UWB at SemEval-2020 Task 1: Lexical Semantic Change Detection

Ondřej Pražák[* 1,2], **Pavel Přibáň**[* 1,2], **Stephen Taylor**[2], and **Jakub Sido**[1,2]

[1]NTIS – New Technologies for the Information Society,
[2]Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
`{ondfa, pribanp, taylor, sidoj}@kiv.zcu.cz`
`http://nlp.kiv.zcu.cz`

Abstract

In this paper, we describe our method for detection of lexical semantic change, i.e., word sense changes over time. We examine semantic differences between specific words in two corpora, chosen from different time periods, for English, German, Latin, and Swedish. Our method was created for the SemEval 2020 Task 1: *Unsupervised Lexical Semantic Change Detection.* We ranked 1st in Sub-task 1: binary change detection, and 4th in Sub-task 2: ranked change detection. Our method is fully unsupervised and language independent. It consists of preparing a semantic vector space for each corpus, earlier and later; computing a linear transformation between earlier and later spaces, using Canonical Correlation Analysis and Orthogonal Transformation; and measuring the cosines between the transformed vector for the target word from the earlier corpus and the vector for the target word in the later corpus.

1 Introduction

Language evolves with time. New words appear, old words fall out of use, the meanings of some words shift. The culture changes as well as the expected audience of the printed word. There are changes in topics, in syntax, in presentation structure. Reading the natural philosophy musings of aristocratic amateurs from the eighteenth century, and comparing with a monograph from the nineteenth century, or a medical study from the twentieth century, we can observe differences in many dimensions, some of which seem hard to study. Changes in word senses are both a visible and a tractable part of language evolution. Computational methods for researching words' stories have the potential of helping us understand this small corner of linguistic evolution. The tools for measuring these diachronic semantic shifts might also be useful for measuring whether the same word is used in different ways in synchronic documents. The task of finding word sense changes over time is called diachronic *Lexical Semantic Change (LSC)* detection. The task is getting more attention in recent years (Hamilton et al., 2016b; Frermann and Lapata, 2016; Schlechtweg et al., 2017). There is also the *synchronic LSC* task, which aims to identify domain-specific changes of word senses compared to general-language usage (Schlechtweg et al., 2019).

Tahmasebi et al. (2018) provides a comprehensive survey of techniques for the *LSC* task, as does Kutuzov et al. (2018). Schlechtweg et al. (2019) evaluated available approaches for *LSC* detection using the *DURel* dataset (Schlechtweg et al., 2018). Some of the methodologies for finding time-sensitive meanings were borrowed from information retrieval techniques in the first place. According to Schlechtweg et al. (2019), there are mainly three types of approaches. (1) Semantic vector spaces approaches (Gulordava and Baroni, 2011; Kim et al., 2014; Xu and Kemp, 2015; Eger and Mehler, 2016; Hamilton et al., 2016a; Hamilton et al., 2016b; Rosenfeld and Erk, 2018) represent each word with two vectors for two different time periods. The change of meaning is then measured by the cosine distance between the two vectors. (2) Topic modeling approaches (Wang and McCallum, 2006; Bamman and Crane, 2011; Wijaya and Yeniterzi, 2011; Mihalcea and Nastase, 2012; Cook et al., 2014; Frermann and Lapata, 2016; Schlechtweg and Walde, 2020) estimate a probability distribution of words over their different senses,

i.e., topics. (3) Clustering models (Mitra et al., 2015; Tahmasebi and Risse, 2017) are used to cluster words into clusters representing different senses.

We participated in the *SemEval 2020 Task 1: Unsupervised Lexical Semantic Change Detection* (Schlechtweg et al., 2020) competition. In this paper, we describe our solution and submitted systems for this competition. The task consists of two sub-tasks, a *binary classification* task (Sub-task 1) and a *ranking* task (Sub-task 2), which involve comparing usage of target words between two lemmatized corpora, each drawn from documents from a different time period for four languages: *English, German, Latin,* and *Swedish*. For both sub-tasks, only the target words and two corpora for each language were provided by organizers, no annotated data. The task is intended to be solved in a completely unsupervised way.

