Diffusion-based Temporal Word Embeddings

Ahnaf Farhan, Roberto Camacho Barranco, M. Shahriar Hossain, Monika Akbar
Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968
afarhan@miners.utep.edu, {rcamachobarranco, mhossain, makbar}@utep.edu

Abstract

Semantics in natural language processing is largely dependent on contextual relationships between words and entities in documents. The context of a word may evolve. For example, the word "apple" currently has two contexts – a fruit and a technology company. The changes in the context of entities in biomedical publications can help us understand the evolution of a disease and relevant scientific interventions. In this work, we present a new diffusion-based temporal word embedding model that can capture short and long-term changes in the semantics of biomedical entities. Our model captures how the context of each entity shifts over time. Existing dynamic word embeddings capture semantic evolution at a discrete/granular level, aiming to study how a language developed over a long period. Our approach provides smooth embeddings suitable for studying short as well as long-term changes. For the evaluation of the proposed model, we track the semantic evolution of entities in abstracts of biomedical publications. Our experiments demonstrate the superiority of the proposed model when compared to its state-of-the-art alternatives.

1 Introduction

Word embeddings are low-dimensional vector space models obtained by training a neural network using contextual information from a large text corpus. There are several variants of word embeddings with different features, such as word2vec (Mikolov et al. 2013b,a) and GloVe (Pennington, Socher, and Manning 2014). However, the research on word embeddings to incorporate temporal shifts of contextual meanings of words is still in its infant stage. This paper focuses on generating word embeddings that account for and take advantage of the temporal nature of timestamped scientific documents (e.g., abstracts of biomedical publications.) Our goal is to obtain a low-dimensional **temporal vector space representation** that allows us to study the semantic and contextual evolution of words/entities. Using the word embeddings generated by our framework, we demonstrate the task of tracking the semantic evolution of entities in a corpus of biomedical abstracts.

To generate word embeddings, our framework trains a model using a diffusion-mechanism for evolving concepts within a scientific text corpus (Camacho et al. 2018; Angulo et al. 1980). A concept generally does not spike on a day and disappear immediately. Rather, concepts evolve with context. Existing temporal low-dimensional language representations fail to integrate the concept of temporal diffusion into language models effectively. Moreover, these existing models (Bamler and Mandt 2017; Marina Del Rey 2018; Rudolph and Blei 2018) cannot simultaneously capture both the short-term and long-term drifts in the meaning of words. As a result, sharply trending concepts, such as *COVID-19* (coronavirus disease 2019), cannot be modeled in the embedding space when long-term drifts are considered. On the other hand, long-range effects – such as the change in the meaning of the word *cloud* – are not captured when these algorithms take only short-term drifts into account.

Our approach uses the model for temporal high-dimensional tf-idf representations introduced by (Camacho et al. 2018) to construct a training set. Construction of the training set is a one-time cost. The temporal high-dimensional tf-idf representation (Camacho et al. 2018) is able to capture sudden short-term changes in the corpus. Additionally, it incorporates diffusion into the modeling to some extent by incorporating the time dimension smoothly. The framework presented by (Camacho et al. 2018) generates smooth tf-idf vectors for each word of a corpus at every timestamp, making it suitable for the generation of training data for our proposed model. One of the challenges of (Camacho et al. 2018) is that each word vector has a length equal to the number of documents in the corpus, which is not practical for analyzing a corpus containing thousands of scientific documents. Our goal is to construct a **contextual low-dimensional temporal embedding space** mimicking this high-dimensional representation without losing the essential temporal diffusion information encoded in the vectors. We introduce a neural-network-based framework that generates **temporal word embeddings** while optimizing for multiple key objectives. The temporal tf-idf representation from (Camacho et al. 2018) is used to obtain a baseline *expected cosine distance* (1.0 - cosine similarity) between pairs of word vectors at each timestamp. The *expected cosine distance* is used in the output layer of our proposed neural network. New low-dimensional embedding vectors – driven by a rigorous objective function to smoothly bring contextual entities close to each other – are generated in the hidden layer. The generated low-dimensional vectors are contextual and allow the discovery of latent (transitive) relationships that can't be observed in the temporal tf-idf representation. For example, if words A and B are close to C, we expect words A and B to be close to each other. We further explain the objective function and the neural-network in Section 4.

The experimental results in Section 5 show that the proposed method performs significantly better than the state-of-the-art dynamic embedding models (Rudolph and Blei 2017; Carlo, Bianchi, and Palmonari 2019) in capturing both short-term and long-term changes in word semantics. Results show that our approach improves the continuity between the vectors across different timestamps. As a result, embeddings for different timestamps combine to a homogeneous space, unlike the state-of-the-art models.

2 Related Work

Meanings of words in a language change over time depending on their use (Aitchison 2013; Yule 2017). Temporal syntactic and semantic shifts are called *diachronic changes* (Hamilton, Leskovec, and Jurafsky 2016). Several probabilistic approaches tackle the problem of modeling the temporal evolution of a vocabulary by converting a set of timestamped documents into a latent variable model (Radinsky, Davidovich, and Markovitch 2012; Yogatama et al. 2014; Tang, Qu, and Chen 2013; Naim, Boedihardjo, and Hossain 2017). Other approaches model diachronic changes using Parts of Speech features (Mihalcea and Nastase 2012) or using graphs where the edges between nodes (that represent words) are stronger based on context information (Mitra et al. 2015). However, **tracking semantic evolution** is not possible using these techniques because they do not generate language models.

