Learning Diachronic Word Embeddings with Iterative Stable Information Alignment

Zefeng Lin[1,2,3], Xiaojun Wan[1,2], and Zongming Guo[1]

[1] Institute of Computer Science and Technology, Peking University
[2] The MOE Key Laboratory of Computational Linguistics, Peking University
[3] Center for Data Science, Peking University
{linzefeng,wanxiaojun,guozongming}@pku.edu.cn

Abstract. Diachronic word embedding aims to reveal the semantic evolution of words over time. Previous works learned word embeddings in different time periods first, and then aligned all the word embeddings into a same vector space. Different from previous works, we iteratively identify stable words, meanings of which remain acceptably stable even in different time periods, as anchors to ensure the performances of both embedding learning and alignment. To learn word embeddings in the same vector space, two different cross-time constraints are used during training. Initially, we identify the most obvious stable words with an unconstrained model, and then use *hard constraint* to restrain them in related stable time periods. In the iterative process, we identify new stable words from previously trained model and use *soft constraint* on them to fine-tune the model. We use COHA dataset [4][14], which consists of texts from 1810*s* to 2000*s*. Both qualitative and quantitative evaluations show our model can capture meanings in each single time period accurately and model the changes of word meaning. Experimental results indicate that our proposed model outperforms all baseline methods in terms of diachronic text evaluation.

Keywords: linguistic change · diachronic word embedding · lexical semantics

1 Introduction

With the influence of technology, culture, as well as policy, words keep evolving all the time. Traditional word embedding [22] does not consider the influence of time. Hence, mistakes are easily to be made, especially on the words that have different meanings over different time periods. For example, "gay" used to mean "cheerful" but people nowadays use it as "homosexual".

More and more researchers realized the importance of time in NLP tasks, which can improve the performance of many tasks, especially those time-related ones. Embedding-based methods for learning word changes usually consist of two steps: first pretraining embeddings using time-specific corpus separately and

[4] https://corpus.byu.edu/coha/

then making alignment between the embeddings. In the alignment process, [26] used orthogonal Procrustes to align the learned embedding. [24] solved a least squares problem to find a similar linear transformation. [28] used the embedding trained at time t to initialize the embeddings at time $t + 1$. The problems of these methods include:

- It is intractable to find the desired linear transformation due to the large embedding size and vocabulary size. Moreover, the degree and content of change for each word is different.
- They can only handle two adjacent time periods simultaneously. For words that have a long-lasting meaning, this long-time stable information should also be taken into consideration.
- The text resources are not sufficient for each time period, which is especially serious for texts in early times. Without sufficient training dataset, it is hard to learn high-quality word embeddings.

To solve problems mentioned above, we extend the skip-gram model to adapt the diachronic situation. In our model, the embedding learning and the alignment process are combined together. The alignment is accomplished by stable words, whose meanings have insignificant changes over time, which means there is no breakage in different time periods. For example, the meaning of "America" is stable over years. "American president", on the other hand, may be used to mean different presidents depending on which period referred to. However, since in most cases it remains stable during presidential tenure of one president, it could also be used as stable word at that particular period. In the diachronic embedding space, stable words should be close in terms of meaning, which builds the bridge between different time periods. Our model can handle all time periods simultaneously under two cross-time constraints. During the learning process, we use the embeddings of stable words at time $t + 1$ to predict the context of those stable words at time t, which enlarges the training texts of word vectors at time $t + 1$. The process is constructed by several iterations. We use stable words identified from previous iteration to fine-tune current embeddings, which help to get new embeddings and new stable words.

The main contributions of our study are summarized as follows:

- We propose a new method of learning diachronic word embedding. Instead of performing embeddings learning and mapping separately, we make alignment during the process of embedding learning.
- We introduce stable words to make alignment across different time periods. During the process, if a word w is regarded as stable through time t_a to time t_b, we push embeddings of word w from t_a to time t_b to become closer.
- We evaluate our proposed model from both qualitative and quantitative perspectives. In task 7 of SemEval 2015 - diachronic text evaluation, our model achieves significantly better results compared to other existing methods.

