Deep Neural Models of Semantic Shift

Alex Rosenfeld
The University of Texas at Austin
Department of Linguistics
alexbrosenfeld@gmail.com

Katrin Erk
The University of Texas at Austin
Department of Linguistics
katrin.erk@mail.utexas.edu

Abstract

Diachronic distributional models track changes in word use over time. In this paper, we propose a deep neural network diachronic distributional model. Instead of modeling lexical change via a time series as is done in previous work, we represent time as a continuous variable and model a word's usage as a function of time. Additionally, we have created a novel synthetic task, which quantitatively measures how well a model captures the semantic trajectory of a word over time. Finally, we explore how well the derivatives of our model can be used to measure the speed of lexical change.

1 Introduction

Diachronic distributional models have provided interesting insights into how words change meaning. Generally, they are used to explore how specific words have changed meaning over time (Sagi et al., 2011; Gulordava and Baroni, 2011; Jatowt and Duh, 2014; Kim et al., 2014; Kulkarni et al., 2015; Bamler and Mandt, 2017; Hellrich and Hahn, 2017), but they have also been used to explore historical linguistic theories (Xu and Kemp, 2015; Hamilton et al., 2016a,b), to predict the emergence of novel senses (Bamman and Crane, 2011; Rohrdantz et al., 2011; Cook et al., 2013, 2014), and to predict world events (Kutuzov et al., 2017a,b).

Diachronic distributional models are distributional models where the vector for a word changes over time. Thus, we can calculate the cosine similarity between the vectors for a word at two different time points to measure how much that word has changed over time and we can perform a nearest neighbor analysis to understand in what direction a word is changing. For example, diachronic distributional models can detect that the word *gay* has greatly changed by comparing the word vector for *gay* across different time points. They can also be used to discover that gay has shifted its meaning from *happy* to *homosexual* by analyzing when those words show up as nearest neighbors to *gay*.

Previous research in diachronic distributional semantics has used models where data is partitioned into time bins and a synchronic model is trained on each bin. A synchronic model is a vanilla, time-independent distributional model, such as skip-gram. However, there are several technical issues associated with data binning. For example, if the bins are too large, you can only achieve extremely coarse grained representations of lexical change over time. However, if the bins are too small, the synchronic models get trained on insufficient data.

In this paper, we have built the first diachronic distributional model that represents time as a continuous variable instead of employing data binning. There are several advantages to treating time as continuous. The first advantage is that it is more realistic. Large scale change in the meaning of a word is the result of change happening one person at a time. Thus, semantic change must be a gradual process. By treating time as a continuous variable, we can capture this gradual shift. The second advantage is that it allows a greater representation of the underlying causes behind lexical change. Words change usage in reaction to real world events and multiple words can be affected by the same event. For example, the usage of *gay* and *lesbian* have changed in similar ways due to changing perceptions of homosexuality in society. By associating time with a vector and having word representations be a function of that vector, we can model a single underlying cause affecting multiple words similarly.

Proceedings of NAACL-HLT 2018, pages 474–484
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

It is difficult to evaluate diachronic distributional models in their ability to capture semantic shift as it is extremely difficult to acquire gold data. Distributional models are traditionally evaluated with word similarity judgments, which we cannot obtain for word usage in the past. Thus, evaluation of diachronic distributional models is a focus of research, such as work done by Hellrich and Hahn (2016) and Dubossarsky et al. (2017). Our approach is to create a synthetic task to measure how well a model captures gradual semantic shifts.

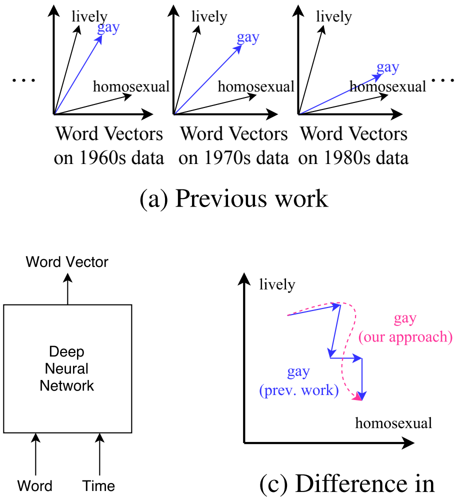
We will also explore how we can use our model to predict the speed at which a word changes. Our model is differentiable with respect to time, which gives us a natural way to measure the velocity, and thus speed, of a word at a given time. We explore the capabilities and limitations of this approach.

In short, our paper provides the following contributions:

- We have developed the first continuous diachronic distributional model. This is also the first diachronic distributional model using a deep neural network.
- We have designed an evaluation of a model's ability to capture semantic shift that tracks gradual change.
- We have used the derivatives of our model as a natural way to measure the speed of word use change.

