Fusion-based Methods for Result Diversification in Web Search

SHENGLI WU, Jiangsu University, China
CHUNLAN HUANG, Jiangsu University, China
JIEYU LI, Jiangsu University, China
FABIO CRESTANI, University of Lugano, Switzerland

In recent years, researchers have investigated search result diversification through a variety of approaches. Result diversification is especially necessary when a user issues a very general or ambiguous query. In such a situation, web search systems need to consider both relevance and diversity for those retrieved documents. On the other hand, previous research has demonstrated that data fusion is useful for improving performance only when we are concerned with relevance. In this article, we propose a few data fusion methods to try to improve performance when both relevance and diversity are concerned. The proposed methods are linear combination of multiple ranked lists of documents. Each list of them is assigned a weight based on a few factors: its performance, its dissimilarity with other lists, and its complementarity on coverage of subtopics with other lists. Experiments are carried out with 3 groups of top-ranked results submitted to the TREC web diversity task. We find that some of the proposed methods are effective and promising. Among three factors, complementarity on coverage of subtopics is the most useful. In addition, a logarithmic model that converts ranking information into scores is also investigated. In our experiments, it is more effective than the reciprocal model, which is a state-of-the-art score normalization method.

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

General Terms: Algorithms, Experimentation, Measurement, Performance

Additional Key Words and Phrases: Data fusion, Web search, Result diversification, Linear combination, Weight assignment, Score normalization

1. INTRODUCTION

In recent years, researchers have taken various approaches to investigate search result diversification [Dang and Croft 2013; Aktolga and Allan 2013]. The rationale behind it is that for some general or ambiguous queries, a good search engine should provide results with a wide coverage of all possible subtopics to the user, rather than a narrow focus on one or very few special subtopics. In such a situation, web search systems need to consider both relevance and diversity for those retrieved documents. In this article, we aim to find out if and how data fusion can help with this.

Previous research on data fusion such as in [Lee 1997; Wu and McClean 2006; Cormack et al. 2009] 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.

According to how they deal with component results, we may divide data fusion methods into two broad categories: equal-treatment and biased methods. As their names 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. If relevance is the only concern, then two factors have been found useful for weight assignment [Wu et al. 2009; Wu 2012b]. One is the performance of every component search system involved, and the other is the dissimilarity (or distance)

between those component systems/results. For the web search 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" [Liu 2011], to train weights by using some training data. This is especially popular for combining results at feature level. However, when diversity is also a concern, we need to consider more factors.

In this article, we are going to investigate data fusion methods, especially linear combination, for result diversification. Novel methods of weight assignment are proposed to accommodate this. Especially the concept of complementarity on coverage of subtopics is taken into consideration. Experiments are carried out to evaluate them with 3 groups of results submitted to the TREC web diversity task between 2009 and 2011 [Clarke et al. 2009; Clarke et al. 2011]. Experiments show that the proposed methods perform well. In all three years the fused results are more effective than the best component results by a clear margin.

The rest of this article is organized as follows: in Section 2 we discuss some related work on search result diversification and data fusion. Several data fusion methods for result diversification are presented in Section 3. Experiments are reported in Section 4 to evaluate a number of data fusion methods. We also explore the relationship of three important concepts introduced in Section 3: performance weight, dissimilarity weight and complementarity weight. In Section 4, we compare three score normalization methods. Section 5 reports the conclusions.

2. RELATED WORK

We divide this section into two parts, in which some related work on result diversification and data fusion are reviewed, respectively.

2.1. Result Diversification

A number of different approaches have been proposed to promote search result diversification. Usually this is done by a two-step procedure: for a given topic first we run a typical web search system to obtain a ranked list of documents, then we apply a result diversification algorithm to re-rank the documents so as to promote diversity.

Result diversification algorithms can be divided into two categories: implicit and explicit. The first approach promotes diversity by comparing the difference of the documents in the list and re-rank them, or by extracting subtopics from all the documents and re-rank them. This means the method does not need any extra information apart from the documents themselves obtained through a traditional retrieval system. [Carbonell and Goldstein 1998] proposed a maximal marginal relevance-based method. The basic idea is to re-rank documents according to a linear combination of each document's relevance to the query and its similarity to other documents that are already selected in the list. Based on the same idea as [Carbonell and Goldstein 1998], [Zhai et al. 2003] used KL-divergence to measure the distance of a new document to those that are already in the list; and both [Rafiei et al. 2010] and [Wang and Zhu 2009] used correlation to measure the novelty of a new document to those already in the list. Some methods extract potential subtopics by analysis of the documents involved, then re-rank them. Analysis can be done in different ways. [Carterette and Chandar 2009] extracts potential subtopics by topic modelling, while [He et al. 2011] does this by query-specific clustering.

The explicit approach needs more information than the implicit approach does. Assuming it is known that the given query has a set of subtopics and other related information, the result diversification algorithm maximizes the coverage to all subtopics in the top-k results. IA_select [Agrawal et al. 2009], xQuAD [Santos et al. 2010], and the Proportionality Model [Dang and Croft 2012] are algorithms in this category. Apart from the re-ranking methods themselves, one key issue for these methods is how to obtain accurate subtopic information from external sources. Different sources such as commercial web search engines and Wikipedia have been investigated.

More recently, a few more methods have been proposed. They use a combination of different types of techniques. In [Zheng and Fang 2013], two representative result diversification methods, xQuAD [Santos et al. 2010] and PM-2 [Dang and Croft 2012], are involved. For a given query, performance of the two methods is predicted based on some factors such as diversity of the documents and number of relevant documents in the results retrieved. The best performer is chosen to present its results based on the prediction.

The approach that is taken in [Liang et al. 2014] is also a combination of different techniques. Their method mainly includes three parts. In the first part they Combine a few results from different search systems. The data fusion methods used are CombSum and CombMNZ. In the second part they infer latent subtopics by topic modelling. The document set used is the output of Step 1 with full text for all the documents. Lastly, result diversification is performed by a typical result diversification method, PM-2 [Dang and Croft 2012].

