Fusion Helps Diversification

Shangsong Liang
University of Amsterdam
Amsterdam, The Netherlands
s.liang@uva.nl

Zhaochun Ren
University of Amsterdam
Amsterdam, The Netherlands
z.ren@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

A popular strategy for search result diversification is to first retrieve a set of documents utilizing a standard retrieval method and then rerank the results. We adopt a different perspective on the problem, based on data fusion. Starting from the hypothesis that data fusion can improve performance in terms of diversity metrics, we examine the impact of standard data fusion methods on result diversification. We take the output of a set of rankers, optimized for diversity or not, and find that data fusion can significantly improve state-of-the art diversification methods. We also introduce a new data fusion method, called diversified data fusion, which infers latent topics of a query using topic modeling, without leveraging outside information. Our experiments show that data fusion methods can enhance the performance of diversification and DDF significantly outperforms existing data fusion methods in terms of diversity metrics.

Categories and Subject Descriptors

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

Keywords

Data fusion; rank aggregation; diversification; ad hoc retrieval

1. INTRODUCTION

Search result diversification is widely being studied as a way of tackling query ambiguity. Instead of trying to identify the "correct" interpretation behind a query, the idea is to make the search results diversified so that users with different backgrounds will find at least one of these results to be relevant to their information need [2]. In contrast to the traditional assumption of independent document relevance, search result diversification approaches typically consider the relevance of a document in light of other retrieved documents [40]. Diversification models try to identify the probable "aspects" of the query and return documents for each aspect, thereby making the result list more diverse.

Data fusion approaches, also called rank aggregation approaches, consist in combining result lists in order to produce a new and hopefully better ranking [16, 42]. Here, results lists can be produced by a wide range of ranking approaches, based, e.g., on different query or document representations. Data fusion methods can improve retrieval performance in terms of traditional relevance-oriented metrics like MAP and precision@k over the methods used to generate the individual result lists being fused [17, 26, 27, 49]. One reason is that retrieval approaches often return very different non-relevant documents, but many of the same relevant documents [49].

We examine the hypothesis that *data fusion can improve performance in terms of diversity metrics by promoting aspects that are found in disparate ranked lists to the top of the fused list*. Our first step in testing this hypothesis is to examine the impact of existing data fusion methods in terms of diversity scores when fusing ranked lists. We find that they tend to improve over individual component runs on nearly all of the diversity metrics that we consider: Prec-IA, MAP-IA, α-NDCG, ERR-IA (all at rank 20).

Building on these findings we propose a new data fusion method, called diversified data fusion (DDF). Based on latent Dirichlet allocation (LDA), it operates on documents in the result lists to be fused, whether the result lists have been diversified or not. DDF infers latent topics, their probabilities of being relevant and a multinomial distribution of topics over the documents being fused. Thus, it integrates topic structure and rank information. DDF does not assume the explicit availability of query aspects, but infers these as well as the latent prior for a given query via the documents being fused. Experimental results show that DDF can aggregate result lists—whether produced by diversification or ad hoc retrieval models—and boost the diversity of the final fused list, outperforming state-of-the-art diversification methods and established data fusion methods, especially in terms of intent-aware precision metrics.

Our contributions in this paper can be summarized as follows:

i. We tackle the challenge of search result diversification in a novel way by using data fusion methods.

ii. We propose a novel data fusion method that aims at optimizing diversification measures and that proves to be especially effective in terms of intent-aware precision metrics.

iii. We analyze the effectiveness of data fusion for result diversification and find that our fusion method as well as other fusion methods can significantly outperform state-of-the-art diversification methods.

§2 discusses related work. §3 describes the fusion models that we use (old and new). §4 describes our experimental setup. §5 is devoted to our experimental results and we conclude in §6.

2. RELATED WORK

We distinguish between three directions of related work: search result diversification, data fusion, and latent topic modeling.

SIGIR '14, July 06–11, 2014, Gold Coast, QLD, Australia.

ACM 978-1-4503-2257-7/14/07 ... $15.00.
http://dx.doi.org/10.1145/2600428.2609561

2.1 Search result diversification

Search result diversification is similar to ad hoc search, but differs in its judging criteria and evaluation measures [8, 12]. The basic premise in search result diversification is that the relevance of a set of documents depends not only on the individual relevance of its members, but also on how they relate to one another [2]. Ideally, users can find at least one relevant document to the underlying information need. Most previous work on search result diversification can be classified as either *implicit* or *explicit* [39, 41].

Implicit approaches to result diversification promote diversity by selecting a document that differs from the documents appearing before it in terms of vocabulary, as captured by a notion of document similarity, such as cosine similarity or Kullback-Leibler divergence. Carbonell and Goldstein [6] propose the *maximal marginal relevance* (MMR) method, which reduces redundancy while maintaining query relevance when selecting a document. Chen and Karger [7] describe a retrieval method incorporating negative feedback in which documents are assumed to be non-relevant once they are included in the result list, with the goal of maximizing diversity. Zhai et al. [51] present a subtopic retrieval model where the utility of a document in a ranking is dependent on other documents in the ranking and documents that cover many different subtopics of a query topic are found. Other implicit work includes, e.g., [1] where set-based recommendation of diverse articles is proposed. We also tackle the problem of search result diversification implicitly, but in a different way, i.e., by data fusion.

Explicit approaches to diversification assume that a set of query aspects is available and return documents for each of them. Past work has shown that explicit approaches are usually somewhat superior to implicit diversification techniques. Well-known examples include xQuAD [39], RxQuAD [45], IA-select [2], PM-2 [13], and, more recently, DSPApprox [14]. Instead of modeling a set of aspects implicitly, these algorithms obtain the set of aspects either manually, e.g., from aspect descriptions [8, 12], or they create them directly from, e.g., suggested queries generated by commercial search engines [13, 39] or predefined aspect categories [44]. We propose an implicit fusion-based diversification model where we do not assume that the aspects of the query are available but do assume that we can infer the underlying topics and the prior relevance of each topic for search result diversification.

2.2 Data fusion

A core concern in data fusion is how to assign a score to a document that appears in one of the lists to be fused [17, 19, 42, 49]. Most previous work on data fusion focuses on optimizing a traditional evaluation metric, like MAP, p@k and nDCG. Fusion approaches can be categorized into supervised or unsupervised: Supervised data fusion approaches, like λ-Merge [43], first extract a number of features, either from documents or lists, and then utilize a machine learning algorithm to train the fusion model [15, 17, 49].

In contrast, unsupervised data fusion methods mainly use either retrieval scores or ranks of documents in the lists to be merged, with the CombSUM family of fusion methods being the oldest and one of the most successful ones in many information retrieval tasks [26, 42]. State-of-the-art data fusion methods ClustFuseCombSUM and ClustFuseCombMNZ (both cluster-based methods) are proposed in [23]. Methods utilizing retrieval scores take score information from the lists to be fused as input, while those utilizing rank information only use order information of the documents appearing in the lists to be fused as input. Data fusion methods utilizing rank information have many uses and applications in information retrieval, including, e.g., expert search [30, 35], query reformulations [43], meta-search [4, 17] and microblog search [31, 32].

We do not make the assumption that labeled data is available but integrate standard unsupervised data fusion information into our diversified fusion model for search result diversification via a latent topic model.

2.3 Topic modeling

Topic models have been proposed for reducing the high dimensionality of words appearing in documents into low-dimensional "latent topics." From the first work on topic models [21], the Probablistic LSI model, topic models have received significant attention [5, 18, 22] and have proved to be effective in many information retrieval tasks [24, 47, 50]. Latent dirichlet allocation (LDA) [5] represents each document as a finite mixture over "latent" topics where each topic is represented as a finite mixture over words existing in that document. Based on LDA, many extensions have been proposed, e.g., to handle users' connections with particular documents and topics [37], to learn relations among different topics [25, 29], for topic over time [46], for dynamic mixture model [48], or tweet summarization [36]. LDA has also been extended to sentiment analysis [28]. We propose a novel topic model where fusion scores of each document appearing in lists to be fused are used to boost the performance of state-of-the-art diversification methods.

Our work adds the following to the work discussed above. We propose a fusion-based approach to the search result diversification task. We find that existing unsupervised fusion methods significantly outperform state-of-the-art diversification methods. In addition, we propose a novel fusion method, diversified data fusion, that uses the output of a fusion step and a topic modeling step as input to a diversification step. To the best of our knowledge, ours is the first attempt to utilize data fusion for diversification.

3. FUSION METHODS

We first review our notation and terminology. Then we introduce the task to be addressed, as well as the baseline fusion methods that we use in this paper plus a new fusion method.

