Fusion-based Methods for Result Diversification in Web Search

Shengli Wu [a,b] Chunlan Huang [a] Liang Li [a] Fabio Crestani [c]

[a] *School of Computer Science, Jiangsu University, China*

[b] *School of Computing & Mathematics, Ulster University, UK*

[c] *Faculty of Informatics, University of Lugano, Switzerland*

Abstract

Search result diversification of text documents is especially necessary when a user issues a faceted or ambiguous query to the search engine. A variety of approaches have been proposed to deal with this issue in recent years. In this article, we propose a group of fusion-based result diversification methods with the aim to improve performance that considers both relevance and diversity. They are linear combinations of scores that are obtained from different component search systems. The weight of each search system is determined by considering three factors: performance, dissimilarity, and complementarity. There are two major contributions. Firstly, we find that all the three factors of performance and complementarity and dissimilarity are useful for effective weighting of linear combination. Secondly, we present the logarithmic function-based model for converting ranking information into scores. Experiments are carried out with four groups of results submitted to the TREC web diversity task. Experimental results show that some of the fusion methods that use the aforementioned techniques perform more effectively than the state-of-the-art fusion methods for result diversification.

Key words: Data fusion, Web search, Result diversification, Linear combination, Weight assignment, Linear score normalization

1 Introduction

In recent years, researchers have taken various approaches to investigate search result diversification (Santos et al., 2015; Naini et al., 2016). The rationale behind it is that for some faceted 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. For example, there are many different interpretations for the query "online mapping sites".

It could mean Goggle Maps, MSN Maps, Yahoo Maps, MapQuest, or other free printable maps. In such a situation, a web search engine needs to consider both relevance and diversity for those retrieved documents. In this article, we investigate this by using the data fusion technique.

Previous research on data fusion demonstrates that it is possible to improve retrieval performance when we only consider relevance (Wu, 2012b). The rationale behind it is: if a document is retrieved by multiple search systems, then it is more likely that the document is relevant to the information need. For the result diversity task, the situation is somewhat different. We may use the same principle of multiple evidence but different explanations for these two situations. In the relevance-related task, data fusion is expected to promote more relevant documents to the top-ranked positions; while in the result diversity task, data fusion is expected to secure wider coverage of different types of relevant documents in the top-ranked positions. As we will see later, different explanations have impact on the fusion algorithms and some of them need to be modified to accommodate for the new situation.

Suppose for a given query, there are a group of search systems and each of them retrieves a ranked list of documents from the same collection of documents. Those ranked lists are referred to as component results later in this article. Data fusion is the technique of combining those component results so as to improve performance. 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 (Fox et al., 1993), CombMNZ (Fox et al., 1993), and the Condorcet fusion (Montague and Aslam, 2002) belong to the first category, while the linear combination method (Vogt and Cottrell, 1998; Wu et al., 2009) 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). 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 heavy weights, while systems performing poorly should be assigned light weights. On the other hand, lighter weights should be assigned to those results that are similar to the others, while heavier weights should be assigned to those results that are more different to the others (Wu and McClean, 2006a). When assigning weights, we may take performance or dissimilarity or even both of them

together into consideration. 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 investigate data fusion methods, especially linear combination, for result diversification. Because we need to balance multiple interpretations of the given query, fusing results balancing both relevance and diversity is more challenging than merely taking relevance into account. Novel methods of weight assignment are proposed to accommodate this. The concept of complementarity on coverage of subtopics is introduced. This is helpful when some of the component results cover more different subtopics than the others. Our investigation shows that using it alone or with other types of weights together can lead to very good weighting schemes. Additionally, we propose a logarithmic function-based method for converting ranking information into scores. Experiments are carried out to evaluate them with four groups of results submitted to the TREC web diversity task between 2009 and 2012 (Clarke et al., 2009, 2011). Experiments show that the proposed methods perform well. For all four groups of results, 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 the proposed data fusion methods. Some discussion about data fusion on result diversification is presented. Besides, the proposed score normalization method is also evaluated. Section 5 is the conclusions.

2 Related Work

This section is divided into two parts: one is some related work on result diversification and the other is on data fusion.

2.1 Result Diversification

Result diversification has been identified as an important problem in many different applications such as web search (Capannini et al., 2011; Santos et al., 2010), recommender systems (Schedl and Hauger, 2015), database systems (Deng and Fan, 2014), among others. In this article we focus on the issue

of result diversification of web search. Santos et al. (2015) is a recent review article about result diversification in web search.

Usually search result diversification 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. Many re-ranking algorithms are greedy algorithms. It means that each time such an algorithm will select one document according to a given criterion. The documents chosen are put to the resultant list one after another until all the positions are occupied.

Result diversification algorithms can be divided into two categories: implicit and explicit. The implicit approach promotes diversity by comparing the difference of the documents in the list and re-ranking them, or by extracting subtopics from all the documents and re-ranking them. This means that such a method does not need any extra information apart from the documents themselves retrieved through a traditional retrieval system and possibly some statistics of the document collection.

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. Analysis can be done in different ways. Carterette and Chandar (2009) extracted potential subtopics by topic modeling, while He et al. (2011) did this by query-specific clustering. Zuccon et al. (2012) modeled the result diversification problem using facility location analysis, which was taken from operations research.

Vieira et al. (2011) evaluated a group of implicit methods including Swap, BSwap, MMR, Motley, MSD (Max-Sum Dispersion), CLT (Clustering), GMC (Greedy Marginal Contribution), and GNE (Grasp with Neighbor Expansion). In their experiments, GNE was the best performer. But on the other hand, it took the longest time for computation. Thang et al. (2015) evaluated 6 implicit methods Swap, Motley, MMR, MSD, GrassHopper, and Affinity Graph. They observed that MMR was the best performer.

The explicit approach needs more information than the implicit approach does but it is usually more effective than the implicit approach. Assuming it is known that the given query has a set of subtopics and other related in-

formation, the result diversification algorithm maximizes the coverage of all subtopics in the top k results. IA_select (Agrawal et al., 2009), xQuAD (Santos et al., 2010), and PM-2 (Dang and Croft, 2012) are algorithms in this category. In the re-ranking stage, xQuAD considers both relevance (relevance of the candidate document to the original query) and diversity (relevance of the candidate document to all sub-topics of that query). A variant of xQuAD is proposed by Ozdemiray and Altingovde (2015). The difference between their method mixCombSum and xQuAD is in the diversity part. Instead of using a greedy algorithm to obtain the documents one by one, their method uses CombSum to sum up subtopic scores of all the documents involved. Then documents are re-ranked by their total scores. IA_select is a simplified version of xQuAD and it only has the diversity part. PM-2 treats the re-ranking problem as a proportional election of the documents for all sub-topics. Similar to IA_select and mixCombSum, PM-2 only considers relevance of the candidate document to all sub-topics without taking the original query into consideration.

Apart from the re-ranking methods themselves, one key issue for these explicit methods is how to obtain accurate subtopic information from external sources. Different sources such as commercial web search engines (Santos et al., 2010) and Wikipedia (Kaptein et al., 2009) have been investigated.

Recently, machine learning has been used to deal with the result diversification problem. Many different machine learning techniques including the maximal marginal relevance model (Xia et al., 2015), neural tensor networks (Xia et al., 2016), recurrent neural networks (Jiang et al., 2017), Markov decision process (Xia et al., 2017), the document repulsion model (Li et al., 2017), word embedding (Ullah et al., 2016), the learning-to-rank algorithm LambdaMART (Wu et al., 2016) have been attempted.

Instead of formulating user intents for a query as a flat list of subtopics, Hu et al. (2015) presented hierarchical diversification models. Hierarchical topic models are estimated for measuring topical diversity of documents in Azarbonyad et al. (2017). Wang et al. (2016a) investigated methods of evaluating search result diversity using intent hierarchies

Result diversification has also been investigated in various specific search applications such as image retrieval (Ionescu et al., 2016), historic entity or event search (Gupta and Berberich, 2016), medical records retrieval (Li et al., 2015), music recommendations (Schedl and Hauger, 2015), search in Twitter (Wang et al., 2016b), among others.

