Fake it Till You Make it: Self-Supervised Semantic Shifts for Monolingual Word Embedding Tasks

Maurício Gruppi, Pin-Yu Chen, Sibel Adalı
1 Rensselaer Polytechnic Institute, NY, USA
2 IBM Research, NY, USA
gouvem@rpi.edu, pin-yu.chen@ibm.com, adalis@rpi.edu

Abstract
The use of language is subject to variation over time as well as across social groups and knowledge domains, leading to differences even in the monolingual scenario. Such variation in word usage is often called lexical semantic change (LSC). The goal of LSC is to characterize and quantify language variations with respect to word meaning, to measure how distinct two language sources are (that is, people or language models). Because there is hardly any data available for such a task, most solutions involve unsupervised methods to align two embeddings and predict semantic change with respect to a distance measure. To that end, we propose a self-supervised approach to model lexical semantic change by generating training samples by introducing perturbations of word vectors in the input corpora. We show that our method can be used for the detection of semantic change with any alignment method. Furthermore, it can be used to choose the landmark words to use in alignment and can lead to substantial improvements over the existing techniques for alignment. We illustrate the utility of our techniques using experimental results on three different datasets, involving words with the same or different meanings. Our methods not only provide significant improvements but also can lead to novel findings for the LSC problem.

1 Introduction
Language use is deeply rooted in the social, cultural and historical context that shapes it. It has been shown that meaning of words change over time, being pushed by cultural and societal transformations, a phenomenon named semantic change (Schmidt 1963). For example, the English word awful was used in the sense of the words impressive or majestic before the year 1800, while, in modern English, it describes something objectionable. Language is also subject to variation across different communities for many different reasons (Schlechtweg et al. 2019). For example, the word model can be used to refer to the design or version of a product (as in model of a car), or it could be used to refer to a mathematical model in a scientific paper.

In this paper, we develop a novel self-supervised semantic shift (S4) method to detect words with different meaning in two corpora from the same language (monolingual).

As methods based on the distributional property of words have been shown to be very effective in encoding semantic relationship between words (Hamilton, Leskovec, and Jurafsky 2016b; Sagi, Kaufmann, and Clark 2009; Bamber and Mandt 2017; Kulkarni et al. 2015) or even biases and stereotypes (Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017), the task of identifying semantic change between words using word embeddings, such as Word2Vec or FastText (Mikolov et al. 2013; Bojanowski et al. 2017) has gained a great deal of popularity. This task is often a difficult one as it involves unsupervised methods (e.g. learning embeddings, alignment and/or mapping of words). For example, in the recent SemEval-2020 competition (Schlechtweg et al. 2020), the highest scores were at about 70% accuracy on a binary classification task to predict occurrence of semantic change across time periods in several languages. The main challenge stems from the unsupervised nature of the problem, as training data is rare or non-existing, and is highly dependent on the input corpora. This impacts multiple aspects of the task. In particular, to compare the embedding matrices $A, B \in \mathbb{R}^{n \times d}$ of two separate corpora $\mathcal{A}$ and $\mathcal{B}$, where $n$ is the size of the common vocabulary and $d$ is the embedding dimension, one must first align them to make them comparable, usually via an Orthogonal Procrustes (OP) method. The goal of OP is to learn an orthogonal transform matrix $Q$ (i.e., $Q^T Q = I_d$, where $I_d$ is the d-dimensional identity matrix), that most closely maps $A$ to $B$, namely, $Q^* = \arg \min_{Q} \| Q^T Q - I_d \| \| AQ - B \|$. It has been shown that this problem accepts a closed-form solution via Singular Value Decomposition (SVD) (Schönemann 1953). The fact that $Q$ is an orthogonal matrix makes it so that $AQ$ is only subject to unitary transformations such as reflection and rotation, preserving the inner product between its word vectors. Words whose vectors are used in the OP are called landmarks (or anchors), these are the words over which we will enforce proximity in the alignment by minimizing the distance of its vectors.