3.1 Notation and terminology

We summarize the main notation used in this paper in Table 1. In the remainder, we distinguish between queries, aspects and topics. A *query* is an expression of an information need; in our experimental evaluation below, queries are provided as part of a TREC test collection. An *aspect* (sometimes called subtopic at the TREC Web track) is an interpretation of an information need. We use *topic* to refer to latent topics as identified by a topic modeling method, in our case LDA. A *component list* is a ranked list that serves as input for a data fusion method. A *fused* list is a list that is the result of applying a fusion method to component lists.

3.2 The diversified data fusion task

The diversified data fusion task that we address is this: given a query, an index of documents, and a set of ranked lists of documents produced in response to a query, aggregate the lists into a final result list where documents should be diversified. The component lists may or may not have been diversified themselves or ranked by relevance only.

The underlying data fusion problem consists of running a ranking function F_{X} that satisfies:

$$\mathbf{L} = \{L_1, L_2, \ldots, L_m\}, q, \mathcal{C} \xrightarrow{F_{\mathrm{X}}} L_f,$$

where $\mathbf{L}$ is a set of components lists, $m = |\mathbf{L}|$ is their number, $\mathcal{C}$ the document corpus, q a query, and L_f the final fused list.

Table 1: Basic notation used in the paper.

Notation	Gloss
$\mathcal{C}$	document corpus
q	query
z	topic
d	document
w	a token
N_d	number of tokens in d
L_i	i-th ranked list of documents
$\mathbf{L}$	set of ranked lists to be fused
m	number of ranked lists to be fused, i.e., $m = \|\mathbf{L}\|$
$\mathcal{C}_\mathbf{L}$	set of documents that appear in the lists $\mathbf{L}$
$\|\mathcal{C}_\mathbf{L}\|$	number of documents in $\mathcal{C}_\mathbf{L}$
F_X	a data fusion method
$F_\mathrm{X}(d;q)$	score of document d for query q according to a data fusion method F_X
$\mathfrak{R}_{L_i d}$	rank-based score of d in list L_i
$\mathrm{rank}(d, L_i)$	rank of d in list L_i
$\|L_i\|$	length of list L_i
R	set of top ranked documents
$qt[z\|q]$	quotient score for z given q in PM-2 algorithm [13]
$v_{z\|q}$	probability of z given q
$s_{z\|q}$	"portion" of seat occupied by z given q in PM-2
λ	a free trade-off parameter in PM-2
α	the parameter of topic Dirichlet prior
β	the parameter of token Dirichlet prior
T	number of topics
V	number of unique tokens in $\mathcal{C}_\mathbf{L}$
θ_d	multinomial distribution of topics specific to d
ϕ_z	multinomial distribution of tokens specific to topic z
μ_z	mean of Log-normal distribution of fusion scores for topic z
σ_z	deviation of Log-normal distribution of fusion scores for z
z_{di}	topic associated with the i-th token in the document d
w_{di}	i-th token in document d
f_{di}	fusion score for token w_{di}

3.3 Baseline data fusion methods

Let $\mathfrak{R}_{L_i d}$ denote the score of document d based on the rank of d in list L_i; in the literature on data fusion, one often finds $\mathfrak{R}_{L_i d} = 0$ if $d \notin L_i$ (d still in the combined set of documents $\mathcal{C}_\mathcal{L} := \bigcup_{i=1}^{m} L_i$). In both CombSUM and CombMNZ, $\mathfrak{R}_{L_i d}$ is often defined as:

$$\mathfrak{R}_{L_i d} = \begin{cases} \frac{(1+|L_i|)-\mathrm{rank}(d,L_i)}{|L_i|} & d \in L_i \\ 0 & d \notin L_i, \end{cases} \quad (1)$$

where $|L_i|$ is the length of L_i and $\mathrm{rank}(d, L_i) \in \{1, \ldots, |L_i|\}$ is the rank of d in L_i. The well-known CombSUM fusion method [17, 49], for instance, scores d by the sum of its rank scores in the lists:

$$F_\mathrm{CombSUM}(d;q) := \textstyle\sum_{L_i} \mathfrak{R}_{L_i d},$$

while CombMNZ [17, 49] rewards d that ranks high in many lists:

$$F_\mathrm{CombMNZ}(d;q) := |\{L_i : d \in L_i\}| \cdot F_\mathrm{CombSUM}(d;q),$$

where $|\{L_i : d \in L_i\}|$ is the number of lists in which d appears.

We consider CombSUM, CombMNZ and two state-of-the-art data fusion methods, ClustFuseCombSUM and ClustFuseCombMNZ [23], that integrate cluster information into CombSUM and CombMNZ, respectively, as baseline fusion methods.

In addition, a natural and direct way of diversifying a result list in the setting of data fusion is this: first rank the documents in the component lists by their estimated relevance to the query through a standard data fusion method, such as CombSUM, and then diversify the ranking through effective search result diversification models, such as MMR [6] and PM-2 [13]. In our experiments, we implement two more baselines, called CombSUMMMR and CombSUMPM-2. They first use CombSUM to obtain a fused list and then use MMR and PM-2, respectively, to diversify the list.

3.4 Diversified data fusion

We propose a diversified data fusion (DDF) method that not only inherits the merits of traditional data fusion methods, i.e., it can improve the performance on relevance orientated metrics, but also considers a query as a compound rather than a single representation of an underlying information need, and regards documents appearing in the component lists as mixtures of latent topics.

3.4.1 Overview of DDF

DDF consists of three main parts: (I) perform standard data fusion; (II) infer latent topics; (III) perform diversification; see Algorithm 1. In the first part ("Part I" in Algorithm 1), DDF computes the fusion scores of the documents in the component lists based on an existing unsupervised data fusion method (steps 1 and 2 in Algorithm 1); in this paper we use CombSUM, as our experimental results in §5.1 and §5.2 show that CombSUM outperforms other plain fusion methods in most cases. In the second part ("Part II" in Algorithm 1), DDF integrates fusion scores into an LDA topic model such that latent topics of the documents, their corresponding estimated relevance scores, and the multinomial distribution of the topics specific to each document can be inferred (steps 3–15 in Algorithm 1). In the last part ("Part III" in Algorithm 1), DDF uses the outputs of Parts I and II as input for an existing diversification method; in this paper, we use PM-2 [13] because it is a the state-of-the-art search result diversification model. Some concepts in PM-2, such as "quotient" and "seat," play important roles in the definition of the diversification step; they will be discussed in §3.4.3.

Below we describe how to infer latent topics ("Part II" in Algorithm 1) in §3.4.2 and how we utilize the information generated from latent topics and fusion scores ("Part III") in §3.4.3.

3.4.2 Part II: Inferring latent topics

Previous work on search result diversification shows that explicitly computing the probabilities of aspects of a query can improve diversification performance [1, 20, 39]. We do not assume that aspect information is explicitly available; we infer latent topics and their probabilities of being relevant using topic modeling.

Topic discovery in DDF is influenced not only by token co-occurrences, but also by the fusion scores of documents in the component lists. To avoid normalization and because fusion scores of the documents theoretically belong to $(0, +\infty)$, we employ a log-normal distribution for fusion scores to infer latent topics of the query via the documents and their relevance probabilities.

The latent topic model used in DDF is a generative model of relevance and the tokens in the documents that appear in the component individual lists. The generative process used in Gibbs sampling [34] for parameter estimation, is as follows:

i. Draw T multinomials ϕ_z from a Dirichlet prior β, one for each topic z;

ii. For each document $d \in \mathcal{C}_\mathbf{L}$, draw a multinomial θ_d from a Dirichlet prior α; then for each token w_{di} in document d:
   (a) Draw a topic z_{di} from multinomial θ_d;
   (b) Draw a token w_{di} from multinomial $\phi_{z_{di}}$;
   (c) Draw a fusion score f_{di} for w_{di} from Log-normal $\mathcal{N}(\mu_{z_{di}}, \sigma_{z_{di}})$.

Fig. 1 shows a graphical representation of our model. In the generative process, the fusion scores of tokens observed in the same document are the same and computed by a data fusion method, like

Algorithm 1: Diversified data fusion

Input : A query q
Ranked lists to be fused, $L_1, L_2, \ldots, L_m$
The combined set of documents $\mathcal{C}_\mathbf{L} := \bigcup_{i=1}^{m} L_i$
A standard fusion method X
A tradeoff parameter λ
Number of latent topics T
Hyperparameters α, β
Output: A final fused diversified list of documents L_f.