2.2 Data Fusion

Data fusion has been widely investigated and used in many different research areas and applications such as classification (Li et al., 2018), artificial neural networks (Chen et al., 2017), the internet of things (Alam et al., 2017), information retrieval/web search (Wu and Crestani, 2015), among many others.

When the search results include text, image, and other types of media, then multimodal data fusion (Lahat et al., 2015) can be used to fuse multiple evidence from different types of media. For some types of applications, data fusion may take place at different levels. For example, in content-based image retrieval (Kaliciak et al., 2014) or multimedia event detection (Lan et al., 2012), data fusion may be carried out at the representation level (early fusion) or at the decision level (late fusion) or both. In these two cases the early fusion strategy fuses some low-level features before performing classification; while the late fusion strategy combines outputs of different classifiers.

In this article, we investigate result diversification via data fusion in which relevance and diversification are considered at the same time. We assume that the component results involved for fusion are ranked list of text documents. We also assume that those component results are already diversified by some special algorithms. This means that it is a late fusion strategy of web search results for result diversification. In the following we review some data fusion work on web search/information retrieval, especially on result diversification.

In information retrieval/web search, data fusion has been applied in many different tasks including the routing task (Bigot et al., 2011), expert search (Macdonald and Ounis, 2006), blog opinion search (Wu, 2012a), query-focused summarization (Wei et al., 2010), and others. Different methods such as CombSum (Fox et al., 1993), CombMNZ (Fox et al., 1993), linear combination (Vogt and Cottrell, 1998; Wu et al., 2009), Borda Count (Aslam and Montague, 2001), Condorcet fusion (Montague and Aslam, 2002), the multiple criteria approach (Farah and Vanderpooten, 2007), cluster-based fusion (Kozorovitzky and Kurland, 2011), genetic algorithm-based method (Ghosh et al., 2015) and others have been investigated. A geometric framework of score-based data fusion methods is presented in Wu and Crestani (2015). However, in all these methods, relevance is the only concern and result diversification is not considered.

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 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 modeling. 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).

Xu et al. (2016) investigated differential evolution-based methods for result diversification. Linear combination is the method used for fusion and differential evolution is applied to train weights for different component search systems.

Instead of fusing results that are already diversified, Xu and Wu (2017) investigated the early fusion strategy. It includes three stages. First a group of results are generated by some typical search algorithms, at this stage only relevance is considered and diversification is not considered. Second, a group of results are fused by a given fusion algorithm such as CombSum. Third, the fused result are re-ranked by a typical result diversification algorithm such as PM-2. Their experiments show that the early fusion strategy is as effective as some late fusion strategies, but can be implemented more efficiently.

Most of the above-mentioned methods are score-based data fusion methods. They need reliable and comparable scores from all component results. Such a requirement is not very often satisfied when various kinds of techniques are used in the implementation of the underlining information search systems. In such a situation, score normalization is necessary before data fusion takes place.

Many score normalization methods have been proposed: the zero-one method (Lee, 1997), the fitting method (Wu et al., 2006), Z-scores (Montague and Aslam, 2001), the reciprocal rank (Cormack et al., 2009), the logistic model (Calvé and Savoy, 2000) and so on. However, these score normalization methods are aimed at improving relevance-based performance and diversity is not an issue. Note that the reciprocal rank and the logistic model convert ranking into scores. Therefore, scores are not required when using either of them.

This piece of work is considerably different from the above-mentioned (Zheng and Fang, 2013; Liang et al., 2014; Ozdemiray and Altingovde, 2015; Xu et al., 2016; Xu and Wu, 2017). 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.

3 Fusion-Based Methods for Result Diversification

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 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 t ranked lists into one. The goal is to make the fused result more effective than those component results.

CombSum (Fox et al., 1993) 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 score that d obtains from ir_i. If those scores from different search systems are not comparable, then it is better to normalize them before the fusion process for better performance. $s_i(d)$ is used to denote either unnormalized or normalized score later in this article. If d does not appear in any r_i, then a default score (e.g., 0) must be assigned to it. 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) 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 category 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; Wu et al., 2009) 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 systems. It is useful when those equal-treatment methods are not able to obtain good results. Weight assignment is a key issue for linear combination to be successful.

When relevance is the only concern, previous research finds that the performance of all component results and dissimilarity between component results are two factors that affect performance of the fused result significantly (Wu and McClean, 2006b). Now both relevance and diversity need to be considered at the same time, we bring a third factor that measures the novelty of a component result relating to other component results. These three factors are referred to as performance, dissimilarity, and complementarity weights later in this article. See Section 3.2 for more details.

A related issue is score normalization, which is required by data fusion when the scores of documents in those component results are not comparable. Previous research finds that the reciprocal function (Cormack et al., 2009) is a good option for a few TREC tasks such as the adhoc task. However, we find that the logarithmic function is a better option when used for the web diversity task. See Section 4.3 for more details.

For convenience, all the symbols used in this article are summarized in Table 1.

3.1 Basic Concepts and Examples

In a diversity task, queries are faceted or 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.

Table 1
Symbols and their meanings used in this article

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 r_i
D	a collection of documents
$g(d)$	final score that d obtain from a fusion algorithm
IR	a list of t 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 measured by a measure like ERR-IA@20
q	a given query in Q
Q	a group of queries
r_i	a list of n documents $<d_{i1},d_{i2},...,d_{in}>$ retrieved from ir_i for q
$rank_k(d)$	rank position of document d in result r_k
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

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, if the following two conditions are satisfied: (1) the component results are equally effective, and (2) subtopics are evenly covered by all the component results, then it is very likely that equal-treatment data fusion methods such as CombSum and CombMNZ can achieve very good results. The following example illustrates this.

Example 2. For a given query q, there are 3 result lists $r_1 = <d_1, d_2, d_3, d_4>$, $r_2 = <d_5, d_6, d_7, d_8>$, and $r_3 = <d_9, d_{10}, d_{11}, d_{12}>$. Among them, d_1 is relevant to subtopics 1 and 2, d_5 is relevant to subtopics 3 and 4, d_9 is relevant

to subtopics 5 and 6, while all others are non-relevant documents. In this case, the above two conditions are well satisfied. Equal-treatment data fusion methods are able to achieve good results. If we use CombSum or CombMNZ to fuse them, then d_1, d_5, and d_9 will be the top-3 documents in the fused result. This is the best possible result.

If either or both of the above two conditions are not well-satisfied, then equal-treatment data fusion methods may not work well and linear combination is a better choice. We may assign a weight to each result so as to reflect its degree of importance in the fusion process. In the following we focus on the coverage of sub-topics by component results and consider profitable weights for fusion performance improvement. Let us see an example to illustrate this.

Example 3. For a given query q, there are 3 result lists $r_1 = <d_1, d_2, d_3, d_4>$, $r_2 = <d_5, d_6, d_2, d_8>$, and $r_3 =<d_7, d_6, d_4, d_8>$. Among them, d_1 is relevant to sub-topics 1 and 2, d_2 is relevant to subtopic 3, d_5 is relevant to subtopics 2 and 3, d_6 is relevant to subtopic 2, d_7 is relevant to subtopics 3 and 4, while d_3, d_4, 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_2			Y	
r_2	d_5		Y	Y	
	d_6		Y		
	d_2			Y	
r_3	d_7			Y	Y
	d_6		Y		

We combine them by CombSum and documents at rank 1, 2, 3, and 4 are given scores of 4, 3, 2, 1, respectively. d_6 obtains a score of 6 (3 for rank position 2 in r_2 and 3 for rank position 2 in r_3) and d_2 obtains a score of 5 (3 for rank position 2 in r_1 and 2 for rank position 3 in r_2). Therefore, the fused result r_f is <(d_6,6),(d_2,5),...>. r_f is not very good because only 2 subtopics are covered in the top 2 documents.

It is worthwhile to investigate why CombSum does not work well. 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. Fusing r_1 and r_3 would cover all four subtopics, while adding r_2 would not be useful. In other words, both r_1 and r_3 cover four subtopics while r_2 only covers two. Therefore, this is not an ideal situation for CombSum to achieve good performance because the second condition is not well-satisfied. This can be improved by applying linear combination with different weights to the component results.

