A Comprehensive Analysis of PMI-based Models for Measuring Semantic Differences

Taichi Aida[1], **Mamoru Komachi**[1], **Toshinobu Ogiso**[2], **Hiroya Takamura**[3] **and Daichi Mochihashi**[4]
[1]Tokyo Metropolitan University
[2]National Institute for Japanese Language and Linguistics
[3]National Institute of Advanced Industrial Science and Technology
[4]The Institute of Statistical Mathematics
aida-taichi@ed.tmu.ac.jp, komachi@tmu.ac.jp, togiso@ninjal.ac.jp
takamura.hiroya@aist.go.jp, daichi@ism.ac.jp

Abstract

The task of detecting words with semantic differences across corpora is mainly addressed by word representations such as word2vec or BERT. However, in the real world where linguists and sociologists apply these techniques, computational resources are typically limited. In this paper, we extend an existing simultaneously optimized model that can be trained on CPU to perform this task. Experimental results show that the extended models achieved comparable or superior results to strong baselines in English corpora and SemEval-2020 Task 1, and also in Japanese. Furthermore, we compared the training time of each model and conducted a comprehensive analysis of Japanese corpora.[1]

1 Introduction

Words can have different meanings at different times and domains. For example, the word *meat* means *food* in Old English but *animal meat* in Modern English; the word *interface* means a *boundary surface*, but in the domain of computer science, it means *software that allows users to communicate with computers*. The task of detecting words with semantic differences provides significant insights into human language (Kutuzov et al., 2018); for instance, linguists often discuss the semantic differences between words in different corpora, such as written and spoken Japanese (Fujimura et al., 2012), British and American English (Lei and Liu, 2014), native speakers and learners of English (McEnery et al., 2019), or web-crawled and traditional corpora in Czech (Cvrček et al., 2020). Automatic methods can facilitate such analyses comprehensively, and can also help lexicographers describe when and how the meanings of words change substantially.

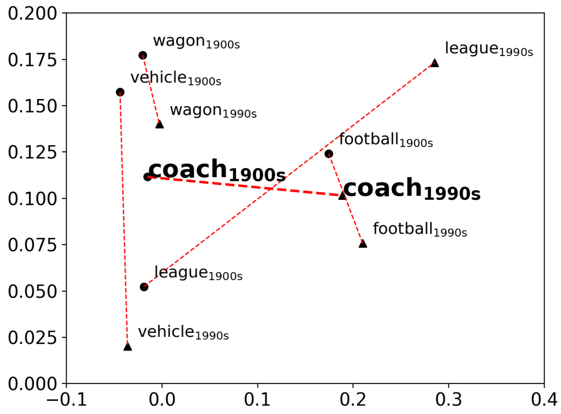


Figure 1: Diachronic differences in the meaning of *coach* and its neighbors, identified by our extended model.

Recent progress in representation learning has provided a useful tool for finding semantic differences in words, known as word embeddings. Figure 1 shows an example of the two-dimensional word embedding space. In this figure, $\overrightarrow{coach}_{1900s}$ and $\overrightarrow{coach}_{1990s}$ are shown in the learned vector space. The shift in the meaning of the word *coach* can be analyzed from the distance between $\overrightarrow{coach}_{1900s}$ and $\overrightarrow{coach}_{1990s}$. Such embeddings are often obtained by training word vectors independently from the corpora of the 1900s and the 1990s, and then aligning them (Kim et al., 2014; Kulkarni et al., 2015; Hamil-

[1] The source code is available at `https://github.com/a1da4/pmi-semantic-difference`

ton et al., 2016). These alignment-based methods learn distributional semantic models efficiently because they use non-contextual word representations, such as word2vec (Mikolov et al., 2013). Thus, researchers can easily introduce them without abundant computational resources (Sommerauer and Fokkens, 2019; Zimmermann, 2019). However, alignment-based methods are based on the strong assumption that they align word representations from different time periods or domains linearly to one another. Recent studies have proposed alignment-independent methods (Yao et al., 2018; Dubossarsky et al., 2019), but existing approaches to this task involve the following problems.

First, one of the alignment-independent methods requires an extensive hyperparameter search. Yao et al. (2018) proposed a model that did not assume linear alignments based on simultaneous learning of word representations. However, as shown later, it includes three sensitive hyperparameters that need to be tuned, which incurs a complex combinatorial optimization problem.

Second, a properly chosen list of target words is not available in a realistic scenario. Dubossarsky et al. (2019) proposed a simultaneously optimized word representation that overcame the abovementioned problems with another simple but not necessarily correct assumption that words other than the target word do not change over time. This model, called Temporal Referencing, is easy to train on CPU without any assumptions on linear transformations, and without an extensive hyperparameter search. However, in the real world, linguists and sociologists may not have a well-selected list of target words.

Third, only a few studies have quantitatively compared each method (Schlechtweg et al., 2019; Shoemark et al., 2019; Tsakalidis and Liakata, 2020; Schlechtweg et al., 2020). The comparisons have been mainly conducted in European languages such as English or German; only a few studies have evaluated these methods in multiple languages (Schlechtweg et al., 2020) due to the lack of evaluation data. In these analyses, many studies cite target words whose meanings have clearly changed, such as the well-known semantic shift of the word *gay* (Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016; Hu et al., 2019). However, few studies have focused on semantic changes of more ordinary words (Gonen et al., 2020).