2 Related Work

There are many researches investigating linguistic changes.

Some of them are based on probabilistic model. For instance, [4] proposed dynamic topic model, which learned the evolution of latent topics over time. [12] proposed a Bayesian model to learn the diachronic meaning changes. [13] proposed dynamic Bernoulli embeddings for language evolution. [23] introduce a novel dynamic Bayesian topic model for semantic change. The evolution was based on distributional information of lexical nature as well as genre.

Some works are intending to build diachronic word embedding [1,29]. [21]. They trained word embeddings on two corpora separately, and then made vectors comparable by transformation. [28] trained the skip-gram model on the annual corpus and initialized the corresponding word embeddings next year using the word embeddings from the previous year. [27] used frequent terms as anchors to find the transformation matrix. In the model proposed by [18], embeddings are connected through a latent diffusion process. [16] proposed an EM algorithm that jointly learned the projection and identified the noisy pairs. They demonstrated the effectiveness on both bilingual and diachronic word embedding.

Based on diachronic word embedding, temporal word analogy aims to find which word w_1 at time t_α is similar to word w_2 at time t_β [9]. [15] focused on capturing global social shifts. [25] used diachronic embedding to detect semantic changes.

Many researches also use changes in the co-occurrence of words or PMI matrix as a tool to discover culture and societal trends [11,7,10,20,5]. Internet linguistics focus on the language changes in media and how they are influenced by the Internet and teen language [6,19,8,3].

3 Proposed Model

3.1 Framework

Our framework includes two parts, initial stage and iterative stage, which is summarised in Figure 1.

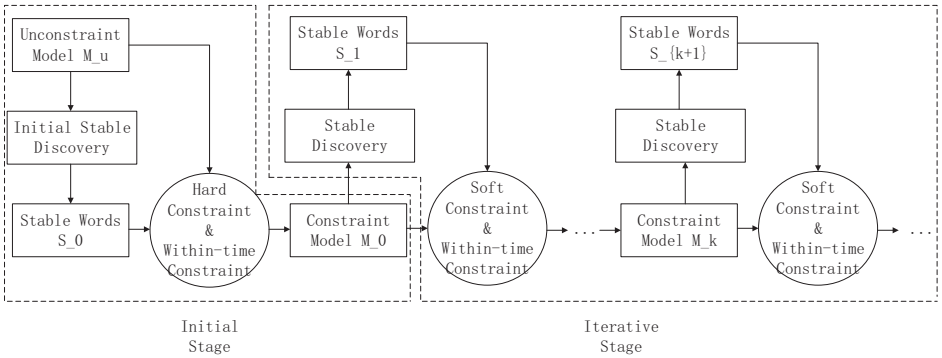

Fig. 1. The framework of our proposed model.

In the initial stage, we find stable words S_0 from unconstrained model M_u. In the iterative stage, we seek new stable words S_{k+1} from the previous model M_k. After fine-tuning the model M_k by stable words S_{k+1}, we get new model M_{k+1}.

We take a subset of the corpus from time $t-1$ to $t+2$ as an example to show the process of building models in Figure 2.

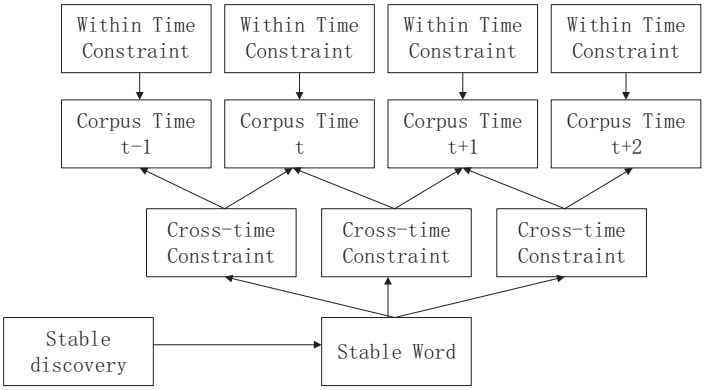

Fig. 2. The process of building models.

