Search Result Diversification via Data Fusion

Shengli Wu and Chunlan Huang
School of Computer Science and Telecommunication Engineering
Jiangsu University, Zhenjiang, China 212013
swu@ujs.edu.cn, palaceo77@163.com

ABSTRACT

In recent years, researchers have investigated search result diversification through a variety of approaches. In such situations, information retrieval systems need to consider both aspects of relevance and diversity for those retrieved documents. On the other hand, previous research has demonstrated that data fusion is useful for improving performance when we are only concerned with relevance. However, it is not clear if it helps when both relevance and diversity are both taken into consideration. In this short paper, we propose a few data fusion methods to try to improve performance when both relevance and diversity are concerned. Experiments are carried out with 3 groups of top-ranked results submitted to the TREC web diversity task. We find that data fusion is still a useful approach to performance improvement for diversity as for relevance previously.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – retrieval models

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Search result diversification, data fusion, linear combination, weight assignment

1. INTRODUCTION

In recent years, researchers have taken various approaches to investigate search result diversification [3, 1]. In such situations, information retrieval systems need to consider both relevance and diversity for those retrieved documents. In this short paper, we aim to find out if and how data fusion can help with this.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609451.

Previous research on data fusion (such as in [2, 6, 8]) demonstrates that it is possible to improve retrieval performance when we only consider relevance. Now with the new dimension of diversification, we need to re-evaluate the technology. In particular, some fusion methods need to be modified to accommodate for the new situation.

We may divide data fusion methods into two broad categories, according to how they deal with component results: equal-treatment and biased methods. As their names would suggest, the former treats all component results equally, while the latter does not. CombSum, CombMNZ, and the Condorcet method belong to the first category, while the linear combination method is a representative of the second category. Equal-treatment methods can likely be used in the new situation without modification, but the linear combination method needs more consideration.

In linear combination, weight assignment is a key issue for achieving good fusion performance and a considerable number of weight assignment methods have been proposed. Generally speaking, we need to consider two factors for weight assignment. One is the performance of every component retrieval system involved, and the other is the dissimilarity (or distance) between those component systems/results. For the information retrieval systems involved, well-performing systems should be given greater weights , while systems performing poorly should be assigned smaller weights. On the other hand, smaller weights should be assigned to those results that are similar to the others, while greater weights should be assigned to those results that are more different to the others. When assigning weights, we may take into consideration performance or dissimilarity, or even both together. It is also possible to use some machine learning techniques, known as "learning to rank", to train weights using some training data. This is especially popular for combining results at the feature level. These methods are aimed at optimizing a goal that is related to retrieval effectiveness measured by a given metric such as average precision. Because the metrics used for result diversification (e.g., ERR-IA@20) are very different from metrics such as average precision, almost all methods in this category cannot be directly used for result diversification.

In this paper, we are going to investigate data fusion methods, especially linear combination, for result diversification. Experiments are carried out to evaluate them with 3 groups of results submitted to the TREC web diversity task between 2009 and 2011. Experiments show that the proposed methods perform well and have the potential to be used for this purpose in practice.

Table 1: Information of 3 groups of results submitted to the web diversity task in TREC

TREC 2009		TREC 2010		TREC 2011	
Run	ERR-IA@20	Run	ERR-IA@20	Run	ERR-IA@20
MSRAACSF	0.2144	qirdcsuog3	0.2051	ICTNET11DVR3	0.4764
MSDiv3	0.2048	THUIR10DvNov	0.3355	liaQEWikiAnA	0.2287
uogTrDYCcsB	0.1922	UAMSD10aSRfu	0.2423	msrsv2011d1	0.4994
UamsDancTFb1	0.1774	UCDSIFTDiv	0.2100	UAmsM705tFLS	0.4378
mudvimp	0.1746	UMd10IASF	0.2546	uogTrA45Nmx2	0.5284
UCDSIFTdiv	0.1733	uogTrB67xS	0.2981	UWatMDSqltsr	0.4939
NeuDiv1	0.1705	cmuWi10D	0.2484	uwBA	0.3986
THUIR03AbClu	0.1665	ICTNETDV10R2	0.3222	CWIcIA2t5b1	0.3487
Average	0.1842	Average	0.2645	Average	0.4265
Variance	0.0003	Variance	0.0024	Variance	0.0098

