Effects of Pre- and Post-Processing on type-based Embeddings in Lexical Semantic Change Detection

Jens Kaiser[*], **Sinan Kurtyigit**[*], **Serge Kotchourko**[*], **Dominik Schlechtweg**
Institute for Natural Language Processing, University of Stuttgart
{jens.kaiser,sinan.kurtyigit,serge.kotchourko,schlecdk}@ims.uni-stuttgart.de

Abstract

Lexical semantic change detection is a new and innovative research field. The optimal fine-tuning of models including pre- and post-processing is largely unclear. We optimize existing models by (i) pre-training on large corpora and refining on diachronic target corpora tackling the notorious small data problem, and (ii) applying post-processing transformations that have been shown to improve performance on synchronic tasks. Our results provide a guide for the application and optimization of lexical semantic change detection models across various learning scenarios.

1 Introduction

In recent years Lexical Semantic Change Detection (LSCD), i.e. the detection of word meaning change over time, has seen considerable developments (Tahmasebi et al., 2018; Kutuzov et al., 2018; Hengchen et al., 2021). The recent publication of multi-lingual human-annotated evaluation data from SemEval-2020 Task 1 (Schlechtweg et al., 2020) makes it now possible to compare LSCD models in a variety of scenarios. The task shows a clear dominance of type-based embeddings, although these are strongly influenced by the size of training corpora. In order to mitigate this problem we propose pre-training models on large corpora and refine them on diachronic target corpora. We further improve the obtained embeddings with several post-processing transformations which have been shown to have positive effects on performance in semantic similarity and analogy tasks (Mu et al., 2017; Artetxe et al., 2018b; Raunak et al., 2019) as well as term extraction (Hätty et al., 2020). Extensive experiments are performed on the German and English LSCD datasets from SemEval-2020 Task 1. According to our findings, pre-training is advisable when the target corpora are small and should be done using diachronic data. We further show that pre-training on large corpora strongly interacts with vector dimensionality and propose a simple solution to avoid drastic performance drops. Post-processing often yields further improvements. However, it is hard to find a reliable parameter that performs well across the board. Our experiments suggest that it is possible to use simple pre- and post-processing techniques to improve the state-of-the-art in LSCD.

2 Related Work

As evident in Schlechtweg et al. (2020) the field of LSCD is currently dominated by Vector Space Models (VSMs), which can be divided into type-based (static) (Turney and Pantel, 2010) and token-based (contextualized) (Schütze, 1998) models. Prominent type-based models include low-dimensional embeddings such as Global Vectors (GloVe, Pennington et al., 2014) and Skip-Gram with Negative Sampling (SGNS, Mikolov et al., 2013a,b). However, as these models come with the deficiency that they aggregate all senses of a word into a single representation, token-based embeddings have been proposed (Peters et al., 2018; Devlin et al., 2019). According to Hu et al. (2019) these models can ideally capture complex characteristics of word use, and how they vary across linguistic contexts. The results of SemEval-2020 Task 1 (Schlechtweg et al., 2020), however, show that contrary to this, the token-based embedding models (Beck, 2020; Kutuzov and Giulianelli, 2020) are heavily outperformed by the type-based ones (Pražák et al., 2020; Asgari et al., 2020). The SGNS model was not only widely used, but also performed best among the participants in the task. This result was recently reproduced in the DIACR-

[*]Authors contributed equally, and their ordering was determined randomly.

Ita shared task (Basile et al., 2020; Laicher et al., 2020; Kaiser et al., 2020b). Its fast implementation and combination possibilities with different alignment types further solidify SGNS as the standard in LSCD (Schlechtweg et al., 2020, 2019a; Shoemark et al., 2019; Kutuzov et al., 2020). Hence, the embeddings used in this work are SGNS-based.

Further increases in performance of type-based VSMs can be achieved by various post-processing transformations. This has been shown for semantic similarity and analogy tasks (Mu et al., 2017; Artetxe et al., 2018b; Raunak et al., 2019) as well as term extraction (Hätty et al., 2020). It is still an open question whether these transformations improve performance in the special setting of LSCD where we typically have several corpora and vector spaces which have to be transformed simultaneously (Schlechtweg et al., 2020). An indication is given by Schlechtweg et al. (2019a) showing that for a simple LSCD model mean centering leads to consistent performance improvements on two German data sets. Whether this result is reproducible on further data sets, more complex models and further post-processing techniques has not been determined yet.

Post-processing methods operate on information already contained in a VSM, rather than adding additional information. Further semantic information can be introduced by pre-training vectors on a larger unspecific collection of text (Kutuzov and Kuzmenko, 2016) or by training a seperate matrix on such text and concatenating the two VSMs (Limsopatham and Collier, 2016). This is especially helpful for cases where only smaller specialized corpora are given. Combining the information from two models is also found in Kim et al. (2014), here it is used for alignment proposes. We operate similarly to Kim et al. but with the motivation of Limsopatham and Collier and Kutuzov and Kuzmenko, as we aim to enrich a VSM prior to the training process.