In this paper, we address these issues. For the first two problems, we modified Temporal Referencing. We first considered all the words in the vocabulary as the target words, to reflect more realistic scenarios. We then proposed an extended model that allows context vectors to change across corpora. For the third problem, we conducted a quantitative comparison between the extended method and strong baselines not only in English and SemEval-2020 Task 1 (Schlechtweg et al., 2020), but also in Japanese. In the experiments, we compared the task performance and training time of each model. To address the lack of evaluation data, we used pseudo words whose meanings were artificially changed (Rosenfeld and Erk, 2018; Shoemark et al., 2019). In the analyses, we focused on ordinary words as well as words with well-known semantic shifts in Japanese.

The contributions of this paper are as follows.

- We extend the existing simultaneously optimized model that can be trained on CPU regarding the real situation.
- Experiments on multiple languages using actual or pseudo-words show that the extended methods learn faster and perform similar to or better than strong baselines.
- We conduct comprehensive analyses, and the experimental results demonstrate that the extended method achieves better results for words with well-known semantic shifts, and detects semantic differences between corpora for words that are not widely known.

2 Related Work

Semantic differences are often detected by comparing word frequencies between corpora (Fujimura et al., 2012; Lei and Liu, 2014; McEnery et al., 2019; Cvrček et al., 2020); however, manually checking all text to be processed is not a straightforward or facile task. Therefore, several automatic methods have been proposed to detect semantic differences across times or domains.

Several studies have been conducted to detect synchronic differences using data from social media. Zhao et al. (2011) compared Twitter and traditional media based on topic modeling. Aoki et al. (2017) used word2vec as a language model to detect non-standard usages from context words in web corpora. Gonen et al. (2020) proposed a metric that compared the nearest neighbors of each target word vector and analyzed the differences in word usage of different age groups on social media. Here, we focus on methods that capture diachronic meaning differences.

2.1 Non-contextual word embeddings

The task of detecting diachronic semantic difference refers to the task of finding words that have different meanings in corpora with different time periods (e.g., SemEval-2020 Task 1). A standard approach involves the comparison of the vectors of the same word over different time periods (e.g., $\overrightarrow{coach}_{1900s}$ and $\overrightarrow{coach}_{1990s}$ in Figure 1).

Early studies on this task often used count-based methods to obtain word vectors for each time period (Sagi et al., 2009; Cook and Stevenson, 2010; Gulordava and Baroni, 2011). However, count-based methods cannot directly model word meanings. Mikolov et al. (2013) proposed word2vec, which solved the abovementioned problem by embedding word meanings into a vector space. To detect semantic difference between corpora, the vector spaces for each corpus must be aligned with one another. For this purpose, Kim et al. (2014) proposed to set the initial word vectors at time t to the corresponding word vectors learned from the corpus for time $t-1$ to train a word2vec model at time t. Then, Kulkarni et al. (2015) and Hamilton et al. (2016) proposed alignment methods with a linear transformation and a rotation, respectively. For each target word w, Kulkarni et al. (2015) used a linear transformation $\mathbf{R}(w)_{t\mapsto t+1}$ to align a target word vector $\mathbf{W}_t(w)$ to an adjacent vector space $\mathbf{W}_{t+1}(w)$. $\mathbf{R}(w)_{t\mapsto t+1}$ was obtained by solving a piecewise linear regression among $\mathbf{W}_t(w)$'s k-nearest neighbors k-NN$(\mathbf{W}_t(w))$:

$$\mathbf{R}(w)_{t\mapsto t+1} = \operatorname*{argmin}_{\mathbf{R}} \sum_{s \,\in\, k\text{-NN}(\mathbf{W}_t(w))} ||\mathbf{W}_t(s)\mathbf{R} - \mathbf{W}_{t+1}(s)||_F^2, \tag{1}$$

where $||\cdot||_F$ is the Frobenius norm. Conversely, Hamilton et al. (2016) introduced a rotation matrix $\mathbf{R}_{t\mapsto t+1}$ to map word representations $\mathbf{W}_t$ to $\mathbf{W}_{t+1}$, which was obtained by solving the orthogonal procrustes problem:

$$\mathbf{R}_{t\mapsto t+1} = \operatorname*{argmin}_{\mathbf{R}:\ \mathbf{R}\mathbf{R}^{\mathsf{T}}=1} ||\mathbf{W}_t\mathbf{R} - \mathbf{W}_{t+1}||_F^2. \tag{2}$$

Alignment-based methods have achieved improved performance compared to count-based methods (Schlechtweg et al., 2019). However, they are based on a strong assumption that word representations are linearly aligned with one another, which might not hold in the actual situations.

