Search Result Diversification in Short Text Streams

SHANGSONG LIANG, University College London
EMINE YILMAZ, University College London
HONG SHEN, Sun Yat-sen University and University of Adelaide
MAARTEN DE RIJKE, University of Amsterdam
W. BRUCE CROFT, University of Massachusetts Amherst

We consider the problem of search result diversification for streams of short texts. Diversifying search results in short text streams is more challenging than in the case of long documents, as it is difficult to capture the latent topics of short documents. To capture the changes of topics and the probabilities of documents for a given query at a specific time in a short text stream, we propose a dynamic Dirichlet multinomial mixture topic model, called D2M3, as well as a Gibbs sampling algorithm for the inference. We also propose a streaming diversification algorithm, SDA, that integrates the information captured by D2M3 with our proposed modified version of the PM-2 (Proportionality-based diversification Method – second version) diversification algorithm. We conduct experiments on a Twitter dataset and find that SDA statistically significantly outperforms state-of-the-art non-streaming retrieval methods, plain streaming retrieval methods, as well as streaming diversification methods that use other dynamic topic models.

CCS Concepts: • **Information systems → Retrieval models and ranking;**

Additional Key Words and Phrases: Diversity, ad hoc retrieval, data streams

ACM Reference Format:
Shangsong Liang, Emine Yilmaz, Hong Shen, Maarten de Rijke, and W. Bruce Croft. 2017. Search result diversification in short text streams. ACM Trans. Inf. Syst. 36, 1, Article 8 (July 2017), 35 pages.
DOI: http://dx.doi.org/10.1145/3057282

1. INTRODUCTION

Search result diversification has been widely studied as a method to tackle query ambiguity [38]. Instead of trying to identify the "correct" interpretation behind an ambiguous query, a diverse ranker identifies the probable "aspects" (also called "subtopics")

This research was supported by the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts, the UCL Big Data Institute, Elsevier, Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Dutch national program COMMIT, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.001.116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Authors' addresses: S. Liang, Department of Computer Science, University College London, UK; email: shangsong.liang@ucl.ac.uk; E. Yilmaz, Department of Computer Science, University College London, UK; and The Alan Turing Institute, UK: email: emine.yilmaz@ucl.ac.uk; H. Shen, School of Data and Computer Science, Sun Yat-sen University, China; and Department of Computer Science, University of Adelaide, Australia; email: shenh3@mail.sysu.edu.cn; M. de Rijke, Informatics Institute, University of Amsterdam, The Netherlands; email: derijke@uva.nl; W. B. Croft, College of Information and Computer Science, University of Massachusetts, U.S.A.; email: croft@cs.umass.edu.

DOI: http://dx.doi.org/10.1145/3057282

of the ambiguous query, retrieves documents for each of these aspects, and makes the search results more diverse. The underlying aspects of queries can be identified in various ways, for example, by query reformation with the help of a commercial search engine [37], by clustering search results [18], or by mining query logs, anchor texts, or the contents of the top-ranked documents, and so on [38]. By diversifying the search results, in the absence of any knowledge of user context or preferences, the chance that any user issuing an ambiguous query will find at least one of these results to be relevant is maximized [8].

To diversify search results, methods such as xQuAD (e*x*plicit *Qu*ery *A*spect *D*iversification) [37], query-specific clustering [18], and PM-2 (proportionality-based diversification Method – second version) [13] identify the underlying aspects of the query, compute the weights of each aspect and the probabilities of each document covering the aspects. With this information, these techniques select documents based on a combination of their relevance to the ambiguous query and relevance to the aspects. These approaches perform well in a large, static set of long documents. However, the problem of how to identify new or emerging aspects, compute their weights and obtain the probabilities of incoming documents in a stream of short texts to retrieve a diversified ranking of documents still needs to be further explored.

We address the problem of *search result diversification for streams of short texts given an ambiguous query at a certain point in time*. The input consists of an ambiguous query while the output varies over time and is a diversified ranked list of short documents covering as many recent aspects of the query as possible. Diversifying search results given a query in short text streams is of importance and has many applications. For instance, top-*k* publish/subscribe systems for tweets [7, 39] are required to return to a subscriber the top-*k* recent tweets that are relevant and diversified given a subscribed keyword. The problem of diversifying search results in long text streams has previously been investigated by Refs. [7, 33]. Both models penalize redundancy in a ranked list of documents in a stream, where redundancy is directly measured as a sum of pairwise similarities between any two documents. However, determining redundancy in a set of short documents, such as tweets or weibos, is challenging precisely because the documents are short [47]. Topic models seem a natural solution to this problem, but in the case of text streams the probabilities of aspects relevant to a given query may change over time.

We develop a dynamic Dirichlet multinomial mixture topic model (D2M3) that is able to capture the evolution of latent topics in a sequentially organized corpus of short documents. We propose a collapsed Gibbs sampling algorithm to infer latent topics for an ambiguous query, their dynamic weights (probabilities) of being relevant to the query, and the probability of a short document being relevant to the topics. Our dynamic mixture model does not assume the explicit availability of dynamic query aspects but infers these as well as the latent prior for a given query via the top-ranked short documents returned by a time-sensitive language model [12]. We also introduce an algorithm to diversify search results for short text streams that uses the information generated by our dynamic topic model. Instead of directly diversifying the search results based on document similarity, we use the dynamic weights of latent topics and the distribution of topics over documents in a text stream.

We evaluate our proposed algorithm for search result diversification in short text streams on a large dataset consisting of a three-month sample of Twitter and compare it to three types of search result diversification methods: (1) algorithms that do not consider data streams, such as xQuAD; (2) streaming diversification methods that work with data streams but that have been developed for long documents, such as in Refs. [7, 33]; and (3) algorithms that combine existing dynamic topic models with effective diversification retrieval models that have not been designed to work with text streams. Our approach outperforms state-of-the-art diversification methods in terms of a range of diversification metrics.

The main contributions of our work are as follows:

(i) We propose a dynamic Dirichlet multinomial mixture topic model that can track the changes of aspects of a given query and the multinomial distribution of aspects over documents.
(ii) We propose a collapsed Gibbs sampling algorithm for our dynamic Dirichlet multinomial mixture topic model to perform inference for search result diversification in text streams.
(iii) We propose a streaming version of the PM-2 diversification algorithm to perform diversification in response to a query at a certain point in time based on the dynamic information captured by our dynamic topic model.
(iv) We systematically analyze the proposed streaming diversification algorithm for short text streams and find that it significantly outperforms state-of-the-art streaming and non-streaming diversification algorithms.

The remainder of this article is organized as follows. Section 2 discusses related work. Section 3 provides an overview of the way we perform diversification in streams. Section 4 details the dynamic Dirichlet multinomial mixture model. Section 5 presents a modification of the PM-2 algorithm for diversification in text streams. Section 6 describes our experimental setup. Section 7 discusses our experimental results, and, finally, Section 8 concludes the article.

2. RELATED WORK

We discuss two lines of related work: search result diversification (streaming or not) and topic modeling (dynamic or not).

2.1. Search Result Diversification

Search result diversification has been studied as a task of re-ranking an initial ranking of documents retrieved for a query. The goal is to produce a more-diverse ranked list with respect to a set of aspects associated with the query [2, 13, 14]. Search result diversification is similar to ad hoc search but differs in its judgment criteria and evaluation measures [38]. The basic premise is that the relevance of a set of documents depends not only on the relevance of its individual 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.

Non-streaming Diversification. Non-streaming approaches to search result diversification work with a collection of documents where the dynamic characteristics of the underlying aspects and the latent topic distribution over documents are usually ignored.

Implicit approaches to search result diversification promote diversity by selecting a document that differs from the documents appearing before it in terms of vocabulary. An early influential article on implicit diversification concerns the Maximal Marginal Relevance (MMR) model [5], which reduces redundancy while maintaining query relevance when selecting a document. Zhai et al. [49] present an implicit subtopic retrieval model where the utility of a document is dependent on other documents in the ranking, and documents that cover many different subtopics of a query are found. Chen and Karger [6] describe a retrieval method incorporating negative feedback in which documents are assumed to be non-relevant once they are included in the result list. He et al. [18] propose a result diversification framework based on query-specific clustering and cluster ranking, in which diversification is restricted to documents belonging to clusters that potentially contain a high percentage of relevant documents. More recent implicit work includes set-based recommendation of diverse articles [1], term-level diversification [14], diversified data fusion [26], and neural-network-based diversification model [46]. Abbar et al. [1] address the problem of providing diverse news recommendations related to an input article by leveraging user-generated data to refine lists of related articles. They explore different diversity distances that rely on the content of user comments on articles such as sentiments and entities. Instead of trying to recover the topics for an ambiguous query, Dang and Croft [14] propose to use a simple greedy multi-document summarization algorithm for identifying topic terms for search result diversification from the initial ranking of documents. Liang et al. [26] start from the hypothesis that data fusion can improve performance in terms of diversity metrics, examine the impact of standard data fusion methods on search result diversification, and propose a diversified data fusion algorithm to infer latent topics of a query using topic modeling model for diversification. Xia et al. [46] propose to model the novelty of a document with a neural tensor network and learn a nonlinear novelty function based on the preliminary representation of the candidate document and other documents for diversification.

Explicit approaches to diversification assume that a set of query aspects are available and return documents for each of them. Well-known examples include xQuAD [37], RxQuAD [41], IA-select [2], PM-2 [13], and learning models for diversification [25, 27, 45]. Instead of modeling a set of aspects implicitly, these algorithms obtain a set of aspects either manually, for example, from aspect descriptions [9, 11], or they create them directly from, for example, suggested queries generated by commercial search engines [13, 37], or predefined aspect categories [40] or directly utilize the human judged labels of aspects for learning [25, 27, 45].

In contrast to previous algorithms, our proposed streaming diversification method is an implicit one and does not assume that aspects of the query are available but does assume that the underlying topics and the dynamic relevance of each topic can be inferred for search result diversification.

Streaming Diversification. Streaming approaches diversify search results in a text stream. To the best of our knowledge, only Minack et al. [33] and Chen and Cong [7] have previously investigated this problem. Minack et al. [33] propose two incremental diversification algorithms for data streams: MaxMinIncremental and MaxSumIncremental. Chen and Cong [7] propose a diversification algorithm for text streams called Diversity-Aware top-*k* Subscription (DAS). These methods process the input as a stream of documents and continuously maintain a diverse subset of documents at each position of the stream. They work with the same objective and try to maintain a set of *k* diversified documents $\mathbf{d}$ in a text stream that maximizes the function $f_{\text{div}}(\mathbf{d} \mid q) = (1 - \lambda) f_1(\mathbf{d} \mid q) + \lambda f_2(\mathbf{d})$, where $f_1(\mathbf{d} \mid q)$ measures the relevance of the set of documents to the query and $f_2(\mathbf{d})$ measures the dissimilarities of the documents as a set. These three streaming diversification methods differ in the way they compute $f_1(\mathbf{d} \mid q)$ and $f_2(\mathbf{d})$. To decide whether an incoming document should replace an old document, MaxMinIncremental only considers the minimum relevance of a document in the diversified document set to the query and the minimum pairwise distance in the set. MaxSumIncremental computes an average of the sum of dissimilarities between this candidate document and other documents and the average of the relevance scores to the query. DAS uses the same objective function as MaxSumIncremental for diversifying the top-*k* subscription for a query. All of the algorithms assume that the content of documents is rich, and it is easy to compute the similarities of document pairs for the objective function.

2.2. Topic Models

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, the Probabilistic LSI model [19], they have received significant attention [4, 17] and have been used in many retrieval tasks [26, 43].

Non-dynamic Topic Models. Non-dynamic topic models infer the topics in a static set of documents, the best-known of which is Latent Dirichlet Allocation (LDA) [4]. LDA represents each document as a finite mixture over "latent" topics where each topic is represented as a finite mixture over words in that document. Based on LDA, many extensions have been proposed, for example, to handle users' connections with particular documents and topics [36], to learn relations among different topics [23, 24], for topics over time [42], for ad hoc retrieval [43], or for rank aggregation [26]. LDA has also been extended to clustering [48] and tweet summarization [35]. The static topic model, Gibbs Sampling Dirichlet Multinomial Mixture model (GSDMM), for clustering proposed by Yin and Wang [48] is of particular interest for us, as this model works with static set of short documents, such as those in Twitter, infers a topic distribution for clustering, and represents each short document through a single topic. How to apply this previous method to streams of short document streams and do the inference is unknown.

Dynamic Topic Models. The Topic over Time (ToT) model [42] infers topics for offline documents with timestamps, makes the assumption that all the documents can only appear in a specific time interval (the time period is fixed), and normalizes the distribution of the timestamps before the inference. The Dynamic Topic Model (DTM) [3] captures the evolution of topics in a sequentially organized corpus of documents. It uses Gaussian series on the natural parameters of the multinomial topics and logistic normal topic proportion models and assumes that the mixture distributions of the documents have a Dirichlet prior that evolves over time. Unlike DTM and ToT, the Dynamic Mixture Model (DMM) [44] assumes that the mixture distribution for each document in streams does not have a Dirichlet prior, and, instead, such a distribution is directly dependent on the mixture distribution of the previous documents. The Topic Tracking Model (TTM) [21] and the online multi-scale topic model track time-varying consumer purchase behavior, in which consumers' interests and items' trends change over time. The Dynamic Clustering Topic (DCT) model [29] aims at clustering short documents rather than diversifying search results by a dynamic topic model, where topic distributions of the documents are assumed to change over time. The dynamic User Clustering Topic (UCT) model [51] and User Collaborative Interest Tracking (UCIT) model [28] propose to tackle the problems of user clustering in the context of streaming short texts by topic models. Twitter-LDA [50] is a topical keyphrase extraction LDA-based topic model and assumes that the content of documents generated from Twitter is rich enough for the inference of topic distributions. However, until now all dynamic topic models except DCT, UCT, UCIT, and Twitter-LDA make the strong assumption that documents arriving in a data stream are relatively long documents and provide a rich context for inference. DCT, UCT, and UCIT do work with streaming short text documents, but the goal of DCT is to cluster documents in streams and the goals of UCT and UCIT are to cluster users in the streams, respectively, rather than dynamically diversify search results for an ambiguous query. The goal of Twitter-LDA is to extract keyphrases from short texts in streams only. Thus, how to automatically diversify search results is still unknown. Our proposed topic model works with a large number of short text documents and is able to perform topic inference for dynamic diversification in streams.

3. DIVERSIFICATION FOR SHORT TEXT STREAMS

We first review our main notation and terminology.

Table I. Main Notations Used in Our Topic Model

Notation	Gloss
q	query
d	document
z	topic
t	time
w	word
V	number of unique words in vocabulary
Z	number of latent topics
z_d	topic assigned to document d
$\|d\|$	length of document d
$\mathbf{d}'_t$	documents arriving at time t
$\mathbf{d}_t$	document stream up to time t
$\mathbf{L}_t$	ranking of documents at time t
α_t	parameter of topic Dirichlet prior at time t
β_t	parameter of word Dirichlet prior at time t
$\mathbf{\Theta}_t$	dynamic topic distribution at time t
$\mathbf{\Phi}_t$	dynamic word distribution at time t
m_t	number of documents up to time t
$m_{t,z}$	number of documents assigned to topic z up to time t
$n_{t,z,v}$	number of words v assigned to topic z up to time t
$N_{d,v}$	number of words v in document d
N_d	length of document d
$n_{t,z,-d}$	number of words assigned to topic z up to time t except those in d
$m_{t,z,-d}$	number of documents assigned to topic z up to time t except document d

3.1. Notation and Terminology

We summarize our main notation in Table I. We distinguish among queries, aspects, and topics. A *query* is an expression of an information need. An *aspect* (sometimes called a *subtopic* of a query at the TREC Web track [11]) is an interpretation of an information need. For an ambiguous query, it usually has at least two aspects. We use *topic* to refer to latent topics as identified by a topic modeling method [4]. We refer to the method that we propose for diversification in short text streams as the *streaming diversification algorithm* (SDA); it builds on the proposed D2M3 (referred to as a *dynamic topic model*) and a modification of the PM-2 diversification algorithm.