2.2. Data Fusion

A lot of research has been done on data fusion in web search and information retrieval. Different methods such as CombSum [Fox et al. 1993; Fox and Shaw 1994], CombMNZ [Fox et al. 1993; Fox and Shaw 1994], linear combination [Bartell et al. 1994; Vogt and Cottrell 1998; 1999; Wu et al. 2009], Borda Count [Aslam and Montague 2001], Condorcet fusion [Montague and Aslam 2002; Wu 2013], the multiple criteria approach [Farah and Vanderpooten 2007], cluster-based fusion [Kozorovitzky and Kurland 2011] and others have been investigated. Data fusion has been used in many different applications including routing tasks [Bigot et al. 2011], expert search [Macdonald and Ounis 2006; Liang and de Rijke 2013], blog opinion search [Wu 2012a], query-focused summarization [Wei et al. 2010], and others.

Score normalization is a related issue to data fusion. The zero-one method [Lee 1997], the fitting method [Wu et al. 2006], Z-scores [Montague and Aslam 2001; Kamps et al. 2005], the reciprocal rank [Lillis et al. 2006; Cormack et al. 2009], the logistic model [Calvé and Savoy 2000] and others have been investigated and evaluated. However, these researches on score normalization aim at improving relevance-based performance: diversity is not an issue.

In this paper, we investigate result diversification via data fusion in which relevance and diversification are considered at the same time. This is an extended work on the same issue as in [Wu and Huang 2014] and more results of empirical investigation are presented. More importantly, not included in [Wu and Huang 2014], there are two major contributions in this article:

(1) Some weighting schemes for the linear combination method are proposed. The proposed weighting schemes take a new factor into consideration. The new factor, complementarity of results on subtopic coverage, is specific to result diversification.
(2) A logarithmic function-based method for converting ranking information into scores is presented for result diversification.

Both of them are effective. Compared with other alternatives, they perform better in our experiments (see Section 4 for details). On the other hand, compared with the work of [Liang et al. 2014], our approach is a pure and novel data fusion method, while theirs is a combination of different methods, in which some traditional data fusion methods like CombSum and CombMNZ are used.

3. FUSION-BASED METHODS FOR RESULT DIVERSIFICATION

We first review some data fusion methods involved in the experiments. Two of them (CombSum and CombMNZ) serve as baselines while the third one (linear combination) provides the basis for our proposed method. Some score normalization methods are also reviewed. After that we discuss some concepts including queries, topics, subtopics, result diversification, complementarity of results, dissimilarity of results. Based on these concepts, we propose a group of data fusion methods.

3.1. Typical Data Fusion Methods

Assume there is a document collection D and a group of search systems $IR = \{ir_i\}$ for $(1 \leq i \leq t)$. All search systems ir_i $(1 \leq i \leq t)$ search D for a given query q and each of them provides a ranked list of documents $r_i = < d_{i1}, d_{i2}, ..., d_{in} >$. We further assume that a score $s_i(d)$ is associated with each of the documents d in the list. The data fusion technique is to use some algorithms to merge these n ranked lists into one. We hope that the fused result is more effective than those component results.

CombSum [Fox et al. 1993; Fox and Shaw 1994] uses the following equation

$$g(d) = \sum_{i=1}^{t} s_i(d) \tag{1}$$

to calculate scores for every document d. Here $s_i(d)$ is the normalized score that d obtains from ir_i. If d does not appear in any r_i, then a default score (e.g., 0) must be assigned to it, or $s_i(d) = 0$. After that, every document d obtain a global score $g(d)$ and all the documents can be ranked according to the global scores they obtain.

Another method CombMNZ [Fox et al. 1993; Fox and Shaw 1994] uses the equation

$$g(d) = m * \sum_{i=1}^{t} s_i(d) \tag{2}$$

to calculate scores. Here m is the number of results in which document d appears.

As aforementioned in Section 1, data fusion methods can be divided into two categories: equal-treatment and biased methods. Methods such as CombSum and CombMNZ belong to the first category. Methods in this categories may be used in many different situations and for different purposes. For example, experiments in [Liang et al. 2014] show that CombSum and CombMNZ perform very well for result diversification.

The linear combination method [Vogt and Cottrell 1998; 1999; Wu et al. 2009; Wu and Crestani 2002] uses the equation below

$$g(d) = \sum_{i=1}^{t} w_i * s_i(d) \tag{3}$$

to calculate scores. w_i is the weight assigned to system ir_i. The linear combination method is very flexible since different weights can be assigned to different web search

Table I. Symbols and their meanings used in this acticle

Symbol	Description
$as(r,i)$	Set of subtopics that the first i documents in r cover
$c_i(j)$	complementarity of subtopic coverage of r_i to r_j
c_i	average complementarity of r_i to other (t-1) results
d	a document in D
d_{ij}	the j-th document in L_i
D	a collection of documents
$g(d)$	final score that d obtain from a fusion algorithm
IR	a list of n search engines that contribute results for fusion
ir_i	the i-th search engine in IR
n	the number of top documents in r_i we use for fusion
p_i, $p(r_i)$	performance of r_i
q	a given query
r_i	a list of m documents $<d_{i1},d_{i2},...,d_{im}>$ retrived from ir_i for q
$rank_A(d)$	rank position of document d in result A
t	the number of search engines in IR ($t > 2$)
$s_i(d)$	normalized score that document d obtains from ir_i
$u(r_i, r_j)$	dissimilarity between result r_i and r_j
v_i	average dissimilarity between result r_i and other $t-1$ results
w_i	the weight assigned to ir_i for linear combination of results

systems. It is helpful when those equal-treatment methods are not able to obtain good results. Weight assignment is a key issue for linear combination to be successful. However in different situations, we may require different mechanisms to deal with this issue. See Subsection 3.3 for more details.

