Dynamic embeddings for user profiling in Twitter

Liang, S.; Zhang, X.; Ren, Z.; Kanoulas, E.

DOI 10.1145/3219819.3220043

 Publication date 2018

Document Version Final published version

Published in KDD '18 : proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining

License Article 25fa Dutch Copyright Act

Dynamic Embeddings for User Profiling in Twitter

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

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

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

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) 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. We utilize Bamler and Mandt’s skip-gram Filtering algorithm [4] for our inference, which 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

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].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
(3) 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 static word embedding models across different time periods. To model changes in word embeddings over time, several approaches have been proposed in the literature [18, 16]. 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 and propose a skip-gram filtering algorithm 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. To the best of our knowledge, this is the first attempt to explicitly model the evolution of users and words 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 operation of the algorithm is essentially a function \( f \) that satisfies:

\[
\mathcal{U}_i, \mathcal{D}_{\leq t} \xrightarrow{f} \mathcal{W}_i,
\]

where \( \mathcal{U}_i = \{u_i^t\}_{i=1}^{U_i} \) is a set of users in the stream at time \( t \), \( |\mathcal{U}_i| \) being the number of users, \( \mathcal{D}_{\leq t} = \{D_j\}_{j=0}^{t} \) is the stream of documents generated by the users up to time \( t \), with \( D_j \) being a set of documents generated by all the users at time \( t \), and \( \mathcal{W}_i = \{W_{u_i^t}, \ldots, W_{u_i^t|D_j}\} \) are all users’ profiling results at time \( t \) with \( W_{u_i^t} = \{w_{u_i^t, 1}, \ldots, w_{u_i^t, K_i}\} \) being the profiling result, i.e., the top-K diversified keywords, for user \( u_i^t \) 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, \( U_i = \{u_i^t\}_{i=1}^{U_i} \), and words, \( V = \{v_k\}_{k=1}^{V} \), over time. Here \( u_i^t \) and \( v_k \) 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 or 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 \( p(z_{k,l} = 1 | v_k, v_l) = s(x) \), 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 \( n_{k,l}^t \) 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

1765
to the likelihood is obtained as \( p(n^+ | v_k, v_l) = s(v_l^T v_k)n^+_k,l \). 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:

\[
p(n^+ n^- | V) = \sum_{k,l=1}^V s(v_l^T v_k)n^+_k,l \times s(-v_l^T v_k)n^-_{k,l}, \tag{1}
\]

where \( n^+, n^- \in \mathbb{R}^{V \times V} \) are the positive and negative indicator matrices for all word pairs with \( n^+_k,l \) and \( n^-_{k,l} \) being their elements, respectively; \( s(-\chi) = 1 - s(\chi) \), and \( V = \{v_k \} \) 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 \( n^- \) is well-defined up to a proportional constant factor that needs to be tuned experimentally. Instead of maximizing the likelihood in (1) directly, the skip-gram model tries to maximize its log likelihood:

\[
\log p(n^+ | V) = \sum_{k,l=1}^V n^+_k,l \log s(v_l^T v_k) + n^-_{k,l} \log s(-v_l^T v_k), \tag{2}
\]

where we denote \( n = (n^+, n^-) \).

The assumptions made in (1) are not realistic when it comes to the setting of streams, where the embeddings 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 apply the skip-gram filtering [4] 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 DUWE, we consider a diffusion process of the 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 \( \alpha_{u,-1} = \{\alpha_{u,t-1}^k \}_{t \leq k} \) with \( \alpha_{u,t-1} \) being the variance of the transition kernel for user \( u \)’s embedding transferring from \( t-1 \) to \( t \). According to Kalman filtering [29], we can define \( \alpha_{u,t-1} \) as:

\[
\alpha_{u,t-1} = \epsilon \cdot g(D_{u,t}, D_{u,t-1})(\tau_t - \tau_{t-1}), \tag{3}
\]

where \( \epsilon \) is a local diffusion constant for user \( u \)’s embedding, \( D_{u,t} \in D_t \cup \cup_u D_{u,t-1} \) is a set of documents generated by user \( u \) at time \( t \), \( g(D_{u,t}, D_{u,t-1}) \) 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 \( D_{u,t} = \{\theta_{u,t}, \theta_{v_1,t}, \ldots, \theta_{v_{\epsilon},t} \} \) be a vector representation for \( u \)’s document set \( D_{u,t} \) at time \( t \), with its element \( \theta_{u,t} \) being computed by an unsupervised language model with Dirichlet smoothing [12, 41] as:

\[
\theta_{u,t} = \frac{c(v_k, D_{u,t}) + \delta \cdot p(v_k | D_{\leq t})}{\sum_v c(v; D_{u,t}) + \delta}, \tag{4}
\]

where \( c(v; D_{u,t}) \) is the total number of times the word \( v \) appearing in the document set \( D_{u,t} \), \( p(v | D_{\leq t}) \) is the probability of the word \( v \) appearing in the whole corpus \( D_{\leq t} \), and \( \delta \) 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:

\[
g(D_{u,t}, D_{u,t-1}) = 1 - \exp \left\{ -\frac{1}{2} (\text{KL}(D_{u,t} || D_{u,t-1}) + \text{KL}(D_{u,t-1} || D_{u,t})) \right\}, \tag{5}
\]

where \( \text{KL}(\|\cdot\|) \) is the Kullback-Leibler (KL) divergence. According to (5), if \( D_{u,t} = D_{u,t-1} \), 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 \( \beta_{t-1} = \{\beta_{t-1}^v \}_{v \leq 1} \) with \( \beta_{t-1}^v \) 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 \( \beta_{t-1} \) as:

\[
\beta_{t-1} = \eta \cdot h(D_t, D_{t-1})(\tau_t - \tau_{t-1}), \tag{6}
\]

where \( \eta \) is a local diffusion constant for words’ embeddings and \( h(D_t, D_{t-1}) \) is a local diffusion value measuring the words’ distribution changes from \( t-1 \) to \( t \). Let \( D_t = \{\phi_{v_1,t}, \phi_{v_2,t}, \ldots, \phi_{v_{\epsilon},t} \} \) be a vector representation for document set \( D_t \), with its element, \( \phi_{v,t} \), being computed by an unsupervised language model with Dirichlet smoothing [12, 41] as:

\[
\phi_{v,t} = \frac{c(v; D_t) + \delta \cdot p(v | D_{\leq t})}{\sum_v c(v; D_t) + \delta}. \tag{7}
\]

According to (8), if \( D_t = D_{t-1} \), 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 \( \tilde{p}(U_t | U_{t-1}) \), that is the the probability of user embedding (before normalization), \( U_t \), at current time step, \( t \), given the user embedding at the previous time step, \( t-1 \), \( U_{t-1} \), we add a Gaussian prior with mean \( \theta \) and variance \( \tilde{\alpha}_u^2 = \{[\tilde{\alpha}_u^2]^k \}_{t \leq k} \), which prevents the user embedding vectors from growing too large. Thus,

\[
\tilde{p}(U_t | U_{t-1}) = \mathcal{N}(U_t; \tilde{\alpha}_u^2 I) \cdot N(0, \tilde{\alpha}_u^2 I). \tag{9}
\]

where \( N(\cdot, \cdot) \) is a Gaussian distribution, and \( I \) is an identity matrix. Similarly, to model \( \tilde{p}(V_t | V_{t-1}) \), that is the probability of word embedding (before normalization) at \( t \), \( V_t \), given the word embedding results at \( t-1 \), \( V_{t-1} \), we add a Gaussian prior with mean \( \theta \) and
where for all user-to-word pairs with the users’ and words’ embeddings 
and positive (observed user-to-word pairs 
and negative (fictitious rejected user-to-word pairs 
\( \alpha \) and \( \beta \), respectively. At 
\( \lambda \), i.e., let \( \lambda \) to be the free parameters of a variational distribution \( q_{k,t}(U_{st}, V_{st}) \). Here, \( \lambda_{st} = \{\lambda_{t} \}_{t=1}^{T} \) 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 \( q_{k,t}(U_{st}, V_{st}) \) by minimizing the KL divergence to the posterior. Minimizing the KL divergence is equivalent to maximizing the following Evidence Lower BOUND (ELBO) [35]:

\[
\mathcal{L}(\lambda_{st}) = \mathbb{E}_{q_{k,t}}(U_{st}, V_{st})[\log p(m_{st}^{+}, n_{st}^{+}, U_{st}, V_{st})] - \mathbb{E}_{q_{k,t}}(U_{st}, V_{st})[\log q_{k,t}(U_{st}, V_{st})].
\]