3.2. The Diversification Task

The search result diversification task we address is this: Given a query q and a short text stream, retrieve a ranking of documents that covers as many aspects of the query as possible and that are relevant to the query. Specifically, we seek a ranking function f that satisfies

$$\mathbf{d}_t = \{\ldots, \mathbf{d}'_{t-2}, \mathbf{d}'_{t-1}, \mathbf{d}'_t\}, q \xrightarrow{f} \mathbf{L}_t,$$

where $\mathbf{d}_t$ is a sequentially organized corpus of short documents, with $\mathbf{d}'_t$ being the most recent set of documents arriving at (the current) time t, $\mathbf{L}_t$ is a ranking of diversified documents in response to query q at time t. A short text stream $\mathbf{d}_t$ comprises a sequence of short text documents, each denoted by a tuple $d = \langle \mathbf{w}_d, t_d \rangle$, where $\mathbf{w}_d$ is a sequence of words appearing in document d from the vocabulary $\mathbf{V} = \{v_1, v_2, \ldots, v_V\}$ and the size of $\mathbf{w}_d$ is no more than a specific predefined small number like in Twitter (where tweets are limited to 140 characters), and t_d is the creation time of d. We also consider the creation time of documents as the time they appear in the streams.

ALGORITHM 1: Streaming Diversification Algorithm

```
   Input:  A query $q$
           A time point $t$
           A short document stream up to $t$, $\mathbf{d}_t$
           Number of latent topics $Z$
           Original hyperparameters $\alpha_0$, $\beta_0$
   Output: A final diversified list of tweets $\mathbf{L}_t$.
   /* Part I: Infer latent topics                                    */
 1 Infer latent topics and their probabilities to $q$ at time $t$
 2 Infer tweets' probabilities to each topic at time $t$
   /* Part II: Perform diversification                               */
 3 Obtain top-$k$ recent and relevant tweets
 4 Diversify the top-$k$ tweets to construct $\mathbf{L}_t$
```

3.3. Overview of the Diversification Algorithm

We propose a search result diversification method for a short text streams, SDA, that can return a ranking of short documents that are recent and relevant to the query and cover as many aspects of the query as possible. Our diversification algorithm consists of two main parts: (i) infer latent topics by the proposed dynamic Dirichlet multinomial mixture model (discussed in Section 4) and (ii) perform diversification (discussed in Section 5); see Algorithm 1. In Part I in Algorithm 1, the diversification algorithm uses the proposed dynamic Dirichlet multinomial mixture model to infer latent topics of the input query and their current probabilities to the query (line 1 in Algorithm 1). These probabilities are likely to change over time, that is, some latent topics become more important but others not. The dynamic mixture model can also infer the relevance probabilities over topics specific to each document in the stream (line 2). In Part II in Algorithm 1, we first apply a time-sensitive retrieval model to obtain the top-k relevant documents (line 3) and then rerank the top-k documents by our proposed diversification algorithm based on PM-2 and the output of the proposed topic model, that is, the dynamic probabilities of latent topics and the probabilities over topics specific to each document (line 4).

Below we describe how to infer latent topics in Section 4, and in Section 5, we show how we use the information generated from latent topics to get a diversified ranking of documents in response to the query.

4. D2M3: A DYNAMIC DIRICHLET MULTINOMIAL MIXTURE TOPIC MODEL

Explicitly computing the probabilities of aspects of a query can improve diversification performance [2, 14, 37]. Following Ref. [26], we do not assume that aspect information is explicitly available; instead, we infer latent topics and their probabilities of being relevant using our proposed dynamic Dirichlet multinomial mixture topic model. We describe the details of the D2M3 model in the following.

Preliminaries. The goal of applying a dynamic model is to infer the dynamics of topics and the dynamics of documents' probabilities for each current topic z at time t. That is, we want to infer the dynamic probabilities of topics for a query q at time t, $P(z \mid t, q)$, and the dynamic probabilities of documents being relevant to the topics and q at time t, $P(d \mid t, z, q)$. For convenience and consistency with the notations used in many topic modeling approaches [3, 4, 21], we put $\mathbf{\Theta}_t = \{\theta_{t,z}\}_{z=1}^{Z}$, where $\mathbf{\Theta}_t$ is the dynamic topic distribution at time t with an element $\theta_{t,z} = P(z|t, q) > 0$, $\sum_{z=1}^{Z} \theta_{t,z} = 1$, and Z is the total number of latent topics. We also let $\mathbf{\Phi}_t = \{\phi_{t,z}\}_{z=1}^{Z}$, where $\mathbf{\Phi}_t$ is the word distribution over topics at time t, $\phi_{t,z}$ is the multinomial distribution of words specific to topic z at time t, the probability of v belonging to z at t, $\phi_{t,z,v} > 0$, and $\sum_{v=1}^{V} \phi_{t,z,v} = 1$. Here, v is a word, and V is the total number of different words in the vocabulary $\mathbf{V}$. In many non-dynamic LDA-style topic models, it is assumed that current topics are independent of the past topics and have a Dirichlet prior. With these assumptions, $\mathbf{\Theta}_t$ can be assumed to have the following Dirichlet prior:

$$P(\mathbf{\Theta}_t \mid \kappa) \propto \prod_{z=1}^{Z} \theta_{t,z}^{\kappa_z - 1}, \tag{1}$$

where $\kappa = \{\kappa_z\}_{z=1}^{Z}$ $(\kappa_z > 0)$ is a set of static Dirichlet parameters, and $\phi_{t,z}$ can be assumed to have the following Dirichlet prior:

$$P(\phi_{t,z} \mid \gamma) \propto \prod_{v=1}^{V} \phi_{t,z,v}^{\gamma_v - 1}, \tag{2}$$

where $\gamma = \{\gamma_v\}_{v=1}^{V}$ $(\gamma_v > 0)$ is a set of static Dirichlet parameters.

Capturing Previous Dependencies. The assumptions made in Equations (1) and (2) are not appropriate when it comes to a streaming datasetting, as the distributions at time t are independent on the past distributions. To model the dynamics of the topics underlying the ambiguous query q, following most dynamic topic models [21, 22, 44], we let the mean of the topics at the current time t be the same as those at a previous time unless otherwise confirmed by the set of newly arriving short documents $\mathbf{d}'_t$. Accordingly, we apply the following Dirichlet distribution for the prior of topics' current trends $\mathbf{\Theta}_t$,

$$P(\mathbf{\Theta}_t \mid \mathbf{\Theta}_{t-1}, \alpha_t) \propto \prod_{z}^{Z} \theta_{t,z}^{\alpha_{t,z}\theta_{t-1,z} - 1}, \tag{3}$$

where the Dirichlet prior κ in Equation (1) is factorized into the mean and precision $\kappa = \alpha_t \mathbf{\Theta}_{t-1}$, and $\alpha_t = \{\alpha_{t,z}\}_{z=1}^{Z}$ is a set of Dirichlet parameters $\alpha_{t,z}$ at time t. Here $\alpha_{t,z}$ represents the topic persistency, which is a measure of how consistently topic z maintains its relevance to query at time t compared with that at the previous time $t-1$. As the relevance of each topic is dynamic, we estimate $\alpha_{t,z}$ for each time period that depends on both t and z. This is a conjugate prior, and the inference can be done by Gibbs sampling [31]. We detail our inference procedure later in this section.

To model the dynamic changes of the multinomial distribution of words specific to topic z, we use the following Dirichlet distribution for the prior of the trends $\phi_{t,z} = \{\phi_{t,z,v}\}_{v=1}^{V}$:

$$P(\phi_{t,z} \mid \phi_{t-1,z}, \beta_{t,z}) \propto \prod_{v=1}^{V} \phi_{t,z,v}^{\beta_{t,z,v}\phi_{t-1,z,v} - 1}, \tag{4}$$

where the Dirichlet prior γ in Equation (2) is factorized into the mean and precision $\gamma = \beta_{t,z}\phi_{t-1,z}$, $\beta_{t,z} = \{\beta_{t,z,v}\}_{v=1}^{V}$ is a set of Dirichlet parameters $\beta_{t,z,v}$ at time t for word v and topic z, and $\beta_t = \{\beta_{t,z}\}_{z=1}^{Z}$. Here, $\beta_{t,z,v}$ represents the topic persistency of word v, which is a measure of how consistently word v maintains its probability of belonging to topic z at time t compared to that at time $t-1$. We detail inference for β_t later in this section.

Suppose we already have the distribution of topics at the previous time $t-1$, $\mathbf{\Theta}_{t-1}$, and the word distribution over topics at $t-1$, $\mathbf{\Phi}_{t-1}$. Our proposed dynamic Dirichlet multinomial mixture topic model is a generative process model that builds on $\mathbf{\Theta}_{t-1}$ and

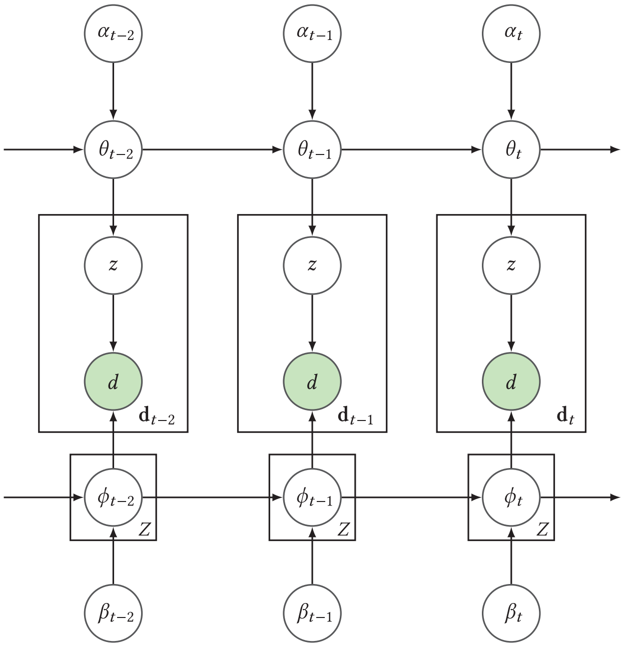


Fig. 1. Graphical representation of D2M3. Green shaded nodes indicate observed variables.

$\mathbf{\Phi}_{t-1}$. For $t = 0$, we can simply let $\theta_{0,z} = 1/Z$ and $\phi_{0,z,v} = 1/V$ as initialization. The generative process used in Gibbs sampling for parameter estimation for documents in stream $\mathbf{d}_t$ at time t is

(i) Draw a multinomial $\mathbf{\Theta}_t$ from a Dirichlet prior $\alpha_t \mathbf{\Theta}_{t-1}$;
(ii) Draw Z multinomials $\phi_{t,z}$ from a Dirichlet prior $\beta_{t,z}\phi_{t-1,z}$, one for each topic z;
(iii) For each document $d \in \mathbf{d}_t$ at time t, draw a topic z_d for a document d from multinomial $\mathbf{\Theta}_t$; then for each word v_{di} in document d:
    (a) Draw a word v_{di} from multinomial ϕ_{t,z_d}.

A graphical representation of this generative process is given in Figure 1. In the process, there is a fixed number of latent topics, Z, although a non-parametric Bayes version of our dynamic topic model that automatically integrates over the number of topics would certainly be possible. In the experiments, we set Z as follows: We vary the number of topics from 2 to 20 in the training dataset. The optimal number of topics is chosen based on the validation dataset and evaluated on the test dataset. See Section 6.5 for more details. We find that when the number of topics is equal to or greater than 8, the performance seems to level off. See Section 7.2 for more details. The posterior distribution of topics depends on the words in the documents. The parameterization of the proposed model is as follows:

$$\begin{aligned} \mathbf{\Theta}_t &\sim \text{Dirichlet}(\alpha_t \mathbf{\Theta}_{t-1}) \\ \phi_{t,z} | \beta_{t,z}\phi_{t-1,z} &\sim \text{Dirichlet}(\beta_{t,z}\phi_{t-1,z}) \\ z_d &\sim \text{Multinomial}(\mathbf{\Theta}_t) \\ v_{di} | \phi_{t,z_d} &\sim \text{Multinomial}(\phi_{t,z_d}). \end{aligned}$$

Inference. Inference is intractable in D2M3. Following References [17, 26, 42], we employ collapsed Gibbs sampling [17] to perform approximate inference. We adopt a conjugate prior (Dirichlet) for the multinomial distributions, and thus we can integrate out $\phi_{t,z}$ and $\boldsymbol{\Theta}_t$, analytically capturing the uncertainty associated with them. Thus, we do need not to sample $\phi_{t,z}$ and $\boldsymbol{\Theta}_t$.

In the Gibbs sampling procedure at time t, we need to calculate the conditional distribution $P(z_d \mid \mathbf{z}_{t,-d}, \mathbf{d}_t, \boldsymbol{\Phi}_{t-1}, \boldsymbol{\Theta}_{t-1}, \alpha_t, \beta_t)$, where $\mathbf{z}_{t,-d}$ represents the topic assignments for all documents in $\mathbf{d}_t$ except document d. We begin with the joint probability of the current document set $\mathbf{d}_t$, and, using the chain rule, we can obtain the conditional probability conveniently as

$$\begin{aligned} &P(z_d \mid \mathbf{z}_{t,-d}, \mathbf{d}_t, \boldsymbol{\Phi}_{t-1}, \boldsymbol{\Theta}_{t-1}, \alpha_t, \beta_t) \\ &\quad\propto \frac{m_{t,z} + \alpha_{t,z}\theta_{t-1,z} - 1}{\sum_{z=1}^{Z}(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - 1} \\ &\quad\times \frac{\prod_{v \in d} \prod_{j=1}^{N_{d,v}} (n_{t,z,v,-d} + \beta_{t,z,v}\phi_{t-1,z,v} + j - 1)}{\prod_{i=1}^{N_d} (n_{t,z,-d} + i - 1 + \sum_{v=1}^{V} \beta_{t,z,v}\phi_{t-1,z,v})}, \end{aligned} \tag{5}$$

where $m_{t,z}$ is the total number of documents in $\mathbf{d}_t$ assigned to topic z, v is a word, $N_{d,v}$ is the total number of the word v in document d, and $n_{t,z,v,-d}$ is the total number of the word v assigned to topic z at t except that in d. Note that in Equation (5), we consider the problem of documents being short in our setting. We tackle it by simply sampling one topic for all words in the same document, which is unlike previous dynamic topic models such as the TTM [21] and DTM [3] that sample different topics for different words in the same document. Previous topic models [48, 50] working with static short text datasets have shown that the strategy of sampling only one topic for the whole document when it is short is simple but effective.

The assumption that short documents tend to be about a single topic and the strategy that each short document is assigned to a single topic is also made and applied in other areas of information retrieval. For instance, Efron et al. [15] build on this assumption to improve the retrieval performance of short texts: Documents are expanded with a set of top-k short documents that are assumed to be about a single topic only. A detailed derivation of Gibbs sampling for our proposed D2M3 model is provided in Appendix A. In the sampling at each iteration, the persistency parameters α_t and β_t can be estimated by maximizing the joint distribution $P(\mathbf{d}_t, \mathbf{z}_t \mid \boldsymbol{\Phi}_{t-1}, \boldsymbol{\Theta}_{t-1}, \alpha_t, \beta_t)$.

We apply fixed-point iteration to get the optimal α_t and β_t at time t. The update rule for α_t for maximizing the joint distribution in our fixed-point iteration is derived by using two bounds in [34]

$$\alpha_{t,z} \leftarrow \frac{\alpha_{t,z}(\Psi(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - \Psi(\alpha_{t,z}\theta_{t-1,z}))}{\Psi(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - \Psi(\sum_{z=1}^{Z} \alpha_{t,z}\theta_{t-1,z})},$$

where $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}$ is the digamma function. To be able to specify the update rule for β_t, we introduce the following abbreviation: $\overline{\phi} = \phi_{t-1,z,v}$. Then, the update rule for β_t is

$$\beta_{t,z,v} \leftarrow \frac{\beta_{t,z,v}(\Psi(n_{t,z,v} + \beta_{t,z,v}\overline{\phi}) - \Psi(\beta_{t,z,v}\overline{\phi}))}{\Psi(\sum_{v=1}^{V} n_{t,z,v} + \beta_{t,z,v}\overline{\phi}) - \Psi(\sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi})},$$

where $n_{t,z,v}$ is the number of words v assigned to topic z in stream $\mathbf{d}_t$. Our derivation of the update rules for α_t and β_t and the two bounds used in deriving the update rules are detailed in Appendix B. An overview of our collapsed Gibbs sampling algorithm, including its input and output and the processes, is given in Algorithm 2.