Any landmark choice incorporates an initial assumption to the solution. An ideal solution to this problem would involve a set of landmark words that are semantically stable (i.e. words that have the same sense) across the input corpora. In the context of diachronic embeddings, where the embedding is learned from adjacent time slices of text, the assumption is that words will change only slightly from time
1. We introduce the novel S4 method and demonstrate its advantage on two tasks in monolingual word embedding: Unsupervised Binary Classification, and Word Embedding Alignment. In classification, S4 simulates semantic changes and learns to predict semantically stable and unstable words via self-supervision. In alignment, S4 learns to determine the landmark set by predicting, selecting, and refining a set of stable words with self-supervision.

2. We evaluate S4-D’s classification performance on a British v.s. American English semantic change detection task. Regardless of the underlying alignment methods, S4 consistently shows significant gain over the baseline methods, attaining up to 2.65× higher F1 score.

3. For the SemEval-2020 task (Schlechtweg et al. 2020), we show landmark words learned by S4-A attains improved performance on lexical semantic change detection (with a simple linear classifier) in four different languages, demonstrating the importance of landmark selection with S4-A for downstream tasks.

4. We also use S4 for discovery of semantic changes in articles within two arXiv subjects: Artificial Intelligence and Classical Physics. We find that S4-based alignment can identify unique semantically changed words in top-ranked word lists that were overlooked by existing alignment methods, providing diversity and novel discovery of lexical semantic changes.

This paper makes the following contributions:

1. We introduce the novel S4 method and demonstrate its advantage on two tasks in monolingual word embedding: Unsupervised Binary Classification, and Word Embedding Alignment. In classification, S4 simulates semantic changes and learns to predict semantically stable and unstable words via self-supervision. In alignment, S4 learns to determine the landmark set by predicting, selecting, and refining a set of stable words with self-supervision.

2. We evaluate S4-D’s classification performance on a British v.s. American English semantic change detection task. Regardless of the underlying alignment methods, S4 consistently shows significant gain over the baseline methods, attaining up to 2.65× higher F1 score.

3. For the SemEval-2020 task (Schlechtweg et al. 2020), we show landmark words learned by S4-A attains improved performance on lexical semantic change detection (with a simple linear classifier) in four different languages, demonstrating the importance of landmark selection with S4-A for downstream tasks.

4. We also use S4 for discovery of semantic changes in articles within two arXiv subjects: Artificial Intelligence and Classical Physics. We find that S4-based alignment can identify unique semantically changed words in top-ranked word lists that were overlooked by existing alignment methods, providing diversity and novel discovery of lexical semantic changes.

Our methods have many applications in which semantic change has been shown to be important such as diachronic linguistic analysis (Hamilton, Leskovec, and Jurafsky 2016a; Dubossarsky, Weinshall, and Grossman 2017),
and predicting armed conflict participants from semantic relations (Kutuzov, Velldal, and Øvrelid 2017). More recently, Bojanowski et al. (2019) have presented a strategy using alignment of monolingual embeddings for updating language models to incorporate changes caused by language evolution or usage in specific domains.

Code for the experiments in this paper are provided in our GitHub repository.

## 2 Related Work

Some of the earliest works on diachronic semantic change discovery analyzes word usage over time with respect to frequency and sense, mostly based on the difference of the distributional property of words between time periods or domains (Sagi, Kaufmann, and Clark 2009; Cook and Stevenson 2010). Distributed word vector representations, such as the ones obtained by skip-gram with negative sampling (Mikolov et al. 2013) allow for learning distributional information into dense continuous vectors. Hamilton et al. (2016b) conducted a diachronic analysis of semantic change on historical text using word embeddings aligned with Orthogonal Procrustes and by measuring the displacement of vectors across time periods using cosine distance. To circumvent the need for alignment, Hamilton et al. (2016a) have proposed a measure of semantic change that compares second-order vectors of distance between a word and its neighbors. Some authors have presented dynamic word embedding techniques to avoid alignment by jointly learning the distributional representations across all time periods (Bamler and Mandt 2017; Rudolph and Blei 2018; Yao et al. 2018), in which words are connected across time via the assumption that the change across periods is smooth. Yin et al. (Yin, Sachidananda, and Prabhakar 2018) introduced the Global Anchor method for corpus-level adaptation, which avoids alignment altogether by using second-order distances. This method is proven to be equivalent to the global alignment method and as a result makes use of the smoothness of change assumption which may lead to over-sharing, especially in cross-domain scenarios.

