Dynamic Embeddings for User Profiling in Twitter

Shangsong Liang
King Abdullah University of
Science and Technology, Saudi Arabia
shangsong.liang@kaust.edu.sa

Xiangliang Zhang
King Abdullah University of
Science and Technology, Saudi Arabia
xiangliang.zhang@kaust.edu.sa

Zhaochun Ren
Data Science Lab, JD.com, China
renzhaochun@jd.com

Evangelos Kanoulas
University of Amsterdam, The Netherlands
e.kanoulas@uva.nl

ABSTRACT

In this paper, we study the problem of dynamic user profiling in Twitter. We address the problem by proposing a dynamic user and word embedding model (DUWE), a scalable black-box variational inference algorithm, and a streaming keyword diversification model (SKDM). DUWE dynamically tracks the semantic representations of users and words over time and models their embeddings in the same space so that their similarities can be effectively measured. Our inference algorithm works with a convex objective function that ensures the robustness of the learnt embeddings. SKDM aims at retrieving top-K relevant and diversified keywords to profile users' dynamic interests. Experiments on a Twitter dataset demonstrate that our proposed embedding algorithms outperform state-of-the-art non-dynamic and dynamic embedding and topic models.

CCS CONCEPTS

• **Mathematics of computing → Probabilistic representations**;

KEYWORDS

Word embeddings; Dynamic model; Profiling

ACM Reference Format:
Shangsong Liang, Xiangliang Zhang, Zhaochun Ren, and Evangelos Kanoulas. 2018. Dynamic Embeddings for User Profiling in Twitter. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220043

1 INTRODUCTION

Twitter is one of the most popular microblogging platforms that allow users to describe their current status, recent activities and opinions in short pieces of texts [21]. Understanding how the interests of users evolve over time is of paramount importance to a variety of downstream applications in microblogging platforms, such as user clustering [25], and news recommendations [33]. In this paper, we study the problem of *user profiling* [3] based on their interests as expressed in *streams of short text*. The task of user profiling was first introduced by Balog et al. [3], where language modeling was used to model users and a set of relevant keywords were selected to represent a user's profile. Similar approaches were used by Rybak et al. [36] and Fang and Godavarthy [14]. These approaches demonstrate a number of major drawbacks: (a) they treat words as atomic units leading to a vocabulary mismatch that harms performance, (b) they represent words and users in disjoint vocabulary spaces making it difficult to measure the similarity between users and words when constructing the profile, and (c) they fail to capture the dynamic nature of user profiles along time (with the exception of Fang and Godavarthy [14]).

Unlike previous work on user profiling that concentrating on words, in this paper we target at building user profiles by embedding users and words in a common semantic space. Embeddings [4, 7, 32, 34, 40] have emerged as a powerful method to encode semantic relations between words and hence bridge the vocabulary gap, while they have led to impressive improvements in natural language processing (NLP) tasks [10, 15, 32]. However, learning embeddings in a dynamic context is a non-trivial task, as trade-offs should be made between computational cost and result optimality. Most current approaches naively group data into time bins and learn embeddings separately for each one of these bins [16, 18, 20], which provides a sub-optimal solution given that they lead to a considerable reduction in training data, while decisions such as the size of the time bins are made ad-hocly.

This work extends embedding models in two directions for the temporal profiling of users: (a) it jointly models words and users in a semantic space that allows measuring the similarity between users and words when constructing a user profile, and (b) it directly models the dynamics of language through time in the embedding space. We propose a dynamic user and word embedding model, abbreviated as **DUWE**. Having inferred the embeddings of words and users, we can generate top-K relevant and diversified keywords to profile users' interests over time in streams of text.

Our contributions can be summarized as follows:

(1) We propose a dynamic user and word embedding algorithm that can jointly and dynamically model user and word representations in the same semantic space in the context of streams of documents in Twitter, such that the semantic similarity between users and words can be effectively measured.

(2) We propose a scalable black-box variational inference algorithm to infer the dynamic embeddings of both users and words in streams. Our inference algorithm works with a convex objective function and does not need to split the data into time bins.

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220043

(3) We propose a streaming keyword diversification model to diversify top-K keywords for characterizing users' profiles over time.
(4) We verify the effectiveness of our proposed embedding model, the inference algorithm, and the streaming keyword diversification model, on user profiling in Twitter, and demonstrate that our method significantly outperforms state-of-the-art methods.

2 RELATED WORK

In what follows, we briefly discuss three lines of related work, user profiling, dynamic topic models, and dynamic embeddings.

2.1 User Profiling

User profiling has been gaining attention after the launch of the expert finding task at TREC 2005 enterprise track [11, 23]. Balog et al. [2, 3] proposed a generative language modeling approach to user profiling with the experiments conducted on a static document collection. Recent studies became aware of the importance of temporal user profiling. Temporal user profiling was first introduced in [36], where topical areas were organized in a predefined taxonomy and users' interests was represented as a weighted static tree built directly by the ACM computing classification system. A probabilistic model was proposed in [14], where authors' academic publications were used to learn how personal research interests evolve over time. All of the previous profiling algorithms are built on word frequencies. To the best of our knowledge, there is no user profiling algorithm that jointly models users and words in a semantic space and diversifies the keywords for profiling.

2.2 Dynamic Topic Models

The proposed DUWE model is a dynamic probabilistic model. A number of dynamic probabilistic topic models have appeared in the literature, including the topic over time model [38], the dynamic mixture model [39], and the topic tracking model [17]. All of these models learn the evolution of latent topics over time. More recent dynamic topic models include dynamic User Clustering Topic model (UCT) [25, 42], dynamic topic model for search result diversification [26], collaborative user clustering topic model for streams [28], and Dynamic Clustering Topic model (DCT) [24]. In this work, we take a different approach that pivots on neural embedding models while at the same time we compare our approach to the state-of-the-art dynamic topic models for user profiling [25].

2.3 Dynamic Embeddings

Kulkarni et al. [20] and Mihalcea and Nastase [30] provide a detailed analysis on how word embeddings change over time by comparing with static word embedding models across different time periods. To model changes in word embeddings over time, several approaches have been proposed in the literature [16, 18]. Kim et al. [18] split the data into separate time bins and train a word2vec model [31, 32] within each bin. Word representations obtained over a time bin are used to initialize the ones to be trained over the next time bin. Similarly, Hamilton et al. [16] also split the data into different time bins and train the embeddings over each bin. They assume that word embeddings at nearby time periods approximately differ by a global rotation in addition to a small semantic drift, and approximately compute this rotation. However, it is challenging to distinguish between semantic drifts of words over time and the artifacts of the approximate rotation. None of the two dynamic word embedding algorithms explicitly model the underlying dynamic process. Further, both of them optimize non-convex objective functions, resulting in an unstable representation of words – training word representations twice in the same time bin leads to different embeddings. Bamler and Mandt [4] extend a bayesian skip-gram model to a dynamic version for word embeddings but not for user embeddings, and how to jointly model two different entities in the same space is still unknown. In contrast, we do not split data into time bins while we train the model optimizing a convex objective function. To the best of our knowledge, this is the first attempt to explicitly model the dynamic process of semantic shifts in language and represent words and users (or any other related item) in a joint embedding space over time.

3 TASK FORMULATION

The task we address in this paper is the following: given a set of users and a stream of short text generated by them in Twitter, infer both user and word semantic representations over time, and dynamically identify a set of top-K relevant and diversified keywords to profile each of the users. The output of the algorithm is essentially a function f that satisfies:

$$\mathcal{U}_t, \mathcal{D}_{\leq t} \xrightarrow{f} \mathcal{W}_t,$$

where $\mathcal{U}_t = \{u_i\}_{i=1}^{|\mathcal{U}_t|}$ is a set of users in the stream at time t, with $|\mathcal{U}_t|$ being the number of users, $\mathcal{D}_{\leq t} = \{\mathcal{D}_j\}_{j=0}^{t}$ is the stream of documents generated by the users up to time t, with $\mathcal{D}_t$ being a set of documents generated by all the users at time t, and $\mathcal{W}_t = \{\mathcal{W}_{u_1,t}, \ldots, \mathcal{W}_{u_{|\mathcal{U}_t|},t}\}$ are all users' profiling results at time t with $\mathcal{W}_{u_i,t} = \{w_{u_i,1,t}, \ldots, w_{u_i,K,t}\}$ being the profiling result, i.e., the top-K diversified keywords, for user u_i at time t.

In what follows, we describe our dynamic embedding method DUWE (See §4), based on which, we generate top-K diversified keywords for profiling each user at time t (See §5).

4 DYNAMIC EMBEDDING MODEL

In this section, we detail our proposed dynamic user and word embedding model, i.e., DUWE.

4.1 Preliminaries

The goal of DUWE is to capture the semantic representations of users, $\mathbf{U}_t = \{\mathbf{u}_{i,t}\}_{i=1}^{|\mathcal{U}_t|}$, and words, $\mathbf{V}_t = \{\mathbf{v}_{k,t}\}_{k=1}^{V}$, over time. Here $\mathbf{u}_{i,t}$ and $\mathbf{v}_{k,t}$ represent the user u_i's and the word v_k's embeddings at time t, respectively; V is the size of the vocabulary $\mathcal{V}$.