The state-of-the-art technique for language modeling is word2vec, introduced by Mikolov et al. (Mikolov et al. 2013a,b). This method generates a *static* language model where every word is represented as a vector (also called *embedding*) by training a neural network to mimic the contextual patterns observed in a text corpus. There are several variants of this method which include probabilistic approaches (Barkan 2017) as well as matrix-factorization-based techniques such as GloVe (Pennington, Socher, and Manning 2014). A major challenge with *static* representations is that they do not incorporate any temporal information that can be used for tracking semantic evolution. Our work focuses on incorporating the temporal dimension of text data into text embedding models so that evolution of a vector space over time can be studied.

A proposed solution to tracking semantic evolution is to obtain a *static* representation for each timestamp in a corpus, and then artificially couple these embeddings over time using regression or similar methods (Hamilton, Leskovec, and Jurafsky 2016; Rosin, Adar, and Radinsky 2017; Carlo, Bianchi, and Palmonari 2019). However, this approach has several drawbacks. First, it requires having a significant number of occurrences for all words at all times, which is usually not the case since words can gain popularity or appear at different times. Second, the artificial coupling of embeddings across timestamps can introduce artifacts in the model that may lead to wrong conclusions. A potential solution to the sparsity problem is introduced by Camacho et al. (Camacho et al. 2018), which leverages diffusion theory (Angulo et al. 1980) to generate a robust temporal representation. The technique uses a temporal tf-idf representation in which the model changes size with the number of documents and as a result, is not extensible.

The drawbacks of using *static* word embedding models to generate temporal representations have led to the development of new techniques that can train the embeddings for different timestamps jointly. The models use filters or regularization terms to connect the embeddings over time. Yao et al. (Marina Del Rey 2018) propose to generate a co-occurrence-based matrix and factorize it to generate temporal embeddings. The embeddings over timestamps are aligned using a regularization term. Rudolph et al. (Rudolph and Blei 2018) apply Kalman filtering to *exponential family embeddings* to generate temporal representations. Bamler et al. (Bamler and Mandt 2017) use similar filtering but apply it to embeddings using a probabilistic variant of *word2vec*. According to Bamler et al. (Bamler and Mandt 2017), using a probabilistic method makes the model less sensitive to noise. All these methods focus primarily on capturing long-term semantic shifts, while our goal is to be able to capture both long and short-term shifts.

3 Problem Description

In this paper, we focus on timestamped text corpora, such as collections of scientific publications that have publication dates. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ be a corpus of $|\mathcal{D}|$ documents and $\mathcal{W} = \{w_1, w_2, \ldots, w_{|\mathcal{W}|}\}$ be the set of $|\mathcal{W}|$ noun phrases and entities extracted from the text corpus $\mathcal{D}$. We consider each of the noun phrases and entities a word. Each document d contains words from the vocabulary ($\mathcal{W}_d \subset \mathcal{W}$) in the same order as they appear in the original document of d. Every document $d \in \mathcal{D}$ is labeled with a timestamp $t_d \in \mathcal{T}$, where $\mathcal{T}$ is the ordered set of timestamps.

The goal of this paper is to obtain a temporal word embedding model $\mathcal{U}$ from corpus $\mathcal{D}$. Thus, for every timestamp $t \in \mathcal{T}$, we seek to obtain a vector representation u_{it} for every word $w_i \in \mathcal{W}$. The word embeddings $\mathcal{U}$ are represented as a 3-dimensional matrix of size $|\mathcal{W}| \times |\mathcal{T}| \times |u|$ where $|u|$ is a user-given parameter that indicates the size of a vector for a particular word at a particular time. We use the shorthand U_i to describe the 2-dimensional matrix of size $|\mathcal{T}| \times |u|$ that represents word $w_i \in \mathcal{W}$ over time.

4 Methodology

Each subsequent subsections below describes a major component of our objective function to generate diffusion-based temporal word embeddings.

4.1 Training data for our model

We use the temporal tf-idf model (Camacho et al. 2018) to obtain high-dimensional time-reflective text representations of size $|\mathcal{W}| \times |\mathcal{T}| \times |\mathcal{D}|$ for training purpose. The vectors are formed using the temporal tf-idf weights of a word for every document in every timestamp. The temporal tf-idf weight of a word is computed using Eq. (1).

$$\hat{w}(w,d,t_d,t,\varsigma) = \left(\frac{1}{\sqrt{2\pi\varsigma^2}} e^{-\frac{(t_d-t)^2}{2\varsigma^2}}\right) \cdot \left(\frac{(1+\log(f_{w,d})(\log\frac{|\mathcal{D}|}{\lambda_w})}{\sum_{w'\in\mathcal{W}_d}\left((1+\log(f_{w',d})(\log\frac{|\mathcal{D}|}{\lambda_{w'}}))\right)^2}\right), \quad (1)$$

where $\hat{w}$ is the weighted tf-idf value at timestamp t for the word $w \in \mathcal{W}$ in document $d \in \mathcal{D}$, which was published at timestamp t_d. The term $f_{w,d}$ represents the term frequency of word w in document d, λ_w is the number of documents that contain word w, and $\mathcal{W}_d$ is the set of words that appear in document d. The standard deviation of the Gaussian distribution function is represented by ς, and is set by the user.

Next, we compute the cosine distance (1.0–cosine similarity) between every pair of words and store these as a distance matrix Δ, where each element can be addressed as $\delta_{ijt} \in \Delta$. This distance matrix Δ becomes the training data for the expected distance between a particular pair of words $(w_i, w_j) \in \mathcal{W}$ at time $t \in \mathcal{T}$. We use the notation δ_{ij} to represent a vector of size $|\mathcal{T}|$ with the temporal tf-idf-based cosine distance between $(w_i, w_j) \in \mathcal{W}$ for all the timestamps. The cosine distances are later used in the output layer of our proposed neural network.

