A Comparative Study of Approaches for the Diachronic Analysis of the Italian Language

Pierluigi Cassotti, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro

Department of Computer Science, University of Bari Aldo Moro
Via E. Orabona, 4 - 70126 Bari (ITALY)
{firstname.surname}@uniba.it

Abstract. In recent years, there has been a significant increase in interest in lexical semantic change detection. Many are the existing approaches, data used, and evaluation strategies to detect semantic drift. Most of those approaches rely on diachronic word embeddings. Some of them are created as post-processing of static word embeddings, while others produce dynamic word embeddings where vectors share the same geometric space for all time slices. The large majority of the methods use English as the target language for the diachronic analysis, while other languages remain under-explored. In this work, we compare state-of-the-art approaches in computational historical linguistics to evaluate the pros and cons of each model, and we present the results of an in-depth analysis conducted using an Italian diachronic corpus. Specifically, several approaches based on both static embeddings and dynamic ones are implemented and evaluated by using the Kronos-It dataset. We train all word embeddings on the Italian Google n-gram corpus. The main result of the evaluation is that all approaches fail to significantly reduce the number of false-positive change points, which confirms that lexical semantic change is still a challenging task.

Keywords: Computational Historical Linguistics · Diachronic word embeddings · Lexical Semantic Change.

1 Background and Motivations

Diachronic Linguistics concerns the investigation of language change over time. Language change involves all levels of linguistic analysis: phonology, morphology, syntax and semantics [6, 5]. In this work, we focus on lexical semantic change. Two recent surveys [11, 19] describe and compare several lexical semantic change models that have been developed in the last years. Several datasets and tasks are employed in the evaluation of those models. In [13], the authors use two corpora of scientific papers and a corpus of senate speeches, both written in

English. They compare Static Bernoulli Embedding [14], Procrustes [10] and Dynamic Bernoulli Embeddings [13] using the held-out likelihood as evaluation metric. In [21], the authors evaluate, by using a rank-based approach, word2vec embeddings and a variant of Procrustes alignment to detect words that have undergone a semantic shift. Solving temporal word analogies is a common task used to evaluate models of lexical semantic change, which consists in detecting words analogies across time slices. In [7], the authors exploit the datasets created by [22] and [18] to compare Temporal Word Embeddings with Compass [7], LinearTrans-Word2vec [18], Procrustes [10], Dynamic Word Embeddings [22] and Geo-Word2vec [1]. However, few standard resources for evaluating lexical semantic change detection models are available. Currently, this gap is tackled by several initiatives. In [17], the authors introduce a framework (DUREL) for the annotation of lexical semantic change and at the same time they make available the annotated data[1]. DUREL is also employed in the annotation process of Semeval 2020 Task 1 [16] that involves four languages: English, Sweden, German and Latin, while the Italian language remains under-explored. Semeval 2020 Task 1 provides corpora in four languages and a gold standard of lexical semantic changes for the evaluation of unsupervised systems. However, the Semeval 2020 Task 1 corpora can only be used to evaluate lexical semantic change across two time periods. Therefore, it cannot be used to perform a more fine-grained analysis of the results. In this work, we describe a systematic evaluation of models for lexical semantic change detection with the Italian Google Ngram as the corpus for training word embeddings and Kronos-it [4] as the gold standard for the evaluation. Kronos-IT is a dataset for the evaluation of semantic change point detection algorithms for the Italian language automatically built by using a web scraping strategy. In particular, it exploits the information presents on the online dictionary "Sabatini Colletti"[2] to create a pool of words that have undergone a semantic change. In the dictionary, some lemmas are tagged with the year of the first attestation of its sense. In some cases, associated with the lemma there are multiple years attesting the introduction of new senses for that word. Kronos-IT uses this information to identify the set of semantic changing words.

Previous works about the Italian Google Ngram corpus and Kronos-it are described in [2, 4], but they are limited to the Temporal Random Indexing model [3] and simple baselines based on word frequencies and collocations ignoring recent approaches based on word embeddings.

The paper is structured as follows: Section 2 describes the approaches under analysis, while Section 3 reports details about the evaluation pipeline used in our work. Results of the evaluation are reported and discussed in Section 4.