`/* Part I: Perform standard data fusion */`
1 **for** $d = 1, 2, \ldots, |\mathcal{C}_\mathbf{L}|$ **do**
2 Initialize $F_\mathrm{X}(d|\mathbf{L}, q)$ using a standard fusion method X
`/* Part II: Infer latent topics */`
3 Randomly initialize topic assignment for all tokens in $\mathbf{w}$
4 **for** $z = 1, 2, \ldots, T$ **do**
5 Initialize μ_z and σ_z randomly for topic z
6 **for** $iter = 1, 2, \ldots, N_{iter}$ **do**
7 **for** $d = 1, 2, \ldots, |\mathcal{C}_\mathbf{L}|$ **do**
8 **for** $i = 1, 2, \ldots, N_d$ **do**
9 draw z_{di} from $P(z_{di}|\mathbf{w}, \mathbf{r}, \mathbf{z}_{-di}, \alpha, \beta, \mu, \sigma, \mathbf{L}, q)$
10 update $n_{z_{di}w_{di}}$ and $m_{dz_{di}}$
11 **for** $z = 1, 2, \ldots, T$ **do**
12 update μ_z and σ_z
13 Compute the posterior estimate of θ
14 **for** $z = 1, 2, \ldots, T$ **do**
15 $v_{z|q} \leftarrow \frac{\exp\{u_z + \frac{1}{2}\sigma_z^2\}}{\sum_{z'=1}^{T} \exp\{u_{z'} + \frac{1}{2}\sigma_{z'}^2\}}$
`/* Part III: Perform diversification */`
16 $L_f \leftarrow \varnothing$
17 $R \leftarrow \mathcal{C}_\mathbf{L}$
18 **for** $z = 1, 2, \ldots, T$ **do**
19 $s_{z|q} \leftarrow 0$
20 **for** *all positions in the ranked list* L_f **do**
21 **for** $z = 1, 2, \ldots, T$ **do**
22 $qt[z|q] = \frac{v_{z|q}}{2s_{z|q}+1}$
23 $z^* \leftarrow \arg\max_z qt[z|q]$
24 $d^* \leftarrow \arg\max_{d \in R}$ $\lambda \times qt[z^*|q] \times P(d|z^*, q)+$
25 $(1-\lambda) \sum_{z \neq z^*} qt[z|q] \times P(d|z, q)$
26 $L_f \leftarrow L_f \cup \{d^*\}$ `/* append` d^* `to` L_f `*/`
27 $R \leftarrow R \backslash \{d^*\}$
28 **for** $z = 1, 2, \ldots, T$ **do**
29 $s_{z|q} \leftarrow s_{z|q} + \frac{P(d^*|z,q)}{\sum_{z'} P(d^*|z',q)}$

CombSUM, for the document, although a fusion score is generated for each token from the log-normal distribution. We use a fixed number of latent topics, T, although a non-parametric Bayes version of DDF that automatically integrates over the number of topics would certainly be possible. The posterior distribution of topics depends on the information from two modalities—both tokens and the fusion scores of the documents.

Inference is intractable in this model. Following [18, 24, 34, 36, 46, 47, 50], we employ Gibbs sampling to perform approximate inference. We adopt a conjugate prior (Dirichlet) for the multinomial distributions, and thus we can easily integrate out θ and ϕ, analytically capturing the uncertainty associated with them. In this way we facilitate the sampling, i.e., we need not sample θ and ϕ at all. Because we use the continuous log-normal distribution rather than discretizing fusion scores, sparsity is not a big concern in fitting the model. For simplicity and speed we estimate these log-normal distributions μ and σ by the method of moments, once per iteration of Gibbs sampling (see the Appendix). We find that the sensitivity of the hyper-parameters α and β is limited. Thus, for simplicity,

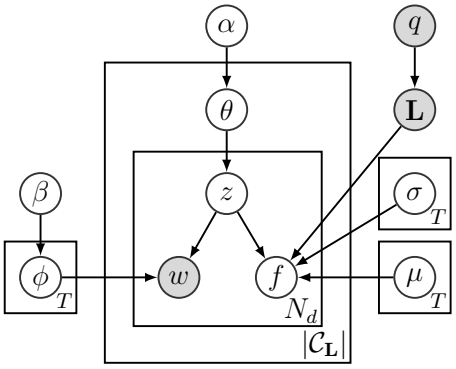

Figure 1: DDF graphical model for Gibbs sampling.

we use fixed symmetric Dirichlet distributions ($\alpha = 50/T$ and $\beta = 0.1$) in all our experiments.

In the Gibbs sampling procedure above, we need to calculate the conditional distribution $P(z_{di}|\mathbf{w}, \mathbf{r}, \mathbf{z}_{-di}, \alpha, \beta, \mu, \sigma, \mathbf{L}, q)$ (step 9 in Algorithm 1), where $\mathbf{z}_{-di}$ represents the topic assignments for all tokens except w_{di}. We begin with the joint probability of documents to be fused, and using the chain rule, we can obtain the conditional probability conveniently as

$$P(z_{di}|\mathbf{w}, \mathbf{r}, \mathbf{z}_{-di}, \alpha, \beta, \mu, \sigma, \mathbf{L}, q) \propto (m_{dz_{di}} + \alpha_{z_{di}} - 1) \times \frac{n_{z_{di}w_{di}} + \beta_{w_{di}} - 1}{\sum_{v=1}^{V}(n_{z_{di}v} + \beta_v) - 1} \times \frac{1}{F_\mathrm{X}(d|\mathbf{L}, q)\sigma_{z_{di}}\sqrt{2\pi}} \exp\{-\frac{(\ln F_\mathrm{X}(d|\mathbf{L}, q) - \mu_{z_{di}})^2}{2\sigma_{z_{di}}^2}\},$$

where n_{zv} is the total number of tokens v that are assigned to topic z, m_{dz} represents the number of tokens in document d that are assigned to topic z. An overview of the Gibbs sampling procedure we use is shown from step 3 to step 12 in Algorithm 1; details are provided in the Appendix.

One merit of our generative model for DDF is that we can predict a fusion score for any document once the tokens in the document have been observed. Given a document, we predict its fusion score by choosing the discretized fusion score that maximizes the posterior which is calculated by multiplying the fusion score probability of all tokens from their corresponding topic-wise log-normal distributions, i.e., $\arg\max_f \prod_{i=1}^{N_d} p(f|\mu_{z_i}, \sigma_{z_i})$.

More importantly, after the Gibbs sampling procedure, we can easily infer the multinomial distribution of topics specific to each document $d \in \mathcal{C}_\mathbf{L}$ as (step 13 in Algorithm 1):

$$\theta_{d,z} = \frac{n_{d,z} + \alpha_z}{\sum_{z=1}^{T}(n_{d,z} + \alpha_z)}, \tag{2}$$

where $n_{d,z}$ is the number of tokens assigned to latent topic z in document d; we can also conveniently estimate the probability of a topic being relevant to the query, denoted as $v_{z|q}$, by (step 15 in Algorithm 1):

$$v_{z|q} := \frac{\mathbb{E}[\mathbf{f}|z]}{\sum_{z'=1}^{T} \mathbb{E}[\mathbf{f}|z']} = \frac{\exp\{u_z + \frac{1}{2}\sigma_z^2\}}{\sum_{z'=1}^{T} \exp\{u_{z'} + \frac{1}{2}\sigma_{z'}^2\}}, \tag{3}$$

where $\mathbb{E}$ denotes the expectation.

3.4.3 *Part III: Diversification*

In Part III of our DDF model we propose a modification of PM-2. Before we discuss the details of this modification, we briefly describe PM-2. PM-2 is a probabilistic adaptation of the Sainte-Laguë method for assigning seats (positions in the ranked list) to

members of competing political parties (aspects) such that the number of seats for each party is proportional to the votes (aspect popularity, also called aspect probabilities, i.e., $p(z|q)$) they receive. PM-2 starts with a ranked list L_f with k empty seats. For each of these seats, it computes the quotient $qt[z|q]$ for each topic z given q following the Sainte-Laguë formula:

$$qt[z|q] = \frac{v_{z|q}}{2s_{z|q}+1}, \tag{4}$$

where $v_{z|q}$ is the probability of topic z given q, i.e., the weight of topic z. According to the Sainte-Laguë method, this seat should be awarded to the topic with the largest quotient in order to best maintain the proportionality of the list. Therefore, PM-2 assigns the current seat to the topic z^* with the largest quotient. The document to fill this seat is the one that is not only relevant to z^* but to other topics as well:

$$d^* = \arg\max_{d\in R} \big(\lambda \times qt[z^*|q] \times P(d|z^*,q) + (1-\lambda)\textstyle\sum_{z\neq z^*} qt[z|q] \times P(d|z,q)\big), \tag{5}$$

where $P(d|z,q)$ is the probability of d talking about topic z for a given q. After the document d^* is selected, PM-2 increases the "portion" of seats occupied by each of the topics z by its normalized relevance to d^*:

$$s_{z|q} \leftarrow s_{z|q} + \frac{P(d^*|z,q)}{\sum_{z'} P(d^*|z',q)}.$$

This process repeats until we get k documents for L_f or we are out of candidate documents. The order in which a document is appended to L_f determines its ranking.

