

SynFlow: A Multidimensional Diachronic Semantic Analysis Toolkit

Bach Phan-Tat¹, Kris Heylen^{1,2}, Dirk Geeraerts¹, Stefano De Pascale^{1,3}, Dirk Speelman¹

¹Department of Linguistics, KU Leuven

²Instituut voor de Nederlandse Taal

³Vrije Universiteit Brussel

Correspondence: ttbach.phan@kuleuven.be

Abstract

Lexical semantic change (LSC) is commonly modelled through vector-space representations, but these approaches often provide limited insight into which aspects of usage are changing. Diachronic corpus research instead examines interpretable dimensions such as syntactic behaviour, morphology, and constructional patterns, but typically through separate analytical workflows. We present **SynFlow**, an open-source toolkit for multidimensional diachronic analysis of linguistic usage. SynFlow represents linguistically enriched word concordances as period-specific distributions and applies a shared workflow across dependency-based co-occurrences, morphological features, constructional configurations, and semantic patterns such as those from Frame Semantics. It supports different distance measures, together with value-level contribution scores, statistical testing, and incremental clustering of lexical slot-fillers. We demonstrate SynFlow through a qualitative case study of the German adjective *viral*, showing how a single semantic development is reflected across syntactic, lexical, constructional, and morphological dimensions. We further report previously published results on SemEval-2020 Task 1 to situate the performance of these representations relative to existing lexical semantic change detection systems.

We introduce SynFlow, an open-source toolkit for multidimensional diachronic semantic analysis. SynFlow represents each dimension as a period-specific distribution of observable values and quantifies semantic change as differences between the diachronic distributions within a dimension. It natively supports automatic extraction and quantification of dependency-based co-occurrences, morphological features, and constructional configurations, while other semantic patterns such as Frame Semantics can be incorporated through the same interface. This allows different dimensions to be analysed within a common diachronic workflow rather than requiring a separate analytical procedure for each representation.

SynFlow compares these distributions across periods using cosine distance, Jensen–Shannon divergence (JSD) (Menéndez et al., 1997), or total variation distance (TVD), and decomposes change scores into contributions from individual values, making it possible to determine **which dimensions change** and **what drives the change**. For lexical fillers, it additionally provides incremental clustering to trace broader thematic developments. The system therefore preserves a direct connection between quantitative change signals and the linguistic evidence underlying them.

The main contribution of SynFlow is a **shared workflow for multidimensional diachronic semantic analysis**, providing temporal comparison, statistical testing, and value-level contribution scores.

We demonstrate ¹ SynFlow with a qualitative case study of the German adjective *viral*, showing how one semantic development is reflected across multiple linguistic dimensions. We also report previously published SemEval-2020 Task 1 results (Tat et al., 2026b,a) to situate the performance of these representations relative to existing LSC sys-

1 Introduction

Recent computational work on lexical semantic change (LSC) is dominated by vector-space modelling approaches that compare vector representations of target words across historical periods (Periti and Montanelli, 2024; Tahmasebi et al., 2021; Tahmasebi and Dubossarsky, 2023). While effective on shared tasks, such methods are difficult to interpret (Lenci et al., 2022), making their outputs difficult to relate directly to linguistic theories of semantic change: a model may indicate that a word has changed without directly revealing which aspects of its usage account for that change.

¹<https://www.youtube.com/watch?v=CR1G2kgdFTE>

tems.

2 Related work

2.1 Linguistic dimensions in diachronic analysis

Theory-driven studies of semantic change have long examined linguistically motivated dimensions such as prototype structure, metaphor, frame semantics and constructions (e.g., Geeraerts, 1993; Jansegers and Gries, 2020; Law, 2019; Pettersson-Traba, 2016; Sovran, 2004; Sullivan, 2007; Sweetser, 1990; Vais, 2022). While these approaches provide detailed accounts of semantic development, their analytical dimensions are often tailored to particular lexical items or change phenomena, limiting systematic comparison across targets and corpora. Computational work has begun to operationalise structured linguistic information at a larger scale (e.g., Kutuzov et al., 2021; Ryzhova et al., 2021). SynFlow extends this direction by providing a reusable workflow for comparing the temporal distributions of such structured usage dimensions within a common framework.