2 Models

Traditional approaches produce word vectors that are not comparable across time due to the stochastic nature of low-dimensional reduction techniques or

[1] http://www.ims.uni-stuttgart.de/data/durel/

[2] https://dizionari.corriere.it/dizionario_italiano/

sampling techniques. To overcome this issue a widely adopted approach is to align the spaces produced for each time step, based on the assumption that only few words change their meaning. Words that turn out to be not aligned after the alignment, changed their semantics. In this work, we investigate two approaches for producing word embeddings that are comparable across time.

The first approach is based on the alignment of computed word embeddings (*bins*). Word vectors are computed before the alignment, once we get the bin (the embeddings matrix for a specific time slice), the different spaces obtained for each time slice are aligned. An example of this kind of approach is Procrustes [10], which aligns word embeddings with a rotation matrix. The assumption is that each word space has axes similar to the axes of the other word spaces, and two word spaces are different due to a rotation of the axes:

$$R = \arg\min_{Q^T Q = I} \left\| Q W^t - W^{t+1} \right\|_F$$

where W^t and W^{t+1} are two word spaces for time slices t and $t+1$, respectively, and Q is an orthogonal matrix that minimizes the Frobenius norm of the difference between W^t and W^{t+1}.

The second approach directly produces aligned word embeddings for each time slice, as it jointly learns word embeddings and aligns them. Dynamic word embeddings (DWE) [22] fall in this second type of approaches and it is based on the positive point-wise mutual information (PPMI) matrix factorization. In a unique optimization function, DWE produces embeddings and tries to align them according to the following equation:

$$\min_{U(t)} \frac{1}{2} \left\| Y(t) - U(t)U(t)^T \right\|_F^2 + \frac{\lambda}{2} \left\| U(t) \right\|_F^2 + \\ \frac{\tau}{2} \left(\left\| U(t-1) - U(t) \right\|_F^2 + \left\| U(t) - U(t+1) \right\|_F^2 \right)$$

where the terms are, respectively, the factorization of the PPMI matrix $Y(t)$, a regularization term and the alignment constraint that keeps the word embeddings similar to the previous and the next word embeddings.

The objective function of static Bernoulli embeddings is closely related to that of the CBOW (Continuous Bag of Words) [12] model, except that static Bernoulli embeddings regularize the embedding placing priors on both the embedding and context vectors. Dynamic Bernoulli Embeddings (DBE) [13] extends static Bernoulli embeddings including the time dimension. Context vectors are shared across all the time slices while embedding vectors are only shared within a time slice. Moreover, Dynamic Bernoulli Embedding uses a Gaussian random walk for obtaining smoothly changing estimates of each term embedding. The random walk penalizes the shifting of consecutive vectors.

Finally, we investigate Temporal Random Indexing (TRI) [3] that is able to produce aligned word embeddings in a single step. Unlike previous approaches, TRI is a count-based method. TRI is based on Random Indexing [15], where a word vector (word embedding) $sv_j^{T_k}$ for the word w_j at time T_k is the sum of random vectors r_i assigned to the co-occurring words taking into account only documents $d_l \in T_k$. Co-occurring words are defined as the set of m words that precede and follow the word w_j. Random vectors are vectors initialized randomly and shared across all time slices so that word spaces are comparable.

3 Methodology

Figure 1 shows the pipeline used for the evaluation, it consists of five modules: corpus pre-processing, computation of bins, bins alignment, construction of time-series and change point detection. The framework is written in Python, we adopt Procrustes[3], DBE[4], DWE[5] and TRI[6] using their original implementation.