4.2 Optimizing for similarity

One of our objectives is to obtain a low-dimensional word embedding model $\mathcal{U}$ such that computing the cosine distance between the word vectors results in a distance matrix that closely resembles Δ. Equation (2) formulates this objective as ϑ. In this case, we are optimizing the vectors in $\mathcal{U}$ to minimize the difference between the cosine distance of each pair of word vectors for every timestamp and the cosine distance from temporal tf-idf model in Δ (Eq. (1)). The minimization of the difference will ensure that our model captures the same similarity as the temporal tf-idf model but ours will provide low-dimensional contextual vectors.

In this paper, the term $dist(A, B)$ refers to the cosine distance between vector A and vector B. The cosine distance between two words vectors is bounded between $[0, 1]$. A cosine distance of 0 between two words vectors means that both words share the same context, while a cosine distance of 1 means that the vectors are completely orthogonal, thus does not share contextual similarities. The variable α is introduced as a scaling factor to avoid numerical stability issues with values close to zero. The simplest form of our objective function is as follows.

$$\vartheta_1(\mathcal{U}) = \sum_{i=1}^{|\mathcal{W}|}\sum_{j=1}^{|\mathcal{W}|}\sum_{t=1}^{|\mathcal{T}|} (\alpha \cdot dist(u_{it}, u_{jt}) - \alpha \cdot \delta_{ijt})^2 \quad (2)$$

4.3 Weighing relevance: Giving more importance to the neighborhood of each word

In our work, we focus on the task of studying the semantic evolution of a word based on changes to its context. Thus, it is more important that our word embedding model correctly captures the *relevant* neighborhood of a word. Our experiments demonstrated that each word has a small number of *relevant* neighbors. That is, each word shares context with a small number of words. To take this into account in the objective function, we introduce a penalty when the temporal tf-idf-based cosine distance δ_{ijt} is small, ensuring that our word embedding model captures the *relevant* context accurately.

$$\vartheta_2(\mathcal{U}) = \sum_{i=1}^{|\mathcal{W}|}\sum_{j=1}^{|\mathcal{W}|}\sum_{t=1}^{|\mathcal{T}|} (\alpha \cdot dist(u_{it}, u_{jt}) - \alpha \cdot \delta_{ijt})^2 \cdot e^{-\beta\delta_{ijt}} \quad (3)$$

where β is a scaling parameter to increase/decrease the importance given to the samples with a smaller distance. Notice that $e^{-\beta\delta_{ijt}}$ in Eq. (3) imposes a higher penalty to examples with smaller baseline distances. The penalty is less when the distance from the temporal tf-idf model is large. Equation (3) supports the phenomenon that, for a specific word, most of the words in the vocabulary are at a relatively large distance. The large distances need not be a part of the penalty because the objective function is only concerned about neighbors that appear in the vicinity for the temporal tf-idf model.

4.4 Temporal diffusion filter

Based on the diffusion theory (Angulo et al. 1980), we assume that the meaning of a word, and consequently its vector representation, diffuses (or drifts) over time. Thus, the word embeddings should evolve smoothly over time. To introduce this concept in our objective function, we model the effect of every word-vector in all timestamps to some degree.

We use a Gaussian filter (Eq. (4)) to *diffuse* the contribution of each vector smoothly before and after the timestamp of the current sample. The filter uses a sliding window, going from the first to the last timestamp. σ is a user-settable parameter representing the standard deviation of the Gaussian distribution. A large value of σ means that the diffusion of word vectors is slow over time. A small standard deviation allows capturing short-term changes in meaning.

$$\gamma(t,\sigma) = \left\langle \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t_i-t)^2}{2\sigma^2}}\right) \text{with } t_i = 1, \ldots, |\mathcal{T}| \right\rangle \quad (4)$$

Equation (5) presents the updated objective ϑ_3 which includes the temporal diffusion of the word embeddings.

$$\vartheta_3(\mathcal{U}) = \sum_{i=1}^{|\mathcal{W}|}\sum_{j=1}^{|\mathcal{W}|}\sum_{t=1}^{|\mathcal{T}|} (\alpha \cdot dist(\gamma(t,\sigma)U_i, \gamma(t,\sigma)U_j) - \alpha \cdot \delta_{ijt})^2 \cdot e^{-\beta\delta_{ijt}} \quad (5)$$

4.5 Smoothness penalty: Creating a homogeneous temporal embedding space

The second important goal that our word embedding model should achieve is to be spatially smooth over time. Continuous or smooth temporal embeddings are those where the

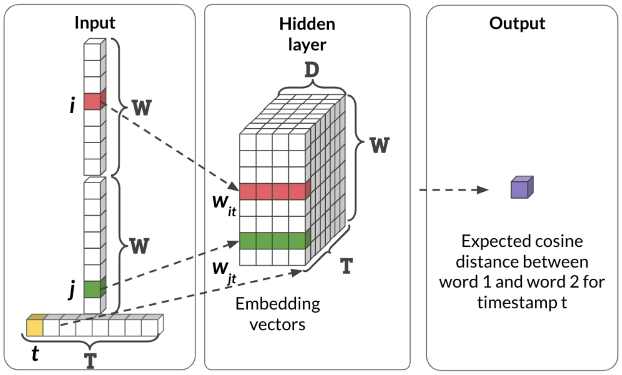

Input
Hidden layer
Output
i
W
W
j
t
T
D
W
w_{it}
w_{jt}
T
Embedding vectors
Expected cosine distance between word 1 and word 2 for timestamp t

Figure 1: The proposed neural network architecture for temporal embedding generation in the hidden layer.

distance (e.g., Manhattan or Euclidean) between two vectors of the same word for consecutive timestamps is small. Equation (6) captures the expected behavior by penalizing significant spatial changes.

$$\varepsilon_{1a}(\mathcal{U}) = \sum_{i=1}^{|\mathcal{W}|} \sum_{t=1}^{|\mathcal{T}|-1} ||u_{it+1}, u_{it}||_2 \tag{6}$$