2. SEVERAL METHODS FOR RESULT DIVERSIFICATION

As aforementioned, weight assignment is a key issue for the linear combination method. In this section, we look at different ways of dealing with this issue. Firstly, we may consider the performance of the retrieval system in question and its similarity with other retrieval systems separately, so that we may obtain two types of weights. Then a combination of these two types of weights can be used to fuse results. Note that performance and dissimilarity are two independent factors. On the one hand, performance of a result concerns the ranking positions of relevant documents and how diversified those relevant documents are in the ranked list of documents; on the other hand, the dissimilarity of two or more results is concerned with how different the ranking positions of the same documents are in two or more individual results, making no distinction between relevant and non-relevant documents.

Suppose there are a group of information retrieval systems ir_1, ir_2,...,ir_t, with some training data available for us to measure their performance and the dissimilarity between them. We further assume that their performances are p_1, p_2,..., p_t, respectively, as measured by a given metric (e.g., ERR-IA@20), so that we may then assign the value of a function of p_i (such as p_i, ${p_i}^2$, and so on) to w_i for $(1 \leq i \leq t)$, as the performance-related weight of ir_i.

Different approaches are possible for calculating the dissimilarity (or similarity) between two results. One approach is to refer to each result as a set of documents and calculate the overlap rate between two results (sets). It is also possible to calculate the correlation (such as Spearman's ranking coefficient or Kendall's tau coefficient) of two ranked lists of documents. If all the documents in the two results are associated with proper scoring information, then score-based methods such as the Euclidean distance or city block distance can be used. In the following we discuss two different ways of doing it.

Let us consider the top-n documents in all component results. Suppose that document d_{ij}, in result r_i, appears or is referred to in c_{ij} of the other $t-1$ results, then all n top-ranked documents of r_i are referred in all other results $ref_i = \sum_{j=1}^{n} c_{ij}$ times. For each document d_{ij}, the maximum times it can appear in the other $t-1$ results is $t-1$, This fact can be used to define the dissimilarity of r_i to other results as

$$dis_i = \frac{1}{n} \sum_{j=1}^{n} \frac{(t-1-c_{ij})}{t-1} \tag{1}$$

dis_i are always in the range of 0 and 1. We may define dis_i or a function of them as the dissimilarity-related weights. Methods using this definition are referred to as reference-based methods later in this paper. One advantage of using such methods for dissimilarity is that we can obtain the weights for component systems by considering all the documents of them together.

An alternative of calculating the dissimilarity between results is to compare documents' ranking difference for each pair of them. Let us consider the n top-ranked documents in both results r_A and r_B. Suppose that m $(m \leq n)$ documents appear in both r_A and r_B, and $(n-m)$ of them appear in only one of them. For those $n-m$ documents that only appear in one of the results, we simply assume that they occupy the places from rank $n+1$ to rank $2n-m$ whilst retaining the same relative orders in the other result. Thus we can calculate the average rank difference of all the documents in both results and use it to measure the dissimilarity of r_A and r_B. To summarize, we have

$$\begin{aligned} v(r_A, r_B) &= \frac{1}{n}\{ \sum_{i=1,2,\ldots,m}^{d_i \in r_A \wedge d_i \in r_B} \frac{|p_A(d_i) - p_B(d_i)|}{m} \\ &+ \sum_{i=1,2,\ldots,n-m}^{d_i \in r_A \wedge d_i \notin r_B} \frac{|p_A(d_i) - (n+i)|}{n-m} \\ &+ \sum_{i=1,2,\ldots,n-m}^{d_i \notin r_A \wedge d_i \in r_B} \frac{|p_B(d_i) - (n+i)|}{n-m} \} \end{aligned} \tag{2}$$

