Negative Feedback: The Forsaken Nature Available for Re-ranking

Yu Hong, Qing-qing Cai, Song Hua, Jian-min Yao, Qiao-ming Zhu
School of Computer Science and Technology, Soochow University
jyao@suda.edu.cn

ABSTRACT

Re-ranking for Information Retrieval aims to elevate relevant feedbacks and depress negative ones in initial retrieval result list. Compared to relevance feedback-based re-ranking method widely adopted in the literature, this paper proposes a new method to well use three features in known negative feedbacks to identify and depress unknown negative feedbacks. The features include: 1) the minor (lower-weighted) terms in negative feedbacks; 2) hierarchical distance (HD) among feedbacks in a hierarchical clustering tree; 3) obstinateness strength of negative feedbacks. We evaluate the method on the TDT4 corpus, which is made up of news topics and their relevant stories. And experimental results show that our new scheme substantially outperforms its counterparts.

1. INTRODUCTION

When we start out an information retrieval journey on a search engine, the first step is to enter a query in the search box. The query seems to be the most direct reflection of our information needs. However, it is short and often out of standardized syntax and terminology, resulting in a large number of negative feedbacks. Some researches focus on exploring long-term query logs to acquire query intent. This may be helpful for obtaining information relevant to specific interests but not to daily real-time query intents. Especially it is extremely difficult to determine whether the interests and which of them should be involved into certain queries. Therefore, given a query, it is important to "locally" ascertain its intent by using the real-time feedbacks.

Intuitively it is feasible to expand the query using the most relevant feedbacks (Chum et al., 2007). Unfortunately search engines just offer "farraginous" feedbacks (viz. pseudo-feedback) which may involve a great number of negative feedbacks. And these negative feedbacks never honestly lag behind relevant ones in the retrieval results, sometimes far ahead because of their great literal similarity to query. These noisy feedbacks often mislead the process of learning query intent.

For so long, there had no effective approaches to confirm the relevance of feedbacks until the usage of the web click-through data (Joachims et al., 2003). Although the data are sometimes incredible due to different backgrounds and habits of searchers, they are still the most effective way to specify relevant feedbacks. This arouses recent researches about learning to rank based on supervised or semi-supervised machine learning methods, where the click-through data, as the direct reflection of query intent, offer reliable training data to learning the ranking functions.

Although the learning methods achieve substantial improvements in ranking, it can be found that lots of "obstinate" negative feedbacks still permeate retrieval results. Thus an interesting question is why the relevant feedbacks are able to describe what we really need, but weakly repel what we do not need. This may attribute to the inherent characteristics of pseudo-feedback, i.e. their high literal similarity to queries. Thus no matter whether query expansion or learning to rank, they may fall in the predicament that "favoring" relevant feedbacks may result in "favoring" negative ones, and that "hurting" negative feedbacks may result in "hurting" relevant ones.

However, there are indeed some subtle differences between relevant and negative feedbacks, e.g. the minor terms (viz. low-weighted terms in texts). Although these terms are often ignored in

Coling 2008: Poster Volume, pages 436–444,
Beijing, August 2010

relevance measurement because their little effect on mining relevant feedbacks that have the same topic or kernel, they are useful in distinguishing relevant feedbacks from negative ones. As a result, these minor terms provides an opportunity to differentiate the true query intent from its counterpart intents (called "opposite intents" thereafter in this paper). And the "opposite intents" are adopted to depress negative feedbacks without "hurting" the ranks of relevant feedbacks. In addition, hierarchical clustering tree is helpful to establish the natural similarity correlation among information. So this paper adopts the hierarchical distance among feedbacks in the tree to enhance the "opposite intents" based division of relevant and negative feedbacks. Finally, an obstinateness factor is also computed to deal with some obstinate negative feedbacks in the top list of retrieval result list. In fact, Teevan (Teevan et al., 2008) observed that most searchers tend to browse only a few feedbacks in the first one or two result pages. So our method focuses on improving the precision of highly ranked retrieval results.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes our new irrelevance feedback-based re-ranking scheme and the HD measure. Section 4 introduces the experimental settings while Section 5 reports experimental results. Finally, Section 6 draws the conclusion and indicates future work.

2. RELATED WORK

Our work is motivated by information search behaviors, such as eye-tracking and click through (Joachims, 2003). Thereinto, the click-through behavior is most widely used for acquiring query intent. Up to present, several interesting features, such as click frequency and hit time on click graph (Craswell et al., 2007), have been extracted from click-through data to improve search results. However, although effective on query learning, they fail to avoid the thorny problem that even when the typed query and the click-through data are the same, their intents may not be the same for different searchers.