Selection of landmarks as words with likely same meaning in two different languages is used in translation tasks. Artetxe et al. (2017) employ a self-supervision approach to refine a small seed dictionary with Arabic numerals. Conneau et al. (2017) use self-supervising adversarial learning to align bilingual embeddings, refining it with Orthogonal Procrustes on the best matching words. Joulin et al. (2018) propose an alternative loss function for the alignment in order to address the conflict between Euclidean alignment and cosine distance mapping and to refine the alignment by matching words that have similar frequency ranking in each language. Lubin et al. (2019) employ a landmark selection by detecting noisy pairs through an iterative EM algorithm.

Most of the previously developed methods do not present a systematic way of detecting semantic change such as in a classification problem. Our proposed method is designed explicitly for lexical semantic change tasks by matching the same word in two corpora, approaching semantic change as a binary classification problem. Our contributions are: the introduction of a self-supervised method for binary semantic change detection, a method for selecting landmark (anchor) words for alignment of semantically changed word vectors, a quantitative test set on British vs. American English for detecting words with similar and distinct senses. We compare our method both to baseline global alignment and noisy pairs methods and show that it provides gains in performance in a number of scenarios.

## 3 Self-Supervised Semantic Shift (S4)

The problem of detecting semantic change using monolingual alignment of word embeddings is defined as follows. Given two input corpora A and B, with vocabularies $V_A$ and $V_B$, let A and B be the word embedding matrices for the common vocabulary $V = V_A \cap V_B$, thus both A and B have dimensions $N \times d$, where $N$ is the size of the common vocabulary, and $d$ is the embedding dimension. A word in the common vocabulary is said to be unstable if it is used in a completely different sense between the corpora, or that has multiple senses but some of which only appear in one corpus, other words are considered stable. One common method for measuring semantic change is to use the cosine distance between two embeddings after aligning A and B on a subset of V. This problem involves two sub-tasks: detecting words with semantic change and choosing landmark words to align on. In this paper, we introduce a self-supervised method that can be used in both tasks.

Given embeddings A and B, the main goal of the self-supervision is to create a modified embedding $B'$ such that $B'$ contains a set of words that are known to be semantically shifted with respect to their meaning in B through explicit perturbations. We can generate (pseudo) training samples for a self-supervised procedure by using these modified embeddings and the fact that they are considered semantically shifted. Suppose t be the target word whose sense we want to add to another word w. We can accomplish this by replacing t with w an arbitrary number of times r in B and re-training the word embeddings, where the parameter $r \in (0, 1)$ defines the proportion of replacements with respect to the number of occurrences of t. To reduce complexity, instead of actually retraining the word embeddings, we move the vector $v_w$ towards $v_t$ by the rule $v_w \leftarrow v_w + rv_t$. This update rule is derived from the skip-gram with negative sampling model (Mikolov et al. 2013), specifically, whenever word t occurs within a neighborhood window of w, vector $v_w$ is updated with $v_w + (1 - z)rv_t$, where $z = \sigma(v_w^T v_t)$. In our perturbation, we replace $\sigma(v_w^T v_t)$ with parameter r, which is used to control the proportion in which word t is found in the neighborhood of w. The resulting word vector for w in $B'$ is now shifted towards that of t and w is now forced to become unstable regardless of its original state. By applying this process to multiple words, we are able to generate positive samples (semantically changed words) to the self-supervision. In contrast, negative samples are drawn from the set of landmarks.

---

1 S4 code repository: https://github.com/IBM/S4_semantic_shift
Algorithm 1: Pseudo-code for self-supervised semantic shift detection (S4-D). Input parameters are word embeddings \( A \) and \( B \), landmark words set \( L \), non-landmark words set \( M \), \( n \) as the number of negative and positive samples in each iteration, and \( r \) is the degree of perturbation. The output of this method is the classifier weights \( W \).