If we fuse them by linear combination, then the key problem is how to assign weights for those component results. 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 let $w_1 = 4$, $w_3 = 4$, and $w_2 = 1$, 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 four subtopics.

Example 3 shows the ability of linear combination over CombSum when the conditions are not favorable for the latter. For a given group of search engines, such a phenomenon may not just happen occasionally for one or two queries, but happen regularly over a large number of queries. For a given set of documents and a given query, search results are related to the implementation technologies used by search engines. Therefore, some specific patterns may occur in those search results from particular search engines. For example, if search engine A uses an implicit result diversification method and search engine B uses an explicit result diversification method, then it is possible that A is less effective than B (Santos et al., 2015); if both search engines A and B use explicit result diversification methods but obtain extra information from different sources (Santos et al., 2010; Kaptein et al., 2009), then it is possible that the subtopic coverage of their results are different. When such patterns happen regularly in results from those search engines across different queries, they can be learnt by applying some training data and used to improve fusion performance by linear combination.

However, how to assign suitable weights to component search systems is challenging. As a matter of fact, it can be defined as an optimization problem. For a collection of training data and a given metric, the best solution can only be found by searching the whole solution space. Instead of doing that, we present some heuristic methods for weight assignment and the goal of this approach is to achieve good result in both performance and efficiency.

3.2 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) Performance of each search system in question.
(2) Similarity between one result and the others.
(3) Complementarity on relevant subtopic coverage between one result and the others.

In previous data fusion experiments (Vogt and Cottrell, 1998; Wu and McClean, 2006b) that only consider relevance, researchers find that performance of each component result and dissimilarity between component results are factors that affect performance of the fused result significantly. The third factor, complementarity, is newly introduced because it only makes sense for results diversification.

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). 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, ranking-based similarity, and set-based metrics. Score-based similarity can be defined by the Euclidean distance or street block distance, while ranking-based similarity can be defined by Spearman's correlation coefficient, Kendall's tau correlation coefficient, Goodman and Kruskal's gamma correlation coefficient, and so on. 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. We should take advantage of this so as to obtain better fusion results by assigning different weights to different search systems.

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, a collection of documents D, a given query q and all its relevant documents in D. Thus all the search systems search the collection and the performance of those results r_1, r_2,..., r_t 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 top-n ranked documents in results r_1 and r_2 for a given query q, respectively. Suppose that m ($m \leq n$) documents appear in both r_1 and r_2, and $(n-m)$ of them appear in only one of them. Without loss of generality, we let d_1, d_2,..., d_m appear in both r_1 and r_2, d_{m+1}, d_{m+2},..., d_n appear in r_1 but not r_2, and d_{n+1}, d_{n+2},..., d_{2n-m} appear in r_2 but not r_1. 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_1 and r_2. To summarize, we have

$$\begin{aligned} u(r_1, r_2) = \frac{1}{n(2n-m)} \{ & \sum_{i=1,2,...,m}^{d_i \in r_1 \wedge d_i \in r_2} |rank_1(d_i) - rank_2(d_i)| \\ & + \sum_{i=m+1,m+2,...,n}^{d_i \in r_1 \wedge d_i \notin r_2} |rank_1(d_i) - (n-m+i)| \\ & + \sum_{i=n+1,n+2,...,2n-m}^{d_i \notin r_1 \wedge d_i \in r_2} |rank_2(d_i) - i| \} \end{aligned} \tag{4}$$

Here $rank_1(d_i)$ and $rank_2(d_i)$ denote the rank positions of d_i in r_1 and r_2, respectively. $\frac{1}{n(2n-m)}$ is the normalization coefficient, which guarantees that $u(r_1, r_2)$ is in the range of [0,1]. Note that $2n-m$ is the number of pairs and n is the maximum rank difference of the same document in r_1 and r_2. Based on Equation 4, the average dissimilarity between r_i ($1 \leq i \leq t$) and other $t-1$ results is defined as

$$v_i = \frac{1}{t-1} \sum_{j=1,2,...,t \wedge j \neq i} u(r_i, r_j) \tag{5}$$

The last factor is complementarity of subtopic coverage between results. Let us consider two results $r_i = < d_{i1}, d_{i2}, ..., d_{in} >$ and $r_j = < d_{j1}, d_{j2}, ..., d_{jn} >$ for a given query q. At rank position k, r_i covers a set of subtopics $as(r_i, k)$, and r_j covers a set of subtopics $as(r_j, k)$. Based on r_i and r_j, we may define a super-result r. Note that r is a virtual result and we only consider its performance. At rank position k, r covers a set of $as(r, k)$ subtopics. Here $as(r, k)$ is defined as the union of $as(r_i, k)$ and $as(r_j, k)$. r's performance can be calculated from r_i and r_j . We may use a metric, such as ERR-IA@20, to measure the performances of r_i, r_j, and r. Suppose that the values obtained are $p(r_i)$, $p(r_j)$, and $p(r)$, then the complementarity of r_i to r_j can be defined as

$$c_i(j) = \frac{p(r) - p(r_j)}{p(r)} \tag{6}$$

From Equation 6, we can see that the complementarity of r_i to r_j (i.e. $c_i(j)$) and the complementarity of r_j to r_i (i.e. $c_j(i)$) are two different quantities.

If there are t results in total, then the average complementarity of r_i to other $t-1$ results r_1,...,r_{i-1}, r_{i+1},...,r_t can be defined as

$$c_i = \frac{1}{t-1} \sum_{j=1,2,...,t \wedge j \neq i} c_i(j) \tag{7}$$

Suppose that the training data set comprises a group of queries Q. For each query q, we may obtain the parameter values of p by evaluating the results directly, and we compute the parameter values of c and v according to Equations 5 and 7, respectively. These values are averaged over all the queries and thus each parameter obtains one value for the whole training data set. In the following we still use p_i, c_i, and v_i to denote those averaged values. There are a few different ways of defining weights. One option is to define weights by considering any individual factor. For example, we may define p_i, p_i^2,...,as performance weights. Some typical values for power is: 0, 1, 2, etc. if power is 0, then all the weights are the same and linear combination becomes CombSum; if power is 1, then it is the performance-level weighting scheme; if power is 2, then it is the performance-square weighting scheme. Note that the larger the power is, the more influential the better-performed system is. Different weighting schemes such as p_i, p_i^2,..., are investigated empirically in Wu et al. (2009) and it is found that on average p_i^2, p_i^3,..., p_i^6 are more effective than p_i when only relevance is considered. Similar weighting schemes can be applied to other two factors dissimilarity and complementarity. 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. We may also define combined weights based on these factors. They are represented by a combination of p, c, and v.

When all the weights are decided by the above-mentioned methods with some training data, it is ready for fusing new results. 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 four successive years

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

from 2009 to 2012. The "ClueWeb09" collection consists of roughly 1 billion web pages crawled from the Web.

Four groups of results are chosen for the experiment. They are top eight results (measured by ERR-IA@20) submitted to the diversity task in the TREC 2009, 2010, 2011, and 2012 web track. Each participant was allowed to submit multiple runs to the same track and the multiple runs from the same participant are much similar than those from different participants. In order to avoid fusing very similar results, we try to take just one run from each participant. However, this is not strictly observed for the 2012 group because not many runs are submitted in that year.

The information about all the selected results is summarized in Table 2. *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. [2] *msrsv3div* is listed in Table 2 but not in Wu and Huang (2014). As a matter of fact, it is the best performer in 2010 and includes 247,778 documents. This number is less than but not far away from 250,000. That is why *msrsv3div* was excluded that time but is taken this time.

In each year, fifty queries are divided into five groups: A (queries 1-10), B (queries 11-20), C (queries 21-30), D (queries 31-40), and E (queries 41-50). All possible combinations of four groups are used as training queries, while the remaining one group is used for fusion test. Here we use the five-fold cross validation method (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: $p * v$, $p^2 * v$, $p * v^2$, $p * c$, $p^2 * c$, $p * c^2$, $p * v * c$, etc.