2 Related work

Previous research in diachronic distributional models has applied a binning approach. In this approach, researchers partition the data into bins based on time and train a synchronic distributional model on that bin's data (See Figure 1). Several authors have used large bin models in their research, such as using five year sized bins (Kulkarni et al., 2015), decade sized bins (Gulordava and Baroni, 2011; Xu and Kemp, 2015; Jatowt and Duh, 2014; Hamilton et al., 2016a,b; Hellrich and Hahn, 2016, 2017), and era sized bins (Sagi et al., 2009, 2011). The synchronic model for each time bin was trained independently of the others. In order to get a fine grained representation of semantic shift, several authors have used small bins. Kim et al. (2014) trained a synchronic model for each time bin. To mitigate data issues, Kim et al. preinitialized a time bin's synchronic model with the model from the previous time bin. Bamler and Mandt (2017) developed a small bin probabilistic approach that used transition probabilities to lessen data issues. They have two versions of their method. The first version trains the distribution in each bin iteratively and the second version trains a joint distribution over all bins. In this paper, we only explore the first version as the second version does not scale well to large vocabulary sizes. Following Bamler and Mandt (2017), we compare to models used by Hamilton et al. (2016b), Kim et al. (2014), and the first version of Bamler and Mandt's's model.



(a) Previous work

(b) Our approach

(c) Difference in trajectories

Figure 1: Difference between our approach and previous work. Previous work in diachronic distributional models (a) has trained synchronic distributional models on consecutive time bins. In our work (b), a neural network takes word and time as input and produces a time specific word vector. In (c), we sketch that previous work produces a jagged semantic trajectory (blue, solid curve) whereas our model produces a smooth semantic trajectory (pink, dotted curve).

There have been other models of lexical change beside distributional ones. Topic modeling has been used to see how topics associated to a word have changed over time (Wijaya and Yeniterzi, 2011; Frermann and Lapata, 2016). Sentiment analysis has been applied to determine how sentiments associated to a word have changed over time (Jatowt and Duh, 2014).

As mentioned in the introduction, it is difficult to quantitatively evaluate diachronic distributional models due to the lack of gold data. Thus, previous research has attempted alternative routes to quantitatively evaluate their models. One route is to use intrinsic evaluations, such as measuring a trajectory's smoothness (Bamler and Mandt, 2017). However, intrinsic measures do not directly measure semantic shift, which is the main use of diachronic distributional models. Hamilton et al. (2016b) use attested shifts generated by historical linguists. However, outside of first attestations, it is a difficult task for historical linguists themselves to accurately detail semantic shifts (Deo, 2015). Additionally, the task used by Hamilton et al. is unusable for model comparison as all but one model had a 100% accuracy in this task. Kulkarni et al. (2015) used a synthetic task to evaluate how well diachronic distributional models can detect semantic shift. They took 20 copies of wikipedia where each is a synthetic version of a time bin and changed several words in the last 10 copies. Models were then evaluated on their ability to detect when those words changed. Our evaluation improves upon this one by having the test data be from a diachronic corpus and we model lexical change as a gradual process rather than searching for a single change point.

3 Models

In this section, we describe the four diachronic distributional models that we analyze in our current work. Three will be from previous research to be used as benchmarks. Each of the four models we analyze are based on skip-gram with negative sampling (**SGNS**). The difference between the four diachronic distributional models we analyze is how they apply SGNS to changes over time.

Skip-gram with negative sampling (SGNS) is a word embedding model that learns a latent representation of word usage (Mikolov et al., 2013). For target words w and context words c, vector representations $\vec{w}$ and $\vec{c}$ are learned to best predict if c will be in context of w in a corpus. k negative contexts are randomly sampled for each positive context. Vector representations are computed by optimizing the following loss function:

$$\sum_{(w,c)\in D} [log(\sigma(\vec{w}\cdot\vec{c}))+\sum_{c_1,\ldots c_k\sim P_D} log(1-\sigma(\vec{w}\cdot\vec{c_i}))] \tag{1}$$

where D is a list of target-context pairs extracted from the corpus, P_D is the unigram distribution on the corpus, σ is the sigmoid function, and k is the number of negative samples.

3.1 Binning by Decade

The first diachronic distributional model we will consider is a large time bin model proposed by Hamilton et al. (2016b). Here, time is partitioned into decades and an SGNS model is trained on each decade's worth of data. We label this model **LargeBin**.

3.2 Preinitialization

The second diachronic distributional model we will consider is a small time bin model proposed by Kim et al. (2014). Here, time is partitioned into years and an SGNS model is trained on each year's worth of data. Data issues are mitigated by preinitializing the model[1] for a given time bin with the vectors of the preceding time bin (Kim et al., 2014). We label this model **SmallBinPreInit**.

3.3 Prior and Transition Probabilities

The third diachronic distributional model we will consider comes from Bamler and Mandt (2017). Bamler and Mandt take a probabilistic approach to modeling semantic change over time. The idea is to transform the SGNS loss function into a probability distribution over the target and context vectors. Then, to create a better diachronic distributional model, they apply priors to this distribution.

The first two priors are Gaussian distributions with mean zero on the vector variables to discourage the vectors from growing too large (Barkan, 2017). More formally:

$$\begin{aligned} P_1(\vec{w}) &= \mathcal{N}(0, \alpha_1 I) \\ P_2(\vec{c}) &= \mathcal{N}(0, \alpha_1 I) \end{aligned} \tag{2}$$

where α_1 is a hyperparameter.