In the *binary classification* task, the goal is for two given corpora C_1 and C_2 (for time t_1 and t_2) and for a set of target words, decide which of these words changed or did not change their sense (semantic) between t_1 and t_2. *Change of sense* is whether the word lost or gained any sense between the two periods (corpora). The objective of the Sub-task 2 is for two given corpora C_1 and C_2, rank a set of target words according to their degree of lexical semantic change between t_1 and t_2. A higher rank means a stronger change. Target words are the same for both sub-tasks.

Because language is evolving, expressions, words, and sentence constructions in two corpora from different time periods about the same topic will be written in languages that are quite similar but slightly different. They will share the majority of their words, grammar, and syntax.

The main idea behind our solution is that we treat each pair of corpora C_1 and C_2 as different *languages* L_1 and L_2 even though that text from both corpora is written in the same language. We believe that these two languages L_1 and L_2 will be extremely similar in all aspects, including semantic. We train separate semantic space for each corpus and subsequently, we map these two spaces into one common cross-lingual space. We use methods for cross-lingual mapping (Brychcín et al., 2019; Artetxe et al., 2016; Artetxe et al., 2017; Artetxe et al., 2018a; Artetxe et al., 2018b) and thanks to the large similarity between L_1 and L_2 the quality of transformation should be high. We compute cosine similarity to classify and rank the target words, see Section 3 for details.

Our systems[1] ranked 1st out of 33 teams in Sub-task 1 with an average accuracy of 0.687, and 4th out of 32 teams in Sub-task 2 with an average Spearman's rank correlation of 0.481.

2 Data

The corpora are drawn from several sources, described in (Schlechtweg et al., 2020). Table 1 shows periods and sizes. For each language, items in the earlier corpus are separated from items in the later one by at least 46 years (German) and as much as 2200 years (Latin). All corpora are lemmatized, punctuation is removed, and sentences are randomly reordered. For English, target words have been marked with their part-of-speech. Two example sentences:(1) "*there be no pleasing any of you do as we may*", and (2) "*rise upon the instant in his stirrup the bold cavalier hurl with a sure and steady hand the discharge weapon in the face_nn of his next opponent*". (1) illustrates a failure of lemmatization – the word 'pleasing', which is a form of the verb 'please'; (2) shows a target word 'face', marked as a noun. Less than 10% of the English sentences contain a target word.

| Language | Corpus 1 | | Corpus 2 | | # Targets |
	Period	# Tokens	Period	# Tokens	
English	1810-1860	6 559 657	1960-2010	6 769 814	37
German	1800-1900	70 244 484	1946-1990	72 397 546	48
Latin	200BC-1BC	1 751 405	100AD-present	9 417 033	40
Swedish	1790-1830	71 091 465	1895-1903	110 792 658	31

Table 1: Corpus statistics. The last column *# Targets* denotes the number of target words.

Lemmatization reduces the vocabulary so that there are more examples of each word. It also introduces

---

[1]Our code is available at `https://github.com/pauli31/SemEval2020-task1`

ambiguity; the decisions to add a POS tag to English target words and retain German noun capitalization shows that the organizers were aware of this problem.

3   System Description

First, we train two semantic spaces from corpus C_1 and C_2. We represent the semantic spaces by a matrix $\mathbf{X}^s$ (i.e., a source space s) and a matrix $\mathbf{X}^t$ (i.e, a target space t)[2] using word2vec Skip-gram with negative sampling (Mikolov et al., 2013). We perform a cross-lingual mapping of the two vector spaces, getting two matrices $\hat{\mathbf{X}}^s$ and $\hat{\mathbf{X}}^t$ projected into a shared space. We select two methods for the cross-lingual mapping *Canonical Correlation Analysis (CCA)* using the implementation from (Brychcín et al., 2019) and a modification of the *Orthogonal Transformation* from *VecMap* (Artetxe et al., 2018b). Both of these methods are linear transformations. In our case, the transformation can be written as follows:

$$\hat{\mathbf{X}}^s = \mathbf{W}^{s \to t} \mathbf{X}^s \tag{1}$$

where $\mathbf{W}^{s \to t}$ is a matrix that performs linear transformation from the source space s (matrix $\mathbf{X}^s$) into a target space t and $\hat{\mathbf{X}}^s$ is the source space transformed into the target space t (the matrix $\mathbf{X}^t$ does not have to be transformed because $\mathbf{X}^t$ is already in the target space t and $\mathbf{X}^t = \hat{\mathbf{X}}^t$).