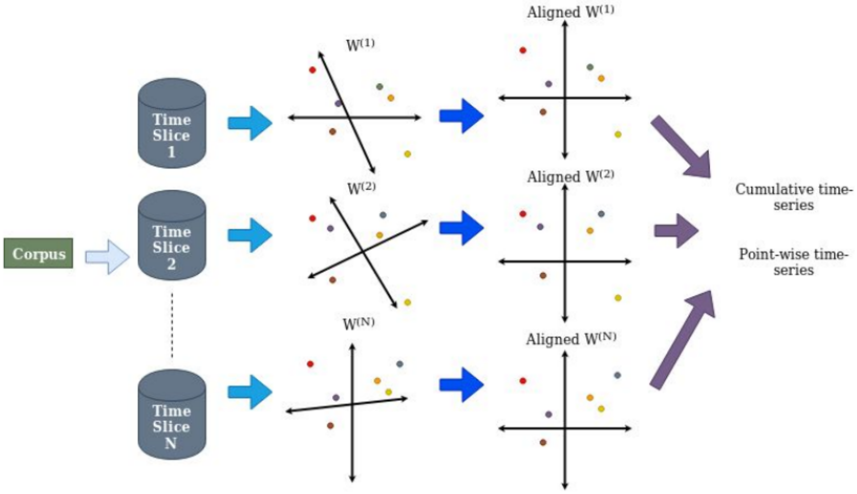

Fig. 1: The evaluation pipeline.

3.1 Corpus pre-processing

The corpus pre-processing module receives as input a corpus annotated with the time label of each document. The first operation is the corpus splitting into temporal slices. During the splitting, the dictionary is computing by keeping track of each new token encountered and its occurrence. The final dictionary is built with all tokens present in each time slice and selecting the first n tokens sorted by the number of occurrences. In our evaluation, we consider $n = 50,000$.

[3] `https://github.com/williamleif/histwords`
[4] `https://github.com/mariru/dynamic_bernoulli_embeddings`
[5] `https://github.com/yifan0sun/DynamicWord2Vec`
[6] `https://github.com/pippokill/tri`

3.2 Bins building

The second module takes as input tokenized documents for each time slice and generates for each approach preliminary information useful for the next steps. It has an execution mode for each approach namely Word2Vec, PPMI, Static Bernoulli and Temporal Random Indexing. Word2Vec mode trains a Word2Vec model on each sub-corpus using Gensim[7], an open-source library for unsupervised topic modelling and natural language processing. The PPMI mode constructs a PPMI matrix for each time slice, which will then be used to create Dynamic Word Embedding. The Bernoulli mode builds static Bernoulli embedding for each time slice that will later be used to construct Dynamic Bernoulli embeddings. The Temporal Random Indexing mode saves the occurrences of words and contexts that we will later be used to create word embeddings.

3.3 Alignment

The aim of the alignment module is the alignment of the bins produced as output in the previous module, and it is composed of several sub-modules: Procrustes Aligner, Bernoulli Aligner, Dynamic word embeddings construction and the TRI sub-module. The Bernoulli Aligner constructs Dynamic Bernoulli Embeddings starting from the static Bernoulli output. Procrustes Aligner is the sub-module that takes each Word2Vec model and applies Procrustes to each time slice. The Dynamic Word Embeddings sub-module takes the PPMI matrices previously created for building the Dynamic Word embeddings model. The TRI sub-module produces word vectors for each time slice by relying on the co-occurrences information built in the previous step.

3.4 Time-series and change point detection

We compute time-series by exploiting the word embeddings created for each time slice. A time-series for each word is built, this result in a matrix W^{VxT} where V is the dictionary size and T is the number of time slices.

We explore two approaches for the computation of the time-series, namely point-wise and cumulative. In the point-wise approach, the element i, j of W^{VxT} represent the cosine similarity

$$W_{i,j} = cos(v_{w_i}^{j-1}, v_{w_i}^{j})$$

where w_i is the i-th word in the dictionary and j is the j-th time slice. While, in the cumulative approach, the element i, j of W is

$$W_{i,j} = cos(\frac{\sum_{k=1}^{j} v_{w_i}^{k-1}}{j}, v_{w_i}^{j})$$

[7] `https://radimrehurek.com/gensim/`

In order to detect change points, we use the algorithm proposed in [20]. According to this model, we define a mean shift of a general time-series W_i pivoted at time period j as:

$$K(W_i) = \frac{1}{l-j} \sum_{k=j+1}^{l} W_{i,k} - \frac{1}{j} \sum_{k=1}^{j} W_{i,k} \tag{1}$$

To understand if a mean shift is statistically significant at time j we use a bootstrapping [8] approach under the null hypothesis. The null hypothesis states there is no change in the mean. We sample B bootstrap examples by permuting $W_{i,j}$. For each bootstrap sample P, $K(P)$ is calculated to provide its corresponding bootstrap statistic and statistical significance (p-value) of observing the mean shift at time j compared to the null distribution. Finally, we estimate the change point by considering the time point j with the minimum p-value score.