2.2 Computational tools for linguistic profiling

SynFlow is related to corpus tools that profile lexical behaviour, including Sketch Engine’s Word Sketch Difference (WSD) (Word-Sketch-Difference, 2019), Korp (Borin et al., 2025), and DiaCollo (Jurish, 2015). DiaCollo models diachronic change through linear-window collocations which often suffer from accidental co-occurrences (Evert, 2008). WSD and Korp use association measures that rank individual collocates whereas SynFlow quantifies redistribution of **complete profiles**, and decomposes change into individual contributions. SynFlow also processes multiple temporal periods within **a single pass** rather than requiring repeated pairwise comparisons like WSD.

There are other tools operating on structured linguistic information in parsed corpora. ANNIS (Chiarcos and Lüdeling, 2009), Grew-match (Guillaume, 2021), and the FrameNet Brasil WebTool (Torrent et al., 2025) support querying, visualisation, or annotation; STARK (Krsnik and Dobrovoljc, 2025) extracts configurable dependency patterns; Profiling-UD (Brunato et al., 2020) derives multi-level linguistic profiles; ComparaTree (Terčon and Dobrovoljc, 2025) compares linguis-

tic properties across treebanks; and DELTA (Estève and Dobrovoljc, 2026) combines dependency-pattern extraction with multi-level diversity measurement across corpora. However, these systems primarily support querying, extraction, profiling, annotation, or corpus-level comparison rather than **target-oriented diachronic semantic analysis**. SynFlow complements them by utilising structured linguistic information to model diachronic semantics. It models linguistic observations as target-conditioned temporal distributions and analyses their redistribution through statistical testing and value-level contribution scores.

3 System architecture

SynFlow takes annotated diachronic subcorpora as input, with each directory representing one temporal period. Given a target lemma, POS, and selected dimension types, it extracts the corresponding linguistic patterns, converts them into period-specific distributions, measures change across periods, and decomposes the resulting scores into contributions from individual values. The overall workflow is demonstrated in figure 1.

SynFlow distinguishes between a *dimension type*, which specifies the linguistic information to extract, and an individual *dimension*, whose observed values form a distribution. For target w , dimension d , and period t , usage is represented as $P_t(v \mid d, w)$ over values v . Some dimension types such as construction define a single distribution, while others define multiple dimensions, such as those conditioned on a specific dependency slot or morphological feature type.

SynFlow natively extracts dependency-based, constructional, and morphological information, while externally annotated dimensions can be analysed through the same downstream workflow provided their dataframe format matches the output of the Dimension Extraction step. These dimensions provide complementary views of linguistic usage. Slot and feature types describe broader structural profiles, slot fillers and morphological features capture variation within particular categories, and constructional configurations capture how multiple syntactic relations combine within the same usage instance. Dedicated notebook templates are provided for the supported analysis types.

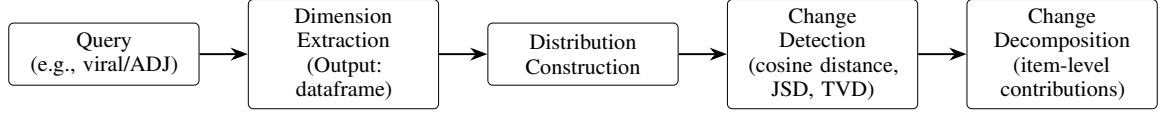


Figure 1: Overview of the workflow.

Type	Dimension	Example values	Interpretation
Slot type	SlotType	chi_nsubj, chi_obj	grammatical profile
Slot filler	FILLER[chi_obj]	cake	lexical preferences
Construction	Construction	[chi_nsubj + chi_obj]	multi-slot configurations
Feature type	FeatureType	Number, Tense	morphological profile
Feature	FEAT[Number]	Sing, Plur	variation within feature

Table 1: Dimensions and example values natively extracted by SynFlow.

3.1 Dimension extraction

Input format SynFlow uses a CoNLL-U-like format in which each sub-corpus is divided into individual sentences. Each sentence begins with `<s=sentence_id>` and ends with `</s>`. Seven tab-separated columns are required in the following order: TOKEN, LEMMA, POS, ID, HEAD_ID, DEPREL, FEATS. We have also provided a notebook and scripts for parsing the raw data (1 sentence/line) into the correct SynFlow format using Stanza (Qi et al., 2020).