**Data:** \( A, B, L, M, n, r, maxiters \)

**Result:** Classifier weights \( W \)

1. \( W \leftarrow init_weights ; \)
2. \( i \leftarrow 0 ; \)
3. **while** \( i < maxiters \) **do**
   1. \( i \leftarrow i + 1 ; \)
   2. **// Sample negatives from \( L \) and positives from \( M \)**
   3. \( S_n = uniform_sample(L, n) ; \)
   4. \( S_p = uniform_sample(M, n) ; \)
   5. \( B' \leftarrow copy(B) ; \)
   6. **for** \( w \in S_p \) **do**
      1. \( t \leftarrow uniform_sample(M) ; \)
      2. **// Simulate change by moving \( w \) towards \( t \)**
      3. \( B'(w) \leftarrow B(w) + rB(t) ; \)
   7. **end**
   8. \( X \leftarrow [A(w), B'(w)] \forall w \in S_n \cup S_p ; \)
   9. \( Y \leftarrow [0 \text{ if } w \in S_n \text{ else } 1] \forall w \in S_n \cup S_p ; \)
10. \( W \leftarrow train(W, X, Y) ; \)
11. **end**
12. **return** \( W \)

3.1 S4-D: S4 for Semantic Change Detection

Self-supervision for semantic change detection can be used in conjunction with any method that uses a subset of words as landmarks. Given an initial alignment of \( A \) to \( B \) on a set of landmarks \( L \) (potentially using Orthogonal Procrustes), a batch of positive samples is generated from the perturbations, and negative samples are uniformly drawn from \( L \). These samples are then used to train a binary classifier to predict stable vs. unstable words. We use a single-layer neural network classifier with 100 hidden units with ReLU activation and sigmoid output. The input to the model is the concatenation of row vectors \( A(w) \) and \( B(w) \) for word \( w \). The model is trained over a predefined number of iterations \( K \) to predict \( y = 0 \) if \( w \) is stable, otherwise \( y = 1 \). A new batch of positive and negative samples is generated in every iteration. The goal is to minimize the average loss \( \frac{1}{K} \sum_{i=1}^{K} L(\hat{y}_i, y_i) \), with \( L \) as the binary cross-entropy function.

Note that, at this point, the self-supervision is done over a fixed alignment of \( A \) and \( B \) and it is trained to predict semantic change on that setup. The hyper-parameters of S4-D are: number of iterations \( K \), number of negative and positive samples to generate in each iteration \( n \) and \( m \), degree of semantic change in the perturbations \( r \). The pseudo-code for S4-D is presented in Algorithm 1.

3.2 S4-A: S4 for Alignment of Word Vectors

In this section, we present an extension of the self-supervised training to refine the landmarks based on the classifier predictions from Section 3.1, resulting in the Self-Supervised Semantic Shift Alignment (S4-A). The general idea is to use stable words for alignment by adding an extra step to each iteration in Algorithm 1. At the end of each training iteration, we update the classifiers weights \( W \) and use the updated model to predict stable/unstable words across \( A \) and \( B \), hence updating the set of landmarks \( L \) with words predicted as stable. Finally, we align \( A \) to \( B \) using the new set of landmarks with orthogonal procrustes and repeat over \( K \) iterations. This method outputs both model weight and the set of landmark words \( L \) and the set of non-landmark words \( N \). Using the final set \( L \) of landmarks, we can align \( A \) to \( B \) using orthogonal procrustes on the words in \( L \). Appendix A.1 contains the pseudo-code for this algorithm.

4 Experiments

4.1 Semantic Change Detection

**Objective** We evaluate the ability of S4-D to correctly detect the occurrence of lexical semantic change across British and American English. We designed binary classification task to evaluate the model’s performance in predicting semantically stable vs. unstable words. The full list of words is given in Appendix B.

**Data set** The corpus used for British English is the British National Corpus (BNC) XML Edition (of Oxford 2007) which contains a mix of news, fiction, and academic texts. The corpus for American English is the Corpus of Contemporary American English (COCA) (Davies 2009) which contains text from newspapers, magazines, fiction, and academic texts. Both corpora are pre-processed by removing stop words, and converting all characters to lower case, resulting in 50M tokens for BNC and 188M tokens for COCA.