A considerable number of studies have explored pseudo-feedback to learn query intent, thus refining page ranking. However, most of them focus on the relevant feedbacks. It is until recently that negative ones begin to receive some attention. Zhang (Zhang et al., 2009) utilize the irrelevance distribution to estimate the true relevance model. Their work gives the evidence that negative feedbacks are useful in the ranking process. However, their work focuses on generating a better description of query intent to attract relevant information, but ignoring that negative feedbacks have the independent effect on repelling their own kind. That is, if we have a king, we will not refuse a queen. In contrast, Wang (Wang et al., 2008) benefit from the independent effect from the negative feedbacks. Their method represents the opposite of query intent by using negative feedbacks and adopts that to discount the relevance of each pseudo-feedback to a query. However, their work just gives a hybrid representation of opposite intent which may overlap much with the relevance model. Although another work (Wang et al., 2007) of them filters query terms from the opposite intent, such filtering makes little effect because of the sparsity of the query terms in pseudo-feedback.

Other related work includes query expansion, term extraction and text clustering. In fact, query expansion techniques are often the chief beneficiary of click-through data (Chum et al., 2007). However, the query expansion techniques via clicked feedbacks fail to effectively repel negative ones. This impels us to focus on un-clicked feedbacks. Cao (Cao et al., 2008) report the effectiveness of selecting good expansion terms for pseudo-feedback. Their work gives us a hint about the shortcomings of the one-sided usage of high-weighted terms. Lee (Lee et al., 2008) adopt a cluster-based re-sampling method to emphasize the core topic of a query. Their repeatedly feeding process reveals the hierarchical relevance of pseudo-feedback.

3. RE-RANKING SCHEME

3.1 Re-ranking Scheme

The re-ranking scheme, as shown in Figure 1, consists of three components: acquiring negative feedbacks, measuring irrelevance feedbacks and re-ranking pseudo-feedback.

Given a query and its search engine results, we start off the re-ranking process after a trigger point. The point may occur at the time when searchers click on "next page" or any hyperlink.

All feedbacks before the point are assumed to have been seen by searchers. Thus the un-clicked feedbacks before the point will be treated as the known negative feedbacks because they attract no attention of searchers. This may be questioned because searchers often skip some hyperlinks that have the same contents as before, even if the links are relevant to their interests. However, such skip normally reflects the true searching intent because novel relevant feedbacks always have more attractions after all.

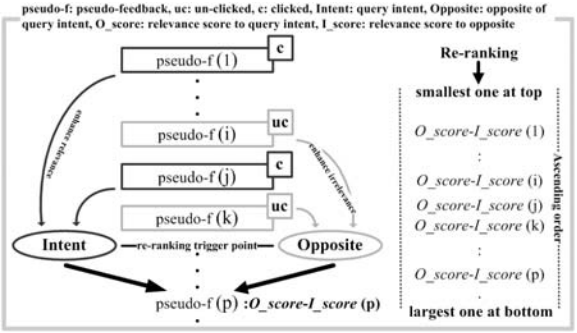


Figure 1. Re-ranking scheme

Another crucial step after the trigger point is to generate the opposite intent by using the known negative feedbacks. But now we temporarily leave the issue to Section 3.2 and assume that we have obtained a good representation of the opposite intent, and meanwhile that of query intent has been composed of the highly weighted terms in the known relevant feedbacks and query terms. Thus, given an unseen pseudo-feedback, we can calculate its overall ranking score predisposed to the opposite intent as follows:

$$R\_score = O\_score - \alpha \cdot I\_score \quad (1)$$

where the *O_score* is the relevance score to the opposite intent, *I_score* is that to the query intent and α is a weighting factor. On the basis, we re-rank the unseen feedbacks in ascending order. That is, the feedback with the largest score appears at the bottom of the ranked list.

It is worthwhile to emphasize that although the overall ranking score, i.e. *R_score*, looks similar to Wang (Wang et al., 2008) who adopts the inversely discounted value (i.e. the relevance score is calculated as $I\_score - \alpha \cdot O\_score$) to re-rank feedbacks in descending order, they are actually quite different because our overall ranking score as shown in Equation (1) is designed to depress negative feedbacks, thereby achieving the similar effect to filtering.

3.2 Representing Opposite Intent

It is necessary for the representation of opposite intent to obey two basic rules: 1) the opposite intent should be much different from the query intent; and 2) it should reflect the independent effect of negative feedbacks.

Given a query, it seems easy to represent its opposite intent by using a vector of high-weighted terms of negative feedbacks. However, the vector is actually a "close relative" of query intent because the terms often have much overlap with that of relevant feedbacks. And the overlapping terms are exactly the source of the highly ranked negative feedbacks. Thus we should throw off the overlapping terms and focus on the rest instead.