Our DUWE is a dynamic skip-gram model – a generalization of the well-known static skip-gram model, i.e., word2vec [32]. Given a corpus of documents, word2vec [32] collects evidence of word pairs for which $z_{k,l} = 1$, i.e., words v_k and v_l co-occur within a context window. Here, $z_{k,l} \in \{0, 1\}$ is an indicator variable that denotes a draw for the word pair (v_k, v_l) from the probability distribution $p(z_{k,l} = 1 \mid v_k, v_l) = s(\mathbf{v}_k^\top \mathbf{v}_l)$, where $s(x)$ is defined by a sigmoid function $s(x) = \frac{1}{1+\exp(-x)}$. The word pair (v_k, v_l) with the indicator being $z_{k,l} = 1$ in a specific context window is called a positive example, while the word pair for which $z_{k,l} = 0$

is called a negative example. Let $n^+_{k,l}$ denote the number of times a word pair (v_k, v_l) is observed in documents in a corpus. This is a sufficient statistic of the skip-gram model, and its contribution to the likelihood is obtained as $p(n^+_{k,l} \mid v_k, v_l) = s(\mathbf{v}_k^\top \mathbf{v}_l)^{n^+_{k,l}}$. The skip-gram model also assumes the probability of rejecting a word pair (v_k, v_l) if $z_{k,l} = 0$, and thus it also constructs a fictitious second training set of rejected word pairs, i.e., the negative examples, the number of which are denoted as $n^-_{k,l}$. The likelihood of both positive and negative examples in the whole corpus is obtained as:

$$p(\mathbf{n}^+, \mathbf{n}^- \mid \mathbf{V}) = \prod_{k,l=1}^{V} s(\mathbf{v}_k^\top \mathbf{v}_l)^{n^+_{k,l}} \times s(-\mathbf{v}_k^\top \mathbf{v}_l)^{n^-_{k,l}}, \tag{1}$$

where $\mathbf{n}^+,\ \mathbf{n}^- \in \mathbb{R}^{V \times V}$ are the positive and negative indicator matrices for all word pairs with $n^+_{k,l}$ and $n^-_{k,l}$ being their elements, respectively; $s(-x) = 1 - s(x)$, and $\mathbf{V} = \{\mathbf{v}_k\}_{k=1}^V$ being the embedding results of all the words in the vocabulary. To construct negative examples, the skip-gram model computes $n^-_{k,l}$ as $n^-_{k,l} \propto p(v_k)p(v_l)^{3/4}$, where $p(v_k)$ is the frequency of word v_k, so that $\mathbf{n}^-$ is well-defined up to a proportional constant factor that needs to be tuned experientially. Instead of maximizing the likelihood in (1) directly, the skip-gram model tries to maximize its log likelihood:

$$\log p(\mathbf{n}^\pm \mid \mathbf{V}) = \sum_{k,l=1}^{V} n^+_{k,l} \log s(\mathbf{v}_k^\top \mathbf{v}_l) + n^-_{k,l} \log s(-\mathbf{v}_k^\top \mathbf{v}_l), \tag{2}$$

where we denote $\mathbf{n}^\pm = (\mathbf{n}^+, \mathbf{n}^-)$.

The assumptions made in (1) an (2) are not realistic when it comes to the setting of streams, where the embeddings of words change over time. In addition, the skip-gram model does not model users' embeddings. In the following subsections, we detail the way we model users' and words' embeddings over time (§4.2) and propose an inference algorithm to obtain their dynamic embeddings (§4.3).

4.2 Modeling Embeddings over Time

To model the dynamic user and word embeddings, following the static/dynamic word embedding models [4, 16, 18, 31, 32], we propose DUWE. DUWE builds up on the skip-gram model [32] and extends it by using a Kalman filter [29] as a prior for the time evolution of user and word embeddings. This allows the algorithm to share information, that is user-to-word and word-to-word co-occurring statistics (in a short document, e.g., a tweet in Twitter, the user and the words in the tweet associated with the user, and word themselves in the tweet are assumed to be co-occurring), across all time steps while still allows the embeddings to drift over time.

In our DUWE model, we consider a diffusion process of the embedding vector representations of both users and words over time, and thus we let variances of the transition kernels for all the embeddings of the users be $\boldsymbol{\alpha}^2_{t-1} = \{\alpha^2_{u,t-1}\}_{u=1}^{|\mathcal{U}_t|}$ with $\alpha_{u_i,t-1}$ being the variance of the transition kernel for user u_i's embedding transferring from $t-1$ to t. According to Kalman filtering [29], we can define $\alpha^2_{u,t-1}$ as:

$$\alpha^2_{u,t-1} = \varepsilon \cdot g(\mathcal{D}_{u,t}, \mathcal{D}_{u,t-1})(\tau_t - \tau_{t-1}), \tag{3}$$

where ε is a local diffusion constant for user u's embedding, $\mathcal{D}_{u,t} \in \mathcal{D}_t \equiv \bigcup_{u'} \mathcal{D}_{u',t}$ is a set of documents generated by user u at time t, $g(\mathcal{D}_{u,t}, \mathcal{D}_{u,t-1})$ is a local diffusion value measuring the word distribution changes from previous time step $t-1$ to the current time step t for user u, and $(\tau_t - \tau_{t-1})$ is the time interval between subsequent observations in the stream. Let $\mathbf{D}_{u,t} = [\theta_{v_1,t}, \theta_{v_2,t}, \ldots, \theta_{v_V,t}]$ be a vector representation for u's document set $\mathcal{D}_{u,t}$ at time t, with its element $\theta_{v_k,t}$ being computed by an unsupervised language model with Dirichlet smoothing [12, 41] as:

$$\theta_{v_k,t} = \frac{c(v_k; \mathcal{D}_{u,t}) + \delta \cdot p(v_k \mid \mathcal{D}_{\le t})}{\sum_v c(v; \mathcal{D}_{u,t}) + \delta}, \tag{4}$$

where $c(v; \mathcal{D}_{u,t})$ is the total number of times the word v appearing in the document set $\mathcal{D}_{u,t}$, $p(v \mid \mathcal{D}_{\le t})$ is the probability of the word v appearing in the whole corpus $\mathcal{D}_{\le t}$, and δ is a smoothing parameter that is set to the average length of the documents in the corpus [41]. Then, we define the local diffusion value in (3) as:

$$g(\mathcal{D}_{u,t}, \mathcal{D}_{u,t-1}) = 1 - \exp\left\{-\frac{1}{2}\Big(\mathrm{KL}(\mathbf{D}_{u,t} \| \mathbf{D}_{u,t-1}) + \mathrm{KL}(\mathbf{D}_{u,t-1} \| \mathbf{D}_{u,t})\Big)\right\}, \tag{5}$$

where $\mathrm{KL}(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence. According to (5), if $\mathcal{D}_{u,t} = \mathcal{D}_{u,t-1}$, we will have the variance $\alpha_{u,t-1} = 0$, which indicates that u's embedding at t does not change; and thus unlike other models, DUWE can avoid inappropriate drifts for user embeddings and distinguishes actual drifts from random noise.

Similarly, let the variance of the transition kernels for embeddings of all the words be $\boldsymbol{\beta}^2_{t-1} = \{\beta^2_{t-1}\}_{v=1}^V$ with β_{t-1} being the variance of the transition kernel for any word embedding transferring from $t-1$ to t. Again, according to Kalman filtering [29], we can define β^2_{t-1} as:

$$\beta^2_{t-1} = \eta \cdot h(\mathcal{D}_t, \mathcal{D}_{t-1})(\tau_t - \tau_{t-1}), \tag{6}$$

where η is a local diffusion constant for words' embeddings and $h(\mathcal{D}_t, \mathcal{D}_{t-1})$ is a local diffusion value measuring the words' distribution changes from $t-1$ to t. Let $\mathbf{D}_t = [\phi_{v_1,t}, \phi_{v_2,t}, \ldots, \phi_{v_V,t}]$ be a vector representation for document set $\mathcal{D}_t$, with its element, $\phi_{v_k,t}$, being computed by an unsupervised language model with Dirichlet smoothing [12, 41] as:

$$\phi_{v_k,t} = \frac{c(v_k; \mathcal{D}_t) + \delta \cdot p(v_k \mid \mathcal{D}_{\le t})}{\sum_v c(v; \mathcal{D}_t) + \delta}, \tag{7}$$

where $c(v; \mathcal{D}_t)$ is the total number of times the word v appearing in the document set $\mathcal{D}_t$. Then, with (7) we define and compute the local diffusion value in (6) as:

$$h(\mathcal{D}_t, \mathcal{D}_{t-1}) = 1 - \exp\left\{-\frac{1}{2}\Big(\mathrm{KL}(\mathbf{D}_t \| \mathbf{D}_{t-1}) + \mathrm{KL}(\mathbf{D}_{t-1} \| \mathbf{D}_t)\Big)\right\}. \tag{8}$$

According to (8), if $\mathcal{D}_t = \mathcal{D}_{t-1}$, we will have the variance $\beta_{t-1} = 0$, which indicates that word embeddings at t are modeled to remain unchanged; and thus unlike other models, our DUWE can avoid inappropriate drifts for word embeddings and distinguish actual drifts from random noise.

To model $\widetilde{p}(\mathbf{U}_t \mid \mathbf{U}_{t-1})$, that is the the probability of user embedding (before normalization), $\mathbf{U}_t$, at current time step, t, given the user embedding at the previous time step, $t-1$, $\mathbf{U}_{t-1}$, we add a Gaussian prior with mean $\mathbf{0}$ and variance $\overline{\boldsymbol{\alpha}}^2_0 = \{\overline{\alpha}^2_{u,0}\}_{u=1}^{|\mathcal{U}_t|}$, which prevents the user embedding vectors from growing too large. Thus,

$$\widetilde{p}(\mathbf{U}_t \mid \mathbf{U}_{t-1}) \propto \mathcal{N}(\mathbf{U}_{t-1}, \boldsymbol{\alpha}^2_{t-1}\mathbf{I}) \cdot \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\alpha}}^2_0 \mathbf{I}), \tag{9}$$

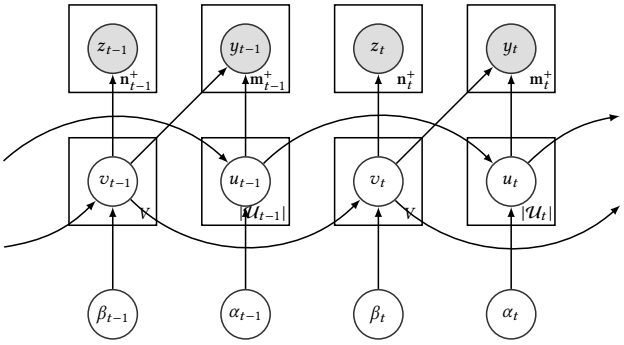

Figure 1: Graphical representation of our dynamic embedding model, DUWE. Shaded nodes represent observed variables.