Generally, the CCA transformation transforms both spaces $\mathbf{X}^s$ and $\mathbf{X}^t$ into a third shared space o (where $\mathbf{X}^s \neq \hat{\mathbf{X}}^s$ and $\mathbf{X}^t \neq \hat{\mathbf{X}}^t$). Thus, CCA computes two transformation matrices $\mathbf{W}^{s \to o}$ for the source space and $\mathbf{W}^{t \to o}$ for the target space. The transformation matrices are computed by minimizing the negative correlation between the vectors $\mathbf{x}_i^s \in \mathbf{X}^s$ and $\mathbf{x}_i^t \in \mathbf{X}^t$ that are projected into the shared space o. The negative correlation is defined as follows:

$$\underset{\mathbf{W}^{s \to o}, \mathbf{W}^{t \to o}}{\operatorname{argmin}} - \sum_{i=1}^{n} \rho(\mathbf{W}^{s \to o} \mathbf{x}_i^s, \mathbf{W}^{t \to o} \mathbf{x}_i^t) = - \sum_{i=1}^{n} \frac{cov(\mathbf{W}^{s \to o} \mathbf{x}_i^s, \mathbf{W}^{t \to o} \mathbf{x}_i^t)}{\sqrt{var(\mathbf{W}^{s \to o} \mathbf{x}_i^s) \times var(\mathbf{W}^{t \to o} \mathbf{x}_i^t)}} \tag{2}$$

where cov the covariance, var is the variance and n is a number of vectors. In our implementation of CCA, the matrix $\hat{\mathbf{X}}^t$ is equal to the matrix $\mathbf{X}^t$ because it transforms only the source space s (matrix $\mathbf{X}^s$) into the target space t from the common shared space with a pseudo-inversion, and the target space does not change. The matrix $\mathbf{W}^{s \to t}$ for this transformation is then given by:

$$\mathbf{W}^{s \to t} = \mathbf{W}^{s \to o}(\mathbf{W}^{t \to o})^{-1} \tag{3}$$

The submissions that use CCA are referred to as **cca-nn, cca-bin, cca-nn-r** and **cca-bin-r** where the **-r** part means that the source and target spaces are reversed, see Section 4. The **-nn** and **-bin** parts refer to a type of threshold used only in the Sub-task 1, see Section 3.1. Thus, in Sub-task 2, there is no difference for the following pairs of submissions: **cca-nn** – **cca-bin** and **cca-nn-r** – **cca-bin-r**.

In the case of the *Orthogonal Transformation*, the submissions are referred to as **ort** & **uns**. We use Orthogonal Transformation with a supervised seed dictionary consisting of all words common to both semantic spaces. (**ort**). The transformation matrix $\mathbf{W}^{s \to t}$ is given by:

$$\underset{\mathbf{W}^{s \to t}}{\operatorname{argmin}} \sum_{i}^{|V|} (\mathbf{W}^{s \to t} \mathbf{x}_i^s - \mathbf{x}_i^t)^2 \tag{4}$$

under the hard condition that $\mathbf{W}^{s \to t}$ needs to be orthogonal, where V is the vocabulary of correct word translations from source to target space. The reason for the orthogonality constraint is that linear transformation with the orthogonal matrix does not squeeze or re-scale the transformed space. It only rotates the space, thus it preserves most of the relationships of its elements (in our case it is important that orthogonal transformation preserves angles between the words, so it preserves the cosine similarity).