In this paper, we propose two simple facilities in representing opposite intent. One is a vector of the weighted terms (except query terms) occurring in the known negative feedbacks, named as $O(-q)$, while another further filters out the high-weighted terms occurring in the known relevant feedbacks, named as $O(-q-r)$. Although $O(-q)$ filters out query terms, the terms are so sparse that they contribute little to opposite intent learning. Thus, we will not explore $O(-q)$ further in this paper (Our preliminary experiments confirm our reasoning). In contrast, $O(-q-r)$ not only differs from the representation of query intent due to its exclusion of query terms but also emphasize the low-weighted terms occurring in negative feedbacks due to exclusion of high-weighted terms occurring in the known relevant feedbacks.

3.3 Employing Opposite Intent

Another key issue in our re-ranking scheme is how to measure the relevance of all the feedbacks to the opposite intent, i.e. *O_score*, thereby the ranking score *R_score*. For simplicity, we only consider Boolean measures in employing opposite intent to calculate the ranking score *R_score*.

Assume that given a query, there are N known relevant feedbacks and $\overline{N}$ known negative ones. First, we adopt query expansion to acquire the representation of query intent. This is done by pouring all terms of the N relevant feedbacks and query terms into a bag of words, where all the occurring weights of each term are

accumulated, and extracting *n* top-weighted terms to represent the query intent as $I(+q+r)$. Then, we use the $\bar{N}$ negative feedbacks to represent the *n*-dimensional opposite intents $O(-q-r)$. For any unseen pseudo-feedback *u*, we also represent it using an *n*-dimensional vector $V(u)$ which contains its *n* top-weighted terms. In all the representation processes, the TFIDF weighting is adopted.

Thus, for an unseen pseudo-feedback *u*, the relevance scores to the query intent and the opposite intent can be measured as:

$$\begin{aligned} I\_score(u) &= B\{\ V(u),\ I(+q+r)\ \} \\ O\_score(u) &= B\{\ V(u),\ O(-q-r)\ \} \end{aligned} \tag{2}$$

where $B\{*,*\}$ indicates Boolean calculation:

$$\begin{aligned} B\{X,Y\} &= \sum b\{x_i,Y\},\ x_i \in X \\ b\{x_i,Y\} &= \begin{cases} 1, & if\ \ x_i \in Y \\ 0, & if\ \ x_i \notin Y \end{cases} \end{aligned} \tag{3}$$

In particular, we simply set the factor α, as mentioned in Equation (1), to 1 so as to balance the effect of query intent and its opposite intent on the overall ranking score. The intuition is that if an unseen pseudo-feedback has more overlapping terms with $O(-q-r)$ than $I(+q+r)$, it will has higher probability of being depressed as an negative feedback.

Two alternatives to the above Boolean measure are to employ the widely-adopted VSM cosine measure and Kullback-Liebler (KL) divergence (Thollard et al., 2000). However, such term-weighting alternatives will seriously eliminate the effect of low-weighted terms, which is core of our negative feedback-based re-ranking scheme.

3.4 Hierarchical Distance (HD) Measure

The proposed method in Section 3.3 ignores two key issues. First, given a query, although search engine has thrown away most opposite intents, it is unavoidable that the pseudo-feedback still involves more than one opposite intent. However, the representation $O(-q-r)$ has the difficulty in highlighting all the opposite intents because the feature fusion of the representation smoothes the independent characteristics of each opposite intent. Second, given several opposite intents, they have different levels of effects on the negative score $O\_score(u)$. And the effects cannot be measured by the unilateral score.

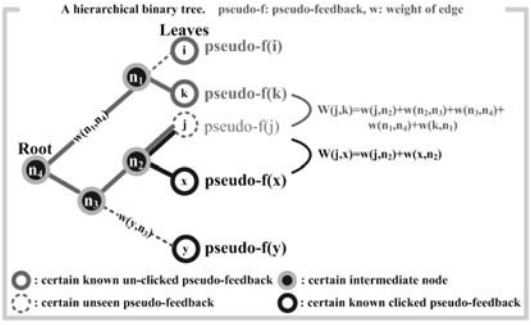


Figure 2. Weighted distance calculation

To solve the issues, we propose a hierarchical distance based negative measure, abbr. HD, which measures the distances among feedbacks in a hierarchical clustering tree, and involves them into hierarchical division of relevance score. Given two random leaves *u* and *v* in the tree, their HD score is calculated as:

$$HD\_score(u,v) = \frac{rel(u,v)}{W(u,v)} \tag{4}$$

where $rel(*,*)$ indicates textual similarity, $W(*,*)$ indicates the weighted distance in the tree, which is calculated as:

$$W(u,v) = \sum_{i \in m} w_i(u,v) \tag{5}$$

where *m* is the total number of the edges between two leaves, $w_i(*,*)$ indicates the weight of the *i-th* edge. In this paper, we adopt CLUTO to generate the hierarchical binary tree, and simply let each $w_i(*,*)$ equal 1. Thus the $W(*,*)$ becomes to be the number of edges *m*, for example, the $W(j,k)$ equals 5 in Figure 2.

