Relation Based Term Weighting Regularization

Hao Wu and Hui Fang

Department of Electrical and Computer Engineering
University of Delaware, USA
{haow,hfang}@ece.udel.edu

Abstract. Traditional retrieval models compute term weights based on only the information related to individual terms such as TF and IDF. However, query terms are related. Intuitively, these relations could provide useful information about the importance of a term in the context of other query terms. For example, query "perl tutorial" specifies that a user look for information relevant to both perl and tutorial. Thus, a document containing both terms should have higher relevance score than the ones with only one of them. However, if the IDF value of "tutorial" is much smaller than "perl", existing retrieval models may assign the document lower score than those containing multiple occurrences of "perl". It is clear that the importance of a term should be dependent on not only collection statistics but also the relations with other query terms. In this work, we study how to utilize semantic relations among query terms to regularize term weighting. Experiment results over TREC collections show that the proposed strategy is effective to improve the retrieval performance.

1 Introduction

Developing effective retrieval functions is always a challenging yet important IR problem. The performance of a retrieval function is closely related to its term weighting strategies [2]. Most commonly used term weighting strategies, such as TF and IDF, utilize only the information of individual query terms. For instance, the IDF value of a term is computed based on the number of documents containing the term and the total number of the documents in the collection. As a result, the IDF value of a term would be the same for all the queries given a document collection. However, since query terms are related, the importance of a term should be regularized based on the relation between the term and other query terms. Let us consider a two-term query "perl tutorial". The relations between these two terms are *conjunctive*, i.e., these terms are used to specify two different concepts of the query, so relevant documents are expected to contain both terms. However, existing retrieval functions may assign higher scores to the documents containing only the term with higher IDF value. On the contrary, the relations of query terms could also be *disjunctive*, e.g., "stolen or lost" art. In these cases, relevant documents do not have to include all terms.

R. Baeza-Yates et al. (Eds.): ECIR 2012, LNCS 7224, pp. 109–120, 2012.

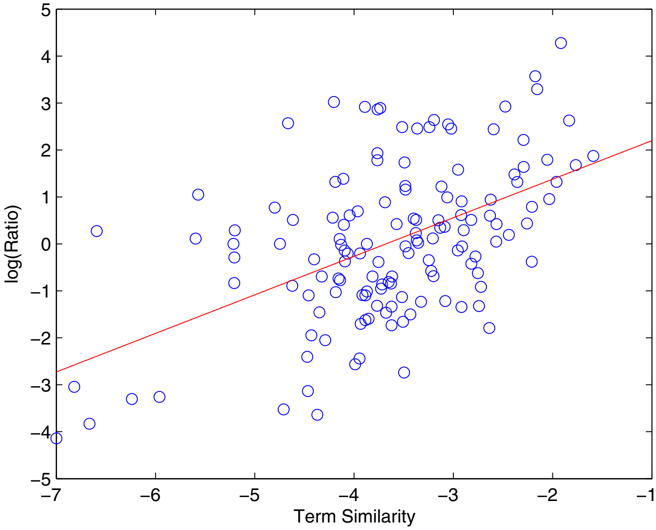

Fig. 1. Plot of $log(\frac{\#Rel\_Docs\_with\_both\_terms}{\#Rel\_Docs\_with\_single\_term})$ vs. $s(t_1, t_2)$

Clearly, the importance of a term should depend on not only term statistics in a collection but also the relations among query terms.

In this paper, we study the problem of regularizing term weights based on conjunctive/disjunctive relations among query terms by exploiting constraint analysis and exploratory data analysis methods. Specifically, we first define a constraint based on the conjunctive/disjunctive relations among query terms, and then analyze existing retrieval functions with the constraint. We find that all the analyzed functions satisfy the constraint conditionally. Guided by the constraint analysis, we then propose a term regularization method that can adjust the IDF value of a term based on its relation with other terms. Empirical results over eight TREC collections show that the proposed method is effective to improve the performance for four of the state of the art retrieval functions over all the collections.

2 Term Weighting Regularization

We now study how to utilize the relation among terms in a query to regularize term weighting. The main idea is to exploit both data exploratory analysis [6,7] and constraint analysis [2], which can provide guidance on how to implement the term regularization method.