The last two priors are also Gaussian distributions on the vector variables. The means are the vector representation from the previous bin. The goal of this prior is to discourage a vector variable from deviating from the previous bin's vectors.

$$\begin{aligned} P_3(\vec{w}) &= \mathcal{N}(\overrightarrow{w_{prev}}, \alpha_2 I) \\ P_4(\vec{c}) &= \mathcal{N}(\overrightarrow{c_{prev}}, \alpha_2 I) \end{aligned} \tag{3}$$

where α_2 is a hyperparameter and $\overrightarrow{w_{prev}}$ and $\overrightarrow{c_{prev}}$ are the vectors from the previous time bin.

We are only exploring point models, thus we take the maximum a posteriori estimate of the

[1] We do not perform preinitialization in LargeBin as large bin models are less susceptible to data issues.

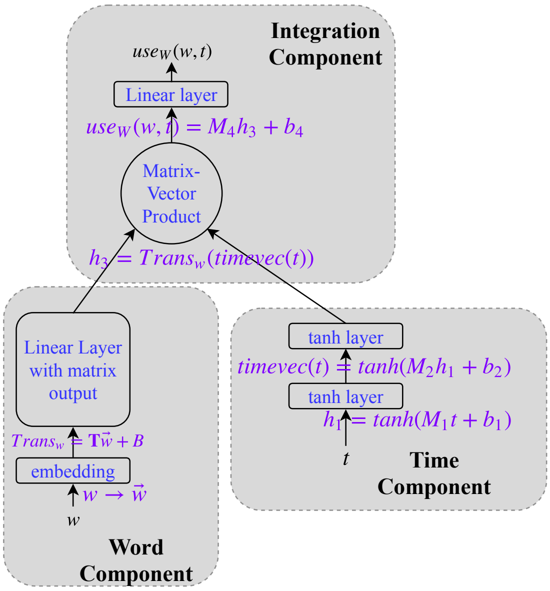

Integration Component
$use_W(w,t)$
Linear layer
$use_W(w,t) = M_4 h_3 + b_4$
Matrix-Vector Product
$h_3 = Trans_w(timevec(t))$
Linear Layer with matrix output
$Trans_w = \mathbf{T}\vec{w} + B$
embedding
$w \rightarrow \vec{w}$
w
Word Component
tanh layer
$timevec(t) = tanh(M_2 h_1 + b_2)$
tanh layer
$h_1 = tanh(M_1 t + b_1)$
t
Time Component

Figure 2: Diagram of DiffTime. $timevec(t)$ encodes temporal information as a vector. M_W encodes lexical information as a matrix. The target vector for w at time t, $use_W(w,t)$, is found by combining $Trans_w$ and $timevec(t)$. Context version $use_C(c,t)$ is the same except that it has its own embedding layer.

joint distribution to recover the vectors for each time bin. We apply a logarithm in constructing the estimate, which transforms the joint probability into the SGNS loss function with four regularizers (each one corresponding to a prior distribution). The prior distribution P_1 becomes $\sum_{w \in W} \frac{\alpha_1}{2}||w||$. The prior distribution P_2 becomes $\sum_{c \in C} \frac{\alpha_1}{2}||c||$. The prior distribution P_3 becomes $\sum_{w \in W} \frac{\alpha_2}{2}||\vec{w} - \overrightarrow{w_{prev}}||$. The prior distribution P_4 becomes $\sum_{c \in C} \frac{\alpha_2}{2}||\vec{c} - \overrightarrow{c_{prev}}||$. W and C are the sets of target and context words. We label this model **SmallBinReg**.

3.4 DiffTime Model

Our model is a modification of the SGNS algorithm to accommodate a continuous time variable. The original SGNS algorithm produces a target embedding $\vec{w}$ for target word w and a context embedding $\vec{c}$ for context word c. Instead, we produce a differentiable function $use_W(w,t)$ that returns a target embedding for target word w at time t and a differentiable function $use_C(c,t)$ that produces a context embedding for context word c at time t.

Our model consists of three components. One component takes time as its input and produces an embedding that characterizes that point in time (lower right). The second component (lower left) takes a word as its input and produces a time-independent word embedding, which is then reshaped into a set of parameters that can modify the time embedding. The third component (top) combines the time embedding and the word embedding.

The first component of our model is a two-layer feed-forward neural network with tanh activation functions. These layers take a time t as input and produces a time embedding $timevec(t)$ as output of those layers:

$$\begin{aligned} h_1 &= tanh(M_1 t + b_1) \\ timevec(t) &= tanh(M_2 h_1 + b_2) \end{aligned} \tag{4}$$

where M_1 and M_2 are the weights of the first two layers and b_1 and b_2 are the biases. To produce the input value t, a timepoint is scaled to a value between 0 and 1, where 0 corresponds to the year 1900, and 1 corresponds to 2009, the last year for which our corpus has data.

The second component incorporates word-specific information into our model. For $use_W(w,t)$, each target word w has a target vector representation $\vec{w}$. The vector $\vec{w}$ is then transformed into a linear transformation $Trans_w$, which in the third component is applied to the time embedding $timevec(t)$. We do this via a modified linear layer where the weights are a three dimensional tensor, the biases are a matrix and the output is a matrix:

$$Trans_w = \mathbf{T}\vec{w} + B \tag{5}$$

where $\mathbf{T}$ is the tensor acting as the weights and B is the matrix acting as the biases.