By contrast, Yao et al. (2018) proposed a model called Dynamic Word Embeddings (DWE) that relaxed the constraint of linear alignment. They did not use any transformations for learning word representations across time periods. Instead, they were learned simultaneously. The word representations $\mathbf{W}_t$ were obtained by minimizing the following objective function using context representations $\mathbf{C}_t$ and word-context positive pointwise mutual information (PMI) matrices $\mathbf{M}_t$:

$$\begin{aligned} &\frac{1}{2}\sum_{t=1}^{T} ||\mathbf{M}_t - \mathbf{W}_t\mathbf{C}_t||_F^2 + \frac{\gamma}{2}\sum_{t=1}^{T} ||\mathbf{W}_t - \mathbf{C}_t^{\mathsf{T}}||_F^2 \\ &+ \frac{\lambda}{2}\sum_{t=1}^{T} ||\mathbf{W}_t||_F^2 + \frac{\tau}{2}\sum_{t=1}^{T-1} ||\mathbf{W}_{t+1} - \mathbf{W}_t||_F^2 \\ &+ \frac{\lambda}{2}\sum_{t=1}^{T} ||\mathbf{C}_t||_F^2 + \frac{\tau}{2}\sum_{t=1}^{T-1} ||\mathbf{C}_{t+1} - \mathbf{C}_t||_F^2, \end{aligned} \tag{3}$$

where γ, λ, and τ are hyperparameters. The parameters γ and τ control the strengths of alignments, and λ controls the strength of regularization. This model assumes that the vectors of the same word in the same time period $(\mathbf{W}_t, \mathbf{C}_t^{\mathsf{T}})$ were close; the vectors of the same word at adjacent time points $(\mathbf{W}_t, \mathbf{W}_{t+1})$, $(\mathbf{C}_t, \mathbf{C}_{t+1})$ were also close. Therefore, the model was sensitive to hyperparameters, and an extensive hyperparameter search was required.

2.2 Contextual word embeddings

Contextual word embeddings, such as BERT (Devlin et al., 2019), can also be used for the task of semantic difference detection. However, methods based on contextual word embeddings have been reported to exhibit lower performance than those based on non-contextual word embeddings in

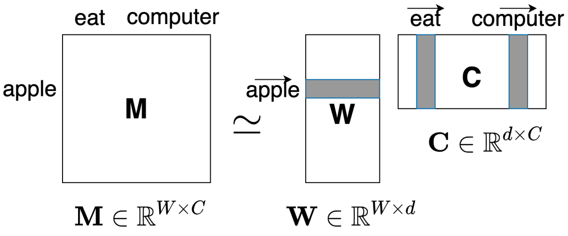

eat
computer
apple
M
$\simeq$
$\overrightarrow{\text{apple}}$
W
$\overrightarrow{\text{eat}}$
$\overrightarrow{\text{computer}}$
C
$\mathbf{C} \in \mathbb{R}^{d \times C}$
$\mathbf{M} \in \mathbb{R}^{W \times C}$
$\mathbf{W} \in \mathbb{R}^{W \times d}$

Figure 2: Overview of PMI-SVD (Levy and Goldberg, 2014) that acquires a word representation $\mathbf{W}$ from the matrix factorization of a PMI matrix by SVD.

the SemEval-2020 Task 1 (Kutuzov and Giulianelli, 2020; Martinc et al., 2020b). Contextual word embeddings are mainly used for polysemous word analysis over time, which cannot be performed using non-contextual word embeddings. Hu et al. (2019) trained each usage-level vector of each word from example sentences in a dictionary using BERT. They tracked each sense of polysemous words, such as *gay*, which can mean either *carefree* or *homosexual* depending on context. Instead of using dictionaries, Giulianelli et al. (2020) performed *k*-means clustering on all token-level vectors obtained by BERT. Their method also provided semantic transitions of polysemous words without any lexicographic supervision.

3 Method: Jointly Optimized Word Representations

Base idea: Temporal Referencing As described in Section 2.1, existing methods involve two problems. First, alignment-based methods (Equations (1) and (2)) are based on the strong assumption that word representations from different periods or domains can be linearly aligned to one another. Second, DWE (Equation (3)) incurs optimizing combinatorial number of its hyperparameters. To address these problems, Dubossarsky et al. (2019) proposed a jointly optimized word representation called Temporal Referencing. This method is based on an assumption that words other than the target word do not change over time. Given a target word list $L = \{w^1, w^2, ..., w^{|L|}\}$, the authors trained a model by distinguishing the target words over time $\{w^i_1, ..., w^i_t, ..., w^i_T | w^i \in L\}$. However, in the real world, there is often no list of well-chosen target words. In this paper, we propose two extensions to Temporal Referencing: (1) considering all words in the vocabulary as target words, and (2) allowing context vectors to change across corpora.

Base model: PMI-SVD We first explain the underlying model introduced by Levy and Goldberg (2014). They show that the model of skip-grams with negative sampling (SGNS) (Mikolov et al., 2013) is equivalent to the factorization of a matrix consisting of PMI between each word and its surrounding context words, as shown in Figure 2. Let $p(w)$, $p(c)$, and $p(w, c)$ denote empirical probabilities of word w, context word c, and their cooccurrence, respectively. Word representations can be learned as follows. First, a PMI matrix[2] $\mathbf{M} \in \mathbb{R}^{W \times C}$ (W and C indicate the total numbers of target words and context words, respectively) is computed.