On the basis, given an unseen feedback *u*, we can acquire its modified re-ranking score R'_score by following steps. First, we regard each known negative feedback as an opposite intent, following the two generative rules (mentioned in section 3.2) to generate its *n*-dimensional representation $O(-q-r)$. Additionally we represent both the known relevant feedbacks and the unseen feedback *u* as *n*-dimensional term vectors. Second, we cluster these feedbacks to generate a hierarchical binary tree and calculate the HD score for each pair of $(u,*)$, where $*$ denotes a leaf in the tree except *u*. Thus the modified ranking score is calculated as:

$$R'\_score = \sum_{i \in \bar{N}} HDI\_score(u,\bar{v}_i) - \sum_{j \in N} HDI\_score(u,v_j) \tag{6}$$

where $\bar{v}_i$ indicates the *i-th* known negative feedback in the leaves, $\bar{N}$ is the total number of

$\bar{v}$, v_j indicates the *j-th* known relevant feedback, N is the total number of v. Besides, we still adopt Boolean value to measure the textual similarity $rel(*,*)$ in both clustering process and ranking score calculation, thus the HD score in the formula (6) can be calculated as follows:

$$HD\_score(u,v) = \frac{B\{\ V(u),\ V(v)\ \}}{W(u,v)}$$
$$HD\_score(u,\bar{v}) = \frac{O\_score(u)}{W(u,\bar{v})} \qquad (7)$$

3.5 Obstinateness Factor

Additionally we involve an interesting feature, i.e. the obstinate degree, into our re-ranking scheme. The degree is represented by the rank of negative feedbacks in the original retrieval results. That is, the more "topping the list" an negative feedback is, the more obstinate it is.

Therefore we propose a hypothesis that if a feedback is close to the obstinate feedback, it should be obstinate too. Thus given an unseen feedback *u*, its relevance to an opposite intent in HD can be modified as:

$$O\_score(u)' = (1 + \frac{\beta}{rnk}) \cdot O\_score(u) \qquad (8)$$

where rnk indicates the rank of the opposite intent in original retrieval results (Note: in HD, every known negative feedback is an opposite intent), β is a smoothing factor. Because ascending order is used in our re-ranking process, by the weighting coefficient, i.e. $(1 + \beta / rnk)$, the feedback close to the obstinate opposite intents will be further depressed. But the coefficient is not commonly used. In HD, we firstly ascertain the feedback closest to *u*, and if the feedback is known to be negative, set to $\bar{v}_{\max}$, we will use the Equation (8) to punish the pair of (u, $\bar{v}_{\max}$) alone, otherwise without any punishment.

4. EXPERIMENTAL SETTING

4.1 Data Set

We evaluate our methods with two TDT collections: TDT 2002 and TDT 2003. There are 3,085 stories in the TDT 2002 collection are manually labeled as relevant to 40 news topics, 30,736 ones irrelevant to any of the topics. And 3,083 news stories in the TDT 2003 collection are labeled as relevant to another 40 news topics, 15833 ones irrelevant to them. In our evaluation, we adopt TDT 2002 as training set, and TDT 2003 as test set. Besides, only English stories are used, both Mandarin and Arabic ones are replaced by their machine-translated versions (i.e. mttkn2 released by LDC).

Corpus	good	fair	poor
TDT 2002	26	7	7
TDT 2003	22	10	8

Table 1. Number of queries referring to different types of feedbacks (Search engine: Lucene 2.3.2)

In our experiments, we realize a simple search engine based on Lucene 2.3.2 which applies document length to relevance measure on the basis of traditional literal term matching. To emulate the real retrieval process, we extract the title from the interpretation of news topic and regard it as a query, and then we run the search engine on the TDT sets and acquire the first 1000 pseudo-feedback for each query. All feedbacks will be used as the input of our re-ranking process, where the hand-crafted relevant stories default to the clicked feedbacks. By the search engine, we mainly obtain three types of pseudo-feedback: "*good*", "*fair*" and "*poor*", where "*good*" denotes that more than 5 clicked (viz. relevant) feedbacks are in the top 10, "*fair*" denotes more than 2 but less than 5, "*poor*" denotes less than 2. Table 1 shows the number of queries referring to different types of feedbacks.

