JCT at SemEval-2020 Task 1: Combined Semantic Vector Spaces Models for Unsupervised Lexical Semantic Change Detection

Efrat Amar, Chaya Liebeskind
Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
21 Havaad Haleumi St., P.O.B. 16031
9116001 Jerusalem, Israel
efrati.amar@gmail.com, liebchaya@gmail.com

Abstract

In this paper, we present our contribution in SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection, where we systematically combine existing models for unsupervised capturing of lexical semantic change across time in text corpora of German, English, Latin and Swedish. In particular, we analyze the score distribution of existing models. Then we define a general classification threshold, adjust it independently to each of the models and measure the models' score certainty. Finally, using both the threshold and score certainty, we aggregate the models for the two sub-tasks: binary classification and ranking.

1 Introduction

Over the last decade, research on detection of lexical semantic change has increased. Many studies were performed on various languages, corpora and periods. Two years ago, two literature surveys on computational approaches to *Lexical Semantic Change (LSC)* (Kutuzov et al., 2018; Tahmasebi et al., 2018) were published. Last year Schlechtweg et al. (2019) first systematically compared a broad variety of LSC detection models on two data sets of different periods and domains and Shoemark et al. (2019) proposed a new evaluation framework for semantic change detection using word embeddings.

To facilitate the comparison of different systems, SemEval-2020 Task 1 (Schlechtweg et al., 2020) introduced a simple evaluation framework for unsupervised lexical semantic change detection in text corpora of German, English, Latin and Swedish. The task relies on the comparison of two time periods for each language. We participated in two sub-tasks: a classification task, where we decide which words lost or gained senses between the periods, and a ranking task, where we rank a set of target words according to their degree of lexical semantic change between the periods.

Given the large number of models that have already been explored, we built a system which systematically combines existing models in the unsupervised setting of the LSC detection task. Since no tuning data is available, we minimized the parameters number of our system.

This paper is organized as follows: First, in Section 2, we describe the existing LSC detection models that we have combined. Then, in Section 3, we analyze the score distribution of the models in order to learn the general behaviour of words in our corpora. We aim to estimate the amount of words that changed their meaning between the periods. Next, we define a general classification threshold percentile (CT) parameter and adapt it to each model separately. We use the CT parameter to measure the models' score certainty too, and filtered models with certainty low than a minimal required decision certainty. The minimal required decision certainty (MCR) is also a parameter of our system. Finally, we present our aggregation methods for the two sub-tasks: classification and ranking. Our system results are detailed in Section 4, followed by conclusions in Section 5.

2 Related Work

In this section, we shortly describe the work of Schlechtweg et al. (2019) which covers a wide range of LSC detection models. Then, we summarize the models that we have combined in our system.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 90–97
Barcelona, Spain (Online), December 12, 2020.

Schlechtweg et al. (2019) made a comprehensive comparison between the results of the diverse existing LSC Detection models and firstly ran the models under one evaluation task and data. Schlechtweg et al. (2019) tested the methods for semantic change detection both across time (Diachronic) and across domains (Synchronic). For the Diachronic task they used the evaluation framework and the DURel German corpora (introduced in (Schlechtweg et al., 2018)). The framework was expanded for the Synchronic task using the SURel German corpora (Hätty et al., 2019).

Existing LSC Detection models are based on three methods for meaning representations: semantic vector spaces (Hamilton et al., 2016a; Hamilton et al., 2016b; Hellrich and Hahn, 2016; Rosenfeld and Erk, 2018), topic distributions (Cook et al., 2014; Fremann and Lapata, 2016), and sense clusters (Mitra et al., 2015). In semantic vector spaces, each term is represented as two vectors indicating its co-occurrence statistics at different eras. Then, the semantic change is commonly measured by either similarity measures, such as Cosine similarity, or contextual measures. In topic distributions, each term is modeled as a probability distribution over different topics, or term senses. Then, the semantic change is measured based on the senses' frequency of use. Whereas, sense clustering and topic models are similar in their mapping (uses to senses) and semantic change measures, in sense clustering, some contextual property is used to assign all uses of a term into sense clusters.