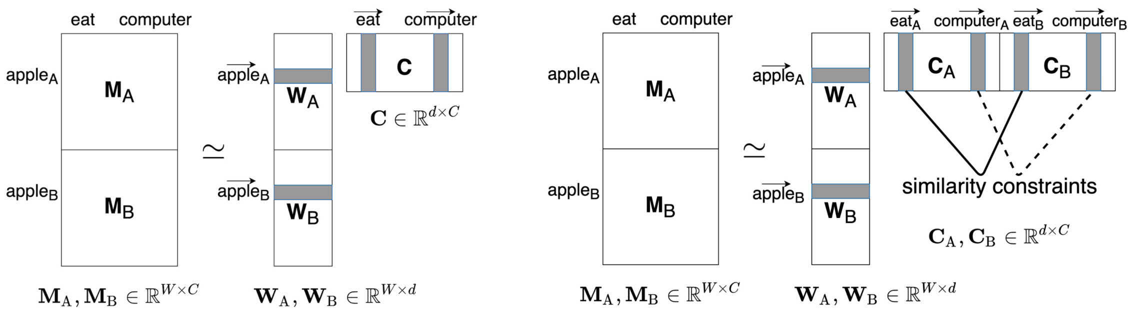
$$\mathrm{M}[w, c] = \max\left(\log \frac{p(w, c)}{p(w)p(c)}, 0\right) \tag{4}$$

Then, $\mathbf{M}$ is decomposed as $\mathbf{M} = \mathbf{U\Sigma V}^\mathsf{T}$ through singular value decomposition (SVD), where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma}$ is a diagonal matrix consisting of singular values of $\mathbf{M}$. Based on this factorization, a d-dimensional matrix $\mathbf{W} \in \mathbb{R}^{W \times d}$ of word vectors and a matrix $\mathbf{C} \in \mathbb{R}^{d \times C}$ of context vectors are obtained by $\mathbf{M} = \mathbf{WC}$ as shown in Figure 2, where $\mathbf{W}$ and $\mathbf{C}$ are computed by $\mathbf{W} = \mathbf{U\Sigma}^{1/2}$ and $\mathbf{C} = \mathbf{\Sigma}^{1/2}\mathbf{V}^\mathsf{T}$.

PMI-SVD$_{\text{joint}}$: To modify Temporal Referencing, we consider all words in the vocabulary as target words. We assume that the context vectors represented by each column in $\mathbf{C}$ are fixed across corpora A and B, in line with the existing approach. Based on this assumption, we can perform matrix factorization on $\mathbf{M} = [\mathbf{M}_\mathrm{A}; \mathbf{M}_\mathrm{B}]$, which are vertically stacked PMI matrices $\mathbf{M}_\mathrm{A}$ and $\mathbf{M}_\mathrm{B}$ for corpora A and B (Figure 3(a)).

$$\begin{bmatrix}\mathbf{M}_\mathrm{A}\\ \mathbf{M}_\mathrm{B}\end{bmatrix} = \begin{bmatrix}\mathbf{W}_\mathrm{A}\\ \mathbf{W}_\mathrm{B}\end{bmatrix}\begin{bmatrix}\mathbf{C}\end{bmatrix}. \tag{5}$$

[2] SGNS has been shown to be equivalent to a shifted version of PMI. However, because Levy et al. (2015) showed that it had no performance benefit in the case of PMI matrix factorization, we simply discarded the shift and used the original PMI.


eat
computer
apple$_A$
$\mathbf{M}_A$
apple$_B$
$\mathbf{M}_B$
$\simeq$
$\overrightarrow{\text{apple}}_A$
$\mathbf{W}_A$
$\overrightarrow{\text{apple}}_B$
$\mathbf{W}_B$
$\overrightarrow{\text{eat}}$
$\overrightarrow{\text{computer}}$
$\mathbf{C}$
$\mathbf{C} \in \mathbb{R}^{d\times C}$
$\mathbf{M}_\mathrm{A}, \mathbf{M}_\mathrm{B} \in \mathbb{R}^{W\times C}$
$\mathbf{W}_\mathrm{A}, \mathbf{W}_\mathrm{B} \in \mathbb{R}^{W\times d}$
$\overrightarrow{\text{eat}}_A$
$\overrightarrow{\text{computer}}_A$
$\overrightarrow{\text{eat}}_B$
$\overrightarrow{\text{computer}}_B$
$\mathbf{C}_A$
$\mathbf{C}_B$
similarity constraints
$\mathbf{C}_\mathrm{A}, \mathbf{C}_\mathrm{B} \in \mathbb{R}^{d\times C}$

(a) PMI-SVD$_{\text{joint}}$

(b) PMI-SVD$_c$

Figure 3: Overview of the slightly modified Temporal Referencing (Dubossarsky et al., 2019) (left) and the extended model (right). They acquired word representations $\mathbf{W}_\mathrm{A}$ and $\mathbf{W}_\mathrm{B}$ for each corpus using the matrix factorization of PMI matrices by SVD.