**Baselines** The baseline methods used are the commonly used cosine distance-based method (Hamilton, Leskovec, and Jurafsky 2016b; Kutuzov et al. 2018; Schlechtweg et al. 2019), and the Noise-Aware method (Yehezkel Lubin, Goldberg 2019), which detects noisy word pairs. For the cosine distance (COS), we use three different thresholds for this measure, specifically we have three cosine-based classifiers at thresholds 0.3, 0.5, and 0.7, above which words are classified as semantically shifted. For Noise-Aware, we treat noisy word pairs as semantically shifted (unstable), and clean word pairs as stable, we will refer to that method as Noisy-Pairs. We compare the baseline methods to our proposed Self-Supervised Semantic Shift Detection (S4-D) with hyper-parameters \( n = 1000 \) positives samples, \( m = 1000 \) negatives samples, rate \( r = 0.25 \), trained over 100 iterations (at each iteration, a batch of 1000 positive and 1000 negative samples are generated and used to update the model’s weights). We train Word2Vec on the input corpora after pre-processing with parameters dimension 300, window size 10, minimum count 100 for British, and dimension 300, window size 10, minimum count 200 for American.
The minimum count for the American corpus is set higher due to its corpus being considerably larger. The common vocabulary contains 26064 words.

**Evaluation and Analysis** Once the embeddings \(X_A, X_B\) are learned from the COCA and BNC corpora, respectively, we get the learned matrices for the common vocabulary \(A, B \in \mathbb{R}^{n \times d}\) by selecting the rows of \(X_A\) and \(X_B\) that correspond to words in the common vocabulary. We learn a transform matrix \(Q\) by aligning \(A\) to \(B\) using a given alignment strategy. The self-supervised classifier is trained on \(A\) and \(B\) using the given alignment. Using the learned matrix \(Q\), we align \(X_B\) to \(X_A\) by doing \(X_B \leftarrow X_BQ\). Finally, we concatenate the vectors of the target words and feed it into the classifier to obtain the binary predictions (i.e. semantically stable or unstable).

The classification scores for this task (Table 1) show that S4-D displays the best accuracy when aligning on the top 10% most frequent words, it also shows high recall and F1 scores when aligning on the 10% least frequent and 5% most frequent words. The scores for S4-D are the average of 10 evaluation rounds, standard deviation is also reported in Table 1. Each evaluation round consists of one execution of the algorithm on the input data. This contrasts with the drop in performance shown by global alignment, this is likely due to the oversharing of words, which makes the separation of stable and unstable words more difficult. The alignment method is irrelevant to Noisy-Pairs since it inherently aligns the input vectors when searching for noise. Noisy-Pairs predicts a total of 24659 pairs as clean, or semantically stable. For that reason, only one pair of words from the target set is predicted as semantically shifted (positive class), which explains the precision score of 1.0 and a recall of 0.03.

Examples of stable words correctly predicted by S4-D are the British-American pairs labour/labor, defence/defense, petrol/gas, football/soccer, and queue/line. This shows we are able to not only to detect identical words but also morphological differences and synonyms. Note that some of these words were not included in the alignment due to not being in the common vocabulary. Yet, we are still able to capture their semantic similarity after the orthogonal transformation. These results show our model’s ability to generalize to words not seen in the self-supervision. Additionally, we were able to correctly predict unstable words such as chips (french fries in the US), biscuit (scone in the UK), cookie in the US, and semi (house in the US, truck in the US). Noisy-Pairs is able to correctly predict the semantically unstable words subway and yankee, and it also predicts all stable words correctly but shows a low recall score.

### 4.2 Evaluating Alignment Strategies

**Objective** We evaluate the Self-Supervised Semantic Shift Alignment (S4-A) described in Section 3.2 by the impact of assessing multiple alignment strategies on the performance of the binary classification problem from the recent SemEval-2020 task on Unsupervised Lexical Semantic Change Detection (Schlechtweg et al. 2020). The task consists of predicting the occurrence of lexical semantic change on a set of target words in four languages: English, German, Latin, and Swedish. For each language there are two input corpora \(C_1\) and \(C_2\) containing text from time periods \(t_1\) and \(t_2\), with \(t_1 < t_2\).