4.2 Evaluation Measure

We use three evaluation measures in experiments, *P@n*, *NDCG@n* and *MAP*. Thereinto, *P@n* denotes the precision of top *n* feedbacks. On the basis, *NDCG* takes into account the influence of position to precision. *NDCG* at position *n* is calculated as:

$$NDCG@n = \frac{1}{Z_n} \cdot DCG@N = \frac{\sum_{i=1}^{n} \frac{2^{r(u_i)} - 1}{\log(1+i)}}{Z_n} \qquad (9)$$

where i is the position in the result list, Z_n is a normalizing factor and chosen so that for the perfect list *DCG* at each position equals one, and $r(u_i)$ equals 1 when u_i is relevant feedback, else 0. While *MAP* additionally takes into account recall, calculated as:

$$MAP = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{R_i} (\sum_{j=1}^{k} r_i(u_j) \cdot (p@j)_i) \qquad (10)$$

where m is the total number of queries, so *MAP* gives the average measure of precision and recall

for multiple queries, R_i is the total number of feedbacks relevant to query *i*, and *k* is the number of pseudo-feedback to the query. Here *k* is indicated to be 1000, thus *Map* can give the average measure for all positions of result list.

4.3 Systems

We conduct experiments using four main systems, in which the search engine based on Lucene 2.3.2, regarded as the basic retrieval system, provides the pseudo-feedback for the following three re-ranking systems.

Exp-sys: Query is expanded by the first *N* known relevant feedbacks and represented by an *n*-dimensional vector which consists of *n* distinct terms. The standard TFIDF-weighted cosine metric is used to measure the relevance of the unseen pseudo-feedback to query. And the relevance-based descending order is in use.

Wng-sys: A system realizes the work of Wang (Wang et al., 2008), where the known relevant feedbacks are used to represent query intent, and the negative feedbacks are used to generate opposite intent. Thus, the relevance score of a feedback is calculated as $I\_score_{wng} - \alpha_w \cdot O\_score_{wng}$, and the relevance-based descending order is used in re-ranking.

Our-sys: A system is approximately similar to *Wng-sys* except that the relevance is measured by $O\_score_{our} - \alpha \cdot I\_score_{our}$ and the pseudo-feedback is re-ranked in ascending order.

Additionally both *Wng-sys* and *Our-sys* have three versions. We show them in Table 2, where "*I*" corresponds to the generation rule of query intent, "*O*" to that of opposite intent, *Rel*. means relevance measure, *u* is an unseen feedback, *v* is a known relevant feedback, $\bar{v}$ is a known negative feedback.

5. RESULTS

5.1 Main Training Result

We evaluate the systems mainly in two circumstances: when both N and $\bar{N}$ equal 1 and when they equal 5. In the first case, we assume that retrieval capability is measured under given few known feedbacks; in the second, we emulate the first page turning after several feedbacks have been clicked by searchers. Besides, the approximately optimal value of *n* for the *Exp-sys*, which is trained to be 50, is adopted as the global value for all other systems. The training results are shown in Figure 3, where the *Exp-sys* never gains much performance improvement when *n* is greater than 50. In fairness to effects of "*I*" and "*O*" on relevance measure, we also make $\bar{n}$ equal 50. In addition, all the discount factors (viz. α, α_{w2} and α_{w3}) initially equal 1, and the smoothing factor β is trained to be 0.5.

Wng-sys1	***"I"***	*n*-dimensional vector for each *v*, Number of *v* in use is *N*
	"O"	None
	Rel.	$R\_score_{w1} = (\sum_{i=1}^{N} \cos(u,v))/N$
Wng-sys2	***"I"***	Number of *v* in use is *N*, all *v* combine into a *n*-dimensional bag of words b_{w2}
	"O"	Number of $\bar{v}$ in use is *N*, all *v* combine into a *n*-dimensional words bag $\bar{b}_{w2}$
	Rel.	$R\_score_{w2} = \cos(u, b_{w2}) - \alpha_{w2} \cdot \cos(u, \bar{b}_{w2})$
Wng-sys3	***"I"***	Similar generation rules to *Wng-sys2* except that query terms are removed from bag of words b_{w3} and $\bar{b}_{w3}$
	"O"	
	Rel.	$R\_score_{w3} = \cos(u, b_{w3}) - \alpha_{w3} \cdot \cos(u, \bar{b}_{w3})$
Our-sys1	***"I"***	$I(+q+r)$ in section 3.3
	"O"	$O(-q-r)$ in section 3.2
	Rel.	$R\_score = O\_score - \alpha \cdot I\_score$
Our-sys2	***"I"***	The same generation rules to *Our-sys1*
	"O"	
	Rel.	HD algorithm: $R'\_score = \sum_{i \in \bar{N}} HDI\_score(u, \bar{v}_i) - \sum_{j \in N} HDI\_score(u, v_j)$
Our-sys3	***"I"***	The same generation rules to *Our-sys1*
	"O"	
	Rel.	HD algorithm + obstinateness factor: $O\_score(u)' = (1 + \frac{\beta}{rnk}) \cdot O\_score(u)$

