Dynamic Embeddings for Language Evolution

Maja Rudolph
Columbia University
maja@cs.columbia.edu

David Blei
Columbia University
david.blei@columbia.edu

ABSTRACT

Word embeddings are a powerful approach for unsupervised analysis of language. Recently, Rudolph et al. [35] developed exponential family embeddings, which cast word embeddings in a probabilistic framework. Here, we develop dynamic embeddings, building on exponential family embeddings to capture how the meanings of words change over time. We use dynamic embeddings to analyze three large collections of historical texts: the U.S. Senate speeches from 1858 to 2009, the history of computer science ACM abstracts from 1951 to 2014, and machine learning papers on the ArXiv from 2007 to 2015. We find dynamic embeddings provide better fits than classical embeddings and capture interesting patterns about how language changes.

KEYWORDS

word embeddings, exponential family embeddings, probabilistic modeling, dynamic modeling, semantic change

ACM Reference Format:
Maja Rudolph and David Blei. 2018. Dynamic Embeddings for Language Evolution. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3178876.3185999

1 INTRODUCTION

Word embeddings are a collection of unsupervised learning methods for capturing latent semantic structure in language. Embedding methods analyze text data to learn distributed representations of the vocabulary. The learned representations are then useful for reasoning about word usage and meaning [16, 36]. With large data sets and approaches from neural networks, word embeddings have become an important tool for analyzing language [3, 6, 21, 24–26, 33, 42].

Recently, Rudolph et al. [35] developed *exponential family embeddings.* Exponential family embeddings distill the key assumptions of an embedding problem, generalize them to many types of data, and cast the distributed representations as latent variables in a probabilistic model. They encompass many existing methods for embeddings and open the door to bringing expressive probabilistic modeling [7, 32] to the task of learning distributed representations.

Here we use exponential family embeddings to develop *dynamic word embeddings*, a method for learning distributed representations that change over time. Dynamic embeddings analyze long-running texts, e.g., documents that span many years, where the way words are used changes over the course of the collection. The goal of dynamic embeddings is to characterize those changes.

Figure 1 illustrates the approach. It shows the changing representation of INTELLIGENCE in two corpora, the collection of computer science abstracts from the ACM 1951–2014 and the U.S. Senate speeches 1858–2009. On the y-axis is "meaning," a proxy for the dynamic representation of the word; in both corpora, its representation changes dramatically over the years. To understand where it is located, the plots also show similar words (according to their changing representations) at various points. Loosely, in the ACM corpus INTELLIGENCE changes from government intelligence to cognitive intelligence to artificial intelligence; in the Congressional record INTELLIGENCE changes from psychological intelligence to government intelligence. Section 3 gives other examples from these corpora, such as for the terms IRAQ, DATA, and COMPUTER.

In more detail, a word embedding uses representation vectors to parameterize the conditional probabilities of words in the context of other words. Dynamic embeddings divide the documents into time slices, e.g., one per year, and cast the embedding vector as a latent variable that drifts via a Gaussian random walk. When fit to data, the dynamic embeddings capture how the representation of each word drifts from slice to slice.

Section 2 describes dynamic embeddings and how to fit them. Section 3 studies this approach on three datasets: 9 years of ArXiv machine learning papers (2007–2015), 64 years of computer science abstracts (1951–2014), and 151 years of U.S. Senate speeches (1858–2009). Dynamic embeddings give better predictive performance than existing approaches and provide an interesting exploratory window into how language changes.

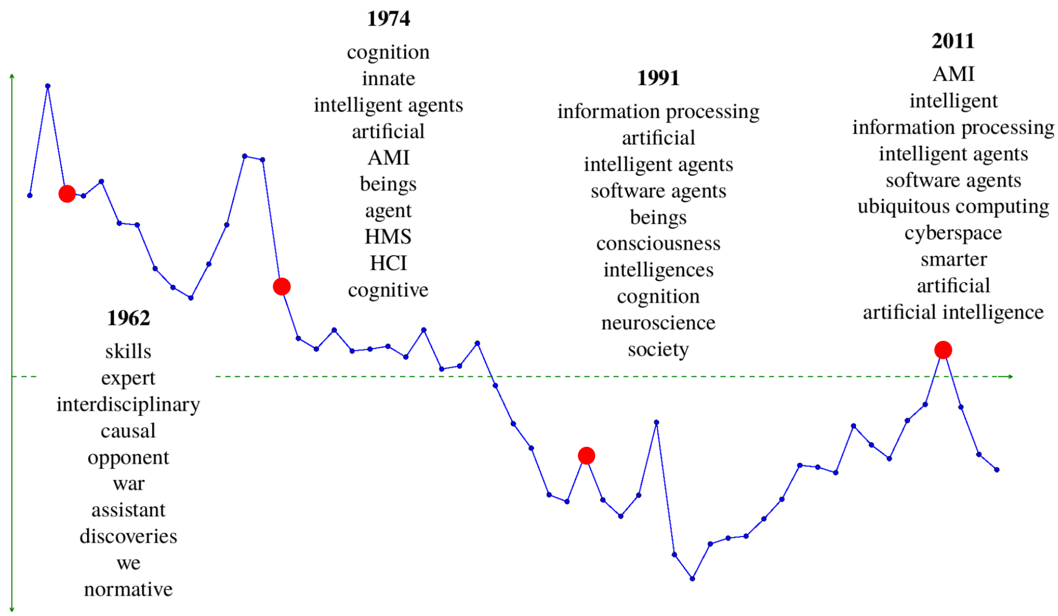
Related work. Language is known to evolve [1, 19] and there have been several lines of research around capturing semantic shifts. Mihalcea and Nastase [23] and Tang et al. [38] detect semantic changes of words using features such as part-of-speech tags and entropy. Sagi et al. [37] and Basile et al. [5] employ latent semantic analysis and temporal semantic indexing for quantifying changes in meaning.