Next let us review a few score normalization methods. Score normalization is required in some high-level retrieval applications such as data fusion [Wu 2012b], federated search [Shokouhi and Si 2011; Hong and Si 2013; Markov and Crestani 2014], among others. The zero-one method [Lee 1997] is very commonly used. In a ranked list of documents r_i=$< d_{i1}, d_{i2}, ..., d_{im} >$, the maximal score is normalized to 1, the minimal score to 0, and other scores are mapped linearly into a score between 0 and 1. Researchers also find that very often converting rankings into scores is a good way of generating scores. According to [Cormack et al. 2009], the reciprocal function is very good for converting rankings into scores. The reciprocal model uses the formula $s(rank) = 1/(rank+60)$ to convert ranks into scores. However, both the zero-one method and the reciprocal function work reasonably well when relevance but not diversity of documents is considered. It needs to be examined how these methods perform when both relevance and diversity are considered.

For convenience, all the symbols used in this article are summarized in Table I.

3.2. Basic Concepts and Examples

In a diversity task, queries are general and ambiguous. Relevant documents are not focused on a single topic, but can be on different subtopics.

Example 1. In TREC's 2009 web track, query 6 is "KCS". "KCS" can be an acronym for Kansas City Southern railroad, or Kanawha County Schools in West Virginia, or Knox County School system in Tennessee, or KCS Energy, Inc. Thus this query has at least 4 subtopics.

For such ambiguous queries, their resultant lists should include documents that are relevant to as many different types of subtopics as possible. With a diversified resultant list, a user is more likely to find the information needed.

As to data fusion, we would hope that those component results include documents relevant to as many subtopics as possible. Based on diversified component results, it is possible to generate diversified final result.

Example 2. For a given query, there are 3 result lists r_1=<d_1,d_3,d_2, d_6>, r_2=<d_4,d_5,d_3,d_8>, and r_3=<d_7,d_5,d_6,d_8>. Among them, d_1 is relevant to subtopics 1 and 2, d_3 is relevant to subtopic 3, d_4 is relevant to subtopics 2 and 3, d_5 is relevant to subtopic 2, d_7 is relevant to subtopics 3 and 4, while d_2, d_6, and d_8 are non-relevant documents. See below for the distribution of relevant documents that cover different subtopics.

Result	Document	Subtopic 1	2	3	4
r_1	d_1	Y	Y		
	d_3			Y	
r_2	d_4		Y	Y	
	d_5		Y		
r_3	d_7			Y	Y
	d_5		Y		
	d_3			Y	

In this example, subtopic 4 is only covered by document d_7 in r_3, but not at all by documents in r_1 or r_2; while subtopic 1 is only covered by document d_1 in r_1. Therefore, r_1 and r_3 are complementary to each other, while the coverage of r_2 is a subset of the subtopics that either r_1 or r_3 covers. Therefore, fusing r_1 and r_3 would cover all 4 subtopics, while adding r_2 would not be useful.

Example 3. We fuse the 3 component results in Example 2 by CombSum. Documents at rank 1, 2, 3, and 4 are given scores of 4, 3, 2, 1, respectively. The fused result r_f is <(d_5,6),(d_3,5),(d_1,4), (d_4,4),(d_7,4),...>. Only 2 subtopics are covered in the top-2 documents of the fused result r_f. This can be improved by applying linear combination with different weights to each component results.

Example 4. We still fuse the 3 component results in Example 2, but this time, by linear combination. The key problem is: in order to obtain better result than that from CombSum, how to decide weights for all the component results involved?

According to our discussion above, we should assign heavy weights to both r_1 and r_3, and assign light weight to r_2. If we assign 4 to both r_1 and r_3, and 1 to r_2, then the fused result r_{f2} is <(d_1,16),(d_7,16),(d_5,15), (d_3,14),(d_6,12),...>. This time, the top-2 documents cover all 4 subtopics.

In previous data fusion experiments [Bartell et al. 1994; Vogt and Cottrell 1998; Wu and McClean 2006] that only consider relevance, performance of each component result and dissimilarity between component results are factors that affect performance of the fused result significantly. Dissimilarity between two results can be calculated in many different ways. One way is to regard each result as a set of documents. Then we are able to calculate how many of them are different.

People may conjecture that there is high correlation between dissimilarity and complementarity. As later our experiment shows this is true. However, They are different in one aspect. When considering dissimilarity of two results, we consider all the documents equally and do not distinguish between relevant documents and nonrelevant documents; when considering complementarity, we ignore all non-relevant documents. Even for those relevant documents, it is required to further considering their coverage of subtopics. Such a difference makes the two factors to perform differently for weight assignment in the linear combination method. See more details in Section 4.

3.3. Weight Assignment for Linear Combination

Weight assignment is a key issue for the linear combination method. In this subsection, we discuss a few different ways of dealing with this issue. Firstly, let us consider the following three different factors:

(1) The performance of each search system in question.
(2) The similarity between one result and the others.
(3) The complementarity on relevant subtopics between one result and the others.

The first factor is straightforward. The only thing we need to consider is the measure for performance evaluation. In this study, we use ERR-IA@20, which is a typical measure for result diversification [Chapelle et al. 2009]. ERR-IA@20 will be used for calculating all types of weights in the training process. Other measures such as α-nDCG@20 [Clarke et al. 2008] may also be used.

As to the second factor, we do not distinguish relevant and non-relevant results when computing the similarity of two results. According to [Wu 2012b], there are different solutions. We may divide them into three categories: score-based similarity such as the Euclidean distance or street block distance, ranking-based similarity such as Spearman's correlation coefficient or Pearson product-moment correlation coefficient, and set-based metrics. In this study, we use a ranking-based measure, which will be described later.

The third factor only makes sense for result diversification in which multiple subtopics exist for the same given query. Some results may cover more or less the same subtopics, while some others may cover very different subtopics. This is determined by the implementation of search systems that we obtain results from. We should be able to distinguish them.

Three types of weighting schemes can be obtained if we consider the above-mentioned three factors separately. Based on these factors, different combinations of them are possible to obtain more weighting schemes. Now let us detail these weighting schemes.

Suppose there are a group of web search systems ir_1, ir_2,...,ir_t, and some training data are available for us to evaluate them and calculate values of some parameters required. First the performance of those results can be calculated by using a given metric (e.g., ERR-IA@20). We use p_1, p_2,..., p_t to represent them.