The third component combines the word-independent time embedding $timevec(t)$ and the time-independent linear transformation $Trans_w$ together to produce the final result. First, $Trans_w$ is applied to $timevec(t)$:

$$h_3 = Trans_w(timevec(t)) \tag{6}$$

Then, an additional linear layer is used as the output layer, taking h_3 as input:

$$use_W(w,t) = M_4 h_3 + b_4 \tag{7}$$

where M_4 and b_4 are the weights and biases of the output layer.

The above details the architecture of $use_W(w,t)$. The corresponding function $use_C(c,t)$ for context words has the same architecture as $use_W(w,t)$ and shares weights with $use_W(w,t)$. The only exception is that $use_C(c,t)$ uses a separate set of vectors $\vec{c}$ in the second component instead of sharing the target vectors $\vec{w}$ with $use_W(w,t)$.

We train our model using a modified version of the SGNS loss function. In particular, our positive samples are now triples (w,c,t) where w is a target word, c is a context word, and t is a time, instead of pairs (w,c) which are typically used in SGNS. For each positive sample (w,c,t), we sample k negative contexts from the unigram distribution, P_D. P_D is trained from all contexts in the entire corpus and is time-independent. Explicitly, the loss function is:

$$\sum_{(w,c,t)\in D} log(\sigma(use_W(w,t)\cdot use_C(c,t)))+ \\ kE_{c_N\sim P_D}[log(\sigma(-use_W(w,t)\cdot use_C(c_N,t)))] \tag{8}$$

3.5 Training

All models are trained on the same training data. We used the English Fiction section of the Google Books ngram corpus (Lin et al., 2012). We use the English fiction specifically, because it is less unbalanced than the full English section and less influenced by technical texts (Pechenick et al., 2015). We only use the years 1900 to 2009 as there is limited data before 1900.

We converted the ngram data for this corpus into a set of (target word, context word, year, frequency) tuples. The frequency is the expected number of times the target word-context word pair is sampled from that year's data using skip-gram. Following Hamilton et al. (2016b), we use subsampling with $t = 10^{-5}$. As the number of texts published since 1900 has increased five fold, we weigh the frequencies so that the sums across each year are equal.

For the binned models, we train each bin's synchronic model using the subset of the training data corresponding to that time bin. For our model, we sample (training word, context word, year) triples from the entire training data as the year is an input to our function.

4 Evaluation

4.1 Synchronic Accuracy

Method	Time	Spearman's ρ
LargeBin	1990s bin	0.615
SmallBinPreInit	1995 bin	0.489
SmallBinReg	1995 bin	0.564
DiffTime	start of 1995	**0.694**

Table 1: Synchronic accuracy of the methods. Time is the point of time we use as our synchronic model.

Before we can evaluate the methods as models of diachronic semantics, we must first ensure that the methods model semantics accurately. To do this, we follow Hamilton et al. (2016b) by performing the MEN word similarity task on vectors extracted from a fixed time point (Bruni et al., 2012). The hope is that the word similarity predictions of a model at that point in time highly correlate with word similarity judgments in the MEN dataset. For the binned models, we used the vectors from the bin best corresponding to 1995 to reflect the 1990s bin chosen by Hamilton et al. (2016b). DiffTime represents time as a continuous variable, so we chose a time t that corresponds to the start of 1995.

The results of MEN word similarity tasks is in Table 1. All of the Spearman's ρ values are comparable to those found in Levy and Goldberg (2014) and Hamilton et al. (2016b). Thus, all of these models reflect human judgments comparable to synchronic models. Thus, the predictions of the models correlate with human judgments.

4.2 Synthetic Task

The goal of creating diachronic distributional models is to help us understand how words change meaning over time. To that end, we have created a synthetic task to compare models by how accurately they track semantic change.

Our task creates synthetic words that change between two senses over time via a sigmoidal path. A sigmoidal path will allow us to emulate a word starting from one sense, shifting gradually to a second sense, then stabilizing on that second sense. By using sigmoidal paths, we can explore how well a model can track words that have switched senses over time such as *gay* (lively to homosexual) and *broadcast* (scattering seeds to televising shows). A similar task is used to evaluate word sense disambiguation (Gale et al., 1992; Schütze, 1992).

The synthetic words are formed by a combination of two real words, e.g. *banana* and *lobster* are combined together to form $banana{\circ}lobster$. The real words are randomly sampled from two distinct semantic classes from the BLESS dataset (Baroni and Lenci, 2011). We use BLESS classes so that we can capture how semantically similar a synthetic word is to its component words by comparing to other words in the same BLESS classes as the component word. For example, we can capture how similar $banana{\circ}lobster$ is to $banana$ by comparing $banana{\circ}lobster$ to words in the fruit BLESS class. See Appendix B for preprocessing details. We denote the synthetic words with $r_1{\circ}r_2$ where r_1 and r_2 are the component real words.

We also randomly generate the sigmoidal path by which a synthetic word changes from one sense to another. For real words r_1 and r_2, this path will be denoted $\mathit{shift}(t; r_1{\circ}r_2)$ and is defined by the following equation:

$$\mathit{shift}(t; r_1{\circ}r_2) = \sigma(s(t - m)) \tag{9}$$

The value s is uniformly sampled from $(\frac{1.0}{110}, \frac{10.0}{110})$ and represents the steepness of the sigmoidal path. The value m is uniformly sampled from $\{1930, \ldots, 1980\}$ and represents the point where the synthetic word is equally both senses. For our example synthetic word $banana{\circ}lobster$, $banana{\circ}lobster$ can transition from meaning banana to meaning lobster via the sigmoidal path $\sigma(0.05(t - 1957))$ where 1957 is the time where $banana{\circ}lobster$ is equally banana and lobster and 0.05 represents how gradually $banana{\circ}lobster$ shifts senses.