Most closely related to our work are methods for dynamic embeddings [15, 18, 20]. These methods train a separate embedding for each time slice of the data. While interesting, this requires enough data in each time slice such that a high quality embedding can be trained for each. Further, because each time slice is trained independently, the dimensions of the embeddings are not comparable across time; they must use initialization [18] or ad-hoc alignment techniques [15, 20, 48] to stitch them together.

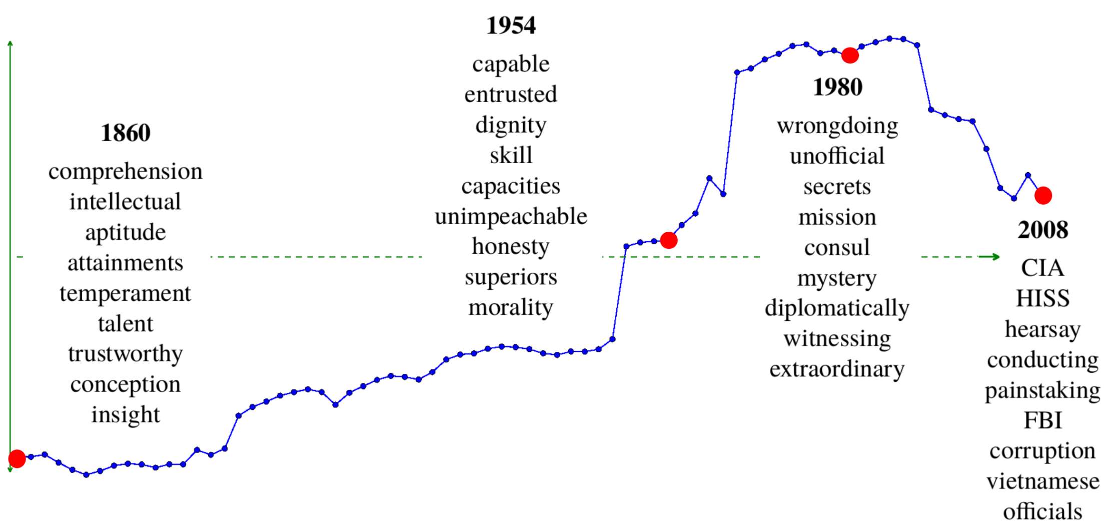
In contrast, the representations of our model for dynamic embeddings are sequential latent variables. This naturally accommodates time slices with sparse data and assures that the dimensions of the embeddings are connected across time. In Section 3, we show that our method provides quantitative improvements over methods that fit each slice independently.

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3185999



(a) INTELLIGENCE in ACM abstracts (1951–2014)



(b) INTELLIGENCE in U.S. Senate speeches (1858–2009)

Figure 1: The dynamic embedding of INTELLIGENCE reveals how the term's usage changes over the years in a historic corpus of ACM abstracts (a) and U.S. Senate speeches (b). The y-axis is "meaning," a one dimensional projection of the embedding vectors. For selected years, we list words with similar dynamic embeddings.

We note that two models similar to ours have been developed independently [4, 46]. Bamler and Mandt [4] model both the embeddings and the context vectors using an Uhlenbeck-Ornstein process [41]. Yao et al. [46] factorize the pointwise mutual information (PMI) matrix at different time slices. Their regularization also resembles an Uhlenbeck-Ornstein process. Both employ the matrix factorization perspective of embeddings [21], while our work builds on exponential family embeddings [35], which generalize embeddings using exponential families. A related perspective is given by Cotterell et al. [10] who show that exponential family PCA can generalize embeddings to higher order tensors.

Another area of related work is dynamic topic models, which are also used to analyze text data over time [8, 12, 13, 27, 28, 43–45, 47]. This class of models describes documents in terms of topics, which are distributions over the vocabulary, and then allows the topics to change. As in dynamic embeddings, some dynamic topic models use a Gaussian random walk to capture drift in the underlying language model; for example, see Blei and Lafferty [8], Wang et al. [43], Gerrish and Blei [13] and Frermann and Lapata [12].

Though topic models and word embeddings are related, they are ultimately different approaches to language analysis. Topic models capture co-occurrence of words at the document level and focus on heterogeneity, i.e., that a document can exhibit multiple topics [9]. Word embeddings capture co-occurrence in terms of proximity in the text, usually focusing on small neighborhoods around each word [26]. Combining dynamic topic models and dynamic word embeddings is an area for future study.

2 DYNAMIC EMBEDDINGS

We develop dynamic embeddings (D-EMB), a type of exponential family embedding (EFE) [35] that captures sequential changes in the representation of the data. We focus on text data and the Bernoulli embedding model. In this section, we review Bernoulli embeddings for text and show how to include dynamics into the model. We then derive the objective function for dynamic embeddings and develop stochastic gradients to optimize it on large collections of text.

Bernoulli embeddings for text. An EFE is a conditional model [2]. It has three ingredients: The *context*, the *conditional distribution* of each data point, and the *parameter sharing structure*.

In an EFE for text, the data is a corpus of text, a sequence of words $(x_1, \dots, x_N)$ from a vocabulary of size V. Each word $x_i \in \{0,1\}^V$ is an indicator vector (also called a "one-hot" vector). It has one nonzero entry at v, where v is the vocabulary term at position i.

In an EFE model, each data point has a *context*. In text, the context of each word is its neighborhood; Each word is modelled conditionally on the words that come before and after. Typical context sizes range between 2 and 10 words and are set in advance.