Four recently proposed fusion methods are also involved for comparison. They are genetic algorithm-based fusion (GA) (Ghosh et al., 2015), differential evolution-based fusion (DE) (Xu et al., 2016), DDF (Liang et al., 2014), and ClustFuseCombSum (Kozorovitzky and Kurland, 2011). For GA, as in Ghosh et al. (2015), we set the population size as 30 and the maximum number of generations as 200. For DE, we set NP=30, F=0.5, CR=0.5, and the maximum number of generations as 200. For DDF, we set α=0.5, β=0.02, and the number of topics is 10. They are the same as in Liang et al.'s experiments. For ClustFuseCombSum, we set δ=10 and λ=0.7. The same δ value of 10 is used in Kozorovitzky and Kurland's experiment and they also recommend that the

[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 2
Information of three 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	TREC 2012
$MSRAACSF$	$msrs3div$	$uogTrA45Nmx2$	$DFalah120D$
(0.2144)	(0.3473)	(0.5284)	(0.4259)
$MSDiv3$	$THUIR10DvNov$	$msrs2011d1$	$DFalah121A$
(0.2048)	(0.3355)	(0.4994)	(0.4290)
$uogTrDYCcsB$	$ICTNETDV10R2$	$UWatMDSqltsr$	$QUTparaBline$
(0.1922)	(0.3222)	(0.4939)	(0.4185)
$UamsDancTFb1$	$uogTrB67xS$	$ICTNET11DVR3$	$uogTrA44xi$
(0.1774)	(0.2981)	(0.4764)	(0.4873)
$mudvimp$	$UMd10IASF$	$UAmsM705tFLS$	$uogTrA44xu$
(0.1746)	(0.2546)	(0.4378)	(0.5048)
$UCDSIFTdiv$	$cmuWi10D$	$uwBA$	$uogTrB44xu$
(0.1733)	(0.2484)	(0.3986)	(0.4785)
$NeuDiv1$	$UAMSD10aSRfu$	$CWIcIA2t5b1$	$utw2012c1$
(0.1705)	(0.2423)	(0.3487)	(0.4046)
$THUIR09AbClu$	$UCDSIFTDiv$	$liaQEWikiAnA$	$utw2012lm09$
(0.1665)	(0.2100)	(0.2287)	(0.4038)
Ave 0.1842	Ave 0.2823	Ave 0.4265	Ave 0.4440
Std 0.0176	Std 0.0502	Std 0.0991	Std 0.0399

value of λ should be between 0.6 and 0.8 for good fusion performance.

4.1 Data Fusion Results

As we know (Wu and McClean, 2006b), it is harder to get improvement over better component results through data fusion. However, the purpose of the experiment is to exam if we can obtain even better results by fusing a number of top-ranked results submitted.

Score normalization is necessary for data fusion to achieve good results. In this experiment the logarithmic function-based method is used to normalize scores of all component results. This method uses the formula $s(rank) = max\{1 - 0.2 * ln(rank), 0\}$ to generate scores for documents at each ranking

Table 3
Performance (measured by ERR-IA@20) of data fusion methods (figures in parentheses indicate the improvement rate of each method over the best component; figures in bold indicate the best three results in each column)

Method	2009	2010	2011	2012	Average
Best	0.2144	0.3473	0.5284	0.5048	0.3987
CombSum	0.2439 (+13.77%)	0.4110 (+18.35%)	0.5457 (+3.27%)	0.5096 (+0.95%)	0.4276 (+7.25%)
CombMNZ	**0.2506** (+16.89%)	0.3985 (+14.77%)	0.5422 (+2.63%)	0.4887 (-3.19%)	0.4200 (+5.34%)
p	0.2438 (+13.69%)	0.4135 (+19.08%)	0.5528 (+4.63%)	0.5055 (+0.14%)	0.4289 (+7.57%)
p^2	0.2450 (+14.25%)	0.4046 (+16.51%)	0.5630 (+6.55%)	0.5099 (+1.01%)	0.4306 (+8.01%)
c	**0.2480** (+15.67%)	**0.4160** (+19.80%)	0.5524 (+4.55%)	0.5069 (+0.42%)	0.4308 (+8.06%)
c^2	0.2472 (+15.31%)	0.4149 (+19.47%)	**0.5635** (+6.65%)	**0.5104** (+1.11%)	**0.4340** (+8.85%)
v	0.2453 (+14.41%)	0.4127 (+18.84%)	0.5442 (+1.05%)	0.5101 (+0.46%)	0.4281 (+7.37%)
v^2	0.2455 (+14.52%)	0.4138 (+19.18%)	0.5465 (+3.44%)	0.5038 (-0.20%)	0.4290 (+7.20%)
pc	**0.2478** (+15.57%)	0.4139 (+19.19%)	0.5633 (+6.62%)	0.5100 (+1.01%)	0.4338 (+8.80%)
p^2c	0.2458 (+14.65%)	0.4032 (+16.10%)	**0.5668** (+7.27%)	0.5078 (+1.01%)	0.4309 (+8.08%)
pv^2	0.2466 (+15.01%)	**0.4205** (+21.10%)	0.5496 (+4.01%)	0.5043 (-0.10%)	0.4303 (+7.93%)
pc^2v	0.2431 (+13.39%)	0.4094 (+17.90%)	**0.5669** (+7.29%)	**0.5170** (+2.42%)	**0.4341** (+8.88%)
pcv	0.2460 (+14.71%)	**0.4202** (+21.00%)	0.5599 (+5.98%)	**0.5113** (+1.29%)	**0.4344** (+8.95%)

Table 4
Performance (measured by α-nDCG@20) of data fusion methods (figures in parentheses indicate the improvement rate of each method over the best component; figures in bold indicate the best three results in each column)

Method	2009	2010	2011	2012	Average
Best	0.3653	0.4909	0.6298	0.6061	0.5230
CombSum	0.4016 (+9.96%)	0.5588 (+13.82%)	0.6610 (+4.97%)	0.6094 (+0.54%)	0.5577 (+6.63%)
CombMNZ	**0.4051** (+10.91%)	0.5502 (+12.09%)	0.6532 (+3.72%)	0.5922 (-2.29%)	0.5502 (+5.20%)
p	0.4009 (+9.75%)	0.5636 (+14.80%)	0.6668 (+5.88%)	0.6100 (+0.64%)	0.5603 (+7.14%)
p^2	0.4035 (+10.47%)	0.5558 (+13.22%)	0.6743 (+7.08%)	0.6137 (+1.25%)	0.5618 (+7.42%)
c	0.4020 (+10.04%)	0.5646 (+15.02%)	0.6662 (+5.79%)	0.6120 (+0.43%)	0.5612 (+7.30%)
c^2	**0.4059** (+11.12%)	0.5673 (+15.56%)	0.6745 (+7.11%)	**0.6146** (+1.40%)	0.5656 (+8.15%)
v	0.4031 (+10.35%)	0.5598 (+14.03%)	0.6561 (+4.19%)	0.6104 (+0.71%)	0.5574 (+6.58%)
v^2	0.4031 (+10.34%)	0.5597 (+14.02%)	0.6575 (+4.41%)	0.6069 (+0.13%)	0.5568 (+6.46%)
pc	**0.4072** (+11.49%)	**0.5677** (+15.64%)	**0.6748** (+7.15%)	**0.6146** (+1.40%)	**0.5661** (+8.24%)
p^2c	0.4049 (+10.84%)	0.5562 (+13.29%)	**0.6776** (+7.60%)	0.6145 (+1.39%)	0.5633 (+7.71%)
pv^2	0.4045 (+10.73%)	**0.5712** (+16.37%)	0.6609 (+4.95%)	0.6109 (+0.79%)	0.5619 (+7.44%)
pc^2v	0.4024 (+10.15%)	0.5633 (+14.76%)	**0.6766** (+7.44%)	**0.6205** (+2.38%)	**0.5657** (+8.16%)
pcv	0.4046 (+10.75%)	**0.5726** (+16.64%)	0.6724 (+6.77%)	**0.6154** (+1.53%)	**0.5663** (+8.28%)

position $rank$. It assigns positive scores {1, 0.8614, 0.7803, 0.7227,..., 0.0006} to top 148 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 Section 4.3.