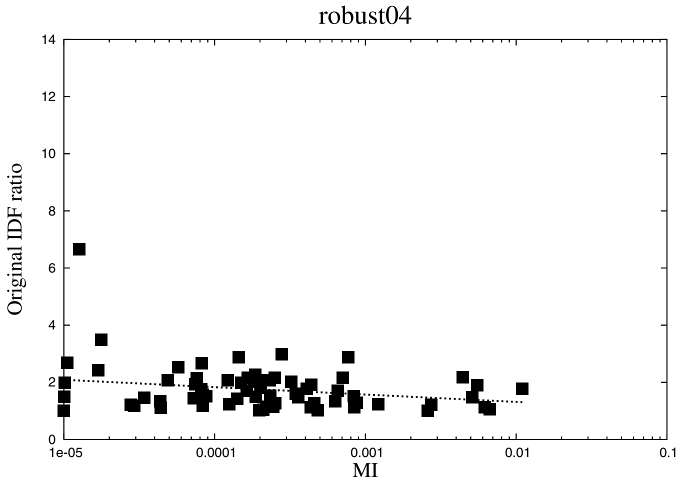

Fig. 2. Plot of the original IDF ratio vs. term similarity (computed using Equation (1))

2.1 Semantic Relation among Query Terms

Let us start with a simple case where a query has only two terms. The relations between terms in the query could be different. For example, some queries require both terms to occur in relevant documents (i.e, AND relation or conjunctive semantics), while others may require at least one query term to occur in relevant documents (i.e., OR relation or disjunctive semantics). Term co-occurrences often indicate the semantic similarity among terms, so a commonly used term semantic similarity function is based on EMIM [17,4] shown as follows:

$$s(t_1, t_2) = \sum_{X_{t_1}, X_{t_2} \in \{0,1\}} p(X_{t_1}, X_{t_2}) \log \frac{p(X_{t_1}, X_{t_2})}{p(X_{t_1}) p(X_{t_2})}. \qquad (1)$$

$s(t_1, t_2)$ is the similarity between the two terms, and X_t is a binary random variable corresponding to the presence/absence of term t in each document.

Intuitively, query terms tend to have AND relation when their semantic similarity is high, and query terms tend to have OR relation when they are not very semantically related. Thus, we hypothesize that, as the semantic similarity between a term pair increases, the number of relevant documents with both query terms increases while the number of relevant documents with only a single query term decreases.

To verify the hypothesis, we conduct data exploratory analysis using TREC data sets. In particular, we first choose all the two-term queries from a data set. For every query, we compute the ratio of the number of relevant document with both query terms to the number of relevant documents with only a single term, and then compute the term semantic similarity using Equation (1). Finally,

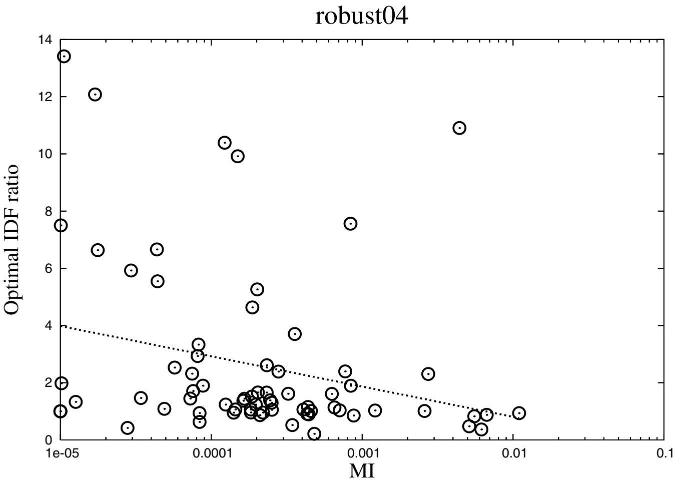

Fig. 3. Plot of the optimal IDF ratio vs. term similarity

we plot the graph for these two variables for every term-term query. Figure 1 shows the results for the data set used in TREC 2004 Robust track [18]. The plots on other data sets are similar. It is clear that the plots are consistent with our hypothesis, i.e., as the value of semantic similarity decreases, the number of relevant documents with both query term increases while the number of relevant documents with only a single query term decreases. Note that we also plot the graphs to observe the relations between the number of relevant documents with both terms vs. term similarity as well as the relations between the number of relevant documents with only a single term vs. term similarity. We do not show these graphs due to the limited space. However, all of these graphs are consistent with our hypothesis. In fact, the hypothesis can be easily generalized to the cases with more than two terms.