Next we discuss how to calculate the dissimilarity of two results. It can be done by comparing documents' ranking difference for each pair of them. Let us consider the n top-ranked documents in 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

$$u(r_A, r_B) = \frac{1}{n}\{ \sum_{i=1,2,\ldots,m}^{d_i \in r_A \wedge d_i \in r_B} \frac{|rank_A(d_i) - rank_B(d_i)|}{m} + \sum_{i=1,2,\ldots,n-m}^{d_i \in r_A \wedge d_i \notin r_B} \frac{|rank_A(d_i) - (n+i)|}{n-m}$$

$$+ \sum_{i=1,2,\ldots,n-m}^{d_i \notin r_A \wedge d_i \in r_B} \frac{|rank_B(d_i) - (n+i)|}{n-m}\} \quad (4)$$

Here $rank_A(d_i)$ and $rank_B(d_i)$ denote the rank position of d_i in r_A and r_B, respectively. $\frac{1}{n}$ is the normalization coefficient, which guarantees that $u(r_A, r_B)$ is in the range of 0 and 1. Based on Equation 4, the average dissimilarity between r_i ($1 \leq i \leq t$) and other $t-1$ results is defined as

$$v_i = \frac{1}{t-1} \sum_{j=1,2,\ldots,t}^{j \neq i} u(r_i, r_j) \quad (5)$$

The last factor is complementarity of subtopic coverage between results. Let us consider two results $r_1 = < d_{11}, d_{12}, ..., d_{1n} >$ and $r_2 = < d_{21}, d_{22}, ..., d_{2n} >$. At rank position i, r_1 covers a set of subtopics $as(r_1, i)$, and r_2 covers a set of subtopics $as(r_2, i)$. Based on r_1 and r_2, we generate a super-result r. At rank position i, r covers a set of $as(r, i)$ subtopics. Here $as(r, i)$ is defined as the union of $as(r_1, i)$ and $as(r_2, i)$. We may use a metric, such as ERR-IA@20, to measure the performances of r_1, r_2, and r. Suppose that the values obtained are $p(r_1)$, $p(r_2)$, and $p(r)$, then the complementarity of r_1 to r_2 can be defined as

$$c_1(2) = \frac{p(r) - p(r_2)}{p(r)} \quad (6)$$

If there are t results in total, then the average complementarity of r_1 to other $t-1$ results r_2, r_3,..., r_t can be defined as

$$c_1 = \frac{1}{t-1} \sum_{i=2}^{t} c_1(i) \quad (7)$$

c_2, c_3,..., c_t can be calculated in the same way.

After calculating values for the desirable parameters, we are able to define weights. One option is to define weights by considering any individual factor. Weights can also be defined by considering those factors together. In the following we will refer to p_i and ${p_i}^2$ (or p and p^2 in short) as performance weight, v_i and ${v_i}^2$ (or v and v^2) as dissimilarity weight, and c_i and ${c_i}^2$ (or c and c^2) as complementarity (or diversity) weight. At the fusion stage, the linear combination method uses Equation 3 to calculate scores for all the documents, then a new ranking of documents can be generated accordingly.

4. EXPERIMENTS

In this section we report the experiments that evaluate the performance of the weighting schemes presented in the previous section. The data set used is "ClueWeb09" [1]. The web track of TREC used it in the 3 successive years from 2009 to 2011. The "ClueWeb09" collection consists of roughly 1 billion web pages crawled from the Web.

Three groups of results are chosen for the experiment. They are 8 top-ranked results (measured by ERR-IA@20) submitted to the diversity task in the TREC 2009, 2010, and 2011 web track. *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

[1] http://www.lemurproject.org/clueweb09.php/

Table II. Information of 3 groups of results submitted to the web diversity task in TREC (the figures in parentheses are ERR-IA@20 values of selected runs)

TREC 2009	TREC 2010	TREC 2011
MSRAACSF (0.2144)	msrsv3div (0.3473)	uogTrA45Nmx2 (0.5284)
MSDiv3 (0.2048)	THUIR10DvNov (0.3355)	msrsv2011d1 (0.4994)
uogTrDYCcsB (0.1922)	ICTNETDV10R2 (0.3222)	UWatMDSqltsr (0.4939)
UamsDancTFb1 (0.1774)	uogTrB67xS (0.2981)	ICTNET11DVR3 (0.4764)
mudvimp (0.1746)	UMd10IASF (0.2546)	UAmsM705tFLS (0.4378)
UCDSIFTdiv (0.1733)	cmuWi10D (0.2484)	uwBA (0.3986)
NeuDiv1 (0.1705)	UAMSD10aSRfu (0.2423)	CWIcIA2t5b1 (0.3487)
THUIR03AbClu (0.1665)	UCDSIFTDiv (0.2100)	liaQEWikiAnA (0.2287)
Average 0.1842	Average 0.2823	Average 0.4265
Variance 0.0003	Variance 0.0025	Variance 0.0098

fusion process as well [2]. The information about all the selected results is summarized in Table II.

In the 3 aforementioned groups of results, the 2009 group has the lowest average effectiveness (0.1842), the lowest best effectiveness (0.2144), and the smallest variance (.0003); the 2011 group has the highest average effectiveness (0.4265), the highest best effectiveness (0.5284), and the largest variance (0.0098); for all three metrics the 2010 group comes second (average is 0.2823, best is 0.3473, variance is 0.0025).

4.1. Data fusion results

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 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 [Kohavi 1995]. Every result is evaluated using ERR-IA@20 over training queries to obtain performance weight p_i and ${p_i}^2$. Dissimilarity weight v_i and ${v_i}^2$ and complementarity weight c_i and ${c_i}^2$ are calculated accordingly. Each of them is used individually as the weight of the corresponding web search system. Different combinations of them are also used: $p * v$, $p^2 * v$, $p * v^2$, $p * c$, $p^2 * c$, $p * c^2$, $p^2 * v * c$, etc.

Score normalization is necessary for data fusion to achieve good results. In this subsection, we present the logarithmic function-based method to normalize scores of all component results. This method uses the formula $s(rank) = max\{1-0.2*ln(rank+1), 0\}$ to generate scores for documents at each ranking position $rank$. It assigns positive scores {0.8614,0.7803,0.7227,...,0.0006} to top 147 documents and zero to the rest of them. More discussion about the logarithmic function-based method and a comparison of this method and two other score normalization methods are reported in the next section.