After the Gibbs sampling procedure, with the fact that a Dirichlet distribution is conjugate to a multinomial distribution, we can easily infer the dynamic topic distribution

ALGORITHM 2: Inference for D2M3 at Time t

```
Input:  Previous topic distribution $\mathbf{\Theta}_{t-1}$
        Previous word distribution specific to topics $\mathbf{\Phi}_{t-1}$
        A set of short documents $\mathbf{d}_t$ at time $t$
        Initialized $\alpha_t$ and $\beta_t$
        Number of iterations $N_{iter}$
Output: Current topic distribution $\mathbf{\Theta}_t$
        Current word distribution specific to topics $\mathbf{\Phi}_t$
        Probabilities of topics relevant to query $q$ at time $t$,
        $P(z \mid t, q)$
        Documents' probabilities to each topic at time $t$,
        $P(z \mid t, d, q)$
Initialize topic assignment randomly for all documents in $\mathbf{d}_t$
for $iter$ = 1 to $N_{iter}$ do
    for $d$ = 1 to $|\mathbf{d}_t|$ do
        draw $z_d$ from $P(z_d|\mathbf{z}_{t,-d}, \mathbf{d}_t, \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)$
        update $m_{t,z_d}$ and $n_{t,z_d,v}$
    update $\alpha_t$ and $\beta_t$
Compute the posterior estimates $\mathbf{\Theta}_t$ and $\mathbf{\Phi}_t$
Compute $P(z \mid t, q)$ and $P(z \mid t, d, q)$
```

at time t, $\mathbf{\Theta}_t$ as

$$\theta_{t,z} = \frac{m_{t,z} + \alpha_{t,z}\theta_{t-1,z}}{\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}} = \frac{m_{t,z} + \alpha_{t,z}\theta_{t-1,z}}{m_t + \sum_{z=1}^{Z} \alpha_{t,z}\theta_{t-1,z}}, \tag{6}$$

where $m_t = |\mathbf{d}_t|$ is the total number of documents in $\mathbf{d}_t$, and infer multinomial distributions over words for topic z at time t as

$$\phi_{t,z,v} = \frac{n_{t,z,v} + \beta_{t,z,v}\overline{\phi}}{\sum_{v=1}^{V} n_{t,z,v} + \beta_{t,z,v}\overline{\phi}} = \frac{n_{t,z,v} + \beta_{t,z,v}\overline{\phi}}{n_{t,z} + \sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}},$$

where $n_{t,z}$ is the number of words assigned to topic z at time t.

For convenience, we write $P(z \mid t, q)$ (the probability of topic z being relevant to q at time t) to denote $\theta_{t,z}$. After the iterations, each short document is assigned to a specific topic z. To compute the probability of a topic z being relevant to a document d given a query q and t, that is, $P(z \mid t, d, q)$, instead of directly setting $P(z \mid t, d, q) = 1$ if d is assigned to z by $P(z_d \mid \mathbf{z}_{t,-d}, \mathbf{d}_t, \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)$ as defined in Equation (5), we set

$$P(z \mid t, d, q) = \frac{P(z \mid \mathbf{z}_{t,-d}, \mathbf{d}_t, \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)}{\sum_{z'=1}^{Z} P(z' \mid \mathbf{z}_{t,-d}, \mathbf{d}_t, \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)}. \tag{7}$$

Online Computational Efficiency Analysis of D2M3. In practice, the retrieval system is required to quickly retrieve a rank list of documents in respond to a given query. Instead of inferring $P(z|t, d)$ for all documents streaming in at query time t using the proposed Gibbs sampling online before in response to the query, we approximately infer $P(z \mid t, d)$ as

$$P(z \mid t, d) = \frac{1}{E} \prod_{v \in d} P(z \mid t-1, v) = \frac{1}{E} \prod_{v \in d} \phi_{t-1,z,v},$$

with computational complexity $O(|\mathbf{d}'_t|)$, which is linear in the number of streaming documents, $|\mathbf{d}'_t|$, at time slice t, where $E = \sum_{z'=1}^{Z} \prod_{v \in d} P(z' \mid t-1, v) = \sum_{z'=1}^{Z} \prod_{v \in d} \phi_{t-1,z',v}$ is a normalization constant. Here $\phi_{t-1,z,v}$ can be exactly inferred using the proposed Gibbs

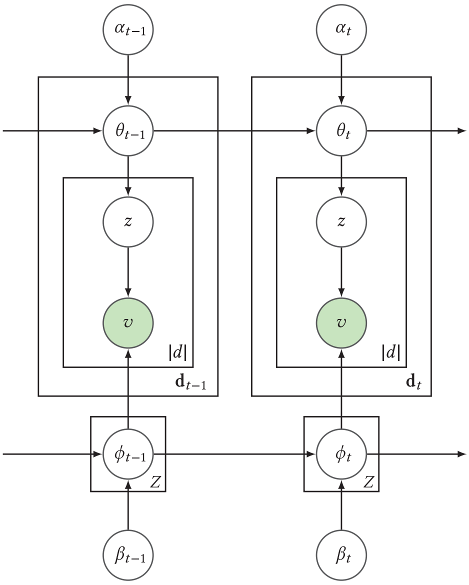


Fig. 2. Graphical representation of the TTM.

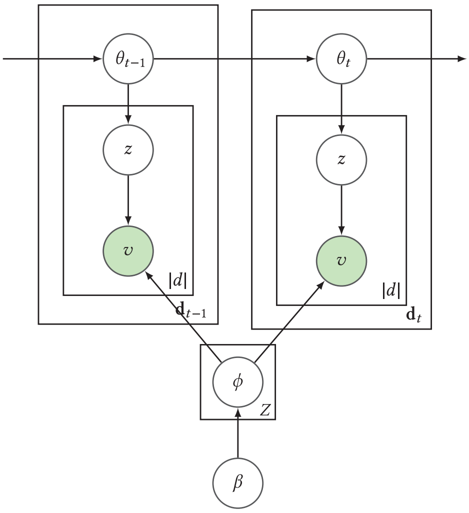


Fig. 3. Graphical representation of the DMM.

sampling algorithm offline before the query time t. In other words, D2M3 can track topic changes over time offline until at least time slice $t-1$, and on the basis of this the current topic changes can be approximately inferred by previous time slices. Tracking topic changes offline is acceptable in many applications such as top-k publish/subscribe for text stream [7], in which the diversified subscription results are only required to be presented to a subscriber once he logs into the system.

Comparison between D2M3 and other Dynamic Topic Models. To further understand our proposed topic model, D2M3, we compare D2M3 with two well-known and effective dynamic topic models that are used as baselines in our experiments, the TTM [21] and DMM [44], the graphical representations of which are shown in Figure 2 and Figure 3, respectively. As can be seen in Figure 2 and Figure 3, compared to the graphical representation of D2M3 in Figure 1, both the TTM and DMM assume that documents are long enough for topic inference, and thus each document is modelled to be a mixture of multiple topics. The generative process of TTM is as follows: At time t, for each document $d \in \mathbf{d}_t$, TTM first draws a multinomial $\theta_{t,d}$ from a Dirichlet distribution with parameter $\alpha_t \mathbf{\Theta}_{t-1}$, and then for each word v in the document, draws a topic z from the multinomial $\theta_{t,d}$ and draws the word v from the multinomial $\phi_{t,z}$ that is drawn from a Dirichlet distribution $\beta_t \mathbf{\Phi}_{t-1}$. In contrast, the generative process of DMM is as follows: at time t, for each document $d \in \mathbf{d}_t$, DMM first draws a multinomial $\theta_{t,d}$ from a multinomial distribution with expectation $\theta_{t-1,d}$ and then, for each word v in the document, draws a topic z from the multinomial $\theta_{t,d}$ and draws the word v from the static multinomial ϕ_z that is drawn from a Dirichlet distribution with parameter β. The main difference, then, between D2M3 and TTM and DMM is that D2M3 works with short documents and thus assumes that words in the same document share a single topic only. Previous work has found that short documents are likely to talk about a single topic only and topic models that assign a single topic to each short document work better than those that assign multiple topics [48]. As the graphical model of D2M3 differs from other dynamic topic models, inference for D2M3 differs as well.

5. SDA: A STREAMING DIVERSIFICATION ALGORITHM

In this section, we provide a way to diversify documents in a stream of short text documents in response to a query. In Section 5.1, we briefly describe how PM-2 works for diversification in a static set of documents. In Section 5.2, we detail our proposed streaming version of PM-2, which performs diversification in response to a query at time t based on the dynamic information captured by D2M3.

5.1. A Diversification Method: PM-2

Before we discuss our proposed modification of the PM-2 diversification algorithm, we briefly describe PM-2 [13, 14]. PM-2 is an election-based approach to search result diversification. It is a probabilistic adaptation of the Sainte-Laguë method for assigning seats (positions in the final 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 or aspect probabilities, that is, $P(z \mid q)$) they receive.

PM-2 starts with a ranked list $\mathbf{L}_f$ with k empty seats, and a set of top-k documents, $\mathcal{R}$, returned by a retrieval model in response to q. For each of the seats, it computes the quotient $qt[z|q]$ for each topic z given q following the Sainte-Laguë formula:

$$qt[z|q] = \frac{v_{z|q}}{2s_{z|q} + 1}, \tag{8}$$

where $v_{z|q}$ is the probability of topic z given q, that is, $P(z \mid q)$, and $s_{z|q}$ is the "number" of seats occupied by topic z (in initialization, $s_{z|q}$ is set to 0 for all topics). According to the Sainte-Laguë method, seats should be awarded to the topic with the largest quotient 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 d^* to fill this seat is the one that is not only relevant to z^* but to other topics as well:

$$d^* = \mathop{\arg\max}_{d \in \mathcal{R}} \left(\lambda \cdot qt[z^*|q] \cdot P(d \mid z^*, q) + (1-\lambda) \sum_{z \neq z^*} qt[z|q] \cdot P(d \mid z, q) \right), \tag{9}$$

where $P(d \mid z, q)$ is the probability of d talking about topic z for a given q. After the document d^* is selected, PM-2 adds d^* as a result document, that is, $\mathbf{L}_f \leftarrow \mathbf{L}_f \cup \{d^*\}$; removes it from $\mathcal{R}$, that is, $\mathcal{R} \leftarrow \mathcal{R} \backslash \{d^*\}$; and increases the "number" of seats occupied by each of the topics z by its normalized relevance to d^*:

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

This process repeats until we get k documents for $\mathbf{L}_f$ or we are out of candidate documents. The order in which a document is appended to $\mathbf{L}_f$ determines its ranking.

5.2. Integrating PM-2 and D2M3

We face three challenges in PM-2: (1) It does not take the changes of distributions of aspects over time into account; (2) it is non-trivial to get the aspect probability $v_{z|q}$, which is often set to be uniform; and (3) it is non-trivial to compute $P(d \mid z, q)$, which usually requires explicit access to additional information.

We add time as a new component into PM-2 and make it time sensitive to address the *first* challenge. The model is described in Algorithm 3. In our time-sensitive version of PM-2, to address the *second* challenge, we compute $v_{z|t,q}$ by (6), that is, $v_{z|t,q} = P(z \mid$

ALGORITHM 3: Modified Version of PM-2. The Differences with the Original Version of PM-2 Are: (1) It Can Diversify Results in Streams; (2) It Can Infer the Aspect Probability to q at t; and (3) It Can Compute Document Probabilities Given Aspects of q at t.

```
Input:  A query $q$
        A set of streaming short documents $\mathbf{d}_t$
        Current topic distribution $\mathbf{\Theta}_t$
        Current word distribution specific to topics $\mathbf{\Phi}_t$
        Probabilities of topics relevant to query $q$ at time $t$,
        $P(z \mid t, q)$
        Documents' probabilities to each topic at time $t$,
        $P(z \mid t, d, q)$
Output: A diversified ranking of documents $\mathbf{L}_t$
$\mathbf{L}_t \leftarrow \varnothing$
$\mathcal{R} \leftarrow \mathbf{d}_t$
for $d = 1, \dots, |\mathbf{d}_t|$ do
    Compute $P(d \mid t, q)$ by a time-sensitive language model
for $z = 1, \dots, Z$ do
    $v_{z|t,q} \leftarrow P(z \mid t, q)$
for all positions in the ranked list $\mathbf{L}_t$ do
    for $z = 1, \dots, Z$ do
        $qt[z|t,q] = \frac{v_{z|t,q}}{2s_{z|t,q} + 1}$
    $z^* \leftarrow \arg\max_z qt[z|t,q]$
    $d^* \leftarrow \arg\max_{d \in \mathcal{R}} \lambda \times qt[z^*|t,q] \times P(d \mid t, z^*, q) + (1-\lambda) \sum_{z \neq z^*} qt[z|t,q] \times P(d \mid t, z, q)$
    $\mathbf{L}_t \leftarrow \mathbf{L}_t \cup \{d^*\}$                /* append $d^*$ to $L_f$ */
    $\mathcal{R} \leftarrow \mathcal{R} \backslash \{d^*\}$
    for $z = 1, 2, \dots, T$ do
        $s_{z|q} \leftarrow s_{z|q} + \frac{P(d^* \mid t, z, q)}{\sum_{z'=1}^{Z} P(d^* \mid t, z', q)}$
```

$t, q) = \theta_{t,z}$, such that (8) at time t is changed to

$$qt[z|t,q] = \frac{P(z \mid t, q)}{2s_{z|t,q} + 1} = \frac{m_{t,z} + \alpha_{t,z}\theta_{t-1,z}}{(2s_{z|t,q} + 1) \cdot (m_t + \sum_{z}^{Z} \alpha_{t,z}\theta_{t-1,z})},$$

where $qt[z|t,q]$ is the quotient for topic z given q at time t, $s_{z|t,q}$ is the "number" of seats occupied by topic z given q at time t (in initialization, $s_{z|t,q}$ is set to 0 for all topics).

For the *third* challenge, instead of explicitly computing the probability of document d being relevant to topic z at time t, $P(d \mid t, z, q)$, we apply Bayes' Theorem so

$$P(d \mid t, z, q) = \frac{P(z \mid t, d, q)P(d \mid t, q)}{P(z \mid t, q)} = \frac{P(z \mid t, d, q)P(d \mid t, q)}{v_{z|t,q}}, \tag{10}$$

where $P(d \mid t, q)$ is the probability of d being relevant to q at time t obtained by a time-sensitive language model, and, similarly, $v_{z|t,q}$ is the probability of topic z relevant to q at time t, that is, $v_{z|t,q} = P(z \mid t, q)$. As a result, after applying Equation (10) to Equation (9) (replacing $P(d \mid z, q)$ in Equation (9) by $P(d \mid t, z, q)$ in Equation (10)), we select a candidate document by

$$\begin{aligned} d^* = & \arg\max_{d \in \mathcal{R}} \ \lambda \cdot qt[z^*|t,q] \cdot \frac{P(z^* \mid t,d,q) \cdot P(d \mid t,q)}{v_{z^*|t,q}} \\ & + (1-\lambda) \cdot \sum_{z \neq z^*} qt[z|t,q] \cdot \frac{P(z \mid t,d,q) \cdot P(d \mid t,q)}{v_{z|t,q}}, \end{aligned} \tag{11}$$

where $P(z \mid t,d,q)$ is the probability of document d belonging to topic z in response to q at time t, which can easily be inferred in our dynamic mixture model by Equation (7). Now, let $\overline{M}(x)$ abbreviate $m_{t,x} + \alpha_{t,x}\theta_{t-1,x}$ and let $\overline{P}(x)$ abbreviate $P(x \mid \mathbf{z}_{t,-d}, \mathbf{d}_t, \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)$. Then, after applying Equation (6) and Equation (7), Equation (11) can be rewritten as

$$\begin{aligned} d^* = & \arg\max_{d \in \mathcal{R}} \ \lambda \cdot qt[z^*|t,q] \cdot \frac{\overline{P}(z^*) \cdot P(d \mid t,q)}{\overline{M}(x) \cdot \sum_{z'}^{Z} \overline{P}(z')} \\ & + (1-\lambda) \cdot \sum_{z \neq z^*} qt[z|t,q] \cdot \frac{\overline{P}(z) \cdot P(d \mid t,q)}{\overline{M}(z) \cdot \sum_{z'}^{Z} \overline{P}(z')}, \end{aligned} \tag{12}$$

where we ignore the constant term $m_t + \sum_{z=1}^{Z} \alpha_{t,z}\theta_{t-1,z}$, as it has no impact on selecting the candidate document d^*.