Schlechtweg et al. (2019) focused on two meaning representations: semantic vector spaces and topic distributions. They tested a list of known LSC Detection models with different combinations of semantic representations, alignment methods and detection measures. They experimented various parameter settings for comparing the models' predictions with the true results. Schlechtweg et al. (2019) concluded that we could use the same modelling methods for both Diachronic and Synchronic LSC Detection. In addition, all model predictions had a strong positive correlation with the true results. The model with the best performance was Skip-Gram with Orthogonal Procrustes alignment and Cosine Distance (SGNS+OP+CD).

Since Schlechtweg et al. (2019) observed that topic distributions models (SCAN) (Frermann and Lapata, 2016) have poor and unstable performance, in our system we integrated only the Semantic Vector Spaces models (reminded at Schlechtweg et al. (2019)).

The combined models' representations were: Raw Count, Positive Pointwise Mutual Information (PPMI) (Church and Hanks, 1990), Singular Value Decomposition (SVD) (Golub and Van Loan, 1996), Random Indexing (RI) (Eckart and Young, 1936; Basile et al., 2015) and Skip-Gram with Negative Sampling (SGNS) (Mikolov et al., 2013; Baroni et al., 2014; Levy et al., 2015).

The integrated alignment methods were: Column Intersection (CI) (Hamilton et al., 2016b), Shared Random Vectors (SRV) (Basile et al., 2015), Orthogonal Procrustes (OP) (Hamilton et al., 2016b; Artetxe et al., 2017), Vector Initialization (VI) (Kim et al., 2014) and Word Injection (WI) (Ferrari et al., 2017).

Two types of measures were combined: Similarity measures (Cosine Distance (CD) and Local Neighbourhood Distance(LND) (Hamilton et al., 2016a)) and Dispersion measures (Frequency Difference (FD), Type Difference (TD) and Entropy Difference (HD) (Shannon, 1948; Santus et al., 2014)). Table 1 summarizes the list of models' combinations used by our system.

For running these models, we used the scripts for vector space representation, alignment, measuring LSC and evaluation that available at `https://github.com/Garrafao/LSCDetection`. We set the same parameters for all models' combination (windowSize = 4, k=1, ts=None, dims=300, eps=5).

3 System Description

Two time-specific corpora for each of the four languages, German, English, Latin and Swedish, were provided by the task organizers. Each line contains one sentence, where the punctuation was eliminated and each token was replaced by its lemma. Within each corpus sentences were shuffled randomly. One-word sentences were removed form the Latin corpus and for the other languages, sentences with less than 10 tokens were removed. Due to the big size of the corpora, we removed low-frequency words for improving the efficiency of the models. The input of the system for each language is a corpus pair and a list of target words. Our system scripts are publicly available on GitHub `https://github.com/efratiamar/CombinedModelsLSC`.

	Representation	Alignment	Measure
1	COUNT	CI	CD
2	COUNT	CI	LND
3	COUNT		FD
4	COUNT		HD
5	COUNT		TD
6	COUNT	WI	CD
7	COUNT	WI	LND
8	PPMI	CI	CD
9	PPMI	CI	LND
10	PPMI	WI	CD
11	PPMI	WI	LND
12	RI	OP	CD
13	RI	OP	LND
14	RI	SRV	CD
15	RI	SRV	LND
16	RI	WI	CD
17	RI	WI	LND
18	SGNS	OP	CD
19	SGNS	OP	LND
20	SGNS	VI	CD
21	SGNS	VI	LND
22	SGNS	WI	CD
23	SGNS	WI	LND
24	SVD	OP	CD
25	SVD	OP	LND
26	SVD	WI	CD
27	SVD	WI	LND

Table 1: Models' combinations integrated by our system.

3.1 Analyzing the score distribution of the LSC detection models

Since we were not given any information on the amount of words that changed their meaning, we analyzed the score distribution of the LSC detection models to learn the general behavior of words in our corpora.