The within-time constraint is applied to every single time period, and the cross-time constraint is applied to adjacent times periods based on stable words, which build up the link between different time periods. We will discuss within-time constraint and cross-time constraint (including hard constraint and soft constraint) below.

3.2 Initial Stage

Initial Stable Word Discovery

The goal of diachronic word embedding is to put embeddings from different time periods into the same vector space. In the all-time space, embeddings of stable words from different time periods keep close. If a word changes its meaning, the embedding at new time is far away from the embedding at the original time. The distance relationship within stable words can be used as the guidance of alignment.

How to identify stable words is quite tricky. Embeddings at each time periods are in different spaces before alignment. This means it is impossible to use vectors to calculate distance and identify stable words directly. Therefore, we use statistical method to calculate stable score. We first train unconstrained model from each corpus of time separately. Then, we calculate the neighbors of each word in the separate space of each time and get the intersection of neighbors in adjacent times. More words in the intersection, more stable the original

word is. This is a sufficient and unnecessary condition of stable words and the algorithm-selected stable words is the subset of all stable words in reality.

We use formulas to illustrate the process. The first step is to calculate the intersection of words listed between adjacent time periods and find the potential suitable words:

$$W_{(t,t+1)} = W_t \cap W_{t+1} \tag{1}$$

$$U_t = W_{(t-1,t)} \cup W_{(t,t+1)} \tag{2}$$

W_t denotes words appearing in time t, $W_{(t,t+1)}$ denotes words appearing in both time t and time $t+1$ and U_t denotes words in time t which also appear in either time $t-1$ or time $t+1$. We calculate neighbors of words in U_t by:

$$N_{(w_i,t)} = \underset{w_j \in W_t, w_j \neq w_i}{\arg top\ n \min} \sqrt{\left\| v_{(w_i,t)} - v_{(w_j,t)} \right\|^2} \qquad \text{for } w_i \in U_t \tag{3}$$

$$N_{(w_i,t)} = \{w_j^{\text{1st most similar}}, w_j^{\text{2nd most similar}}, \ldots, w_j^{n-\text{th most similar}}\} \tag{4}$$

$v_{(w_i,t)}$ denotes the vector of word w_i at time t. "$\arg top\ n \min$" denotes the top n words that have the smallest distance. $N_{(w_i,t)}$ denotes the set of top n smallest words from $w_j^{\text{1st most similar}}$ to $w_j^{n-\text{th most similar}}$. It also denotes the similar neighbors of word w_i at time t. Then we calculate the intersection of neighbors in adjacent times by:

$$C_{(w_i,t,t+1)} = N_{(w_i,t)} \cap N_{(w_i,t+1)} \tag{5}$$

$C_{(w_i,t,t+1)}$ denotes the intersection of most similar words of w_i at time t and $t+1$. If the number of $C_{(w_i,t,t+1)}$ is larger than a threshold value, the word w_i will be chosen as stable word during time t and time $t+1$. Time t and time $t+1$ build a stable time period for stable word w_i.

Within-time Constraint

The basic requirement of diachronic word embedding is that they should hold the relations of embeddings for each time as previous unconstrained model. So we learn word embeddings of each time by the skip-gram model, and the loss is defined as follows:

$$LS_1 = -\sum_{t=1}^{T} \sum_{i=1}^{L_t} \sum_{k=-n}^{n} \log p(w_{(i+k,t)} | w_{(i,t)}) \tag{6}$$

T is the number of time periods. $w_{(i,t)}$ is the word of i at time t. k is a word in the context (a window size of $2n$) of word i. L_t is the total number of words in the corpus at time t.