We use **SDA** to refer to our streaming diversification method as described in Algorithm 1, with D2M3 and modified version of PM-2 as detailed in Algorithms 2 and 3, respectively.

6. EXPERIMENTAL SETUP

6.1. Research Questions

The research questions guiding the remainder of the article are as follows.

Concerning the performance of SDA:

RQ1 How does SDA compare against a baseline time-sensitive retrieval run, against non-streaming diversification methods, and against streaming diversification methods on short text streams, in terms of traditional retrieval measures?

RQ2 How does SDA compare against a baseline time-sensitive retrieval run, against non-streaming diversification methods, and against streaming diversification methods on short text streams, in terms of diversity measures?

Concerning the contribution of D2M3 to SDA:

RQ3 How does the contribution of our topic model D2M3 to the overall performance of SDA compare to the contribution of other topic models, in terms of traditional retrieval measures?

RQ4 Do the latent topics generated by D2M3 enhance the diversity performance of SDA compared to other topic models?

RQ5 How does the contribution of our topic model D2M3 to the overall performance of SDA compare to the contribution of other topic models, in terms of diversity measures?

RQ6 Can our SDA retrieve a competitive number of subtopics per query?

RQ7 Does our SDA outperform the best diversification baseline method on each query?

RQ8 Is the performance of SDA and the baseline models sensitive to the number of latent topics?

To answer **RQ1** and **RQ2**, we run a series of contrastive experiments, see Section 7.1, and report on the outcomes in terms of relevance and diversity retrieval performance measures. To answer **RQ3–RQ8**, we modify SDA, replace D2M3 with other topic models, report on relevance, clustering, and diversity retrieval performance, and analyze the outcomes per query and in terms of subtopics retrieved and sensitivity to the number of latent topics; see Section 7.2.

6.2. Dataset

There are publicly available labeled corpora, such as the Tweets 2011 and Tweets 2013 datasets, that have been used for ad hoc retrieval in the TREC 2011–2015 Microblog track [30]. However, they have not been created for a diversification task, the queries that make up the datasets are too long and specific, and no aspects of the queries have been identified for evaluation purposes. Furthermore, the timespan of the collection is relatively small and the ground truth is static for all the queries over the time. Thus they are unsuitable for our experimental purposes.

We work with publicly downloadable posts that were a 1% sample from Twitter as a short text stream.[1] The tweets were posted between February 1, 2015, and April 30, 2015, covering a period of about 90 days. Most tweets are written in English. We remove non-English tweets and retweets (which increase redundancy in the retrieved documents for a given query), leaving us with 396 million tweets.

To evaluate the performance of our proposed diversification algorithm, SDA, and the baseline algorithms, we need to manually create a set of queries, their corresponding aspects, and the ground-truth judgments, that is, whether a document is relevant to a given query and to which aspect of the query. To create the ground truth, we follow the process in Reference [16] to generate ambiguous queries that contain no more than two keywords, the aspects and the relevant documents specific to the aspects. The process we used is as follows:

(1) Generate a set of ambiguous queries by manual selection from a list of hashtags in the whole dataset. Hashtags related to topics of general interest were selected. This created a list of hashtags such as "#Apple" and "#Egypt." Text queries were created from these tags manually, resulting in queries such as "Apple" and "Egypt" that will be used for the whole time period.
(2) For each query at time t, find a list of k associated hashtags. This was done by simply identifying the tags with co-occurrence with the hashtag that was the basis of the query. Aspects were manually generated based on these k associated hashtags, resulting in 2 to at most 10 aspects for this query at time t.
(3) Given a query at time t, manually labeled the top-k documents retrieved by a time-sensitive language model (see Section 6.3) for its aspects, resulting in the query-aspect-document ground truth used in our experiments.

As the way we generated our ground truth is time consuming, for each query we only manually labeled the data every 20 days, resulting in 5 sets of ground truth, for February 9, March 1, March 21, April 10, and April 30, 2015, respectively. To complete the process for generating the ground truth, 23 students with different backgrounds but all in possession of intermediate or high-level English certifications at a Chinese university were invited as annotators to label the data. They were given a list of ambiguous queries generated in step 1 in the process across the whole days and were asked to pick up queries based on the hashtags they were interested in. After that, on each day, that is, February 9, March 1, March 21, April 10, and April 30, 2015, for each

[1] The dataset can be downloaded from https://archive.org/details/twitterstream.

Table II. Aspects Used to Evaluate the Ranking of Documents in Response to Three Example Ambiguous Queries over the 5 Evaluation Days, February 9, March 1, March 21, April 10, and April 30, 2015, Respectively

Queries	Aspects on Feb. 9, 2015	Aspects on Mar. 1, 2015	Aspects on Mar. 21, 2015	Aspects on Apr. 10, 2015	Aspects on Apr. 30, 2015
Boston	Snow, Job, News, Education, Sports, Business	Snow, Chinese-newyear, Job, News, Education, Sports, Business	Bombing, Snow, Job, News, Education, Sports, Business	Boston2024, Job, News, Education, Sports, Business	Boston2024, Job, News, Education, Sports, Business
Apple	Report, Macbook, Food, iPhone, iPad	Investment, Update, Macbook, Food, iPhone, iPad	ResearchKit, AppleWatch, Macbook, Food, iPhone, iPad	CareKit, AppleWatch, Macbook, Food, iPhone, iPad	Conference, Update, AppleWatch, Macbook, Food, iPhone, iPad
Obama	Germany, Jordan, NHLChampion, BilateralMeeting	Law, Remark, Qatar, Liberia, BilateralMeeting	WhiteHouse, Law, BilateralMeeting, Ireland, videoconference	memorandum, Iraq, Panama, CARICOM, Afghanistan	Remark, Energy, WhiteHouse, Japan, Honor, BilateralMeeting

query a result list consisting of the top-500 retrieved documents produced by our time-sensitive language model (LM) as a baseline algorithm was provided to the annotators, respectively. Annotators were required to identify a number of associated tags with co-occurrence with the hashtag that was the basis of the query as aspects of the queries in step 2, based on the content of the top-*k* retrieved documents and the associated tags. They produced judgements on whether the documents were relevant to the queries and to which aspects in step 3 in the process. Hence, for a specific date, the annotators only saw the tweets up to that date, and the tags obtained by the annotators only represent the aspects specific to that date, as desired. In our evaluation, for all the baselines and our SDA algorithm, we assume that any documents that were not observed by annotators in the labeling process, that is, documents that were ranked lower than the top-500 position by the LM baseline, are non-relevant. To reduce annotators' workload for the labeling task, all tweets retrieved in response to a query at a specific time for a given aspect were labeled once.

The process resulted in a total of 107 ambiguous queries on each test day. For some queries, new aspects may appear and old ones may be ignored by the annotators and the decision of which were made by annotators themselves. The number of aspects per query changes over time. On average, we have 3.7, 4.4, 5.2, 6.0, and 6.8 aspects per query on the 5 selected dates, respectively.

Table II shows dynamic aspects of three ambiguous queries over the 5 test days. Aspects used to evaluate the ranking of documents in response to the ambiguous query "Boston" over the 5 different test days were generated by the annotators based on the following events: It was snowing heavily in Boston in February and the seasonal snowfall record was broken with 108.6 inches on March 16, 2015. Many people talked about the Chinese new year festival that started from February 19 and ended around March 1, 2015, in Boston. Dzhokhar Tsarnaev was found guilty on all charges in the Boston Marathon bombing event on April 8, 2015, and afterwards people recalled and discussed the bombing event that happened in Boston in 2013. In early 2015, Boston was chosen by the United States Olympic Committee to compete with other candidates around the world to bid for the 2024 Summer Olympics. News about Boston 2024 became popular from April 2015.

Table III. Our Diversification Methods and the Baselines Used for Comparison

Acronym	Gloss
The proposed streaming diversification methods integrating with different topic models	
SDA	Streaming diversification algorithm integrating with D2M3
SDA$_{TTM}$	Streaming diversification algorithm integrating with TTM
SDA$_{DMM}$	Streaming diversification algorithm integrating with DMM
Non-streaming diversification methods	
LM	Time-sensitive language model
MMR	Maximal marginal relevance model
xQuAD	Explicit query aspect diversification model
PM-2	An election-based approach to search result diversification model
Traditional streaming diversification methods	
MMInc	Incremental diversification algorithm with MaxMin objective
MSInc	Incremental diversification algorithm with MaxSum objective
PM-2 framework-based diversification methods integrating with different topic models	
PM-2$_{GSDMM}$	PM-2 diversification method integrating with GSDMM
PM-2$_{LDA}$	PM-2 diversification method integrating with LDA

6.3. Baselines

We list our proposed diversification methods and the baselines that we consider for comparison in Table III. To address **RQ1** and **RQ2**, we compare SDA to (1) a non-streaming non-diversified retrieval baseline, viz. a time-sensitive language model (LM) [12]; (2) three non-streaming diversification baselines, viz. MMR [5], xQuAD [37], and PM-2 [13]; and (3) two state-of-the art streaming diversification algorithms, MMInc, which abbreviates MaxMinIncremental, and MSInc, which abbreviates MaxSumIncremental [33]. Diversity-Aware top-*k* Subscription (DAS) [7] uses the same objective function as MSInc for diversifying the top-*k* subscription for a query and generates the same results; hence, we do not report on experimental results for DAS.

To address **RQ3**–**RQ8**, we contrast SDA (with D2M3) with two variations of SDA obtained by swapping out D2M3: SDA$_{TTM}$ and SDA$_{DMM}$. SDA$_{TTM}$ first utilizes a dynamic topic model, viz. the TTM [21], to infer the multinomial distribution of topics specific to each document in the top-*k* results returned by the LM model and then applies the modified PM-2 algorithm to diversify the top-*k* results. SDA$_{DMM}$ utilizes the dynamic topic model, DMM [44], to infer topics for the top-*k* documents returned by LM and then applies the modified PM-2 method for diversification. To understand whether dynamic topic modeling is more effective than static topic modeling when using the same diversification framework, that is, the PM-2 framework in RQ2, and whether the performance improvement of SDA is simply due to amelioration of vocabulary mismatch, we consider two additional baseline diversification algorithms PM-2$_{GSDMM}$ and PM-2$_{LDA}$. Here, PM-2$_{GSDMM}$ and PM-2$_{LDA}$ first apply static topic models, the GSDMM [48] and the LDA, to the top-*k* documents retrieved by our LM baseline, respectively. Then, they diversify the top-*k* documents[2] by Equation (9), where $P(d \mid z^{*}, q)$ is set to be the distribution $\theta_{z,d}$ inferred by GSDMM and LDA, respectively.

6.4. Evaluation Metrics

For evaluating regular retrieval performance, we use nDCG, ERR, Prec@*k*, and MAP. We use the following diversity metrics for evaluation, most of which have been used as official evaluation metrics at the TREC Web track [11] and in the literature on search result diversification: normalized discounted cumulative gain at *k* (α-nDCG@*k*) [8], subtopic recall at *k* (S-Recall@*k*) [49], intent-aware expected reciprocal rank at *k*

[2] We let $k = 500$ in our experiments and found that when $k \geq 300$ the performance levels off.

(ERR-IA@k) [2, 13], intent-aware precision at k (Prec-IA@k) [2], intent-aware MAP at k (MAP-IA@k) [2], and novelty- and rank-biased precision (NRBP) [10].

For evaluating the quality of the latent topics generated by our D2M3 topic model, we use Purity [32], Normalized Mutual Information (NMI) [32], and Adjusted Rank Index (ARI) [32], which are widely used in the literature of traditional clustering. Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_G\}$ be a set of ground-truth clusters (aspects that the documents are assigned to according to the ground truth) and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_j, \ldots, \mathbf{y}_Z\}$ be the set of output clusters (topics that the documents are assigned to by D2M3) at time t, where G and Z are the total number of the clusters in the ground truth and the output clusters, respectively. Then, these metrics can be computed as follows:

Purity. To compute purity, each output cluster $\mathbf{y}$ is assigned to the ground-truth cluster $\mathbf{x}$ that is most frequent in the cluster, and the the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N. Here, N is the total number of documents in $\mathbf{X}$. Formally it is defined as

$$\mathrm{Purity}(\mathbf{X}, \mathbf{Y}) = \frac{1}{N} \sum_j \max_i |\mathbf{y}_j \cap \mathbf{x}_i|,$$

where $|\mathbf{y}_j \cap \mathbf{x}_i|$ is the number of documents in the intersection $\mathbf{y}_j \cap \mathbf{x}_i$.

NMI. High purity is easy to achieve when the number of clusters is large. In particular, purity is 1.0 if each document gets its own cluster. Thus, we cannot simply use purity to trade off the quality of the clustering against the number of clusters. NMI is a measure that does allow us to make this tradeoff:

$$\mathrm{NMI}(\mathbf{X}, \mathbf{Y}) = \frac{I(\mathbf{X}; \mathbf{Y})}{[E(\mathbf{X}) + E(\mathbf{Y})]/2} = \frac{\sum_{i,j} \frac{|\mathbf{y}_j \cap \mathbf{x}_i|}{N} \log \frac{N|\mathbf{y}_j \cap \mathbf{x}_i|}{|\mathbf{y}_j||\mathbf{x}_i|}}{\left(-\sum_i \frac{|\mathbf{x}_i|}{N} \log \frac{|\mathbf{x}_i|}{N} - \sum_j \frac{|\mathbf{y}_j|}{N} \log \frac{|\mathbf{y}_j|}{N}\right) \Big/ 2},$$

where $I(\mathbf{X}; \mathbf{Y})$, $E(\mathbf{X})$, and $E(\mathbf{Y})$ are the mutual information, the entropy of $\mathbf{X}$ and of $\mathbf{Y}$, respectively. According to NMI, when $\mathbf{Y}$ is the same to $\mathbf{X}$, NMI achieves a value of 1, its largest value.

ARI. Consider a situation where one clusters documents based on a series of pairwise decisions. If two documents both in the same cluster are aggregated into the same cluster and two documents in different clusters are aggregated into different clusters, then the decision is considered to be correct. The Rand index shows the percentage of decisions that are correct while the adjusted Rand index is the corrected-for-chance version of the Rand index [20]. The maximum value is 1 for an exact match; larger values mean better performance for clustering. ARI($\mathbf{X}, \mathbf{Y}$) is computed as

$$\mathrm{ARI}(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i,j} \binom{|\mathbf{y}_j \cap \mathbf{x}_i|}{2} - \left[\sum_j \binom{|\mathbf{y}_j|}{2} \sum_i \binom{|\mathbf{x}_i|}{2}\right] \Big/ \binom{N}{2}}{\frac{1}{2}\left[\sum_j \binom{|\mathbf{y}_j|}{2} + \sum_i \binom{|\mathbf{x}_i|}{2}\right] - \left[\sum_j \binom{|\mathbf{y}_j|}{2} \sum_i \binom{|\mathbf{x}_i|}{2}\right] \Big/ \binom{N}{2}}.$$

We assign only one topic $z = \arg\max_z P(z \mid t, d, q)$ to document d when we evaluate the quality of the topics generated by the underlying topic model (for the purpose of getting the purity, NMI, and ARI evaluation results only).

We follow previous work [11, 13, 26, 46] on search result diversification and compute the metric scores at depth 20. We report on scores per day and on scores averaged over the 5 days.