We face two challenges in PM-2: it is non-trivial to get the aspect probability $v_{z|q}$ (i.e., $p(z|q)$), which is often set to be uniform, and it is non-trivial to compute $p(d|z,q)$, which usually requires explicit access to additional information. To address the first challenge, we compute $v_{z|q}$ by (3), such that (4) can be modified as:

$$qt[z|q] = \frac{p(z|q)}{2s_{z|q}+1} = \frac{\exp\{u_z + \frac{1}{2}\sigma_z^2\}}{(2s_{z|q}+1)\sum_{z'=1}^{T}\exp\{u_{z'} + \frac{1}{2}\sigma_{z'}^2\}}.$$

For the second challenge, instead of computing $P(d|z,q)$ explicitly, we modify $P(d|z,q)$ and apply Bayes' Theorem so that

$$P(d|z,q) = \frac{p(z|d,q)p(d|q)}{p(z|q)} = \frac{p(z|d,q)p(d|q)}{v_{z|q}}. \tag{6}$$

Then we integrate the fused score generated by CombSUM into our model, i.e., we set

$$p(d|q) \overset{rank}{=} F_{\text{CombSUM}}(d;q)$$

in (6). As a result, after applying (6) to (5), DDF selects a candidate document by:

$$d^* = \arg\max_{d\in R} \lambda \cdot qt[z^*|q] \cdot \frac{p(z^*|d,q)\cdot F_{\text{CombSUM}}(d;q)}{v_{z^*|q}} + (1-\lambda)\textstyle\sum_{z\neq z^*} qt[z|q] \cdot \frac{p(z|d,q)\cdot F_{\text{CombSUM}}(d;q)}{v_{z|q}}, \tag{7}$$

where $p(z|d;q)$ is the probability of document d belonging to topic z, which can easily be inferred in our DDF model by (2) (i.e., $p(z|d,q) = \theta_{d,z}$). Therefore, after applying (2) and (3), (7) can be rewritten as:

$$d^* = \arg\max_{d\in R} \lambda \cdot qt[z^*|q] \cdot \frac{\theta_{d,z^*}\cdot F_{\text{CombSUM}}(d;q)}{\exp\{\mu_z^* + \frac{1}{2}\sigma_z^{*2}\}} + (1-\lambda)\textstyle\sum_{z\neq z^*} qt[z|q] \cdot \frac{\theta_{d,z}\cdot F_{\text{CombSUM}}(d;q)}{\exp\{\mu_z + \frac{1}{2}\sigma_z^2\}}, \tag{8}$$

where it should be noted that we ignore the constant term

$$\textstyle\sum_{z=1}^{T}\exp\{\mu_z + \frac{1}{2}\sigma_z^2\},$$

as it has no impact on selecting the candidate document d^*.

4. EXPERIMENTAL SETUP

In this section, we describe our experimental setup; §4.1 lists our research questions; §4.2 describes our data set; §4.3 lists the metrics and the baselines; §4.4 details the settings of the experiments.

4.1 Research questions

The research questions guiding the remainder of the paper are:

RQ1 Do fusion methods help improve state-of-the-art search diversification methods? Do they help in terms of intent-aware precision, as our main metric? Does DDF beat standard and state-of-the-art fusion methods? (See §5.1 and §5.2.)

RQ2 What is the effect on the diversification performance of DDF and fusion methods of the number of component lists? Does the contribution of fusion to diversification performance depend on the quality of the component lists? (See §5.3)

RQ3 Does DDF outperform the best diversification and fusion methods on each query? (See §5.4.)

RQ4 How do the rankings of DDF differ from those produced by other fusion methods? (See §5.5.)

RQ5 What is the effect on the diversification performance of DDF of the number of latent topics used by DDF? (See §5.6.)

4.2 Data set

In order to answer our research questions we work with the runs submitted to the TREC 2009, 2010, 2011 and 2012 Web tracks, and the billion-page ClueWeb09 collection.[1] There are two tasks in these tracks: an ad hoc search task and a search result diversification task [8, 10–12]. We only focus on the diversification task, where the top-k documents returned should not only be relevant but also cover as many aspects as possible in response to a given query. In total, we have 200 ambiguous queries from the four years, with 2 queries (#95 and #100 in the 2010 edition) not having relevant documents. Typically, each query has 2 to 5 aspects, and some relevant documents are relevant to more than 2 aspects of the query.

Many of the runs submitted to these four years of the Web track for the diversification task were generated by state-of-the-art diversification methods. In total, we have 119, 88, 62 and 48 runs from the 2009, 2010, 2011 and 2012 editions, respectively.[2]

4.3 Evaluation metrics and baselines

We evaluate our component runs and fused runs using several standard metrics that are official evaluation metrics in the diversification tasks at TREC Web tracks [8, 10–12] and are widely used in the literature on search result diversification [2, 3, 13, 14, 38, 40]: Prec-IA@k [2], MAP-IA@k [2], ERR-IA@k [2] and α-nDCG@k [9]. The former two are set-based and indicate, respectively, the precision and mean average precision across all aspects of the query in the search results, whereas the remaining ones are cascade measures that penalize redundancy at each position in the ranked list based on how much of that information the user has already seen from documents at earlier ranks.

[1] Available from http://boston.lti.cs.cmu.edu/Data/clueweb09.

[2] All runs are available from http://trec.nist.gov.

We follow published work on search result diversification and mainly compute the metric scores at depth 20. Statistical significance of observed differences between the performance of two runs is tested using a two-tailed paired t-test and is denoted using $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for significant differences for $\alpha = .01$, or $^{\triangle}$ (and $^{\triangledown}$) for $\alpha = .05$.

When assessing a fusion method X we will prefer fusion methods that are safe, where we say that X is *safe for metric* M if applying X to a set of component runs always yields a fused run that scores at least as high as the highest scoring component run in the set (according to M).

We consider several baselines. Two standard fusion methods [26], CombSUM and CombMNZ; two state-of-the-art fusion methods [23], ClustFuseCombSUM and ClustFuseCombMNZ; each year's best performing runs in the diversification tasks at the TREC Web track [8, 10–12], and state-of-the-art plain diversification methods, xQuAD [39] and PM-2 [13]. As DDF builds on both fusion and diversification methods, we also consider two fusion methods, CombSUMMMR and CombSUMPM-2, that integrate plain diversification methods MMR [6] and PM-2 into CombSUM for diversification, respectively.

4.4 Experiments

We report on five main experiments aimed at answering the research questions listed in §4.1. In our first experiment, aimed at determining whether fusion methods help diversification, we fuse the five top performing diversification result lists from the TREC Web 2009, 2010, 2011 and 2012 submitted runs (some lists are generated by the implementation of PM-2) by our baselines, viz., CombSUM, CombMNZ, ClustFuseCombSUM, ClustFuseCombMNZ, CombSUMMMR and CombSUMPM-2 (see §4.3). The performance of the baselines is compared against that of DDF.

Our second experiment is aimed at understanding the effect on the diversification performance of DDF and fusion methods of the number of component lists; we randomly sample $k \in \{2, 4, \ldots, 26\}$ component runs from the submitted runs in the TREC Web 2012 track and fuse them. We repeat the experiments 20 times and report the average results and the standard deviations. We also show one sample's result when fusing 4 runs.

Next, in order to understand how DDF outperforms the best component run and the fusion methods per query, our third experiment provides a query-level analysis. Our fourth experiment is aimed at understanding how the runs generated by DDF differ from those produced by other fusion methods; we zoom in on the differences between DDF and the next best performing fusion method, CombSUMPM-2, in terms of the documents (and aspects) retrieved by one, but not the other, or by both.

Finally, to understand the influence of the number of latent topics used in DDF, we vary the number of latent topics and assess the performance of DDF. We also use an oracle variant of DDF, called DDF2, where for every test query we consider as many latent topics as there are aspects according to the ground truth. The number of topics used in DDF is set to 10, unless stated otherwise.

5. RESULTS

In §5.1 we examine the performance of baseline fusion methods on the diversification task, which we follow with a section on the performance of DDF in §5.2. §5.3 details the effect of the number of lists; §5.4 provides a query-level analysis; §5.5 zooms in on the effect on ranking of DDF compared to the next best fusion method; §5.6 examines the effect of the number of latent topics on DDF.

5.1 Performance of baseline fusion methods

In Table 2 we list the diversity scores of the baseline fusion methods on the diversity task: CombSUM, CombMNZ, ClustFuseCombSUM, ClustFuseCombMNZ, CombSUMMMR, CombSUMPM-2, with the 5 best performing component lists from the TREC Web 2009, 2010, 2011 and 2012 tracks, respectively.[3] For all metrics and in all years, almost all baseline fusion methods outperform the state-of-the-art diversification methods, and in many cases significantly so. Note, however, that none of the baseline methods is safe in the sense defined in §4.3. Additionally, Table 3 shows the diversity scores of the baseline fusion methods when we fuse 4 randomly sampled runs from the 2012 data set, which confirms that fusion does help diversification.