3 Data and Tasks

We train SGNS-based VSMs on various corpora and use a word similarity task and an LSCD task for evaluation. The two tasks share a common aspect: the vector representations of two words need to be compared with some metric (e.g. cosine similarity), and word pairs need to be ranked according to that metric. In the word similarity task, we have the vectors of two different words in the same vector space (w_i, w_j), while for LSCD we have the vectors of the same word but from different vector spaces representing different time periods (w_i^{t1}, w_i^{t2}).

Modern Data. We use two large modern English and German corpora, PukWaC (Baroni et al., 2009) and SdeWaC (Faaß and Eckart, 2013) to validate the post-processing methods on the word similarity task and to create pre-trained embeddings for the LSCD task. PukWaC and SdeWaC are web-crawled corpora from the *.uk* and *.de* domain respectively. Resulting in fairly large corpora, 2B tokens and 750M tokens (see Table 1). We evaluate vector representations created on the two corpora on a standard dataset of human similarity judgments, WordSim353 (Finkelstein et al., 2002), by measuring Spearman's rank correlation coefficient of the cosine similarity of vectors for target word pairs with human judgments.

Diachronic Data. We utilize the English and German datasets provided by SemEval-2020 Task 1 Subtask 2 (Schlechtweg et al., 2020). Each dataset contains two *target corpora* from different time periods, t_1 and t_2, as well as a list of target words. The corpora originate mostly from newspaper articles and books. Their biggest difference to PukWaC and SdeWaC is their approximately 10 to 100 times smaller size, according to token counts (see to Table 1). The task is to rank the list of target words according to their word sense divergence, gradually from 0 (no change) to 1 (total change). The rank predictions are compared against gold data which is based on human judgments. Once again Spearman's rank correlation coefficient is used to measure performance on the task.

4 Models

Following the popular approach taken for type-based vector space models in LSCD, we combine three sub-systems: (i) creating semantic word representations, (ii) aligning them across corpora, and (iii) measuring differences between the aligned representations (Schlechtweg et al., 2019a; Dubossarsky et al., 2019; Shoemark et al., 2019). Alignment is needed as columns from different vector spaces may not correspond to the same coordinate axes, due to the stochastic nature of many low-dimensional word representations (Hamilton et al., 2016). Additionally, we aim to refine subsystem (i) by adding pre-trained semantic word representations and using post-processing methods

	DIACHRON				MODERN	
	GER_{t1}	GER_{t2}	ENG_{t1}	ENG_{t2}	SDEWAC	PUKWAC
source	DTA	BZ+ND	CCOHA	CCOHA	web	web
time period	1800 – 1899	1946 – 1990	1810 – 1860	1960 – 2010	∼2005 – 2005	∼2005 – 2005
# of tokens	66.9M	67.2M	6.48M	6.62M	750M	1.92B
# of types	51.1K	59.1K	25.9K	37.5K	44.6K	51.9K
min word freq.	39	39	4	4	450	750

Table 1: Corpus statistics. GER_{t1} and GER_{t2} are sampled from DTA (Deutsches Textarchiv, 2017), BZ (Berliner Zeitung, 2018) and ND (Neues Deutschland, 2018). DTA contains texts from different genres, BZ and ND are collections of newspaper articles. Clean Corpus of Historical American English (CCOHA) (Davies, 2012; Alatrash et al., 2020) is a genre balanced collection of texts from a wide variety of time periods and the basis for ENG_{t1} and ENG_{t2}.

to improve the quality of the created semantic word representations.[1]

We use SGNS (Mikolov et al., 2013a,b) to create type-based word representations in combination with three different alignment methods, Orthogonal Procrustes (OP), Vector initialization (VI), and Word Injection (WI). The three alignment methods combined with SGNS have been proven to be state-of-the-art, even when competing against token-based embeddings (Schlechtweg et al., 2020; Kaiser et al., 2020a; Basile et al., 2020). Cosine Distance (CD) is used to measure differences between word vectors.[2]

4.1 Alignment

Vector initialization (VI). In VI we first train SGNS on one corpus and then use the learned word and context vectors to initialize the model for training on the second corpus (Kim et al., 2014; Kaiser et al., 2020a). The motivation is that the vector of a word with similar contexts across both corpora will not deviate much from its initialized value. On the other hand, vectors of words with different contexts across both corpora, will be updated to accommodate the new semantic properties. Words which only appear in the second corpus are initialized on random vectors.

Orthogonal Procrustes (OP). SGNS is trained on each corpus separately, resulting in word matrices A and B. To align them, we follow Hamilton et al. (2016) and calculate an orthogonally-constrained matrix W^*:

$$W^* = \arg\min_{W \in O(d)} \|BW - A\|_F .$$

Prior to this alignment step both matrices are length-normalized and mean-centered (Artetxe et al., 2017; Schlechtweg et al., 2019a).

Word Injection (WI). The sentences of both corpora are shuffled into one joint corpus, but all occurrences of target words are substituted by the target word concatenated with a tag indicating the corpus it originated from (Ferrari et al., 2017; Schlechtweg et al., 2019a; Dubossarsky et al., 2019). This leads to the creation of two vectors for each target word in one vector space, while non-target words receive only one vector encoding information from both corpora.