Here, we will build on Bernoulli embeddings, which provide a conditional model for the individual entries of the indicator vectors $x_{iv} \in \{0,1\}$. Let c_i be the set of positions in the neighborhood of position i and let $\mathbf{x}_{c_i}$ denote the collection of data points indexed by those positions. The conditional distribution of x_{iv} is

$$x_{iv} | \mathbf{x}_{c_i} \sim \text{Bern}(p_{iv}), \tag{1}$$

where $p_{iv} \in (0,1)$ is the Bernoulli probability.[1]

Bernoulli embeddings specify the natural parameter of this distribution, the log odds $\eta_{iv} = \log \frac{p_{iv}}{1-p_{iv}}$, as a function of the representation of term v and the terms in the context of position i. Specifically, each index (i, v) in the data is associated with two parameter vectors, the *embedding vector* $\rho_v \in \mathbb{R}^K$ and the *context vector* $\alpha_v \in \mathbb{R}^K$. Together, the embedding vectors and context vectors form the natural parameter of the Bernoulli. It is

$$\eta_{iv} = \rho_v^\top \left(\textstyle\sum_{j \in c_i} \sum_{v'} \alpha_{v'} x_{jv'} \right). \tag{2}$$

This is the inner product between the embedding ρ_v and the context vectors of the words that surround position i. (Because x_j is an indicator vector, the sum over the vocabulary selects the appropriate context vector α at position j.) The goal is to learn the embeddings and context vectors.

The index on the parameters does not depend on position i, but only on term v; the embeddings are shared across all positions in the text. This is what Rudolph et al. [35] call the *parameter sharing structure*. It ensures, for example, that the embedding vector for INTELLIGENCE is the same wherever it appears. (Dynamic embeddings partially relax this restriction.)

Finally, Rudolph et al. [35] regularize the Bernoulli embedding by placing priors on the embedding and context vectors. They use Gaussian priors with diagonal covariance, i.e., ℓ_2 regularization. Without the regularization, fitting a Bernoulli embedding closely relates to other embedding techniques such as CBOW [24] and negative sampling [25]. But the probabilistic perspective of EFE —and in particular the priors and the parameter sharing—allows us to extend this setting to capture dynamics.

Dynamic Bernoulli embeddings (D-EMB) extend Bernoulli embeddings to text data over time. Each observation x_{iv} is associated with a time slice t_i, such as the year of the observation. Context vectors are shared across all positions in the text but the embedding vectors are only shared within a time slice. Thus dynamic embeddings posit a sequence of embeddings for each term $\rho_v^{(t)} \in \mathbb{R}^K$ while the static context vectors help ensure that consecutive embeddings are grounded in the same semantic space.

The natural parameter of the conditional likelihood is similar to Equation (2) but with the embedding vector ρ_v replaced by the per-time-slice embedding vector $\rho_v^{(t_i)}$,

$$\eta_{iv} = \rho_v^{(t_i)\top} \left(\textstyle\sum_{j \in c_j} \sum_{v'} \alpha_{v'} x_{jv'} \right). \tag{3}$$

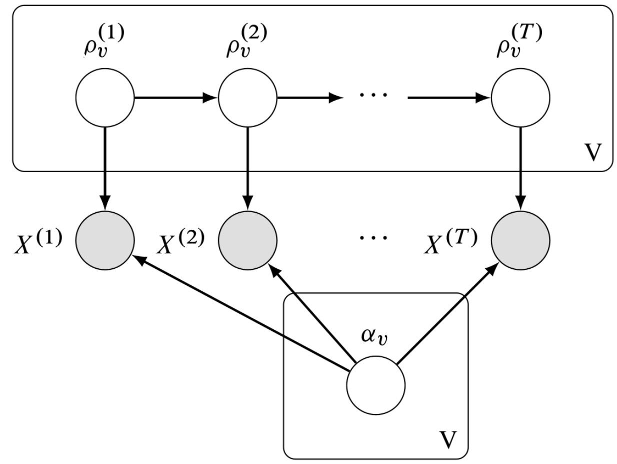


Figure 2: Graphical representation of a D-EMB for text data in T time slices, $X^{(1)}, \cdots, X^{(T)}$. The embedding vectors ρ_v of each term evolve over time. The context vectors are shared across all time slices.

[1]Multinomial embeddings [35] model each indicator vector x_i with a categorical conditional distribution, but this requires expensive normalization in form of a softmax function. For computational efficiency, one can replace the softmax with the hierarchical softmax [25, 29, 31] or employ approaches related to noise contrastive estimation [14, 30]. Bernoulli embeddings relax the one-hot constraint of x_i, and work well in practice; they relate to the negative sampling [25].

Finally, dynamic embeddings use a Gaussian random walk as a prior on the embedding vectors,

$$\alpha_v, \rho_v^{(0)} \sim \mathcal{N}(0, \lambda_0^{-1} I) \tag{4}$$

$$\rho_v^{(t)} \sim \mathcal{N}(\rho_v^{(t-1)}, \lambda^{-1} I). \tag{5}$$

Given data, this leads to smoothly changing estimates of each term's embedding.[2]

Figure 2 gives the graphical model for dynamic embeddings. Dynamic embeddings are a conditionally specified model, which in general are not guaranteed to imply a consistent joint distribution. But dynamic Bernoulli embeddings model binary data, and thus a joint exists [2].

Fitting dynamic embeddings. Calculating the joint is computationally intractable. Rather, we fit dynamic embeddings with the *pseudo log likelihood*, the sum of the log conditionals, a commonly used objective for conditional models [2].

In detail, we regularize the pseudo log likelihood with the log priors and then maximize to obtain a pseudo MAP estimate. For dynamic Bernoulli embeddings, this objective is the sum of the log priors and the conditional log likelihoods of the data x_{iv}.