For comparison, the best component result and two traditional data fusion methods, CombSum and CombMNZ, are used as baseline. Experimental results are shown in Tables 3 and 4. Two metrics, ERR-IA@20 and α-nDCG@20 ($\alpha = 0.5$), are used to evaluate all the fusion methods. From Tables 3 and 4, we can see that all the data fusion methods involved perform better than the best component result. Both CombSum and CombMNZ perform quite well, although CombSum is a little better than CombMNZ.

In this experiment, we apply three types of weights (performance, dissimilarity, and complementarity) separately with two options (linearly or squared). Most of the time using these weights we obtain better results than the best component result. It shows that using performance or complementarity weights can achieve better results than using dissimilarity weights. 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 both performance and complementarity weights, the square function is slightly better than the linear function.

Different types of combined weights are also tested. On average the combinations pcv, pc^2v, pc, and c^2 are very close in performance. They are better than the other schemes. They outperform CombSum and CombMNZ by 1% to 3% when either of the two measures is used. The improvement rates over the best component results are over 8% when either ERR-IA@20 or α-nDCG@20 is used.

In order to investigate the generalizability and robustness of these fusion methods, we carry out more experiments by the following procedure: from all the runs submitted to the web diversity task of TREC in the same year, we randomly select 3-20 runs to test the effectiveness of these methods. For any given number (3-20), 200 combinations are tested. Figures 1 and 2 present the results with the TREC 2010 web diversity data set, with metrics ERR-IA@20 and α-nDCG@20, respectively. From Figures 1 and 2, we can see that for both metrics CombSum is better than the best component results at sixteen points except four points (3, 4, 5, and 7). p, c, and v are better than CombSum, while pcv is the best. Similar results are observed for the other three data sets 2009, 2011, and 2012. Therefore, we do not present them.

Results are shown in Tables 5 and 6 for the comparison between our methods

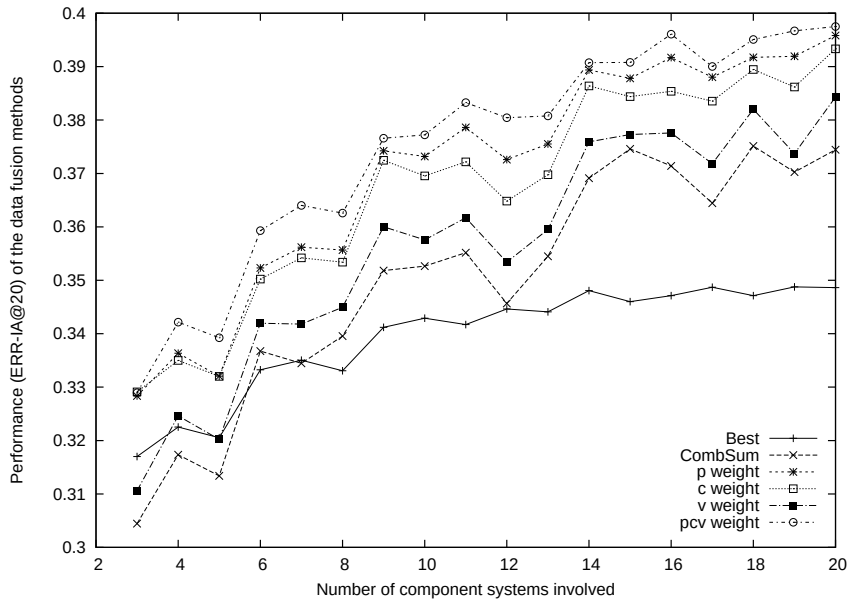

Fig. 1. Average performance (in ERR-IA@20) of the fusion methods with the TREC 2010 web diversity data set (3-20 component results, 200 combinations for each given number of component results)

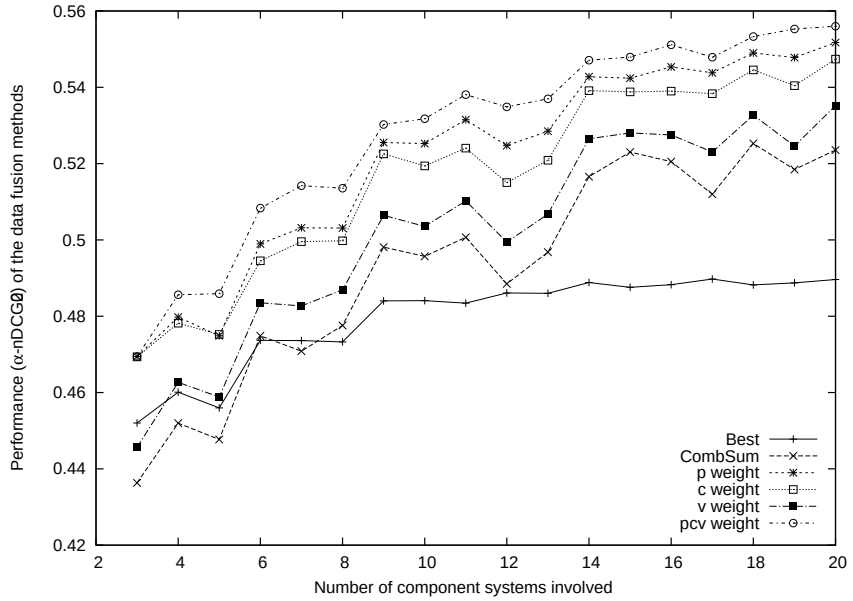

Fig. 2. Average performance (in α-nDCG@20) of the fusion methods with the TREC 2010 web diversity data set (3-20 component results, 200 combinations for each given number of component results)

and six other fusion methods that were proposed recently: ClustFuseCombSum (Kozorovitzky and Kurland, 2011), DDF (Liang et al., 2014), GA (Ghosh et al., 2015), $dis * p^2(Eq.1)$ (Wu and Huang, 2014), and DE (Xu et al., 2016). For convenience, one of our best methods *pc* is also shown in these two tables. We can see that *pc*, DE, and GA are close in performance, while DDF,

Table 5
Performance (measured by ERR-IA@20) of data fusion methods *pc* and four other methods (figures in bold indicate the best result in each column and a "*" symbol indicates the difference between that method and *pc* is significant at the 0.05 level)

Method	2009	2010	2011	2012	Average
pc	0.2478	0.4139	**0.5633**	0.5100	**0.4338**
GA	0.2474	**0.4204**	0.5472*	0.5091	0.4310
DE	**0.2551***	0.4067*	0.5571	**0.5108**	0.4324
DDF	0.2394*	0.4059*	0.5439*	0.5081	0.4243
ClustFuseCombSum	0.2393*	0.3585*	0.5157*	0.5025	0.4040
$dis * p^2(Eq.1)$	0.2492	0.3905*	0.5410*	0.4973*	0.4195

Table 6
Performance (measured by α-nDCG@20) of data fusion methods *pc* and four other methods (figures in bold indicate the best result in each column and a "*" symbol indicates the difference between that method and *pc* is significant at the 0.05 level)

Method	2009	2010	2011	2012	Average
pc	0.4072	0.5677	**0.6748**	0.6146	**0.5661**
GA	0.4067	**0.5718**	0.6655	0.6182	0.5656
DE	**0.4117**	0.5619	0.6691	**0.6190**	0.5654
DDF	0.3956*	0.5544*	0.6549*	0.6074	0.5531
ClustFuseCombSum	0.3924*	0.5071*	0.6293*	0.5926*	0.5303
$dis * p^2(Eq.1)$	0.4100	0.5515*	0.6477*	0.6012*	0.5526

$dis * p^2(Eq.1)$, and ClustFuseCombSum are not as good as the other three. A two-tailed T test is conducted to compare the difference between *pc* and the other methods. A "*" symbol in Tables 5 and 6 indicates that the difference between that method and *pc* is significant at the 0.05 level. In most cases, the difference between *pc* and either GA or DE is not significant, while the difference between *pc* and either of DDF, ClustFuseCombSum and $dis * p^2(Eq.1)$ is significant.