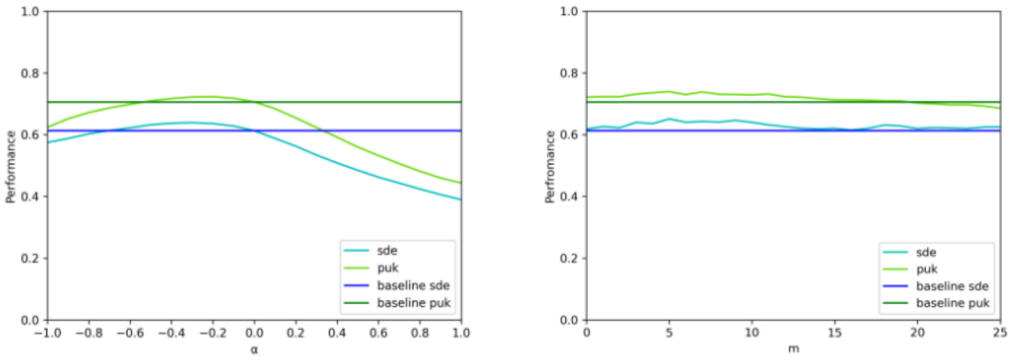
No Alignment (NO). Comparing two vector spaces without aligning them results in poor performance on LSCD (Schlechtweg et al., 2019a). As VI shows, initializing the model with weights from the previous run, results in aligned vector spaces. We expand on this concept by initializing two models on the same pre-trained weights assuming that the resulting vector spaces are aligned to one another. The difference to VI is that instead of initializing model B with the weights from model A, the weights from a third pre-trained model C are used to initialize both models A and B.

4.2 Pre-training

The corpora used in the context of LSCD are often small, as they are restricted by the length of time periods or availability of historical data. For example the English corpora of SemEval-2020 Task 1 only have 6.6M tokens each, compared to 1.9G of PUKWAC. This reduced corpus size limits the amount of semantic information encoded into VSMs trained on the corpus. Pre-training addresses this problem by first training SGNS on a large, possibly external corpus, and then using these vectors

[1] Find a comprehensive overview of type-based LSCD models including semantic representations, alignments and measures in Schlechtweg et al. (2019a).

[2] We provide our code at: https://github.com/Garrafao/LSCDetection.



(a) (b)

Figure 1: Performance on modern data (wordsim353) **left**: SOT for $\alpha \in [-1, 1]$, **right:** MC+PCR across different amounts of PCs removed. Zero PCs removed indicates only mean centering. Baselines are performances without PP.

to initialize the model for training on the smaller diachronic target corpora. The idea is that the model first learns very broad and general semantic properties followed by the training on the target corpora, where corpus and time specific details are picked up, i.e., a form of refinement. This procedure is applicable to all alignment types.

We use PUKWAC and SDEWAC for pre-training, later referenced as MODERN. However, pre-training on modern corpora is only advisable if the assumption can be made that the meanings of words in the pre-training corpus roughly correspond to the meanings of words in the target corpora. It is unclear to which extent this assumption holds for our data. Hence, we also combine the two target corpora into a bigger corpus, referenced as DIACHRON, which is then used for pre-training.

4.3 Post-processing (PP)

Similarity Order Transformation (SOT). In 2nd order similarity, the similarity of two words is assessed in terms of how similar they are to a third word (Schütze and Pedersen, 1993; Artetxe et al., 2018b; Schlechtweg et al., 2019b). This can analogously be done for higher (3rd, 4th, etc.) orders. According to Artetxe et al. (2018b) these orders capture different aspects of language. Artetxe et al. propose a linear transformation deriving higher or lower orders of similarity from a given matrix X. For this, the product with the transpose matrix is split into its eigendecomposition $X^T X = Q\lambda Q^T$, so that λ is a positive diagonal matrix whose entries are the eigenvalues of $X^T X$ and Q is an orthogonal matrix with their respective eigenvectors as columns. The linear transformation matrix is then defined as $W_\alpha = Q\lambda^\alpha$, where α is the parameter that adjusts the desired similarity order. Applying this to the original embeddings X yields the transformed embeddings $X' = XW^\alpha$.

Mean Centering (MC). The centroid of a matrix is the average vector over all vectors in a matrix: $\vec{c} = \frac{1}{|V|}\sum_i^V \vec{w_i}$. MC refers to subtracting $\vec{c}$ from each $\vec{w_i}$ in the matrix. MC alters all dimensions so that the mean of all columns is zero. Artetxe et al. provide the intuitive motivation for MC that it moves randomly similar vectors further apart and Mu and Viswanath (2018) consider mean centering as an operation making vectors "more isotropic", i.e., more uniformly distributed across the vector space. Mu and Viswanath indicate that isotropy of word vectors is positively correlated to performance.