**4.3 Inference**

The following derivation follows closely the skip-gram filtering algorithm proposed by Bamber and Mandt in [4]. To infer the users’ and words’ embedding results at time step \( t \), \( U_{t} \) and \( V_{t} \), we start by formulating a joint distribution of our DUWE model, i.e., (13), over \( m_{st}^{+} \) and \( n_{st}^{+} \), and the users’ and words’ embeddings \( U_{st} \) and \( V_{st} \) across all the time. Accordingly, we are interested in the posterior distribution over \( U_{st} \) and \( V_{st} \) conditioned on the statistics information \( m_{st}^{+} \) and \( n_{st}^{+} \) as follows:

\[
p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+}) = \frac{p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+})}{\int p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+}) \, dU_{st} \, dV_{st}}.
\]

**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 \( U_{st} \) and \( V_{st} \). Let \( \lambda_{st} \) be the free parameters of a variational distribution \( q_{k,t}(U_{st}, V_{st}) \). Here, \( \lambda_{st} = \{\lambda_{t} \}_{t=1}^{T} \) 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 \( q_{k,t}(U_{st}, V_{st}) \) by minimizing the KL divergence to the posterior. Minimizing the KL divergence is equivalent to maximizing the following Evidence Lower BOUND (ELBO) [35]:

\[
\mathcal{L}(\lambda_{st}) = \mathbb{E}_{q_{k,t}}(U_{st}, V_{st})[\log p(m_{st}^{+}, n_{st}^{+}, U_{st}, V_{st})] - \mathbb{E}_{q_{k,t}}(U_{st}, V_{st})[\log q_{k,t}(U_{st}, V_{st})].
\]

**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 cannot 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 \( q_{k,t}(U_{st}, V_{st}) \) given the statistics \( m_{st}^{+} \) and \( n_{st}^{+} \) from \( 0 \) to \( t \). Therefore, we define a variational distribution that is factorized across all time steps up to \( t \), i.e., let \( q_{k,t}(U_{st}, V_{st}) = \prod_{t'=1}^{t} q_{k,t'}(U_{st}, V_{st}) \). We adopt the mean-field approximation inference strategy, and thus factorize the distribution \( q_{k,t}(U_{t}, V_{t}) \) at \( t \) as:

\[
q_{k,t}(U_{t}, V_{t}) = q_{k,t}(U_{t}) \cdot q_{k,t}(V_{t})
\]

\[
\approx \prod_{t=1}^{T} \prod_{k,l=1}^{N} (m_{k,l,t'}^{+}, n_{k,l,t'}^{-}) \cdot \prod_{k,l=1}^{N} N(u_{l,t'}; \mu_{u_{l,t'}}, \sigma_{u_{l,t'}}^{2}), \prod_{k,l=1}^{N} N(v_{k,t'}; \mu_{v_{k,t'}}, \sigma_{v_{k,t'}}^{2}).
\]

\[
\text{where } \mu_{u_{l,t'}}, \sigma_{u_{l,t'}}, \mu_{v_{k,t'}}, \sigma_{v_{k,t'}} \text{ are the means of the user } u_{l} \text{’s and the word } v_{k} \text{’s embeddings at } t \text{, respectively; and } \sigma_{u_{l,t'}}^{2} \text{ and } \sigma_{v_{k,t'}}^{2} \text{ are the corresponding variances, respectively. At } t, \text{ given } m_{st}^{+} \text{ and } n_{st}^{+}, \text{ and the fact that } \lambda_{st-1}, U_{st-1} \text{ and } V_{st-1} \text{ have been obtained, the goal of our inference algorithm is to infer the variational parameters in } q_{k,t}(U_{t}, V_{t}) \text{ at } t \text{, i.e., } \lambda_{t} = \{\mu_{u_{l,t}}, \sigma_{u_{l,t}}^{2} \}_{l=1}^{V}, \{\mu_{v_{k,t}}, \sigma_{v_{k,t}}^{2} \}_{k=1}^{V}.
\]

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

\[
p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+}) \propto p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+}, p(m_{st}^{+}, n_{st}^{+}) = p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+})
\]

\[
\propto p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+}) = p(U_{st}, V_{st} | m_{st}^{+}, n_{st}^{+})
\]
\[
\begin{align*}
&= \prod_{t' = 0}^{t} p(U_{t'}, V_{t'} | m_{t'}^\pm) \cdot \prod_{t' = 0}^{t} p(V_{t'} | n_{t'}^\pm) \\
&\propto \prod_{t' = 0}^{t} \left( p(m_{t'}^+ | U_{t'}, V_{t'}) p(U_{t'}, V_{t'} | m_{t'}^+, n_{t'}^+) \right) \times p(m_{t'}^- | \Omega(t') \cup \psi_{\Omega(t')}^2).
\end{align*}
\]