PMI-SVD$_c$: The method introduced above is based on the assumption that context word vectors remain unchanged across corpora. We relax this assumption to propose a model that allows the vectors of context words to change, as in Figure 3(b). In contrast to PMI-SVD$_{\text{joint}}$, context representations $\mathbf{C}_\mathrm{A}$ and $\mathbf{C}_\mathrm{B}$ are also computed in the decomposition of the stacked PMI matrix $\mathbf{M}$. One straightforward method to obtain word and context embeddings is to factorize $\mathbf{M}_y$ in each corpus y. However, the word vectors obtained for different corpora would not correspond to each other. Hence, we added an additional constraint that the context representations of adjacent corpora are close to each other. Therefore, the objective function used to obtain word representations $\mathbf{W}_y$ is as follows:

$$\sum_{y\in\{A,B\}} \|\mathbf{M}_y - \mathbf{W}_y\mathbf{C}_y\|_F + \tau\|\mathbf{C}_\mathrm{B} - \mathbf{C}_\mathrm{A}\|_F, \quad (6)$$

where τ is the only hyperparameter that controls the strength of the constraint. This model seems close to DWE, but our model has only one hyperparameter, whereas DWE has three hyperparameters with an exponential number of combinations. Moreover, we show later that PMI-SVD$_c$ achieved the same or better performance experimentally than DWE, yet it runs several orders of magnitude faster than DWE.

4 Preliminary Experiment: Detecting Semantic Change from a List of Words

We performed the SemEval-2020 Task 1 using PMI-SVD$_c$. The SemEval-2020 Task 1 has two subtasks: one is a binary classification task that detects whether or not the meanings of target words have changed, and the other is a ranking task that sorts the target words by the degree of change in meaning. Classification was evaluated by accuracy, and ranking was evaluated using the Spearman's rank correlation coefficient. For overall performance, the average over the four languages (English, German, Latin, and Swedish) were evaluated.

Task		Oracle	PMI-SVD$_c$	
Classification (Accuracy)	Avg	0.713	0.645	(5)
	En	0.676	0.649	(4)
	De	0.750	0.667	(10)
	La	0.650	0.650	(4)
	Sv	0.774	0.613	(16)
Ranking (Spearman)	Avg	N/A	0.433	(6)
	En	N/A	0.424	(2)
	De	N/A	0.597	(9)
	La	N/A	0.328	(10)
	Sv	N/A	0.328	(11)

Table 1: Results for the extended model PMI-SVD$_c$ in the SemEval-2020 Task 1. Oracle used an optimal threshold for classification in each language.

In our models for SemEval-2020 Task 1, we mainly used the cosine similarity between two time periods of each target word. For classification, we used the average cosine similarity of the target words as the threshold for each language. We used the optimal threshold for classification in each language as an oracle, similarly to previous reports that used a test set to adjust hyperparameters. For ranking, the target words were ranked in ascending order of the cosine similarity.

From Table 1, we confirmed that our model worked consistently across the four languages. At the oracle, the model was able to achieve high performance, with an average score of 0.713.

5 Experiments: Detecting Semantic Change from All Words

In this section, we describe the experimental setup and results of quantitative and qualitative evaluations performed on English and Japanese.

5.1 Data and preprocessing

English: We used the Corpus of Historical American English (COHA)[3]. We selected documents from the 1900s and 1990s. After removing stopwords and proper nouns, we chose nouns, verbs, adjectives, and adverbs that appeared more than 100 times in both documents, following (Hamilton et al., 2016). We regarded the chosen words as target words.

Japanese: We used the Corpus of Historical Japanese (CHJ) and the Showa-Heisei Corpus of Written Japanese[4]. We merged these two corpora and split them into two periods based on World War II because the Japanese language has changed significantly since that war. Target words were selected similar to the experiments in English.

5.2 Models

We compared the model with minor modifications (PMI-SVD$_{\mathrm{joint}}$) and our extended model (PMI-SVD$_c$) with the following previous methods. For all non-contextual word representations, we used a window size of 4, 100 dimensions, and contextual distributional smoothing of 0.75. Then, we performed a post-processing called all-but-the-top (Mu and Viswanath, 2018) simultaneously for the representation of each period (Kaiser et al., 2021).

Word2Vec$_{\mathbf{align}}$ (Hamilton et al., 2016): We trained word2vec SGNS models on different time periods separately. Then, we aligned these models with a rotation matrix using Equation (2).

PMI-SVD$_{\mathbf{align}}$ (Hamilton et al., 2016): We trained PMI-SVD models instead. Subsequently, these models were aligned similarly with Word2Vec$_{\mathrm{align}}$.

DWE (Yao et al., 2018): In line with a previous study, we minimized Equation (3) with block coordinate descent to obtain word representations. To find the best setting for this model and PMI-SVD$_c$, a grid search was performed out of seven values 10^x, $-3 \leq x \leq 3$ for each hyperparameter by taking the hyperparameters with the highest AUC.

BERT (Martinc et al., 2020a): Target word vectors in each period were obtained by averaging usage-level vectors computed by a BERT model. For both languages, we used pre-trained *bert-base-uncased* models published in the Huggingface[5].

5.3 Evaluation

To evaluate the proposed approach, we computed the mean reciprocal rank (MRR) (Kulkarni et al., 2015; Yao et al., 2018). Each model first ranked all words in the vocabulary in ascending order of the cosine similarity between the two periods. Subsequently, MRR is computed as the average of the inverse of the rank of each word in a reference list that contains words with known semantic change. The Spearman's rank correlation coefficient used in SemEval-2020 Task 1 could not be used because the evaluation lists of the words were not annotated with the degree of semantic change.

To visualize the detection of words with changed meanings in the reference list, we calculated the recall with top-k words and a reference list called Recall@k (Kulkarni et al., 2015).