We divide the data likelihood into two parts, the contribution of nonzero data entries $\mathcal{L}_{\text{pos}}$ and of zero data entries $\mathcal{L}_{\text{neg}}$,

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{neg}} + \mathcal{L}_{\text{prior}}. \tag{6}$$

The likelihoods are

$$\mathcal{L}_{\text{pos}} = \sum_{i=1}^{N} \sum_{v=1}^{V} x_{iv} \log \sigma(\eta_{iv})$$

$$\mathcal{L}_{\text{neg}} = \sum_{i=1}^{N} \sum_{v=1}^{V} (1 - x_{iv}) \log(1 - \sigma(\eta_{iv})),$$

where $\sigma(\cdot)$ is the sigmoid, which maps natural parameters to probabilities. The prior is

$$\mathcal{L}_{\text{prior}} = \log p(\boldsymbol{\alpha}) + \log p(\boldsymbol{\rho}),$$

where

$$\log p(\boldsymbol{\alpha}) = -\frac{\lambda_0}{2} \sum_v ||\alpha_v||^2$$

$$\log p(\boldsymbol{\rho}) = -\frac{\lambda_0}{2} \sum_v ||\rho_v^{(0)}||^2 - \frac{\lambda}{2} \sum_{v,t} ||\rho_v^{(t)} - \rho_v^{(t-1)}||^2.$$

The parameters $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$ appear in the natural parameters η_{iv} of Equations (2) and (3) and in the log prior. The random walk prior penalizes consecutive word vectors $\rho_v^{(t-1)}$ and $\rho_v^{(t)}$ for drifting too far apart. It prioritizes parameter settings for which the norm of their difference is small.

The most expensive term in the objective is $\mathcal{L}_{\text{neg}}$, the contribution of the zeroes to the conditional log likelihood. The objective is cheaper if we subsample the zeros. Rather than summing over all words which are not at position i, we sum over a subset of n negative samples $\mathcal{S}_i$ drawn at random. Mikolov et al. [25] call this negative sampling and recommend sampling from $\hat{p}$, the unigram distribution raised to the power of 0.75.

With negative sampling, we redefine $\mathcal{L}_{\text{neg}}$ as

$$\mathcal{L}_{\text{neg}} = \sum_{i=1}^{N} \sum_{v \in \mathcal{S}_i} \log(1 - \sigma(\eta_{iv})). \tag{7}$$

This sum has fewer terms and reduces the contribution of the zeros to the objective. In a sense, this incurs a bias—the expectation with respect to the negative samples is not equal to the original objective—but "downweighting the zeros" can improve prediction accuracy [17, 22] and leads to significant computational gains.

Algorithm 1: SGD for dynamic embeddings.

Input: T time slices of text data $X^{(t)}$ of size m_t respectively. Context size c, size of embedding K, number of negative samples n, number of minibatch fractions m, initial learning rate η, precision λ, vocabulary size V, smoothed unigram distribution $\hat{p}$.

for $v = 1$ **to** V **do**
- Initialize entries of α_v and entries of $\rho_v^{(t)}$ (using draws from a normal distribution with zero mean and standard deviation 0.01).

end for

for number of passes over the data **do**
- **for** number of minibatch fractions m **do**
  - **for** $t = 1$ **to** T **do**
    - Sample minibatch of m_t/m consecutive words $\{x_1^{(t)}, \cdots, x_{m_t/m}^{(t)}\}$ from each time slice $X^{(t)}$, and construct
    $$C_i^{(t)} = \sum_{j \in c_i} \sum_{v'=1}^{V} \alpha_{v'} x_{jv'}.$$
    - For each text position in the minibatch, draw a set $\mathcal{S}_i^{(t)}$ of n neg. samples from $\hat{p}$.
  - **end for**
  - update the parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\rho}\}$ by ascending the stochastic gradient
  $$\nabla_{\boldsymbol{\theta}} \Bigg\{ \sum_{t=1}^{T} m \sum_{i=1}^{m_t/m} \Bigg(\sum_{v=1}^{V} x_{iv}^{(t)} \log \sigma(\rho_v^{(t)\top} C_i^{(t)}) + \sum_{x_j \in \mathcal{S}_i^{(t)}} \sum_{v=1}^{V} (1 - x_{jv}) \log(1 - \sigma(\rho_v^{(t)\top} C_i^{(t)})) \Bigg) - \frac{\lambda_0}{2} \sum_v ||\alpha_v||^2 - \frac{\lambda_0}{2} \sum_v ||\rho_v^{(0)}||^2 - \frac{\lambda}{2} \sum_{v,t} ||\rho_v^{(t)} - \rho_v^{(t-1)}||^2 \Bigg\}.$$
- **end for**

end for

We use Adagrad [11] to set rate η.

[2]Because $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ appear only as inner products in Equation (2), we can capture that their interactions change over time even by placing temporal dynamics on the embeddings $\boldsymbol{\rho}$ only. Exploring dynamics in $\boldsymbol{\alpha}$ is a subject for future study.

Table 1: Time range and size of the three corpora analyzed in Section 3.

	ArXiv ML 2007 – 2015	ACM 1951 – 2014	Senate speeches 1858 – 2009
slices	9	64	76
slice size	1 year	1 year	2 years
vocab size	50k	25k	25k
words	6.5M	21.6M	13.7M

We fit the objective (Equation (6) with Equation (7)) using stochastic gradients [34] and with adaptive learning rates [11]. The negative samples are resampled at each gradient step. Pseudo code is in Algorithm 1. To avoid deriving the gradients of Equation (6), we implemented the algorithm in Edward [40]. Edward is based on tensorflow [39] and employs automatic differentiation.[3]

3 EMPIRICAL STUDY

This empirical study has two parts. In a quantitative evaluation we benchmark dynamic embeddings against static embeddings [24, 25, 35]. We found that dynamic embeddings improve over static embeddings in terms of the conditional likelihood of held-out predictions. Further, dynamic embeddings perform better than embeddings trained on the individual time slices [15]. In a qualitative evaluation we use fitted dynamic embeddings to extract which word vectors change most and we visualize their dynamics. Dynamic embeddings provide a new window into how language changes.

3.1 Data

We study three datasets. Their details are summarized in Table 1.

Machine Learning Papers (2007 - 2015). This dataset contains the full text from all machine learning papers (tagged "stat.ML") published on the ArXiv between April 2007 and June 2015. It spans 9 years and we treat each year as a time slice. The number of ArXiv papers about machine learning has increased over the years. There were 101 papers in 2007, while there were 1, 573 papers in 2014.