First, for each language, we randomly selected $n = 200$ words with more that 30 appearances in each of the two periods. Then, for each of the LSC detection models, we applied the following steps:

1. Calculate the scores for all the 200 words.
2. Draw a histogram of the scores, as illustrated in Figure 1.
3. Calculate the skewness of the score distribution. Skewness characterizes the degree of asymmetry of a distribution around its mean. Positive skewness indicates a distribution with an asymmetric tail extending toward more positive values. Negative skewness indicates a distribution with an asymmetric tail extending toward more negative values. Skewness is defined as

$$Skewness = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \overline{x}}{s}\right)^3 \tag{1}$$

   where $\overline{x}$ is the sample average and s is the standard derivation.

Our exploration revealed that most of the models have positive skewness. This implies that there are more words that preserved their meaning than words that changed their meaning. The percentage of models with positive skewness for English, German, Latin, and Swedish are 62.96%, 81.48%, 74.07% and 62.96%, respectively. Over 62% of the models in all languages have positive skewness. Therefore, in our system, we combined only models with positive skewness.

3.2 Setting thresholds in an unsupervised setting

The setting of the LSC detection task is unsupervised, there is no labeled data nor training set. To eliminate the need for tuning parameters for each LSC model separately, we defined a general classification threshold (CT) parameter in terms of percentile and adjusted it to the various LSC models. For example, if the threshold parameter is set to 90%, for each model, we calculated its numeric value which corresponds to the score that 90% of the scores in our random sample ($n = 200$) are lower than it.

For each model separately, we set a numeric classification threshold (NCT). The NCT is used for making a decision whether a word has changed it meaning or not as well as for measuring the certainty of our decision, as detailed next.

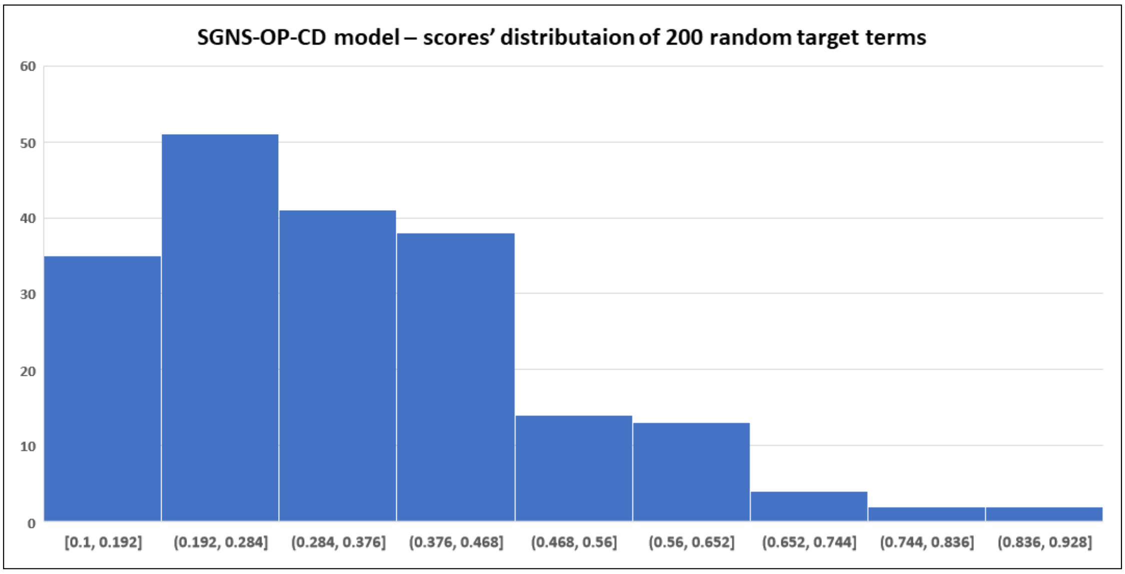

Figure 1: A histogram with positive skewness of the SGNS-OP-CD model on $n = 200$ random terms

3.3 Measuring the score's decision certainty

After setting a numeric classification threshold (NCT) for each model based on our random sample and the CT parameter, we took the input of the LSC detection task, a list of target terms and calculate the models' scores. Then, for each target and model, we compared its score with its numeric classification threshold. If the score was higher than the numeric classification threshold, we assumed the target term has changed its meaning. Next, for each target and model, we measured the decision certainty for each score in the following way:

1. Calculate the percentile of the model score ($percentile(score)$) based on our random sample.

2. Calculate the distance between the score percentile and the model's classification threshold (CT). $distance =\mid percentile(score) - CT \mid$ (i.e., the integral of some part of the curve).