The main issue with this expression is that by forcing consecutive vectors to be very close together, we might be losing important information when the vectors drift apart in the original data. Thus, we introduce weights, ω_ϑ, and ω_ε to control the effect of each objective. The final objective function takes the form of Eq. (7).

$$\mathcal{F}_a(\mathcal{U}) = \vartheta_3(\mathcal{U})^{\omega_\vartheta} \varepsilon_1(\mathcal{U})^{\omega_\varepsilon} \tag{7}$$

An alternative form would be:

$$\mathcal{F}_b(\mathcal{U}) = \omega_\vartheta \log \vartheta_3(\mathcal{U}) + \omega_\varepsilon \log \varepsilon_1(\mathcal{U}) \tag{8}$$

or

$$\mathcal{F}_c(\mathcal{U}) = \omega_\vartheta \vartheta_3(\mathcal{U}) + \omega_\varepsilon \varepsilon_1(\mathcal{U}) \tag{9}$$

4.6 Implementation

We implemented a neural network-based model using Tensorflow to generate our **low-dimensional temporal word embeddings**. An overall view of the architecture of our neural network is shown in Fig. 1. The goal of the neural network is to minimize Eq. (7). The embeddings for all words in all timestamps are generated in the hidden layer. We initialize the weights in the hidden layer in the range [0, 1]. The data used for training the model contains three inputs (one-hot encoding of a pair of words for which the cosine distance is known, and the timestamp) and one target value (cosine distance). The inputs are the indices for two random words w_{it} and w_{jt}, at timestamp t. The target value is the expected cosine distance between w_{it} and w_{jt}, obtained using the temporal tf-idf representations of Eq. (1).

5 Experimental Results

We performed experiments using three different datasets: a synthetic dataset, PubMed Pandemic dataset, and PubMed COVID dataset.

We generated **the synthetic dataset** consisting of 10,000 words and ten timestamps. For this dataset, we already know the 10-nearest neighbors of each word in every timestamp. Neighborhoods of larger sizes will contain random words starting at the 11th nearest neighbor.

The **PubMed pandemic dataset**, contains **328,908** abstracts of *pandemic and epidemic*-related biomedical publications. The abstracts were published between years 2000 to 2020. We selected 3,000 most frequent biomedical entities for this dataset.

The PubMed COVID dataset contains **41,571** abstracts of biomedical papers related to COVID-19, published in 2020. The corpus was collected from Kaggle COVID19 Open Research Dataset Challenge (Wang et al. 2020). We selected 2,000 most frequent biomedical entities for this dataset. We extracted the biomedical entities for the PubMed abstracts using *scispaCy*'s Biomedical Named Entity Recognition (Neumann et al. 2019). In this paper, we used the phrase *temporal word embedding* or *temporal embedding* to describe the core concepts, while in practice we performed *temporal biomedical entity embedding*.

We evaluate our temporal word embedding method by comparing its performance with that of a regular tf-idf model, the temporal tf-idf model (Camacho et al. 2018), dynamic Bernoulli embeddings (Rudolph and Blei 2018), and temporal word embeddings with a compass (TWEC) (Carlo, Bianchi, and Palmonari 2019). In all our experiments we used an embedding size of 64.

We seek to answer the following questions.

1. What is the effect of introducing different penalty terms in our objective function? (Section 5.1)
2. How well do the models perform in terms of capturing the neighborhood of entities over time, compared to the temporal tf-idf? (Section 5.2)
3. How well do the models perform in terms of capturing changes in the neighborhood over time in the respective embedding spaces? (Section 5.3)
4. How well does our algorithm track the quick evolution of a specific entity, such as *COVID*, compared to other methods? (Section 5.4)
5. How well does our algorithm capture semantic evolution of a general term, such as *pandemic*, compared to other methods? (Section 5.5)

5.1 Effect of penalty terms

In this experiment, we study the effect of the different versions of our objective function on the quality of the temporal word embedding model, focusing on the task of tracking semantic evolution. The versions under this study correspond to ϑ_1 (2), ϑ_2 (3), ϑ_3 (5), $\mathcal{F}_a$ (7), $\mathcal{F}_b$ (8), and $\mathcal{F}_c$ (9). We quantify the quality of the resulting vectors with two different metrics: similarity and continuity.

The similarity is measured as the number of intersections between the word neighborhoods obtained using the temporal tf-idf model and each of the different versions of our objective function. The goal of the similarity evaluation is to quantify how well our model mimics the temporal tf-idf

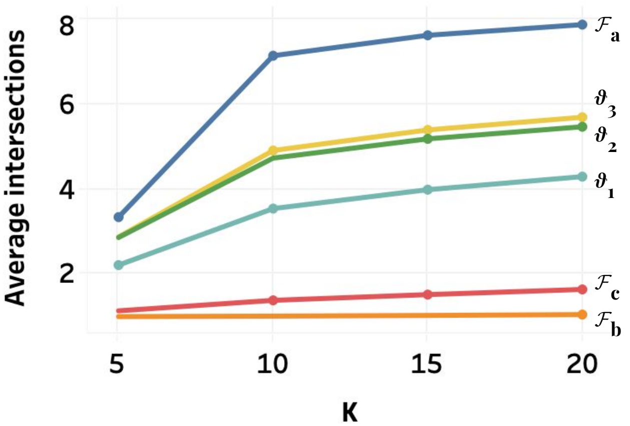


Figure 2: Average number of intersections per timestamp for different neighborhood sizes (k) between the neighborhoods obtained with the baseline method and those obtained using the different versions of our objective function (embedding size = 64).