2.2 Term Relation Based Analysis

With the verified hypothesis, we now discuss how to adjust term weighting based on term semantic relations. We first define a constraint based on term relations, and then discuss how to utilize the results of constraint analysis to regularize the term weighting.

AND Relation Constraint: Let $Q = \{q_1, q_2\}$ be a query with two terms, where $td(q_1) \geq td(q_2)$ and $td(t)$ is the term discriminative value of t such as IDF. Let D_1 and D_2 be two documents, where $c(q_1, D_1) = 1$, $c(q_2, D_1) = 1$, $c(q_1, D_2) = c$ and $c \geq 1$. Assuming that the user issuing the query expects that a relevant document contains both terms, thus, we have $S(Q, D_1) > S(Q, D_2)$.

The constraint says that, if the terms in a query has an AND relation, we require the documents with both query terms be ranked higher than those with only one query term. Specifically, given a two-term query, if D_1 contains both terms and both terms only occur once and D_2 contains only the term with the higher IDF value, D_1 should always have higher relevance score than D_2 no matter how many times the term with higher IDF occur.

We then analyze four representative retrieval functions with the constraint. The functions are *pivoted normalization function* derived from vector space models [13,14]), *Okapi BM25* derived from classical probabilistic models [16,5,12], *Dirichlet prior* derived from language models [11,19] and *F2-EXP* derived from axiomatic retrieval models [3]. The retrieval functions are shown as follows.

- Pivoted Normalization:

$$S(Q,D) = \sum_{t \in D \cap Q} \frac{1 + \log(1 + \log(c(t,D)))}{1 - s + s\frac{|D|}{avdl}} \times c(t,Q) \times \log \frac{N+1}{df(t)}$$

- Okapi BM25:

$$\begin{aligned} S(Q,D) = \sum_{t \in Q \cap D} \log \frac{N - df(t) + 0.5}{df(t) + 0.5} \times \frac{(k_3 + 1) \times c(t,Q)}{k_3 + c(t,Q)} \\ \times \frac{(k_1 + 1) \times c(t,D)}{k_1((1-b) + b\frac{|D|}{avdl}) + c(t,D)} \end{aligned} \tag{2}$$

- Dirichlet Prior:

$$S(Q,D) = \sum_{t \in Q \cap D} c(t,Q) \times \log(1 + \frac{c(t,D)}{\mu \times p(t|C)}) + \log \frac{\mu}{|D| + \mu} \tag{3}$$

- Axiomatic:

$$S(Q,D) = \sum_{t \in Q \cap D} c(t,Q) \times (\frac{N}{df(t)})^{0.35} \times \frac{c(t,D)}{c(t,D) + b + \frac{b \times |D|}{avdl}} \tag{4}$$

$c(t,D)$ is the count of term t in document D, $c(t,Q)$ is the count of term t in query Q, $df(t)$ is the number of documents with term t, $|D|$ is the length of document d, $avdl$ is the average document length of the document collection, and $p(t|C)$ is the probability of term t in collection C. k_1 is set to 1.2 and k_3 is set to 1000. μ, b, s are parameters in the original retrieval functions.

The constraint analysis results show that none of the functions satisfies the constraint conditionally. In order to satisfy the constraint, the term discriminative ratio between the two terms needs to be smaller than a certain threshold as follows:

- Pivoted:
$$\frac{td(q_1)}{td(q_2)} < \frac{LN(c)}{LN(2)(1+\log(1+\log(c))) - LN(c)},$$
where $LN(x) = 1 - s + s\frac{x}{avdl}$ and s is the retrieval parameter in Pivoted.
- Okapi:
$$\frac{td(q_1)}{td(q_2)} < \frac{LN(c)+c}{c \cdot LN(2) - LN(c)},$$
where $LN(x) = 1 - b + b\frac{x}{avdl}$ and b is the retrieval parameter in Okapi.
- Dirichlet:
$$\frac{td(q_1)}{td(q_2)} < \frac{(\mu y(\mu y+1)(c+\mu)^2 - (\mu y)^2(2+\mu)^2}{c \cdot \mu y(2+\mu)^2 - (\mu y+1)(c+\mu)^2},$$
where $y = p(q_1|C)$.
- Axiomatic:
$$\frac{td(q_1)}{td(q_2)} < \frac{LN_a(c)+c}{c \cdot LN_a(2) - LN_a(c)},$$
where $LN_a(x) = b + b\frac{x}{avdl}$ and b is the retrieval parameter in Okapi.