Artetxe et al. (2018b) also proposed a method for automatic dictionary induction. This is a fully unsupervised method for finding orthogonal cross-lingual transformations. We used this approach for our **uns** submissions.

---

[2] The source space $\mathbf{X}^s$ is created from the corpus C_1 and the target space $\mathbf{X}^t$ is created from the corpus C_2.

Finally in all transformation methods, for each word w_i from the set of target words T, we select its corresponding vectors $\mathbf{v}_{w_i}^s$ and $\mathbf{v}_{w_i}^t$ from matrices $\hat{\mathbf{X}}^s$ and $\hat{\mathbf{X}}^t$, respectively ($\mathbf{v}_{w_i}^s \in \hat{\mathbf{X}}^s$ and $\mathbf{v}_{w_i}^t \in \hat{\mathbf{X}}^t$), and we compute cosine similarity between these two vectors. The cosine similarity is then used to generate an output for each sub-task. For Sub-task 2, we compute the degree of change for word w_i as $1 - cos(\mathbf{v}_{w_i}^s, \mathbf{v}_{w_i}^t)$. The settings of hyper-parameters for both methods give us several combinations, see sections 4 and 5.

3.1 Binary System

The organizers provided a definition for binary change in terms of specific numbers of usages of senses of a target word. We decided against attempting to model and group individual word usages. Instead, we decided to use the continuous scores from Sub-task 2 for the binary task. We assume that there is a threshold t for which the target words with a continuous score greater than t changed meaning and words with the score lower than t did not. We know that this assumption is generally wrong (because using the threshold we introduce some error into the classification), but we still believe it holds for most cases and it is the best choice. In order to examine this assumption after the evaluation period, we computed the accuracy of the gold ranking scores with the optimal threshold (selected to maximize test accuracy). As you can see in Table 2, even if an optimal threshold is chosen, the best accuracy that can be achieved is, on average, 87.6%.

	Avg	Eng	Ger	Lat	Swe
acc	.876	.838	.854	.875	.936
t	-	.207	.190	.235	.244

Table 2: Sub-task 1 accuracy with gold ranking and optimal thresholds.

In order to find the threshold t, we tried several approaches. We call the first approach *binary-threshold* (**-bin** in Table 3). For each target word w_i we compute cosine similarity of its vectors $\mathbf{v}_{w_i}^s$ and $\mathbf{v}_{w_i}^t$, then we average these similarities for all words. The resulting averaged value is used as the threshold. Another approach called *global-threshold* (**-gl**) is done similarly, but the average similarity is computed across all four languages. The last approach, called *nearest-neighbors* (**-nn**), compares sets of nearest neighbours. For target word w_i we find 100 nearest (most similar) words from both transformed spaces (matrices $\hat{\mathbf{X}}^s$ and $\hat{\mathbf{X}}^t$), getting two sets of nearest neighbours N^s and N^t for each target word. Then, we compute the size of the intersection of these two sets for each target word. From the array of intersection sizes, we select the second highest value[3], and we divide it by two. The resulting number is a threshold for all target words. If the size of a target word's intersection is greater or equal to the threshold, we classify the word *unchanged*; otherwise *changed*. This threshold is set for each language independently.

4 Experimental Setup

To obtain the semantic spaces, we employ Skip-gram with negative sampling (Mikolov et al., 2013). We use the Gensim framework (Řehůřek and Sojka, 2010) for training the semantic spaces. For the final submission, we trained the semantic spaces with 100 dimensions for five iterations with five negative samples and window size set to five. Each word has to appear at least five times in the corpus to be used in the training. For all **cca-** submissions, we build the translation dictionary for the cross-lingual transformation of the two spaces by removing the target words from the intersection of their vocabularies.

For **cca-nn-r** and **cca-bin-r** we change the direction of the cross-lingual transformation. The initial setup for the transformation is that the source space is the space of the older corpus C_1 (represented by the matrix $\mathbf{X}^s$), and the target space is the semantic space of the later corpus C_2 (represented by the matrix $\mathbf{X}^t$). We reversed the order, and we use the matrix $\mathbf{X}^t$ as the source space, which is transformed into semantic space (matrix $\mathbf{X}^s$) of the older corpus, i.e. into the original source space.