Change points together with the year, the p-value and the word are stored in a file used for the evaluation.

4 Evaluation

4.1 Data

For the training, we use the Google Ngram, a dataset of ngrams extracted by 305,763 Google Books. Google Ngram covers the period from 1500 to 2012. OCR errors can occur more in older historical documents, then we extract a sub-corpus concerning the period 1900-2010. We split Google Ngram corpus into ten slices with a range of ten years, starting from 1900 to 2010. We chose a time span of ten years for reducing the computational complexity since semantic changes are not frequent and generally require a large time span to be observed. Since the full text is not available in the Google Ngram, we use the method described in [9] for extracting co-occurrences between words. As gold standard, we use Kronos-it [4], a dataset for the Italian lexical change detection task. Kronos-it provides for each lemma a set of years indicating the semantic change for that lemma. Kronos-it is extracted by the Sabatini Coletti, an Italian dictionary that contains for some word meanings the year of the first appearance. The Kronos-it dataset contains 13,818 lemmas and 13,932 change points. Lemmas reported in Kronos-it have, on average, one change point.

4.2 Hyper-parameters

We use the same hyper-parameters values shared by two or more models. We use the same values for the *context-window* and the *dimension* of the embeddings. Table 1 reports training strategies and hyper-parameters values. We adopt default values used by the authors of the models.

In particular, in DWE we specify the number of *iterations* over the data, the alignment weight τ, the regularization weights λ and γ. In TRI, we set the *down*

DWE		TRI		DBE		Procrustes	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
dimension	300	dimension	300	dimension	300	dimension	300
window	4	window	4	window	4	window	4
iters	5	down-sampling	0.001	negatives	2	min-count	1
λ	10	seeds	10	minibatch	1000	negatives	20
γ	100			n epochs	4	sample	1e-5
τ	50					iter	4

Table 1: Models hyper-parameters.

sampling factor, and the number of *seeds*. In DBE, we set the number of *negative samples*, the *minibatch* size and the *number of epochs*. In Procrustes, we set the minimum number of occurrences a token must have to appear in the dictionary *min-count*, the number of *negative samples*, the downsampling parameter *sample* and the number of *iterations* over the data.

4.3 Metrics

We compute the performance of each approach by using Precision, Recall and F-measure. In the evaluation, a true positive is a change point for a word reported in the gold standard that belongs to the range of the ten years predicted by the system for that word. Change points provided by the systems are compared to the change points reported in the gold standard. The false negatives (FN) are the number of change points in the gold standard minus the true positives. The false positives (FP) are the number of change points provided by the system minus the true positives.

4.4 Results

Table 2 reports Precision (P), Recall (R) and F-measure (F) for each system. We can observe that generally, we obtain a low F-measure. This is due to the large numbers of change points detected by each system (false positive). We can observe that the best approach is DWE point-wise. However, the results of DWE point-wise are close to those obtained by Procrustes point-wise and TRI cumulative. A remarkable aspect is the worse performance of DBE respect those of TRI and DWE, the entries of DBE time-series are very close to 1, this highlights a heavy alignment. This is maybe due to the choice of hyper-parameters used to train the DBE. We use, as mentioned above, the default hyper-parameters and the type of datasets used by the authors is different from Google Ngrams, mainly due to the large amount of data in the Google Ngrams. This could have affected results obtained by DBE. The results of the evaluation prove that the task of semantic change detection is very challenging, in particular, the large number of detected change points (false positive) drastically affects the performance. Sometimes change points are detected before or after the change point reported