Principal Component Removal (PCR). Given a n-dimensional matrix X, Principal Component Analysis (PCA, Pearson, 1901) returns n vectors where each vector describes a best fitting line for the data while being orthogonal to the first $n-1$ vectors. Thus, the first PC describes the greatest variance in the first direction, the second PC describes the second greatest variance in the second direction, and the nth PC describes the nth greatest variance in the nth direction. Mu and Viswanath (2018) use PCA to compute the top m PCs from a mean centered word embedding $\bar{M}$: $p_1, ..., p_m = \text{PCA}(\bar{M})$. Subsequently these PCs

are used to project each vector $v \in M$ onto the subspace spanned by the PCs. This projection is then subtracted from the original mean centered word vector $\tilde{v}$ by $v' = \tilde{v} - \sum_{i=1}^{m}(p_i^\intercal v)p_i$, which results in nullifying the top m PCs in M. This is similar to the approach of Bullinaria and Levy (2012). Mu and Viswanath combine both MC and PCR into one PP transformation (MC+PCR).

As for MC Mu and Viswanath's main motivation for PCR is to make vectors more isotropic. They also demonstrate empirically that the top PCs encode word frequency and offer the removal of this noise from the matrix as an alternative explanation for observed performance improvements.

Stacking. VI and OP alignment result in two matrices, and hence, a proper way for applying PP to both of them is needed. The naïve way of simply post-processing both matrices separately (SEP) may violate the assumption that they are represented in the same space. Therefore, in a second approach, we apply PP to both matrices simultaneously by stacking them vertically beforehand (STA). Preliminary experiments showed that following the naïve way of PP (SEP) led to severe decrease in performance for SOT (but not for MC+PCR). Hence, applying SOT on two matrices separately is followed by an orthogonal post-alignment (SEP+PA).

5 Experiments

For the most part, we chose common model hyper-parameter settings in order to keep our results comparable to previous research (Hamilton et al., 2016; Schlechtweg et al., 2019a; Kaiser et al., 2020a). We fine-tune for different alignment methods and datasets by varying dimensionality d, window size w and number of training epochs e.[3]

5.1 Validation

We validate the results reported by Artetxe et al. (2018a) and Mu and Viswanath (2018) on PUKWAC and SDEWAC. The performance peaks for negative α-values around -0.2 as well as the slight performance increase over the baseline for SOT are in line with the findings of Artetxe et al. (see Figure 1a). For MC+PCR we observe the greatest performance improvement when the number of removed PCs is around $m = \frac{d}{100}$ (see Figure 1b). This fits the rule of thumb as stated by Mu and Viswanath.

5.2 LSCD

5.2.1 Pre-training

We tune SGNS models for each alignment method with and without pre-training (baseline), see Table 2. Recall from Section 4.2 that we use the corpora MODERN and DIACHRON for pre-training. Table 2 lists the maximum and mean performances of the baseline and pre-training with different alignment methods, as well as the standard deviation (for a visual representation of the max values see Figure 2). The mean is calculated across different d, e and w, giving the expected performance in a realistic scenario where fine-tuning hyper-parameters is not possible (Schlechtweg et al., 2020; Basile et al., 2020). For German, the baseline max and mean scores could not be significantly improved by pre-training across alignments. For English, pre-training on DIACHRON results in better max and mean scores for OP and WI, with max improvements up to .10. Also, the overall best result is achieved with OP and pre-training on DIACHRON. The usage of MODERN does not improve on the maximum, while reducing the mean. The overall lower performance as well as the observed performance improvements compared to German, may be attributed to the roughly 10 times smaller target corpora. That is, pre-training is helpful on the smaller target corpora.

5.2.2 Post-processing

For every combination of alignment and pre-training method, the matrix with the highest performance across parameters is chosen as the baseline. SOT and MC+PCR are applied individually to these matrices within a wide parameter range (see Appendix B) for both stacking methods (STA and SEP/SEP+PA). Table 3 presents the mean optimal performance gains after PP, which is calculated by extracting the best performance after PP for every matrix, subtracting the baseline values and averaging the values per language. Averaging the respective parameter values yields the mean argmax. Figure 3a and 3d show the highest performances for every baseline matrix after SOT and MC+PCR respectively.

SOT. As we see in Figure 3a, SEP+PA and STA perform similarly. We find small mean performance gains across the board (.013 for GER+STA,

[3] For a detailed overview on SGNS parameters see Appendix B.