Finally, we test the time consumed by GA, DE, DDF, ClustFuseCombSum, and four representative weighting schemes of our method: p, v, c, and pcv. [3] Other weighting schemes that are not presented have almost the same time complexity as one of the presented. For example, p^2 is similar to p, v^2 is

[3] A personal computer with Intel Core i7 quad-core 3.4 GHz CPU and 32 GB of RAM is used for the test.

Table 7
Time in seconds for training (40 queries) and run-time (per query) of several fusion methods (CF denotes ClustFuseCombSum)

Method	TREC 2009		TREC 2010		TREC 2011	
	training	run-time	training	run-time	training	run-time
p	1.30	0.0074	1.16	0.0075	1.19	0.0074
v	0.25	0.0075	0.24	0.0077	0.23	0.0078
c	7.06	0.0074	7.30	0.0074	7.21	0.0075
pcv	8.66	0.0074	8.82	0.0074	8.79	0.0075
GA	169.75	0.0074	168.31	0.0075	169.52	0.0075
DE	180.33	0.0075	179.58	0.0075	181.02	0.0075
DDF	-	15,379	-	13,478	-	18,986
CF	-	93.03	-	94.08	-	93.96

similar to v, pc^2v is similar to pcv, and so on. Table 7 shows the result. In those four weighting schemes, pcv needs more time than p, c, and v. All of the four weighting schemes of our method, GA, and DE need more or less the same time for fusion, but GA and DE need more time for training than our methods do. For example, GA requires almost 20 times as much time as pcv does. DDF and ClustFuseCombSum are very different from GA, DE, and the methods presented in this paper. Both of them do not need training but take long time in the fusion process, although DDF needs much longer time than ClustFuseCombSum does.

4.2 Three Types of Weights and Their Relationships

In this subsection we further investigate the relationship between three types of weights. We still look at the eight runs for each of the three groups listed in Table 2. As in the experiment presented in Section 4.1, three types of weights are calculated five times for each group of component results due to the five-fold cross validation method used. To estimate the similarity of a pair of weighting schemes, we 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 8 shows the results. The distance between p and c is always much shorter than the distance between c and v, or between p and v. If we regard each weighting scheme as a point in a space and use a triangle to represent the relationship (distance) of them, then the triangle is roughly an isosceles one. The distance between the dissimilarity weighting and the other two are long,

Table 8
Euclidean's distance between two different types of weights (average of 5 groups)

Pair of Weights	2009	2010	2011	2012	Average
c and v	0.0960	0.1180	0.2363	0.3362	0.1966
p and v	0.1104	0.1614	0.2497	0.3572	0.2197
p and c	0.0334	0.0670	0.0300	0.2845	0.1037

Table 9
Different types of weights for three results (average of five groups, the figures in parentheses indicate the normalized weights by setting the weight of MSRAACSF to 1)

Type	MSRAACSF	UamsDancTFb1	mudvimp
p	0.2144(1)	0.1774(0.8274)	0.1746(0.8142)
c	0.5512(1)	0.4554(0.8261)	0.4129(0.7492)
v	0.8925(1)	0.9183(1.0289)	0.8938(1.0014)

Table 10
Performance of data fusion with three results (figures in bold indicate the top three values in the column)

Method	ERR-IA@20	α-nDCG@20
CombSum	0.2097	0.3373
p	0.2129	0.3432
p^2	0.2184	0.3490
c	0.2129	0.3433
c^2	**0.2188**	**0.3503**
v	0.2092	0.3369
v^2	0.2093	0.3364
pc	**0.2189**	**0.3501**
pcv	**0.2186**	**0.3496**

while the performance weighting and the complementarity weighting are close. As we already see, the dissimilarity weights is not as useful as the performance weights and complementarity weights. This phenomenon suggests that the area around performance and complementarity weighting is a profitable one for fusion performance improvement.

In Section 3.2 we used a few toy examples to illustrate the concept of complementarity of results on different subtopics and its effect on data fusion. In this subsection we take a real example to further explore this.

From those eight submitted results in 2009, we choose three of them for further investigation. They are *MSRAACSF* (Dou et al., 2009), *UamsDancTFb1* (Anh and Moffat, 2010), and *mudvimp* (Kaptein et al., 2009). They are chosen because they are implemented by using different retrieval models. *MSRAACSF* is implemented using WebStudio (an augmented BM25 model) with a clustering-based implicit diversification method, *UamsDancTFb1* is implemented using Indri (a combination of the language modeling and inference network) and a document similarity-based implicit diversification method, while *mudvimp* is implemented using IMP (a variation of the vector-space model), but no diversification method is used. Another difference is: both *MSRAACSF* and *mudvimp* use "ClueWeb09", while *UamsDancTFb1* uses "ClueWeb09B", which is a subset of "ClueWeb09". Interestingly, all of them index anchor text of all the web documents for the search task.

After calculating each result's complementarity of subtopic coverage to another result using Equations 5 and 6, we obtain: $c_a(b)$: 0.5621, $c_a(c)$: 0.5403, $c_b(a)$: 0.4163, $c_b(c)$: 0.4944, $c_c(a)$: 0.3542, $c_c(b)$: 0.4717. Here a, b, and c denotes *MSRAACSF*, *UamsDancTFb1*, and *mudvimp*, respectively. We can see that *MSRAACSF* obtains the heaviest weight, *UamsDancTFb1* is in the second place, and *mudvimp* obtains the lightest. Two other types of weights are also calculated, and all the weights are shown in Table 9. In this example, performance weights and complementarity weights are strongly correlated, while dissimilarity is quite different from the other two. Because all three dissimilar weights are very close to each other, their power for distinguishing systems/results is limited.

Using the same methodology of five-fold cross validation, we fuse these three results by CombSum and linear combination. The results are shown in Table 10. Not surprisingly, using c or c^2 can achieve similar fusion performance as using p or p^2. pc, c^2 and pcv are the top three. This is consistent with the experimental results reported in Section 4.1.

4.3 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 $score(i) = max\{1 - 0.2 * ln(i), 0\}$ 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.

Table 11
Estimation accuracy of the two rank-to-score converting models

Group	Logarithmic		Reciprocal	
	R^2	F	R^2	F
2009	0.907	959.079	0.900	879.491
2010	0.926	1226.342	0.890	794.127
2011	0.959	2312.175	0.936	1424.361
2012	0.957	2168.538	0.952	1954.654

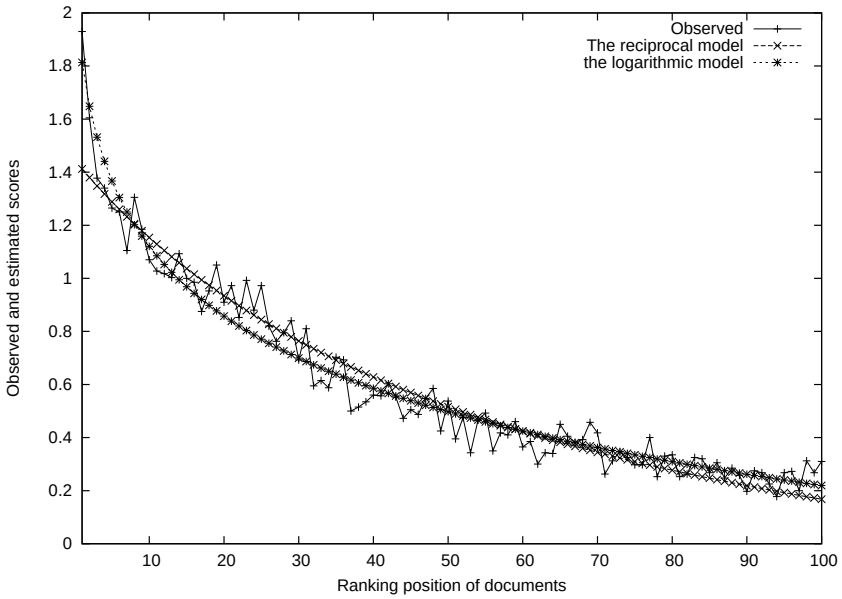

Fig. 3. Score estimation of two different models with the TREC 2011 web diversity data set

For those top-ranked documents, it is possible that some of them are relevant to multiple subtopics. Thus we hypothesize that the reciprocal model does not fit well on some top-ranked documents, while the logarithmic model can be a better option for this.