Hard Constraint

In initial stage, stable words are identified from the unconstrained model. Those words are most obviously stable. We directly reduce the distance of two word vectors called "hard constraint". The distance is made by Euclidean distance as follows:

$$LS_2 = \sum_{w_q \in S} \sum_{(t_i,t_j) \in T_{w_q}} \sqrt{\left\| v_{(w_q,t_i)} - v_{(w_q,t_j)} \right\|^2} \tag{7}$$

$v_{(w_q,t_n)}$ denotes the vector of w_q at time t_n. w_q denotes one of the stable words. T_{w_q} denotes the set of stable time pairs (t_i, t_j), which builds the stable time period of the stable word of w_q . S is the list of stable words. The total loss is the linear combination of LS_1 and LS_2.

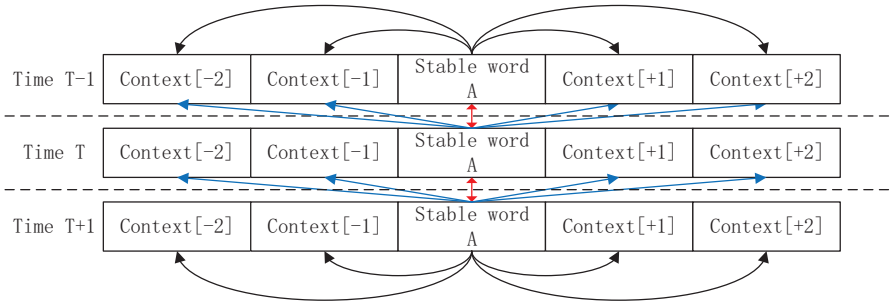

Fig. 3. Red line denotes hard constraint, blue line denotes soft constraint and black curved line denotes within-time constraint.

3.3 Iterative Stage

Iterative Stable Word Discovery

Iteratively training under constraints of stable words brings time-related information. As a result, we can identify new stable words from the model trained on the previous step with a stable discovery algorithm discussed above.

Soft Constraint

Then, we use soft constraint on new stable words to fine-tune word embeddings. Instead of directly controlling the distance of two vectors, soft constraint uses vector of a stable word at time t_j to predict its context words at time t_i if the word keeps stable from time t_i to time t_j. The loss is:

$$LS_3 = -\sum_{w_q \in S} \sum_{(t_i,t_j) \in T_{w_q}} \sum_{k=-n}^{n} \log p(w_{(q+k,t_i)} | w_{(q,t_j)}) \tag{8}$$

$w_{(q+k,t)}$ is a word in the context (a window size of $2n$) of the word $w_{q,t}$ at time t. w_q is the stable word and S is a list of stable words.

As the blue line shown in Figure 3, we only use vectors to predict the former context. Time is unidirectional. If a word remains stable, it means that its meaning is similar to that in the former time period. At the same time, they may be possible to be influenced by the new meanings. Soft constraint not only builds connections between stable words and the context in former time period, but also brings two time periods of stable words closer. Furthermore, they enlarge the size of training texts.

The iterative process is described as below:

ALGORITHM 1: Diachronic word embedding with constraints.

Input: Text of 20 decades, Max iteration N
Output: Word embeddings of 20 decades
$iteration\_index = 0$; $Max\_iteration =$ N;
Learn unconstrained models separately;
Find initial stable words S_0 from unconstrained model;
Learn embedding $E_{1,1}$ to $E_{1,20}$ with stable words S_0; using within-time constraint and hard constraint; $iteration\_index$ ++;
repeat
  Find stable words $S_{iteration\_index}$ from embedding $E_{iteration\_index,1}$ to $E_{iteration\_index,20}$;
  Learn embedding $E_{iteration\_index+1,1}$ to $E_{iteration\_index+1,20}$ with stable words $S_{iteration\_index}$; using within-time constraint and soft constraint;
  $iteration\_index$ ++;
until $iteration\_index > Max\_iteration$;

4 Evaluation

4.1 Dataset and Parameters

The corpus we use is COHA[14], which consists of texts from 1810 to 2000, including fiction, non-fiction, newspapers, and magazines. And we choose decade as the granularity of time period .