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 $^{\vartriangle}$ (and $^{\triangledown}$) for $\alpha = .05$.

6.5. Training and Parameter Settings

For the time-sensitive language model baseline, LM, we rank the documents by $P(d \mid t, q) = P(d \mid q) \cdot b^{-(t-t_d)}$, where b is a base parameter that determines the rate of the recency decay and t_d is the creation time of document d. The other baselines also adopt this setting to compute $P(d \mid t, q)$ to obtain the relevance of a document to a query at time t. For our diversification model for short text streams, SDA, we perform the proposed dynamic Dirichlet multinomial mixture topic model, D2M3, on the top-k documents (we found that the performance of utilizing more than top 100 documents is almost the same; we use the top-500 documents for inference in our experiments) retrieved by the time-sensitive language model LM at time t from the short text stream $\mathbf{d}_t$ up to time t.

For evaluation purposes, for our proposed algorithm SDA and all the baseline algorithms, we use a 60/30/10 split of all the 107 ambiguous queries for our training, validation, and test sets, respectively. Specifically, on each test day, for each split of the 107 ambiguous queries, we conduct our training using the queries in the training set that make up 60% of the ambiguous queries and all tweets posted on or before the test day; we validate the algorithms using the queries in the validation set consisting of 30% of the ambiguous queries and the posts on or before the day, and we report the performance of the algorithms using the remainder of the queries, that is, 10% of the ambiguous queries and the posts on or before the day. We also report the mean performance of all these 5 test days when necessary in the analysis.

We train SDA, the baseline PM-2, and the variants SDA$_{TTM}$, SDA$_{DMM}$, PM-2$_{GSDMM}$, and PM-2$_{LDA}$ using values of λ (see Equations (9) and (12)) varying from 0 to 1.0 and varying the number of topics from 2 to 20. The best λ value and the number of topics are then chosen based on the validation set and evaluated on the test queries.

Similarly, for the baseline MMR, in the training we vary the parameter λ from 0 to 1.0; recall that it governs the linear mixture of a candidate document's relevance to the input query and the minimal similarity of the candidate document to the previously selected documents. The best λ value is then chosen based on the validation set and evaluated on the test queries.

For the baseline xQuAD, we vary the parameter λ from 0 to 1.0 that governs the probability of a candidate document's relevance to the input query and $p(d, \bar{\mathcal{S}}|q)$, that is, the probability of observing the candidate document but not the documents already in the previous selected document set $\mathcal{S}$. Again, the best λ value is then chosen based on the validation set and evaluated on the test queries. The same setting is applied for parameter λ used in the two streaming diversification algorithms, MMInc and MSInc.

In terms of aspects used for each query in the baseline xQuAD, we follow Ref. [37] and apply query reformulation techniques for the aspect generation. Specifically, we directly append each aspect of the initial query that is manually identified in the ground truth at time t to the initial ambiguous query q itself as a sub-query $q_{i,t}$ in xQuAD. We estimate the sub-query importance component, $p(q_{i,t}|q)$, in our baseline xQuAD as $p(q_{i,t}|q) = \frac{1}{|\mathcal{Q}_{t,q}|}$, where $\mathcal{Q}_{t,q}$ is the set of sub-queries for query q at time t. There are a number of ways to estimate $p(q_{i,t}|q)$ as indicated in Ref. [37], but we found that this is the most effective way in our experiments.

In the baseline PM-2, we also directly append each aspect of the initial query that is manually identified in the ground truth at t to the initial ambiguous query q itself as a sub-query. Other settings for PM-2 are the same as in Ref. [13].

Table IV. Mean Performance of SDA and the Baselines on Relevance Metrics. The Best Performance Per Metric Is in Bold. Statistically Significant Differences between SDA and the Best Baseline, PM-2, Are Marked in the Upper Right-Hand Corner of SDA's Scores

	nDCG	ERR	Prec	MAP
LM	.4287	.9624	.3835	.2108
MMR	.4058	.9466	.3663	.1950
MSInc	.4363	.9614	.3907	.2177
MMInc	.4440	.9614	.3962	.2240
xQuAD	.4527	.9660	.4041	.2370
PM-2	.4781	.9798	.4194	.2502
SDA	**.5408**$^{\blacktriangle}$	**.9869**	**.4728**$^{\blacktriangle}$	**.2954**$^{\blacktriangle}$

Table V. Mean of Performance of SDA and the Baselines on Diversification Metrics. The Best Performance per Metric Is in Bold. Statistically Significant Differences Between SDA and the Best Baseline, PM-2, Are Marked in the Upper Right-Hand Corner of SDA's Scores

	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	NRBP
LM	.2560	.7548	.1749	.0604	.1079	.1075
MMR	.2714	.7826	.1816	.0642	.1135	.1114
MSInc	.2760	.7873	.1864	.0675	.1193	.1182
MMInc	.2856	.8009	.1983	.0739	.1283	.1296
xQuAD	.2977	.8300	.2132	.0807	.1402	.1460
PM-2	.3262	.8503	.2272	.0874	.1491	.1587
SDA	**.3783**$^{\blacktriangle}$	**.9214**$^{\blacktriangle}$	**.2610**$^{\vartriangle}$	**.1074**$^{\vartriangle}$	**.1676**$^{\vartriangle}$	**.1886**$^{\vartriangle}$

For SDA and all the baselines, the training/validation/test splits are permuted until all 107 queries have been chosen once for the test set. We repeat the experiments 10 times and report the average evaluation results.

7. RESULTS

We start by comparing the retrieval (**RQ1**) and diversity (**RQ2**) performance of SDA against that of the other methods. We then examine the retrieval (**RQ3**), clustering (**RQ4**), and diversity (**RQ5**) performance of SDA integrated with D2M3 and other topic models and analyze their outcomes per query (**RQ6**) in terms of subtopics retrieved (**RQ7**) and sensitivity to the number of latent topics (**RQ8**).

7.1. The Performance of SDA

RQ1: *Retrieval Performance*. To start, we contrast the retrieval performance of SDA against the baselines in terms of traditional relevance-oriented evaluation metrics. Table IV shows the performance averaged over all 5 test days.

Except for ERR, for every relevance metric, we find the following order between methods: SDA > PM-2 > xQuAD ∼ MMInc ∼ MSInc ∼ LM > MMR. Here > denotes statistically significantly higher performance and A ∼ B denotes that we did not observe a significant differences between A and B. For ERR we observe the following partial order: SDA > PM-2 > xQuAD > MMInc ∼ MSInc ∼ LM > MMR. This relative ordering of methods is mostly consistent across the 5 testing days. In addition, LM outperforms MMR, and the differences are statistically significant. We observe the same relative order of methods (in terms of performance) for each of the 5 individual test dates.

RQ2: *Diversification Performance*. We start by considering the average diversification performance of SDA and our baselines across the 5 testing days. See Table V. SDA outperforms all baselines, on all metrics, and significantly so. LM, not MMR, is the worst-performing method now. The performance of MSInc and MMInc is similar to that of MMR. This is because these methods are quite similar: They work with an objective that tries to return a set of relevant and diversified documents by directly computing the relevance of the documents and their similarities. Although xQuAD and PM-2 are non-streaming diversification methods, they outperform the streaming diversification methods, MSInc and MMInc. The reason is that both xQuAD and PM-2 model the underlying aspects of the queries and try to maintain a diversified and relevant document set, while MSInc and MMInc simply try to make the content of the documents in the returned set differ from each other. SDA statistically significantly outperforms xQuAD and PM-2: It not only tries to maintain a relevant and diversified document set but also updates the probabilities of latent topics to the query, which can be utilized for the online diversification process.

Next, we turn to the diversification performance per day. Rather than presenting five copies of Table V, one per day, we present six heat maps, one per metric, so the relative performance per method and per day can be observed. See Figure 4. The relative order of methods is the same as in Table V. One interesting thing that can be found in these tables is that, as time goes by, in terms of the performance evaluated by some metrics, SDA is more likely to beat the performance of the best baseline, PM-2. For instance, on February 9, 2015, the difference in α-nDCG scores between SDA and PM-2 is only 2.2% (0.4607–0.4387), while the difference on April 30, 2015 is 7.4% (0.3115–0.2374), which is significant at a level of 0.99. The reason is obvious: As time goes by, more aspects are associated with each test topic (see Section 6.2), which provides more room for improvement as evidenced by SDA. In addition, in the heat maps in Figure 4, we find that the performance of the methods diminish over the 5 testing days on all the metrics. The reason is that as time moves forward, on average there are more aspects per query. Recall that on average we have 3.7, 4.4, 5.2, 6.0, and 6.8 aspects per query on the 5 testing dates, respectively. Diversification performance of the representative methods, SDA, PM-2, and xQuAD, and the average number of aspects per query across the 5 testing days are shown in Figure 5.

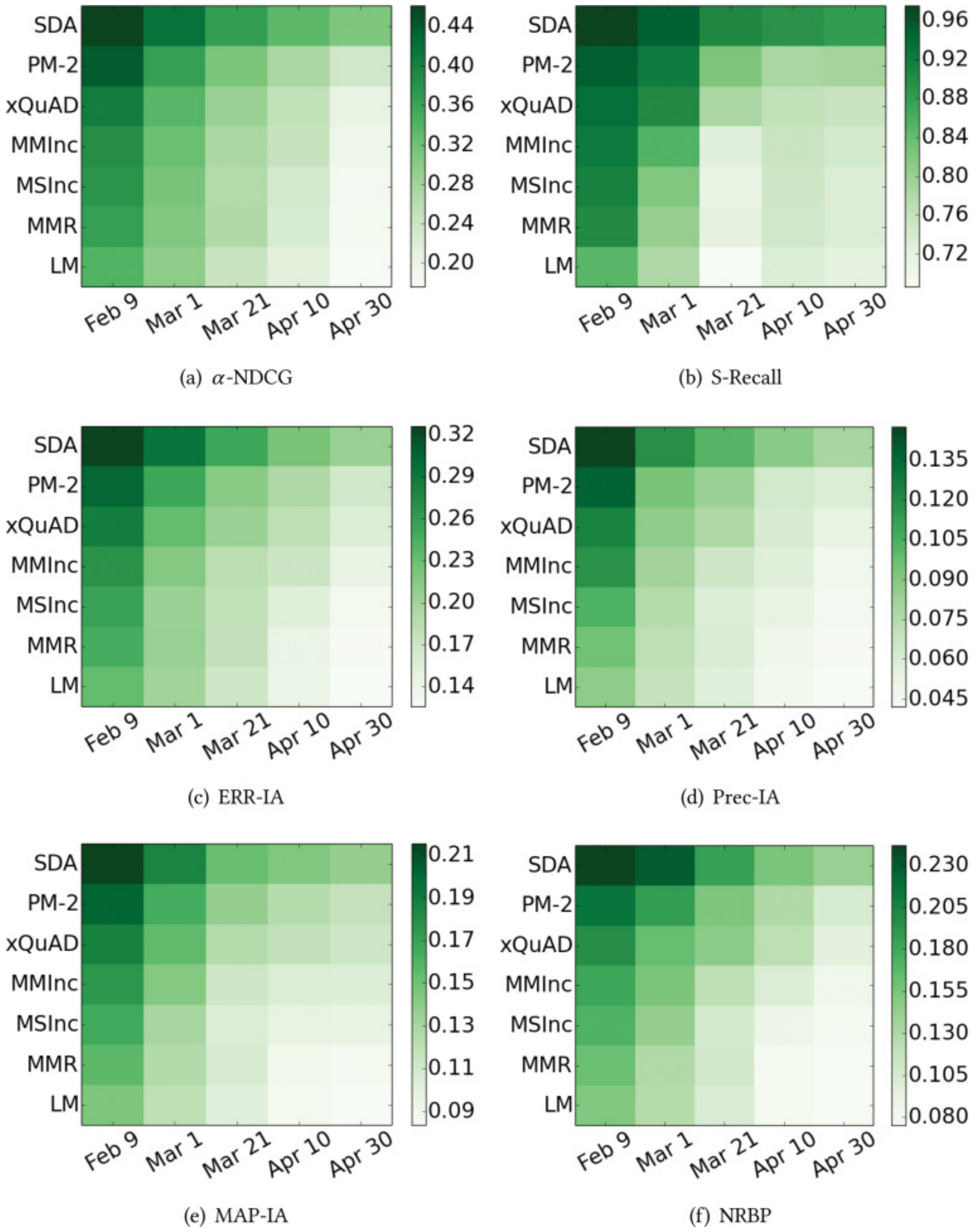
The answers to research questions **RQ1** and **RQ2** are clear. SDA outperforms state-of-the-art streaming diversification algorithms on short text streams, non-streaming ones, and time-sensitive language models on both relevance and diversity-oriented evaluation metrics.

7.2. Contribution of D2M3 to SDA

We compare SDA against variants with a different topic model.

RQ3: *Retrieval Performance*. We report on the retrieval performance, averaged over the 5 test days, of SDA, SDA$_{TTM}$, SDA$_{DMM}$, PM-2$_{GSDMM}$, and PM-2$_{LDA}$ in Table VI. SDA significantly outperforms SDA$_{TTM}$ and SDA$_{DMM}$ that integrate the dynamic topic models TTM and DMM, respectively, and PM-2$_{GSDMM}$ and PM-2$_{LDA}$ that integrate the static topic models GSDMM and LDA, respectively, on all metrics except ERR, where SDA does not significantly differ from SDA$_{TTM}$ and SDA$_{DMM}$ but does significantly differ from PM-2$_{GSDMM}$ and PM-2$_{LDA}$. Thus, D2M3's contribution to the retrieval performance of SDA is bigger than that of the dynamic topic models TTM and DMM and the static topic models GSDMM and LDA.

RQ4: *Clustering Performance*. To compare the clustering performance, given a query, we regard relevant documents associated with the same aspect according to the ground truth as being in the same cluster. We further regard the documents assigned to the same topic $z = \arg\max_z P(z \mid t, d, q)$ (for this purpose only, see Section 6.4) by the underlying topic model as being in the same cluster. The comparison result is shown in Table VII. SDA again significantly outperforms SDA$_{TTM}$, SDA$_{DMM}$, PM-2$_{GSDMM}$, and PM-2$_{LDA}$ on all clustering evaluation metrics, which indicates that the quality of latent



(a) α-NDCG

(b) S-Recall

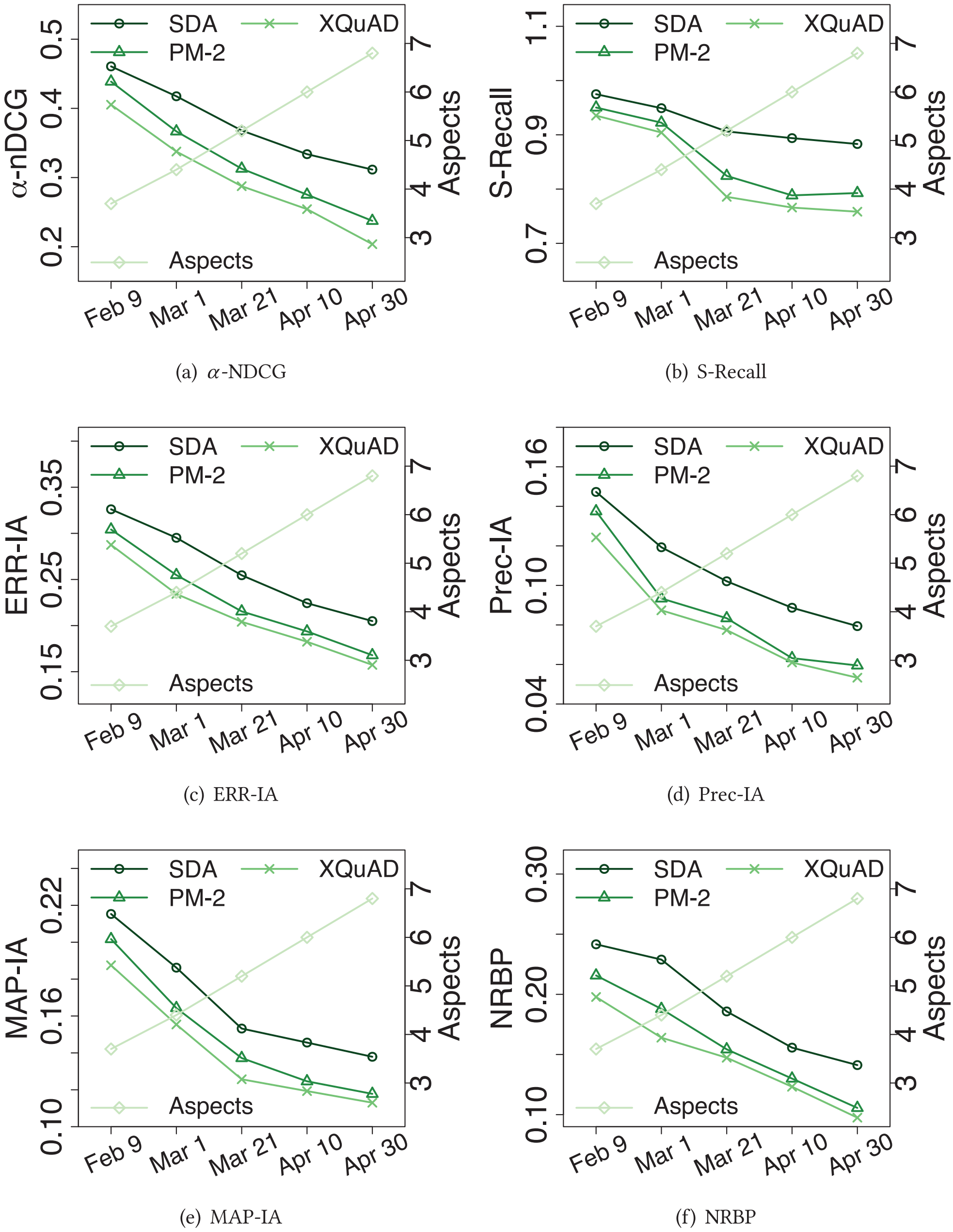
(c) ERR-IA

(d) Prec-IA

(e) MAP-IA

(f) NRBP

Fig. 4. Heat maps of diversification performance. One heat map per metric; columns represent days (February 9, March 1, March 21, April 10, and April 30, 2015, from left to right); rows represent methods (SDA, PM-2, xQuAD, MMInc, MSInc, MMR, LM, from top to bottom).