[2]As a matter of fact, *uwgym* in 2010 includes 12,719 documents, and *UDCombine2* in 2011 includes 48,951 documents, while other runs include 250,000 or close to 250,000 documents.

Table III. Performance (measured by ERR-IA@20) of a group of data fusion methods (p and p^2 are performance weights and v and v^2 are dissimilarity weights and c and c^2 are complementarity weight; 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	0.2144	0.3473	0.5284	0.3634
CombSum	0.2440 (+13.81%)	0.4110 (+18.34%)	0.5457 (+3.27%)	0.4002 (+10.12%)
CombMNZ	0.2493 (+16.28%)	0.4052 (+16.67%)	0.5368 (+1.59%)	0.3971 (+9.27%)
p	0.2446 (+14.09%)	0.4135 (+19.06%)	0.5528 (+4.62%)	0.4037 (+11.09%)
p^2	0.2433 (+13.48%)	0.4046 (+16.50%)	0.5629 (+6.53%)	0.4036 (+11.06%)
c	0.2482 (+15.76%)	0.4160 (+19.78%)	0.5522 (+4.50%)	0.4054 (+11.56%)
c^2	0.2472 (+15.30%)	0.4149 (+19.46%)	0.5635 (+6.64%)	**0.4085** (+12.41%)
v	0.2453 (+14.41%)	0.4131 (+18.95%)	0.5448 (+3.10%)	0.4011 (+10.37%)
v^2	0.2433 (+13.48%)	0.4136 (+19.09%)	0.5426 (+2.69%)	0.3998 (+10.02%)
pc	0.2475 (+15.44%)	0.4139 (+19.18%)	0.5633 (+6.60%)	0.4082 (+12.33%)
p^2c	0.2465 (+14.97%)	0.4032 (+16.10%)	**0.5668** (+7.27%)	0.4055 (+11.59%)
pc^2	**0.2503** (+16.74%)	0.4091 (+17.79%)	0.5654 (+7.00%)	0.4083 (+12.36%)
pv	0.2432 (+13.43%)	0.4167 (+19.98%)	0.5512 (+4.31%)	0.4037 (+11.09%)
p^2v	0.2444 (+13.99%)	0.4049 (+16.59%)	0.5633 (+6.60%)	0.4042 (+11.23%)
pv^2	0.2441 (+13.85%)	**0.4176** (+20.24%)	0.5512 (+4.31%)	0.4043 (+11.25%)
p^2cv	0.2487 (+16.00%)	0.4029 (+16.01%)	0.5664 (+7.19%)	0.4061 (+11.75%)

Experimental results are shown in Tables III and IV. Two metrics, ERR-IA@20 and α-nDCG@20 ($\alpha = 0.5$), are used to evaluate all the fusion methods. The best component result is used as the baseline. Apart from linear combination with different types of weights, two traditional data fusion methods, CombSum and CombMNZ, are also tested.

From Tables III and IV, we can see that all the data fusion methods involved perform better than the best component result. Both CombSum and CombMNZ perform very well, although CombSum is a little better than CombMNZ (about 1%).

In this experiment, three types of weights (performance, dissimilarity, and complementarity) are applied separately with two options (linearly or squared). Although all of them perform better than the best component result, their effect on fusion results are quite different. Comparing them with CombSum and CombMNZ, we find that using complementarity weight or performance weight is useful for performance improvement, while dissimilarity weight does not make much difference. We also observe that complementarity weight is more useful than performance weight, when using one of them alone. For performance weight, square function p^2 is worse than linear function p; while for complementarity weight, square function c^2 is more beneficial than linear function c.

Table IV. Performance (measured by α-nDCG@20) of a group of data fusion methods (p and p^2 are performance weights and v and v^2 are dissimilarity weights and c and c^2 are complementarity weight; 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	0.3653	0.4909	0.6298	0.4953
CombSum	0.4019 (+10.02%)	0.5588 (+13.83%)	0.6610 (+4.95%)	0.5406 (+9.15%)
CombMNZ	**0.4111** (+12.54%)	0.5555 (+13.16%)	0.6471 (+2.75%)	0.5345 (+7.91%)
p	0.4015 (+9.91%)	0.5636 (+14.81%)	0.6668 (+5.87%)	0.5440 (+9.83%)
p^2	0.4025 (+10.18%)	0.5558 (+13.22%)	0.6742 (+7.05%)	0.5442 (+6.66%)
c	0.4020 (+10.05%)	0.5646 (+15.01%)	0.6660 (+5.75%)	0.5442 (+9.87%)
c^2	0.4055 (+11.00%)	0.5673 (+15.56%)	0.6745 (+7.10%)	0.5491 (+10.86%)
v	0.4031 (+10.35%)	0.5595 (+13.97%)	0.6584 (+4.54%)	0.5404 (+9.11%)
v^2	0.4016 (+9,94%)	0.5608 (+14.24%)	0.6545 (+3.92%)	0.5393 (+8.88%)
pc	0.4070 (+11.42%)	0.5677 (+15.64%)	0.6748 (+7.15%)	**0.5498** (+11.00%)
p^2c	0.4063 (+11.22%)	0.5562 (+13.30%)	**0.6776** (+7.59%)	0.5467 (+10.38%)
pc^2	0.4089 (+11.94%)	0.5623 (+14.54%)	0.6759 (+7.32%)	0.5490 (+10.84%)
pv	0.3998 (+9.44%)	0.5674 (+15.58%)	0.6653 (+5.64%)	0.5442 (+9.87%)
p^2v	0.4027 (+10.24%)	0.5568 (+13.42%)	0.6747 (+7.13%)	0.5448 (+9.99%)
pv^2	0.4006 (+9.66%)	**0.5690** (+15.91%)	0.6642 (+5.46%)	0.5446 (+9.95%)
p^2cv	0.4069 (+11.39%)	0.5558 (+13.22%)	0.6760 (+7.34%)	0.5463 (+10.30%)

In the experiment we also use different types of combined weights. On average combining performance weight and complementarity weight is the best option. Combining dissimilarity weight with others is not very good. Combining all three types of weights is also considered and the result is no better than the combination of performance weight and complementarity weight. Only the result of one form of combination p^2cv, which is the best among all such combinations, is presented.