Since the AND relation indicates that the two terms have higher semantic similarity, it means that the ratio of the discrimination values needs to be smaller when the semantic similarity is higher. In an extreme case, when two terms have the highest semantic similarity, i.e., they always co-occur in the collection, the original term discrimination values do not matter at all.

Furthermore, we conduct another set of data exploratory analysis to validate the previous constraint. For a data set, we select all queries with only two query terms. We use the traditional IDF, i.e., $log\frac{N+1}{df(t)}$ as the term discrimination value for term t. For every query, we compute the original IDF ratio among terms as well as the optimal IDF ratio among terms. The original IDF ratio is computed using collection statistics, while the optimal IDF ratio is obtained by changing the IDF ratio arbitrarily and recording the values yielding to the best retrieval performance. We then plot the relations between original IDF ratio vs. term similarity as well as the relations between optimal IDF ratio vs. term similarity. Figure 2 and 3 show the plots on ROBUST04 data set, and the curves on other data sets are similar. Clearly, the original IDF ratio is more uniformly distributed, while the optimal IDF ratio increases as the term similarity decreases. The observation is consistent with the constraint analysis results.

To summarize, when the semantic similarities among terms are large, we should emphasize all the terms by making the ratio of term discrimination values smaller; when the similarity decreases, we should put more trust on the terms with higher term discrimination values by making the ratio larger.

2.3 Term Weighting Regularization Function

Following the spirit of axiomatic approaches [3], we can modify the retrieval functions to make them satisfy the proposed constraint. The assumption is a retrieval function would perform better when it satisfies more reasonable constraints. We propose the following modification for the term discrimination part of each function to make them depend on semantic relations between a query term and others:

$$td_{reg}(q) = td_{old}(q)(1 + \beta \cdot \frac{TD_{old}(Q)}{td_{old}(q)}^{sgn(sim(q,Q)-\delta)}) \tag{5}$$

where $td_{old}(q)$ is existing implementation of term discrimination value of q in a retrieval function, and $td_{reg}(q)$ denotes the proposed regularized term discrimination part for the corresponding retrieval function. $TD_{old}(Q) = \frac{\sum_{t \in Q} td_{old}(t)}{|Q|}$ measures the important of query Q, which is computed as the average term discrimination of all the query terms. $s(q, Q)$ denotes the similarity between q and other terms in Q and is computed as $sim(q, Q) = \frac{\sum_{t \in Q - \{q\}} s(q,t)}{|Q-1|}$, where $s(q, t)$ is computed as shown in Equation (1). $sgn(x) \in \{0, 1\}$ is a sign function defined as $sgn(x) = 1$ if $x > 0$, and $sgn(x) = -1$ if $x \leq 0$.

The main idea of the proposed modification is to diminish the effect of term discrimination values if a term has AND relation with other query terms. There are two parameters in Equation (5).

- β controls how much we trust the regularized term weighting. When β is set to 0, we do not regularize the term weighting.
- δ is a threshold used to determine when two terms are considered to have AND relation based on their semantic similarities. For example, when $sim(q, D)$ is larger than δ, we assume that q has AND relation with other terms.

The proposed function regularize the term weighting in the following way. When the terms have AND relation and β is set to a very large value, all the term weights would be the same, i.e., the average IDF of query terms. This could avoid the cases where documents containing only terms with higher IDF values are favored. When they have OR relation, the differences of term weights become larger.

Finally, we analyze the modified retrieval functions with the regularized term discrimination part as shown in Equation (5) with the proposed constraint, and find that the modified retrieval functions would satisfy the constraint unconditionally. Note that the proposed method is only one possible way of solving the problem. We plan to explore other options in our future work.

3 Experiments

3.1 Experiment Design

We conduct experiments over eight representative TREC data sets, which include news articles, technical reports, government documents, Web data and ad hoc data as follows.

- *Robust04:* the collection used in TREC Robust 2004 track;
- *Robust05:* the collection used in TREC Robust 2005 track;
- *Web:* the collection used in the Web track of TREC8;
- *Trec8:* the collection used for the ad hoc retrieval track of TREC8;
- *Trec7:* the collection used for the ad hoc retrieval track of TREC7;
- *Ap88-89:* AP news articles with a set of TREC queries;
- *Fr88-89:* government documents with a set of TREC queries;
- *Doe:* technical reports with a set of TREC queries.