(a) α-NDCG

(b) S-Recall

(c) ERR-IA

(d) Prec-IA

(e) MAP-IA

(f) NRBP

Fig. 5. Diversification performance of the representative methods, SDA, PM-2, and xQuAD, and the average number of aspects per query across the 5 testing days (February 9, March 1, March 21, April 10, and April 30, 2015, respectively). One plot per metric. In each plot, the Y-axes on the left- and right-hand sides are for diversification performance and the average number of aspects per query, respectively. Figures are best viewed in color.

Table VI. Mean Performance of SDA and Alternative Versions of SDA with D2M3 Replaced by the Dynamic Topic Model TTM or DMM, and PM-2 with Static Topic Model GSDMM or LDA, Using Relevance Metrics. The Best Performance Per Metric Is in Bold. Statistically Significant Differences Between SDA and the Best Performing Alternative System, SDA$_{TTM}$, Are Marked in the Upper Right-Hand Corner of the SDA Scores

	nDCG	ERR	Prec	MAP
PM-2$_{LDA}$	.4482	.9631	.3975	.2293
PM-2$_{GSDMM}$	.4523	.9677	.4052	.2412
SDA$_{DMM}$	.5030	.9856	.4397	.2640
SDA$_{TTM}$	.5198	.9813	.4512	.2770
SDA	**.5408**$^{\blacktriangle}$	**.9869**	**.4728**$^{\blacktriangle}$	**.2954**$^{\blacktriangle}$

Table VII. Mean Performance of SDA and Alternative Versions of SDA with D2M3 Replaced by Dynamic Topic Model TTM or DMM, and PM-2 with Static Topic Model GSDMM or LDA, Using Clustering Metrics. The Best Performance per Metric Is in Bold. Statistically Significant Differences Between SDA and the Best Performing Alternative System, SDA$_{TTM}$, Are Marked in the Upper Right-Hand Corner of the SDA Scores

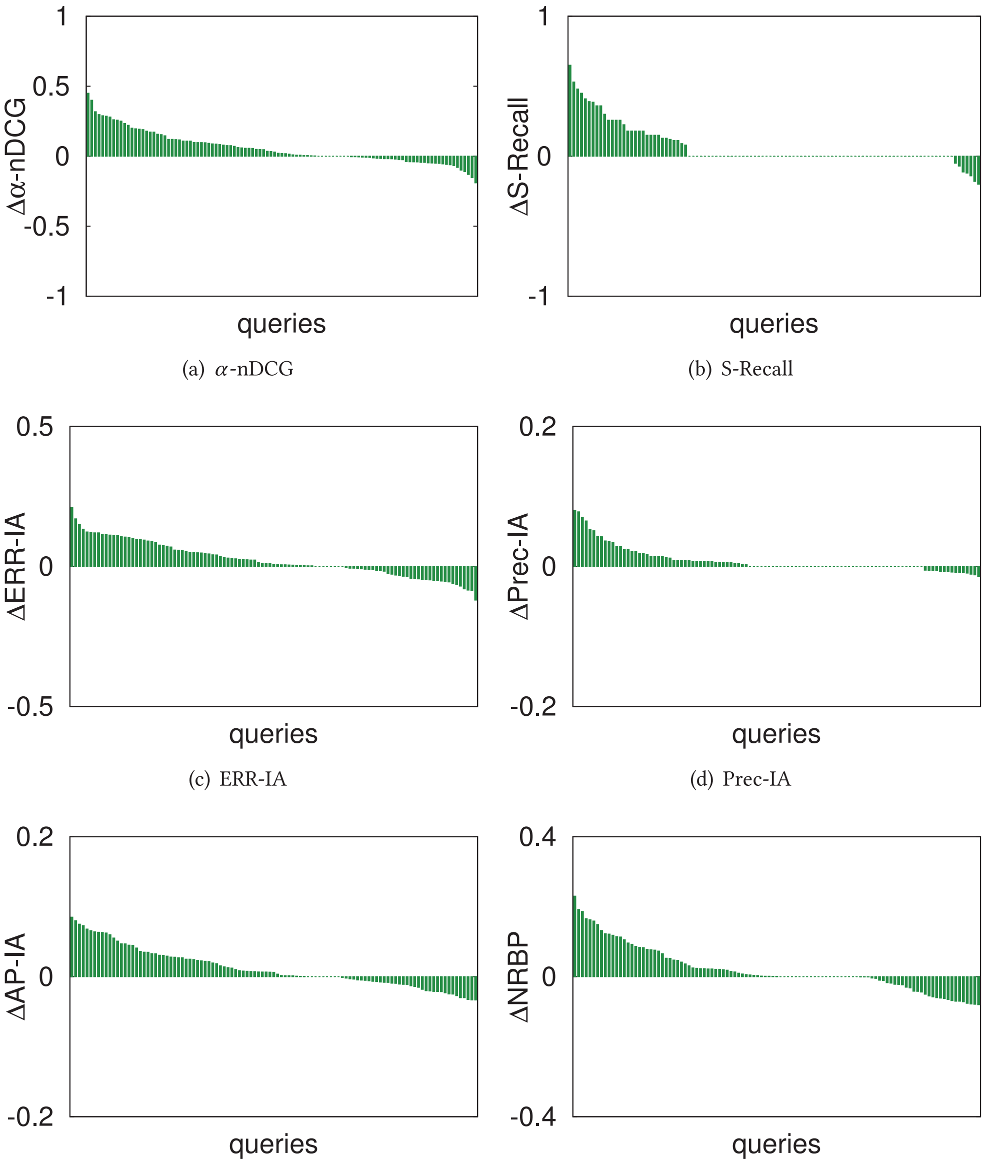
	Purity	NMI	ARI
PM-2$_{LDA}$	.3174	.7024	.6047
PM-2$_{GSDMM}$	.3425	.7234	.6352
SDA$_{DMM}$	.3689	.7616	.6957
SDA$_{TTM}$	.3749	.7828	.7210
SDA	**.3936**$^{\blacktriangle}$	**.8560**	**.7742**$^{\blacktriangle}$

Table VIII. Mean Performance of SDA and Alternative Versions of SDA with D2M3 Replaced by Dynamic Topic Model TTM or DMM, and PM-2 with Static Topic Model GSDMM or LDA, Using Diversification Metrics. The Best Performance per Metric Is in Bold. Statistically Significant Differences Between SDA and the Best-Performing Alternative System, SDA$_{TTM}$, Are Marked in the Upper Right-Hand Corner of the SDA Scores

	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	NRBP
PM-2$_{LDA}$	.2937	.8274	.2031	.0784	.1354	.1433
PM-2$_{GSDMM}$	.3151	.8425	.2174	.0843	.1470	.1537
SDA$_{DMM}$	.3486	.8700	.2402	.0974	.1579	.1724
SDA$_{TTM}$	.3593	.8828	.2500	.0996	.1633	.1818
SDA	**.3783**$^{\blacktriangle}$	**.9214**$^{\blacktriangle}$	**.2610**$^{\vartriangle}$	**.1074**$^{\vartriangle}$	**.1676**$^{\vartriangle}$	**.1886**$^{\vartriangle}$

topics produced by D2M3 for SDA is better than that of the two dynamic topic models TTM and DMM and the two static topic models GSDMM and LDA.

RQ5: *Diversification Performance*. Table VIII lists the diversity scores of SDA, SDA$_{TTM}$, SDA$_{DMM}$, PM-2$_{GSDMM}$, and PM-2$_{LDA}$, averaged over the 5 test days. SDA significantly outperforms PM-2$_{GSDMM}$, which ignores time information of the documents during inference and assigns one single topic to each document, and PM-2$_{LDA}$, which also ignores time information of the documents in the inference and, in contrast, assumes that each document is a mixture of multiple latent topics. SDA also outperforms SDA$_{DMM}$ and SDA$_{TTM}$, in which the dynamic topic models, DMM and TTM, respectively, assume that topics are changed over time, and update the probabilities of topics to the queries, but assume that each document is long enough for inference. We omit per test day results; they show qualitatively similar trends as Table VIII. Also, as visualized in Figure 4 for the baseline approaches, the relative differences between SDA on the one hand and SDA$_{TTM}$ and SDA$_{DMM}$ on the other grow from the first test day (February 9) to the last (April 30), albeit not as dramatically as between SDA and PM-2: from 0.7% to 3%.



(a) α-nDCG

(b) S-Recall

(c) ERR-IA

(d) Prec-IA

(e) AP-IA

(f) NRBP

Fig. 6. Per-query performance differences between SDA and SDA$_{TTM}$, diversity measures, averaged over all test days. One plot per metric. A bar extending above the center of a plot indicates that SDA outperforms SDA$_{TTM}$ and vice versa for bars extending below the center. Figures are not to the same scale.

In the analyses that we provide below, we contrast SDA with the best-performing alternative, SDA$_{TTM}$.

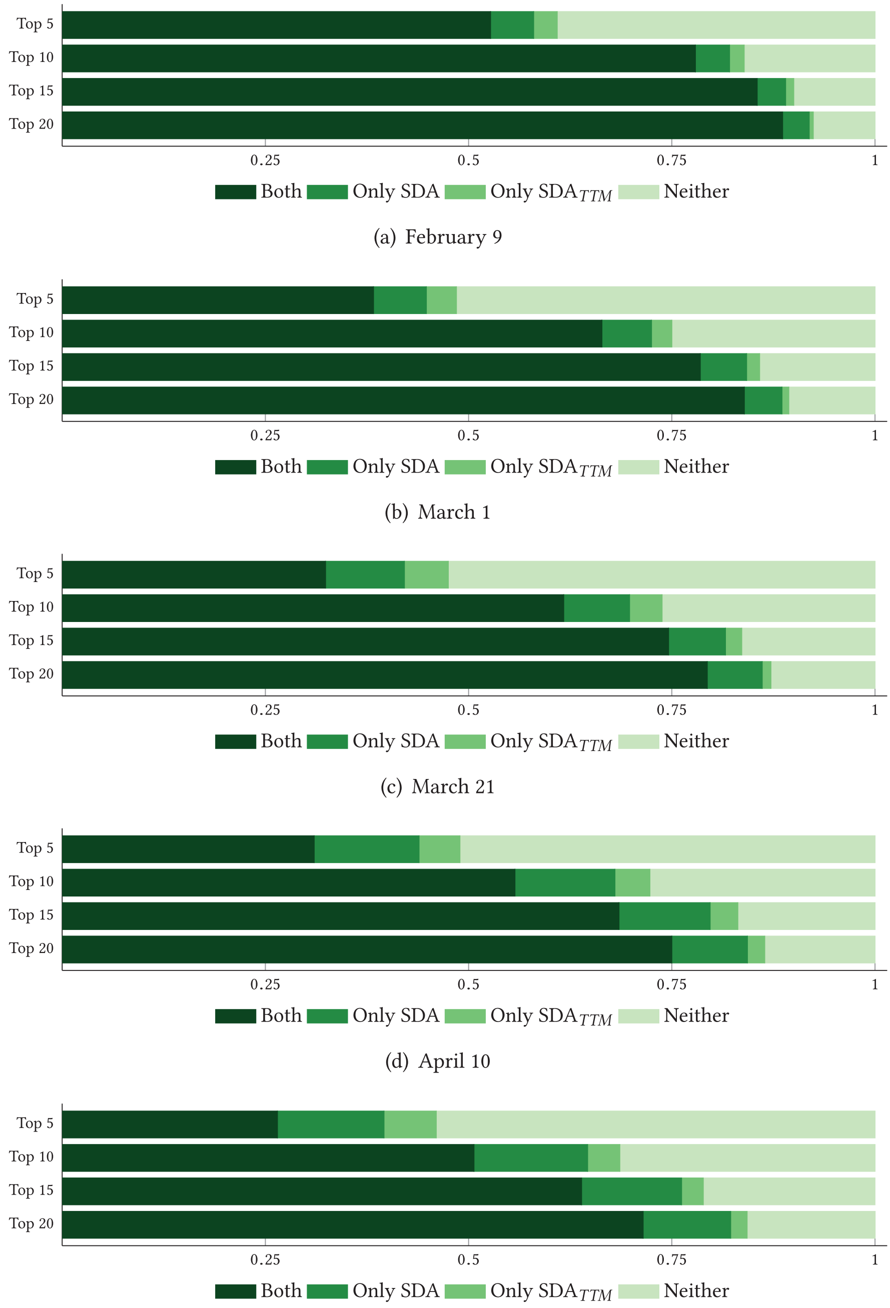
RQ6: *Query-level Analysis*. To begin, we take a closer look at per-test query improvements of SDA vs. SDA$_{TTM}$. Figure 6 shows the per-query performance differences between SDA and SDA$_{TTM}$ in terms of the diversity metrics, averaged over all test days. The number of queries on which SDA outperforms SDA$_{TTM}$ is larger than the number of queries on which SDA$_{TTM}$ outperforms SDA for every metric. This again illustrates that the integration of time information, for example, changes of probabilities of topics to a query captured by D2M3, is able to enhance the diversification performance in short text streams. In a very small number of cases, SDA is outperformed by SDA$_{TTM}$. This appears to be due to the fact that SDA sometimes promotes non-relevant documents when it tries to retrieve as many subtopics as possible for a given query.

RQ7: *Subtopic-level Analysis*. Next, we focus on the fractions of subtopics retrieved by SDA and SDA$_{TTM}$. Figures 7 and 8 show the fractions of subtopics retrieved by both SDA and SDA$_{TTM}$, only SDA, only SDA$_{TTM}$, or neither of the two. Figure 7 shows how runs produced by SDA and SDA$_{TTM}$ differ in terms of subtopic retrieval on all the test days at depth $k = 5$, 10, 15, and 20, respectively. Clearly, on average, as we go deeper down the result lists, the fraction of subtopics retrieved by both methods increases. For example, on April 30, in the top 5 the fraction is 26.5%, while in the top 20 the fraction goes up to 71.5%. However, the fraction of subtopics retrieved by SDA only seems to remain stable: At the top 5 the fraction is 13.1% and at the top 15 the fraction is almost the same, 12.3%; the fraction for SDA$_{TTM}$ only drops down, from 6.4% at the top 5 to 2.6% for the top 15. This shows that, on average, SDA is able to return more subtopics and maintain a stable improvement over SDA$_{TTM}$ as we go down the result lists, while maintaining the relevance.