**Data sets** The data sets used in this experiment are provided in the aforementioned SemEval task. The corpus used for English is the Clean Corpus of Historical American English (CCHOA) (Alatrash et al. 2020), a pre-processed and lemmatized version of the Corpus of Historical American English (COHA) (Davies 2009). For this task, the corpus was split into time periods \(t_1\) with texts from years 1810 through 1860, and \(t_2\) with text from years 1960 through 2010. The German data set consists of the DTA corpus (von der Berlin-Brandenburgischen Akademie der Wissenschaften 2017) for the first time period, and a combination of the BZ and ND corpora (zu Berlin 2018a,b) for the second time period. Specifically, text in \(t_1\) pertains to years 1800 through 1899, and text in \(t_2\) pertains to years 1946 through 1990. For Latin we use the LatinSE corpus (McGillivray and Kilgarrif 2013) with time periods \(t_1\) from 200 B.C. through 0 A.D., and \(t_2\) from years 0 through 2000 (A.D.). For Swedish we use the KubHist corpus (Borin, Forsberg, and Roxendal 2012; Adesam, Dannells, and Tahmasebi 2019). Time period \(t_1\) is from 1790 through 1830, time period \(t_2\) is from 1895 through 1903. Along with each data set, a list of target words is provided for evaluation. Further details about the data sets can be found in Appendix C.

**Baselines** We include common alignment methods for comparison to S4-A. Particularly, we adopt the global alignment strategy as used in diachronic embeddings (Hamilton, Leskovec, and Jurafsky 2016b), as well as frequency based selection of landmarks, aligning at the 5% and 10% most and least frequent words (Bojanowski et al. 2017), and the Noise-Aware method for selecting landmarks.

**Evaluation and Analysis** We begin by training Word2Vec on \(C_1\) and \(C_2\), generating embeddings \(X_1\) and \(X_2\). We set the embedding dimension to 300 and use a window of size 10 for all languages. The minimum word count is 20, 30, 10, and 50 for English, German, Latin, and Swedish, respectively, these are chosen based on the amount of data provided for each language.

Let \(A \subset X_1\) and \(B \subset X_2\) be the embedding matrices for the common vocabulary terms in \(C_1\) and \(C_2\). Our experiment consists of aligning \(A\) to \(B\) using different alignment strategies and evaluating the alignments with respect to its performance in the binary classification task. Particularly, we evaluate our self-supervised alignment and compare it to doing global alignment, aligning on most and least frequent words, and selecting clean words as landmarks (Yehezkel Lubin, Goldberger, and Goldberg 2019).
in the leave-one-out tests, \( t \) is searched in \((0, 1)\) in increments of 0.1. The prediction is \( \hat{y} = 1 \) if \( P(x < X) > t \), otherwise \( \hat{y} = 0 \).

Results from this experiment are shown in Table 2. We report the accuracy on the evaluation set for each language and alignment method. Top/bot alignments are done over the top 5%/10% most/least frequent words in \( C_1 \). S4-A is able to achieve the best accuracy scores at maximum scores for English and German, matching some of the top performing scores in the post-evaluation phase of the SemEval-2020 Task 1 competition.

### 4.3 Discovery of Semantic Change

#### Objective

We conduct an experiment on the arXiv data provided by Yin et al. (2018) to show how we can use S4-A for word embedding alignment for the discovery of semantic change, and how the results differ across alignment methods. We select the subjects of Artificial Intelligence (cs.AI) and Classical Physics (physics.class-ph) and train embeddings \( A \) and \( B \), respectively, with Word2Vec (dimension 300, window size 10, minimum count 20). The embedding matrices are aligned using each alignment strategy, and the semantic shift measured by \( d_i = \| A_i Q - B_i \| \) for each word \( w_i \) in the common vocabulary, where \( Q \) is the transform matrix learned in the alignment.

#### Baselines

We compare the most semantically shifted words as discovered by the Global and Noise-Aware alignments (Hamilton, Leskovec, and Jurafsky 2016b; Yehezkel Lubin, Goldberger, and Goldberg 2019). We also compare our results to the top 3 high scoring entries from post-evaluation phase of the SemEval-2020 Task 1 competition, these methods may use distinct sets of features that go beyond just using word embeddings.