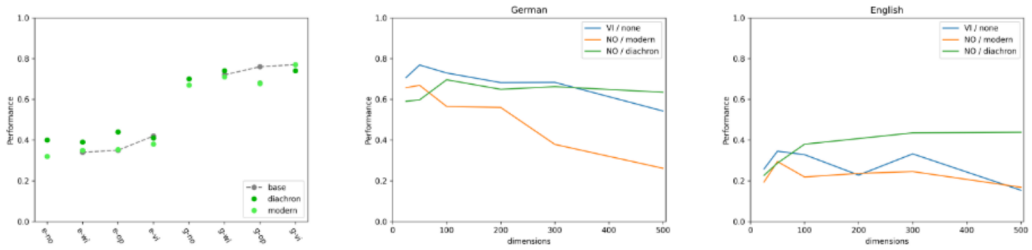


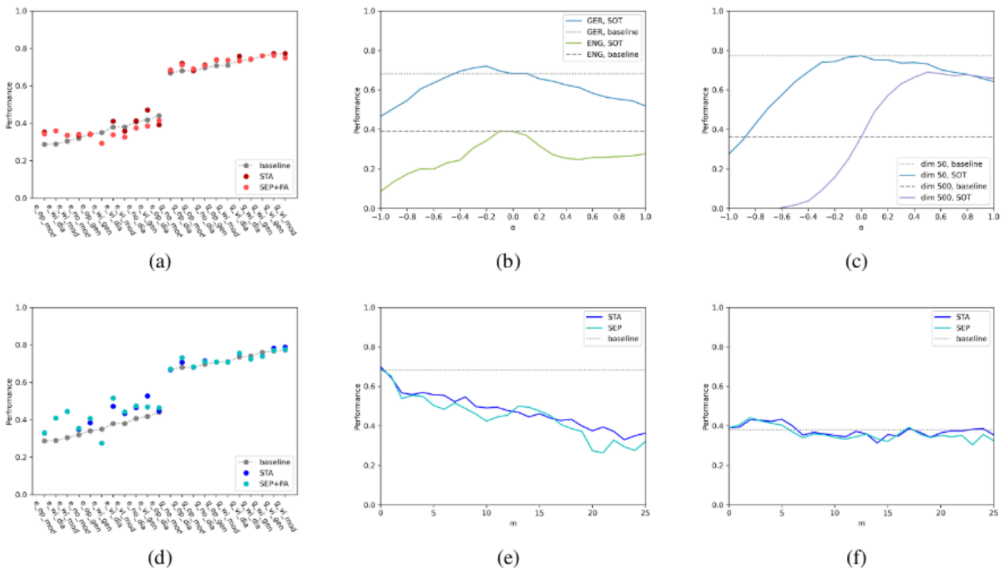
Figure 2: **Left**: max scores from Table 2, **middle** and **right**: Performance (Spearman's rho) of NO alignment method on LSCD task across different dimensionalities and pre-training corpora. VI without pre-training as comparable baseline.

	align.	baseline		DIACHRON		MODERN	
		max	*mean/std*	*max*	*mean/std*	*max*	*mean/std*
GER	VI	**.77**	.72 / .063*	.74	.61 / .067*	**.77**	.70 / .060*
	OP	**.72**	.69 / .022	.68	.59 / .049*	.68	.61 / .051*
	WI	**.76**	.70 / .033	.74	.69 / .037*	.71	.66 / .043*
	NO	-	- / -	.70	.58 / .081*	.67	.60 / .050*
ENG	VI	**.42**	.30 / .067	.41	.28 / .073	.38	.26 / .060
	OP	.34	.28 / .041	**.44**	.31 / .071	.35	.27 / .047
	WI	.35	.28 / .041	**.39**	.29 / .053	.35	.24 / .055
	NO	-	- / -	.40	.34 / .080	.32	.24 / .060

Table 2: max and mean performance on LCSD task (Spearman's rho) for all alignment methods. Note: mean values marked with (*) ignore results utilizing d <100 due to consistent performance drops at higher d.

.008 for GER+SEP+PA, .013 for ENG+STA), except for ENG+SEP+PA where a minuscule decrease (-.005) can be seen. Overall, STA outperforms SEP+PA slightly. We now further examine the effect of SOT+STA on individual matrices. In general, the data can approximately be described as a downward opening parabola (see Figure 3b), with different peaks for both languages and slight differences between alignment methods. Averaging the argmax for α shows us where these peaks are. The calculations yield a mean optimal α of 0 for GER+STA, and -0.2 for ENG+STA. For GER the peak performance always lies in the interval $[-0.2, 0.3]$. This changes to $[-0.4, 0.1]$ for ENG, except for one outlier, where the peak is at -0.8. Moving α away from this parameter range results in severe performance decreases. This behaviour can also be seen on the MODERN corpora (see Figure 1) and is in line with the findings of Artetxe et al. (2018b). In order to predict a high-performing parameter, independent from the underlying matrix, we calculate mean performance gains for fixed parameter values. The values are chosen according to the the above-described peak intervals for the respective languages. However, on average, using a fixed parameter results in slight performance losses, notwithstanding the α-value, and hence, finding a high-performing fixed parameter value was not possible. We observe similar findings for individual alignment methods and varying dimensionality. However, GER+VI alignment represents an interesting exception: With high dimensionality (> 300) base performance drops heavily (Kaiser et al., 2020a), and is then "repaired" by the PP, bringing it close to the baseline of the best performing dimension (see Figure 3c).

MC+PCR. As we see in Figure 3d, MC+PCR yields small improvements over the baselines for German. This is also reflected in the mean gain in Table 3. We find that no single value for m yields consistent improvements. However, we find that for m=0 (only MC) MC+PCR consistently improves the baseline slightly (see Figure 3e), while for higher m the performance decreases consistently. For English we see greater improvements, see Fig-



(a) (b) (c)

(d) (e) (f)

Figure 3: **Top:** SOT (3a, 3c, 3b), **Bottom:** MC+PCR (3d, 3e, 3f). Performance over high-scores (3a, 3d). Representative results after SOT+STA over german and english dataset (3b). Representative plot of "repair" effect after SOT+STA for GER+VI (3c). Representative result after MC+PCR over German and English dataset (3e, 3f). Note for 3a and 3d: where data points overlap only lighter colour visible; dashed line between baseline data points only a visual aid.