In this experiment, c^2, $p * c$ and $p * c^2$ perform better than the other options on average. They outperform CombSum and CombMNZ by 1% to 3% when either of the two measures is used. Although the difference is not big, it is statistically significant at the .05 level. Comparing them with the best result submitted, the difference is larger. The improvement rates are about 12% and 11% when ERR-IA@20 or α-nDCG@20 are used, respectively.

4.2. Relationship between Three Types of Weights

In this subsection we explore the relationship among three different types of weights: performance weights, dissimilarity weights, and complementarity weights. In the experiments presented in Section 4.1, these three types of weights are calculated five times for each group of component results. To estimate their correlation, we first rank them separately by value, and then calculate the Spearman's rank coefficient between

Table V. Spearman's rank coefficients between two different types of weights (average of 5 groups)

Pair of Weights	TREC 2009	TREC 2010	TREC 2011
c and v	0.8762	0.9048	0.9238
p and v	-0.1738	0.1476	-0.8238
p and c	-0.4953	0.0524	-0.6857

Table VI. Euclidean's distance between two different types of weights (average of 5 groups)

Pair of Weights	TREC 2009	TREC 2010	TREC 2011
c and v	0.0124	0.0189	0.0187
p and v	0.1048	0.1608	0.2420
p and c	0.1104	0.1602	0.2487

two rankings. See Table V for their rank coefficients. The rank coefficients for c and v are always above 0.8, but that for other two pairs vary. As an alternative measure, we also calculate the Euclidean distance of any two types of weights. For consistency, each group of weights are normalized to unit length, thus the Euclidean distances are comparable across different cases. Table VI shows the results. The distance between c and v is around 1/10 of the distance between c and p, or between v and p.

The above observation can explain why using combined weight of c and v is not a good solution. Additionally, although dissimilarity weights and complementarity weights are very similar, experimental results presented in Section 4.1 show that their effects on data fusion are different. complementarity weights are beneficial but dissimilarity weights are not. It also suggests that the performance of linear combination are sensitive to weights assigned to component results. There is a small beneficial area around complementarity weights. When weights drift away a little from the beneficial area, fusion results may deteriorate quite quickly.

5. CONVERTING RANKINGS INTO SCORES

In this section we compare two methods of converting rankings into scores: the logarithmic model and the reciprocal model. The logarithmic model uses the formula $score(i) = max\{1 - 0.2 * ln(i + 1), 0\}$ to normalize score of documents at rank i to $score(i)$, while the reciprocal model uses the formula $score(i) = 1/(i + 60)$ for the same purpose. According to [Cormack et al. 2009], the reciprocal function is very good for converting rankings into scores. However, their scenario is different from ours here. In their experiments, a topic does not include any subtopic and binary relevance judgment is used. This is common in many historical TREC tasks. But this time we are in a different situation.

Firstly, we try to see which of them fits the observed data better. The procedure is as follows: in a year group, we select all 8 runs. We check all the documents involved to see if they are relevant or not to the given query. If a document is non-relevant to any subtopics, then a score of 0 is assigned to that document; while if a document is relevant to s ($s \geq 1$) subtopics at the same time, then a score of s is assigned to that document. Scores are averaged for all the documents at each rank (1,2,...,100). Thus we obtain an observed curve that indicates the average number of subtopics to which a document at a certain rank would be relevant. Then we run curve estimation using a statistic software "SPSS" to see how accurate the two models are. Table VII shows the results.

From Table VII we can see that in all three year groups, the logarithmic model is slightly more accurate than the reciprocal model with larger R^2 and F values.

Table VII. Estimation accuracy of the two ranking converting models

Group	Logarithmic		Reciprocal	
	R^2	F	R^2	F
2009	.907	959.079	.900	879.491
2010	.926	1226.342	.890	794.127
2011	.959	2312.175	.936	1424.361

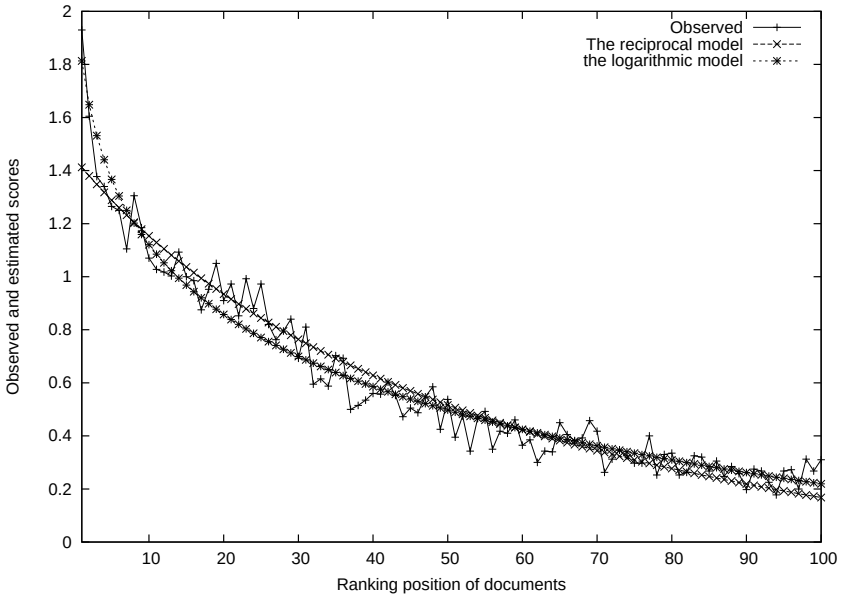

Fig. 1. Score estimation of two different models with the TREC 2011 web diversity data set

To have a closer look at how these two models perform, we depict the observed curves and the two estimated curves for the data of 2011 group in Figure 1 [3]. The curves for the other two groups are not presented due to similarity. In Figure 1, the observed curve is not very smooth and follows a zigzag pattern. This indicates that the number of relevant subtopics varies from one rank to next and the number of cases (8 runs * 50 queries) are not large enough to stabilize them. In Figure 1, we can also see that both logarithmic and reciprocal models fit the observed curve quite well. Probably there is only one noticeable phenomenon: the logarithmic model fits the observed curve better than the reciprocal model at a few very top ranks. This phenomenon is significant since top-ranked documents are more important than others. It may explain why the logarithmic model leads to better results than the reciprocal model in the fusion experiment, as we discuss now.