Here $p_A(d_i)$ and $p_B(d_i)$ denote the rank position of d_i in r_A and r_B, respectively. $\frac{1}{n}$ is the normalization coefficient which guarantees that $v(r_A, r_B)$ is in the range of 0 and 1. Based on Equation 2, the dissimilarity weight of r_i $(1 \leq i \leq t)$ is defined as

$$dis_i = \frac{1}{t-1} \sum_{j=1,2,\ldots,t}^{j \neq i} v(r_i, r_j) \tag{3}$$

Methods that use this definition are referred to as ranking difference based methods. No matter how we obtain

the weights for dissimilarity, we may combine dissimilarity-related weights with performance-related weights. Different options, such as $p_i * dis_i$, ${p_i}^2 * dis_i$, $p_i * {dis_i}^2$, and so on, might be used to obtain weight w_i for information retrieval system ir_i. At the fusion stage, the linear combination method uses the following equation to calculate scores:

$$g(d) = \sum_{i=1}^{t} w_i * s_i(d) \quad (4)$$

where $g(d)$ is the global score that document d obtains during data fusion, $s_i(d)$ is the (normalized) score that document d obtains from information retrieval system ir_i ($1 \leq i \leq t$), and w_i is the weight assigned to system ir_i. All the documents can be ranked according to the global scores they obtain.

3. EXPERIMENTS

In the 3 successive years from 2009 to 2011, the web track of TREC used the collection of "ClueWeb09". The collection consists of roughly 1 billion web pages crawled from the Web.

3 groups of results are chosen for the experiment. They are 8 top-ranked results [1] (measured by ERR-IA@20) submitted to the diversity task in the TREC 2009, 2010, and 2011 web track. The information about all the selected results is summarized in Table 1.

As we know, it is harder to get improvement over better component results through data fusion. However, the purpose of the experiments is going to see if we can obtain even better results by fusing a number of top-ranked results submitted.

In the 3 aforementioned groups of results, the 2009 group has the lowest average effectiveness (.1842), the lowest best effectiveness (.2144), and the smallest variance (.0003); the 2011 group has the highest average effectiveness (.4265), the highest best effectiveness (.5284), and the largest variance (.0098); for all three metrics the 2010 group comes second (average is .2645, best is .3356, variance is .0024). We will see that the effectiveness of the fused results is affected by these factors.

In the 2009 group, $MSRAACSF$ [4] is the best performer (ERR_IA@20: 0.2144). This run is submitted by Microsoft Research Asia in Beijing. For a given query, sub-topics are mined from different sources including anchor texts, search result clusters, and web sites at which search results are located; and documents are ranked by considering both relevance and diversity of mined sub-topics.

In the 2010 group, $THUIR10DvNov$ is the best among all 8 runs selected for the experiment. Its performance is 0.3355 when measured by ERR_IA@20. Two other runs $msrsv3div$ and $uwgym$ (baseline) are slightly better than $THUIR10DvNov$. The technical details of this run are not known because we cannot find the corresponding report for this. We speculate that this run is submitted by a research group in Tsinghua University.

In the 2011 group, $uogTrA45Nmx2$ [7] is the best performer (ERR_IA@20: 0.5284). This run is submitted by the IR research group at Glasgow University. It uses Terrier with a component xQuAD for search result diversification. The primary idea is to find useful information of sub-topics by sending the initial query to three commercial web search engines.

[1] *msrsv3div* and *uwgym* in 2010 and *UDCombine2* in 2011 are not chosen because they include much fewer documents than the others and using them would cause problems in calculating weights for the linear combination method and in the fusion process as well.