	PP + STA/SEP	argmax *mean/std*	gain *mean/std*
GER	SOT + STA	0.0/0.2	.013/.013
	SOT + SEP+PA	0.1/0.3	.008/.015
	MC+PCR + STA	1.2/1.6	.004/.042
	MC+PCR + SEP	0.7/1.1	.004/.043
ENG	SOT + STA	-0.2/0.2	.013/.041
	SOT + SEP+PA	-0.2/0.3	-.005/.043
	MC+PCR + STA	3.0/3.8	.049/.068
	MC+PCR + SEP	6.2/7.1	.058/.077

Table 3: Mean of best-performing parameters and mean performance gain compared to baseline on LSCD task. Parameter range for SOT [-1,1] and MC+PCR [0,25]

ure 3f, 3d and mean gain in Table 3. A range of parameters shows improvements with m=3 yielding the highest (.0175). This can also be seen in Figure 3f where several parameters yield improvements. We conclude that predicting a parameter for likely performance improvement is possible for English, but not for German. However, if this PP should be used, we recommend using a parameter space of $m \in [0, 5]$, as this parameter space is most likely to produce improvements on English, while not harming performance too much on German. This also roughly corresponds to the recommendation of Mu and Viswanath (2018), as they predict that the parameter should be chosen around $\frac{d}{100}$. Furthermore, we suggest using STA, as this does on average show better performance over SEP for the aforementioned parameter space. We see that the effects of SOT as well as MC+PCR are highly dependent on the underlying matrix.

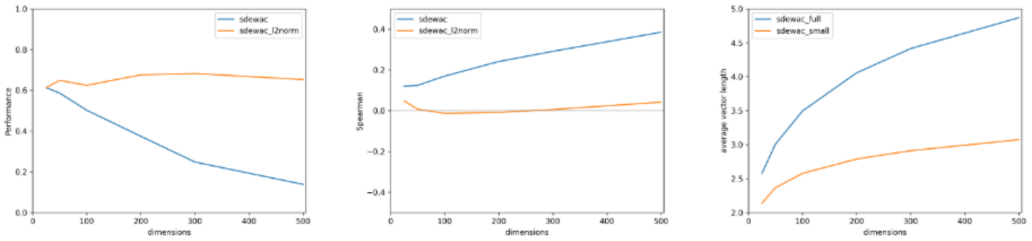
6 Analysis

Test Statistics. The effects of pre-training and PP methods on word embeddings are not limited to performance differences in word similarity or LSCD tasks. We use two test statistics to further analyse vector spaces: (i) isotropy (Mu and Viswanath, 2018), i.e., uniformity of vector distribution and (ii) frequency bias (Dubossarsky et al., 2017; Kaiser et al., 2020a), i.e., correlation between cosine distance and frequency.[4]

6.1 Pre-training

On the German dataset it is noticeable that pre-training on DIACHRON often results in slight drop in performance at higher d. This behaviour is more pronounced, consistent and even visible on the En-

[4]We compute correlation based on frequency in the second target corpus, results were similar for the first target corpus.



(a) (b) (c)

Figure 4: Test-statistics for result analysis, **left**: Performance after pre-training on different corpora, **middle**: Correlation between CD and frequency, **right**: average vector length of the weights created on different-sized pre-training corpora.

glish dataset when pre-training on MODERN, see Figure 2.[5] Such a drop in performance after initializing on pre-trained vectors has already been observed by Kaiser et al. (2020a). The authors relate the drop to an increased frequency bias and reduce it by increasing e/w. It is noteworthy that the drop is much more pronounced for pre-training on MODERN compared to DIACHRON. This can be attributed to a difference in word vector lengths of the SGNS model used for initialization. We make the following observation: average word vector length increases with the amount of training word pairs. The difference more training data makes is amplified at higher d, see Figure 4c. By length-normalizing the word vectors between the initialization and training step, the drop in performance can be completely circumvented. Additionally, the frequency bias is reduced to 0, see Figure 4b.

For English, we expected a higher performance gain from pre-training when using MODERN because of the small data size. However, we observe no improvements over the baseline. Using length-normalized word vectors for initialization does result in slightly improved max and mean values for MODERN but these are still lower than max and mean values of DIACHRON.

6.2 SOT

SOT has a clear effect on isotropy, which has not been described in previous research. Isotropy shows the same behaviour across both languages and all models, and is best described as a vertically mirrored S-curve (see Figure 5a). Decreasing α increases isotropy close to 1, while increasing α decreases isotropy close to 0. The average correlation (Pearson) between α and isotropy over all matrices is -.89 for both languages. However, the performance correlates only slightly with isotropy (-.25, .35). Moreover, α correlates only weakly with frequency bias (.19, -.12, however with high variance). In order to explain the above-described "repair" effect we take a closer look at the three GER+VI models. Applying SOT brings large performance increases, as stated in Section 5.2.2. For all three models a considerably higher baseline frequency bias for d=500 is visible. SOT strongly reduces this bias for MODERN, and results in a huge performance gain (see Figure 5b).