Table 2. All versions of both *Wngs* and *Ours*

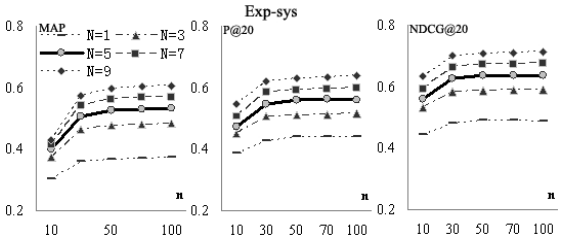


Figure 3. Parameter training of *Exp-sys*

For each query we re-rank all the pseudo-feedback, including that defined as known, so P@20 and NDCG@20 are in use to avoid over-fitting (such as P@10 and NDCG@10 given both N and $\bar{N}$ equal 5). We show the main training results in Table 3, where our methods achieve much better performances than the re-ranking methods based on relevant feedback learning when ***N***= $\bar{N}$ =5. Thereinto, our basic system, i.e. *Our-sys1*, at least achieves approximate 5% improvement on P@20, 3% on NDCG@20 and 1% on MAP than the optimal *wng-sys* (viz. *wng-sys1*). And obvi-

ously the most substantial improvements are contributed by the HD measure which even increases the P@20 of *Our-sys1* by 8.5%, NDCG@20 by 13% and MAP by 9%. But it is slightly disappointing that the obstinateness factor only has little effectiveness on performance improvement, although *Our-sys3* nearly wins the best retrieval results. This may stem from "soft" punishment on obstinateness, that is, for an unseen feedback, only the obstinate companion closest to the feedback is punished in relevance measure.

-	*Our-sys1*	*Our-sys2*	*Exp-sys*	*Wng-sys1*	*Basic*
P@20	0.6603	0.8141	0.63125	0.7051	0.6588
NDCG@20	0.7614	0.8587	0.8080	0.7797	0.6944
MAP	0.6583	0.7928	0.5955	0.7010	0.6440

Table 3. Main training results

It is undeniable that all the re-ranking systems work worse than the basic search engine when the known feedbacks are rare, such as $N = \overline{N} = 1$. This motivates an additional test on the higher values of both N and $\overline{N}$ ($N = \overline{N} = 9$), as shown in Table 4. Thus it can be found that most of the re-ranking systems achieve much better performance than the basic search engine. An important reason for this is that more key terms can be involved into representations of both query intent and its opposite intent. So it seems that more manual intervention is always reliable. However in practice, seldom searchers are willing to use an unresponsive search engine that can only offer relatively satisfactory feedbacks after lots of click-through and page turning. And in fact at least two pages (if one page includes 10 pseudo-feedback) need to be turned in the training corpus when both N and $\overline{N}$ equal 9. So we just regard the improvements benefiting from high click-through rate as an ideal status, and still adopt the practical numerical value of N and $\overline{N}$, i.e. $N = \overline{N} = 5$, to run following test.

5.2 Constraint from Query

A surprising result is that *Exp-sys* always achieves the worst MAP value, even worse than the basic search engine even if high value of *N* is in use, such as the performance when *N* equal 9 in Table 4. It seems to be difficult to question the reasonability of the system because it always selects the most key terms to represent query intent by query expansion. But an obvious difference between *Exp-sys* and other re-ranking systems could explain the result. That is the query terms consistently involved in query representation by *Exp-sys*.

systems	$N = \overline{N}$	*P@20*	*NDCG@20*	*MAP*	*Factor*
Basic	-	**0.6588**	**0.6944**	**0.6440**	-
Exp-sys	1	0.4388	0.4887	0.3683	-
	5	**0.5613**	**0.6365**	**0.5259**	-
Wng-sys1	1	0.5653	0.6184	0.5253	-
	5	**0.6564**	**0.7361**	**0.6506**	-
Wng-sys2	1	0.5436	0.6473	0.4970	α_{w2} =1
	5	**0.5910**	**0.7214**	**0.5642**	α_{w2} =1
Wng-sys3	1	0.5436	0.6162	0.4970	α_{w3} =1
	5	**0.5910**	**0.6720**	**0.5642**	α_{w3} =1
Our-sys1	1	0.5628	0.6358	0.4812	α =1
	5	**0.7031**	**0.7640**	**0.6603**	α =1
Our-sys2	1	0.6474	0.6761	0.5967	α =1
	5	**0.7885**	**0.8381**	**0.7499**	α =1
Our-sys3	1	0.6026	0.6749	0.5272	β =0.5
	5	**0.7897**	**0.8388**	**0.7464**	β =0.5

Table 4. Effects of N and $\overline{N}$ on re-ranking performance (when $N = \overline{N} = 9$, $n = \overline{n} = 50$)

In fact, *Wng-sys1* never overly favor the query terms because they are not always the main body of an independent feedback, and our systems even remove the query terms from the opposite intent directly. Conversely *Exp-sys* continuously enhances the weights of query terms which result in over-fitting and bias. The visible evidence for this is shown in Figure 4, where *Exp-sys* achieves better Precision and NDCG than the basic search engine at the top of result list but worse at the subsequent parts. The results illustrate that too much emphasis placed on query terms in query expansion is only of benefit to elevating the originally high-ranked relevant feedback but powerless to pull the straggler out of the bottom of result list.