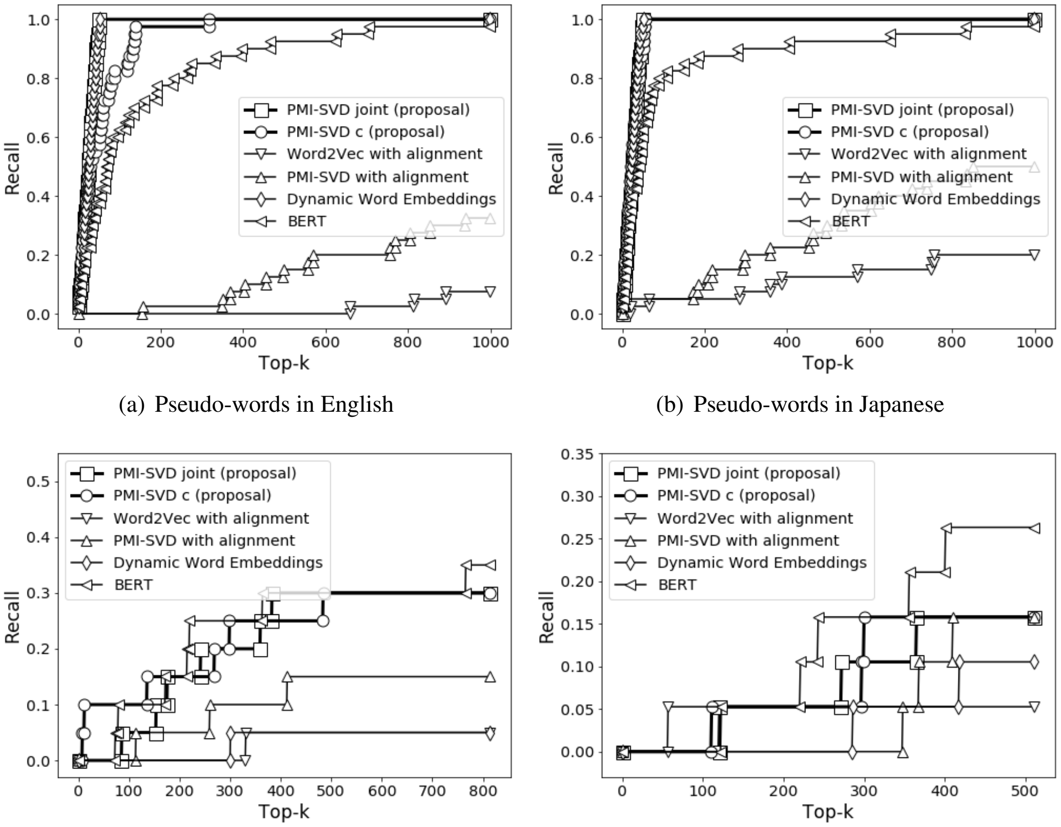
5.4 Quantitative results on pseudo-words

Settings For a theoretical investigation, we generated words with semantic changes artificially, similar to Shoemark et al. (2019). The pseudo-word α, whose meaning changes from α to β, was generated following by replacing of all occurrences of the word β in the last time period with α and deleting the original occurrence. In this paper, we randomly sampled 50 pairs of words whose absolute cosine similarity of word vectors was 0.01 or less in both periods. We used 10 words for the hyperparameter search and the rest for evaluations.

[3] https://www.english-corpora.org/coha/

[4] https://ccd.ninjal.ac.jp/chj/overview-en.html

[5] https://github.com/huggingface/transformers



(a) Pseudo-words in English

(b) Pseudo-words in Japanese

(c) Actual words in English

(d) Actual words in Japanese

Figure 4: Plot of Recall@k for words that have changed semantically. For English and Japanese, reference lists of words with semantic changes (see text) were employed.

Models	English	Japanese
PMI-SVD$_{\text{joint}}$	**0.0933**	0.0737
PMI-SVD$_c$	0.0870	0.0781
Word2Vec$_{\text{align}}$	0.0004	0.0022
PMI-SVD$_{\text{align}}$	0.0010	0.0171
DWE	0.0835	**0.0913**
*BERT**	*0.0590*	*0.0776*

Table 2: Mean Reciprocal Rank (MRR) in pseudo-words. *Using external datasets in pre-training.

Models	English	Japanese	Time
PMI-SVD$_{\text{joint}}$	0.00186	0.00131	**2m58s**
PMI-SVD$_c$	**0.01045**	0.00120	26m01s
Word2Vec$_{\text{align}}$	0.00040	**0.00137**	6m22s
PMI-SVD$_{\text{align}}$	0.00100	0.00091	3m26s
DWE	0.00047	0.00058	30h20m
*BERT**	*0.00250*	*0.00163*	2h23m
BERT-tiny	0.00100	0.00078	12days
BERT-mini	0.00135	0.00119	2weeks

Table 3: Mean Reciprocal Rank (MRR) in actual words. The time indicates the training time for each model in the English experiment. BERT models (BERT-tiny, BERT-mini) were trained from scratch using diachronic corpora. *Using external datasets in pre-training.

The extended methods vs. baselines Figures 4(a) and 4(b) show the Recall@k of each language. Our models perfectly detected pseudo-words with semantic change in the reference list, as in DWE and BERT. These figures and MRR (Table 2) show that our models performed better than or comparable to the existing models.