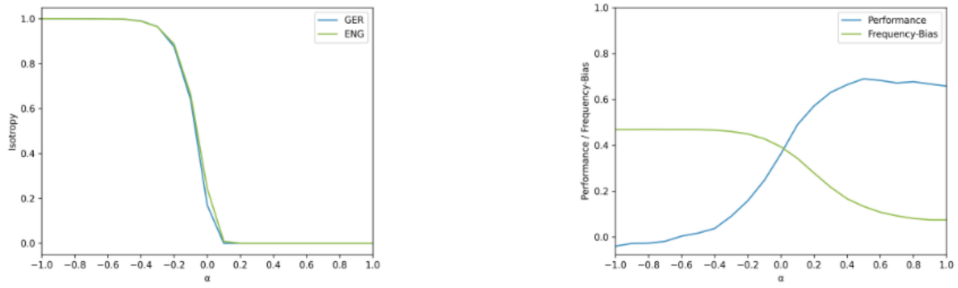
6.3 MC+PCR

As Mu and Viswanath (2018)'s main motivation behind MC+PCR is to increase isotropy of a vector space as well as removal of word frequency noise through PCR, we examine how isotropy and frequency bias develop with m. While PCR has the predicted effect on frequency bias (GER: -.94, ENG: -0.6), PCR does in fact not increase isotropy, contrary to Mu and Viswanath's motivation of "rounding towards isotropy", but has a consistent reducing effect (GER -.75, ENG: -.7). Thus, we believe that rounding towards isotropy is not suitable for explaining performance. Furthermore, we observe that MC not only exhibits effects on isotropy, but also acts on frequency bias, thus Mu and Viswanath's PCR motivation can be extended to MC.

7 Conclusion

We tested the effects of pre-training and post-processing on a variety of LSCD models. We performed extensive experiments on a German and an

[5] Although not depicted, the other alignment techniques in combination with pre-training show very similar behaviour to NO.



(a) (b)

Figure 5: Representative plot for the isotropy after SOT+STA (5a). Performance and frequency bias after SOT+STA for GER+VI+BIG (5b).

English LSCD dataset. According to our findings, pre-training is advisable when the target corpora are small and should be done using diachronic data. The size of the pre-training corpus is crucial, as a large number of training pairs leads to performance drops, which are probably caused by their effect on vector length. Length-normalization may be used on pre-trained vectors to counteract this effect.

Further performance improvements may be reached by post-processing. While SOT+STA yielded moderate improvements for both languages, MC+PCR showed larger improvements, but only on English. However, for neither we were able to find a reliable parameter that performed well across the board. Instead, we found that a well-performing parameter value is highly dependent on the underlying matrix. Both post-processing methods affect isotropy and frequency bias.

The methods we tested are particularly helpful when tuning data is available, as performance can be optimized and becomes more predictable. Hence, we recommend to obtain a small annotated sample of target words for the target corpora and to tune pre-training, model and post-processing parameters on the sample before performing predictions for semantic changes on unseen data. With the recent upsurge of digitized historical corpora and diachronic semantic annotation efforts (Tahmasebi and Risse, 2017; Schlechtweg et al., 2018, 2020; Basile et al., 2020; Rodina and Kutuzov, 2020) this may often be a likely and feasible scenario.

Acknowledgments

Dominik Schlechtweg was supported by the Konrad Adenauer Foundation and the CRETA center funded by the German Ministry for Education and Research (BMBF) during the conduct of this study. We thank the reviewers for their insightful feedback.

References

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2020. CCOHA: Clean Corpus of Historical American English. In *Proceedings of the 12th International Conference on Language Resources and Evaluation*, Marseille, France. European Language Resources Association.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 451–462. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, Iñigo Lopez-Gazpio, and Eneko Agirre. 2018b. Uncovering divergent linguistic information in word embeddings with lessons for intrinsic and extrinsic evaluation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 282–291, Brussels, Belgium. Association for Computational Linguistics.

Ehsaneddin Asgari, Christoph Ringlstetter, and Hinrich Schütze. 2020. EmbLexChange at SemEval-2020 Task 1: Unsupervised Embedding-based Detection of Lexical Semantic Changes. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43:209–226.

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. Overview of the EVALITA 2020 Diachronic Lexical Semantics (DIACR-Ita) Task. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

Christin Beck. 2020. DiaSense at SemEval-2020 Task 1: Modeling sense change via pre-trained BERT embeddings. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Berliner Zeitung. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin [online]. 2018.

John Bullinaria and Joseph Levy. 2012. Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and svd. *Behavior research methods*, 44:890–907.

Mark Davies. 2012. Expanding Horizons in Historical Linguistics with the 400-Million Word Corpus of Historical American English. *Corpora*, 7(2):121–157.

Deutsches Textarchiv. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Herausgegeben von der Berlin-Brandenburgischen Akademie der Wissenschaften [online]. 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-Out: Temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1147–1156, Copenhagen, Denmark.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – A corpus of parsable sentences from the web. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 61–68. Springer Berlin Heidelberg.