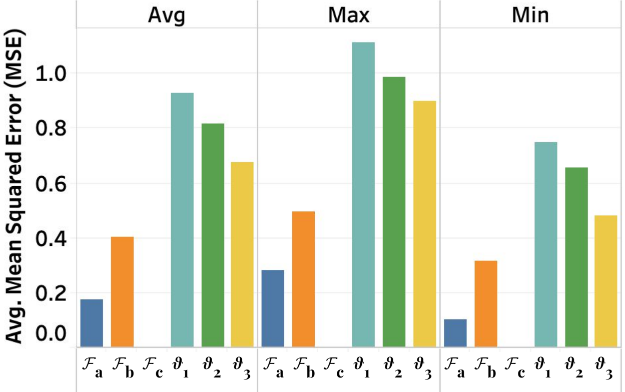


Figure 3: Average mean squared error (MSE) for different versions of our objective function. The average MSE is computed from obtaining the squared difference between vectors for the same word for every pair of consecutive timestamps (embedding size = 64).

model. It must be noted that we did not expect to have a perfect match in the neighborhoods of words since the temporal tf-idf model representation does not take into account latent contextual relationships between words.

The continuity is measured using the average, maximum, and minimum mean squared errors (MSE) across consecutive timestamps for the word vectors obtained using the different versions of our objective function.

Fig. 2 shows the results for the similarity evaluation with the synthetic dataset described at the beginning of Section 5. The objective function labeled as $\mathcal{F}_a$ on the figure performs significantly better than the other formulations. If we discard $\mathcal{F}_b$ and $\mathcal{F}_c$, it is possible to see how the similarity improves with the progression in which we developed our objective function. Furthermore, taking into account that only the top-10 nearest neighbors are known and set as *accurate* in the synthetic data and the rest of the neighbors are random, having an average of 8 intersections means that our model can correctly capture the semantic evolution of the synthetic dataset.

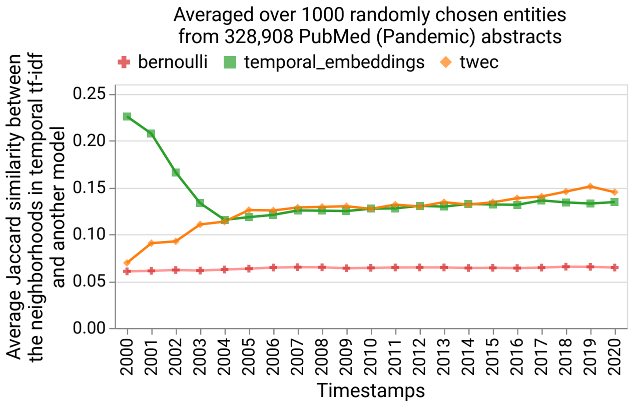


Figure 4: Jaccard similarity in the neighborhoods between temporal tf-df (Camacho et al. 2018) and each of the three models – TWEC, Bernoulli embeddings, and our temporal word embedding model. (PubMed (pandemic) dataset. Embedding size = 64.)

Evaluating continuity is required to ensure that there is a smooth transition between timestamps for the vectors of the same word. A high average or minimum MSE value indicates that there is a significant movement of the word vectors over time in the embedding space. However, a small maximum MSE value would mean that the word embeddings are not following the trends observed in the temporal tf-idf model-based training. Thus, the best model is one that has high similarity with the temporal tf-idf model while maintaining a low MSE value.

Fig. 3 shows the results for the continuity evaluation. In this case, F_c has a continuity of 0.0, which, in conjunction with the similarity results, indicates that this objective function produces static, unusable vectors. The second smallest average MSE value is obtained with $\mathcal{F}_a$, which also showed the best performance in terms of similarity. Thus, the **final objective function** is $\mathcal{F}_a$ (Eq. (7)), and we confirm that the smoothness penalty (Eq. (6)) has a positive effect both on the similarity and continuity results.

5.2 Capability to capture content neighborhood

A major purpose of any temporal word modeling is to capture content similarity over time. We compare three models – TWEC, Bernoulli embeddings, and our temporal word embedding – with Temporal tf-idf (Camacho et al. 2018) in Fig. 4, using PubMed (pandemic) dataset. We use temporal tf-idf (Camacho et al. 2018) for this comparison because it models content smoothly over time. Each line in the figure represents average set-based Jaccard similarity between the 10-nearest neighbors of 1000 randomly selected entities using temporal tf-idf and the 10-nearest neighbors of the same entities using one of the three models. Fig. 4 demonstrates that our embedding model and TWEC have closer similarity with temporal tf-idf than Bernoulli embeddings. Additionally, our model has greater similarity with the neighborhood of temporal tf-idf in the earlier timestamps, compared to both TWEC and Bernoulli embeddings. Our model

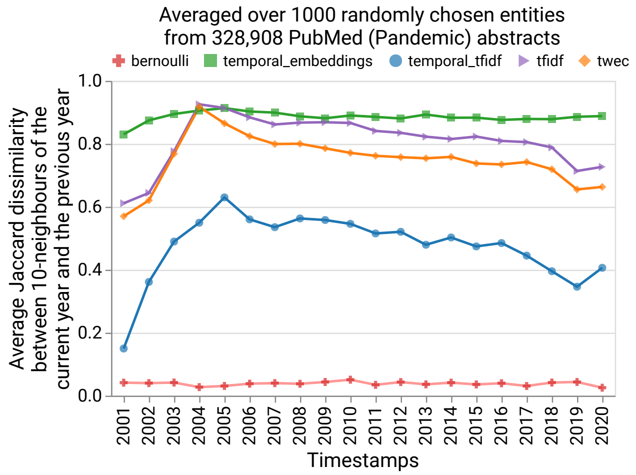


Figure 5: Comparison of average Jaccard dissimilarity (change) between 10-neighbors of the current year and the previous year.

smoothly spreads word-influence using diffusion over the years. As a result, our embedding model performs significantly better than other methods even when the vocabulary is smaller in the earlier timestamps.