---

[3]We observed that the largest size of intersection is usually significantly greater than the other sizes so we select the second highest number. We are aware that this is a very simple and data-dependent approach and could be further improved.

The two other methods are Orthogonal Transformations with identical words as the seed dictionary (**ort**) and unsupervised transformation (**uns**) from the *VecMap* tool. In these methods, we use median similarity as the threshold for the binary task. We experimented with a separate threshold for each language (**bin** suffix) and with a single global threshold for all the languages (**gl** suffix).

The sample data provided with the task did not contain any labeled development target words, so we used the DURel (Schlechtweg et al., 2018) corpus and WOCC (Schlechtweg et al., 2019) corpus[4] to validate our solutions' performance. The WOCC corpus contains lemmatized German sentences from *Deutsches Textarchiv* (Deutsches Textarchiv, 2017) for two time periods *1750-1799* and *1850-1899* , which we used to train the semantic spaces. We evaluated our system on the DURel corpus, and we achieved 0.78 of Spearman's correlation rank with settings that correspond to the column named **cca-nn-r** in Table 3.

5 Results

We submitted eight different submissions for the Sub-task 1 and four for the Sub-task 2 obtained by CCA and the VecMap tool. We also submitted predictions based on Latent Dirichlet Allocation (Blei et al., 2003) (LDA), but because of its poor results and limited paper size, we do not include the description here. The results, along with the ranking[5] are shown in Table 3. The bold results denote the best result we achieved for each language, and the underlined results were used in our team's final ranking.

Task		cca-nn	cca-nn-r	cca-bin	cca-bin-r	ort-bin	ort-gl	uns-bin	uns-gl
	Avg	.572 (65)	.607 (28)	.663 (6)	<u>**0.687 (1)**</u>	.634 (21)	.639 (16)	.659 (11)	.655 (13)
	Eng	.595 (6)	**.730 (1)**	.595 (6)	.622 (5)	.568 (7)	.622 (5)	.568 (7)	.622 (5)
1	Ger	.521 (14)	.542 (13)	**.812 (1)**	.750 (3)	.750 (3)	.688 (6)	.750 (3)	.646 (8)
	Lat	.625 (5)	.575 (7)	.600 (6)	**.700 (2)**	.575 (7)	.600 (6)	.675 (3)	.675 (3)
	Swe	.548 (8)	.581 (7)	.645 (5)	**.677 (4)**	.645 (5)	.645 (5)	.645 (5)	.677 (4)
	Avg	.411 (14)	.469 (7)	-	-	<u>**.481 (6)**</u>	-	.455 (8)	-
	Eng	.279 (33)	**.421 (5)**	-	-	.367 (11)	-	.365 (12)	-
2	Ger	**.707 (4)**	.696 (7)	-	-	.697 (6)	-	.687 (9)	-
	Lat	.181 (66)	.251 (58)	-	-	**.254 (55)**	-	.181 (65)	-
	Swe	.476 (10)	.509 (8)	-	-	**.604 (1)**	-	.587 (3)	-

Table 3: Results for our final submissions.

The *cca-bin-r* system settings achieved the first-place rank out of 189 submissions by 33 teams on Sub-task 1, with an absolute accuracy of 0.687. The majority class was *unchanged*, and always choosing it would have given a score of 0.571. The submissions with the top 15 best accuracies (four teams) had scores ranging from 0.659 to 0.687 and assorted scores on the different languages. The four of our systems using the **-bin** approach, which included the top-scoring system, have a mean percentile of 94.6; the two **-gl** strategy systems had a mean percentile of 92.7, and the two systems with the **-nn** strategy had a mean percentile of 75.5. On Sub-task 2, our best system achieved the fourth-place rank out of 33 teams. It had an average correlation against the gold ranking for all four languages of 0.481.