The performance is measured in terms of MAP (mean average precision). *BL* denotes the original retrieval function without regularization, and *Reg* denotes the proposed regularization method. We incorporate the proposed methods into four representative retrieval functions, i.e., Pivoted normalization (**Piv.**), Okapi BM25 (**Okapi**), Dirichlet Prior (**Dir.**) and axiomatic retrieval function F2-EXP (**AX**).

Table 1. Optimal performance comparison for keyword queries (MAP) (‡ and † indicate that the improvement is statistically significant according to the Wilcoxin signed rank test at the level of 0.05 and 0.1 respectively)

Function		**Robust04**	**Robust05**	**Trec7**	**Trec8**	**Web**	**Ap88-89**	**Doe**	**Fr88-89**
Piv.	BL	0.2406	0.1999	0.1762	0.2438	0.2883	0.2267	0.1788	0.2183
	Reg_{Opt}	**0.2423**	**0.2007†**	**0.1812**	**0.2470†**	**0.2919**	**0.2270**	**0.1817†**	**0.2270**
Okapi	BL	0.2477	0.2013	0.1857	0.2512	0.3105	0.2255	0.1847	0.2247
	Reg_{Opt}	**0.2508‡**	**0.2027**	**0.1881**	**0.2552‡**	**0.3141‡**	**0.2274**	**0.1890†**	**0.2321†**
Dir.	BL	0.2504	0.1957	0.1860	0.2567	0.3024	0.2216	0.1803	0.2022
	Reg_{Opt}	**0.2523‡**	**0.1961**	**0.1884‡**	**0.2575**	**0.3052**	**0.2219**	**0.1811**	**0.2036**
AX	BL	0.2505	0.1932	0.1872	0.2537	0.2878	0.2250	0.1747	0.2168
	Reg_{Opt}	**0.2523‡**	**0.1935**	0.1872	**0.2570‡**	**0.2956‡**	**0.2255**	**0.1812‡**	**0.2191**

Table 2. Optimal performance comparison for description-only queries (MAP)(‡ and † indicate that the improvement is statistically significant according to the Wilcoxin signed rank test at the level of 0.05 and 0.1 respectively)

Function		**Robust04**	**Robust05**	**Trec7**	**Trec8**	**Web**	**Ap88-89**	**Doe**	**Fr88-89**
Piv.	BL	0.2145	0.1406	0.1461	0.2032	0.2122	0.1931	0.1031	0.1424
	Reg_{Opt}	**0.2411‡**	**0.1488†**	**0.1803‡**	**0.2364‡**	**0.2354†**	**0.2151‡**	**0.1269‡**	**0.1683‡**
Okapi	BL	0.2114	0.1391	0.1527	0.2014	0.2371	0.1812	0.1037	0.1526
	Reg_{Opt}	**0.2460‡**	**0.1489**	**0.1866‡**	**0.2432‡**	**0.2665‡**	**0.2034‡**	**0.1249**	**0.1854‡**
Dir.	BL	0.2326	0.1598	0.1811	0.2279	0.2693	0.1990	0.1253	0.1522
	Reg_{Opt}	**0.2496‡**	**0.1619**	**0.1948‡**	**0.2434‡**	**0.2818**	**0.2007**	**0.1283**	**0.1635‡**
AX	BL	0.2421	0.1612	0.1864	0.2357	0.2715	0.2016	0.1161	0.1674
	Reg_{Opt}	**0.2463‡**	0.1612	**0.1903**	**0.2406‡**	**0.2748**	**0.2118‡**	**0.1236‡**	**0.1885‡**

3.2 Performance Comparison

We first compare the optimal performance of the proposed methods. For each collection, we use two types of queries: keyword-only and descrption-only queries. Table 2 shows the optimal performance of different functions with relation-based regularization for descrption-only queries, and Table 1 shows the optimal performance for keyword-only queries. Note that Reg_{Opt} denotes the optimal performance with the proposed term weighting regularization function.

The proposed method can improve the performance for keyword-only queries, but the improvement is not consistent and sometimes not statistically significant. It is clear that the proposed term regularization methods are more effective on description-only queries, where these methods can significantly and consistently improve the retrieval performance for all four retrieval functions over almost all of the eight data collections. Another interesting observation is that the improvement is not consistent across different retrieval functions. The improvement is more substantial for pivoted normalization and Okapi functions.