where $\mathcal{N}(\cdot,\cdot)$ is a Gaussian distribution, and $\mathbf{I}$ is an identity matrix. Similarly, to model $\widetilde{p}(\mathbf{V}_t \mid \mathbf{V}_{t-1})$, that is the probability of word embedding (before normalization) at t, $\mathbf{V}_t$, given the word embedding results at $t-1$, $\mathbf{V}_{t-1}$, we add a Gaussian prior with mean $\mathbf{0}$ and variance $\overline{\boldsymbol{\beta}}_0^2 = \{\overline{\beta}_0^2\}_{v=1}^V$ that prevents the word embedding vectors from growing too large. Thus,

$$\widetilde{p}(\mathbf{V}_t \mid \mathbf{V}_{t-1}) \propto \mathcal{N}(\mathbf{V}_{t-1}, \boldsymbol{\beta}_{t-1}^2\mathbf{I}) \cdot \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\beta}}_0^2\mathbf{I}). \tag{10}$$

We apply normalizations to (9) and (10), and according to the theory that the convolution of two Gaussian distributions is a Gaussian distribution [8], we obtain the corresponding normalized distributions for $\mathbf{U}_t \mid \mathbf{U}_{t-1}$ and $\mathbf{V}_t \mid \mathbf{V}_{t-1}$, respectively:

$$p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) = \mathcal{N}\left(\frac{\mathbf{U}_{t-1}}{1+\boldsymbol{\alpha}_{t-1}^2/\overline{\boldsymbol{\alpha}}_0^2}, \frac{(\boldsymbol{\alpha}_{t-1}^2)^\top \cdot \overline{\boldsymbol{\alpha}}_0^2}{\boldsymbol{\alpha}_{t-1}^2 + \overline{\boldsymbol{\alpha}}_0^2}\mathbf{I}\right), \tag{11}$$

$$p(\mathbf{V}_t \mid \mathbf{V}_{t-1}) = \mathcal{N}\left(\frac{\mathbf{V}_{t-1}}{1+\boldsymbol{\beta}_{t-1}^2/\overline{\boldsymbol{\beta}}_0^2}, \frac{(\boldsymbol{\beta}_{t-1}^2)^\top \cdot \overline{\boldsymbol{\beta}}_0^2}{\boldsymbol{\beta}_{t-1}^2 + \overline{\boldsymbol{\beta}}_0^2}\mathbf{I}\right), \tag{12}$$

The proof that the normalizations of (9) and (10) are (11) and (12), respectively, is omitted here but similar proof is found in [8]. For initialization, i.e., at time $t=0$, we define $p(\mathbf{U}_1 \mid \mathbf{U}_0) \equiv \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\alpha}}_0^2\mathbf{I})$ and $p(\mathbf{V}_1 \mid \mathbf{V}_0) \equiv \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\beta}}_0^2\mathbf{I})$.

According to the graphical representation of DUWE (shown in Fig. 1), the joint distribution of our DUWE can be factorized as:

$$p(\mathbf{m}^\pm_{\le t}\mathbf{n}^\pm_{\le t}, \mathbf{U}_{\le t}, \mathbf{V}_{\le t}) = \prod_{t'=1}^{t}\Big(p(\mathbf{U}_{t'} \mid \mathbf{U}_{t'-1})\, p(\mathbf{V}_{t'} \mid \mathbf{V}_{t'-1}) \times \Big(\prod_{k,l=1}^{V} p(\mathbf{n}^\pm_{k,l,t'} \mid \mathbf{v}_k, \mathbf{v}_l)\Big) \cdot \Big(\prod_{i=1}^{|\mathcal{U}_{t'}|}\prod_{k=1}^{V} p(\mathbf{m}^\pm_{u_i,k,t'} \mid \mathbf{u}_i, \mathbf{v}_k)\Big)\Big), \tag{13}$$

where $\mathbf{n}^\pm_{k,l,t'} = \{\mathbf{n}^+_{k,l,t'}, \mathbf{n}^-_{k,l,t'}\}$, $\mathbf{m}^\pm_{u_i,k,t'} = \{\mathbf{m}^+_{i,k,t'}, \mathbf{m}^-_{i,k,t'}\}$. Here $\mathbf{n}^+_{k,l,t'}, \mathbf{n}^-_{k,l,t'} \in \mathbb{R}^{V\times V}$ are the positive and negative indicator matrices for all word-to-word pairs with $n^+_{k,l,t'}$ and $n^-_{k,l,t'}$ being the number of word-to-word positive (observed word-to-word pairs (v_k, v_l)) and negative (fictitious rejected pairs (v_k, v_l); see §6.3 for their constructions) examples at time t', respectively; and $\mathbf{m}^+_{i,k,t'}$, $\mathbf{m}^-_{i,k,t'} \in \mathbb{R}^{|\mathcal{U}_{t'}|\times V}$ are the positive and negative indicator matrices for all user-to-word pairs with $m^+_{i,k,t'}$ and $m^-_{i,k,t'}$ being the number of user-to-word positive (observed user-to-word pairs (u_i, v_k)) and negative (fictitious rejected user-to-word pairs (u_i, v_k); see §6.3 for constructions) examples.

4.3 Inference

To infer the users' and words' embedding results at time step t, $\mathbf{U}_t$ and $\mathbf{V}_t$, we start by formulating a joint distribution of our DUWE model, i.e., (13), over $\mathbf{m}^\pm_{\le t}$ and $\mathbf{n}^\pm_{\le t}$, and the users' and words' embeddings $\mathbf{U}_{\le t}$ and $\mathbf{V}_{\le t}$ across all the time. Accordingly, we are interested in the posterior distribution over $\mathbf{U}_{\le t}$ and $\mathbf{V}_{\le t}$ conditioned on the statistics information $\mathbf{m}^\pm_{\le t}$ and $\mathbf{n}^\pm_{\le t}$ as follows:

$$p(\mathbf{U}_{\le t}, \mathbf{V}_{\le t} \mid \mathbf{m}^\pm_{\le t}, \mathbf{n}^\pm_{\le t}) = \frac{p(\mathbf{m}^\pm_{\le t}\mathbf{n}^\pm_{\le t}, \mathbf{U}_{\le t}, \mathbf{V}_{\le t})}{\iint p(\mathbf{m}^\pm_{\le t}\mathbf{n}^\pm_{\le t}, \mathbf{U}_{\le t}, \mathbf{V}_{\le t}) d\mathbf{U}_{\le t}\, d\mathbf{V}_{\le t}}. \tag{14}$$

The Evidence Lower BOund. It is intractable to compute the denominator in (14), i.e., the normalization term. Variational inference transforms the problem of approximating a posterior (conditional) distribution into an optimization problem [5, 35]. The idea is to posit a simple family of distributions over the latent variables and find the member of the family that is closest in KL divergence to the posterior distribution. Accordingly, we propose a variational inference algorithm to approximately infer $\mathbf{U}_{\le t}$ and $\mathbf{V}_{\le t}$. Let $\boldsymbol{\lambda}_{\le t}$ be the free parameters of a variational distribution $q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t})$. Here, $\boldsymbol{\lambda}_{\le t} = \{\boldsymbol{\lambda}_{t'}\}_{t'=0}^t$ summarizes all parameters of the variational distribution from time 0 to t. The goal of our inference algorithm is to approximate the posterior, i.e., (14), with the simpler variational distribution $q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t})$ by minimizing the KL divergence to the posterior. Minimizing the KL divergence is equivalent to maximizing the following Evidence Lower BOund (ELBO) [35]:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\lambda}_{\le t}) =& \mathbb{E}_{q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t})}[\log p(\mathbf{m}^\pm_{\le t}, \mathbf{n}^\pm_{\le t}, \mathbf{U}_{\le t}, \mathbf{V}_{\le t})] - \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t})}[\log q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t})].\end{aligned} \tag{15}$$

Black Box Variational Inference. For a restricted class of models, their ELBO can be computed in a closed-form [5]. However, our embedding model is non-conjugate and thus can not be computed in a closed-form. Instead, we propose a variational inference algorithm, which is based on black-box variational inference [35]. Our inference algorithm iteratively updates the variational distribution $q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t})$ given the statistics $\mathbf{m}^\pm_{\le t}$ and $\mathbf{n}^\pm_{\le t}$ from 0 to t. Therefore, we define a variational distribution that is factorized across all time steps up to t, i.e., let $q_{\boldsymbol{\lambda}_{\le t}}(\mathbf{U}_{\le t}, \mathbf{V}_{\le t}) = \prod_{t'=0}^t q_{\boldsymbol{\lambda}_{t'}}(\mathbf{U}_{t'}, \mathbf{V}_{t'})$. We adopt the mean-field approximation inference strategy, and thus factorize the distribution $q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)$ at t as:

$$\begin{aligned} & q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t) = q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t) \cdot q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t) \\ &= \prod_{i=1}^{|\mathcal{U}_t|} \mathcal{N}(\mathbf{u}_{i,t}; \boldsymbol{\mu}_{u_i,t}, \boldsymbol{\sigma}^2_{u_i,t}\mathbf{I}) \cdot \prod_{k=1}^{V} \mathcal{N}(\mathbf{v}_{k,t}; \boldsymbol{\mu}_{v_k,t}, \boldsymbol{\sigma}^2_{v_k,t}\mathbf{I}), \end{aligned} \tag{16}$$

where $\boldsymbol{\mu}_{u_i,t}$ and $\boldsymbol{\mu}_{v_k,t}$ are the means of the user u_i's and the word v_k's embeddings at t, respectively; and $\boldsymbol{\sigma}^2_{u_i,t}$ and $\boldsymbol{\sigma}^2_{v_k,t}$ are the corresponding variances, respectively. At t, given $\boldsymbol{m}^\pm_{\le t}$ and $\boldsymbol{n}^\pm_{\le t}$, and the fact that $\boldsymbol{\lambda}_{\le t-1}$, $\mathbf{U}_{t-1}$ and $\mathbf{V}_{t-1}$ have been obtained, the goal of our inference algorithm is to infer the variational parameters in $q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)$ at t, i.e., $\boldsymbol{\lambda}_t = \left\{\{\boldsymbol{\mu}_{u_i,t}, \boldsymbol{\sigma}^2_{u_i,t}\}_{i=1}^{|\mathcal{U}_t|}, \{\boldsymbol{\mu}_{v_k,t}, \boldsymbol{\sigma}^2_{v_k,t}\}_{k=1}^V\right\}$.