We build the model with the gensim tool[5]. As for the parameters, we set the embedding size to 300, window size to 5 and the number of negative samples to 5. In stable discovery, the number of neighbors for testing is 10 and the threshold in both initial stage and iterative stage is 5. And the total number of iteration is 30.

4.2 Qualitative Evaluation

History in Diachronic Word Embedding

[5] https://radimrehurek.com/gensim/

History in Diachronic Word Embedding	
	Neighbors of word "war" in Single Time Space
1830s	**revolution**, contest, struggle, peace, hostilities, **France**, battle
1870s	contest, wars, **France**, battle, hostilities, rebellion, Revolution
1910s	struggle, conflict, crisis, victory, **Allies**, battle, hostilities
1930s	War, **depression**, peace, crisis, wars, conflict, struggle, battle
1970s	conflict, struggle, **Vietnam**, fighting, battle, wars, **terror**
2000s	battle, **Iraq**, democracy, **Saddam**, disaster, **terrorists**, **9/11**
Evolution in Diachronic Word Embedding	
	Neighbors of word "war" in All Time Space
1830s	revolution(1830s), war(1840s), contest(1830s), struggle(1830s)
1870s	war(1860s), war(1880s), war(1890s), contest(1870s)
1910s	war(1920s), struggle(1910s), conflict(1910s), crisis(1910s)
1930s	war(1920s), War(1930s), depression(1930s), war(1940s)
1970s	war(1960s), conflict(1970s), struggle(1970s), war(1980s)
2000s	battle(2000s), Iraq(2000s), democracy(2000s), Saddam(2000s)
Semantic Change in Diachronic Word Embedding	
	Change of word "gay"
1850s	joyous, merry, pleasant, cheerful, **happy**, merriest, graceful
1930s	happy, merry, **bright**, excited, alluring, pleasant, sweet
1940s	pleasant, friendly, happy, excited, cheerful, bright, **charming**
1980s	**bisexual**, **homosexual**, heterosexual, gifted, lesbian
1990s	**lesbian**, feminist, antiabortion, conservative, **homosexual**l
	Change of word "energy"
1850s	strength, **vigor**, activity, sagacity, ability, force, skill, **ardor**
1930s	**vitality**, **energies**, imagination, strength, intelligence, ability
1940s	**power**, substance, **radium**, imagination, material, energies
1980s	**oil**, waste, economy, **fuel**, force, power, wealth, water, tax
1990s	**electricity**, power, **fuel**, oxygen, **gas**, calcium, ethanol

Table 1. Qualitative Evaluation of Diachronic Word Embedding.

We can learn history from diachronic word embedding. As presented in Table 1, using "war" as example, there are mainly two types of neighbors. One is associated with its commonly acknowledged meaning, such as "battle" and "conflict". The other is associated with history. "war" in 1830s reveals the July **Revolution** in **France**. "war" in 1910s reveals the World War I, which **Allies** participated in. **"9/11"** event began the U.S. war on **terror**. They attacked **Iraq** and defeated **Saddam** in 2000s.

Evolution in Diachronic Word Embedding

Evolution of words is a smooth process. From Table 1, the meaning of "war" in 1930s is similar to "war" in the 1920s and 1940s and the meaning in 1970s is similar to "war" in 1960s and 1980s. This, to some extent, shows that words evolve over time in our diachronic word embedding is relatively smooth.

Semantic Change in Diachronic Word Embedding

The similarity results from diachronic word embedding show that the meaning of "gay" changed from "happy", "charming" to "homosexual". In early times, "energy" was more commonly used to describe human. Then, the meaning of "fuel" increased.

4.3 Quantitative Evaluation

Compared Models

We compare our proposed model with following baselines:

- **SCAN model[12]** used the Bayesian model to build the diachronic model.
- **HISTWORD[26]** trained word vectors separately and then found orthogonal alignment matrices between adjacent years[6]. They released three groups of embeddings trained on COHA, lemma of COHA and Google books n-gram[7].
- **Dynamic Word Embeddings[30]** is a dynamic statistical model which learned time-aware word vector representation by solving a joint optimization problem.
- **Unconstrained model** is a model without cross-time alignment on the same diachronic corpus. The comparison between unconstrained model and our model aims to evaluate the effectiveness of cross-time constraint.
- **Our model** is trained on fiction, non-fiction, newspapers, and magazines separately, which aims to evaluate the effectiveness on different types of literature.