Computer Science Abstracts (1951 - 2014). This dataset contains abstracts of computer science papers published by the Association of Computing Machinery (ACM) from 1951 to 2014. We treat each year as a time slice and here too, the amount of data increases over the years. For 1953, there are only around 10 abstracts and their combined length is only 471 words; the total length of the abstracts from 2009 is over 2M.

Senate Speeches (1858 - 2009). This dataset contains all U.S. Senate speeches from 1858 to mid 2009. Here we treat every 2 years as a time slice. Unlike the other datasets, this is a transcript of spoken language. It contains many infrequent words that occur only in a few of the time slices.

Preprocessing. We convert the text to lowercase and strip it of all punctuation. Frequent n-grams such as UNITED STATES are treated as a single term. The vocabulary consists of the 25, 000 most frequent terms and all words which are not in the vocabulary are removed. As in [25], we additionally remove each word with probability $p = 1 - \sqrt{(\frac{10^{-5}}{f_i})}$ where f_i is the frequency of the word. This effectively downsamples especially the frequent words and speeds up training.

3.2 Quantitative evaluation

We compare dynamic embeddings (D-EMB) to time-binned embeddings (T-EMB) [15] and static embeddings (S-EMB) [35]. There are many embedding techniques, without dynamics, that enjoy comparable performance. For the S-EMB, we study Bernoulli embeddings [35], which are similar to continuous bag-of-words (CBOW) with negative sampling [24, 25]. For time-binned embeddings, Hamilton et al. [15] train a separate embedding on each time slice.

Evaluation metric. From each time slice 80% of the words are used for training. A random subsample of 10% of the words is held out for validation and another 10% for testing. We evaluate models by held-out Bernoulli probability. Given a model, each held-out position (validation or testing) is associated with a Bernoulli probability for each vocabulary term. At that position, a better model assigns higher probability to the observed word and lower probability to the others. This metric is straightforward because the competing methods all produce Bernoulli conditional likelihoods (Equation (1)). Since we hold out chunks of consecutive words usually both a word and its context are held out. All methods require the words in the context to compute the conditional likelihoods.

We report $\mathcal{L}_{\text{eval}} = \mathcal{L}_{\text{pos}} + \frac{1}{n}\mathcal{L}_{\text{neg}}$, where n is the number of negative samples. Renormalizing with n assures that the metric is balanced. It equally weights the positive and negative examples. To make results comparable, all methods are trained with the same number of negative samples.

Model training and hyperparameters. Each method takes a maximum of 10 passes over the data. (The corresponding number of stochastic gradient steps depends on the size of the minibatches.) The parameters of S-EMB are initialized randomly. We initialize both D-EMB and T-EMB from a fit of S-EMB which has been trained from one pass, and then train for 9 additional passes.

We set the dimension of the embeddings to 100 and the number of negative samples to 20. We study two context sizes, 2 and 8.

Other parameters are set by validation error. All methods use validation error to set the initial learning rate η and minibatch sizes m. The model selects $\eta \in [0.01, 0.1, 1, 10]$ and $m \in [0.001N, 0.0001N, 0.00001N]$, where N is the size of training data. The only parameter specific to D-EMB is the precision of the random drift. To have one less hyper parameter to tune, we fix the precision on the context vectors and the initial dynamic embeddings to $\lambda_0 = \lambda/1000$, a constant multiple of the precision on the dynamic embeddings. We choose $\lambda \in [1, 10]$ by validation error.

Results. We train each model on each training set and use each validation set for selecting parameters like the minibatch size and the learning rate. Table 2 reports the results on the test set. Dynamic embeddings consistently have higher held-out likelihood.

3.3 Qualitative exploration

There are different reasons for a word's usage to change over the course of a collection. Words can become obsolete or obtain a new meaning. As society makes progress and words are used to describe that progress, that progress also gradually changes the meaning of

[3]Code is available at http://github.com/mariru/dynamic_bernoulli_embeddings

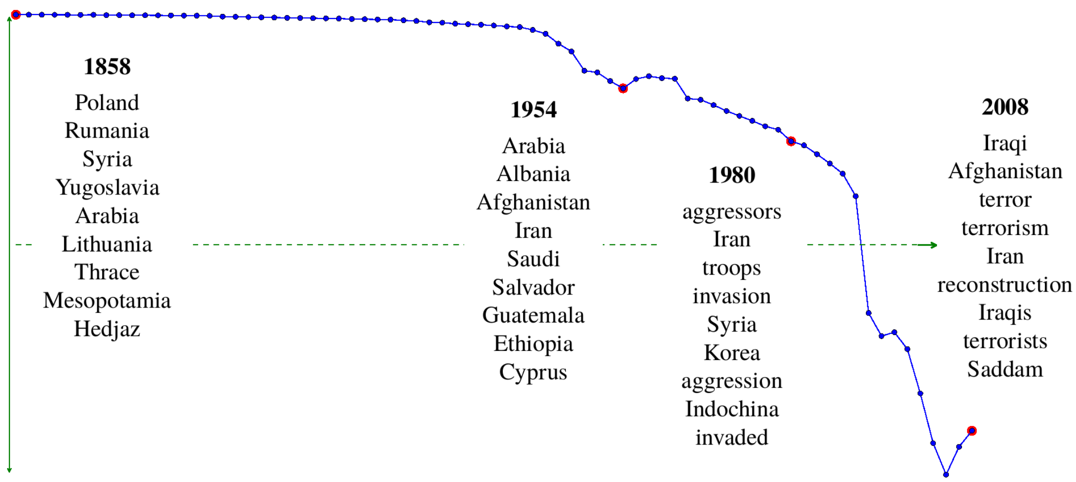


Figure 3: The dynamic embedding captures how the usage of the word IRAQ changes over the years (1858-2009). The x-axis is time and the y-axis is a one-dimensional projection of the embeddings using principal component analysis (PCA). We include the embedding neighborhoods for IRAQ in the years 1858, 1954, 1980 and 2008.