As our model is a Markovian dynamic system (see Fig. 1), we have the following recursion:

$$p(\mathbf{m}^\pm_{\le t}, \mathbf{n}^\pm_{\le t}, \mathbf{U}_{\le t}, \mathbf{V}_{\le t}) = p(\mathbf{U}_{\le t}, \mathbf{V}_{\le t} \mid \mathbf{m}^\pm_{\le t}, \mathbf{n}^\pm_{\le t})\, p(\mathbf{m}^\pm_{\le t}, \mathbf{n}^\pm_{\le t})$$

$$\propto p(\mathbf{U}_{\leq t}, \mathbf{V}_{\leq t} \mid \mathbf{m}^{\pm}_{\leq t}, \mathbf{n}^{\pm}_{\leq t}) = p(\mathbf{U}_{\leq t}, \mathbf{V}_{\leq t} \mid \mathbf{m}^{\pm}_{\leq t})\, p(\mathbf{V}_{\leq t} \mid \mathbf{n}^{\pm}_{\leq t})$$
$$= \prod_{t'=0}^{t} p(\mathbf{U}_{t'}, \mathbf{V}_{t'} \mid \mathbf{m}^{\pm}_{\leq t'}) \cdot \prod_{t'=0}^{t} p(\mathbf{V}_{t'} \mid \mathbf{n}^{\pm}_{\leq t'})$$
$$\propto \prod_{t'=0}^{t} \Big(p(\mathbf{m}^{\pm}_{t'} \mid \mathbf{U}_{t'}, \mathbf{V}_{t'})\, p(\mathbf{U}_{t'}, \mathbf{V}_{t'} \mid \mathbf{m}^{\pm}_{\leq t'-1}) \times p(\mathbf{n}^{\pm}_{t'} \mid \mathbf{V}_{t'})\, p(\mathbf{V}_{t'} \mid \mathbf{n}^{\pm}_{\leq t'-1}) \Big). \quad (17)$$

Substituting (17) into (15), the ELBO in (15) therefore separates into a sum of terms from time 0 to the current time t, i.e., $\mathcal{L}(\boldsymbol{\lambda}_{\leq t}) = \sum_{t'=0}^{t} \mathcal{L}(\boldsymbol{\lambda}_{t'})$ with $\mathcal{L}(\boldsymbol{\lambda}_t)$ for time step t being the following:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\lambda}_t) =& \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)}[\log p(\mathbf{m}^{\pm}_t \mid \mathbf{U}_t, \mathbf{V}_t)] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)}[\log p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{m}^{\pm}_{\leq t-1})] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t)}[\log p(\mathbf{n}^{\pm}_t \mid \mathbf{V}_t)] + \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t)}[\log p(\mathbf{V}_t \mid \mathbf{n}^{\pm}_{\leq t-1})] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)}[\log q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)], \end{aligned} \quad (18)$$

where similar to (2), $\log p(\mathbf{m}^{\pm}_t \mid \mathbf{U}_t, \mathbf{V}_t)$ can be computed as:

$$\log p(\mathbf{m}^{\pm}_t \mid \mathbf{U}_t, \mathbf{V}_t) = \sum_{i=1}^{|\mathcal{U}_t|} \sum_{k=1}^{V} \Big(m^{+}_{i,k,t} \log s(\mathbf{u}_i^{\top} \mathbf{v}_k) + m^{-}_{i,k,t} \log s(-\mathbf{u}_i^{\top} \mathbf{v}_k) \Big), \quad (19)$$

$\log p(\mathbf{n}^{\pm}_t \mid \mathbf{V}_t)$ is computed by (2), and $\log q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)$ can be computed according to (16), respectively; and thus their corresponding expectations can be computed easily. However $p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{m}^{\pm}_{\leq t-1})$ and $p(\mathbf{V}_t \mid \mathbf{n}^{\pm}_{\leq t-1})$ in (18) are still intractable. Applying the variational inference results in the previous time step, $q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1}, \mathbf{V}_{t-1}) \approx p(\mathbf{U}_{t-1}, \mathbf{V}_{t-1})$ and $q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{V}_{t-1}) \approx p(\mathbf{V}_{t-1})$, we can approximate these two probabilities, respectively:

$$\begin{aligned} p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{m}^{\pm}_{\leq t-1}) &\equiv \mathbb{E}_{p(\mathbf{U}_{t-1}, \mathbf{V}_{t-1} \mid \mathbf{m}^{\pm}_{\leq t-1})}[p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{U}_{t-1}, \mathbf{V}_{t-1})] \\ &\approx \mathbb{E}_{q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1}, \mathbf{V}_{t-1})}[p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{U}_{t-1}, \mathbf{V}_{t-1})], \end{aligned} \quad (20)$$

$$\begin{aligned} p(\mathbf{V}_t \mid \mathbf{n}^{\pm}_{\leq t-1}) &\equiv \mathbb{E}_{p(\mathbf{V}_{t-1} \mid \mathbf{n}^{\pm}_{\leq t-1})}[p(\mathbf{V}_t \mid \mathbf{V}_{t-1})] \\ &\approx \mathbb{E}_{q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{V}_{t-1})}[p(\mathbf{V}_t \mid \mathbf{V}_{t-1})], \end{aligned} \quad (21)$$

where $p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{U}_{t-1}, \mathbf{V}_{t-1}) = p(\mathbf{U}_t \mid \mathbf{U}_{t-1})\, p(\mathbf{V}_t \mid \mathbf{V}_{t-1})$, which are obtained by (11) and (12), respectively. Thus, the resulting approximate probability of (20) is a fully factorized distribution:

$$p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{m}^{\pm}_{\leq t-1}) \approx \prod_{i=1}^{|\mathcal{U}_t|} \mathcal{N}(\mathbf{u}_{i,t}; \widetilde{\boldsymbol{\gamma}}_{u_i,t}, \widetilde{\boldsymbol{\psi}}^2_{u_i,t} \mathbf{I}) \times \prod_{k=1}^{V} \mathcal{N}(\mathbf{v}_{k,t}; \widetilde{\boldsymbol{\gamma}}_{v_k,t}, \widetilde{\boldsymbol{\psi}}^2_{v_k,t} \mathbf{I}), \quad (22)$$

where the mean $\widetilde{\boldsymbol{\gamma}}_{u_i,t}$ and the variance $\widetilde{\boldsymbol{\psi}}^2_{u_i,t}$ are:

$$\widetilde{\boldsymbol{\gamma}}_{u_i,t} = \widetilde{\boldsymbol{\psi}}^2_{u_i,t} \left(\boldsymbol{\sigma}^2_{u_i,t-1} + \boldsymbol{\alpha}^2_{u_i,t-1} \mathbf{I} \right)^{-1} \boldsymbol{\mu}_{u_i,t-1}, \quad (23)$$

$$\widetilde{\boldsymbol{\psi}}^2_{u_i,t} = \left[\left(\boldsymbol{\sigma}^2_{u_i,t-1} + \boldsymbol{\alpha}^2_{u_i,t-1} \mathbf{I} \right)^{-1} + (1/\overline{\boldsymbol{\alpha}}^2_0) \mathbf{I} \right]^{-1}. \quad (24)$$

The equations for the word v_k's mean $\widetilde{\boldsymbol{\gamma}}_{v_k,t}$ and variance $\widetilde{\boldsymbol{\psi}}^2_{v_k,t}$ applied in both (20) and (21) are analogous to (23) and (24), respectively. The proof of (23) and (24) is detailed in Appendix A. Similarly, the resulting approximate probability of (21) is:

$$p(\mathbf{V}_t \mid \mathbf{n}^{\pm}_{\leq t-1}) \approx \prod_{k=1}^{V} \mathcal{N}(\mathbf{v}_{k,t}; \widetilde{\boldsymbol{\gamma}}_{v_k,t}, \widetilde{\boldsymbol{\psi}}^2_{v_k,t}). \quad (25)$$

Inserting (22) and (25) into (18) results in the fact that all the expectations in (18) now involve only Gaussians and can be carried-out analytically; and more importantly, our ELBO in (18) becomes a convex objective and thus training the embeddings twice on the same data would result in the same embedding results in contrast to other embedding models with non-convex objectives. We, therefore, can optimize our ELBO at time t via applying the black-box variational inference using the reparameterization trick [19, 35, 37]. We develop an unbiased estimator of the gradient of (18), which can be computed from samples from the variation posterior. To do this, we write the gradient of our ELBO in (18) as expectation with respect to the variational distributions as: $\nabla_{\boldsymbol{\lambda}_t} \mathcal{L}(\boldsymbol{\lambda}_t) =$

$$\begin{aligned} & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)}[\nabla_{\boldsymbol{\lambda}_t(\mathbf{U}_t, \mathbf{V}_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t) \log p(\mathbf{m}^{\pm}_t \mid \mathbf{U}_t, \mathbf{V}_t)] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)}[\nabla_{\boldsymbol{\lambda}_t(\mathbf{U}_t, \mathbf{V}_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t) \log p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{m}^{\pm}_{\leq t-1})] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t)}[\nabla_{\boldsymbol{\lambda}_t(\mathbf{V}_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t) \log p(\mathbf{n}^{\pm}_t \mid \mathbf{V}_t)] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t)}[\nabla_{\boldsymbol{\lambda}_t(\mathbf{V}_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{V}_t) \log p(\mathbf{V}_t \mid \mathbf{n}^{\pm}_{\leq t-1})] + \\ & \mathbb{E}_{q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)}[\nabla_{\boldsymbol{\lambda}_t(\mathbf{U}_t, \mathbf{V}_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t) \log q_{\boldsymbol{\lambda}_t}(\mathbf{U}_t, \mathbf{V}_t)], \end{aligned} \quad (26)$$

The proof that the gradient of (18) is (26) is omitted here but analogous proof can be found in [35]. With (26) we compute noisy unbiased gradients of our ELBO at time t with S Monte Carlo samples from the variational distribution:

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}_t} \mathcal{L}(\boldsymbol{\lambda}_t) \approx \frac{1}{S} \sum_{s=1}^{S} \Big(& \nabla_{\boldsymbol{\lambda}_t(\mathbf{U}^s_t, \mathbf{V}^s_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{U}^s_t, \mathbf{V}^s_t) \Big(\log p(\mathbf{m}^{\pm}_t \mid \mathbf{U}^s_t, \mathbf{V}^s_t) \\ & + \log p(\mathbf{U}^s_t, \mathbf{V}^s_t \mid \mathbf{m}^{\pm}_{\leq t-1}) + \log q_{\boldsymbol{\lambda}_t}(\mathbf{U}^s_t, \mathbf{V}^s_t) \Big) \\ & + \nabla_{\boldsymbol{\lambda}_t(\mathbf{V}^s_t)} \log q_{\boldsymbol{\lambda}_t}(\mathbf{V}^s_t) \Big(\log p(\mathbf{n}^{\pm}_t \mid \mathbf{V}^s_t) + \log p(\mathbf{V}^s_t \mid \mathbf{n}^{\pm}_{\leq t-1}) \Big) \Big), \\ & \text{where} \quad (\mathbf{U}^s_t, \mathbf{V}^s_t) \sim q_{\boldsymbol{\lambda}_t}(\mathbf{U}^s_t, \mathbf{V}^s_t). \end{aligned} \quad (27)$$

With (27), we can use stochastic optimization to optimize our ELBO in (18) and update the parameter $\boldsymbol{\lambda}^{i+1}_t$ at the $i+1$-th iteration with:

$$\boldsymbol{\lambda}^{i+1}_t = \boldsymbol{\lambda}^{i}_t + \rho^{i+1}_t \cdot \nabla_{\boldsymbol{\lambda}^{i}_t} \mathcal{L}(\boldsymbol{\lambda}^{i}_t), \quad (28)$$

where ρ^{i+1}_t is the learning rate at the $i+1$-th iteration. Once the iterations converge, we obtain the optimal embeddings for users and words at time t, $\mathbf{U}^*_t$ and $\mathbf{V}^*_t$ as:

$$(\mathbf{U}^*_t, \mathbf{V}^*_t) = \arg\max_{(\mathbf{U}_t, \mathbf{V}_t)} q_{\boldsymbol{\lambda}^*_t}(\mathbf{U}_t, \mathbf{V}_t) = (\{\boldsymbol{\mu}^*_{u_i,t}\}_{i=1}^{|\mathcal{U}_t|}, \{\boldsymbol{\mu}^*_{v_k,t}\}_{k=1}^{V}), \quad (29)$$

where $\boldsymbol{\lambda}^*_t$ is the optimal parameter at t after the iterations have been converged. In practice, to speed up the convergence, we apply reparameterization trick [37] to (28) during the iterations.

5 STREAMING KEYWORD DIVERSIFICATION MODEL

After user and word embeddings are obtained, inspired by PM-2 [13], a static diversification method, we propose a streaming

Algorithm 1: SKDM to generate top-K keywords for profiling.

Input : Users' and words' dynamic embedding at time t, $\mathbf{U}_t$ and $\mathbf{V}_t$
Output: All users' profiling results at time t, $\mathcal{W}_t$

1 **for** $u = 1, \ldots, |\mathcal{U}_t|$ **do**
2 $\quad \mathcal{W}_{u,t} \leftarrow \varnothing$ /* $\mathcal{W}_{u,t} \in \mathcal{W}_t$ */
3 $\quad \mathcal{N}_u \leftarrow$ retrieve top-N words with their embeddings similar to $\mathbf{u}_t$
4 $\quad C \leftarrow$ perform K-means on $\mathcal{N}_u$
5 $\quad$ **for** $c = 1, \ldots, |C|$ **do**
6 $\quad\quad \pi_{c,u,t} \leftarrow P(c \mid u, t)$
7 $\quad\quad s_{c|u,t} \leftarrow 0$
8 $\quad$ **for** *all positions in the ranked list* $\mathcal{W}_{u,t}$ **do**
9 $\quad\quad$ **for** $c = 1, \ldots, |C|$ **do**
10 $\quad\quad\quad qt[c|u,t] = \frac{\pi_{c,u,t}}{2s_{c|u,t}+1}$
11 $\quad\quad c^* \leftarrow \arg\max_c qt[c|u,t]$
12 $\quad\quad v^* \leftarrow \arg\max_{v \in c^*}(\varkappa P(v|c^*,u,t) + (1-\varkappa)\,\mathrm{tfidf}(v|u,t))$
13 $\quad\quad \mathcal{W}_{u,t} \leftarrow \mathcal{W}_{u,t} \cup \{v^*\}$ /* append v^* to $\mathcal{W}_{u,t}$ */
14 $\quad\quad c^* \leftarrow c^* \backslash \{v^*\}$ /* remove v^* from the cluster c^* */
15 $\quad\quad s_{c^*|u,t} \leftarrow s_{c^*|u,t} + P(v^*|c^*,u,t)$

keyword diversification model [26], SKDM, to generate top-K relevant and diversified keywords for profiling users' interests at time t. The overview of SKDM is shown in Algorithm 1.

For each user u at t, SKDM starts with an empty keyword set $\mathcal{W}_{u,t}$ with K empty seats (step 2 of Algorithm 1), and a set of candidate keywords (step 3), which is the top-N ($N \gg K$) relevant words $\mathcal{N}_u$ whose embeddings have highest cosine similarities to user u' embedding. It then performs K-means on $\mathcal{N}_u$ to obtain clusters of the words (step 4). For each of the seats, it computes the quotient $qt[c|u,t]$ for each cluster c given a user u at time t by: $qt[c|u,t] = \frac{\pi_{c,u,t}}{2s_{c|u,t}+1}$, where $\pi_{c,u,t}$ is the probability of the user u being interested in cluster c at t denoted as $P(c|u,t)$ and is set to be $\pi_{c,u,t} = P(c|u,t) = \frac{1}{Z_C}\cos(\mathbf{u}_t, \mathbf{v}_c)$ (step 6), and $s_{c|u,t}$ is the "number" of seats occupied by cluster c (in initialization, $s_{c|u,t}$ is set to 0 for all clusters (step 7)). Here $\mathbf{v}_c$ is the average embedding of all words in cluster c, Z_C is a normalization term. According to PM-2, seats should be awarded to the cluster with the largest quotient in order to best maintain the proportionality of the result list. Therefore, SKDM assigns the current seat to the cluster c^* with the largest quotient (step 11). The keyword to fill this seat is the one from c^* and its embedding has the highest cosine similarity to $\mathbf{u}_t$, defined as $P(v|c^*,u,t) = \frac{1}{Z_{c^*}}\cos(\mathbf{u}_t, \mathbf{v}_v)$, where Z_{c^*} is a normalization term. Thus we propose to obtain the keyword v^* for user u's profiling at t as (step 12): $v^* \leftarrow \arg\max_{v \in c^*}(\varkappa P(v|c^*,u,t) + (1-\varkappa)\,\mathrm{tfidf}(v|u,t))$, where $0 \le \varkappa \le 1$ is a trade-off free parameter, and $\mathrm{tfidf}(v|t,u)$ is a time-sensitive term frequency-inverse document frequency function for user u at t, which can be defined as: $\mathrm{tfidf}(v|t,u) = \mathrm{tf}(v|\mathcal{D}_{u,t}) \times \mathrm{idf}(v|u,\mathcal{D}_t)$, where $\mathrm{tf}(v|\mathcal{D}_{u,t}) = \frac{|\{d \in \mathcal{D}_{u,t} : v \in d\}|}{|\mathcal{D}_{u,t}|}$ is the term frequency function that computes how many percents of the documents that contain the word v in the whole document set $\mathcal{D}_{u,t}$, and $\mathrm{idf}(v|u,\mathcal{D}_t) = \log \frac{|\mathcal{D}_t|}{|\{d \in \mathcal{D}_t : v \in d\}|+\epsilon}$ is the inverse document frequency function with ϵ being set to 1 to avoid the division-by-zero error. According to the tfidf function, if the word v frequently appears in $\mathcal{D}_{u,t}$ generated by user u but not frequently appears in the document set $\mathcal{D}_t$ generated by all the users in $\mathcal{U}_t$, $\mathrm{tfidf}(v|t,u)$ will return a high score. After the word v^* is selected, SKDM adds v^* as a result keyword to $\mathcal{W}_{u,t}$ for profiling the user u at t, i.e., $\mathcal{W}_{u,t} \leftarrow \mathcal{W}_{u,t} \cup \{v^*\}$ (step 13), removes it from the cluster c^* (step 14), and increases the "number" of seats occupied by the cluster c^* as (step 15): $s_{c^*|u,t} \leftarrow s_{c^*|u,t} + P(v^*|c^*,u,t)$. The process (steps 8 to 15) repeats until we get K diversified keywords for $\mathcal{W}_{u,t}$. The order in which a keyword is appended to $\mathcal{W}_{u,t}$ determines its ranking for the profiling. After the process is done, we obtain a set of diversified keywords $\mathcal{W}_{u,t}$ that profile the interest of a user at t.

Obviously, the user and word dynamic embeddings can be computed offline and the top-K keywords obtained by SKDM can be performed offline as well. Thus, our profiling algorithm is efficient.

6 EXPERIMENTAL SETUP

6.1 Research Questions

We seek to answer the following research questions that guide the remainder of the paper:
(RQ1) Can DUWE capture better semantic representations of users and words for user profiling, compared to state-of-the-art non-dynamic and dynamic embedding and topic models?
(RQ2) How the length of time bins affects the DUWE model?
(RQ3) How good the representations inferred by DUWE are?
(RQ4) Can DUWE capture the dynamics of both user and word embeddings and make the embedding results explainable?
(RQ5) Is DUWE sensitive to the embedding dimensions?

6.2 Dataset

In order to answer our research questions, we work with a publicly available dataset collected from Twitter [25].[1] In details, the dataset randomly sampled 1,375 users from Twitter, and all users' tweets posted from the beginning of their registrations up to May 31, 2015. Totally, it has 3.78 million tweets with each tweet having its own timestamp. The average length of the tweets is 12 words.