We then use $\mathit{shift}(t; r_1{\circ}r_2)$ to integrate $r_1{\circ}r_2$ into the real diachronic corpus data. Our training data is a set of (target word, context word, year, frequency) tuples extracted from a diachronic corpus (see 3.5). For every tuple where r_1 is the target word, we replace the target word with $r_1{\circ}r_2$ and we multiply the frequency by $\mathit{shift}(t; r_1{\circ}r_2)$. For every tuple where r_2 is the target word, we replace the target word with $r_1{\circ}r_2$ and we multiply the frequency by $1 - \mathit{shift}(t; r_1{\circ}r_2)$. In other words, in the modified corpus, $r_1{\circ}r_2$ has $\mathit{shift}(t; r_1{\circ}r_2)$ percent of r_1's contexts at time t and $1 - \mathit{shift}(t; r_1{\circ}r_2)$ percent of r_2's contexts at time t.

We train a model mod on this modified training data. This provides a representation for $r_1{\circ}r_2$ over time. We can capture how much a model predicts $r_1{\circ}r_2$ is more semantically similar to r_1 than r_2 by comparing mod's representation of $r_1{\circ}r_2$ to words in the same semantic category as r_1 and r_2. We use BLESS classes as our notion of semantic category. If cls_1 is the BLESS class of r_1 and cls_2 is the BLESS class of r_2, then mod's prediction for how much more similar $r_1{\circ}r_2$ is to r_1 than r_2, $rec(t; r_1{\circ}r_2, mod)$, is defined as follows:

$$\begin{aligned} rec(t; r_1{\circ}r_2, mod) &= \\ &\frac{1}{|cls_1|} \sum_{r_1' \in cls_1} sim^{mod}(r_1{\circ}r_2, r_1', t) \\ &- \frac{1}{|cls_2|} \sum_{r_2' \in cls_2} sim^{mod}(r_1{\circ}r_2, r_2', t) \end{aligned} \tag{10}$$

$sim^{mod}(r_1{\circ}r_2, r_1', t)$ is the cosine similarity between mod's word vector for $r_1{\circ}r_2$ at time t and mod's word vector for r_1' at time t.

Method	AMSE 1900–2009	AMSE 1950–2009
LargeBin	62.52	51.71
SmallBinPreInit	171.43	49.88
SmallBinReg	106.79	42.67
DiffTime	**25.67**	**11.48**

Table 2: Model performance under the synthetic evaluation. The values are the mean sum of squares error (MSSE) for each method. Lower value is better. The first column is MSSE using all times. The second column is MSSE using years 1950 to 2009.

To evaluate a model in its ability to capture semantic shift, we use the mean sum of squares error (MSSE) between $rec(t; r_1{\circ}r_2, mod)$ and $\mathit{shift}(t; r_1{\circ}r_2)$ across all synthetic words. The function $rec(t; r_1{\circ}r_2, mod)$ is model mod's prediction of how much more similar $r_1{\circ}r_2$ is to r_1 than r_2. The gold value of $rec(t; r_1{\circ}r_2, mod)$ would then be the sigmoidal path that defines how $r_1{\circ}r_2$ semantically shifts from r_1 to r_2 over time, $\mathit{shift}(t; r_1{\circ}r_2)$. To evaluate how accurately mod predicted the semantic trajectory of $r_1{\circ}r_2$, we calculate the mean squared error between $rec(t; r_1{\circ}r_2, mod)$ and $\mathit{shift}(t; r_1{\circ}r_2)$ as follows:

$$\sum_{t=1900}^{2009} (rec(t; r_1 \circ r_2, mod) - \mathit{shift}(t; r_1 \circ r_2))^2 \tag{11}$$

As $rec(t; r_1 \circ r_2, mod)$ and $\mathit{shift}(t; r_1 \circ r_2)$ have different scales, we Z-scale both the $rec(t; r_1 \circ r_2, mod)$ values and the $\mathit{shift}(t; r_1 \circ r_2)$ values before calculating the mean squared error.

We use three sets of 15 synthetic words and the average is calculated over all 45 words. The synthetic words and BLESS classes we used are contained in the supplementary material. The results are in Table 2. The column AMSE is MSSE when all years are taken into account. Kim et al. (2014) noted that small bin models require an initialization period, so the column AMSE (1950-) is MSSE when only years 1950 to 2009 are taken into account and the years 1900 to 1949 are used as the initialization period. From the table, we see our model outperforms the three benchmark models in both cases. Using a paired t-test, we found that the reduction in MSSE between our model and the benchmark models are statistically significant.