5.2 The performance of DDF

Inspired by the success of baseline fusion methods on the diversification task, we now consider our newly proposed fusion method, DDF. Returning to Tables 2 and 3, two types of conclusion emerge. First, DDF outperforms all component runs (note that component runs in Table 2 are the best runs in the tracks), on all metrics, for all years. In other words, it is *safe* in the sense defined in Section 4.3. The difference between DDF and the best performing component run is always significant. We believe that the strong performance of DDF is due to the fact that DDF not only focuses on improving the relevance score of fused run but also explicitly tries to diversify the fused run.

Second, DDF outperforms all baseline fusion methods, on all metrics. In many cases, CombSUMPM-2 and CombSUM yield the second and third best performance, respectively, but DDF outperforms them in every case, and often significantly so. DDF can beat CombSUMPM-2 as it tackles two main challenges in PM-2 (see §3.4.3), although they build on the same framework. CombSUMMMR follows a similar strategy as DDF but its performance is worse than that of DDF. This is due to the fact that MMR models documents as if they are centered around a single topic only. It is clear from Tables 2 and 3 that cluster-based data fusion methods (ClustFuseCombSUM, ClustFuseCombMNZ) sometimes perform a little worse than the standard fusion method they build on (CombSUM, CombMNZ). This is because cluster-based fusion focuses on relevance of the documents rather than on diversification.

5.3 Effect of the number of component lists

Next, we zoom in on DDF. In particular, we explore the effect of varying the number of lists to be fused on its performance. Fig. 2 shows the fusion results of randomly sampling $k \in \{2, 4, \ldots, 26\}$ lists from the 48 runs submitted to the TREC Web 2012 track plus the PM-2 runs (due to space limitations, we only report results using the 2012 runs; the findings on other years are qualitatively similar). For each k, we repeat the experiment 20 times and report on the average scores and the corresponding standard deviations indicated by the error bars in the figure. We use CombSUM as a representative example for comparison with DDF, as the results of other baseline fusion methods are worse or have qualitatively similar results to those of CombSUM. As shown in Fig. 2, DDF always outperforms CombSUM in terms of the Prec-IA, α-nDCG and ERR-IA evaluation metrics and the performance gaps remain almost unchanged, in absolute terms, no matter how many component lists are fused. One reason for this is that as DDF builds on CombSUM, it inherits the merits of the fusion method, and more importantly, at the same time it tries to infer latent topics and rerank the high

[3]The run "PM-2 (TREC)" is the run that utilizes aspect information from the ground truth in the PM-2 model and the run "PM-2 (engine)" is produced using information from a commercial search engine. The run "xQuAD (uogTrX)" is a uogTrX TREC edition run generated using the xQuAD algorithm; see [33].

Table 2: Performance obtained using the 2009–2012 editions of the TREC Web tracks. The best performing run per metric per year is in boldface. Statistically significant differences between fusion method and the best component run, between DDF and CombSUM, and between DDF and CombSUMPM-2, are marked in the upper right hand corner of the fusion method score, in the upper left hand corner of DDF's score, and in the lower left hand corner of DDF's score, respectively.

		Prec-IA	MAP-IA	α-nDCG	ERR-IA
2012	DFalah120A	.3241	.0990	.5291	.4259
	DFalah120D	.3241	.0990	.5291	.4259
	xQuAD (uogTrA44xi)	.3349	.1345	.5917	.4873
	xQuAD (uogTrA44xu)	.3504	.1360	.6061	.5048
	xQuAD (uogTrB44xu)	.3389	.1339	.5795	.4785
	ClustFuseCombMNZ	.3533	.1488$^{\blacktriangle}$	.6010	.5105
	ClustFuseCombSUM	.3545	.1495$^{\blacktriangle}$	.5965	.5049
	CombSUMMMR	.3558	.1544$^{\blacktriangle}$	.6106	.5115
	CombSUMPM-2	.3718$^{\blacktriangle}$	.1826$^{\blacktriangle}$	.6228$^{\blacktriangle}$	.5179$^{\triangle}$
	CombMNZ	.3663$^{\blacktriangle}$	.1785$^{\blacktriangle}$	.6154$^{\triangle}$	.5153$^{\triangle}$
	CombSUM	.3592$^{\triangle}$	.1767$^{\blacktriangle}$	.6114$^{\triangle}$	.5126$^{\triangle}$
	DDF	**.3904**$^{\blacktriangle\blacktriangle}$	**.1910**$^{\blacktriangle\blacktriangle}$	**.6334**$^{\blacktriangle\blacktriangle}$	**.5266**$^{\blacktriangle\blacktriangle}$
2011	ICTNET11ADR2	.2993	.1328	.5725	.4658
	umassGQdist	.3003	.1313	.5513	.4530
	xQuAD (uogTrA45Nmx2)	.3039	.1365	.6298	.5284
	xQuAD (uogTrA45Vmx)	.3030	.1323	.6304	.5238
	UWatMDSdm	.3214	.1350	.5979	.4875
	ClustFuseCombMNZ	.3303$^{\blacktriangle}$	.1757$^{\blacktriangle}$	.6221$^{\triangledown}$	.5001
	ClustFuseCombSUM	.3296$^{\triangle}$	.1775$^{\blacktriangle}$	.6307	.5110
	CombSUMMMR	.3395$^{\blacktriangle}$	.1830$^{\blacktriangle}$	.6341	.5107
	CombSUMPM-2	.3450$^{\blacktriangle}$	.2024$^{\blacktriangle}$	.6448$^{\blacktriangle}$	.5196
	CombMNZ	.3413$^{\blacktriangle}$	.1943$^{\blacktriangle}$	.6430$^{\blacktriangle}$	.5209
	CombSUM	.3376$^{\blacktriangle}$	.1966$^{\blacktriangle}$	.6423$^{\blacktriangle}$	.5216
	DDF	**.3596**$^{\blacktriangle\blacktriangle}$	**.2102**$^{\blacktriangle\blacktriangle}$	**.6496**$^{\blacktriangle}$	**.5295**$^{\blacktriangle}$
2010	CSE.pm2.run	.1832	.0351	.4165	.3052
	cmuWi10D	.1879	.0599	.3452	.2484
	xQuAD (uogTrA42x)	.1845	.0529	.3558	.2454
	PM-2 (engine)	.2009	.0414	.3660	.2581
	PM-2 (TREC)	.2026	.0430	.4449	.3320
	ClustFuseCombMNZ	.2105	.0845$^{\blacktriangle}$	.4313	.3221
	ClustFuseCombSUM	.2072	.0825$^{\blacktriangle}$	.4257$^{\triangledown}$	.3148$^{\triangledown}$
	CombSUMMMR	.2115$^{\triangle}$	.0836$^{\blacktriangle}$	.4366	.3189
	CombSUMPM-2	.2129$^{\blacktriangle}$	.0839$^{\blacktriangle}$	.4379	.3193
	CombMNZ	.2177$^{\blacktriangle}$	.0899$^{\blacktriangle}$	.4471	.3411$^{\triangle}$
	CombSUM	.2159	.0875$^{\blacktriangle}$	.4454	.3350
	DDF	**.2285**$^{\blacktriangle\blacktriangle}$	**.0910**$^{\triangle\blacktriangle}$	**.4627**$^{\blacktriangle\blacktriangle}$	**.3406**$^{\blacktriangle\blacktriangle}$
2009	NeuDiv1	.1343	.0458	.2781	.1705
	NeuDivW75	.1239	.0397	.2501	.1598
	xQuAD(uogTrDPCQcdB)	.1302	.0463	.2968	.1848
	xQuAD (uogTrDYCcsB)	.1268	.0444	.3081	.1922
	uwgym	.1224	.0456	.2798	.1701
	ClustFuseCombMNZ	.1381	.0681$^{\blacktriangle}$	.3076	.1937
	ClustFuseCombSUM	.1379	.0680$^{\blacktriangle}$	.3223$^{\blacktriangle}$	.2005
	CombSUMMMR	.1424$^{\triangle}$	.0682$^{\blacktriangle}$	.3343$^{\blacktriangle}$	.2028$^{\triangle}$
	CombSUMPM-2	.1588$^{\blacktriangle}$	.0754$^{\blacktriangle}$	.3887$^{\blacktriangle}$	.2674$^{\blacktriangle}$
	CombMNZ	.1400$^{\triangle}$	.0666$^{\blacktriangle}$	.3343$^{\blacktriangle}$	.2033$^{\triangle}$
	CombSUM	.1400$^{\triangle}$	.0664$^{\blacktriangle}$	.3482$^{\blacktriangle}$	.2080$^{\triangle}$
	DDF	**.1631**$^{\blacktriangle\blacktriangle}$	**.0731**$^{\blacktriangle\blacktriangle}$	**.4005**$^{\blacktriangle\blacktriangle}$	**.2713**$^{\blacktriangle\blacktriangle}$

ranked documents in terms of novelty of the documents. For the MAP-IA metric, however, the gaps increase with more component lists being fused. The performance of both DDF and CombSUM increases faster when the number of component lists increases but is ≤ 10 than when the number of component lists is > 10, for all the metrics. This seems to be inherent to the underlying CombSUM method and is due to the fact that with smaller numbers of component lists, there is simply more space available at depth 20 to obtain improvements than with larger numbers of component lists.