Since the threshold strategy was used only for Sub-task 1, there is no difference in results for Sub-task 2 in Table 3 in the following pairs of columns **cca-nn** – **cca-bin**, **cca-nn-r** – **cca-bin-r**, **ort-bin** – **ort-gl** and **uns-bin** – **uns-gl**. Thus, for Sub-task 2 we provide numbers only in the **cca-nn**, **cca-nn-r**, **ort-bin** and **uns-bin** columns.

Examining Table 3, the ranking scores for Latin are not only worse than the other languages in absolute terms, but their position relative to other submissions is also much worse. The Latin corpora have several anomalies, but only the small size of the earlier corpus (a third the size of the next smallest corpus) seems likely to be a problem. For example, although both Latin corpora have a much larger proportion of short lines than the others, a measurement of the mean context size for target words shows that for all of the corpora, the mean context size for target words is between 8.38 (German 2) and 8.95 (German 1).

---

[4]Both available at www.ims.uni-stuttgart.de/en/research/resources/corpora/wocc

[5]The ranking in the table is the ranking among 186 submissions and does not correspond to ranking among 33 teams because team ranks are based on their best submission.

A different problem is syntax. The lemmatization of the corpora is no obstacle to the English reader, who can usually guess the correct form, because word-order in English (and also in Swedish and German) is somewhat rigid, and word inflexions are largely replaced by separate words, which have their own lemmas. In Latin, word-order is quite flexible, and different authors have different preferred schemes for emphasis and euphony. Two adjacent words are not required to be part of each other's context, and the most semantically related words in a sentence may be separated precisely to emphasize them.

We performed post-evaluation experiments with word embedding vector size in order to discover its effect on system performance, see Figure 1 containing visualization of results for Sub-task 2. It shows that the optimal size for this task is in most cases between 100 and 175 for Sub-task 2, and between 75 and 125 for Sub-task 1 (not shown here). We also tried using fastText (Bojanowski et al., 2017) instead of word2vec Skip-gram to get the semantic space, with settings that correspond to the ones we used for the final submissions. The performance using fastText was not improved in general, but for some settings we obtained slightly better results. Other experiments suggest that Latin results can benefit when the context size is increased during the training of the semantic spaces. According to submitted results, it seems that CCA method with reversed order (columns **cca-nn-r**, **cca-bin-r** in Table 3) works better than without reversed but from the Figure 1 is evident that it is valid only for English and Latin.
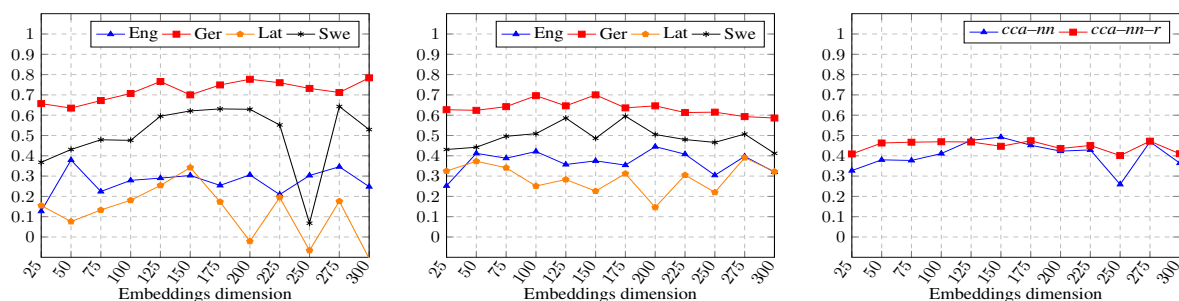
(a) Nearest neighbors mapping – *cca-nn* (b) Reversed nn. mapping – *cca-nn-r* (c) Average of all languages

Figure 1: Performance using the CCA method with different word embeddings size for Sub-task 2.