In Figure 8 we fix another dimension, looking only at the top 20, and contrast different dates, February 9, March 1 and 21, and April 10 and 30, 2015. As time progresses, the fraction of subtopics not retrieved by either SDA or SDA$_{TTM}$ increases. This is because more new subtopics appear as time progresses, which makes the diversification task harder. It is interesting to see that the documents returned by SDA cover more subtopics than those returned by SDA$_{TTM}$, especially during later days. On March 1, SDA covers only 4.6% more subtopics than SDA$_{TTM}$, while on April 30 it covers as many as 9.3% more subtopics. These findings confirm that considering dynamic changes as integrated in D2M3 can improve the performance of diversification in short text streams.

RQ8: *Effect of the Number of Topics*. Finally, we examine the effect on the overall performance of the number of latent topics used in SDA and the baselines SDA$_{TTM}$, SDA$_{DMM}$, and PM-2. We vary the number of latent topics used in SDA and the alternatives just listed, and examine their performnce using diversity metrics. The results are shown in Figure 9, where we take April 30 as representative (findings on other days are qualitatively similar). When only two latent topics are used, the performance of the four methods is almost the same. With 4 to 8 latent topics, the performance of all four increases dramatically. And when the number of latent topics varies between 8 and 16, the performance of both SDA and the baselines seems to level off. A similar pattern was found in many LDA-based topic models (the models integrated into SDA and the baselines here are also LDA-based) in terms of, for example, generalization performance measured by perplexity, where generalization performance becomes better when more latent topics are applied and then levels off when the number of topics applied is large enough [4], and thus it is no surprise to see a similar pattern in terms of diversification performance. This shows the merit of the proposed streaming version of the PM-2 algorithm: It is robust and insensitive to the number of latent topics once this is "large enough."

Importantly, SDA, which integrates D2M3, outperforms SDA$_{TTM}$ and SDA$_{DMM}$, which integrate the tracking topic model and dynamic mixture model, respectively. Latent topics can enhance the performance, and the findings confirm the merit of the proposed dynamic topic model D2M3, that is, it beats the TTM and the DMM when applied in a short text stream for diversification.



(a) February 9

(b) March 1

(c) March 21

(d) April 10

(e) April 30

Fig. 7. Fraction of subtopics retrieved by both SDA and SDA$_{TTM}$, only SDA, only SDA$_{TTM}$, or neither. Results for (a) February 9, (b) March 1, (c) March 21, (d) April 10, and (e) April 30, 2015, averaged over all queries for different top N's, respectively. The figures are best viewed in color.

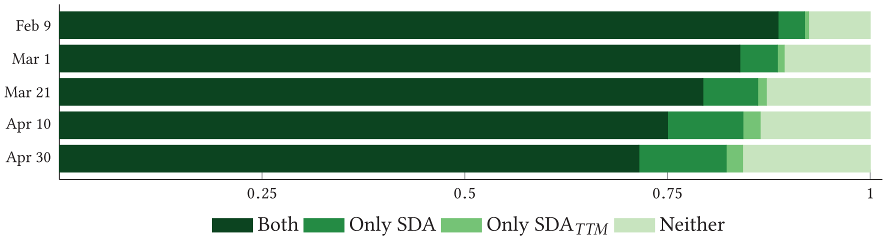


Fig. 8. Fraction of subtopics retrieved by both SDA and SDA$_{TTM}$, only SDA, only SDA$_{TTM}$, or neither. Results for the top 20, averaged over all queries, for different days. The figure is best viewed in color.

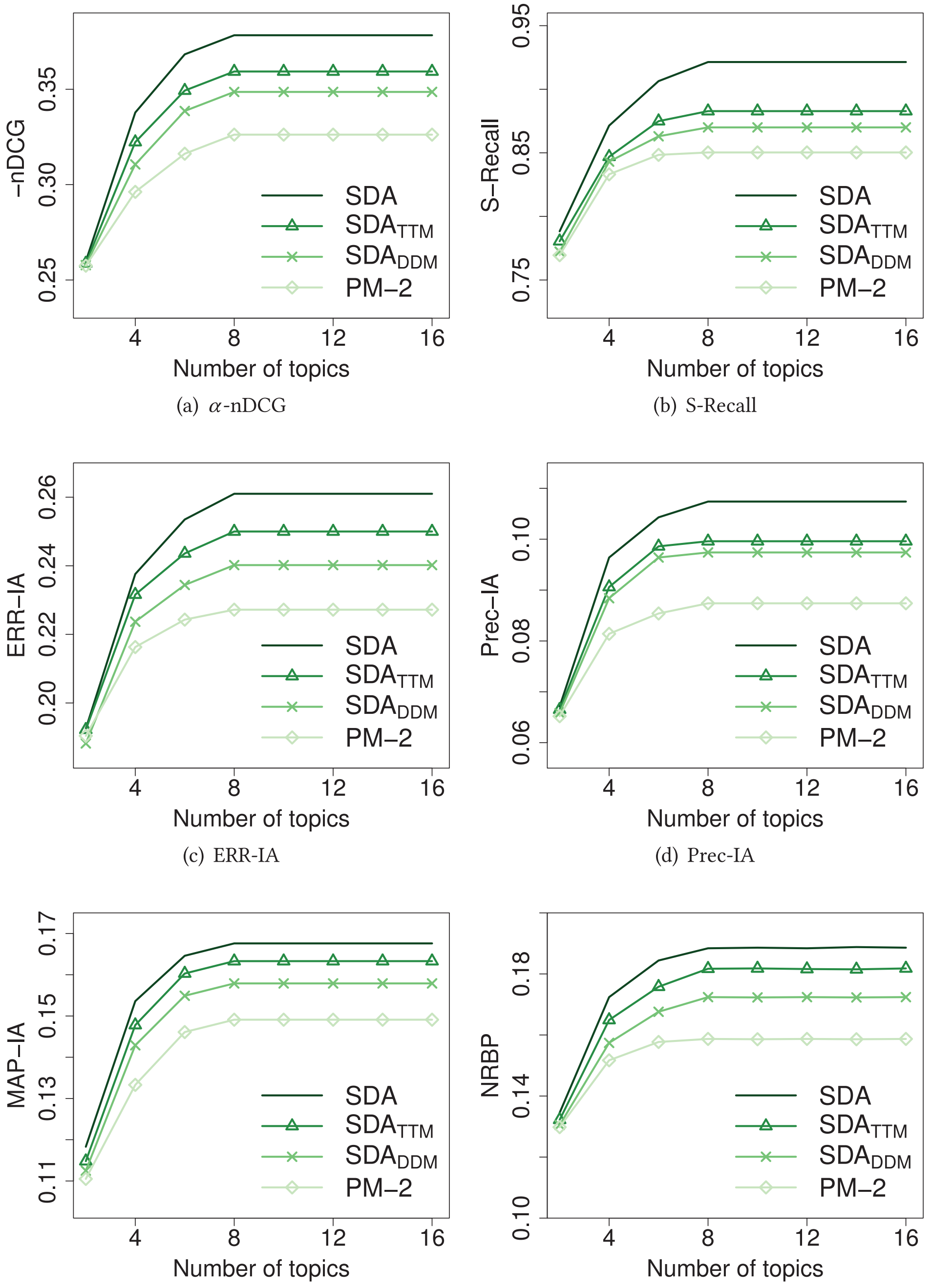
The answers to research questions **RQ2–RQ8** are clear. In terms of both retrieval and diversification performance, our topic model D2M3 as integrated in our SDA works better than any other topic model that we have considered for diversification on short text streams, including state-of-the-art dynamic and non-dynamic topic models.

8. CONCLUSION

We have studied the problem of diversifying search results in short text streams and have proposed a streaming diversification algorithm, SDA, to deal with the problem. Specifically, we propose a dynamic Dirichlet multinomial mixture model, D2M3, to capture the evolution of latent topics in a sequentially organized corpus of short documents and a collapsed Gibbs sampling algorithm to infer the probabilities of topics and documents for a given query. To diversify search results in a stream, we have proposed a modification of the PM-2 diversification algorithm in which the dynamic information of latent topics and the probabilities of documents inferred by D2M3 are integrated while diversifying results.

We have conduced experiments on a Twitter dataset. Our evaluation results have shown that SDA outperforms state-of-the-art non-streaming diversification algorithms, plain streaming diversification methods, as well as variants that integrate other dynamic topic models instead of D2M3. We have found that D2M3 is able to capture the dynamic weights of topics, their probability of relevance to the query, and the probability of documents of being relevant to the query. Moreover, we have found that the proposed modified PM-2 algorithm does aid the performance of diversification in short text streams. Our proposed model works better than the baselines for most queries and is able to return more subtopics. We also found that SDA and the baselines SDA$_{TTM}$, SDA$_{DMM}$, and PM-2 are insensitive to the number of latent topics of a query, once a sufficiently large number was chosen.

As to future work, we aim to automatically estimate the dynamic number of aspects to set the number of latent topics in our dynamic Dirichlet multinomial mixture topic model and let the number of latent topics utilized in modeling documents change from one query to another, as restricting a uniform number of latent topics in our proposed topic model for all the queries may not be the best option. We plan to utilize alternative diversification algorithms instead of the modified PM-2 diversification algorithm in SDA and apply other machine-learning technologies such as deep learning for diversification in short text streams. Also, we intend to apply our SDA to other search applications such as diversifying search results in academic search using article abstracts only but not the full text of the articles. Until now, no streaming long document datasets have been available for dynamic search result diversification; in the future, we plan to collect such a dataset and test whether SDA is also effective for streaming long documents. We also plan to test our model on a larger dataset with short text streams.



(a) α-nDCG

(b) S-Recall

(c) ERR-IA

(d) Prec-IA

(e) MAP-IA

(f) NRBP

Fig. 9. Comparisons among SDA, SDA$_{TTM}$, SDA$_{DMM}$, and PM-2 when varying the number of latent topics, for (a) α-nDCG, (b) S-Recall, (c) ERR-IA, (d) Prec-IA, (e) MAP-IA, and (f) NRBP, respectively, averaged over all test days. Figures are not to the same scale. Figures are best viewed in color.

APPENDIXES

A. GIBBS SAMPLING DERIVATION FOR D2M3

We begin with the joint distribution $P(\mathbf{d}_t, \mathbf{z}_t|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)$. We can take advantage of conjugate priors to simplify the integrals. All other symbols are defined in Sections 3 and 4.

$$\begin{aligned}
&P(\mathbf{d}_t, \mathbf{z}_t|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t) = P(\mathbf{d}_t|\mathbf{z}_t, \mathbf{\Phi}_{t-1}, \beta_t)P(\mathbf{z}_t|\mathbf{\Theta}_{t-1}, \alpha_t)\\
&\quad= \int P(\mathbf{d}_t|\mathbf{z}_t, \mathbf{\Phi}_t)P(\mathbf{\Phi}_t|\mathbf{\Phi}_{t-1}, \beta_t)d\mathbf{\Phi}_t \int P(\mathbf{z}_t|\mathbf{\Theta}_t)P(\mathbf{\Theta}_t|\mathbf{\Theta}_{t-1}, \alpha_t)d\mathbf{\Theta}_t\\
&\quad= \int \prod_{d=1}^{|\mathbf{d}_t|}\prod_{i=1}^{N_d} P(v_{t,di}|\phi_{t,z_{di}}) \prod_{z=1}^{Z} P(\phi_{t,z}|\phi_{t-1,z}, \beta_t)d\Phi_t \times \int \prod_{d=1}^{|\mathbf{d}_t|} P(z_{t,d}|\theta_t)P(\theta_t|\theta_{t-1}, \alpha_t)d\Theta_t\\
&\quad= \int \prod_{z=1}^{Z}\prod_{v=1}^{V} \phi_{t,z,v}^{n_{t,z,v}} \prod_{z=1}^{Z} P(\phi_{t,z}|\phi_{t-1,z}, \beta_t)d\Phi_t \times \int \prod_{d=1}^{|\mathbf{d}_t|} P(z_{t,d}|\theta_t)P(\theta_t|\theta_{t-1}, \alpha_t)d\Theta_t\\
&\quad= \int \prod_{z=1}^{Z}\prod_{v=1}^{V} \phi_{t,z,v}^{n_{t,z,v}} \prod_{z=1}^{Z} \left(\frac{\Gamma\left(\sum_{v=1}^{V}\beta_{t,z,v}\overline{\phi}\right)}{\prod_{v=1}^{V}\Gamma(\beta_{t,z,v}\overline{\phi})} \prod_{v=1}^{V} \phi_{t,z,v}^{\beta_{t,z,v}\overline{\phi}-1}\right) d\Phi_t\\
&\qquad\times \int \prod_{z=1}^{Z} \theta_{t,z}^{m_{t,z}} \left(\frac{\Gamma\left(\sum_{z=1}^{Z}\alpha_{t,z}\theta_{t-1,z}\right)}{\prod_{z=1}^{Z}\Gamma(\alpha_{t,z}\theta_{t-1,z})}\right) \prod_{z=1}^{Z} \theta_{t,z}^{\alpha_{t,z}\theta_{t-1,z}-1} d\Theta_t\\
&\quad= \prod_{z=1}^{Z} \frac{\Gamma\left(\sum_{v=1}^{V}\beta_{t,z,v}\overline{\phi}\right)}{\prod_{v=1}^{V}\Gamma(\beta_{t,z,v}\overline{\phi})} \prod_{z=1}^{Z} \int \prod_{v=1}^{V} \phi_{t,z,v}^{n_{t,z,v}+\beta_{t,z,v}\overline{\phi}-1} d\Phi_t\\
&\qquad\times \frac{\Gamma\left(\sum_{z=1}^{Z}\alpha_{t,z}\theta_{t-1,z}\right)}{\prod_{z=1}^{Z}\Gamma(\alpha_{t,z}\theta_{t-1,z})} \int \prod_{z=1}^{Z} \theta_{t,z}^{m_{t,z}+\alpha_{t,z}\theta_{t-1,z}-1} d\Theta_t\\
&\quad= \prod_{z=1}^{Z} \frac{\Gamma\left(\sum_{v=1}^{V}\beta_{t,z,v}\overline{\phi}\right)}{\prod_{v=1}^{V}\Gamma(\beta_{t,z,v}\overline{\phi})} \prod_{z=1}^{Z} \frac{\prod_{v=1}^{V}\Gamma(n_{t,z,v}+\beta_{t,z,v}\overline{\phi})}{\Gamma\left(\sum_{v=1}^{V}n_{t,z,v}+\beta_{t,z,v}\overline{\phi}\right)}\\
&\qquad\times \frac{\Gamma\left(\sum_{z=1}^{Z}\alpha_{t,z}\theta_{t-1,z}\right)}{\prod_{z=1}^{Z}\Gamma(\alpha_{t,z}\theta_{t-1,z})} \frac{\prod_{z=1}^{Z}\Gamma(m_{t,z}+\alpha_{t,z}\theta_{t-1,z})}{\Gamma\left(\sum_{z=1}^{Z}m_{t,z}+\alpha_{t,z}\theta_{t-1,z}\right)}.
\end{aligned}$$