Table 3: Performance obtained using the 2012 editions of the TREC Web track. The best performing run per metric is in boldface. Other notational conventions as in Table 2.

		Prec-IA	MAP-IA	α-nDCG	ERR-IA
2012	QUTparaBline	.2261	.0639	.5270	.4185
	xQuAD (uogTrA44xl)	.2957	.1077	.5161	.4009
	utw2012c1	.1637	.0439	.5075	.4046
	PM-2 (TREC)	.2631	.0601	.5245	.4155
	ClustFuseCombMNZ	.2735$^{\triangledown}$	.1155$^{\blacktriangle}$	.5717$^{\blacktriangle}$	.4608$^{\blacktriangle}$
	ClustFuseCombSUM	.2752	.1172$^{\blacktriangle}$	.5726$^{\blacktriangle}$	.4674$^{\blacktriangle}$
	CombSUMMMR	.2783$^{\triangledown}$	.1189$^{\blacktriangle}$	.5799$^{\blacktriangle}$	.4633$^{\blacktriangle}$
	CombSUMPM-2	.2934	.1305$^{\blacktriangle}$	.6013$^{\blacktriangle}$	.4877$^{\blacktriangle}$
	CombMNZ	.2864	.1267$^{\blacktriangle}$	.5851$^{\blacktriangle}$	.4708$^{\blacktriangle}$
	CombSUM	.2884	.1275$^{\blacktriangle}$	.5944$^{\blacktriangle}$	.4803$^{\blacktriangle}$
	DDF	**.3193**$^{\blacktriangle\blacktriangle}$	**.1409**$^{\blacktriangle\blacktriangle}$	**.6107**$^{\triangle\blacktriangle}$	**.4919**$^{\blacktriangle\blacktriangle}$

5.4 Query-level analysis

We take a closer look at per test query improvements of DDF over the best baseline fusion run when fusing the best 5 runs in 2012, viz., CombSUMPM-2, which outperforms the best component list. Fig. 3 shows the per query performance differences in terms of Prec-IA, MAP-IA, α-nDCG and ERR-IA, respectively, of DDF against CombSUMPM-2. DDF achieves performance improvements for many queries when compared against CombSUMPM-2, although the differences are sometimes relatively small.

In a very small number of cases, DDF performs poorer than CombSUMPM-2. This appears to be due to the fact that DDF "over-diversifies" documents in runs produced by CombSUM that have very few relevant document to start with, so that DDF ends up promoting different but non-relevant documents.

5.5 Zooming in on Prec-IA@k

Next, we zoom in on one of the metrics that shows the biggest relative differences between DDF and the next best performing fusion method, Prec-IA, so as to understand how the runs generated by DDF differ from those by other fusion-based methods. Here, again, we use CombSUMPM-2 as a representative, as it tends to outperform or equal the other fusion methods. Specifically, we report changes in the number of relevant documents for DDF against CombSUMPM-2 when fusing the 2012 runs in Table 2 in 2012; see Fig. 4. Red bars indicate the number of relevant documents that appear in the run of DDF but not the run of CombSUMPM-2, white bars indicate the number of relevant documents in both runs, whereas blue bars indicate the number of relevant documents that appear not in DDF but in CombSUMPM-2; topics are ordered first by the size of the red bar, then the size of the white bar, and finally the size of the blue bar.

Clearly, the differences between DDF and CombSUMPM-2 in the top 5 and 10 are more limited than the differences in the top-15 and 20, but in all cases DDF outperforms CombSUMPM-2. E.g., in total there are 45 more relevant documents in the top 20 of the run produced by DDF than those in the CombSUMPM-2 run (49 relevant documents in DDF but not in CombSUMPM-2, 4 relevant documents in CombSUMPM-2 but not in DDF). We examine the matter further by comparing the Prec-AI@5, 10, 15, 20 scores of the DDF and CombSUMPM-2 runs for the 2012 data; see Table 4. The differences at small depths (5, 10) are weakly statistically significant while those at bigger depths are significant, confirming our observations in Fig. 4; we also find that DDF statistically significantly outperforms CombSUMPM-2 in terms of Prec-IA scores at depth 5, 10, 15 and 20, which again confirms the above observations based on Fig. 4.

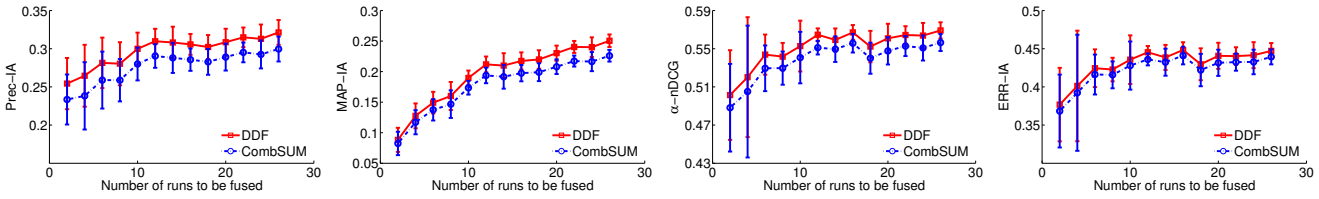

Figure 2: Effect on performance (in terms of Prec-IA, MAP-IA, α-nDCG and ERR-IA) of the number of component lists, using runs sampled from the TREC 2012 Web track. We plot averages and standard deviations. Note: the figures are not to the same scale.

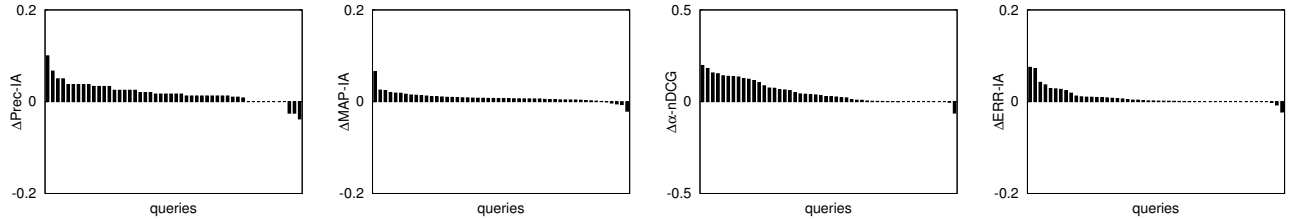

Figure 3: Per query performance differences of DDF against CombSUMPM-2 (second row). The figures shown are for fusing the runs in TREC Web 2012 track, for Prec-IA@20, MAP-IA@20, α-nDCG@20 and ERR-IA@20 (from left to right). A bar extending above the center of a plot indicates that DDF outperforms CombSUMPM-2, and vice versa for bars below the center.

Table 4: Prec-IA@5, 10, 15, 20 performance comparison between CombSUMPM-2 and DDF. A statistically significant difference between DDF and CombSUMPM-2 is marked in the upper left hand corner of the DDF score.

Prec-IA@	5	10	15	20
CombSUMPM-2	.4367	.4066	.3887	.3718
DDF	△.4555	△.4194	▲.4060	▲.3904

5.6 Effect of the number of topics

Finally, we examine the effect on the overall performance of the number of latent topics used in DDF, and contrast the performance of DDF with varying number of latent topics against DDF2, CombSUM and CombSUMPM-2. Here, DDF2 is the same algorithm as DDF except that for every test query it considers as many latent topics as there are aspects according to the ground truth. We use DDF2, DDF, CombSUM and CombSUMPM-2 to fuse the component result runs listed in Table 2 in 2012 as an example. We vary the number of latent topics in DDF from 2 to 16. See Fig. 5.

When the number of latent topics used in DDF increases from 2 to 6, the performance of DDF increases dramatically. When only 2 latent topics are used, the performance is worse than that of CombSUM and CombSUMPM-2; e.g., Prec-IA@20 for DDF is 0.3404, while the scores of CombSUM and CombSUMPM-2 are 0.3592 and 0.3718, respectively. In contrast, when the number of latent topics varies between 8 to 16, the performance of DDF seems to level off. This demonstrates another merit of our fusion model, DDF: it is robust and not sensitive to the number of latent topics once the number of latent topics is "large enough." Another important finding from Fig. 5 is that DDF2 always enhances the performance of DDF, CombSUM and CombSUMPM-2, for all metrics, which demonstrates the fact that latent topics can enhance the performance. The performance differences between DDF2 and DDF are quite marginal and not statistically significant. We leave it as future work to dynamically estimate the number of aspects (and latent topics) of an incoming query and to use this estimate in DDF.