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., \(\mathcal{L}(\lambda_{t-1}) = \sum_{t'=0}^{t} \mathcal{L}(\lambda_{t'})\) with \(\mathcal{L}(\lambda_{t'})\) for time step \(t\) being the following:

\[
\mathcal{L}(\lambda_{t'}) = \log p(U_{t'}, V_{t'}) + \sum_{t' = 0}^{t} \log p(U_{t'}, V_{t'} | m_{t'}^+ + n_{t'}^-) + \sum_{t' = 0}^{t} \log p(V_{t'} | m_{t'}^+) + \sum_{t' = 0}^{t} \log q_{\lambda_{t'}}(U_{t'}, V_{t'})
\]

where similar to (2), \(\log p(m_{t'}^+ | U_{t'}, V_{t'})\) can be computed as:

\[
\log p(m_{t'}^+ | U_{t'}, V_{t'}) = \sum_{i=1}^{V} \sum_{k=1}^{m_{t'}^+} s(u_{t'}^T v_k) + m_{t'}^- \log s(-u_{t'}^T v_k). \tag{19}
\]

\[
\log p(n_{t'}^- | V_{t'}) \text{ is computed by (2), and } \log q_{\lambda_{t'}}(U_{t'}, V_{t'}) \text{ can be computed according to (16), respectively; and thus their corresponding expectations can be computed easily. However } p(U_{t'}, V_{t'} | m_{t'}^+) \text{ and } p(V_{t'} | n_{t'}^-) \text{ in (18) are still intractable. Applying the variational inference results in the previous step, } q_{\lambda_{t-1}}(U_{t-1}, V_{t-1}) \approx p(U_{t-1}, V_{t-1}) \text{ and } q_{\lambda_{t-1}}(V_{t-1}) \approx p(V_{t-1}), \text{ we can approximate these two probabilities, respectively:}
\]

\[
\begin{align*}
p(U_{t'}, V_{t'} | m_{t'}^+) &\approx \mathbb{E}_{q_{\lambda_{t-1}}(U_{t-1}, V_{t-1})} [p(U_{t'}, V_{t'} | U_{t-1}, V_{t-1})] \\
&= \mathbb{E}_{q_{\lambda_{t-1}}(U_{t-1}, V_{t-1})} [p(U_{t'} | U_{t-1}, V_{t-1})], \\
p(V_{t'} | n_{t'}^-) &\approx \mathbb{E}_{q_{\lambda_{t-1}}(V_{t-1})} [p(V_{t'} | V_{t-1})] \\
&= \mathbb{E}_{q_{\lambda_{t-1}}(V_{t-1})} [p(V_{t'} | V_{t-1})].
\end{align*}
\]

The resulting approximate probability of (21) is:

\[
\begin{align*}
p(V_{t'} | n_{t'}^-) &\approx \prod_{k=1}^{V} \mathcal{N}(v_{k,t}; \tilde{\psi}_{v_{k,t}}, \psi_{v_{k,t}}^2).
\end{align*}
\]

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:

\[
\nabla_{\lambda_{t}} \mathcal{L}(\lambda_{t}) = \mathbb{E}_{q_{\lambda_{t}}(U_{t}, V_{t})} \left[ \nabla_{\lambda_{t}} \log q_{\lambda_{t}}(U_{t}, V_{t}) \log p(U_{t}, V_{t}) \right] + \mathbb{E}_{q_{\lambda_{t}}(U_{t}, V_{t})} \left[ \nabla_{\lambda_{t}} \log q_{\lambda_{t}}(U_{t}, V_{t}) \log p(U_{t}, V_{t} | m_{t-1}^+) \right] + \mathbb{E}_{q_{\lambda_{t}}(U_{t}, V_{t})} \left[ \nabla_{\lambda_{t}} \log q_{\lambda_{t}}(V_{t}) \log p(V_{t} | n_{t-1}^-) \right] + \mathbb{E}_{q_{\lambda_{t}}(U_{t}, V_{t})} \left[ \nabla_{\lambda_{t}} \log q_{\lambda_{t}}(U_{t}, V_{t}) \right].
\]