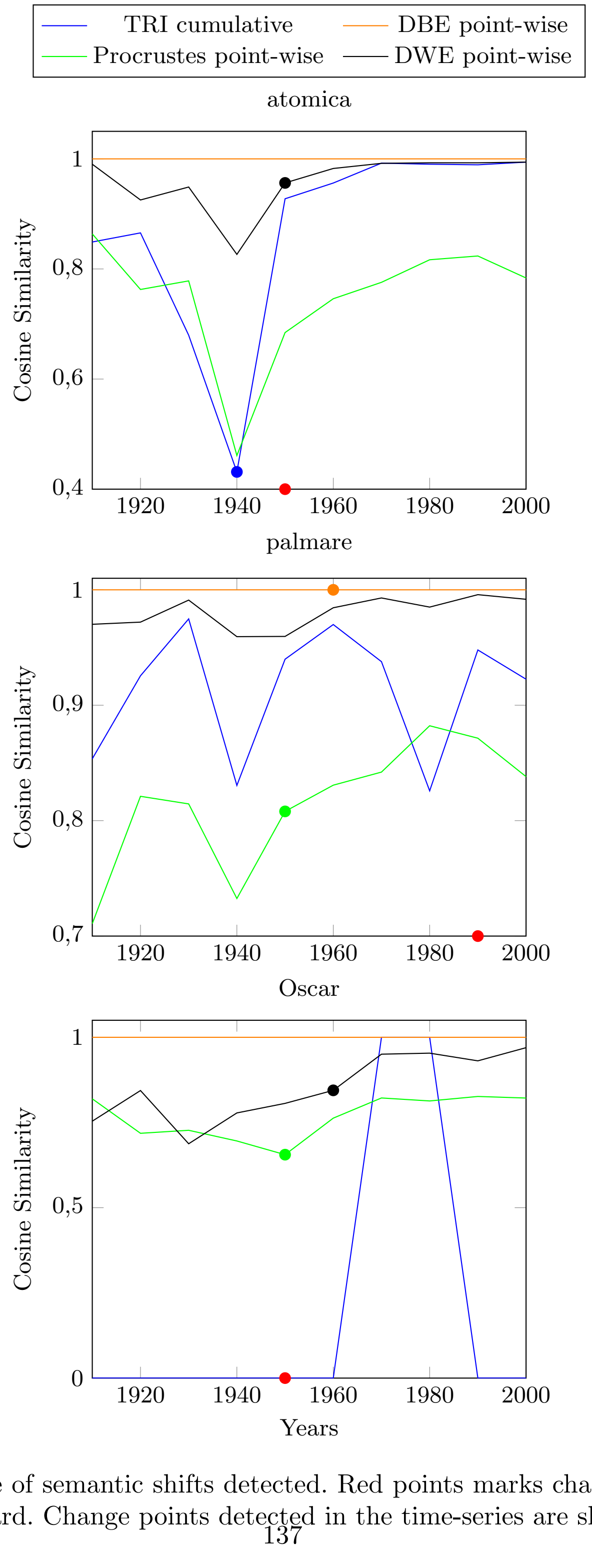

Fig. 2: Example of semantic shifts detected. Red points marks change points in the gold standard. Change points detected in the time-series are shown.

Model	Precision	Recall	F-Measure	Change points detected
DWE cumulative	.0016	.0840	.0031	13207
DWE point-wise	**.0020**	**.0880**	**.0039**	11115
TRI cumulative	.0017	.0680	.0033	10233
TRI point-wise	.0016	.0680	.0032	10315
DBE cumulative	.0000	.0000	.0000	255
DBE point-wise	.0019	.0200	.0035	2815
Procrustes cumulative	.0016	.0640	.0033	9652
Procrustes point-wise	.0019	.0200	.0036	2757

Table 2: Results of the evaluation.

in the gold standard, this supports the hypothesis that the change of semantics of a word is a continuous process, which involves long periods before reaching a stabilization. More studies are necessary to understand which component affects the performance, such an in-depth and explicit analysis of time-series. Moreover, it is important to underline that the year reported in the dictionary may be incorrect.

In Figure 2, we show some examples of time-series. For the word 'atomica', DWE cumulative is the only approach that fits the change point in the gold standard, indicating the change point as the decade 1950-1959, after 1945, year of Hiroshima and Nagasaki. We do not detect change points in the time-series produced by Procrustes point-wise and DBE point-wise, while we find a change point in the TRI-cumulative time-series in the 1950-1959 decade. For the word 'palmare', in the DBE point-wise and Procrustes cumulative time-series, two change points are detected that are too early compared to the change point in the gold standard 1998. Procrustes provided the right range 1950-1959 for the word 'Oscar', years in which for the first time an Italian film director, Vittorio De Sica, won the Oscar. TRI cumulative and DBE point-wise do not detect change points, while in the DWE point-wise time-series a change point is founded in the decade 1960-1969.