We examine the effectiveness of score normalization by fusing the same results with different score normalization methods. Apart from the logarithmic model and the reciprocal model, the zero-one method is also included for comparison. The zero-one method [Lee 1997] is a typical method for score normalization, which normalizes scores of a resultant list of documents into the range of 0-1. The experimental result is shown in Tables VIII and IX.

[3]In Figure 1, the curves of the reciprocal model and the logarithmic model are magnified linearly to best fit the observed curve. Thus it is easier to compare them.

Table VIII. Performance (measured by ERR-IA@20) of data fusion using 3 different score normalization methods

Fusion method	Logarithmic	Reciprocal	Zero-one
CombSum	0.4002	**0.4014**	0.3642
CombMNZ	**0.3971**	0.3845	0.3507
p	**0.4037**	0.3959	0.3763
p^2	**0.4036**	0.3915	0.3724
c	**0.4054**	0.3974	0.3759
c^2	**0.4085**	0.3917	0.3733
v	**0.4010**	0.4008	0.3746
v^2	0.3998	**0.4023**	0.3746
pc	**0.4082**	0.3911	0.3740
pc^2	**0.4052**	0.3924	0.3725
p^2c	**0.4055**	0.3937	0.3732
pv	**0.4082**	0.3958	0.3777
p^2v	**0.4055**	0.3907	0.3722
pv^2	**0.4083**	0.3961	0.3757
p^2cv	**0.4060**	0.3949	0.3737
Average	0.4044	0.3947	0.3721

Table IX. Performance (measured by α-nDCG@20) of data fusion using 3 different score normalization methods

Fusion method	Logarithmic	Reciprocal	Zero-one
CombSum	**0.5406**	0.5403	0.5060
CombMNZ	**0.5379**	0.5232	0.4863
p	**0.5440**	0.5382	0.5085
p^2	**0.5442**	0.5358	0.5071
c	**0.5442**	0.5391	0.5079
c^2	**0.5491**	0.5352	0.5062
v	**0.5404**	0.5399	0.5061
v^2	0.5393	**0.5413**	0.5063
pc	**0.5498**	0.5360	0.5073
pc^2	**0.5490**	0.5363	0.5082
p^2c	**0.5467**	0.5386	0.5092
pv	**0.5442**	0.5384	0.5100
p^2v	**0.5448**	0.5353	0.5063
pv^2	**0.5446**	0.5386	0.5090
p^2cv	**0.5463**	0.5397	0.5093
Average	0.5443	0.5371	0.5062

From Tables VIII and IX we can see that both the logarithmic model and the reciprocal model are better than the zero-one method in all the cases by a clear margin (5%-8%). On average and in most cases, the logarithmic model is slightly better than the reciprocal model. One exception is for one form of weighting (v^2), when the reciprocal model is better than the logarithmic model. This experiment shows it is very likely that the logarithmic model is a better option than the reciprocal model for score normalization when multiple subtopics are considered.

6. CONCLUSIONS

In this article we have reported our investigation on search result diversification via data fusion. Especially we focus on the linear combination method in which weight assignment is a key issue. In order to achieve better fusion results, complementarity of results on subtopic coverage has been identified as an important factor that affects the weight of an component search system. Using it alone or combining it with other factors, such as performance, can achieve very good results.

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 variety of combined weights of performance and complementarity outperform 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. The proposed methods are promising and have the potential to be used in such applications.

REFERENCES

R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. 2009. Diversifying search results. In *Proceedings of the Second International Conference on Web Search and Web Data Mining*. Barcelona, Spain, 5–14.

E. Aktolga and J. Allan. 2013. Sentiment Diversification with Different Biases. In *Proceedings of the 36th Annual International ACM SIGIR Conference*. Dublin, Ireland, 593–602.

J. A. Aslam and M. Montague. 2001. Models for Metasearch. In *Proceedings of the 24th Annual International ACM SIGIR Conference*. New Orleans, Louisiana, USA, 276–284.

B. T. Bartell, G. W. Cottrell, and R. K. Belew. 1994. Automatic combination of multiple ranked retrieval systems. In *Proceedings of ACM SIGIR'94*. Dublin, Ireland, 173–184.

A. Bigot, C. Chrisment, T. Dkaki, G. Hubert, and J. Mothe. 2011. Fusing different information retrieval systems according to query-topics: a study based on correlation in information retrieval systems and TREC topics. *Information. Retrieval* 14, 6 (2011), 617–648.

A. L. Calvé and J. Savoy. 2000. Database Merging Strategy Based on Logistic Regression. *Information Processing & Management* 36, 3 (2000), 341–359.

J. Carbonell and J. Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. Melbourne, Australia, 335–336.

B. Carterette and P. Chandar. 2009. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. Hong Kong, China, 1287–1296.

O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. Hong Kong, China, 621–630.

C. Clarke, N. Craswell, and I. Soboroff. 2009. Overview of the TREC 2009 Web Track. In *Proceedings of The Eighteenth Text REtrieval Conference*. National Institute of Standards and Technology, USA, Gaithersburg, Maryland, USA.

C. Clarke, N. Craswell, I. Soboroff, and E. Voorhees. 2011. Overview of the TREC 2011 Web Track. In *Proceedings of The Twentieth Text REtrieval Conference*. National Institute of Standards and Technology, USA, Gaithersburg, Maryland, USA.

C. Clarke, M. Kolla, G. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Singapore, 659–666.

G. V. Cormack, C. L. A. Clarke, and S. Büttcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Mthods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference*. Boston, MA, USA, 758–759.

V. Dang and W. B. Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th Annual International ACM SIGIR Conference*. Portland, OR, USA, 65–74.