3. Divide the distance by the range size, where the range depends on the placement of the score percentile in relation to the classification threshold (CT):

$$certainty(score) = \begin{cases} \frac{distance}{100-CT}, & \text{if } score \geq NCT \\ \frac{distance}{CT}, & \text{otherwise} \end{cases}$$

The impact of this division is illustrated in Figure 2. Since we uniformly divided all the values on the same side of the classification threshold, scores above the threshold were weakly affected, while scores below the threshold were strongly affected. In future work, we plan to apply a differential division method.

3.4 Binary Classification

As detailed in the previous section, the binary classification for each model was determined by comparison of the model score with the model numeric classification threshold. Additionally, for each classification, we calculated its decision certainty.

A minimal required decision certainty (in percentage) is a parameter of our system, termed MRC. First, For each target term, we ran all the models and got a binary classification for each of them. Then, we filtered models with decision certainty below the MRC parameter. Finally, we applied the majority rule, a decision rule that selects alternatives with a majority, i.e. more than half of the votes.

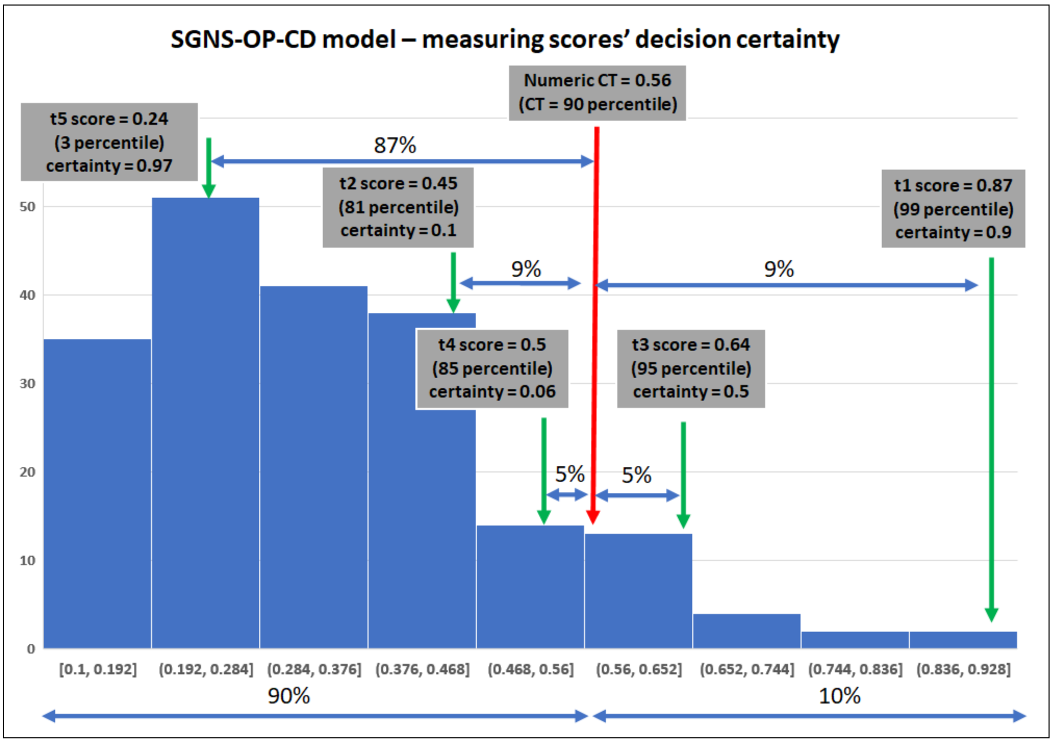

Figure 2: An illustration of the certainty calculation for 5 target terms scores, with CT=90%

3.5 Ranking

For the ranking task, to rank a set of target words according to their degree of lexical semantic change, we applied a similar approach.