Linear alignment The linear alignment (Hamilton et al., 2016) performed poorly in this experiment where the words were completely changed in meaning. Therefore, we conclude that the assumption that separately trained models can be aligned by a linear transformation is too strong.

	BERT		PMI-SVD$_c$	
rank	word	description	word	description
1	若く	comparable, young → young	行い	behavior → behavior, execute
2	触れ	fall, mention, violate → mention, touch	かねて	before → before, simultaneous
3	行い	behavior → behavior, execute	おまけ	in addition → in addition, discount
4	公明	fairness → [organization], fairness	無論	[adverb] → [adverb]
5	思い	thinking, emotion → thinking	年中	year around, officer → year around
6	削除	delete → delete	キー	music, [person] → music, key
7	在り	physical existence → conceptual existence	欠け	missing → lack
8	参議	participate → [organization]	皆無	nothing → nothing
9	欠け	missing → lack	馬場	[person], turf → [person], turf
10	幼稚	childish → kindergarten, childish	反面	opposite, while → while

Table 4: Top 10 actual words with the smallest cosine similarity that have changed semantically in Japanese. We excluded single-character words that are less meaningful.

5.5 Quantitative results on actual words

Settings Next, we evaluated each model using actual words. For English, we used the word sense change testset[6] for the hyperparameter search and the list of Kulkarni et al. (2015) for the evaluation. For Japanese, we used the list of words with semantic differences by Mabuchi and Ogiso (2021) for both the hyperparameter search and the evaluation.

Proposed methods vs. baselines The performance is shown in Figures 4(c) and 4(d), and Table 3. Overall, the results were worse than those obtained with the use of pseudo-words. According to these figures and MRR (Table 3), PMI-SVD$_c$ outperformed previous works with the exception of Word2Vec$_{\text{align}}$ and BERT in Japanese. In addition, Table 3 shows that PMI-SVD$_c$ is computationally more efficient than DWE and BERT[7].

Pre-training BERT from diachronic corpus We mainly used BERT-base models (12 layers, 768 hidden sizes) pre-trained with huge amounts of data. In this part, we trained BERT models from scratch with the diachronic corpora used in Section 5.1. Due to the small amount of diachronic corpora, the availability of which is limited, we trained BERT-tiny (2 layers, 128 hidden sizes) and BERT-mini (4 layers, 256 hidden sizes) models. Table 3 shows that our models perform better than BERT-tiny and BERT-mini when they were trained with the same amount of data. Moreover, our models required only minutes to hours to train on CPU, as opposed to BERT models, which require tremendous computational resources and more than two weeks to train from scratch.

5.6 Qualitative results

Top-10 words found by BERT and PMI-SVD$_c$ We compared the top-10 words with the highest degree of semantic differences sorted by the cosine similarity in each of BERT and the proposed method (PMI-SVD$_c$), which performed the best in a quantitative evaluation (Section 5.5). In Japanese, Table 4 shows that both methods included ordinary words with semantic differences like "行い (behavior)" and "欠け (missing)." In particular, BERT generally captured semantic-level differences, such as "若く (young)," "触れ (touch)," "在り (existence)," and "幼稚 (childish)," and PMI-SVD$_c$ captures syntactic-level differences such as "おまけ (in addition)" and "反面 (while)." This may be attributed to the difference in the window size; BERT creates a word vector from an entire sentence, whereas the proposed method creates a word vector from the information obtained from surrounding words.

Analyzing (non-)famous words Next, we compared neighbors of each word (Kim et al., 2014; Hamilton et al., 2016). Again, we compared BERT and PMI-SVD$_c$. We investigated a famous word "了解 (understand)" in the list of Mabuchi and Ogiso (2021) and the ordinary word "欠け (missing)" in Table 4. Tables 5(a) and 5(b) show the top-5 similar words, "了解" and "欠け," in the prewar and

[6] https://zenodo.org/record/495572

[7] The time was measured on a machine with 2 CPUs (Intel Xeon 2.60 GHz, with a total of 56 cores) and 512 GB of RAM.

(a) 了解 (*understand*→*consent*)

BERT		PMI-SVD$_c$	
prewar	postwar	prewar	postwar
承諾	承諾	理解	承諾
(consent)	(consent)	(understand)	(consent)
承知	承知	納得	承知
(consent)	(consent)	(understand)	(consent)
納得	承認	推測	納得
(understand)	(consent)	(estimation)	(understand)
理解	同意	判断	同意
(understand)	(agreement)	(decision)	(agreement)
断定	納得	断定	理解
(conclusion)	(understand)	(decision)	(understand)

(b) 欠け (*missing*→*lack*)

BERT		PMI-SVD$_c$	
prewar	postwar	prewar	postwar
マイナス	欠如	切り	有し
(minus)	(lack)	(cut)	(have)
決まり	乏しい	切ら	欠如
(rule)	(poor)	(cut)	(lack)
構え	不足	諦め	富ん
(posture)	(lack)	(give up)	(rich)
重み	崩れ	箸	づけ
(weight)	(collapse)	(fleeting)	(attach)
当て	破れ	つける	把握
(aim)	(tear)	(attach)	(grasp)

Table 5: Top-5 similar words for each period. We excluded single-character words that are less meaningful.

the postwar sets using BERT and PMI-SVD$_c$. First, considering the word "了解," both methods found words with the meanings *understand* ("納得 (understand)," and "理解 (understand)") in the prewar, and *consent* ("承諾 (consent)," "承知 (consent)," "承認 (consent)," and "同意 (agreement)") in the postwar (Table 5(a)). However, BERT found some words that have a meaning *consent* in the prewar ("承諾 (consent)" and "承知 (consent)"). Second, in the case of the word "欠け," both methods yielded words such as *missing* ("マイナス (minus)," "切り (cut)," or "切ら (cut)") in the prewar, and words meaning *lack* ("欠如 (lack)" and "不足 (lack)") in the postwar (Table 5(b)). From these results, PMI-SVD$_c$ detected differences in the meanings of words between corpora, even for word that are not widely known.