#### Evaluation and Analysis

To quantify the difference between different alignments, we measured the ranking correlation using the Spearman’s \( \rho \) coefficient of the ranked lists of words according to each method (ranked in descending order of semantic shift) at varying top-\( k \) thresholds with \( k \) in \([10, 300]\) in increments of 10. Figure 2 shows the ranked correlation coefficient between each alignment strategy. Higher values of \( \rho \) indicate that the order of semantic shift is more consistent between the two alignment strategies. These re-

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>Global</td>
<td>0.38/0.30/0.25</td>
<td>0.48/0.36/0.00</td>
<td>0.26/0.10/0.00</td>
<td>0.32/0.15/0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.45 ± 0.03</td>
<td>0.81 ± 0.06</td>
<td>0.28 ± 0.05</td>
<td>0.43 ± 0.06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>Top-5%</td>
<td>0.39/0.30/0.25</td>
<td>0.47/0.36/0.00</td>
<td>0.29/0.10/0.00</td>
<td>0.33/0.15/0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.67 ± 0.02</td>
<td>0.81 ± 0.02</td>
<td>0.79 ± 0.02</td>
<td>0.82 ± 0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>Top-10%</td>
<td>0.37/0.30/0.25</td>
<td>0.47/0.36/0.00</td>
<td>0.26/0.10/0.00</td>
<td>0.31/0.15/0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.70 ± 0.03</td>
<td>0.83 ± 0.02</td>
<td>0.74 ± 0.03</td>
<td>0.78 ± 0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>Bot-5%</td>
<td>0.40/0.29/0.24</td>
<td>0.46/0.32/0.00</td>
<td>0.31/0.10/0.00</td>
<td>0.35/0.16/0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.45 ± 0.02</td>
<td>0.62 ± 0.01</td>
<td>0.21 ± 0.02</td>
<td>0.31 ± 0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>Bot-10%</td>
<td>0.37/0.30/0.25</td>
<td>0.53/0.45/0.25</td>
<td>0.29/0.12/0.02</td>
<td>0.34/0.18/0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.65 ± 0.02</td>
<td>0.71 ± 0.01</td>
<td>0.89 ± 0.02</td>
<td>0.79 ± 0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>S4-A</td>
<td>0.44/0.34/0.28</td>
<td>0.66/0.62/0.57</td>
<td>0.40/0.18/0.06</td>
<td>0.44/0.27/0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.70 ± 0.01</td>
<td>0.72 ± 0.01</td>
<td>0.93 ± 0.01</td>
<td>0.81 ± 0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>S4-A</td>
<td>0.44/0.34/0.28</td>
<td>0.66/0.62/0.57</td>
<td>0.40/0.18/0.06</td>
<td>0.44/0.27/0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.70 ± 0.01</td>
<td>0.72 ± 0.01</td>
<td>0.93 ± 0.01</td>
<td>0.81 ± 0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Alignment</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>S4-D</td>
<td>0.67 ± 0.02</td>
<td>0.81 ± 0.02</td>
<td>0.28 ± 0.05</td>
<td>0.43 ± 0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.70 ± 0.03</td>
<td>0.83 ± 0.02</td>
<td>0.74 ± 0.03</td>
<td>0.78 ± 0.02</td>
</tr>
</tbody>
</table>

Table 1: Classification scores for the British vs. American English task. The baseline cosine (COS) method outputs unstable words whose cosine distance is greater than 0.3/0.5/0.7. Top-N/bot-N alignments use the most/least frequent words. S4-D (our method) is the self-supervised semantic shift detection. The scores for S4-D are given as the mean and standard deviation over 10 evaluation rounds. The initial alignment is irrelevant to noisy-pairs as it necessarily searches an alignment.

Table 2: Classification accuracy of the unsupervised lexical semantic change detection on the SemEval-2020 Task 1 data set. The results were obtained by aligning the embedding matrices using different alignment strategies, and applying a threshold to the cosine distance of the aligned vectors, selected by cross-validation. Top-fr. and Bot-fr. are alignments using OP on the top and bottom 5% and 10% frequent words. The bottom rows shows the top 3 high scoring submissions to SemEval-2020 Task 1 in the post evaluation phase.
results reveal that Global and Noise-Aware produce very similar rankings, with rho approaching 1 even for small values of k. On the other hand, the ranking correlation between S4-A and the others is in the most shifted words, with the ranking of the remaining words being similar to Global and Noise-Aware. In summary, S4-A can be used to find novel shifted words that are overlooked by existing methods such as Global and Noise-Aware. Table 3 shows the list of uniquely discovered words among the top most shifted for Global and S4-A. Noise-Aware because it does not show any novel words when compared to Global, i.e., its predictions are the same between arXiv subjects of Artificial Intelligence and Classical Physics. We find that words uniquely discovered by S4-A can be naturally explained in context of their subjects, for instance, mass is likely more often used as probability mass in AI, and as physical mass in classical physics. More comparisons are shown in Appendix D.