Task Description

Diachronic text evaluation is Task 7 of Semeval 2015[8]. We choose subtask 2 for evaluation. This task aims to predict the time when the text was most likely written.

The inputs of this multi-class classification task are the document vectors [2]. The document vectors at time t are built by the average of word embeddings in the text. The final document vector is the concatenation of the all-time document vectors. After building the document vector, we classify it with random forest. All parameters are same among different models.

There are two evaluation criteria. Precision(P) refers to the percentage of results which are perfectly true. The score takes time distance into consideration. The loss of different distance from 0 to bigger than 9 are 0, 0.1, 0.15, 0.2, 0.4, 0.5, 0.6, 0.8, 0.9 and 0.99 [17], respectively. The final score is:

$$Score = 1 - \frac{sum(loss)}{count(all)} \tag{9}$$

[6] https://nlp.stanford.edu/projects/histwords/
[7] https://books.google.com/ngrams/
[8] http://alt.qcri.org/semeval2015/task7/

Diachronic Text Evaluation						
	6-years		12-years		20-years	
Model	Precision	Score	Precision	Score	Precision	Score
SCAN [[12]]	0.053	0.376	0.091	0.572	0.135	0.719
SVM SCAN [[12]]	0.331	0.573	0.368	0.667	0.428	0.790
Dynamic Word Embeddings[[30]]	0.365	0.559	0.385	0.666	0.431	0.783
HISTWORD coha-word [[26]]	0.364	0.592	0.388	0.695	0.448	0.802
HISTWORD coha-lemma [[26]]	0.346	0.561	0.368	0.669	0.428	0.787
HISTWORD eng-fiction [[26]]	0.338	0.562	0.360	0.672	0.420	0.784
unconstrained model [fiction]	0.347	0.559	0.366	0.672	0.415	0.788
unconstrained model [non-fiction]	0.343	0.565	0.368	0.673	0.425	0.790
unconstrained model [magazine]	0.343	0.572	0.365	0.683	0.419	0.790
unconstrained model [news]	0.324	0.557	0.344	0.674	0.401	0.791
Our model [fiction] (Iteration 30)	0.379	**0.698**	0.411	**0.799**	0.490	**0.876**
Our model [non-fiction] (Iteration 30)	0.379	0.668	0.410	0.771	0.484	0.856
Our model [magazine] (Iteration 30)	**0.390**	0.687	**0.420**	0.786	**0.493**	0.868
Our model [news] (Iteration 30)	0.372	0.667	0.403	0.767	0.478	0.855

Table 2. Result of Diachronic text evaluation.

Where $sum(loss)$ means the amount of all losses for the prediction results and $count(all)$ denotes the number of results. The higher the score, the better the performance.

Results and Discussions

The better result on diachronic text evaluation means the model can capture the word meanings each time and the changes cross time periods more accurately. From Table 2, our proposed model achieves the best score and precision.

Unconstrained model can study the meaning of each word under every individual time period, but it is hard to capture the change cross time. Comparing to other methods such as SCAN model, Dynamic Word Embeddings and HISTWORD model, our model captures word changes better with the help of constraints. The high performance consistently achieved in all types of literature proves the universality of our model.

5 Conclusion

In order to solve the problem of modeling word changes, we introduce a novel method to train diachronic embeddings. Unlike former works which first trained separately and then found a transformation matrix, we make alignment during the process of embedding learning. We propose two cross-time constraints and iteratively extract stable words from corpus as anchors to build diachronic constraints.

We evaluate our embeddings qualitatively and quantitatively. In the qualitative evaluation, the embedding of our model can reflect the history of different time periods, show the smooth process of evolution and capture the word

changes. During the Diachronic Text Evaluation, our trained embedding achieves significantly better results in both scores and precision.