6 Conclusion

We have extended an existing simultaneously optimized method to address real-world situations in which there is no target word list or abundant computational resources are available for semantic change detection. For a theoretical investigation, we conducted quantitative evaluations to measure diachronic meaning differences with pseudo- and actual word lists in two languages. Experimental results show that our extended methods can be learned faster, required less hyperparameter search, and achieved better or comparable performances than strong baselines. In the future work, we plan to apply these models to different domains, such as books and social media datasets.

Acknowledgments

The work reported in this article was supported by the NINJAL collaborative research project "Statistic and Statistical Mechanics Approach to Semantic Drift of Words in Modern and Contemporary Languages," NINJAL Diachronic Corpus project, and JSPS KAKENHI Grant Number 19H00531.

References

Tatsuya Aoki, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2017. Distinguishing Japanese non-standard usages from standard ones. In *EMNLP 2017*, pages 2323–2328.

Paul Cook and Suzanne Stevenson. 2010. Automatically identifying changes in the semantic orientation of words. In *LREC 2010*, pages 28–34.

Václav Cvrček, Zuzana Komrsková, David Lukeš, Petra Poukarová, Anna Řehořková, Adrian Jan Zasina, and Vladimír Benko. 2020. Comparing web-crawled and traditional corpora. *LRE*, 54(3):713–745.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019*, pages 4171–4186.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-out: Temporal referencing for robust modeling of lexical semantic change. In *ACL 2019*, pages 457–470.

Itsuko Fujimura, Shoju Chiba, and Mieko Ohso. 2012. Lexical and grammatical features of spoken and written Japanese in contrast: Exploring a lexical profiling approach to comparing spoken and written corpora. In *GSCP 2012*, pages 393–398.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *ACL 2020*, pages 3960–3973.

Hila Gonen, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, interpretable and stable method for detecting words with usage change across corpora. In *ACL 2020*, pages 538–555.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *GEMS 2011*, pages 67–71.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *ACL 2016*, pages 1489–1501.

Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *ACL 2019*, pages 3899–3908.

Jens Kaiser, Sinan Kurtyigit, Serge Kotchourko, and Dominik Schlechtweg. 2021. Effects of pre- and post-processing on type-based embeddings in lexical semantic change detection. In *EACL 2021*, pages 125–137.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *LTCSS 2014*, pages 61–65.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *WWW 2015*, pages 625–635.

Andrey Kutuzov and Mario Giulianelli. 2020. UiO-UvA at SemEval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In *SemEval 2020*, pages 126–134.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *COLING 2018*, pages 1384–1397.

Lei Lei and Zehua Liu. 2014. A word type-based quantitative study on the lexical change of American and British English. *JQL*, 21(1):36–49.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *NIPS 2014*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Yoko Mabuchi and Toshinobu Ogiso. 2021. An attempt to construct a dataset of words for semantic change analysis of modern Japanese. In *ANLP 2021*, pages 1166–1170.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020a. Leveraging contextual embeddings for detecting diachronic semantic shift. In *LREC 2020*, pages 4811–4819.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020b. Discovery team at SemEval-2020 task 1: Context-sensitive embeddings not always better than static for semantic change detection. In *SemEval 2020*, pages 67–73.

Tony McEnery, Vaclav Brezina, Dana Gablasova, and Jayanti Banerjee. 2019. Corpus linguistics, learner corpora, and SLA: Employing technology to analyze language use. *ARAL*, 39:74–92.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR 2013*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *ICLR 2018*.

Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. In *NAACL 2018*, pages 474–484.

Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2009. Semantic density analysis: Comparing word meaning across time and phonetic space. In *GEMS 2009*, pages 104–111.

Dominik Schlechtweg, Anna H"atty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *ACL 2019*, pages 732–746.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *SemEval 2020*, pages 1–23.

Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. 2019. Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. In *EMNLP-IJCNLP 2019*, pages 66–76.

Pia Sommerauer and Antske Fokkens. 2019. Conceptual change and distributional semantic models: an exploratory study on pitfalls and possibilities. In *LChange 2019*, pages 223–233.

Adam Tsakalidis and Maria Liakata. 2020. Sequential modelling of the evolution of word representations for semantic change detection. In *EMNLP 2020*, pages 8485–8497.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *WSDM 2018*, page 673–681.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing Twitter and traditional media using topic models. In *ECIR 2011*, pages 338–349.

Richard Zimmermann. 2019. Studying semantic chain shifts with Word2Vec: FOOD>MEAT>FLESH. In *LChange 2019*, pages 23–28.