Firstly, we try to see which of them fits the observed data better. The procedure is as follows: consider eight runs in the same year in Table 2 together, 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. Over all fifty queries and all eight runs, scores are averaged for 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 11 shows the results.

From Table 11 we can see that in all four years, the logarithmic model is slightly more accurate than the reciprocal model with larger R^2 and F values.

To have a closer look at how these two models perform, we depict the observed curve and the two estimated curves for the data of 2011 group in Figure 3. [4] The curves for the other three groups are not presented due to similarity. In Figure 3, 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) is not large enough to stabilize them. In Figure 3, we can also see that both logarithmic and reciprocal models fit the observed curve quite well. However, we can observe that 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 the 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 performance 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 12 and 13.

From Tables 12 and 13 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, the logarithmic model is slightly better than the reciprocal model. If we compare them per year, then the reciprocal model performs better than the logarithmic model in 2009, while it is worse than the logarithmic model in the other three years. This is mainly because in 2009, fewer documents are relevant to multiple subtopics as in the other three years. 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.

5 Conclusions

In this article we have reported our investigation on search result diversification via data fusion. We focus on the linear combination method in which weight assignment is a key issue. In order to achieve better fusion results,

[4] In Figure 3, 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 12
Performance (measured by ERR-IA@20) of data fusion using 3 different score normalization methods (figures in bold indicate the highest performance for a specific fusion method)

Fusion method	Logarithmic	Reciprocal	Zero-one
CombSum	**0.4276**	0.4266	0.3990
CombMNZ	0.4200	**0.4205**	0.3827
p	**0.4289**	0.4208	0.3983
p^2	**0.4306**	0.4172	0.3974
c	**0.4308**	0.4211	0.3987
c^2	**0.4340**	0.4196	0.3976
v	**0.4281**	0.4243	0.3983
v^2	**0.4290**	0.4247	0.3978
pc	**0.4338**	0.4175	0.3973
p^2c	**0.4309**	0.4215	0.3958
pc^2	**0.4316**	0.4197	0.3961
pv	**0.4296**	0.4195	0.3982
pv^2	**0.4303**	0.4185	0.3980
p^2v	**0.4307**	0.4180	0.3989
p^2cv	**0.4316**	0.4203	0.3985
pc^2v	**0.4341**	0.4200	0.3972
pcv	**0.4344**	0.4189	0.3976
Average	0.4303	0.4205	0.3969
	±0.0%	-2.28%	-7.76%

complementarity of results on subtopic coverage has been identified as an important factor that can be used to decide the weight of a component search system. Using it alone or combining it with other factors, such as performance and dissimilarity, can achieve very good results.

Experiments with four 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 and dissimilarity outperform the others on average. Our experiments also demonstrate that the logarithmic model is very likely better than the reciprocal model for converting rank information into scores.

Table 13
Performance (measured by α-nDCG@20) of data fusion using 3 different score normalization methods (figures in bold indicate the highest performance for a specific fusion method)

Fusion method	Logarithmic	Reciprocal	Zero-one
CombSum	**0.5577**	0.5562	0.5233
CombMNZ	**0.5502**	0.5443	0.5055
p	**0.5603**	0.5540	0.5235
p^2	**0.5618**	0.5516	0.5233
c	**0.5612**	0.5542	0.5230
c^2	**0.5656**	0.5532	0.5227
v	**0.5574**	0.5545	0.5228
v^2	**0.5568**	0.5550	0.5233
pc	**0.5661**	0.5524	0.5224
p^2c	**0.5633**	0.5553	0.5229
pc^2	**0.5639**	0.5530	0.5232
pv	**0.5611**	0.5527	0.5229
pv^2	**0.5619**	0.5528	0.5237
p^2v	**0.5629**	0.5530	0.5242
p^2cv	**0.5619**	0.5541	0.5248
pc^2v	**0.5657**	0.5533	0.5234
pcv	**0.5663**	0.5535	0.5231
Average	0.5622	0.5531	0.5222
	±0.0%	-1.62%	-7.11%

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. However, there are conditions for our methods to work more effectively than CombSum and CombMNZ: either some of the component results are more effective than the others or subtopics are unevenly covered by all the component results. If the component results are generated by search systems with diversified technologies, then it is very likely that the methods proposed in this paper are able to achieve better performance. On the other hand, one deep question is: why the data fusion methods work for diversified results as well as for relevance-oriented results? In this article we have addressed it by the introduction of complementarity weights in Section

3.2 and by some analysis of the three types of weights in Section 4.2. Some more theoretical work is desirable and it remains to be our future work.

6 Acknowledgement

This research has been partially supported by Natural Science Foundation of Jiangsu Province of China (number BK20171303).

References

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

Alam, F., Mehmood, R., Katib, I., Albogami, N. N., Albeshri, A., 2017. Data fusion and iot for smart ubiquitous environments: A survey. IEEE Access 5, 9533–9554.

Anh, V., Moffat, A., November 2010. The role of anchor text in clueweb09 retrieval. In: Proceedings of The Nineteenth Text REtrieval Conference, TREC 2010. Gaithersburg, Maryland, USA.

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

Azarbonyad, H., Dehghani, M., Kenter, T., Marx, M., Kamps, J., de Rijke, M., 2017. Hierarchical re-estimation of topic models for measuring topical diversity. In: Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings. pp. 68–81.

Bigot, A., Chrisment, C., Dkaki, T., Hubert, G., Mothe, J., 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), 617–648.

Calvé, A. L., Savoy, J., 2000. Database merging strategy based on logistic regression. Information Processing & Management 36 (3), 341–359.

Capannini, G., Nardini, F. M., Perego, R., Silvestri, F., 2011. Efficient diversification of web search results. PVLDB 4 (7), 451–459.

Carbonell, J., Goldstein, J., August 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, pp. 335–336.

Carterette, B., Chandar, P., November 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, pp. 1287–1296.

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

Chen, G., Ye, D., Xing, Z., Chen, J., Cambria, E., 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017. pp. 2377–2383.

Clarke, C., Craswell, N., Soboroff, I., Voorhees, E., November 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.

Clarke, C., Kolla, M., Cormack, G., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I., July 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, pp. 659–666.

Clarke, C. L. A., Craswell, N., Soboroff, I., November 2009. Overview of the trec 2009 web track. In: Proceedings of The Eighteenth Text REtrieval Conference. Gaithersburg, Maryland, USA.

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

Dang, V., Croft, W. B., August 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, pp. 65–74.

Deng, T., Fan, W., 2014. On the complexity of query result diversification. ACM Transactions on Database Systems 39 (2), 15.

Dou, Z., Chen, K., Song, R., Ma, Y., Shi, S., Wen, J., November 2009. Microsoft research asia at the web track of TREC 2009. In: Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009. Gaithersburg, Maryland, USA.

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

Fox, E. A., Koushik, M. P., Shaw, J., Modlin, R., Rao, D., March 1993. Combining evidence from multiple searches. In: The First Text REtrieval Conference (TREC-1). Gaitherburg, MD, USA, pp. 319–328.

Ghosh, K., Parui, S. K., Majumder, P., 2015. Learning combination weights in data fusion using genetic algorithms. Information Processing & Manage-

ment 51 (3), 306–328.

Gupta, D., Berberich, K., 2016. Diversifying search results using time - an information retrieval method for historians. In: Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings. pp. 789–795.

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

Hu, S., Dou, Z., Wang, X., Sakai, T., Wen, J., 2015. Search result diversification based on hierarchical intents. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015. pp. 63–72.

Ionescu, B., Popescu, A., Radu, A., Müller, H., 2016. Result diversification in social image retrieval: a benchmarking framework. Multimedia Tools Appl. 75 (2), 1301–1331.

Jiang, Z., Wen, J., Dou, Z., Zhao, W., Nie, J., Yue, M., 2017. Learning to diversify search results via subtopic attention. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017. pp. 545–554.