V. Dang and W. B. Croft. 2013. Term Level Search Result Diversification. In *Proceedings of the 36th Annual International ACM SIGIR Conference*. Dublin, Ireland, 603–612.

M. Farah and D. Vanderpooten. 2007. An outranking approach for rank aggregation in information retrieval. In *Proceedings of the 30th ACM SIGIR Conference*. Amsterdam, The Netherlands, 591–598.

E. A. Fox, M. P. Koushik, J. Shaw, R. Modlin, and D. Rao. 1993. Combining Evidence from Multiple Searches. In *The First Text REtrieval Conference (TREC-1)*. Gaitherburg, MD, USA, 319–328.

E. A. Fox and J. Shaw. 1994. Combination of Multiple Searches. In *The Second Text REtrieval Conference (TREC-2)*. Gaitherburg, MD, USA, 243–252.

J. He, E. Meij, and M. de Rijke. 2011. Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology* 62, 3 (January 2011), 550–571.

D. Hong and L. Si. 2013. Search Result Diversification in Resource Selection for Federated Search. In *Proceedings of the 36th Annual International ACM SIGIR Conference*. Dublin, Ireland, 613–622.

J. Kamps, M. de Rijke, and B. Sigurbjörnsson. 2005. Combination Methods for Crosslingual Web Retrieval. In *Accessing Multilingual Information Repositories, 6th Workshop of the CrossLanguage Evalution Forum, CLEF 2005, Revised Selected Papers (LNCS 4022 Springer 2006)*. Vienna, Austria, 856–864.

R. Kohavi. 1995. 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)*. Montreal, Canada, 1137–1145.

A. K. Kozorovitzky and O. Kurland. 2011. Cluster-based fusion of retrieved lists. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Beijing, China, 893–902.

J. H. Lee. 1997. Analysis of Multiple Evidence Combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference*. Philadelphia, Pennsylvania, USA, 267–275.

S. Liang and M. de Rijke. 2013. Finding knowledgeable groups in enterprise corpora. In *Proceedings of the 36th International ACM SIGIR conference on research and development in Information Retrieval*. Dublin, Ireland, 1005–1008.

S. Liang, Z. Ren, and M. de Rijke. 2014. Fusion Helps Diversification. In *Proceedings of the 37th Annual International ACM SIGIR Conference*. Cold Coast, QLD, Australia, 303–312.

D. Lillis, F. Toolan, R. Collier, and J. Dunnion. 2006. ProbFuse: a probabilistic approach to data fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference*. Seattle, Washington, USA, 139–146.

T. Liu. 2011. *Learning to Rank for Information Retrieval*. Springer. 292 pages.

C. Macdonald and I. Ounis. 2006. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management*. Arlington, Virginia, USA, 387–396.

I. Markov and F. Crestani. 2014. Theoretical, Qualitative, and Quantitative Analyses of Small-Document Approaches to Resource Selection. *ACM Transactions on Information Systems* 32, 2 (April 2014).

M. Montague and J. A. Aslam. 2001. Relevance Score Normalization for Metasearch. In *Proceedings of ACM CIKM Conference*. Berkeley, USA, 427–433.

M. Montague and J. A. Aslam. 2002. Condorcet fusion for improved retrieval. In *Proceedings of ACM CIKM Conference*. McLean, VA, USA, 538–548.

D. Rafiei, K. Bharat, and A. Shukla. 2010. Diversifying web search results. In *Proceedings of the 19th International Conference on World Wide Web*. Raleigh, North Carolina, USA, 781–790.

R. Santos, C. Macdonald, and I. Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*. Raleigh, North Carolina, USA, 881–890.

M. Shokouhi and L. Si. 2011. Federated Search. *Foundations and Trends in Information Retrieval* 1, 5 (2011), 1–102.

C. C. Vogt and G. W. Cottrell. 1998. Predicting the Performance of Linearly combined IR Systems. In *Proceedings of the 21st Annual ACM SIGIR Conference*. Melbourne, Australia, 190–196.

C. C. Vogt and G. W. Cottrell. 1999. Fusion Via a Linear Combination of Scores. *Information Retrieval* 1, 3 (October 1999), 151–173.

J. Wang and J. Zhu. 2009. Portfolio theory of information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Boston, MA, USA, 115–122.

F. Wei, W. Li, and S. Liu. 2010. iRANK: A rank-learn-combine framework for unsupervised ensemble ranking. *Journal of the American Society for Information Science and Technology* 61, 6 (2010), 1232–1243.

S. Wu. 2012a. Applying the Data Fusion Technique to Blog Opinion Retrieval. *Expert Systems with Applications* 39, 1 (January 2012), 1346–1353.

S. Wu. 2012b. *Data Fusion in Information Retrieval*. Springer. 220 pages.

S. Wu. 2013. The Weighted Condorcet Fusion in Information Retrieval. *Information Processing & Managment* 49, 1 (January 2013), 114–126.

S. Wu, Y. Bi, X. Zeng, and L. Han. 2009. Assigning Appropriate Weights for the Linear Combination Data Fusion Method in Information Retrieval. *Information Processing & Management* 45, 4 (July 2009), 413–426.

S. Wu and F. Crestani. 2002. Data Fusion with Estimated Weights. In *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management*. McLean, VA, USA, 648–651.

S. Wu, F. Crestani, and Y. Bi. 2006. Evaluating Score Normalization Methods in Data Fusion. In *Proceedings of the 3rd Asia Information Retrieval Symposium (LNCS 4182)*. Singapore, 642–648.

S. Wu and C. Huang. 2014. Search Result Diversification via Data Fusion. In *Proceedings of the 37th Annual International ACM SIGIR Conference*. Cold Coast, QLD, Australia, 827–830.

S. Wu and S. McClean. 2006. Performance Prediction of Data Fusion for Information Retrieval. *Information Processing & Management* 42, 4 (July 2006), 899–915.

C. Zhai, W. Cohen, and J. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Toronto, Canada, 10–17.

W. Zheng and H. Fang. 2013. A Diagnostic Study of Search Result Diversification Methods. In *Proceedings of International Conference on the Theory of Information Retrieval, ICTIR '13*. Copenhagen, Denmark, 17.