Dimensions SynFlow natively extracts five types of linguistic information (Table 1) using different tree-traversal algorithms. **Slot types** are individual dependency relations involving the target, with direction encoded as `chi_` for children and `pa_` for parents. For example, in *the boy eats the cake*, *eats* is associated with `chi_nsubj` and `chi_obj` as it is the head of *boy* and *cake*, while *boy* is associated with `pa_nsubj` and `chi_det`. **Slot fillers** are the lexical items occupying a dependency slot, with each slot defining a separate dimension. **Constructional configurations** represent combinations of dependency slots co-occurring around the same target instance, e.g. `[chi_nsubj + chi_obj]`, within a user-specified dependency-path depth. **Morphological feature types** are categories extracted from the target’s FEATS field, while **morphological features** represent the observed values of each category as separate dimensions.

3.2 Change detection

Counts within each dimension and period are normalised to sum to one, yielding probability distributions. This differs from occurrence-normalised

frequencies used in some descriptive visualisations, where counts are divided by the number of target occurrences. SynFlow then measures distribution change between consecutive periods using cosine distance, JSD, or TVD. Because distances estimated from sparse observations can be unstable, SynFlow optionally applies support weighting:

$$WeightedDist = RawDist \times \min\left(1, \frac{c}{k}\right),$$

where $c = \min(c_1, c_2)$ is the smaller number of observations in the two periods, and k is a user-defined support threshold. Distances receive full weight (i.e., 1) when both periods contain at least k observations and are linearly down-weighted otherwise.

SynFlow also supports permutation testing to help distinguish meaningful changes from random fluctuations. Alongside the magnitude of the detected change, it provides a p -value indicating how likely a change of similar or greater size would be observed by chance. This gives users an interpretable measure of whether a detected change is statistically significant.

For multi-period analyses, SynFlow applies the Benjamini–Yekutieli (BY) procedure (Benjamini and Yekutieli, 2001) to control the False Discovery Rate (FDR). For each dimension, correction is applied jointly to the p -values from all adjacent-period comparisons.

3.2.1 Incremental slot filler clustering

This section only applies to slot fillers. To distinguish lexical replacement from broader thematic change, SynFlow provides incremental clustering of slot fillers. The analysis uses static word2vec embeddings (Mikolov et al., 2013) trained separately

for each period and Procrustes aligned (Smith et al., 2017) to a common space, following diachronic embedding approaches (Hamilton et al., 2016). Fillers in the first period are grouped using hierarchical agglomerative clustering, with each cluster represented by the frequency-weighted centroid of its members. In subsequent periods, fillers are assigned to sufficiently similar existing centroids, while unmatched fillers form new clusters. Active centroids are then updated from the current-period fillers. Previously established clusters are retained when temporarily inactive, allowing them to be reactivatable in later periods. This incremental memory enables SynFlow to trace the emergence, persistence, decline, and reactivation of broader filler themes over time. A dedicated notebook is provided for training and aligning the required word2vec embeddings.

3.3 Change decomposition

After computing the distance between two period-specific distributions, SynFlow can further decompose the observed change into contributions from individual values. This allows users to move from a single aggregate change score to the specific linguistic items responsible for that difference. For a dimension d , SynFlow reports how much each value v contributes to the overall distance between two consecutive periods. For example, if the distribution of FILLER[chi_amod] changes substantially between two periods, decomposition can reveal which adjectival fillers account for most of that shift. This step is particularly useful for qualitative analysis, because it links the numerical change score back to interpretable linguistic evidence. Rather than only indicating that a dimension has changed, SynFlow shows which individual values drive the observed redistribution.

3.4 Output and visualisation

Most intermediate and final outputs produced by SynFlow are stored as comma-separated values (CSV) files. This allows users to inspect the extracted information directly, verify intermediate processing steps, and reuse the outputs in external analyses. The accompanying notebooks also provide visualisations at different stages of the workflow to facilitate exploration and interpretation. These include time-series plots for tracking changes across periods, heatmaps for comparing patterns across dimensions or time, and bar plots for inspecting distributions and item-level contri-

butions. Together, the tabular outputs and visualisations allow users to move between quantitative summaries and the linguistic evidence underlying them.

3.5 Availability and execution

SynFlow is freely available as open-source software under the MIT license. The repository² provides installation instructions, dependency specifications, documentation of the expected corpus format, and ready-to-use notebook templates for the main analysis types. SynFlow is currently distributed as a notebook-based Python workflow. Users install the system by cloning the repository, creating a Python environment, installing the packages specified in requirements.txt, and running the notebooks. Python 3.10 or newer is recommended.