Table 2: Dynamic embeddings (D-EMB) consistently achieve highest held-out $\mathcal{L}_{\text{eval}}$. We compare to static embeddings (S-EMB) [25, 35], and time-binned embeddings (T-EMB) [15]. The largest standard error on the held-out predictions is 0.002 which means all reported results are significant.

ArXiv ML	context size 2	context size 8
S-EMB [35]	−2.77	−2.54
T-EMB [15]	−2.97	−2.81
D-EMB [this paper]	**−2.58**	**−2.44**

Senate speeches	context size 2	context size 8
S-EMB [35]	−2.41	−2.29
T-EMB [15]	−2.44	−2.46
D-EMB [this paper]	**−2.33**	**−2.28**

ACM	context size 2	context size 8
S-EMB [35]	−2.48	−2.30
T-EMB [15]	−2.55	−2.42
D-EMB [this paper]	**−2.45**	**−2.27**

words. A word might also have multiple alternative meanings. Over time, one meaning might become more relevant than the other. We now show how to use dynamic embeddings to explore text data and to discover such changes in the usage of words.

A word's *embedding neighborhood* helps visualize its usage and how it changes over time. It is simply a list of other words with similar usage. For a given query word (e.g., COMPUTER) we take its index v and select the top ten words according to

$$\text{neighborhood}(v, t) = \text{argsort}_w \left(\frac{\text{sign}(\rho_v^{(t)})^\top \rho_w^{(t)}}{||\rho_v^{(t)}|| \cdot ||\rho_w^{(t)}||} \right). \tag{8}$$

As an example, we fit a dynamic embedding fit to the Senate speeches. Table 3 gives the embedding neighborhoods of COMPUTER for the years 1858 and 1986. Its usage changed dramatically over the years. In 1858, a COMPUTER was a profession, a person who was hired to compute things. Now the profession is obsolete; COMPUTER refers to the electronic device.

Table 3 provides another example, BUSH. In 1858 this word always referred to the plant. A BUSH still is a plant, but in the 1990's, it usually refers to a politician. Unlike COMPUTER, where the embedding neighborhoods reveal two mutually exclusive meanings, the embedding neighborhoods of BUSH reflect which meaning is more prevalent in a given period.

A final example in Table 3 is the word DATA, from a D-EMB of the ACM abstracts. The evolution of the embedding neighborhoods of DATA reflects it changes meaning in the computer science literature.

Finding changing words with absolute drift. We have highlighted example words whose usage changes. However, not all words have changing usage. We now define a metric to discover which words change most.

One way to find words that change is to use *absolute drift*. For word v, it is

$$\text{drift}(v) = ||\rho_v^{(T)} - \rho_v^{(0)}||. \tag{9}$$

This is the Euclidean distance between the word's embedding at the last and at the first time slice.

In the Senate speeches, Table 4 shows the 16 words that have largest absolute drift. The word IRAQ has largest drift. Figure 3 highlights IRAQ's embedding neighborhood in four time slices: 1858,

Table 3: Embedding neighborhoods (Equation (8)) reveal how the usage of a word changes over time. The embedding neighborhoods of COMPUTER and BUSH were computed from a dynamic embedding fitted to Congress speeches (1858-2009). COMPUTER used to be a profession but today it is used to refer to the electronic device. The word BUSH is a plant but eventually in congress BUSH is used to refer to the political figures. The embedding neighborhood of DATA comes from a dynamic embedding fitted to ACM abstracts (1951-2014).

COMPUTER (Senate)		BUSH (Senate)	
1858	**1986**	**1858**	**1990**
draftsman	software	barberry	cheney
draftsmen	computers	rust	nonsense
copyist	copyright	bushes	nixon
photographer	technological	borer	reagan
computers	innovation	eradication	george
copyists	mechanical	grasshoppers	headed
janitor	hardware	cancer	criticized
accountant	technologies	tick	clinton

DATA (ACM)			
1961	**1969**	**1991**	**2014**
directories	repositories	voluminous	data streams
files	voluminous	raw data	voluminous
bibliographic	lineage	repositories	raw data
formatted	metadata	data streams	warehouses
retrieval	snapshots	data sources	dws
publishing	data streams	volumes	repositories
archival	raw data	dws	data sources
archives	cleansing	dsms	data mining

Table 4: A list of the top 16 words whose dynamic embedding on Senate speeches changes most. The number represents the absolute drift (Equation (9)). The dynamics of the capitalized words are in Table 5 and discussed in the text.

words with largest drift (Senate)			
IRAQ	3.09	coin	2.39
tax cuts	2.84	social security	2.38
health care	2.62	FINE	2.38
energy	2.55	signal	2.38
medicare	2.55	program	2.36
DISCIPLINE	2.44	moves	2.35
text	2.41	credit	2.34
VALUES	2.40	UNEMPLOYMENT	2.34

1950, 1980, and 2008. At first the neighborhood contains other countries and regions. Later, Arab countries move to the top of the neighborhood, suggesting that the speeches start to use rhetoric more specific to Arab countries. In 1980, Iraq invades Iran and the Iran-Iraq war begins. In these years, words such as TROOPS, and INVASION appear in the embedding neighborhood. Eventually, by 2008, the neighborhood contains TERROR, TERRORISM, and SADDAM.

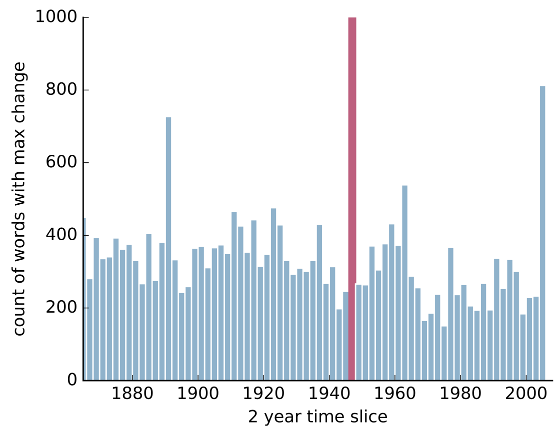


Figure 4: According to D-EMB fitted to the Senate Speeches, most words change most in the 1947-1947 time slice.