First, for each target term, we ran all the models, got a score for each of them and normalized the models' scores to values between 0 and 1. Then, we filtered models with certainty below the MRC parameter. Finally, we calculated a weighted average. Our weighted average takes into account the scores' certainty by multiplying each score with its certainty. Thus, models with higher certainty had more affect on the target term ranking.

Since the scores of each model had a different scale, it was essential to normalize the scores before averaging them.

4 Results

Given the unsupervised LSC detection task, our concept was to use a minimum number of parameters beyond the models' parameters. Our system used two parameters (see Section 3): general classification threshold (CT) and minimal required decision certainty (MRC). In Table 2 we report our system results in the post-evaluation phase for the two sub-tasks with different configuration settings. For task 1, binary classification, we report the Accuracy (ACC) and for task 2, ranking, we report the Spearman correlation (SPR).

In the evaluation phase our system achieved the highest score with configuration number 4 (CT=90, MRC=0.5). Our system was ranked 8^{th} in task 1 (ACC=0.636) and 13^{th} in task 2 (SPR=0.254). As seen in Table 2, the score for this configuration in the post evaluation phase, is higher (ACC=0.647, SPR=0.283). The reason for this gap is that in the post-evaluation phase we improved our system by normalizing the models' scores to values between 0 and 1, as explained in Section 3.5.

In the post-evaluation phase, we realized that low CT and MCR achieved higher accuracy, but lower spearman correlation. As seen in Table 2, the best configuration for task 1 is configuration no. 8 (CT=80, MRC=0.4) with ACC=0.664 and SPR=0.321. Whereas, in task 2, the best configuration is configuration no. 12 (CT=70, MRC=0.3) with ACC=0.632, SPR=0.355.

	System Conf.		English		German		Latin		Swedish		AVG for all languages	
	CT	MRC	ACC	SPR	ACC	SPR	ACC	SPR	ACC	SPR	ACC	SPR
1	95	0.5	0.649	**0.35**	0.625	0.367	0.525	0.588	0.71	0.047	0.627	0.338
2	95	0.4	0.649	0.263	0.625	0.352	0.525	0.607	0.71	0.069	0.627	0.323
3	95	0.3	0.622	0.172	0.604	0.298	0.5	**0.644**	**0.742**	0.03	0.617	0.286
4	90	0.5	0.649	0.208	0.729	0.322	0.5	0.633	0.71	-0.032	0.647	0.283
5	90	0.4	**0.676**	0.126	0.667	0.429	0.525	0.59	0.71	-0.001	0.644	0.286
6	90	0.3	0.649	0.082	0.625	0.416	0.5	0.569	0.71	0.071	0.621	0.285
7	80	0.5	0.649	0.219	0.75	0.491	0.5	0.53	0.645	0.058	0.636	0.325
8	80	0.4	0.649	0.212	**0.771**	0.452	0.525	0.513	0.71	0.107	**0.664**	0.321
9	80	0.3	0.622	0.186	0.729	0.478	**0.55**	0.507	0.71	0.111	0.653	0.32
10	70	0.5	0.595	0.139	0.708	0.553	0.525	0.513	0.613	0.104	0.61	0.328
11	70	0.4	0.595	0.161	0.75	0.556	0.525	0.494	0.613	0.13	0.621	0.335
12	70	0.3	0.622	0.168	0.729	**0.571**	0.5	0.501	0.677	**0.18**	0.632	**0.355**

Table 2: Our system results in the post-evaluation phase for the two sub-tasks with different configuration settings. In each column, the result with the maximum score is bolded.