6   Conclusion

Applying a threshold to semantic distance is a sensible architecture for detecting the binary semantic change in target words between two corpora. Our *binary-threshold* strategy succeeded quite well. We did have a small advantage in that the list of target words turned out to be nearly equally divided between *changed* (0.57) and *unchanged* (0.43) words. Thus, choosing thresholds assuming that the division was 50/50 was not a severe problem. Our experiments also reveal the limits of a threshold strategy, as shown in Table 2. Second, although our systems did not win Sub-task 2 change-ranking competition, they show that our architecture is a strong contender for this task when there is sufficient data to build good vector semantic spaces. The results for Latin illustrate that there is still room for other, less statistical approaches; one might have predicted, falsely, that 1.7M tokens was plenty of data. The variety of experiments which were possible in the post-evaluation period suggest that the corpora developed for the task will encourage work in the area. In future work, we expect to focus on other cross-lingual techniques and other methods of measuring similarity between words besides cosine similarity.

Acknowledgments

References

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *Computing Research Repository*, arXiv:1602.01925.

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, November. 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 (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada, July. 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 (AAAI-18)*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia, July. Association for Computational Linguistics.

David Bamman and Gregory Crane. 2011. Measuring historical word sense variation. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, JCDL '11, page 1–10, New York, NY, USA. Association for Computing Machinery.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tomáš Brychcín, Stephen Taylor, and Lukáš Svoboda. 2019. Cross-lingual word analogies using linear transformations between semantic spaces. *Expert Systems with Applications*, 135:287–295.

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

Deutsches Textarchiv. 2017. Grundlage für ein referenzkorpus der neuhochdeutschen sprache. Herausgegaben von der Berlin-Brandenburgischen Akademie der Wissenschaften. http://deutschestextarchiv.de.

Steffen Eger and Alexander Mehler. 2016. On the linearity of semantic change: Investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 52–58, Berlin, Germany, August. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471.

Lea Frermann and Mirella Lapata. 2016. A Bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, 4:31–45.

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

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121, Austin, Texas, November. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany, August. Association for Computational Linguistics.

David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, December.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 61–65, Baltimore, MD, USA, June. Association for Computational Linguistics.

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, August. Association for Computational Linguistics.

Rada Mihalcea and Vivi Nastase. 2012. Word epoch disambiguation: Finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 259–263, Jeju Island, Korea, July. Association for Computational Linguistics.

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

Sunny Mitra, Ritwik Mitra, Suman Kalyan Maity, Martin Riedl, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2015. An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering*, 21(5):773–798.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. http://is.muni.cz/publication/884893/en.

Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. 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 474–484, New Orleans, Louisiana, June. Association for Computational Linguistics.

Dominik Schlechtweg and Sabine Schulte im Walde. 2020. Simulating lexical semantic change from sense-annotated data. In A. Ravignani, C. Barbieri, M. Martins, M. Flaherty, Y. Jadoul, E. Lattenkamp, H. Little, K. Mudd, and T. Verhoef, editors, *The Evolution of Language: Proceedings of the 13th International Conference (EvoLang13)*.

Dominik Schlechtweg, Stefanie Eckmann, Enrico Santus, Sabine Schulte im Walde, and Daniel Hole. 2017. German in flux: Detecting metaphoric change via word entropy. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 354–367, Vancouver, Canada, August. Association for Computational Linguistics.

Dominink Schlechtweg, Sabine Schulte im Wlade, and Stefanie Eckmann. 2018. Diachronic usage relatedness (durel): A framework for the annotation of lexical semantic change. In *Proceedings of NAACL-HLT 2018*, pages 169–174.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. 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, July. 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 (SemEval-2020)*, Barcelona, Spain, Sep. Association for Computational Linguistics.

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, RANLP 2017*, pages 741–749, Varna, Bulgaria, September. INCOMA Ltd.

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

Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 424–433, New York, NY, USA. Association for Computing Machinery.

Derry Tanti Wijaya and Reyyan Yeniterzi. 2011. Understanding semantic change of words over centuries. In *Proceedings of the 2011 International Workshop on DETecting and Exploiting Cultural DiversiTy on the Social Web*, DETECT '11, page 35–40, New York, NY, USA. Association for Computing Machinery.

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