Four other words with large drift are DISCIPLINE, VALUES, FINE and UNEMPLOYMENT (Table 4). Table 5 shows their embedding neighborhoods. Of these words, DISCIPLINE, VALUES and, FINE have multiple meanings. Their neighborhoods reflect how the dominant meaning changes over time. For example, VALUES can be either a numerical quantity or can be used to refer to moral values and principles. In contrast, IRAQ and UNEMPLOYMENT are both words which have always had the same definition. Yet, the evolution of their neighborhood captures changes in the way they are used.

Changepoint analysis. We use the fitted dynamic embeddings to find instances in time where a word's usage changes drastically. We make no assumption that a word's meaning makes only a single phase transition [20]. Since in our formulation of D-EMB the context vectors are shared between all time slices, the embeddings are grounded in one semantic space and no postprocessing is needed to align the embeddings. We can directly compute large jumps in word usage on the learned embedding vectors.

For each word, we compute a list of time slices where the word's usage changed most.

$$\text{max change}(v) = \text{argsort}_t \left(\frac{||\rho_v^{(t)} - \rho_v^{(t-1)}||}{\sum_w ||\rho_w^{(t)} - \rho_w^{(t-1)}||} \right). \quad (10)$$

The changes in time slice t are normalized by how much all other words changed within the same time slice. The normalization, makes the *max change* ranking sensitive to time slices in which a word's embedding drifted farthest, compared to how far other words drifted within the time slice.

For example, for the word IRAQ the largest change is in the years 1990-1992. Indeed, that year the Gulf war started. Note that this is consistent with Figure 3 where we see in the one dimensional projection of the trajectory of the embedding a large jump around the year 1990. The trajectory in the Figure captures only the variation in the first principal component, while Equation 10 measures difference of embedding vectors in all the dimensions combined.

Table 5: Embedding neighborhoods extracted from a dynamic embedding fitted to Senate speeches (1858 - 2009). DISCIPLINE, VALUES, FINE, and UNEMPLOYMENT are within the 16 words whose dynamic embedding has largest absolute drift. (Table 4).

DISCIPLINE		VALUES		FINE		UNEMPLOYMENT	
1858	2004	1858	2000	1858	2004	1858	2000
hazing	balanced	fluctuations	sacred	luxurious	punished	unemployed	jobless
westpoint	balancing	value	inalienable	finest	penitentiaries	depression	rate
assaulting	fiscal	currencies	unique	coarse	imprisonment	acute	depression

Table 6: Using dynamic embeddings we can study a social phenomenon of interest. We pick a target word of interest, such as JOBS or PROSTITUTION and create their embedding neighborhoods (Equation (8)).

JOBS		
1858	1938	2008
employment	unemployed	job
unemployed	employment	create
overtime	job	creating

PROSTITUTION					
1858	1930	1945	1962	1988	1990
punishing	punishing	indecent	indecent	intimidation	servitude
immoral	immoral	vile	harassment	prostitution	harassment
illegitimate	bootlegging	immoral	intimidation	counterfeit	intimidation

Next, we examine in which years many words changed most in terms of their usage. In Figure 4 is a histogram of the years in which each word changed the most. For example, IRAQ falls into the 1990-1992 bin, together with almost 300 other words which also had their largest relative change in 1990 - 1992. We can see that the bin with the most words (marked in red) is 1946-1947 which marks the end of the Second World War. Almost 1000 words had their largest relative change in that time slice.

In Table 7 is a list of the 10 words with the largest change in the years 1946-1947. On top of the list is MARSHALL, the middle name of John Marshall Harlan, and John Marshall Harlan II, father and son who both served as U.S. Supreme Court Justices. It is also the last name of George Marshall who became the U.S. Secretary of State in 1947. He conceived and carried out the Marshall plan, an economic relief program to aid post-war Europe. In Table 7 are the embedding neighborhoods for MARSHALL before and after the 1946-1947 time bin. In 1944-1945, MARSHALL is similar to other names with importance to the U.S. judicial system but by 1948-1950 the most similar word is PLAN as the Marshall plan is now frequently discussed in the U.S. Senate Speeches.

Table 7: Dynamic embeddings identify MARSHALL, as the word changing most in 1946-1947. On the left is a list of the top words with largest change in the 1946-1947 time bin (marked red in Figure 4). On the right, are the embedding neighborhoods of MARSHALL before and after the jump.

top change in 1946
1. marshall
2. atlantic
3. korea
4. douglas
5. holland
6. steam
7. security
8. truman
9. plan

MARSHALL (Senate)	
1944	1948
wheaton	plan
taney	satellites
harlan	britain
vs	great britain
gibbons	acheson
mcreynolds	democracies
waite	france
webster	western europe

Dynamic embeddings as a tool to study a text. Our hope is that dynamic embeddings provide a suggestive tool for understanding change in language. For example, researchers interested in UNEMPLOYMENT can complement their investigation by looking at the embedding neighborhood of related words such as EMPLOYMENT, JOBS or LABOR. In Table 6 we list the neighborhoods of JOBS for the years 1858, 1938, and 2008. In 2008 the embedding neighborhood contains words like CREATE and CREATING, suggesting a different outlook on JOBS than in earlier years.

Another interesting example is PROSTITUTION. It used to be IMMORAL and VILE, went to INDECENT, and in modern days it is considered HARASSMENT. We note the word PROSTITUTION is not a frequent word. On average, it is used once per time slice and, in two thirds of the time slices, it is not mentioned at all. Yet, the model is able to learn about PROSTITUTION and the temporal evolution of the embedding neighborhood reveals how over the years a judgemental stance turns into concern over a social issue.

4 SUMMARY

We described dynamic embeddings, distributed representations of words that drift over the course of the collection. Building on Rudolph et al. [35], we formulate word embeddings with conditional probabilistic models and then incorporate dynamics with a Gaussian random walk prior. We fit dynamic embeddings to language data using stochastic optimization.