Table 2: Performance (measured by ERR-IA@20) of a group of data fusion methods (p denotes performance-related weight and dis denotes dissimilarity-related weight; dis is calculated using either Equation 1 or Equation 3; the figures in parentheses indicate the improvement rate of each method over the best component result; the figures in bold indicate the highest value in the column)

Group	2009	2010	2011	Ave.
best result	0.2144	0.3355	0.5284	0.3551
p	0.2544 (18.66%)	0.3567 (6.32%)	0.5398 (2.16%)	0.3836 9.05%
p^2	0.2499 (16.56%)	0.3684 (9.81%)	0.5343 (1.12%)	0.3842 9.16%
$dis * p$(Eq.1)	0.2552 (19.03%)	0.3548 (5.75%)	0.5398 (2.16%)	0.3833 8.60%
$dis * p^2$(Eq.1)	0.2492 (16.23%)	**0.3705** (10.43%)	0.5355 (1.34%)	0.3851 9.33%
$dis^2 * p$(Eq.1)	0.2548 (18.84%)	0.3533 (5.31%)	**0.5410** (2.38%)	0.3830 8.84%
$dis * p$(Eq.3)	**0.2553** (19.08%)	0.3531 (5.25%)	0.5388 (1.97%)	0.3824 8.77%
$dis * p^2$(Eq.3)	0.2503 (16.74%)	0.3658 (9.03%)	0.5347 (1.19%)	0.3836 8.99%
$dis^2 * p$(Eq.3)	0.2534 (18.19%)	0.3562 (6.17%)	0.5330 (0.87%)	0.3809 8.41%

In each year group, 50 queries are divided into 5 groups: 1-10, 11-20, 21-30, 31-40, and 41-50. 4 arbitrary groups of them are used as training queries, while the remaining one group is used for fusion test. This is referred to as the five-fold cross validation method in statistics and machine learning [5]. Every result is evaluated using ERR-IA@20 over training queries to obtain the performance weight p_i. On the other hand, either Equation 1 or Equations 2 and 3 are used with training data to obtain dis_i for the dissimilarity weight. After that, we try 5 different ways of combining the weights: p_i, ${p_i}^2$, $p_i * dis_i$, ${p_i}^2 * dis_i$, and $p_i * {dis_i}^2$.

In order to fuse component results by linear combination, reliable scores are required for all the documents required. In this study, we use the reciprocal function [2]. According to [2], the reciprocal function is very good for converting rankings into scores. For any resultant list $r = <d_1, d_2, ..., d_n>$, a score of $\frac{1}{i+60}$ is assigned to document d_i at rank i.

Experimental results are shown in Tables 2 and 3. Two metrics, ERR-IA@20 and α-nDCG@20, are used to evaluate all the fusion methods. The best component result is used as the baseline. When calculating dissimilarity weights by reference based method, or Equation 1, we use the top 100 documents in all component results. We have also tried some other options, including the top 50 and the top 200, though the experimental results are omitted here since they are so similar to what we observed for the top 100. When using rank difference based method, or Equations 2 and 3,

Table 3: Performance (measured by α-nDCG@20) of a group of data fusion methods (p denotes performance-related weight and dis denotes dissimilarity-related weight; dis is calculated using either Equation 1 or Equation 3; the figures in parentheses indicate the improvement rate of each method over the best component result; the figures in bold indicate the highest value in the column)

Group	2009	2010	2011	Ave.
best result	0.3653	0.4745	0.6298	0.4869
p	0.4130 (13.06%)	0.5071 (6.87%)	0.6510 (3.37%)	0.5237 7.77%
p^2	0.4108 (12.46%)	0.5226 (10.14%)	0.6468 (2.70%)	0.5267 8.43%
$dis * p$(Eq.1)	0.4141 (13.36%)	0.5057 (6.58%)	0.6513 (3.41%)	0.5237 7.78%
$dis * p^2$(Eq.1)	0.4100 (12.24%)	**0.5241** (10.45%)	0.6477 (2.84%)	0.5273 8.51%
$dis^2 * p$(Eq.1)	**0.4150** (13.61%)	0.5045 (6.32%)	**0.6522** (3.56%)	0.5239 7.83%
$dis * p$(Eq.3)	0.4138 (13.28%)	0.5054 (6.51%)	0.6506 (3.30%)	0.5233 7.70%
$dis * p^2$(Eq.3)	0.4108 (12.46%)	0.5202 (9.63%)	0.6478 (3.00%)	0.5263 8.36%
$dis^2 * p$(Eq.3)	0.4126 (12.95%)	0.5084 (7.14%)	0.6488 (3.17%)	0.5233 7.75%

to calculate dissimilarity weights, we use all the documents in each component result.