5.3 Capability to detect changes in neighborhood

An objective of a temporal embedding technique is to capture changes in the neighborhood of each word over time. The ability to capture changes allows us to study the evolution of concepts. This subsection provides an experiment to investigate how much change occurs from one year to another in the neighborhood using different models. We quantify change in terms of set-based Jaccard dissimilarity (1.0-Jaccard similarity) between the neighborhood of a word in the current year and the neighborhood of the same word in the previous year. Average Jaccard dissimilarity over many words in a certain year for a model gives an overall idea of how much the model can detect changes in the neighborhood. Fig. 5 demonstrates average Jaccard dissimilarity (change) at each year for five different models – our temporal embedding model, Bernoulli embeddings, TWEC, and vanilla tf-idf computed independently at each year, and temporal tf-idf using 1000 randomly selected entities from the PubMed (pandemic) dataset. The plot shows that our temporal embedding model detects more changes in terms of average Jaccard dissimilarity compared to other models.

The Bernoulli embeddings capture the least amount of changes. Based on further investigation (not covered in this paper), we noticed that Bernoulli embeddings rarely capture any changes. These embeddings capture only a few long-term changes, whereas our temporal embedding model significantly captures both long-term and short-term changes. TWEC captures more changes than Bernoulli and temporal tf-idf, but lesser changes than the vanilla tf-idf. Our temporal word embedding performs even better than the vanilla tf-idf. Contextual changes are best-captured using our temporal embedding because the objective function of our model spreads the effect of each word smoothly from the current year to other years. As a result, our model captures changes, in terms of average Jaccard dissimilarity, better than regular tf-idf and temporal tf-idf models.

Our model is clearly superior in terms of the ability to capture changes. In subsection 5.4, we explain how the superiority in the detection of changes in the neighborhood helps in analyzing evolving concepts, such as *COVID-19*.

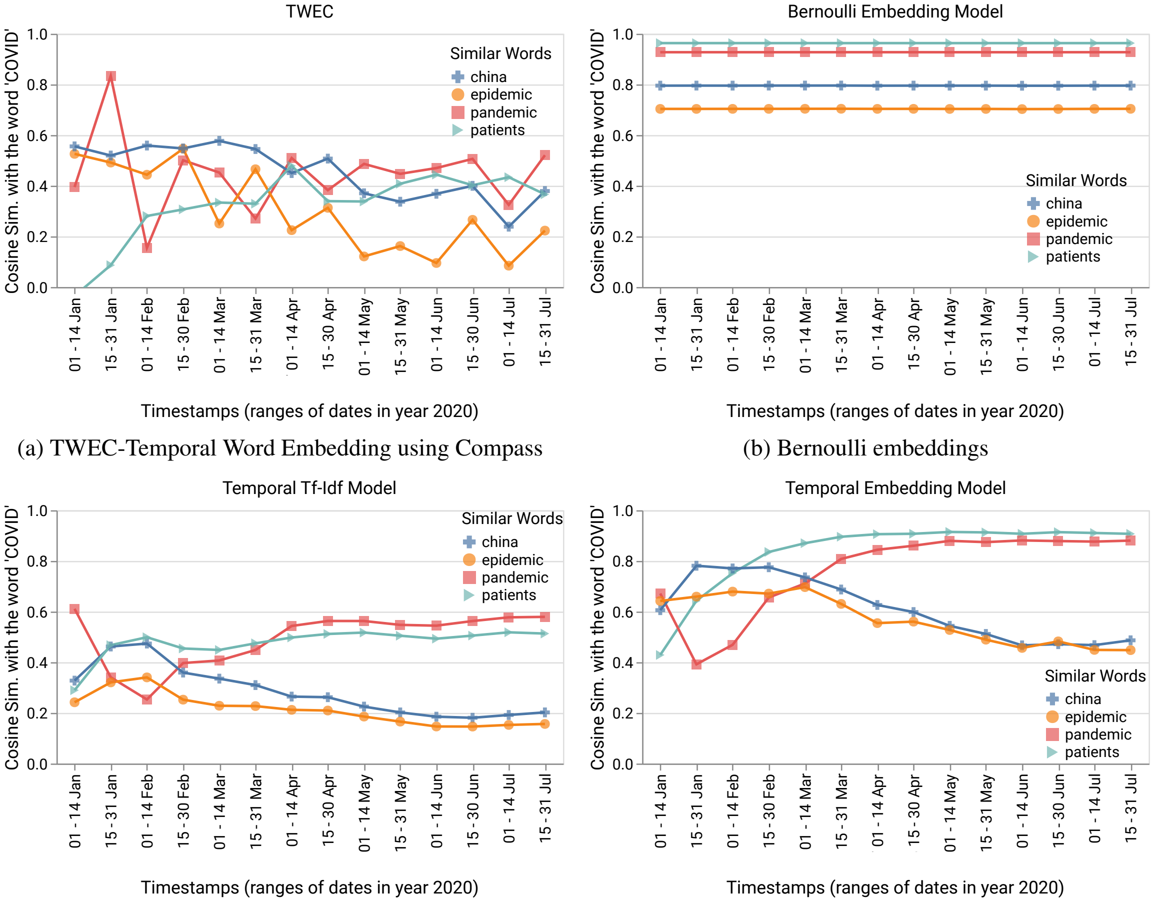
5.4 Analyzing the neighborhood of COVID-19