There are two parameters in the proposed function. We used Fr88-89 as a training collection to train the parameters. The optimal value of β is 1, and that of δ is 0.001. We then plug these values in the function and report the performance for description only queries on other data collections in Table 3. Note that $Reg_{trained}$ denotes the performance when we set the parameter to the trained values. The results show that, with the trained parameter values, the proposed methods can still improve the retrieval performance for description-only queries.

Table 3. Performance comparison with trained parameter for description-only queries (MAP) (‡ and † indicate that the improvement is statistically significant according to the Wilcoxin signed rank test at the level of 0.05 and 0.1 respectively)

Function		**Robust04**	**Robust05**	**Trec7**	**Trec8**	**Web**	**Ap88-89**	**Doe**
Piv.	BL	0.2145	0.1406	0.1461	0.2032	0.2122	0.1931	0.1031
	$Reg_{trained}$	**0.2411‡**	**0.1486 †**	**0.1803‡**	**0.2364‡**	**0.2343†**	**0.2151‡**	**0.1252 ‡**
Okapi	BL	0.2114	0.1391	0.1527	0.2014	0.2371	0.1812	0.1037
	$Reg_{trained}$	**0.2458‡**	**0.1469**	**0.1866‡**	**0.2418 †**	**0.2615 †**	**0.2034‡**	**0.1249**
Dir.	BL	0.2326	0.1598	0.1811	0.2279	0.2693	0.1990	0.1253
	$Reg_{trained}$	**0.2496‡**	**0.1619**	**0.1948‡**	**0.2434‡**	**0.2818**	**0.2000**	**0.1276**
AX	BL	0.2421	0.1612	0.1864	0.2357	0.2715	0.2016	0.1161
	$Reg_{trained}$	**0.2448 †**	0.1612	**0.1873**	**0.2382**	0.2693	**0.2092†**	**0.1223‡**

3.3 Parameter Sensitivity

We examine the parameter sensitivity for the two parameters in the proposed method. Figure 4 shows the sensitivity curve for the parameter β on Robust04. We observe that the performance is not very sensitive when β is close to 1.

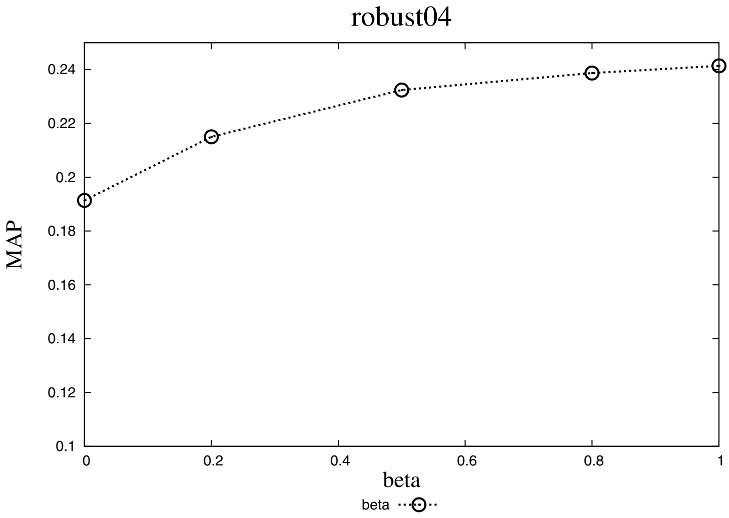

Fig. 4. Performance Sensitivity: β

We only show results on one data set due to the space limit. However, the results are similar for other data sets. Similar to the observation based on Table 3, setting $\beta = 1$ in Equation (5) would lead to good performance.

Figure 5 shows the parameter sensitivity for δ. It is clear that the performance is not very sensitive to the parameter when δ is smaller than 0.0001. The value depends on the collection statistics, and we plan to automatically learn the value in our future work.

4 Related Work

The most commonly used term weighting strategies, such as TF-IDF, are based on only the statistic information of individual terms. The semantic relations among terms are often ignored. There have been quite a few recent efforts that exploit the relations among query terms to improve search quality. A few studies focused on using phrases in retrieval models [1,10,8]. Metzler and Croft [9] proposed a term dependence model based on language modeling approaches. Tao and Zhai [15] exploited term proximity in retrieval models. Zheng and Fang [20] focused on utilizing the relations among different query aspects to regularize the term weighting.