6. CONCLUSION

Most previous work on search result diversification focuses on the content of the documents returned by an ad hoc algorithm to diversify the results implicitly or explicitly, i.e., using implicit or explicit representations of aspects. In this paper we have adopted a different perspective on the search result diversification problem, based on data fusion. We proposed to use traditional unsupervised and state-of-the-art data fusion methods, CombSUM, CombMNZ, ClustFuseCombSUM, ClustFuseCombMNZ, CombSUMMMR and CombSUMPM-2 to diversify result lists. This led to the insight that fusion does aid diversification. We also proposed a fusion-based diversification method, DDF, which infers latent topics from ranked lists of documents produced by a standard fusion method, and combines this with a state-of-the-art result diversification model. We found that data fusion approaches outperform state-of-the-art search result diversification algorithms, with DDF invariably giving rise to the highest scores on all of the metrics that we have considered in this paper. DDF was shown to behave well with different numbers of component lists. We also found that DDF is insensitive to the number of latent topics of a query, once a sufficiently large number was chosen, e.g., 10.

As to future work, we aim to incorporate into DDF methods for automatically estimating the number of aspects, which will be used to set the number of latent topics. The last and third part of DDF is based on a particular choice of method, viz. PM-2, and we only apply rank-based fusion methods for diversification. In future work we plan to compare these choices with alternative choices, and apply other fusion alternatives, e.g., score-based fusion methods.

Acknowledgements. We thank Van Dang for generating the PM-2 runs for us. This research was partially supported by the China Scholarship Council, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nrs 288024 and 312827, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

7. REFERENCES

[1] S. Abbar, S. Amer-Yahia, P. Indyk, and S. Mahabadi.

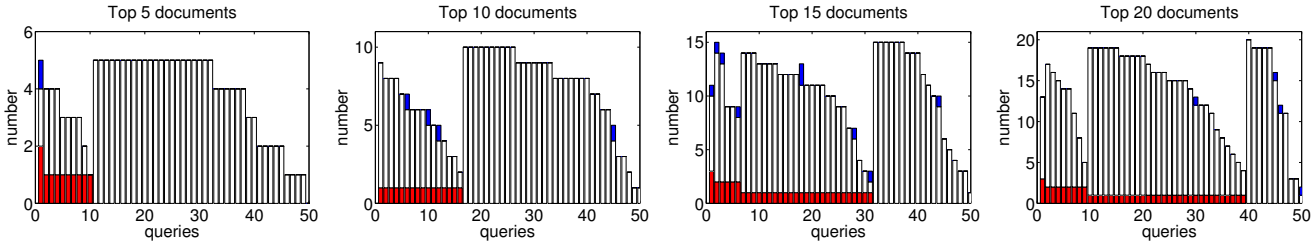

Figure 4: How runs produced by DDF and CombSUMPM-2 differ. Red, white, blue bars indicate the number of relevant documents that appear in DDF but not in CombSUMPM-2, in both runs and not in DDF but in CombSUMPM-2, respectively, at corresponding depth k (for $k = 5, 10, 15, 20$). Figures should be viewed in color.

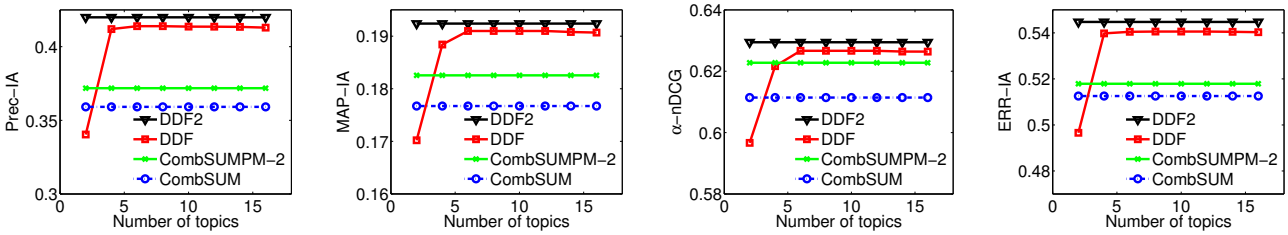

Figure 5: Performance comparison between DDF2, DDF, CombSUMPM-2 and CombSUM when varying the number of latent topics used in DDF. Note: the figures are not to be the same scale.

Real-time recommendation of diverse related articles. In *WWW*, pages 1–12, 2013.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.

[3] E. Aktolga and J. Allan. Sentiment diversification with different biases. In *SIGIR*, pages 593–600, 2013.

[4] J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR'01*, pages 276–284, 2001.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[6] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[7] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR*, pages 429–436, 2006.

[8] C. L. A. Clarke and N. Craswell. Overview of the TREC 2011 web track. In *TREC*, pages 1–9, 2011.

[9] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.

[10] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 web track. In *TREC*, pages 1–9, 2009.

[11] C. L. A. Clarke, N. Craswell, I. Soboroff, and G. V. Cormack. Overview of the TREC 2010 web track. In *TREC*, pages 1–9, 2010.

[12] C. L. A. Clarke, N. Craswell, and E. M. Voorhees. Overview of the TREC 2012 web track. In *TREC*, pages 1–8, 2012.

[13] V. Dang and W. B. Croft. Diversity by proportionality: An election-based approach to search result diversification. In *SIGIR*, pages 65–74, 2012.

[14] V. Dang and W. B. Croft. Term level search result diversification. In *SIGIR*, pages 603–612, 2013.

[15] M. Efron. Information search and retrieval in microblogs. *J. Am. Soc. for Inform. Sci. and Techn.*, 62(6):996–1008, 2011.

[16] M. Farah and D. Vanderpooten. An outranking approach for rank aggregation in information retrieval. In *SIGIR'07*, 2007.

[17] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *TREC-2*, 1994.

[18] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101:5228–5235, 2004.

[19] D. He and D. Wu. Toward a robust data fusion for document retrieval. In *IEEE NLP-KE'08*, 2008.

[20] J. He, V. Hollink, and A. de Vries. Combining implicit and explicit topic representations for result diversification. In *SIGIR*, pages 851–860, 2012.

[21] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[22] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *CIKM*, pages 775–784, 2011.

[23] A. K. Kozorovitsky and O. Kurland. Cluster-based fusion of retrieved lists. In *SIGIR'11*, pages 893–902, 2011.

[24] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *WSDM*, pages 375–384, 2013.

[25] J. D. Lafferty and D. M. Blei. Correlated topic models. In *NIPS'05*, pages 147–154, 2005.

[26] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In *SIGIR'95*, pages 180–188, 1995.

[27] J. H. Lee. Analyses of multiple evidence combination. In *SIGIR*, 1997.

[28] F. Li, M. Huang, and X. Zhu. Sentiment analysis with global topics and local dependency. In *AAAI*, 2010.

[29] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML*, pages 577–584. ACM, 2006.

[30] S. Liang and M. de Rijke. Finding knowledgeable groups in enterprise corpora. In *SIGIR'13*, pages 1005–1008, 2013.

[31] S. Liang, M. de Rijke, and M. Tsagkias. Late data fusion for microblog search. In *ECIR'13*, pages 743–746, 2013.

[32] S. Liang, Z. Ren, and M. de Rijke. The impact of semantic document expansion on cluster-based fusion for microblog

search. In *ECIR'14*, pages 493–499, 2014.
[33] N. Limsopatham, R. McCreadie, and M.-D. Albakour. University of Glasgow at TREC 2012: Experiments with Terrier in medical records, microblog, and web tracks. In *TREC*, 2012.
[34] J. S. Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *J. Am. Stat. Assoc.*, 89(427):958–966, 1994.
[35] C. Macdonald and I. Ounis. Voting for candidates: Adapting data fusion techniques for an expert search task. In *CIKM*, 2006.
[36] Z. Ren, S. Liang, E. Meij, and M. de Rijke. Personalized time-aware tweets summarization. In *SIGIR*, 2013.
[37] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494, 2004.
[38] T. Sakai, Z. Dou, and C. L. A. Clarke. The impact of intent selection on diversified search result. In *SIGIR*, 2013.
[39] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *WWW*, pages 881–890, 2010.
[40] R. L. Santos, C. Macdonald, and I. Ounis. Intent-aware search result diversification. In *SIGIR*, pages 595–604, 2011.
[41] R. L. T. Santos, J. Peng, C. Macdonald, and I. Ounis. Explicit search result diversification through sub-queries. In *ECIR*, 2010.
[42] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *TREC 1992*, pages 243–252. NIST, 1993.
[43] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell. LambdaMerge: merging the results of query reformulations. In *WSDM*, pages 795–804, 2011.
[44] I. Szpektor, Y. Maarek, and D. Pelleg. When relevance is not enough: promoting diversity and freshness in personalized question recommendation. In *WWW '13*, 2013.
[45] S. Vargas, P. Castells, and D. Vallet. Explicit relevance models in intent-oriented information retrieval diversification. In *SIGIR*, pages 75–84, 2012.
[46] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD'06*, pages 424–433, 2006.
[47] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.
[48] X. Wei, J. Sun, and X. Wang. Dynamic mixture models for multiple time-series. In *IJCAI*, pages 2909–2914, 2007.
[49] S. Wu. *Data fusion in information retrieval*, volume 13 of *Adaptation, Learning and Optimization*. Springer, 2012.
[50] Z. Xu, Y. Zhang, Y. Wu, and Q. Yang. Modeling user posting behavior on social media. In *SIGIR*, pages 545–554, 2012.
[51] C. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR*, pages 10–17, 2003.