We also looked at what the results could have been if we were trying to fine-tune the system parameters (CT and MCR) and set for each language the configuration with the highest performance. In this case, our system obtains an average accuracy of 0.685 and average Spearman correlation of 0.436. This can be deduced from Table 2 as follows: if for each language we select the bolded configuration with the maximum ACC, we can see that their averaged ACC is 0.685. In the same way an average of 0.436 will be obtained in the SPR column.

To test the performance of our system, we analyzed the results of the best configurations (no.8 and no. 12). We compared the score of each model separately to the score of our system that weighs all the models' scores together. We found that there is no consistent behaviour. In other words, there was not any model that consistently outperformed all the other models or our weighted score. In the case of the German and the Swedish languages, the SGNS-based models produced better results than the other models. In the other languages, other models performed better. For example: In task 1, the PPMI-WI-LND and COUNT-TD models produced the best results in English, while in the Swedish language, the best results were obtained by the PPMI-WI-LND and COUNT-TD models. In task 2, the SGNS-WI-CD model produced the best results in English, while in the Swedish language, it was the SGNS-OP-CD model.

We also noticed that dispersion measures models, which strongly rely on frequency, had low performance. This could be resulted from the fact that the organizers' controlled each test set for frequency (which we could not know before they published the task description paper).

5 Conclusions and Future Work

We have implemented a system that systematically incorporates existing models to identify LSC over time in text corpora of four languages. We evaluated the score distribution of existing models, suggested a general classification threshold and applied it to each of the models individually. We calculated the models' score certainty and used it to aggregate the models. In the evaluation phase of the SemEval-2020 Task 1, our system was ranked 8^{th} and 13^{th} in the classification and ranking sub-tasks, respectively.

We plan to investigate additional aggregation methods and explore the impact of the individual models on the combined system to improve our system results. We also plan to try our system on other languages of different families, such as Semitic languages (Liebeskind and Liebeskind, 2020) and use LSC models to construct diachronic thesaurus, which bridges the lexical gap between modern and ancient language (Zohar et al., 2013; Liebeskind and Dagan, 2015; Liebeskind et al., 2016; Liebeskind et al., 2019).

References

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.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2015. Temporal random indexing: A system for analysing word meaning over time. *Italian Journal of Computational Linguistics*, 1(1):55–68.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

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.

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Alessio Ferrari, Beatrice Donati, and Stefania Gnesi. 2017. Detecting domain-specific ambiguities: an nlp approach based on wikipedia crawling and word embeddings. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 393–399. IEEE.

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

Gene H Golub and Charles Van Loan. 1996. *Matrix computations*. Johns Hopkins Univ.

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 Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 2116. NIH Public Access.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*.

Anna Hätty, Dominik Schlechtweg, and Sabine Schulte im Walde. 2019. Surel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 1–8.

Johannes Hellrich and Udo Hahn. 2016. Bad company—neighborhoods in neural embedding spaces considered harmful. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2785–2796.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515*.

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.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Chaya Liebeskind and Ido Dagan. 2015. Integrating query performance prediction in term scoring for diachronic thesaurus. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 89–94.

Chaya Liebeskind and Shmuel Liebeskind. 2020. Deep learning for period classification of historical hebrew texts. *Journal of Data Mining & Digital Humanities*, 2020, June.

Chaya Liebeskind, Ido Dagan, and Jonathan Schler. 2016. Semiautomatic construction of cross-period thesaurus. *J. Comput. Cult. Herit.*, 9(4), December.

Chaya Liebeskind, Ido Dagan, and Jonathan Schler. 2019. An algorithmic scheme for statistical thesaurus construction in a morphologically rich language. *Applied Artificial Intelligence*, 33(6):483–496.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

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.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte Im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic usage relatedness (durel): A framework for the annotation of lexical semantic change. *arXiv preprint arXiv:1804.06517*.

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.

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.

Claude E Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

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 (EMNLP-IJCNLP)*, pages 66–76, Hong Kong, China, November. Association for Computational Linguistics.

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

Hadas Zohar, Chaya Liebeskind, Jonathan Schler, and Ido Dagan. 2013. Automatic thesaurus construction for cross generation corpus. *J. Comput. Cult. Herit.*, 6(1), April.