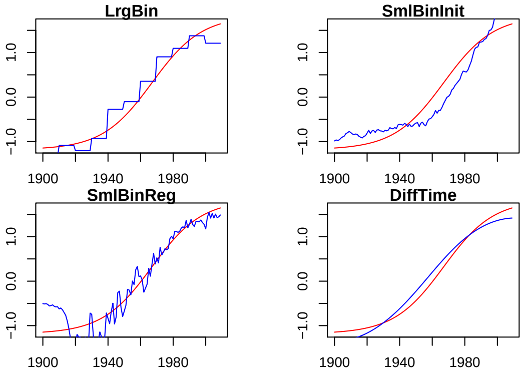


Figure 3: Graph Comparisons between $\mathit{shift}(t; r_1 \circ r_2)$ (red) and $rec(t; r_1 \circ r_2,)$ (blue) for the synthetic word $pistol \circ elm$. The x-axis are the years and the y-axis are the values of $\mathit{shift}(t; r_1 \circ r_2)$. $rec(t; r_1 \circ r_2, mod)$ and $\mathit{shift}(t; r_1 \circ r_2)$ have been Z-scaled.

In Figure 3, we plot $\mathit{shift}(t; r_1 \circ r_2)$ and $rec(t; r_1 \circ r_2, mod)$ for the synthetic word $pistol \circ elm$. Each method has a subgraph. The predictions of the large bin model LargeBin appear as a step function with large steps (top left graph). These large steps seem to cause the predicted shift (blue curve) to poorly correlate with the gold shift (red curve). Next, we consider the small bin models SmallBinPreInit (top right graph) and SmallBinReg (bottom left graph). Both predicted shifts have an initial portion that poorly fits the generated shift (between 1900 and 1950). From Kim et al. (2014), it takes several iterations for small bin models to stabilize due to each bin being fed limited data. Additionally, there are fluctuations in the graphs of the predicted shift, which we attribute to the high variance of data per bin. In contrast to the other models, our predicted shift tightly fits the gold shift (bottom right graph).

Although this evaluation provides useful information on the quality of an diachronic distributional model, it has some weaknesses. The first is that it is a synthetic task that operates on synthetic words. Thus, we have limited ability to understand how well a model will perform on real world data. Second, we only generate words that shift from one sense to another. This fails to account for other common changes, such as gaining/losing senses and narrowing/broadening. Finally, by using a sigmoidal function to generate how words change meaning, we may have privileged continuous models that incorporate a sigmoidal function in their architecture. We are working towards improving this evaluation to remove these issues.

4.3 Speed of word use change

In this section, we evaluate our model's ability to measure the speed at which a word is changing. Our model is differentiable with respect to time. Thus, we can get the derivative of $use_W(w, t)$ with respect to t to model how word w is changing usage at time t. We $l2$-normalize $use_W(w, t)$ beforehand to reduce frequency effects. We then get the magnitude of this normalized derivative to model the speed at which a word is changing at a given time.

We explore the connection between speed and the nearest neighbors to a word in Figure 4. First, we use *apple* as a baseline for discussion. We chose *apple*, because the meaning of the word has remained relatively stable throughout the 1900s. With *apple*, we see a low speed over time and a consistency in the cosine similarity to *apple*'s nearest neighbors. While it is true that *apple* has other meanings beyond the fruit, such as referring to Apple Inc., those meanings are much rarer, especially in the fiction corpus we use.

In contrast to *apple*, the word *gay* has a very high speed and a drastic change for *gay*'s nearest

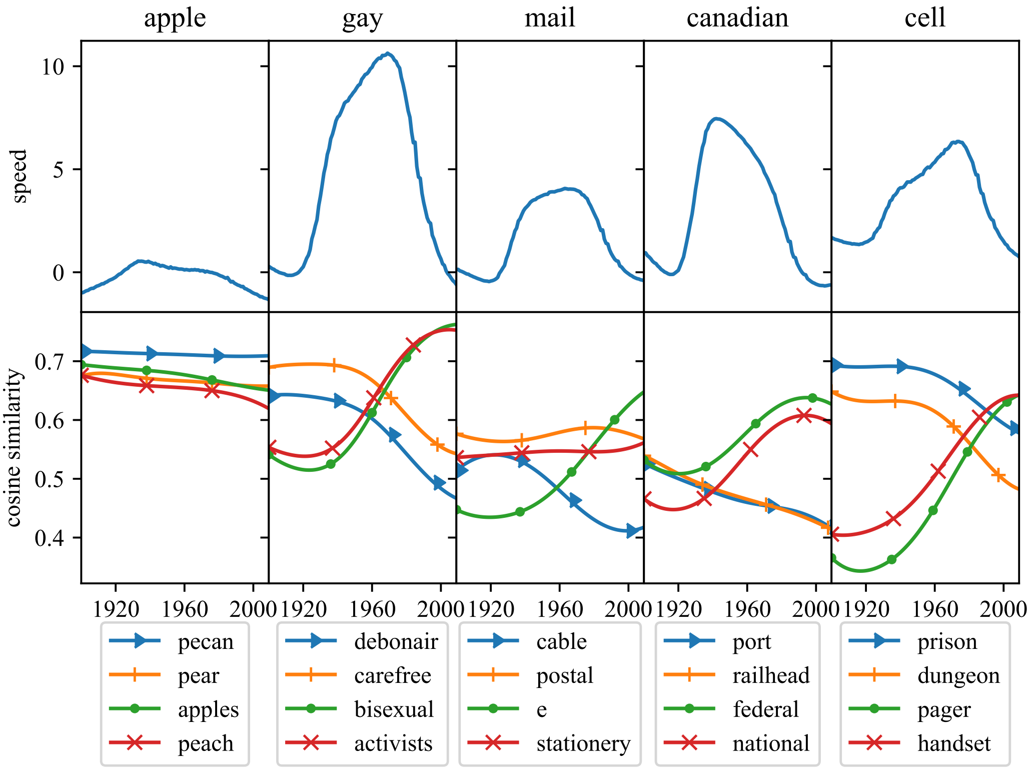


Figure 4: Speed and nearest neighbors over time of selected words. The top graphs as the speed at which a word changes usage according to our model. The bottom graphs are selected nearest neighbors for those words. Each of the chosen nearest neighbors appear as a top 10 nearest neighbor to the word at some year.