Figure 2: Ranking correlation between Global, Noise-Aware, and S4-A alignments at varying top-k levels. Works are ranked from most to least shifted according to each method and the ranking correlation is measured with Spearman’s rho. The semantic shifts are between arXiv subjects cs.AI and physics.class-ph.

<table>
<thead>
<tr>
<th>Global/Noise-Aware</th>
<th>S4-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>component</td>
</tr>
<tr>
<td>approximation</td>
<td>element</td>
</tr>
<tr>
<td>boundary</td>
<td>mass</td>
</tr>
<tr>
<td>conceptual</td>
<td>order</td>
</tr>
<tr>
<td>knowledge</td>
<td>solution</td>
</tr>
<tr>
<td>plane</td>
<td>space</td>
</tr>
<tr>
<td>reference</td>
<td>term</td>
</tr>
<tr>
<td>rules</td>
<td>time</td>
</tr>
<tr>
<td>system</td>
<td>vector</td>
</tr>
</tbody>
</table>

Table 3: Unique words discovered by each alignment method among the top 50 most shifted, between arXiv cs.AI and physics-class-ph corpora. Global and Noise-Aware show the same predictions in the top 50 words.

5 Conclusions

We introduced S4-D and S4-A as self-supervised approaches to detect word-level semantic shifts on monolingual corpora. Motivated by the unsupervised nature of this problem, we introduce self-supervision based on the perturbation of word vectors and apply it to binary classification and vector alignment. S4-D is presented as an alternative to baseline unsupervised methods for semantic shift detection, particularly in the case of binary classification. We show, through experiments in Section 4.1, that it achieves over 2× higher F1-scores than baselines in the classification settings. Moreover, we show how the alignment of word embeddings affect the outcome of such methods. Particularly, we show that global alignment uses the assumption of smooth transition, which may not hold true in the scenario of cross-domain semantic shift, where many words can be highly shifted. For that reason, we present an extension of our method, named S4-A, that uses its predictions to refine the alignment of the input embeddings. We demonstrate its usefulness quantitatively, through the detection task in Section 4.2, where S4-A allows for the detection of unique words when using a simple cosine distance baseline. Qualitatively, we demonstrate that S4-A is able to discovery novel shifts when compared to other alignment methods.

There are still open questions on how the self-supervised model is affected according to part-of-speech, frequency range, and degree of polysemy of words. In addition, factors such as number of tokens, vocabulary size, and degree of change of the language may impact the quality of the embeddings, therefore, affect the semantic shift detection.

While this remains a difficult task, we believe that this work will help numerous applications of semantic shift detection and alignment that have been recently explored, especially in the monolingual and cross-domain setting.

Ethical Impact

Prior work has shown that embeddings in a single corpus can encode many cultural stereotypes such as gender and racial bias. This is not particularly surprising as language is a tool for creating common meaning. Stereotypes, gender and racial bias are social constructs that have been quite prominent in many cultural artifacts including languages. Code words and dog-whistles are words often used to mean different things only for a specific community. A word that looks quite common and benign can be very offensive when evaluated in context. Some of these differences can be rooted in social justice work of reclaiming controversial words. In short, language codes rich and complex cultural and historical differences that are lost when treated as a monolith. Our emphasis is in creating tools not to smooth over these differences, but help identify them. To accomplish this, we use two corpora in the same language to understand which words in one corpus have a different meaning compared to another corpus. Our work leverages potential bias in languages by using one corpus as a point of reference for the other to highlight the differences.
Acknowledgements
This work was supported by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons). We also thank the anonymous reviewers who contributed with our work with their invaluable feedback.

References


Cook, P.; and Stevenson, S. 2010. Automatically Identifying Changes in the Semantic Orientation of Words. In LREC.