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:

\[
\begin{align*}
\nabla_{\lambda_{t}} \mathcal{L}(\lambda_{t}) &\approx \frac{1}{S} \sum_{s=1}^{S} \left[ \nabla_{\lambda_{t}} \log q_{\lambda_{t}}(U_{t}, V_{t}) \left( \log p(U_{t}^s, V_{t}^s) \right) \right. \\
&\quad + \log p(U_{t}^s, V_{t}^s | m_{t-1}^+) + \log q_{\lambda_{t}}(U_{t}^s, V_{t}^s) \left. \right]
\end{align*}
\]

\[
\text{where } (U_{t}^s, V_{t}^s) \sim q_{\lambda_{t}}(U_{t}, V_{t}).
\]

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

\[
\lambda_{t+1}^i = \lambda_{t}^i + \rho_{t}^{i+1} \cdot \nabla_{\lambda_{t}} \mathcal{L}(\lambda_{t}^i),
\]

where \(\rho_{t}^{i+1}\) 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 \(t\), \(U_{t}^*\) and \(V_{t}^*\) as:

\[
(U_{t}^*, V_{t}^*) = \arg \max_{U_{t}^*, V_{t}^*} \lambda_{i}^t(U_{t}, V_{t}) = (|\mu_{u_{t-1}}| V_{t}, |\mu_{v_{t-1}}| V_{t})
\]

where \(\lambda_{i}^t\) 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 PM2 [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 $t$, $U_t$ and $V_t$.
Output: All users’ profiling results at time $t$, $W_{u,t}$.

1. for $u = 1, \ldots, |U_t|$ do
2. \hspace{1em} $W_{u,t} \leftarrow \emptyset$ \hspace{1em} /* $W_{u,t} \in W_t$ */
3. \hspace{1em} $N_u \leftarrow$ retrieve top-$N$ words with their embeddings similar to $u_t$
4. \hspace{1em} $C \leftarrow$ perform K-means on $N_u$
5. \hspace{1em} for $c = 1, \ldots, |C|$ do
6. \hspace{2em} $P_{c,u,t} \leftarrow P(c \mid u, t)$
7. \hspace{2em} $s_{c,u,t} \leftarrow 0$
8. \hspace{1em} end
9. \hspace{1em} for all positions in the ranked list $W_{u,t}$ do
10. \hspace{2em} $q_t[c,u,t] \leftarrow \frac{P_{c,u,t}}{s_{c,u,t}^{1/2}}$
11. \hspace{2em} $c^* \leftarrow \arg \max_c q_t[c,u,t]$ \hspace{1em} /* append $c^*$ to $W_{u,t}$ */
12. \hspace{2em} $v^* \leftarrow \arg \max_{v \in C^c} (\kappa P(v|c^*, u, t) + (1 - \kappa) \text{tfidf}(v|t, u))$
13. \hspace{2em} $W_{u,t} \leftarrow W_{u,t} \cup \{v^*\}$
14. \hspace{2em} $s_{c^*,u,t} \leftarrow s_{c^*,u,t} + \text{tfidf}(v^*|t, u)$
15. \hspace{1em} end