Alessio Ferrari, Beatrice Donati, and Stefania Gnesi. 2017. Detecting domain-specific ambiguities: An NLP approach based on wikipedia crawling and word embeddings. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference Workshops*, pages 393–399.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20:116–131.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Anna Hätty, Dominik Schlechtweg, Michael Dorna, and Sabine Schulte im Walde. 2020. Predicting degrees of technicality in automatic terminology extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington. Association for Computational Linguistics.

Simon Hengchen, Nina Tahmasebi, Dominik Schlechtweg, and Haim Dubossarsky. 2021. Challenges for computational lexical semantic change. In Nina Tahmasebi, Lars Borin, Adam Jatowt, Yang Xu, and Simon Hengchen, editors, *Computational Approaches to Semantic Change*, volume Language Variation, chapter 11. Language Science Press, Berlin.

Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3899–3908, Florence, Italy. Association for Computational Linguistics.

Jens Kaiser, Dominik Schlechtweg, Sean Papay, and Sabine Schulte im Walde. 2020a. IMS at SemEval-2020 Task 1: How low can you go? Dimensionality in Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Jens Kaiser, Dominik Schlechtweg, and Sabine" Schulte im Walde. 2020b. OP-IMS @ DIACR-Ita: Back to the Roots: SGNS+OP+CD still rocks Semantic Change Detection. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

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

Andrey Kutuzov, Vadim Fomin, Vladislav Mikhailov, and Julia Rodina. 2020. Shiftry: Web service for diachronic analysis of russian news. In *Proceedings of the International Conference "Dialog"*.

Andrey Kutuzov and Mario Giulianelli. 2020. UiO-UvA at SemEval-2020 Task 1: Contextualised Embeddings for Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Andrey Kutuzov and Elizaveta Kuzmenko. 2016. Cross-lingual trends detection for named entities in news texts with dynamic neural embedding models. In *NewsIR@ECIR*, pages 27–32.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Severin Laicher, Gioia Baldissin, Enrique Castaneda, Dominik Schlechtweg, and Sabine Schulte im Walde. 2020. CL-IMS @ DIACR-Ita: Volente o Nolente: BERT does not outperform SGNS on Semantic Change Detection. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

Nut Limsopatham and Nigel Collier. 2016. Modelling the combination of generic and target domain embeddings in a convolutional neural network for sentence classification. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 136–140, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, Nevada, USA.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. Representing sentences as low-rank subspaces. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 629–634, Vancouver, Canada. Association for Computational Linguistics.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

Neues Deutschland. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin [online]. 2018.

Karl Pearson. 1901. *On Lines and Planes of Closest Fit to Systems of Points in Space*. University College.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ondřej Pražák, Pavel Přibáků, Stephen Taylor, and Jakub Sido. 2020. UWB at SemEval-2020 Task 1: Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Julia Rodina and Andrey Kutuzov. 2020. Rusemshift: a dataset of historical lexical semantic change in russian. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*. Association for Computational Linguistics.

Dominik Schlechtweg, Anna Hätty, Marco del Tredici, and Sabine Schulte im Walde. 2019a. A Wind of Change: Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy. Association for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Dominik Schlechtweg, Cennet Oguz, and Sabine Schulte im Walde. 2019b. Second-order co-occurrence sensitivity of skip-gram with negative sampling. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 24–30, Florence, Italy. Association for Computational Linguistics.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana, USA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Hinrich Schütze and Jan Pedersen. 1993. A vector model for syntagmatic and paradigmatic relatedness. In *Proc. of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113, Oxford, England.

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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to diachronic conceptual change. *arXiv:1811.06278*.

Nina Tahmasebi and Thomas Risse. 2017. Finding individual word sense changes and their delay in appearance. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 741–749, Varna, Bulgaria.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188.

A Corpus details

The corpora are lemmatized and contain no punctuation, further pre-processing on the corpora by us is limited to removing low-frequency words. All words with a frequency below the value listed in row *min word freq.* in Table 1 are removed from the corpora. This is done to reduce noise and unwanted artifacts.

B Parameter settings

SGNS. We use common hyper-parameter settings: initial learning rate of 0.025, number of negative samples k=5 and no sub-sampling. Vector dimensionality d, window size w and number of training epochs e are varied in order to fine-tune model and methods. This is important as alignment methods like VI are highly dependent on the choice of e and d (Kaiser et al., 2020a). The following values are used: $w \in \{5, 10\}$, $e \in \{5, 10, 20, 30\}$, $d \in \{25, 50, 100, 200, 300, 500\}$. Due to the immense amount of possible parameter combinations we only ran each setting once.

PP was performed on the high-scores of each language, where we differentiate between different combinations of alignment, pre-training as well as if the matrices were STA or SEP post-processed.

SOT. As stated in Section 4.3, SEP is used in combination with post-alignment. We apply SOT with α values ranging from -1 to 1 in 0.1 increments on every baseline matrix with $d \in \{25, 50, 100, 200, 300, 500\}$.

MC+PCR. MC+PCR is performed using a parameter space of [0, 25] in order to examine the performance development over a growing number of PCs removed. It is important to note that using the parameter 0 results in only applying MC.