From Tables 2 and 3, we can see that all the data fusion methods involved perform better than the best component result. However, improvement rates vary from one year group to another. For all the data fusion methods involved, the largest improvement of over 10% occurs in the 2009 year group, which is followed by the 2010 year group with improvement between 5% and 11%, while the smallest improvement of less than 4% occurs in the 2011 year group. According to [8], the target variable of performance improvement of the fused result over the best component result is affected by a few factors. Among other factors, the variance of performance of all the component results and the performance of the best component result (see Table 1) have negative effect on the target variable. This can partially explain what we observe: all data fusion methods do the best in the 2009 data set, the worst in the 2011 data set, and the medium in the 2010 data set.

Intuitively, such a phenomenon is understandable. If a component result is very good and a large percentage of relevant documents in multiple categories are retrieved and top-ranked, then it must be very difficult to make any further improvements over this result; on the other hand, if some of the results are much poorer than the others, then it is very difficult for the fused result to outperform the best component result. Anyway, in all 3 data sets, all of the fused results exhibit improvements over the best component result.

If we compare performance-related weights to combined weights, it is not always the case that combined weights can achieve greater improvement. However, if we examine the greatest improvement in each case, it always happens when some form of combined weights is used. On average over three year groups, $dis * p^2$(Eq.1) performs the best no matter if ERR-IA@20 or α-nDCG@20 is used for evaluation. This suggests that $dis * p^2$(Eq.1) is a very good option for the combined weight.

4. CONCLUSIONS

In this short paper we have reported our investigation on the search result diversification problem via data fusion. Especially we focus on the linear combination method. Two options of calculating dissimilarity weights and several options of combining performance-related weights and dissimilarity-related weights have been proposed. Experiments with 3 groups of results submitted to the TREC web diversity task show that all the data fusion methods perform well and better than the best component result. Among those methods proposed, a combined weight of square performance and dissimilarity (calculated by comparing ranking difference of pair-wise results) outperforms the others on average.

In summary, the experiments demonstrate that data fusion is still a useful technique for performance improvement when addressing search result diversification.

5. REFERENCES

[1] E. Aktolga and J. Allan. Sentiment diversification with different biases. In *Proceedings of the 36th Annual International ACM SIGIR Conference*, pages 593–602, Dublin, Ireland, July 2013.

[2] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning mthods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference*, pages 758–759, Boston, MA, USA, July 2009.

[3] V. Dang and W. B. Croft. Term level search result diversification. In *Proceedings of the 36th Annual International ACM SIGIR Conference*, pages 603–612, Dublin, Ireland, July 2013.

[4] Z. Dou, K. Chen, R. Song, Y. Ma, S. Shi, and J. Wen. Microsoft Research Asia at the web track of TREC 2009. In *Proceedings of The Eighteenth Text REtrieval Conference*, Gaithersburg, Maryland, USA, November 2009.

[5] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (Volumn 2)*, pages 1137–1145, Montreal, Canada, August 1995.

[6] J. H. Lee. Analysis of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference*, pages 267–275, Philadelphia, Pennsylvania, USA, July 1997.

[7] R. McCreadie, C. Macdonald, R. Santos, and I. Ounis. University of Glasgow at TREC 2011: Experiments with terrier in crowdsourcing, microblog, and web tracks. In *Proceedings of The Twentieth Text REtrieval Conference*, Gaithersburg, Maryland, USA, November 2011.

[8] S. Wu and S. McClean. Performance prediction of data fusion for information retrieval. *Information Processing & Management*, 42(4):899–915, July 2006.