For each user $u$ at $t$, SKDM starts with an empty keyword set $W_{u,t}$ 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 $N_u$ whose embeddings have highest cosine similarities to user $u$’ embedding. It then performs K-means on $N_u$ to obtain clusters of the words (step 4). For each of the clusters, it computes the quotient $q_t[c,u,t]$ for each cluster $c$ given a user $u$ at time $t$ by: $q_t[c,u,t] = \frac{P_{c,u,t}}{s_{c,u,t}^{1/2}}$, where $P_{c,u,t}$ is the probability of the user $u$ being interested in cluster $c$ at time $t$ as denoted by $P(c|u, t)$ and is set to be $P_{c,u,t} = P(c|u, t) = \frac{1}{Z_C} \cos(v_C, v_c)$ (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 $v_C$ 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 $u$, defined as $P(v|c^*, u, t) = \frac{1}{Z_C} \cos(v_C, v_c)$, where $Z_C$ is a normalization term. We then propose to obtain the keyword $v^*$ for user $u$’s profiling at $t$ as (step 12): $v^* \leftarrow \arg \max_{v \in C^c} (\kappa P(v|c^*, u, t) + (1 - \kappa) \text{tfidf}(v|t, u))$, where $0 \leq \kappa \leq 1$ is a trade-off free parameter, and $\text{tfidf}(v|t, u)$ is a time-sensitive term frequency-inverse document frequency function for user $u$ at $t$, which can be defined as: $\text{tfidf}(v|t, u) = \frac{\text{idf}(v|t, u)}{|D_u|}$, where $\text{idf}(v|u, D_t) = \log \frac{|D_u|}{|\{d \in D_u, v \in d\}|/|D_t|}$ is the term frequency function that computes how many percent of the documents that contain the word $v$ in the whole document set $D_u$, and $\text{idf}(v|u, D_t) = \log \frac{|D_u|}{|\{d \in D_u, v \in d\}|/|D_t|}$ is the inverse document frequency function with $v$ being set to 1 to avoid the division-by-zero error. According to the tfidf function, if the word $v$ frequently appears in $D_u$ generated by user $u$ but not frequently appears in the document set $D_t$ generated by all the users in $U_t$, tfidf($v|t, u$) will return a high score. After the word $v^*$ is selected, SKDM adds $v^*$ as a result keyword to $W_{u,t}$ for profiling the user $u$ at $t$, i.e., $W_{u,t} = W_{u,t} \cup \{v^*\}$ (step 13), removes it from the cluster $c^*$ (step 14), and increases the “number” of seats occupied by the cluster $c^*$ as (step 15): $s_{c^*,u,t} = s_{c^*,u,t} + P(v^*|c^*, u, t)$. The process (steps 8 to 15) repeats until we get $K$ diversified keywords for $W_{u,t}$. The order in which a keyword is appended to $W_{u,t}$ determines its ranking for the profiling. After the process is done, we obtain a set of diversified keywords $W_{u,t}$ 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]. 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 Ground Truth: one for evaluating Relevance-oriented (RGT) performance and another for evaluating Diversity-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.
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].

Distribution 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 word vectors are 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 in both 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 the embedding baselines, we set the number of negative word pairs \((v_k, v_j)\) samples at time \(t\) to

\[
N_{k,j,t} = \left( \sum_{k',l,t=1}^V n_{k',l,t} \right) \cdot \xi \cdot p_t(k) \cdot p'_t(l),
\]

where \(\xi\) is the ratio of negative to positive word pairs. We set it to 1.0, \(p_t(k) = \sum_{l,t=1}^V n_{k,l,t} \), and \(p'_t(l) = \frac{p_t(l)}{\sum_{r,t=1}^V (p_t(r))^\gamma}\). We follow the same way to define the number of negative user-word pair \((u_i, v_j)\) samples in \(m^-\) 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 \(\epsilon\) in (3), \(\eta\) in (6), \(\alpha\) in (9) and \(\beta\) in (10); \(\epsilon, \eta, \alpha, \beta\) take values in \([0.001, 0.01, 0.1, 0.2, \ldots, 1, 2, \ldots, 10]\), respectively (elements in \(\alpha\) are set to be equal at each training time; the same setting to those in \(\beta\)). The optimal \(\epsilon, \eta, \alpha\) and \(\beta\) 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 \(\rho_t^{l+1}\) 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], and 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 \(\cos(v^*, v_{gt})\). 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 ~ DR ~ 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 $\boldsymbol{\bigtriangleup}$ for $\alpha = .01$, and $\boldsymbol{\bigtriangledown}$ for $\alpha = .05$.