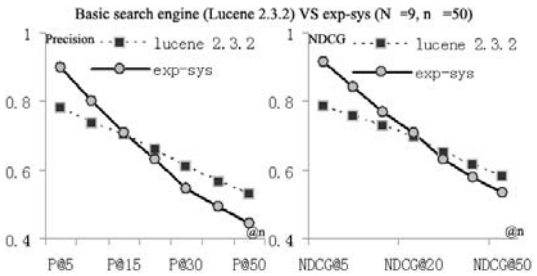


Figure 4. MAP comparison (basic vs *Exp*)

5.3 Positive Discount Loss

Obviously Wang (Wang et al., 2008) has noticed the negative effects of query terms on re-ranking. Therefore his work (reproduced by *Wng-sys1*, *2*, *3* in this paper) avoids arbitrarily enhancing the terms in query representation, even removes them as *Wng-sys3*. This indeed contributes to the

improvement of the re-ranking system, such as the better performances of *Wng-sys1, 2, 3* shown in Table 3, although *Wng-sys3* has no further improvement than *Wng-sys2* because of the sparsity of query terms. On the basis, the work regards the terms in negative feedbacks as noises and reduces their effects on relevance measure as much as possible. This should be a reasonable scheme, but interestingly it does not work well in our experiments. For example, although *Wng-sys2* and *Wng-sys3* eliminate the relevance score calculated by using the terms in negative feedbacks, they perform worse than *Wng-sys1* which never make any discount.

systems	$\alpha_* = 0.5$	$\alpha_* = 1$	$\alpha_* = 2$
Our-sys1	0.4751	0.6603	0.6901
Wng-sys2	0.6030	0.5642	0.4739
Wng-sys3	0.6084	0.5642	0.4739

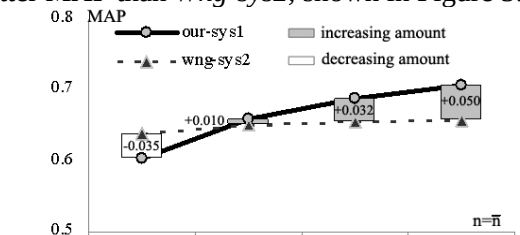
Table 5. Effects on MAP

Additionally when we increase the discount factor α_{w2} and α_{w3}, as shown in Table 5, the performances (MAP) of *Wng-sys2* and *Wng-sys3* further decrease. This illustrates that the high-weighted terms of high-ranked negative feedbacks are actually not noises. Otherwise why do the feedbacks have high textual similarity to query and even to their neighbor relevant feedbacks? Thus it actually hurts real relevance to discount the effect of the terms.

Conversely *Our-sys1* can achieve further improvement when the discount factor α increases, as shown in Table 5. It is because the discount contributes to highlighting minor terms of negative feedbacks, and these terms always have little overlap with the kernel of relevant feedbacks. Additionally the minor terms are used to generate the main body of opposite intent in our systems, thus the discount can effectively separate opposite intent from positive query representation. Thereby we can use relatively pure representation of opposite intent to detect and repel subsequent negative feedbacks.

5.4 Availability of Minor Terms

Intuitively we can involve more terms into query representation to alleviate the positive discount loss. But it does not work in practice. For example, *Wng-sys2* shown in Figure 5 has no obvious improvement no matter how many terms are included in query representation. Conversely *Our-sys1* can achieve much more improvement when it involves more terms into the opposite intent. For example, when the number of terms increases to 150, *Our-sys1* has approximately 5% better MAP than *Wng-sys2*, shown in Figure 5.



	30	50	100	150
our-sys1	0.6034	0.6603	0.6873	0.706
wng-sys2	0.6386	0.6506	0.6556	0.6563

Figure 5. Effects on MAP in modifying the dimensionality n (when $N=\overline{N}=5$, $\alpha=1$)

This result illustrates that minor terms are available for repelling negative feedbacks, but too weak to recall relevant feedbacks. In fact, the minor terms are just the low-weighted terms in text. Current text representation techniques often ignore them because of their marginality. However minor terms can reflect fine distinctions among feedbacks, even if they have the same topic. And the distinctions are of great importance when we determine why searchers say "Yes" to some feedbacks but "No" to others.