In this experiment, we analyze the changes in the neighborhood of the word *COVID* in the PubMed (COVID) dataset. Fig. 6 presents how the similarities between the entity *COVID* and some of its nearest neighbors–*China*, *epidemic*, *pandemic*, and *patients*– change over time using (a) TWEC model, (b) Bernoulli embeddings, (c) temporal tf-idf, and (d) our temporal embedding model. The data contains ranges of two-weeks from January to July of 2020. As we already know, COVID-19 is, as of August 2020, considered a pandemic – which is a global outbreak rather than a local epidemic (Steffens 2020). In Fig. 6, we observe that (c) temporal tf-idf and (d) our temporal embedding can detect the rising trends of *pandemic* and falling trends of the word *epidemic*. This observation matches with our known knowledge regarding COVID-19. TWEC (Fig. 6a) is able to track this to some degree but with zigzag-patterns in the trends. Bernoulli embeddings (Fig. 6b) give higher similarity for *pandemic* than *epidemic* with the word *COVID*, which is correct in July but the timeline does not demonstrate any rising and falling trends of the words *pandemic* and *epidemic*.

Our temporal embedding (Fig. 6d) demonstrates that the word *China* had high similarity with *COVID* in the beginning. The similarity started to fall by the end of March. According to our model, starting at the end of march the word *epidemic* started to exhibit lesser similarity with *COVID* and the word *pandemic* started to show higher similarity. The temporal tf-idf model (Fig. 6c) demonstrates a similar trend. The trends match with our common knowledge regarding the COVID-19 pandemic. Also, TWEC (Fig. 6a) has an overall downward trend for the word *China*, but with zigzag movements over the timeline. Bernoulli embeddings (Fig. 6 b) do not demonstrate any change and capture a static similarity for the entire timeline. We noticed that the underlying vectors in Bernoulli embeddings change but the neighbors of a word do not change much.

We know that the number of COVID infected *patients* increased over the months of 2020. Our temporal embedding model (as well as the temporal tf-idf) captures the rising-similarity of the word *patients* in the context of *COVID* quite smoothly (Fig. 6d). TWEC also has an upward trend which is less smooth. However, the Bernoulli embeddings do not demonstrate any changes in the similarity between the words *patients* and *COVID*.

This experiment demonstrates that our temporal embedding model captures the short-term changes in content (as shown by temporal tf-idf) while also capturing the context that we can track smoothly to study the evolution of a concept, such as *COVID*. In contrast, Bernoulli embeddings construct a context that is intractable in terms of similarity. TWEC provides noisy patterns that are difficult to interpret.



(a) TWEC-Temporal Word Embedding using Compass

(b) Bernoulli embeddings

(c) Temporal tf-idf model (vector size = 32,000)

(d) Our model (Eq. (7), vector size = 64)

Figure 6: Evolution of the word *COVID* in PubMed COVID-19-related abstracts published in 2020 using four different models – TWEC, Bernoulli embeddings, temporal tf-idf, and our temporal embedding model. *Cosine similarity* is used to compute the similarity between the vectors of the word *COVID* and any other word.

5.5 Analysis of the the word *Pandemic*

With the rise of the COVID-19 pandemic, it has become essential to study how biomedical scientists have dealt with a pandemic in the past years. Such an analysis requires a model that can capture long term changes. In this experiment, we attempt to track the closest term to the word *pandemic* in each year of the PubMed (pandemic) dataset, which spans biomedical abstracts from 2000 to 2020.

Each line of Fig. 7 plots the similarity of the top nearest-neighbor of the word *pandemic* in each year. The five lines represent similarities using five different models – Bernoulli embeddings, our temporal embedding model, temporal tf-idf, vanilla tf-idf, and TWEC. Notice that our temporal embedding model demonstrates peak similarities in 2009/2010 and in 2020, when H1N1 influenza (swine flu) and COVID-19, respectively became prominent. This signal from our temporal embedding model reflects the fact that the worst pandemics in the last 20 years are the H1N1 influenza in 2009 (Sullivan et al. 2010) and COVID-19 in 2020 (Cucinotta and Vanelli 2020). Note that other words like *concerns* in 2004 and *public* in 2015 are detected as the top nearest neighbors, which are not highly similar to the word *pandemic*. This indicates that no entities appeared too close to the word *pandemic* in those years.

TWEC captures *influenza* and *H1N1* in the middle of the timeline but fails to capture *COVID*-related keywords in 2020 as the closest entity to *pandemic*. In Fig. 7, the Bernoulli model can pick up *coronavirus* as the nearest neighbor of *pandemic* but it was not able to pick up influenza in its trend. Moreover, *coronovirus* appears in all the years as the top nearest neighbor of *pandemic* which is not correct because the fact is that the coronavirus spread started in 2019 and became a pandemic in 2020 (Cucinotta and Vanelli 2020). Temporal tf-idf and vanilla tf-idf were able to pick up *coronavirus*/*COVID*. Temporal tf-idf and vanilla tf-idf were also able to pick up influenza subtype *H1N1* (swine flu) but the respective similarities were not high.

Based on the experiment presented in this subsection, our temporal embedding model has the ability to separate highly contextual words (such as H1N1 and COVID) of a concept (such as *pandemic*) via similarity-peaks. Our model helps in determining prominent neighbors of a concept in the past.