<table>
<thead>
<tr>
<th></th>
<th>Pre</th>
<th>NDCG</th>
<th>MRR</th>
<th>Pre</th>
<th>NDCG-S</th>
<th>MRR-S</th>
<th>MAP-S</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SGM</td>
<td>.264</td>
<td>.232</td>
<td>.662</td>
<td>.135</td>
<td>.410</td>
<td>.388</td>
<td>.865</td>
<td>.208</td>
</tr>
<tr>
<td>DRD</td>
<td>.268</td>
<td>.234</td>
<td>.662</td>
<td>.137</td>
<td>.412</td>
<td>.392</td>
<td>.864</td>
<td>.210</td>
</tr>
<tr>
<td>PLM</td>
<td>.273</td>
<td>.239</td>
<td>.668</td>
<td>.140</td>
<td>.417</td>
<td>.398</td>
<td>.870</td>
<td>.212</td>
</tr>
<tr>
<td>DISG</td>
<td>.295</td>
<td>.279</td>
<td>.721</td>
<td>.150</td>
<td>.426</td>
<td>.417</td>
<td>.873</td>
<td>.220</td>
</tr>
<tr>
<td>DDDR</td>
<td>.308</td>
<td>.287</td>
<td>.725</td>
<td>.153</td>
<td>.438</td>
<td>.425</td>
<td>.876</td>
<td>.225</td>
</tr>
<tr>
<td>UCT</td>
<td>.335</td>
<td>.324</td>
<td>.787</td>
<td>.172</td>
<td>.462</td>
<td>.457</td>
<td>.883</td>
<td>.237</td>
</tr>
<tr>
<td>DPFG</td>
<td>.352</td>
<td>.338</td>
<td>.792</td>
<td>.178</td>
<td>.470</td>
<td>.459</td>
<td>.897</td>
<td>.242</td>
</tr>
<tr>
<td>DPDR</td>
<td>.355</td>
<td>.342</td>
<td>.812</td>
<td>.184</td>
<td>.476</td>
<td>.463</td>
<td>.905</td>
<td>.247</td>
</tr>
<tr>
<td>DUWE</td>
<td>.383</td>
<td>.375</td>
<td>.854</td>
<td>.207</td>
<td>.493</td>
<td>.483</td>
<td>.932</td>
<td>.262</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Pre</th>
<th>$\alpha$-NDCG</th>
<th>MRR</th>
<th>Pre</th>
<th>$\alpha$-NDCG</th>
<th>MAP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-IA</td>
<td>.158</td>
<td>.188</td>
<td>.482</td>
<td>.183</td>
<td>.260</td>
<td>.325</td>
<td>.734</td>
</tr>
<tr>
<td>CG</td>
<td>.159</td>
<td>.189</td>
<td>.485</td>
<td>.183</td>
<td>.262</td>
<td>.328</td>
<td>.738</td>
</tr>
<tr>
<td>-IA</td>
<td>.162</td>
<td>.192</td>
<td>.487</td>
<td>.187</td>
<td>.265</td>
<td>.332</td>
<td>.742</td>
</tr>
<tr>
<td>-IA-S</td>
<td>.183</td>
<td>.217</td>
<td>.504</td>
<td>.198</td>
<td>.294</td>
<td>.352</td>
<td>.760</td>
</tr>
<tr>
<td>SGM</td>
<td>.187</td>
<td>.228</td>
<td>.517</td>
<td>.205</td>
<td>.306</td>
<td>.364</td>
<td>.765</td>
</tr>
<tr>
<td>DRD</td>
<td>.209</td>
<td>.245</td>
<td>.543</td>
<td>.224</td>
<td>.328</td>
<td>.395</td>
<td>.778</td>
</tr>
<tr>
<td>PLM</td>
<td>.223</td>
<td>.256</td>
<td>.573</td>
<td>.234</td>
<td>.337</td>
<td>.412</td>
<td>.783</td>
</tr>
<tr>
<td>DISG</td>
<td>.225</td>
<td>.260</td>
<td>.580</td>
<td>.242</td>
<td>.348</td>
<td>.426</td>
<td>.787</td>
</tr>
<tr>
<td>DDDR</td>
<td>.257</td>
<td>.293</td>
<td>.617</td>
<td>.258</td>
<td>.387</td>
<td>.447</td>
<td>.808</td>
</tr>
</tbody>
</table>

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 $\alpha$-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 those of simply splitting the data into separate time bins and then obtaining embeddings from each of the bins as dynamic embedded 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.

7.3 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]. 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 Perplexity$(D_{t}) = \exp \left( \frac{- \sum_{t' = 0}^{t} |D_{t'}|}{\sum_{d=1}^{N_d} \sum_{v=1}^{D_{t'}} \log p(v|u_{d}, t')} \right)$, where $N_d$ is the number of words in document $d$, and $p(v|u_{d}, t') = \cos(v, u_{d})$. Here $u_{d}$ 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...
words to profile the user at different quarters, which demonstrates that the embedding model, DUWE, is able to consistently improve user profiling performance with the number of dimensions going from 10 to 300. At all different sizes, DUWE keeps outperforming other baselines with various sizes of dimensions of embeddings.

### 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. We closely follow Bamler and Mandt’s skip-gram filtering algorithm [4] to infer the dynamic embeddings of users and words in Twitter. 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. We thank Dr. R. Bamler and Dr. S. Mandt for their suggestion on the derivation of the inference.