Applying the chain rule, we can obtain the following conditional probability:

$$\begin{aligned}
&P(z_d = z|\mathbf{z}_{t,-d}, \mathbf{d}_t, \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t) = \tfrac{P(\mathbf{z}_t, \mathbf{d}_t|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)}{P(\mathbf{z}_{t,-d}, \mathbf{d}_t|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)}\\
&\quad\propto \frac{P(\mathbf{z}_t, \mathbf{d}_t|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)}{P(\mathbf{z}_{t,-d}, \mathbf{d}_{t,-d}|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)}\\
&\quad= \prod_{z=1}^{Z} \frac{\prod_{v=1}^{V}\Gamma(n_{t,z,v}+\beta_{t,z,v}\overline{\phi})}{\Gamma\left(\sum_{v=1}^{V}n_{t,z,v}+\beta_{t,z,v}\overline{\phi}\right)} \times \frac{\prod_{z=1}^{Z}\Gamma(m_{t,z}+\alpha_{t,z}\theta_{t-1,z})}{\Gamma\left(\sum_{z=1}^{Z}m_{t,z}+\alpha_{t,z}\theta_{t-1,z}\right)} \Bigg/\\
&\quad\prod_{z=1}^{Z} \frac{\prod_{v=1}^{V}\Gamma(n_{t,z,v,-d}+\beta_{t,z,v}\overline{\phi})}{\Gamma\left(\sum_{v=1}^{V}n_{t,z,v,-d}+\beta_{t,z,v}\overline{\phi}\right)} \times \frac{\prod_{z=1}^{Z}\Gamma(m_{t,z,-d}+\alpha_{t,z}\theta_{t-1,z})}{\Gamma\left(\sum_{z=1}^{Z}m_{t,z,-d}+\alpha_{t,z}\theta_{t-1,z}\right)}.
\end{aligned}$$

Because document d is associated with its own topic z, it becomes

$$
\begin{aligned}
&= \frac{\prod_{v=1}^{V} \Gamma(n_{t,z,v} + \beta_{t,z,v}\overline{\phi})}{\Gamma\big(\sum_{v=1}^{V} n_{t,z,v} + \beta_{t,z,v}\overline{\phi}\big)} \times \frac{\Gamma(m_{t,z} + \alpha_{t,z}\theta_{t-1,z})}{\Gamma\big(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\big)} \Bigg/ \\
&\quad \frac{\prod_{v=1}^{V} \Gamma(n_{t,z,v,-d} + \beta_{t,z,v}\overline{\phi})}{\Gamma\big(\sum_{v=1}^{V} n_{t,z,v,-d} + \beta_{t,z,v}\overline{\phi}\big)} \times \frac{\Gamma(m_{t,z,-d} + \alpha_{t,z}\theta_{t-1,z})}{\Gamma\big(\sum_{z=1}^{Z} m_{t,z,-d} + \alpha_{t,z}\theta_{t-1,z}\big)} \\
&= \frac{\Gamma(m_{t,z} + \alpha_{t,z}\theta_{t-1,z})}{\Gamma(m_{t,z} + \alpha_{t,z}\theta_{t-1,z} - 1)} \frac{\Gamma\big(\sum_{z=1}^{Z}(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - 1\big)}{\Gamma\big(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\big)} \\
&\quad \times \frac{\prod_{v=1}^{V} \Gamma(n_{t,z,v} + \beta_{t,z,v}\overline{\phi})}{\prod_{v=1}^{V} \Gamma(n_{t,z,v,-d} + \beta_{t,z,v}\overline{\phi})} \frac{\Gamma\big(\sum_{v=1}^{V} n_{t,z,v,-d} + \beta_{t,z,v}\overline{\phi}\big)}{\Gamma\big(\sum_{v=1}^{V} n_{t,z,v} + \beta_{t,z,v}\overline{\phi}\big)} \\
&= \frac{\Gamma(m_{t,z} + \alpha_{t,z}\theta_{t-1,z})}{\Gamma(m_{t,z} + \alpha_{t,z}\theta_{t-1,z} - 1)} \frac{\Gamma\big(\sum_{z=1}^{Z}(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - 1\big)}{\Gamma\big(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\big)} \\
&\quad \times \frac{\prod_{v\in d} \Gamma(n_{t,z,v} + \beta_{t,z,v})}{\prod_{v\in d} \Gamma(n_{t,z,v,-d} + \beta_{t,z,v})} \frac{\Gamma\big(n_{t,z,-d} + \sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}\big)}{\Gamma\big(n_{t,z,-d} + N_d + \sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}\big)}.
\end{aligned}
$$

Applying $\Gamma(x) = (x-1)\Gamma(x-1)$ and $\Gamma(x+m) = \prod_{i=1}^{m}(x+i-1)\Gamma(x)$, the above becomes

$$
\begin{aligned}
&= \frac{m_{t,z} + \alpha_{t,z}\theta_{t-1,z} - 1}{\sum_{z=1}^{Z}(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - 1} \frac{\frac{\prod_{v\in d} \Gamma(n_{t,z,v}+\beta_{t,z,v}\overline{\phi})}{\prod_{v\in d} \Gamma(n_{t,z,v,-d}+\beta_{t,z,v}\overline{\phi})}}{\prod_{i=1}^{N_d} \big(n_{t,z,-d} + i - 1 + \sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}\big)} \\
&= \frac{m_{t,z} + \alpha_{t,z}\theta_{t-1,z} - 1}{\sum_{z=1}^{Z}(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - 1} \frac{\prod_{v\in d} \prod_{j=1}^{N_{d,v}}(n_{t,z,v,-d} + \beta_{t,z,v}\overline{\phi} + j - 1)}{\prod_{i=1}^{N_d} \big(n_{t,z,-d} + i - 1 + \sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}\big)}.
\end{aligned}
$$

B. DERIVATION OF THE UPDATE RULES

We apply a fixed-point iteration for estimating the parameters α_t and β_t by maximizing the joint distribution $P(\mathbf{d}_t, \mathbf{z}_t | \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t)$. Instead of maximizing the joint distribution directly, we try to maximize the following:

$$
\begin{aligned}
&\log P(\mathbf{d}_t, \mathbf{z}_t | \mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \alpha_t, \beta_t) \\
&\quad = \sum_{z=1}^{Z} \log \Gamma\left(\sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}\right) - \sum_{z=1}^{Z} \log \Gamma\left(\sum_{v=1}^{V} n_{t,z,v} + \beta_{t,z,v}\overline{\phi}\right) \\
&\quad\quad + \sum_{z=1}^{Z}\sum_{v=1}^{V} \log \Gamma(n_{t,z,v} + \beta_{t,z,v}\overline{\phi}) - \sum_{z=1}^{Z}\sum_{v=1}^{V} \log \Gamma(\beta_{t,z,v}\overline{\phi}) \\
&\quad\quad + \log \Gamma\left(\sum_{z=1}^{Z} \alpha_{t,z}\theta_{t-1,z,v}\right) - \log \Gamma\left(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\right) \\
&\quad\quad + \sum_{z=1}^{Z} \log \Gamma(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - \sum_{z=1}^{Z} \log \Gamma(\alpha_{t,z}\theta_{t-1,z}).
\end{aligned}
$$

Using the bounds [34]: For any $x^* \in \mathbb{R}^+$, $n \in \mathbb{Z}^+$, and x^*'s estimation x,

$$
\log \Gamma(x^*) - \log \Gamma(x^* + n) \geq \log \Gamma(x) - \log \Gamma(x + n) + (\Psi(x+n) - \Psi(x))(x - x^*)
$$

and

$$
\log \Gamma(x^* + n) - \log \Gamma(x^*) \geq \log \Gamma(x + n) - \log \Gamma(x) + x(\log x^* - \log x),
$$

supposing $\alpha^*_{t,z}$ is the optimal parameter in the next fixed-point iteration, it follows that

$$\begin{aligned}\log P(\mathbf{d}_t, \mathbf{z}_t|\mathbf{\Phi}_{t-1}, \mathbf{\Theta}_{t-1}, \{\alpha_{t,1}, \dots \alpha^*_{t,z}, \dots, \alpha_{t,Z}\}, \beta_t) &\geq B(\alpha^*_{t,z}) \\ = \alpha_{t,z}\theta_{t-1,z}(\Psi(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - \Psi(\alpha_{t,z}\theta_{t-1,z})) &\log \alpha^*_{t,z}\theta_{t-1,z} \\ - \alpha^*_{t,z}\theta_{t-1,z}\left(\Psi\left(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\right)\right) &+ C,\end{aligned}$$

where C is function not containing the term $\alpha^*_{t,z}$ and thus will be integrated out by taking $\frac{\partial(\cdot)}{\partial \alpha^*_{t,z}}$ to $\alpha^*_{t,z}$. Then, we let

$$\begin{aligned}\frac{\partial B(\alpha^*_{t,z})}{\partial \alpha^*_{t,z}} &= \frac{\alpha_{t,z}\theta_{t-1,z}(\Psi(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - \Psi(\alpha_{t,z}\theta_{t-1,z}))}{\alpha^*_{t,z}} \\ &\quad - \theta_{t-1,z}\left(\Psi\left(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\right) - \Psi\left(\sum_{z=1}^{Z} \alpha_{t,z}\theta_{t-1,z}\right)\right), \\ &= 0,\end{aligned}$$

which results in

$$\alpha^*_{t,z} = \frac{\alpha_{t,z}(\Psi(m_{t,z} + \alpha_{t,z}\theta_{t-1,z}) - \Psi(\alpha_{t,z}\theta_{t-1,z}))}{\Psi\big(\sum_{z=1}^{Z} m_{t,z} + \alpha_{t,z}\theta_{t-1,z}\big) - \Psi\big(\sum_{z=1}^{Z} \alpha_{t,z}\theta_{t-1,z}\big)},$$

where $\Psi(\cdot)$ is the digamma function defined by $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}$.

Following the same derivation, again supposed $\beta^*_{t,z,v}$ is the optimal parameter in the next fixed-point iteration, we have

$$\beta^*_{t,z,v} = \frac{\beta_{t,z,v}(\Psi(n_{t,z,v} + \beta_{t,z,v}\overline{\phi}) - \Psi(\beta_{t,z,v}\overline{\phi}))}{\Psi\big(\sum_{v=1}^{V} n_{t,z,v} + \beta_{t,z,v}\overline{\phi}\big) - \Psi\big(\sum_{v=1}^{V} \beta_{t,z,v}\overline{\phi}\big)}.$$

ACKNOWLEDGMENTS

We are grateful to our reviewers for providing very detailed and helpful feedback on earlier versions of this article.

REFERENCES

[1] Sofiane Abbar, Sihem Amer-Yahia, Piotr Indyk, and Sepideh Mahabadi. 2013. Real-time recommendation of diverse related articles. In *WWW*. ACM, 1–12.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *WSDM*. ACM, 5–14.

[3] David M. Blei and John D Lafferty. 2006. Dynamic topic models. In *ICML*. 113–120.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[5] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*. ACM, 335–336.

[6] Harr Chen and David R. Karger. 2006. Less is more: Probabilistic models for retrieving fewer relevant documents. In *SIGIR*. ACM, 429–436.

[7] Lisi Chen and Gao Cong. 2015. Diversity-aware top-k publish/subscribe for text stream. In *SIGMOD*. ACM, 347–362.

[8] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *SIGIR*. ACM, 659–666.

[9] Charles L. A. Clarke, Nick Craswell, and Ellen M. Voorhees. 2012. Overview of the TREC 2012 web track. In *TREC*. NIST, 1–8.

[10] Charles L. A. Clarke, Maheedhar Kolla, and Olga Vechtomova. 2009. An effectiveness measure for ambiguous and underspecified queries. In *ECIR*. Springer, 188–199.

[11] Kevyn Collins-Thompson, Craig Macdonald, Paul Bennett, and others. 2015. TREC 2014 web track overview. In *TREC*. NIST, 1–21.

[12] W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2015. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Reading.

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

[14] Van Dang and W. Bruce Croft. 2013. Term level search result diversification. In *SIGIR*. ACM, 603–612.

[15] Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving retrieval of short texts through document expansion. In *SIGIR*. ACM, 911–920.

[16] David Fisher, Ashish Jain, Mostafa Keikha, W. Bruce Croft, and Nedim Lipka. 2015. *Evaluating Ranking Diversity and Summarization in Microblogs Using Hashtags*. Technical Report. University of Massachusetts.

[17] Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proc. Natl. Acad. Sci. U.S.A.* 101, suppl 1 (2004), 5228–5235.

[18] Jiyin He, Edgar Meij, and Maarten de Rijke. 2011. Result diversification based on query-specific cluster ranking. *J. Am. Soc. Inf. Sci. Technol.* 62, 3 (March 2011), 550–571.

[19] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR*. ACM, 50–57.

[20] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *J. Classif.* 1, 2 (1985), 193–218.

[21] Tomoharu Iwata, Shinji Watanabe, Takeshi Yamada, and Naonori Ueda. 2009. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*, Vol. 9. 1427–1432.

[22] Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. 2010. Online multiscale dynamic topic models. In *SIGKDD*. ACM, 663–672.

[23] John D. Lafferty and David M. Blei. 2005. Correlated topic models. In *NIPS*. 147–154.

[24] Wei Li and Andrew McCallum. 2006. Pachinko allocation: DAG-structured mixture models of topic correlations. In *ICML*. 577–584.

[25] Shangsong Liang, Fei Cai, Zhaochun Ren, and Maarten de Rijke. 2016. Efficient structured learning for personalized diversification. *IEEE Trans. Knowl. Data Eng.* 28, 11 (2016), 2958–2973.

[26] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. 2014a. Fusion helps diversification. In *SIGIR*. 303–312.

[27] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. 2014b. Personalized search result diversification via structured learning. In *KDD*. 751–760.

[28] Shangsong Liang, Zhaochun Ren, Emine Yilmaz, and Evangelos Kanoulas. 2017. Collaborative user clustering for short text streams. In *AAAI*.

[29] Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. 2016. Dynamic clustering of streaming short documents. In *KDD*. ACM, 995–1004.

[30] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2015. Overview of the TREC 2014 microblog track. In *TREC'15*. NIST.

[31] Jun S. Liu. 1994. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *J. Amer. Statist. Assoc.* 89, 427 (1994), 958–966.

[32] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[33] Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2011. Incremental diversification for very large sets: A streaming-based approach. In *SIGIR*. ACM, 585–594.

[34] Thomas Minka. 2000. *Estimating a Dirichlet Distribution*. Technical Report. MIT.

[35] Zhaochun Ren, Shangsong Liang, Edgar Meij, and Maarten de Rijke. 2013. Personalized time-aware tweets summarization. In *SIGIR*. ACM, 513–522.

[36] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *UAI*. 487–494.

[37] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *WWW*. ACM, 881–890.

[38] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search result diversification. *Found. Trends Inf. Retriev.* 9, 1 (2015), 1–90.

[39] Alexander Shraer, Maxim Gurevich, Marcus Fontoura, and Vanja Josifovski. 2013. Top-k publish-subscribe for social annotation of news. In *VLDB*. 385–396.

[40] Idan Szpektor, Yoelle Maarek, and Dan Pelleg. 2013. When relevance is not enough: Promoting diversity and freshness in personalized question recommendation. In *WWW*. ACM, 1249–1260.

[41] Saúl Vargas, Pablo Castells, and David Vallet. 2012. Explicit relevance models in intent-oriented information retrieval diversification. In *SIGIR*. ACM, 75–84.

[42] Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-markov continuous-time model of topical trends. In *KDD*. ACM, 424–433.

[43] Xing Wei and W. Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *SIGIR*. ACM, 178–185.

[44] Xing Wei, Jimeng Sun, and Xuerui Wang. 2007. Dynamic mixture models for multiple time-series. In *IJCAI*. 2909–2914.

[45] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *SIGIR*. ACM, 113–122.

[46] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2016. Modeling document novelty with neural tensor network for search result diversification. In *SIGIR*. ACM, 395–404.

[47] Hongzhi Yin, Bin Cui, Ling Chen, and others. 2015. Dynamic user modeling in social media systems. *ACM Trans. Inf. Syst.* 33, 3 (2015), Article 10.

[48] Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *SIGKDD*. ACM, 233–242.

[49] ChengXiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *SIGIR*. ACM, 10–17.

[50] Wayne Xin Zhao, Jing Jiang, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from twitter. In *ACL*. ACL, 379–388.

[51] Yukun Zhao, Shangsong Liang, Zhaochun Ren, Jun Ma, Emine Yilmaz, and Maarten de Rijke. 2016. Explainable user clustering in short text streams. In *SIGIR*. 155–164.

Received May 2016; revised February 2017; accepted February 2017