References

1. Andrei Kutuzov, Erik Velldal, and Lilja Øvrelid. Tracing armed conflicts with diachronic word embedding models. Association for Computational Linguistics, 2017.
2. Chiraag Lala. Word vector-space embeddings of natural language data over time. 2014.
3. David Crystal. *Internet linguistics: A student guide*. Routledge, 2011.
4. David M. Blei and John D. Lafferty. Dynamic topic models. In *Proc. International Conference on Machine Learning*, pages 113–120, 2006.
5. Derry Tanti Wijaya and Reyyan Yeniterzi. Understanding semantic change of words over centuries. In *Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web*, pages 35–40. ACM, 2011.
6. Diane J Schiano, Coreena P Chen, Ellen Isaacs, Jeremy Ginsberg, Unnur Gretarsdottir, and Megan Huddleston. Teen use of messaging media. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 594–595. ACM, 2002.
7. Gerhard Heyer, Florian Holz, and Sven Teresniak. Change of topics over time-tracking topics by their change of meaning. *KDIR*, 9:223–228, 2009.
8. Guy Merchant. Teenagers in cyberspace: An investigation of language use and language change in internet chatrooms. *Journal of Research in Reading*, 24(3):293–306, 2001.
9. Guy D Rosin, Eytan Adar, and Kira Radinsky. Learning word relatedness over time. *arXiv preprint arXiv:1707.08081*, 2017.
10. Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.
11. Kristina Gulordava and Marco Baroni. 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*, pages 67–71. Association for Computational Linguistics, 2011.
12. Lea Frermann and Mirella Lapata. A bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, 4:31–45, 2016.
13. Maja Rudolph and David Blei. Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052*, 2017.
14. Mark Davies, Irén Hegedűs, and Alexandra Fodor. The 400 million word corpus of historical american english (1810–2009). In *English Historical Linguistics 2010: Selected Papers from the Sixteenth International Conference on English Historical Linguistics (ICEHL 16), Pécs, 23-27 August 2010*, volume 325, page 231. John Benjamins Publishing, 2012.
15. Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.
16. Noa Yehezkel Lubin, Jacob Goldberger, and Yoav Goldberg. Aligning vector-spaces with noisy supervised lexicons. *arXiv preprint arXiv:1903.10238*, 2019.

17. Octavian Popescu and Carlo Strapparava. Semeval 2015, task 7: Diachronic text evaluation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 870–878, 2015.
18. Robert Bamler and Stephan Mandt. Dynamic word embeddings via skip-gram filtering. *stat*, 1050:27, 2017.
19. Sali A Tagliamonte and Derek Denis. Linguistic ruin? lol! instant messaging and teen language. *American speech*, 83(1):3–34, 2008.
20. Sunny Mitra, Ritwik Mitra, Martin Riedl, Chris Biemann, Animesh Mukherjee, and Pawan Goyal. That's sick dude!: Automatic identification of word sense change across different timescales. *arXiv preprint arXiv:1405.4392*, 2014.
21. Terrence Szymanski. Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 448–453, 2017.
22. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
23. Valerio Perrone, Marco Palma, Simon Hengchen, Alessandro Vatri, Jim Q Smith, and Barbara McGillivray. Gasc: Genre-aware semantic change for ancient greek. *arXiv preprint arXiv:1903.05587*, 2019.
24. Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee, 2015.
25. William L Hamilton, Jure Leskovec, and Dan Jurafsky. 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*, volume 2016, page 2116. NIH Public Access, 2016.
26. William L Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*, 2016.
27. Yating Zhang, Adam Jatowt, Sourav S Bhowmick, and Katsumi Tanaka. The past is not a foreign country: Detecting semantically similar terms across time. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2793–2807, 2016.
28. Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515*, 2014.
29. Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. Discovery of evolving semantics through dynamic word embedding learning. arxiv preprint. *arXiv preprint arXiv:1703.00607*, 2017.
30. Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 673–681. ACM, 2018.