We used dynamic embeddings to analyze 3 datasets: 8 years of machine learning papers, 63 years of computer science abstracts, and 151 years of U.S. Senate speeches. Dynamic embeddings provide a better fit than static embeddings and other methods that account for time. In addition, dynamic embeddings can help identify interesting ways in which language changes. A word's meaning can change (e.g., COMPUTER); its dominant meaning can change (e.g., VALUES); or its related subject matter can change (e.g., IRAQ).

ACKNOWLEDGEMENTS

We thank Francisco Ruiz and Liping Liu for discussion and helpful suggestions, Elliot Ash and Suresh Naidu for access to the Congress speeches, and Aaron Plasek and Matthew Jones for access to the ACM abstracts. This work is supported by ONR N00014-11-1-0651, DARPA PPAML FA8750-14-2-0009, the Alfred P. Sloan Foundation, and the John Simon Guggenheim Foundation.

REFERENCES

[1] Jean Aitchison. 2001. *Language change: progress or decay?* Cambridge University Press.

[2] Barry C Arnold, Enrique Castillo, Jose Maria Sarabia, et al. 2001. Conditionally specified distributions: an introduction (with comments and a rejoinder by the authors). *Statist. Sci.* 16, 3 (2001), 249–274.

[3] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2015. RAND-WALK: A latent variable model approach to word embeddings. *arXiv preprint arXiv:1502.03520* (2015).

[4] Robert Bamler and Stephan Mandt. 2017. Dynamic Word Embeddings via Skip-gram Filtering. *arXiv preprint arXiv:1702.08359* (2017).

[5] Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. Analysing word meaning over time by exploiting temporal random indexing. In *First Italian Conference on Computational Linguistics CLiC-it.*

[6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[7] Christopher M Bishop. 2006. Machine learning and pattern recognition. *Information Science and Statistics. Springer, Heidelberg* (2006).

[8] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning.* ACM, 113–120.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[10] Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. 2017. Explaining and Generalizing Skip-Gram through Exponential Family Principal Component Analysis. *EACL 2017* (2017), 175.

[11] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

[12] Lea Frermann and Mirella Lapata. 2016. A Bayesian Model of Diachronic Meaning Change. *Transactions of the Association for Computational Linguistics* 4 (2016), 31–45.

[13] S. Gerrish and D. Blei. 2010. A Language-based Approach to Measuring Scholarly Impact. In *International Conference on Machine Learning.*

[14] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS.*

[15] William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. *arXiv preprint arXiv:1605.09096* (2016).

[16] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[17] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.* Ieee, 263–272.

[18] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515* (2014).

[19] Simon Kirby, Mike Dowman, and Thomas L Griffiths. 2007. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences* 104, 12 (2007), 5241–5245.

[20] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web.* ACM, 625–635.

[21] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Neural Information Processing Systems.* 2177–2185.

[22] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 951–961.

[23] Rada Mihalcea and Vivi Nastase. 2012. Word epoch disambiguation: Finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2.* Association for Computational Linguistics, 259–263.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR Workshop Proceedings. arXiv:1301.3781* (2013).

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems.* 3111–3119.

[26] Tomas Mikolov, Wen-T au Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations.. In *HLT-NAACL.* 746–751.

[27] Sunny Mitra, Ritwik Mitra, Suman Kalyan Maity, Martin Riedl, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2015. An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering* 21, 05 (2015), 773–798.

[28] Sunny Mitra, Ritwik Mitra, Martin Riedl, Chris Biemann, Animesh Mukherjee, and Pawan Goyal. 2014. That's sick dude!: Automatic identification of word sense change across different timescales. *arXiv preprint arXiv:1405.4392* (2014).

[29] Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems.* 1081–1088.

[30] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Neural Information Processing Systems.* 2265–2273.

[31] Frederic Morin and Yoshua Bengio. 2005. Hierarchical Probabilistic Neural Network Language Model.. In *Aistats*, Vol. 5. Citeseer, 246–252.

[32] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective.* MIT press.

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *Conference on Empirical Methods on Natural Language Processing*, Vol. 14. 1532–1543.

[34] Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics* (1951), 400–407.

[35] Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. 2016. Exponential Family Embeddings. In *Advances in Neural Information Processing Systems.* 478–486.

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (1986), 9.

[37] Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2011. Tracing semantic change with latent semantic analysis. *Current methods in historical semantics* (2011), 161–183.

[38] Xuri Tang, Weiguang Qu, and Xiaohe Chen. 2016. Semantic change computation: A successive approach. *World Wide Web* 19, 3 (2016), 375–415.

[39] Tensorflow Team. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.

[40] Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. 2016. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787* (2016).

[41] George E Uhlenbeck and Leonard S Ornstein. 1930. On the theory of the Brownian motion. *Physical review* 36, 5 (1930), 823.

[42] Luke Vilnis and Andrew McCallum. 2015. Word representations via Gaussian embedding. In *International Conference on Learning Representations.*

[43] C. Wang, D. Blei, and D. Heckerman. 2008. Continuous Time Dynamic Topic Models. In *Uncertainty in Artificial Intelligence (UAI).*

[44] Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 424–433.

[45] Derry Tanti Wijaya and Reyyan Yeniterzi. 2011. Understanding semantic change of words over centuries. In *Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web.* ACM, 35–40.

[46] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2017. Discovery of Evolving Semantics through Dynamic Word Embedding Learning. *arXiv preprint arXiv:1703.00607* (2017).

[47] D. Yogatama, C. Wang, B. Routledge, N. A Smith, and E. Xing. 2014. Dynamic Language Models for Streaming Text. *Transactions of the Association for Computational Linguistics* 2 (2014), 181–192.

[48] Yating Zhang, Adam Jatowt, Sourav S Bhowmick, and Katsumi Tanaka. 2016. The Past is Not a Foreign Country: Detecting Semantically Similar Terms across Time. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2793–2807.