5 Conclusions

In this paper, we present a systematic evaluation of Dynamic Word Embeddings, Dynamic Bernoulli Embeddings, Procrustes and Temporal Random Indexing for the lexical semantic change detection for the Italian language. The results show that detect lexical semantic change is a complex task. A large number of change points is detected by systems, affecting the performance. A qualitative analysis of words time-series highlights that some change points are detected just before or after the correct period. This behaviour requires some further linguistic analysis for understanding the reasons behind.

This work can be extended in two directions: 1) including some recent models of lexical semantic change that involve contextual embeddings and a hyperparameter search optimized on the Italian Google Ngram dataset; 2) investigating other diachronic Italian corpora as training data. Moreover, we plan to investigate further methods for detecting changes in time-series.

Acknowledgments

This research has been partially funded by ADISU Puglia under the post-graduate programme ”Emotional city: a location-aware sentiment analysis platform for mining citizen opinions and monitoring the perception of quality of life”.

References

1. Bamman, D., Dyer, C., Smith, N.A.: Distributed representations of geographically situated language. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 828–834 (2014)
2. Basile, P., Caputo, A., Luisi, R., Semeraro, G.: Diachronic analysis of the Italian language exploiting google ngram. In: Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016). p. 56. CEUR.org (2016)
3. Basile, P., Caputo, A., Semeraro, G.: Analysing word meaning over time by exploiting temporal random indexing. In: First Italian Conference on Computational Linguistics CLiC-it (CLiC-it 2014). CEUR.org (2014)
4. Basile, P., Semeraro, G., Caputo, A.: Kronos-it: a Dataset for the Italian Semantic Change Detection Task. In: Proceedings of the 6th Italian Conference on Computational Linguistics (CLiC-it 2019). CEUR.org (2019)
5. Blank, A.: Why do new meanings occur? A cognitive typology of the motivations for lexical semantic change. Historical semantics and cognition (1999)
6. Bybee, J.L.: Diachronic linguistics. In: The Oxford handbook of cognitive linguistics. Oxford University Press (2010), https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199738632.001.0001/oxfordhb-9780199738632-e-36
7. Di Carlo, V., Bianchi, F., Palmonari, M.: Training temporal word embeddings with a compass. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6326–6334 (2019)
8. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. CRC Press (1994)
9. Ginter, F., Kanerva, J.: Fast Training of word 2 vec Representations Using N-gram Corpora (2014), https://www2.lingfil.uu.se/SLTC2014/abstracts/sltc2014_submission_27.pdf
10. Hamilton, W.L., Leskovec, J., Jurafsky, D.: 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). pp. 1489–1501 (2016)
11. Kutuzov, A., Øvrelid, L., Szymanski, T., Velldal, E.: Diachronic word embeddings and semantic shifts: a survey. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1384–1397 (2018)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013)
13. Rudolph, M., Blei, D.: Dynamic embeddings for language evolution. In: Proceedings of the 2018 World Wide Web Conference. pp. 1003–1011 (2018)

14. Rudolph, M., Ruiz, F., Mandt, S., Blei, D.: Exponential family embeddings. In: Advances in Neural Information Processing Systems. pp. 478–486 (2016)
15. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop at the 7th International conference on Terminology and Knowledge Engineering (2005)
16. Schlechtweg, D., McGillivray, B., Hengchen, S., Dubossarsky, H., Tahmasebi, N.: SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In: Proceedings of the 14th International Workshop on Semantic Evaluation. Association for Computational Linguistics (2020)
17. Schlechtweg, D., im Walde, S.S., Eckmann, S.: 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, Volume 2 (Short Papers). pp. 169–174 (2018)
18. Szymanski, T.: Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 448–453 (2017)
19. Tahmasebi, N., Borin, L., Jatowt, A.: Survey of computational approaches to lexical semantic change. arXiv preprint arXiv:1811.06278 (2018)
20. Taylor, W.A.: Change-point analysis: a powerful new tool for detecting changes
21. Tsakalidis, A., Bazzi, M., Cucuringu, M., Basile, P., McGillivray, B.: Mining the UK Web Archive for Semantic Change Detection. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). pp. 1212–1221 (2019)
22. Yao, Z., Sun, Y., Ding, W., Rao, N., Xiong, H.: Dynamic word embeddings for evolving semantic discovery. In: Proceedings of the eleventh ACM International Conference on Web Search and Data Mining. pp. 673–681 (2018)