neighbors. This makes sense as *gay* is well established to have experienced a drastic sense change in the mid to late 1900s (Harper, 2014).

Next, we explore the word *mail*. The word *mail* has a moderately high speed. This may be reflective of the fact that there have been incredible changes in the medium by which we send mail, e.g. changing from cables to email. A possible reason for the speed only being moderately high is that, even though the medium by which we send mail has changed, many of the same uses of mail, e.g. sending, receiving, opening, etc., remain the same. We see this reflected in the nearest neighbors as well as *mail* shifts from a high similarity to *cable* to a high similarity to *e* (as in email), yet *mail* is consistently similar to *postal* and *stationery*.

The next word we will explore is the word *canadian*. We chose this word as we were surprised to find that *canadian* has one of the fastest speeds in the 1930s to 1940s. The nearest neighbors to *canadian* have shifted from geographic terms like *port* and *railhead* to civil terms like *federal* and *national*. In further analysis, we discovered that this may be reflective of a larger push to form a Canadian identity in the early 1900s (Francis, 1997). The nearest neighbors to *canadian* may reflect the change from being a part of the British Empire to having its own unique national identity.

The final word we will explore is *cell*. The word *cell* also has a high speed over time. However, there is a spike in the speed during the 1980s. Analyzing the nearest neighbors we see a rapid rise in similarity to *pager* and *handset*, which indicates that this spike may be related to the rapid rise of cell phone use. Additionally, this example demonstrates a weakness in our approach. Our graph shows that our model predicts that the word *cell* gradually changed meaning over time and that *cell* started changing meaning much earlier than expected. This prediction error comes from the smoothing out of the output caused by representing time as a continuous variable.

Even though we are able to extract interesting insights from the speed of word use change, Figure 4 also exhibits some limitations. In particular, most words have a sharp rise in speed in the 1930s and a steep decline in speed in the 1980s. We believe this is an artifact of our representation of word use as a function of time as there is a single time vector that influences all words. In the future, we will explore model variants to address this.

4.4 Automatic extraction of time periods

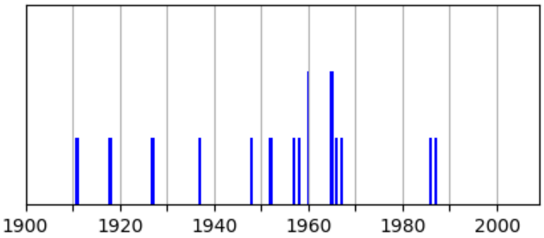


Figure 5: Distribution of time points where a node in h_1 is zero. We could interpret these points as barriers between time periods.

We can inspect h_1, the first layer in the time subnetwork, to gain further understanding of what our model is doing. We do this by analyzing the time points where a node in h_1 is zero.

As the activation function in h_1 is tanh, a node in h_1 switches from positive to negative (or vice versa) at the time points where it is zero. Thus, the time points where a node is zero should indicate barriers between time periods.

We visualize the time points where a node is zero in Figure 5. We see that we have a fairly even distribution of points until the 1940s, a large burst of points in the 1950s-1960s, and two points in the 1980s. Thus, there are many time periods before the 1940s (which may be caused by noisiness of the data in the first half of the century), a big transition between time periods in the 1950s-1960s, and a transition between time periods in the 1980s. Thus, these are time points that the model perceives as having increased semantic change.

However, there is a weakness to this analysis. Only 16% of the 100 nodes in h_1 are zero for time points between 1900 and 2009. Thus, a vast majority of nodes do not correspond to transitions between time periods.

5 Conclusion

Diachronic distributional models are a helpful tool in studying semantic shift. In this paper, we introduced our model of diachronic distributional semantics. Our model incorporates two hypotheses that better help the model capture how words change usage over time. The first hypothesis is that semantic change is gradual and the second hypothesis is that words can change usage due to common causes.

Additionally, we have developed a novel synthetic task to evaluate how accurately a model tracks the semantic shift of a word across time. This task directly measures semantic shift, is quantifiable, allows model comparison, and focuses on the trajectory of a word over time.

We have also used the fact that our model is differentiable to create a measure of the speed at which a word is changing. We then explored this measure's capabilities and limitations.

Acknowledgments

We would like to thank the University of Texas Natural Language Learning reading group as well as the reviewers for their helpful suggestions. This research was supported by the NSF grant IIS 1523637, and by a grant from the Morris Memorial Trust Fund of the New York Community Trust. We acknowledge the Texas Advanced Computing Center for providing grid resources that contributed to these results.