#### A DERIVATIONS OF $\overline{Y}$ AND $\overline{Y^*}$

According to (20), we have $p(U_t, V_t | m^z_{(t−1)}) \approx \int q_{\lambda_t}(U_{t−1}, V_{t−1}) p(U_t, V_t | U_{t−1}, V_{t−1}, m_{(t−1)})dU_{t−1}dV_{t−1} = \int q_{\lambda_t}(U_{t−1})p(U_t | U_{t−1})dU_{t−1} \int q_{\lambda_t}(V_{t−1})p(V_t | U_{t−1})dV_{t−1}$. In the following, we only show the derivation for $\mathbb{E}_{q_{\lambda_t}(U_{t−1})}p(U_t | U_{t−1}) = \int q_{\lambda_t}(U_{t−1})p(U_t | U_{t−1})dU_{t−1}$, as the derivation for $\mathbb{E}_{q_{\lambda_t}(V_{t−1})}p(V_t | V_{t−1}) = \int q_{\lambda_t}(V_{t−1})p(V_t | V_{t−1})dV_{t−1}$ is essentially the same. Applying (9) and (16), and inserting the expressions for the Gaussian distributions, we have the following:

$$\mathbb{E}_{q_{\lambda_t}(U_{t−1})}p(U_t | U_{t−1}) = \int q_{\lambda_t}(U_{t−1})p(U_t | U_{t−1})dU_{t−1}$$

$$\propto \int N(U_{t−1}; \mu_{t−1}, \sigma_{t−1}^2)N(U_t; \mu_t, \sigma_t^2)N(U_{t−1}; 0, \sigma_0^2)du_{t−1}dU_{t−1}$$

$$\propto \int \exp \left[ -\frac{1}{2} \left( \frac{(U_{t−1} - \mu_{t−1})^2}{\sigma_{t−1}^2} + \frac{(U_t - \mu_t)^2}{\sigma_t^2} + \frac{U_{t−1}^2}{\sigma_0^2} \right) \right] dU_{t−1},$$

(30)

where let $\sigma_{t−1}^2$ and $\mu_t−1$ be abbreviated for the variances and means for all users’ embeddings $U_{t−1}$, respectively, and drop the constant prefactors and use a notation that is suitable for scalar values. In reality, $\sigma_{t−1}^2$ 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 $U_{t−1}$ out of it, and then (30) becomes:

$$\propto \exp \left[ -\frac{1}{2} \frac{\mu_t^2}{\sigma_t^2} \right] \times \exp \left[ -\frac{1}{2} \frac{1}{\sigma_{t−1}^2} \frac{1}{\sigma_0^2} \right] \times \int \exp \left[ -\frac{1}{2} \left( \frac{U_{t−1}^2}{\sigma_{t−1}^2} + \frac{(U_t - \mu_t)^2}{\sigma_t^2} \right) \right] dU_{t−1},$$

(31)

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

$$\propto \exp \left[ -\frac{1}{2} \frac{1}{\sigma_{t−1}^2} \frac{1}{\sigma_0^2} \right] \times \exp \left[ \frac{1}{2} \left( \frac{1}{\sigma_{t−1}^2} + \frac{1}{\sigma_0^2} \right) \right] \times \int \exp \left[ -\frac{1}{2} \left( \frac{1}{\sigma_{t−1}^2} + \frac{1}{\sigma_0^2} \right) (U_{t−1} - A)^2 \right] dU_{t−1},$$

(32)
which results in:

where we let $U_t$ abbreviate for the means and variances for all users’ embeddings $U_t$, respectively, and let:

$$
\frac{1}{\tilde{q}_t^2} = \frac{1}{\sigma_0^2} + \frac{1}{\sigma_1^2 - 1} + \frac{1}{\sigma_2^2 - 1}, \quad \text{and} \quad \frac{1}{\tilde{q}_t^2} = \frac{\mu_{t-1}}{\sigma_0^2 - 1} + \frac{1}{\sigma_1 - 1} + \frac{1}{\sigma_2 - 1},
$$

which results in:

$$
\tilde{y}_t = \frac{q_t^2}{\tilde{q}_t^2} \left( \sigma_1^2 - 1 \right)^{-1} \mu_{t-1},
$$

$$
\tilde{q}_t^2 = \left( \sigma_1^2 - 1 \right)^{-1} + \left( 1/\tilde{q}_t^2 \right)^{-1}.
$$

REFERENCES