We obtain two types of **G**round **T**ruth: one for evaluating **R**elevance-oriented (RGT) performance and another for evaluating **D**iversity-oriented (DGT) performance. To create the RGT, we split the dataset into 5 different partitions of time periods, i.e., a week, a month, a quarter, half a year and a year, respectively. For each Twitter user at every specific time period, an annotator was asked to generate a ranked list of top-K relevant keywords (the number of which was decided by the annotators) that can summarize the user's interests at that time period. In total, 68 annotators took part in the labelling with each of them labelling about 5 Twitter users for these 5 different partitions. To create DGT, as it is expensive to manually obtain aspects of the keywords from annotators, we cluster the relevant keywords with their embeddings [2] into 15 categories [3] by K-means. Relevant keywords within a cluster are regarded as being relevant to the same aspect in the DGT ground truth.

6.3 Baselines and Settings

We make comparisons between the proposed DUWE model and the following state-of-the-art algorithms for user profiling:

[1] Available from https://bitbucket.org/sliang1/uct-dataset/get/UCT-Dataset.zip.
[2] Publicly available from https://nlp.stanford.edu/projects/glove/.
[3] Information of the categories is available from http://dmoztools.net.

Non-dynamic embedding models:

Skip-Gram Model (SGM): This is the popular static word2vec embedding model [31, 32].

Distributed Representations of Documents (DRD): This is the popular static doc2vec embedding model [22].

Dynamic traditional profiling model:

Predictive Language Model (PLM). It models the dynamics of personal interests via a probabilistic language model [14].

Dynamic topic model:

User Clustering Topic model (UCT). This is a dynamic multinomial Dirichlet mixture user clustering topic model [25], which can capture users' time-varying topic distributions.

Dynamic embedding models:

Dynamic Independent Skip-Gram model (DISG). It splits the data into different time bins, independently initializes words' representations and obtains the words' embeddings at each bin by word2vec [31, 32]. Word embeddings at nearby bins are then made comparable by approximating orthogonal transformations [16].

Dynamic Pre-initialized Skip-Gram model (DPSG). This approach [18] is the same as DISG, but with word vectors being initialized with values from the previous time bin.

Dynamic Independent Distributed Representations of documents (DIDR). This approach is the same as DISG, but obtains embeddings of documents rather than words at each bin by doc2vec [22].

Dynamic Pre-initialized Distributed Representations of documents (DPDR). This approach is the same as DIDR, but with document vectors being initialized with the average values of the words in the documents from the previous bin.

SGM and DRD are static methods, while the others are dynamic ones. We do not include other dynamic topic models as baselines, e.g., topic tracking model [17], since Liang et al. [25] have demonstrated that UCT outperforms these topic models. For all the word embedding baseline models, the average of the embeddings has been used to represent users.

Following previous work on embeddings [16, 18, 31, 32], we set the number of dimensions both in DUWE and in the baseline methods, to 300. For fair comparisons, we set the number of topics in the baseline topic models to 300 as well (we found that the performance is almost consistent once the dimensions of the representations in embeddings and the number of topics in topic models are as large as ∼100). Following word2vec [31, 32], for both DUWE and all the embedding baselines, we set the number of negative word pairs (v_k, v_j) samples at time t to $n^-_{k,l,t} = \left(\sum_{k',l'=1}^{V} n^+_{k',l',t}\right) \cdot \xi \cdot p_t(k) \cdot p'_t(l)$, where ξ is the ratio of negative to positive word pairs and is set to 1.0, $p_t(k) = \frac{\sum_{l=1}^{V} n^+_{k,l,t}}{\sum_{k',l=1}^{V} n^+_{k',l,t}}$, and $p'_t(l) = \frac{(p_t(l))^{3/4}}{\sum_{l'=1}^{V}(p_t(l'))^{3/4}}$. We follow the same way to define the number of negative user-word pair (u_i, v_k) samples in $\mathbf{m}^-$ in our DUWE. For tuning the parameters in DUWE and all the baselines, we use a 70%/20%/10% split of the users in the dataset for our training, validation and test sets, respectively. We train DUWE for different values of the parameters ε in (3), η in (6), $\overline{\boldsymbol{\alpha}}_0^2$ in (9) and $\overline{\boldsymbol{\beta}}_0^2$ in (10); ε, η, and all elements in $\overline{\boldsymbol{\alpha}}_0^2$ and $\overline{\boldsymbol{\beta}}_0^2$ take values in $[0.001, 0.01, 0.1, 0.2, \ldots, 1, 2, \ldots, 10]$, respectively (elements in $\overline{\boldsymbol{\alpha}}_0^2$ are set to be equal at each training time; the same setting to those in $\overline{\boldsymbol{\beta}}_0^2$). The optimal ε, η, $\overline{\boldsymbol{\alpha}}_0^2$ and $\overline{\boldsymbol{\beta}}_0^2$ values are decided based on the validation set, and evaluated on the test set. The train/validation/test splits are permuted until all users were chosen once for the test set. We repeat the experiments 10 times and report the average results. For initialization in DUWE and other embedding baselines, we let words' embeddings at $t = 0$ be those pre-trained by word2vec and users' embeddings be the average of the words embeddings associated with the users. We adopt Adadelta for setting our learning rates ρ_t^{i+1} in (28).

6.4 Evaluation Metrics

For evaluation purpose, we use standard relevance-oriented evaluation metrics, Pre@k (Precision at k), NDCG@k (Normalized Discounted Cumulative Gain at k), MRR@k (Mean Reciprocal Rank at k), and MAP@k (Mean Average precision at k) [12], and diversity-oriented metrics, Pre-IA@k (Intent-Aware Precision at k) [1], α-NDCG@k [9], MRR-IA@k [1], MAP-IA@k [1]. We also propose semantic versions of the original metrics, denoted as Pre-S@k, NDCG-S@k, MRR-S@k, MAP-S@k, Pre-IA-S@k, α-NDCG-S@k, MRR-IA-S@k, and MAP-IA-S@k, respectively. Here the only difference between the original metrics and the corresponding semantic ones is the way to compute the relevance score of a retrieved keyword v^* to ground truth keyword v_{gt}. For original metrics, we let the relevance score be 1 if and only if $v^* = v_{gt}$, otherwise be 0; whereas for the semantic versions, we let the relevance score be the cosine similarity between the word embedding vectors of v^* and v_{gt}, computed as $\cos(\mathbf{v}^*, \mathbf{v}_{gt})$. Since we are usually restricted to choose a small number of keywords to describe a user's profile, we compute the scores at depth 10, i.e., let $k = 10$ in evaluation. For simplifying the notation we use M to refer to $M@k$, where M is any metrics. Additionally, we adopt Perplexity [6] to evaluate the generalization performance of the models.

7 RESULTS AND DISCUSSIONS

In this section, we answer the research questions listed in §6.1, analyze the experimental results, and provide discussions.

7.1 Overall Profiling Performance

RQ1: We compare the profiling performance of our DUWE with that of the baselines listed in §6.3.

Tables 1 and 2 compare the relevance and diversity profiling performance of DUWE to that of the baselines, averaged across all the testing time periods on every month, and evaluated by the relevance and diversity ground truths, RGT and DGT, respectively. The ranking of models with respect to the relevance and diversity performance is consistent across different evaluation metrics, and in particular this order is observed: DUWE > DPDR ∼ DPSG > UCT > DIDR ∼ DISG ∼ PLM ∼ DRD ∼ SGM. Here > denotes the performance difference is statistically significant at a significance level of 95% with Student's two tailed t-test, and ∼ denotes the difference is not statistically significant. The finding DUWE > DPDR ∼ DPSG indicates that embeddings of both users and words produced by our DUWE work better than those by state-of-the-art dynamic embedding models. The finding that DUWE, DPDR, DPSG and UCT outperform DIDR and DISG indicates that the strategies

Table 1: *Relevance* Performance on time periods of each month. Statistically significant differences between DUWE and the best baseline DPDR are marked in the upper right hand corner of DUWE scores. Statistical significance is tested using a two-tailed paired t-test and is denoted using ▲ for $\alpha = .01$, and $^{\triangle}$ for $\alpha = .05$.

	Pre	NDCG	MRR	MAP	Pre-S	NDCG-S	MRR-S	MAP-S
SGM	.264	.232	.662	.135	.410	.388	.863	.208
DRD	.268	.234	.662	.137	.412	.392	.864	.210
PLM	.273	.239	.668	.140	.417	.398	.870	.212
DISG	.295	.279	.721	.150	.426	.417	.873	.220
DIDR	.308	.287	.725	.153	.438	.425	.876	.225
UCT	.335	.324	.787	.172	.462	.457	.883	.237
DPSG	.352	.338	.792	.178	.470	.459	.897	.242
DPDR	.355	.342	.812	.184	.476	.463	.905	.247
DUWE	.383▲	.375▲	.854▲	.207▲	.495▲	.483▲	.932▲	.262▲

Table 2: *Diversification* Performance on time periods of every month. Notations for the statistical significances are as in Table 1.

	Pre -IA	α-ND CG	MRR -IA	MAP -IA	Pre -IA-S	α-ND CG-S	MRR -IA-S	MAP -IA-S
SGM	.158	.188	.482	.183	.260	.325	.734	.150
DRD	.159	.189	.485	.183	.262	.328	.738	.150
PLM	.162	.192	.487	.187	.265	.332	.742	.152
DISG	.183	.217	.504	.198	.294	.352	.760	.172
DIDR	.187	.228	.517	.205	.306	.364	.765	.178
UCT	.209	.245	.543	.224	.328	.395	.778	.194
DPSG	.223	.256	.573	.234	.337	.412	.783	.207
DPDR	.225	.260	.580	.242	.348	.426	.787	.213
DUWE	.257▲	.293▲	.617▲	.258▲	.387▲	.447▲	.808$^{\triangle}$	.227▲

of tracking embeddings over time upon the embeddings at previous time steps work better than those of simply splitting the data into separate time bins and then obtaining embeddings from each of the bin as dynamic embedding results. All dynamic models work better than static models, i.e., DRD and SGM, which illustrates that embeddings need to be modeled over time for user profiling.

7.2 Length of Time Bins

RQ2: We vary the length of time bins to analyze if the models are sensitive to the length and the performance is consistent over time.