systems	*metric*	*good*	*fair*	*poor*	*global*	*Factor*
Wng-sys1	**P@20**	0.7682	0.5450	0.2643	0.6205	-
	NDCG@20	0.8260	0.6437	0.4073	0.7041	
	MAP	**0.6634**	**0.4541**	**0.9549**	**0.6620**	
Our-sys1	**P@20**	0.8273	0.5700	0.2643	0.6603	$\alpha=2$, $\beta=0.5$
	NDCG@20	0.8679	0.6620	0.4017	0.7314	
	MAP	**0.6740**	**0.4573**	**0.9184**	**0.6623**	
Our-sys2	**P@20**	0.8523	0.7600	0.2714	0.7244	$\alpha=2$, $\beta=0.5$
	NDCG@20	0.8937	0.8199	0.4180	0.7894	
	MAP	**0.7148**	**0.6313**	**0.9897**	**0.7427**	
Our-sys3	**P@20**	0.8523	0.7600	0.2714	0.7244	$\alpha=2$, $\beta=0.5$
	NDCG@20	0.8937	0.8200	0.4180	0.7894	
	MAP	**0.7145**	**0.6292**	**0.9897**	**0.7420**	

Table 6. Main test results

5.5 Test Result

We run all systems on test corpus, i.e. TDT2003, but only report four main systems: *Wng-sys1*, *Our-sys1*, *Our-sys2* and *Our-sys3*. Other systems are omitted because of their poor performances. The test results are shown in Table 6 which includes not only global performances for all test queries but also local ones on three distinct types of queries, i.e. "good", "fair" and "poor". Thereinto, *Our-sys2* achieves the best performance around all types of queries. So it is believable

that hierarchical distance of clustering tree always plays an active role in distinguishing negative feedbacks from relevant ones. But it is surprising that *Our-sys3* achieves little worse performance than *Our-sys2*. This illustrates poor robustness of obstinateness factor.

Interestingly, the four systems all achieve very high MAP scores but low P@20 and NDCG@20 for "poor" queries. This is because the queries have inherently sparse relevant feedbacks: less than 6‰ averagely. Thus the highest p@20 is only approximate 0.3, i.e. 6/20. And the low NDCG@20 is in the same way. Besides, all MAP scores for "fair" queries are the worst. We find that this type of query involves more macroscopic features which results in more kernels of negative feedbacks. Although we can solve the issue by increasing the dimensionality of opposite intent, it undoubtedly impairs the efficiency of re-ranking.

6. CONCLUSION

This paper proposes a new re-ranking scheme to well explore the opposite intent. In particular, a hierarchical distance-based (HD) measure is proposed to differentiate the opposite intent from the true query intent so as to repel negative feedbacks. Experiments show substantial outperformance of our methods.

Although our scheme has been proven effective in most cases, it fails on macroscopic queries. In fact, the key difficulty of this issue lies in how to ascertain the focal query intent given various kernels in pseudo-feedback. Fortunately, click-through data provide some useful information for learning real query intent. Although it seems feasible to generate focal intent representation by using overlapping terms in clicked feedbacks, such representation is just a reproduction of macroscopic query since the overlapping terms can only reflect common topic instead of focal intent. Therefore, it is important to segment clicked feedbacks into different blocks, and ascertain the block of greatest interest to searchers.

References

Allan, J., Lavrenko, V., and Nallapati, R. 2002. UMass at TDT 2002, Topic Detection and Tracking: Workshop.

Craswell, N., and Szummer, M. Random walks on the click graph. 2007. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '30. ACM Press, New York, NY, 239-246.

Cao, G. H., Nie, J. Y., and Gao, J. F. 2008. Stephen Robertson. Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 243-250.

Chum, O., Philbin, J., Sivic, J., and Zisserman, A. 2007. Automatic query expansion with a generative feature model for object retrieval. In Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 1–8.

Joachims, T., Granka, L., and Pan, B. 2003. Accurately Interpreting Clickthrough Data as Implicit Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '28. New York, NY, 154-161.

Lee, K. S., Croft, W. B., and Allan, J. 2008 A Cluster-Based Resampling Method for Pseudo-Relevance Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 235-242.

Thollard, F., Dupont, P., and Higuera, L.2000. Probabilistic DFA Inference Using Kullback-Leibler Divergence and Minimality. In Proceedings of the 17th Int'l Conf on Machine Learning. San Francisco: Morgan Kaufmann, 975-982.

Teevan, J. T., Dumais, S. T., and Liebling, D. J. 2008. To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. New York, NY, 163-170.

Wang, X. H., Fang, H., and Zhai, C. X. 2008. A Study of Methods for Negative Relevance Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 219-226.

Wang, X. H., Fang, H., and Zhai, C. X. 2007. Improve retrieval accuracy for difficult queries using negative feedback. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM press, New York, NY, USA, 991-994.

Zhang, P., Hou, Y. X., and Song, D. 2009. Approximating True Relevance Distribution from a Mixture Model based on Irrelevance Data. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 107-114.