APPENDIX

Gibbs sampling derivation for DDF model

We begin with the joint distribution $P(\mathbf{w}, \mathbf{f}, \mathbf{z}|\alpha, \beta, \mu, \sigma, \mathbf{L})$ and use conjugate priors to simplify the integrals. Notation defined in §3.

$$
\begin{aligned}
&P(\mathbf{w}, \mathbf{f}, \mathbf{z}|\alpha, \beta, \mu, \sigma, \mathbf{L}, q) = P(\mathbf{w}|\mathbf{z}, \beta)p(\mathbf{f}|\mu, \sigma, \mathbf{z}, \mathbf{L})P(\mathbf{z}|\alpha) \\
&= \int P(\mathbf{w}|\mathbf{\Phi}, \mathbf{z})p(\mathbf{\Phi}|\beta)d\mathbf{\Phi} \times p(\mathbf{f}|\mu, \sigma, \mathbf{z}, \mathbf{L}, q) \int P(\mathbf{z}|\mathbf{\Theta})P(\mathbf{\Theta}|\alpha)d\mathbf{\Theta} \\
&= \int \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\prod_{i=1}^{N_d} P(w_{di}|\phi_{z_{di}}) \prod_{z=1}^{T} p(\phi_z|\beta)d\mathbf{\Phi} \\
&\quad \times \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\prod_{i=1}^{N_d} p(f_{di}|\mu_{z_{di}}, \sigma_{z_{di}}, \mathbf{L}, q) \\
&\quad \times \int \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\left(\prod_{i=1}^{N_d} P(z_{di}|\theta_d)p(\theta_d|\alpha)\right) d\mathbf{\Theta} \\
&= \int \prod_{z=1}^{T}\prod_{v=1}^{V} \phi_{z_v}^{n_{z_v}} \prod_{z=1}^{T}\left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\prod_{v=1}^{V}\phi_{z_v}^{\beta_v-1}\right) d\mathbf{\Phi} \\
&\quad \times \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\prod_{i=1}^{N_d} p(f_{di}|\mu_{z_{di}}, \sigma_{z_{di}}, \mathbf{L}, q) \\
&\quad \times \int \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\prod_{z=1}^{T} \theta_{d_z}^{m_{d_z}} \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\left(\frac{\Gamma(\sum_{z=1}^{T}\alpha_z)}{\prod_{z=1}^{T}\Gamma(\alpha_z)}\prod_{z=1}^{T}\theta_{d_z}^{\alpha_z-1}\right) d\mathbf{\Theta} \\
&= \left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\right)^T \left(\frac{\Gamma(\sum_{z=1}^{T}\alpha_z)}{\prod_{z=1}^{T}\Gamma(\alpha_z)}\right)^{|\mathcal{C}_\mathbf{L}|} \\
&\quad \times \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\prod_{i=1}^{N_d} p(f_{di}|\mu_{z_{di}}, \sigma_{z_{di}}, \mathbf{L}, q) \\
&\quad \times \prod_{z=1}^{T}\frac{\prod_{v=1}^{V}\Gamma(n_{z_v}+\beta_v)}{\Gamma(\sum_{v=1}^{V}(n_{z_v}+\beta_v))} \prod_{d=1}^{|\mathcal{C}_\mathbf{L}|}\frac{\prod_{z=1}^{T}\Gamma(m_{d_z}+\alpha_z)}{\Gamma(\sum_{z=1}^{T}(m_{z_d}+\alpha_z))}
\end{aligned}
$$

Using the chain rule, we can obtain the conditional probability conveniently,

$$
\begin{aligned}
&P(z_{di}|\mathbf{w}, \mathbf{f}, \mathbf{z}_{-di}, \alpha, \beta, \mu, \sigma, \mathbf{L}, q) \\
&= \frac{P(z_{di}, w_{di}, f_{di}|\mathbf{w}_{-di}, \mathbf{f}_{-di}, \mathbf{z}_{-di}, \alpha, \beta, \mu, \sigma, \mathbf{L}, q)}{P(w_{di}, f_{di}|\mathbf{w}_{-di}, \mathbf{f}_{-di}, \mathbf{z}_{-di}, \alpha, \beta, \mu, \sigma, \mathbf{L}, q)} \\
&= \frac{P(\mathbf{w}, \mathbf{f}, \mathbf{z}|\alpha, \beta, \mu, \sigma, \mathbf{L}, q)}{P(\mathbf{w}, \mathbf{f}, \mathbf{z}_{-di}|\alpha, \beta, \mu, \sigma, \mathbf{L}, q)}
\end{aligned}
$$

because z_{di} depends only on w_{di} and f_{di}

$$
\begin{aligned}
&\propto \frac{P(\mathbf{w}, \mathbf{f}, \mathbf{z}|\alpha, \beta, \mu, \sigma, \mathbf{L}, q)}{P(\mathbf{w}_{-di}, \mathbf{f}_{-di}, \mathbf{z}_{-di}, |\alpha, \beta, \mu, \sigma, \mathbf{L}, q)} \\
&\propto (m_{dz_{di}} + \alpha_{z_{di}} - 1)\frac{n_{z_{di}w_{di}} + \beta_{w_{di}} - 1}{\sum_{v=1}^{V}(n_{z_{di}v} + \beta_v) - 1} \\
&\quad \times \frac{1}{f_{di}\sigma_{z_{di}}\sqrt{2\pi}} \exp\{-\frac{(\ln f_{di} - \mu_{z_{di}})^2}{2\sigma_{z_{di}}^2}\} \\
&\propto (m_{dz_{di}} + \alpha_{z_{di}} - 1)\frac{n_{z_{di}w_{di}} + \beta_{w_{di}} - 1}{\sum_{v=1}^{V}(n_{z_{di}v} + \beta_v) - 1} \\
&\quad \times \frac{1}{F_\mathrm{X}(d|\mathbf{L}, q)\sigma_{z_{di}}\sqrt{2\pi}} \exp\{-\frac{(\ln F_\mathrm{X}(d|\mathbf{L}, q) - \mu_{z_{di}})^2}{2\sigma_{z_{di}}^2}\},
\end{aligned}
$$

where $F_\mathrm{X}(d|\mathbf{L}, q) \in (0, +\infty)$ is a fusion score generated by a standard fusion method F_X for document $d \in \mathcal{C}_\mathbf{L}$ given the observation of lists $\mathbf{L}$ to be merged and query q. We use $F_{\mathrm{CombSUM}}(d|\mathbf{L}, q)$.

Since the data fusion score of a token that appears in d when fusing all the lists in $\mathbf{L}$ given a query q and the latent topics of which is z_{di}, is drawn from log-normal distributions, sparsity is not a big problem for parameter estimation of both $\mu_{z_{di}}$ and $\sigma_{z_{di}}$. For simplicity, we update both $\mu_{z_{di}}$ and $\sigma_{z_{di}}$ after each Gibbs sample iteration by maximum likelihood estimation:

$$
\begin{aligned}
\hat{\mu}_{z_{di}} &= \frac{\sum_{d'=1}^{|\mathcal{C}_\mathbf{L}|}\sum_{i' \wedge (z_{d'i'}=z_{di})}^{N_d} \ln f_{d'i'}}{n_{z_{di}}} \\
&= \frac{\sum_{d'=1}^{|\mathcal{C}_\mathbf{L}|}\sum_{i' \wedge (z_{d'i'}=z_{di})}^{N_d} \ln F_\mathrm{X}(d'|\mathbf{L}, q)}{n_{z_{di}}} \\
\hat{\sigma}_{z_{di}}^2 &= \frac{\sum_{d'=1}^{|\mathcal{C}_\mathbf{L}|}\sum_{i' \wedge (z_{d'i'}=z_{di})}^{N_d} (\ln f_{d'i'} - \hat{\mu})^2}{n_{z_{di}}} \\
&= \frac{\sum_{d'=1}^{|\mathcal{C}_\mathbf{L}|}\sum_{i' \wedge (z_{d'i'}=z_{di})}^{N_d} (\ln F_\mathrm{X}(d'|\mathbf{L}, q) - \hat{\mu})^2}{n_{z_{di}}}
\end{aligned}
$$