References

Robert Bamler and Stephan Mandt. 2017. Dynamic word embeddings. In *International Conference on Machine Learning*. pages 380–389.

David Bamman and Gregory Crane. 2011. Measuring historical word sense variation. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*. ACM, pages 1–10.

Oren Barkan. 2017. Bayesian neural word embedding. In *AAAI*. pages 3135–3143.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Association for Computational Linguistics, pages 1–10.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 136–145.

Paul Cook, Jey Han Lau, Diana McCarthy, and Timothy Baldwin. 2014. Novel word-sense identification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. pages 1624–1635.

Paul Cook, Jey Han Lau, Michael Rundell, Diana McCarthy, and Timothy Baldwin. 2013. A lexicographic appraisal of an automatic approach for detecting new word senses. *Proceedings of eLex* pages 49–65.

Ashwini Deo. 2015. Diachronic semantics. *Annu. Rev. Linguist.* 1(1):179–197.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1136–1145.

Daniel Francis. 1997. *National dreams: Myth, memory, and Canadian history*. Arsenal Pulp Press.

Lea Frermann and Mirella Lapata. 2016. A Bayesian Model of Diachronic Meaning Change. *Transactions of the Association for Computational Linguistics* 4:31–45.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*. volume 54, page 60.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Association for Computational Linguistics, pages 67–71.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Cultural shift or linguistic drift? Comparing two computational measures of semantic change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. NIH Public Access, volume 2016, page 2116.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096* .

Douglas Harper. 2014. gay. In *Online Etymology Dictionary*. http://www.etymonline.com/word/gay.

Johannes Hellrich and Udo Hahn. 2016. Bad company-neighborhoods in neural embedding spaces considered harmful. In *COLING*. pages 2785–2796.

Johannes Hellrich and Udo Hahn. 2017. Exploring diachronic lexical semantics with JESEME. *Proceedings of ACL 2017, System Demonstrations* pages 31–36.

Adam Jatowt and Kevin Duh. 2014. A framework for analyzing semantic change of words across time. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*. IEEE Press, pages 229–238.

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* .

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*. International World Wide Web Conferences Steering Committee, pages 625–635.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2017a. Temporal dynamics of semantic relations in word embeddings: An application to predicting armed conflict participants. *arXiv preprint arXiv:1707.08660* .

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2017b. Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*. pages 31–36.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*. pages 2177–2185.

Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the Google Books Ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics, pages 169–174.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Eitan Adam Pechenick, Christopher M Danforth, and Peter Sheridan Dodds. 2015. Characterizing the Google Books corpus: Strong limits to inferences of socio-cultural and linguistic evolution. *PloS one* 10(10):e0137041.

Christian Rohrdantz, Annette Hautli, Thomas Mayer, Miriam Butt, Daniel A Keim, and Frans Plank. 2011. Towards tracking semantic change by visual analytics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pages 305–310.

Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2009. Semantic density analysis: Comparing word meaning across time and phonetic space. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*. Association for Computational Linguistics, pages 104–111.

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

Hinrich Schütze. 1992. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*. pages 113–120.

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, pages 35–40.

Yang Xu and Charles Kemp. 2015. A computational evaluation of two laws of semantic change. In *CogSci*.

A Hyperparameters and preprocessing details

We used data from the English Fiction section of the Google Books ngram corpus (Lin et al., 2012). We use the English fiction specifically, because it is less unbalanced than the full English section and less influenced by technical texts (Pechenick et al., 2015). We only use the years 1900 to 2009 as there is limited data before 1900. Both the set of target words and the set of context words are the top 100,000 words by average frequency across the decades as generated by Hamilton et al. (2016b). We take a sampling approach to generating word vectors, so the corpus was converted into a list of (target word, context word, year, frequency) tuples. Frequency is the expected number of times the target word is in context of the context word that year. As the number of texts published since 1900 has increased five fold, we weigh the the frequencies so that the sums across each year are equal.

For every model, the representation of a word's use at time t is a 300 dimensional vector. For $SmallBinReg$, α_1 is set to 1000 and α_2 is set to 1. This choice of hyperparameters comes from Bamler and Mandt (2017). For $DiffTime$, every hidden layer is 100 dimensional, except for $embed_W(w)$ which is 300 dimensional.

We trained each method using random minibatching with 10,000 samples each iteration and 990 epochs total. For LargeBin, since our study spans 11 decades (1900-2009), the synchronic model for each decade is trained for 99 epochs. For SmallBinPreInit and SmallBinReg, since our study spans 110 years, the synchronic model for each year is trained for 9 epochs.

B BLESS class preprocessing

BLESS Class	Size
bird	10
building	7
clothing	10
fruit	7
furniture	8
ground_mammal	17
tool	12
tree	6
vehicle	6
weapon	7

Table 3: BLESS classes with the number of elements in each class after our preprocessing.

In this section, we discuss the BLESS preprocessing details. In the original dataset, there are 200 words categorized into 17 classes. However, we remove words that do not rank in the top 20,000 by frequency in any decade in our training data to ensure that the synthetic words do not lack context words at a given time. We then remove BLESS classes with less than 6 members to ensure that there are a sufficient number of words in each class. See Table 3 for the resulting list of BLESS classes and the number of members of each class.