Fig. 2 reports the relevance and diversity performance using Precision, NDCG, Pre-IA and α-NDCG as representative metrics. As it can be seen, DUWE significantly and consistently outperforms the best three baselines, DPDR, DPSG and UCT, for the different lengths of testing time cutoffs which vary from a week to a year, which illustrates that dynamic embeddings generated by our DUWE work better than the state-of-the-art. When the length of time periods increases from a quarter to a year, DUWE reaches a plato but still outperforms the best baselines. All of these findings demonstrate that DUWE is robust and it is able to maintain significant improvements of user profiling performance over the state-of-the-art.

7.3 Quality of Semantic Representations

RQ3: We now evaluate the performance of DUWE and the baseline models in terms of perplexity, which is widely used as an evaluation metric to evaluate the generation of representations [6].

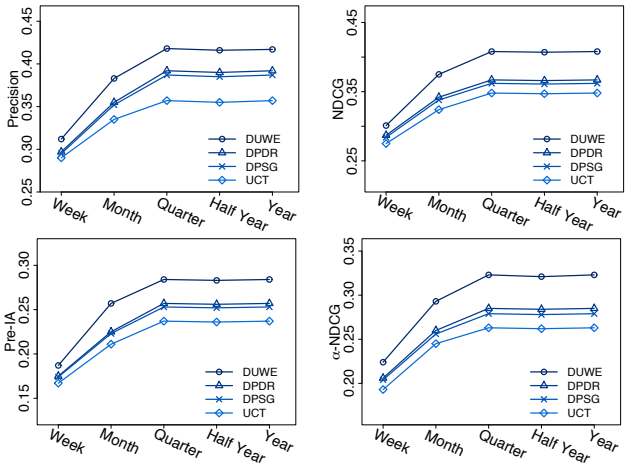

Figure 2: Relevance and diversity performance of DUWE, DPDR, DPSG and UCT on time periods of a week, a month, a quarter, half a year, and a year, respectively.

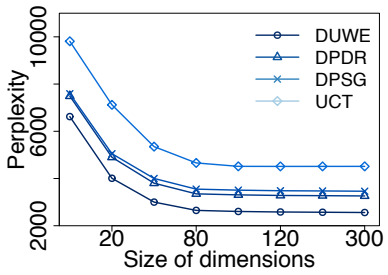

Figure 3: Mean perplexity of DUWE and state-of-the-art models with varying sizes of dimensions.

Perplexity is monotonically decreasing with the likelihood of the documents, and is algebraically equivalent to the inverse of the geometric mean per-word (in our case per-user and per-word) likelihood. To evaluate the quality of representations, we follow that in [6, 24] and compute the perplexity [6, 24] as $\text{Perplexity}(\mathcal{D}_{\leq t}) = \exp\Big(-\frac{\sum_{t'=0}^{t}\sum_{d=1}^{|\mathcal{D}_{t'}|}\sum_{v\in d}\log p(v|u_d,t')}{\sum_{t'=0}^{t}\sum_{d=1}^{|\mathcal{D}_{t'}|}N_d}\Big)$, where N_d is the number of words in document d, and $p(v|u_d,t') = \cos(\mathbf{v},\mathbf{u}_d)$. Here $\mathbf{u}_d$ is the embedding of the user associated with d. A lower perplexity score indicates better generalization performance. Fig. 3 shows the mean perplexity performance of DUWE and the baseline models, over different sizes of dimensions ranging between 10 and 300. We report the result with the length of each time period being a month as a representative. As it can be observed, DUWE consistently performs better than the rest of the models, with the performance flattening out when the dimensions are equal or more than ~100.

7.4 Dynamic Representations

RQ4: Next, we examine whether DUWE outperforms baselines on capturing the dynamics of embeddings to user profiling in streams.

We randomly choose one example user and display the top-K words from the ground truth and the top-K words generated by DUWE and the best baseline DPDR for profiling the user at every

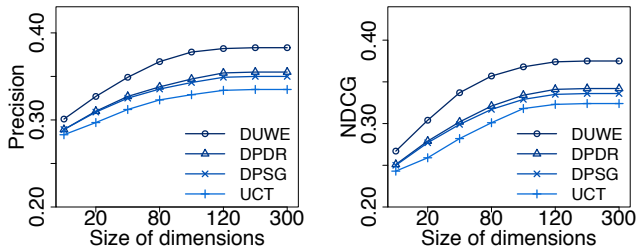

Figure 4: Precision and NDCG performance of DUWE and the baselines with various sizes of dimensions of embeddings.

quarter, respectively. Looking as the ground truth keywords in Table 3, the user's interests first center on the aspects "sports" and "plant" from April to June 2014 and then move to the aspects "education" and "electronic products" from April to May 2015. Compared to the best baseline, DUWE is more effective to track the user's interests over time and retrieve top-K relevant and diverse keywords to profile the user at different quarters, which demonstrates the high quality of the dynamic representations generated by our dynamic user and word embedding model, DUWE.

7.5 Dimensions of Representations

RQ5: Finally, we vary the sizes of dimensions of the embeddings in the models and evaluate their performance.

Fig. 4 shows the Precision and NDCG performance of DUWE and the best baselines, DPDR, DPSG and UCT, on different sizes of dimensions varying between 10 and 300. Performance evaluated by other metrics is not reported here, as it follows the same pattern. It is clear from the figure that the performance increases with the number of dimensions both in DUWE and the baselines, when the number of dimensions goes from 10 to ~100. The performance of all the models seems to be reaching a plateau when dimensions increase from ~100 to 300. At all different sizes, DUWE keeps outperforming all other baselines. All of these findings demonstrate another merit of our DUWE: it is not sensitive to the size of dimensions of the embeddings when the size is set to be large enough, and it is able to consistently improve user profiling performance with various sizes of the dimensions over the best embedding models.

8 CONCLUSION

We have studied the problem of user profiling over time in Twitter. To tackle the problem, we have proposed a dynamic user and word embedding model, DUWE, that is the first attempt to simultaneously model user and word embeddings over time in the same space. DUWE adopts a skip-gram model to a dynamic setup and trains on all the data up to the current time step, which allows end-to-end training. This leads to stable, continuous embedding trajectories, smooth out noise, avoid inappropriate semantic drifts, and share user-to-word and word-to-word statistics information across all the steps. To infer the dynamic embeddings, we have proposed a scalable black-box variational inference algorithm, which works with a convex objective. Our inference strategy guarantees that training the dynamic embeddings twice on the same data would result in the same results in contrast to other embedding models with non-convex objectives. To diversify top-K keywords for users' profiling over time, we have proposed a streaming keyword diversification model, SKDM. Experimental results on a publicly available dataset demonstrate the effectiveness of the proposed algorithms.

There are many aspects to be explored in future work, e.g., how to generate phrases instead of keywords for profiling, whether there other ways to model and infer user embeddings over time, whether there other datasets to verify our embedding model, or whether we can apply DUWE to other applications, e.g., rank aggregation [27], and verify the effectiveness of the embeddings there.

Acknowledgments

This work was supported by the King Abdullah University of Science and Technology (KAUST), Saudi Arabia.

A DERIVATIONS OF $\widetilde{\gamma}$ AND $\widetilde{\psi}$

According to (20), we have $p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{m}^{\pm}_{\leq t-1}) \approx \iint q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1}, \mathbf{V}_{t-1})$ $p(\mathbf{U}_t, \mathbf{V}_t \mid \mathbf{U}_{t-1}, \mathbf{V}_{t-1}) d\mathbf{U}_{t-1} d\mathbf{V}_{t-1} = \int q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1})\, p(\mathbf{U}_t \mid \mathbf{U}_{t-1})$ $d\mathbf{U}_{t-1} \cdot \int q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{V}_{t-1})\, p(\mathbf{V}_t \mid \mathbf{V}_{t-1}) d\mathbf{V}_{t-1}$. In the following, we only show the derivation for $\mathbb{E}_{q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1})} p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) = \int q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1})$ $p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) d\mathbf{U}_{t-1}$, as the derivation for $\mathbb{E}_{q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{V}_{t-1})} p(\mathbf{V}_t \mid \mathbf{V}_{t-1}) =$ $\int q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{V}_{t-1})\, p(\mathbf{V}_t \mid \mathbf{V}_{t-1}) d\mathbf{V}_{t-1}$ is essentially the same. Applying (9) and (16), and inserting the expressions for the Gaussian distributions, we have the following:

$$\begin{aligned}
&\mathbb{E}_{q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1})} p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) = \int q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{U}_{t-1})\, p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) d\mathbf{U}_{t-1} \\
&\propto \int \mathcal{N}(\mathbf{U}_{t-1}; \boldsymbol{\mu}_{t-1}, \boldsymbol{\sigma}^2_{t-1}\mathbf{I}) \mathcal{N}(\mathbf{U}_t; \mathbf{U}_{t-1}, \boldsymbol{\alpha}^2_{t-1}\mathbf{I}) \mathcal{N}(\mathbf{U}_t; \mathbf{0}, \overline{\boldsymbol{\alpha}}^2_0 \mathbf{I}) d\mathbf{U}_{t-1} \\
&\propto \int \exp\left[-\frac{1}{2}\left(\frac{(\mathbf{U}_{t-1} - \boldsymbol{\mu}_{t-1})^2}{\boldsymbol{\sigma}^2_{t-1}} + \frac{(\mathbf{U}_t - \mathbf{U}_{t-1})^2}{\boldsymbol{\alpha}^2_{t-1}} + \frac{\mathbf{U}_t^2}{\overline{\boldsymbol{\alpha}}^2_0}\right)\right] d\mathbf{U}_{t-1},
\end{aligned} \tag{30}$$

where let $\boldsymbol{\sigma}^2_{t-1}$ and $\boldsymbol{\mu}_{t-1}$ be abbreviated for the variances and means for all users's embeddings $\mathbf{U}_{t-1}$, respectively, and drop the constant prefactors and use a notation that is suitable for scalar values. In reality, $\boldsymbol{\sigma}^2_{t-1}$ is a matrix, but since it is diagonal we can treat each component as an independent scalar. To carry out the integral in (30), we pull all terms that are independent of $\mathbf{U}_{t-1}$ out of it, and then (30) becomes:

$$\begin{aligned}
&\propto \exp\left(-\frac{1}{2}\frac{\boldsymbol{\mu}^2_{t-1}}{\boldsymbol{\sigma}^2_{t-1}}\right) \times \exp\left[-\frac{1}{2}\left(\frac{1}{\boldsymbol{\alpha}^2_{t-1}} + \frac{1}{\overline{\boldsymbol{\alpha}}^2_0}\right)\mathbf{U}_t^2\right] \times \\
&\int \exp\left[-\frac{1}{2}\left(\frac{1}{\boldsymbol{\sigma}^2_{t-1}} + \frac{1}{\boldsymbol{\alpha}^2_{t-1}}\right)\mathbf{U}^2_{t-1} + \left(\frac{\boldsymbol{\mu}_{t-1}}{\boldsymbol{\sigma}^2_{t-1}} + \frac{\mathbf{U}_t}{\boldsymbol{\alpha}^2_{t-1}}\right)\mathbf{U}_{t-1}\right] d\mathbf{U}_{t-1},
\end{aligned} \tag{31}$$

where the first factor is a constant (independent of $\mathbf{U}_t$), which will be cancelled out. In the last factor, we sort in powers of $\mathbf{U}_{t-1}$, and can carry out the Gaussian integral in (31) by completing the square. Thus, (31) becomes:

$$\begin{aligned}
&\propto \exp\left[-\frac{1}{2}\left(\frac{1}{\boldsymbol{\alpha}^2_{t-1}} + \frac{1}{\overline{\boldsymbol{\alpha}}^2_0}\right)\mathbf{U}_t^2\right] \times \exp\left[\frac{1}{2}\left(\frac{1}{\boldsymbol{\sigma}^2_{t-1}} + \frac{1}{\boldsymbol{\alpha}^2_{t-1}}\mathbf{A}^2\right)\right] \times \\
&\int \exp\left[-\frac{1}{2}\left(\frac{1}{\boldsymbol{\sigma}^2_{t-1}} + \frac{1}{\boldsymbol{\alpha}^2_{t-1}}\right)(\mathbf{U}_{t-1} - \mathbf{A})^2\right] d\mathbf{U}_{t-1},
\end{aligned} \tag{32}$$

Table 3: Top six keywords of an example user's dynamic profile with the time being five quarters from April 2014 to May 2015. The keywords from the DGT ground truth, generated by the best baseline DPDR and our DUWE are presented for the user in the rows, respectively.

	Apr. 2014 to Jun. 2014	Jul. 2014 to Sep. 2014	Oct. 2014 to Dec. 2014	Jan. 2015 to Mar. 2015	Apr. 2015 to May 2015
Ground Truth	badminton leaf basketball flower bicycling root	muscle apple heart kiwi lungs pomelo	freezer fly toaster cockroach cabinet ant	injury clothes joint slacks immune slippers	school macbook teacher ipad assignment iphone
DPDR	badminton sky basketball herb coach grass	heart apple ankle pomelo finger peach	freezer water muffin fly toaster cockroach	injury clothes dose slacks food slippers	class ipad garden update teacher system
DUWE	badminton flower basketball leaf bicycling fruit	heart apple muscle kiwi breath pomelo	freezer ant dishwaster fly toaster cockroach	injury clothes ankle trousers doctor slacks	teacher laptop student apple school ipad

where $\mathbf{A} = \left(1/\boldsymbol{\sigma}_{t-1}^2 + 1/\boldsymbol{\alpha}_{t-1}^2\right)^{-1}\left(\boldsymbol{\mu}_{t-1}/\boldsymbol{\sigma}_{t-1}^2 + \mathbf{U}_t/\boldsymbol{\alpha}_{t-1}^2\right)$. The integral in (32) leads to a constant factor (independent of $\mathbf{U}_t$) because it is invariant under a constant shift of the integration variable, which will be cancelled out as well. Thus, (32) becomes:

$$\begin{aligned}
&\propto \exp\left[-\frac{1}{2}\left(\frac{1}{\boldsymbol{\alpha}_{t-1}^2} + \frac{1}{\overline{\boldsymbol{\alpha}}_0^2}\right)\mathbf{U}_t^2\right] \times \exp\left[\frac{1}{2}\left(\frac{1}{\boldsymbol{\sigma}_{t-1}^2} + \frac{1}{\boldsymbol{\alpha}_{t-1}^2}\mathbf{A}^2\right)\right] \\
&= \exp\left[-\frac{1}{2}\left(\frac{1}{\overline{\boldsymbol{\alpha}}_0^2} + \frac{1}{\boldsymbol{\sigma}_{t-1}^2 + \boldsymbol{\alpha}_{t-1}^2}\right)\mathbf{U}_t^2 + \frac{\boldsymbol{\mu}_{t-1}}{\boldsymbol{\sigma}_{t-1}^2 + \boldsymbol{\alpha}_{t-1}^2}\mathbf{U}_t\right] \\
&\propto \mathcal{N}(\mathbf{U}_t; \widetilde{\boldsymbol{\gamma}}_t, \widetilde{\boldsymbol{\psi}}_t^2\mathbf{I}),
\end{aligned} \tag{33}$$

where we let $\widetilde{\boldsymbol{\gamma}}_t$ and $\widetilde{\boldsymbol{\psi}}_t^2$ abbreviate for the means and variances for all users' embeddings $\mathbf{U}_t$, respectively, and let:

$$\frac{1}{\widetilde{\boldsymbol{\psi}}_t^2} = \frac{1}{\overline{\boldsymbol{\alpha}}_0^2} + \frac{1}{\boldsymbol{\sigma}_{t-1}^2 + \boldsymbol{\alpha}_{t-1}^2}, \quad \text{and} \quad \frac{\widetilde{\boldsymbol{\gamma}}_t}{\widetilde{\boldsymbol{\psi}}_t^2} = \frac{\boldsymbol{\mu}_{t-1}}{\boldsymbol{\sigma}_{t-1}^2 + \boldsymbol{\alpha}_{t-1}^2}, \tag{34}$$

which results in:

$$\widetilde{\boldsymbol{\gamma}}_t = \widetilde{\boldsymbol{\psi}}_t^2\left(\boldsymbol{\sigma}_{t-1}^2 + \boldsymbol{\alpha}_{t-1}^2\mathbf{I}\right)^{-1}\boldsymbol{\mu}_{t-1}, \tag{35}$$

$$\widetilde{\boldsymbol{\psi}}_t^2 = \left[\left(\boldsymbol{\sigma}_{t-1}^2 + \boldsymbol{\alpha}_{t-1}^2\mathbf{I}\right)^{-1} + (1/\overline{\boldsymbol{\alpha}}_0^2)\mathbf{I}\right]^{-1}. \tag{36}$$

REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.
[2] K. Balog and M. de Rijke. Determining expert profiles (with and application to expert finding). In *IJCAI*, pages 2657–2662, 2007.
[3] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *SIGIR*, pages 551–558, 2007.
[4] R. Bamler and S. Mandt. Dynamic word embeddings. In *ICML*, pages 380–389, 2017.
[5] C. Bishop. Pattern recognition and machine learning (information science and statistics). *Springer, New York*, 2007.
[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. 2016.
[8] P. Bromiley. Products and convolutions of gaussian probability density functions. *Tina-Vision Memo*, 3(4), 2014.
[9] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.
[10] R. Collobert, J. Weston, L. Bottou, and et al. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.
[11] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC 2005 enterprise track. In *TREC'05*, pages 1–7, 2005.
[12] W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2015.
[13] V. Dang and W. B. Croft. Diversity by proportionality: An election-based approach to search result diversification. In *SIGIR*, pages 65–74, 2012.
[14] Y. Fang and A. Godavarthy. Modeling the dynamics of personal expertise. In *SIGIR*, pages 1107–1110, 2014.
[15] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, pages 1764–1772, 2014.
[16] W. L. Hamilton, J. Leskovec, and D. Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *ACL*, pages 1489–1501, 2016.
[17] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*, volume 9, pages 1427–1432, 2009.
[18] Y. Kim, Y.-I. Chiu, K. Hanaki, D. Hegde, and S. Petrov. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515*, 2014.
[19] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
[20] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically significant detection of linguistic change. In *WWW*, pages 625–635, 2015.
[21] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW'10*, pages 591–600, 2010.
[22] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.
[23] S. Liang and M. de Rijke. Formal language models for finding groups of experts. *Information Processing & Management*, 52(4):529–549, 2016.
[24] S. Liang, E. Yilmaz, and E. Kanoulas. Dynamic clustering of streaming short documents. In *KDD*, pages 995–1004, 2016.
[25] S. Liang, Z. Ren, Y. Zhao, J. Ma, E. Yilmaz, and M. D. Rijke. Inferring dynamic user interests in streams of short texts for user clustering. *ACM Trans. Inf. Syst.*, 36(1):10:1–10:37, 2017.
[26] S. Liang, E. Yilmaz, H. Shen, M. D. Rijke, and W. B. Croft. Search result diversification in short text streams. *ACM Trans. Inf. Syst.*, 36(1):8:1–8:35, 2017.
[27] S. Liang, I. Markov, Z. Ren, and M. de Rijke. Manifold learning for rank aggregation. In *WWW*, pages 1735–1744, 2018.
[28] S. Liang, E. Yilmaz, and E. Kanoulas. Collaboratively tracking interests for user clustering in streams of short texts. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–16, 2018.
[29] R. Mihalcea and V. Nastase. An introduction to the kalman filter. In *SIGGRAPH*, pages 27599–23175, 2001.
[30] R. Mihalcea and V. Nastase. Word epoch disambiguation: Finding how words change over time. In *ACL*, pages 259–263, 2012.
[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
[32] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
[33] S. Okura, Y. Tagami, S. Ono, and A. Tajima. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942, 2017.
[34] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.
[35] R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.
[36] J. Rybak, K. Balog, and K. Nørvåg. Temporal expertise profiling. In *ECIR*, pages 540–546, 2014.
[37] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, pages 901–909, 2016.
[38] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD*, pages 424–433, 2006.
[39] X. Wei, J. Sun, and X. Wang. Dynamic mixture models for multiple time-series. In *IJCAI*, pages 2909–2914, 2007.
[40] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. 2015.
[41] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, 2001.
[42] Y. Zhao, S. Liang, Z. Ren, J. Ma, E. Yilmaz, and M. de Rijke. Explainable user clustering in short text streams. In *SIGIR*, pages 155–164, 2016.