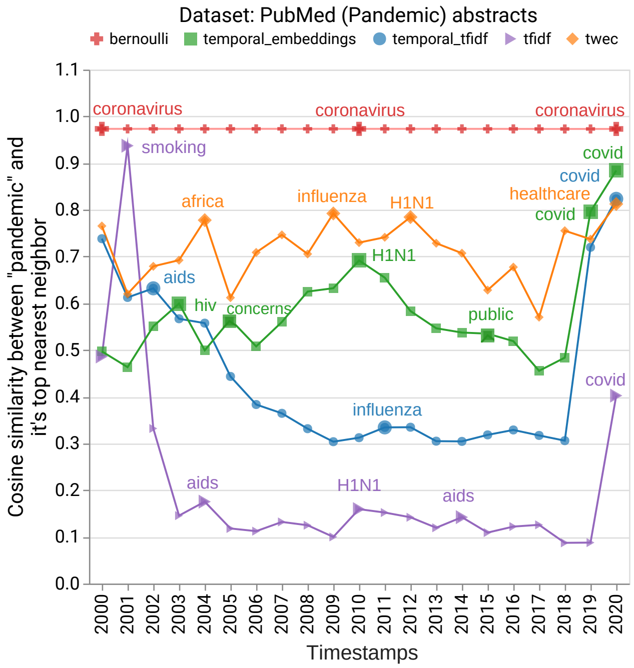


Figure 7: Cosine similarity of the top nearest neighbor of *pandemic* in each year using all methods. Nearest neighbors are written with selected peaks. Our temporal embedding method provides better context for *pandemic*. Embedding size = 64.

Our vectors are able to construct a peak for a prominent nearest neighbor because our method models diffusion. That is, a concept that appears today affects the past and the future to some extent, regardless of whether the concept directly appears in the contents or not.

6 Conclusions

This paper introduces a new technique to generate low-dimensional temporal word embeddings for timestamped scientific documents. We compare our model with existing temporal word embeddings. Our method generates a representation that: (1) can track changes observed within a short period, (2) provides a smooth evolution of the word vectors over a continuous temporal vector space, (3) uses the concept of *diffusion* to capture trends better than the existing models, and (4) is low-dimensional. Unlike previous models, our proposed model creates a homogeneous space over every timestamp of the embeddings. As a result, the generated vectors over timestamps can be used for prediction using conventional algorithms for predicting signals. Extrapolation of the embedding vectors to forecast a future neighborhood of a scientific concept is a future direction of this work.

References

Aitchison, J. 2013. *Language change: progress or decay?* Cambridge University Press.

Angulo, J.; Pederneiras, C.; Ebner, W.; Kimura, E.; and Megale, P. 1980. Concepts of diffusion theory and a graphic approach to the description of the epidemic flow of contagious disease. *Public Health Rep* 95(5): 478–485.

Bamler, R.; and Mandt, S. 2017. Dynamic Word Embeddings. In *Proceedings of the 34th ICML*, volume 70, 380–389. PMLR.

Barkan, O. 2017. Bayesian Neural Word Embedding. In *AAAI*, 3135–3143. San Francisco, California, USA: AAAI Press.

Camacho, R.; Dos Santos, R. F.; Hossain, M. S.; and Akbar, M. 2018. Tracking the Evolution of Words with Time-reflective Text Representations. In *2018 IEEE International Conference on Big Data (Big Data)*, 2088–2097. Seattle, WA, USA: IEEE.

Carlo, V. D.; Bianchi, F.; and Palmonari, M. 2019. Training Temporal Word Embeddings with a Compass. In *AAAI*.

Cucinotta, D.; and Vanelli, M. 2020. WHO Declares COVID-19 a Pandemic. *Acta bio-medica : Atenei Parmensis* 157—160.

Hamilton, W. L.; Leskovec, J.; and Jurafsky, D. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proc. of ACL*, volume 1, 1489–1501. Berlin, Germany.

Marina Del Rey, CA, U. 2018. Dynamic Word Embeddings for Evolving Semantic Discovery. In *Proc. of ACM WSDM*, 673–681.

Mihalcea, R.; and Nastase, V. 2012. Word Epoch Disambiguation: Finding How Words Change over Time. In *Proc. of ACL*, 259–263.

Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *ArXiv* abs/1310.4546.

Mitra, S.; Mitra, R.; Maity, S.; Riedl, M.; Biemann, C.; Goyal, P.; and Mukherjee, A. 2015. An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering* 21: 773–798.

Naim, S. M.; Boedihardjo, A. P.; and Hossain, M. S. 2017. A scalable model for tracking topical evolution in large document collections. In *IEEE BigData*, 726–735.

Neumann, M.; King, D.; Beltagy, I.; and Ammar, W. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *BioNLP@ACL*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global Vectors for Word Representation. In *Proc. of Conf. on EMNLP*, 1532–1543.

Radinsky, K.; Davidovich, S.; and Markovitch, S. 2012. Learning Causality for News Events Prediction. In *Proc. of Int. Conf. on WWW*, 909–918.

Rosin, G. D.; Adar, E.; and Radinsky, K. 2017. Learning Word Relatedness over Time. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1168–1178.

Rudolph, M.; and Blei, D. 2018. Dynamic Embeddings for Language Evolution. In *Proceedings of the 2018 World Wide Web Conference*, 1003–1011.

Rudolph, M. R.; and Blei, D. M. 2017. Dynamic Bernoulli Embeddings for Language Evolution. *ArXiv* abs/1703.08052.

Steffens, I. 2020. A hundred days into the coronavirus disease (COVID-19) pandemic. *Euro Surveill.* 25(14).

Sullivan, S. J.; Jacobson, R. M.; Dowdle, W. R.; and Poland, G. A. 2010. 2009 H1N1 influenza. *Mayo Clin. Proc.* 85(1): 64–76.

Tang, X.; Qu, W.; and Chen, X. 2013. Semantic Change Computation: A Successive Approach 68–81.

Wang, L. L.; Lo, K.; Chandrasekhar, Y.; Reas, R.; and et al. 2020. CORD-19: The COVID-19 Open Research Dataset.

Yogatama, D.; Wang, C.; Routledge, B. R.; Smith, N. A.; and Xing, E. 2014. Dynamic Language Models for Streaming Text. *Transactions of the Association for Computational Linguistics* 181–192.

Yule, G. 2017. *The study of language*. Cambridge University Press.