Although the motivation is similar, our work differs from existing studies in that (1) we aim to study the feasibility of utilizing conjunctive/disjunctive relations among query terms in term weighting regularization; and (2) the proposed methods are quite general and can be combined with almost all existing retrieval models.

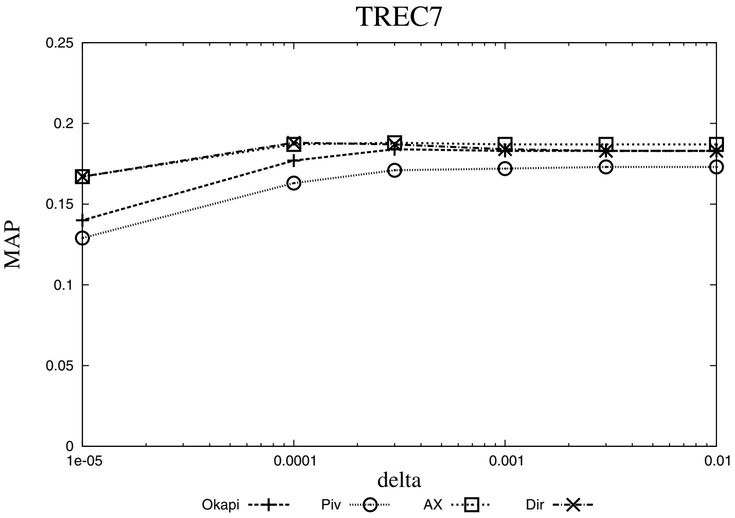

Fig. 5. Performance Sensitivity: δ

5 Conclusions

We propose a new method that can regularize the term weighting based on the semantic relation among query terms. Specifically, we propose a method that can adjust the IDF value of a term based on its relation with other terms. Empirical results over eight TREC collections show that the proposed method is effective to improve the performance for two existing retrieval functions over all the collections.

There are many interesting directions for the future work. First, we plan to study more sophisticated methods to derive new regularization methods based on the proposed constraints. Second, we plan to study more reasonable retrieval constraints based on term semantic relations. Finally, it would be interesting to study how to combine the proposed relation based term weighting with existing inter-aspect term weighting to further improve the performance.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant Number IIS-1017026 and HP Labs Innovation Research Program. We thank the anonymous reviewers for their useful comments.

References

1. Croft, W., Turtle, H., Lewis, D.: The use of phrases and structured queries in information retrieval. In: Proceedings of SIGIR 1991 (1991)
2. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: Proceedings of SIGIR 2004 (2004)

3. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proceedings of SIGIR 2005 (2005)
4. Fang, H., Zhai, C.: Semantic term matching in axiomatic approaches to information retrieval. In: Proceedings of SIGIR 2006 (2006)
5. Fuhr, N.: Probabilistic models in information retrieval. The Computer Journal 35(3), 243–255 (1992)
6. Grieff, W.R.: A theory of term weighting based on exploratory data analysis. In: Proceedings of SIGIR 1998 (1998)
7. Hartiwig, F., Dearing, B.E.: Exploratory Data Analysis. Sage Publications (1979)
8. Liu, S., Liu, F., Yu, C., Meng, W.: An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: Proceedings of SIGIR 2004 (2004)
9. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of SIGIR 2005 (2005)
10. Mitra, M., Buckley, C., Singhal, A., Cardie, C.: An analysis of statistical and syntactic phrases. In: Proceedings of RIAO (1997)
11. Ponte, J., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the ACM SIGIR 1998, pp. 275–281 (1998)
12. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at TREC-3. In: Proceedings of TREC-3 (1995)
13. Salton, G.: Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley (1989)
14. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: Proceedings of SIGIR 1996 (1996)
15. Tao, T., Zhai, C.: An exploration of proximity measures in information retrieval. In: Proceedings of SIGIR 2007 (2007)
16. van Rijbergen, C.J.: A theoretical basis for theuse of co-occurrence data in information retrieval. Journal of Documentation, 106–119 (1977)
17. van Rijsbergen, C.J.: Information Retrieval. Butterworths (1979)
18. Voorhees, E.M.: Overview of the trec 2005 robust retrieval track. In: Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005 (2006)
19. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR 2001 (2001)
20. Zheng, W., Fang, H.: Query Aspect Based Term Weighting Regularization in Information Retrieval. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 344–356. Springer, Heidelberg (2010)