Kaliciak, L., Myrhaug, H. I., Göker, A., Song, D., 2014. On the duality of specific early and late fusion strategies. In: 17th International Conference on Information Fusion, FUSION 2014, Salamanca, Spain, July 7-10, 2014. pp. 1–8.

Kaptein, R., Koolen, M., Kamps, J., November 2009. Result diversity and entity ranking experiments: Anchors, links, text and wikipedia. In: Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009. Gaithersburg, Maryland, USA.

Kohavi, R., August 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, pp. 1137–1145.

Kozorovitzky, A. K., Kurland, O., July 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, pp. 893–902.

Lahat, D., Adali, T., Jutten, C., 2015. Multimodal data fusion: An overview of methods, challenges, and prospects. Proceedings of the IEEE 103 (9), 1449–1477.

Lan, Z., Bao, L., Yu, S., W., Hauptmann, A. G., 2012. Double fusion for multimedia event detection. In: Advances in Multimedia Modeling - 18th International Conference, MMM 2012, Klagenfurt, Austria, January 4-6, 2012. Proceedings. pp. 173–185.

Lee, J. H., July 1997. Analysis of multiple evidence combination. In: Proceedings of the 20th Annual International ACM SIGIR Conference. Philadelphia,

Pennsylvania, USA, pp. 267–275.
Li, J., Fong, S., Wong, R. K., Chu, V. W., 2018. Adaptive multi-objective swarm fusion for imbalanced data classification. Information Fusion 39, 1–24.
Li, J., Liu, C., Liu, B., Mao, R., Wang, Y., Chen, S., Yang, J., H., Wang, Q., 2015. Diversity-aware retrieval of medical records. Computers in Industry 69, 81–91.
Li, J., Wu, Y., Zhang, P., Song, D., Wang, B., 2017. Learning to diversify web search results with a document repulsion model. Inf. Sci. 411, 136–150.
Liang, S., Ren, Z., de Rijke, M., July 2014. Fusion helps diversification. In: Proceedings of the 37th Annual International ACM SIGIR Conference. Cold Coast, QLD, Australia, pp. 303–312.
Liu, T., 2011. Learning to Rank for Information Retrieval. Springer.
Macdonald, C., Ounis, I., November 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, pp. 387–396.
Montague, M., Aslam, J. A., November 2001. Relevance score normalization for metasearch. In: Proceedings of ACM CIKM Conference. Berkeley, USA, pp. 427–433.
Montague, M., Aslam, J. A., November 2002. Condorcet fusion for improved retrieval. In: Proceedings of ACM CIKM Conference. McLean, VA, USA, pp. 538–548.
Naini, K. D., Altingovde, I. S., Siberski, W., 2016. Scalable and efficient web search result diversification. ACM Transactions on the Web 10 (3), 15:1–15:30.
Ozdemiray, A. M., Altingovde, I. S., 2015. Explicit search result diversification using score and rank aggregation methods. Journal of the Association for Information Science and Technology 66 (6), 1212–1228.
Rafiei, D., Bharat, K., Shukla, A., April 2010. Diversifying web search results. In: Proceedings of the 19th International Conference on World Wide Web. Raleigh, North Carolina, USA, pp. 781–790.
Santos, R., Macdonald, C., Ounis, I., April 2010. Exploiting query reformulations for web search result diversification. In: Proceedings of the 19th International Conference on World Wide Web. Raleigh, North Carolina, USA, pp. 881–890.
Santos, R. L. T., MacDonald, C., Ounis, I., 2015. Search result diversification. Foundations and Trends in Information Retrieval 9 (1), 1–90.
Schedl, M., Hauger, D., 2015. Tailoring music recommendations to users by considering diversity, mainstreaminess, and novelty. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015. pp. 947–950.
Thang, D. C., Tam, N. T., Hung, N. Q. V., Aberer, K., 2015. An evaluation of diversification techniques. In: Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September

1-4, 2015, Proceedings, Part II. pp. 215–231.

Ullah, M. Z., Shajalal, M., Chy, A. N., Aono, M., 2016. Query subtopic mining exploiting word embedding for search result diversification. In: Information Retrieval Technology - 12th Asia Information Retrieval Societies Conference, AIRS 2016, Beijing, China, November 30 - December 2, 2016, Proceedings. pp. 308–314.

Vieira, M. R., Razente, H. L., Barioni, M. C. N., Hadjieleftheriou, M., Srivastava, D., Jr., C. T., Tsotras, V. J., 2011. On query result diversification. In: Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany. pp. 1163–1174.

Vogt, C. C., Cottrell, G. W., August 1998. Predicting the performance of linearly combined IR systems. In: Proceedings of the 21st Annual ACM SIGIR Conference. Melbourne, Australia, pp. 190–196.

Wang, J., Zhu, J., July 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, pp. 115–122.

Wang, X., Dou, Z., Sakai, T., Wen, J., 2016a. Evaluating search result diversity using intent hierarchies. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016. pp. 415–424.

Wang, Y., Luo, Z., Yu, Y., 2016b. Learning for search results diversification in twitter. In: Web-Age Information Management - 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part II. pp. 251–264.

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

Wu, S., January 2012a. Applying the data fusion technique to blog opinion retrieval. Expert Systems with Applications 39 (1), 1346–1353.

Wu, S., 2012b. Data Fusion in Information Retrieval. Springer.

Wu, S., Bi, Y., Zeng, X., Han, L., July 2009. Assigning appropriate weights for the linear combination data fusion method in information retrieval. Information Processing & Management 45 (4), 413–426.

Wu, S., Crestani, F., 2015. A geometric framework for data fusion in information retrieval. Information Systems 50, 20–35.

Wu, S., Crestani, F., Bi, Y., October 2006. Evaluating score normalization methods in data fusion. In: Proceedings of the 3rd Asia Information Retrieval Symposium (LNCS 4182). Singapore, pp. 642–648.

Wu, S., Huang, C., July 2014. Search result diversification via data fusion. In: Proceedings of the 37th Annual International ACM SIGIR Conference. Cold Coast, QLD, Australia, pp. 827–830.

Wu, S., McClean, S., December 2006a. Improving high accuracy retrieval by eliminating the uneven correlation effect in data fusion. Journal of American Society for Information Science and Technology 57 (14), 1962–1973.

Wu, S., McClean, S., July 2006b. Performance prediction of data fusion for information retrieval. Information Processing & Management 42 (4), 899–915.

Wu, Y., Li, J., Zhang, P., Song, D., 2016. Learning to improve affinity ranking for diversity search. In: Information Retrieval Technology - 12th Asia Information Retrieval Societies Conference, AIRS 2016, Beijing, China, November 30 - December 2, 2016, Proceedings. pp. 335–341.

Xia, L., Xu, J., Lan, Y., Guo, J., Cheng, X., 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015. pp. 113–122.

Xia, L., Xu, J., Lan, Y., Guo, J., Cheng, X., 2016. Modeling document novelty with neural tensor network for search result diversification. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016. pp. 395–404.

Xia, L., Xu, J., Lan, Y., Guo, J., Zeng, W., Cheng, X., 2017. Adapting markov decision process for search result diversification. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017. pp. 535–544.

Xu, C., Huang, C., Wu, S., 2016. Differential evolution-based fusion for results diversification of web search. In: Web-Age Information Management - 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part I. pp. 429–440.

Xu, C., Wu, S., 2017. The early fusion strategy for search result diversification. In: Proceedings of the ACM Turing 50th Celebration Conference - China, TUR-C 2017, Shanghai, China, May 12-14, 2017. pp. 47:1–47:6.

Zhai, C., Cohen, W., Lafferty, J., August 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, pp. 10–17.

Zheng, W., Fang, H., September-October 2013. A diagnostic study of search result diversification methods. In: Proceedings of International Conference on the Theory of Information Retrieval, ICTIR '13. Copenhagen, Denmark, p. 17.

Zuccon, G., Azzopardi, L., Zhang, D., Wang, J., 2012. Top-k retrieval using facility location analysis. In: Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings. pp. 305–316.