There are dedicated notebooks³ for slot-level, constructional, morphological-feature, diachronic embedding and qualitative analysis. Within each notebook, users specify the corpus location, target lemma and POS, temporal periods, and relevant analysis parameters. Input corpora are organised into directories corresponding to temporal periods and follow the seven-column CoNLL-U-like format described in 3.1. The notebooks then guide users through the complete workflow from dimension extraction and distribution construction to change measurement, decomposition, statistical analysis, and visualisation. This design allows the different linguistic dimensions to be analysed through a common workflow without requiring users to modify SynFlow’s underlying extraction and analysis functions.

3.6 Runtime and scalability

SynFlow supports multiprocessing at every possible step. As a practical benchmark, we ran SynFlow with a representative target (plane/NOUN) from Corpus of Historical American English (Davies, 2010) (1810s - 2000s, divided into 20 periods and containing 116,613 parsed files), on a personal laptop (32 GB RAM, AMD Ryzen 7 8845HS). Direct slot-level exploration, which reads the parsed corpus, counts dependency slots within each period, and produces the corresponding visual-

²<https://github.com/phantaatbach/SynFlow>

³Because we cannot upload the whole dataset, we have provided dataframes of different dimensions so that users can explore Change Detection and Change Decomposition.

isations, took approximately 7 minutes 30 seconds. Extracting all slot fillers and constructing the slot-filler dataframe took 4 minutes 55 seconds. For each adjacent-period transition, we ran 1,000 permutations for each pair of slot-filler distributions of every extracted slot. Processing all slot-specific permutation tests for one transition required only 3 minutes. These results indicate that SynFlow remains practical for large diachronic corpora on consumer hardware.

4 Evaluation

4.1 Demonstrative case study

We demonstrate SynFlow with the German adjective *viral* in the Leipzig German News corpus (Leipzig, 2026), covering 1995–2025 and divided into six five-year periods. Over this period, *viral* shifts from predominantly virus-related uses, as in *virale Infektion* ‘viral infection’, towards the newer sense ‘widely circulated or popular’, especially in online and social-media contexts, as in *viral gehen* ‘go viral’.

In the slot type dimension, occurrence-normalised frequencies of different slots (Figure 2) remain largely stable except for a marked shift around 2011–2015: *pa_amod* decreases while *pa_advmod* increases, reflecting the growing use of *viral* as an adverbial modifier in expressions such as *viral gehen* and a relative decrease of the use as an adjectival modifier like in *virale Infektion*.

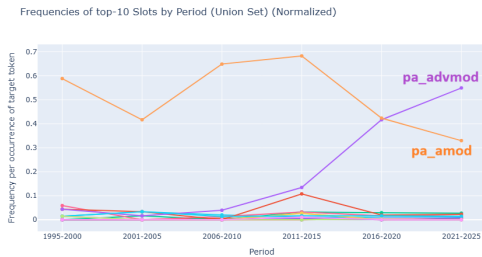


Figure 2: Change of normalised frequencies of all slot types of *viral*.

This redistribution produces a large TVD between the two periods, and decomposition shows that the redistribution of *pa_amod* and *pa_advmod* accounts for much of the overall TVD (Figure 3).

The slot-filler dimensions of *pa_amod* and *pa_advmod* show the largest weighted TVD increases around 2011–2015 (Figure 4), while most other slots remain comparatively stable. Further decomposition and clustering analysis (Figure 5)

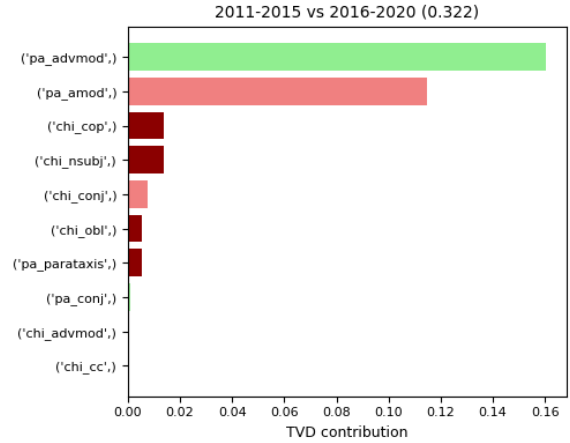


Figure 3: Contributions of individual slot types to TVD of *viral*. Green bars indicate an increase in relative frequency; red bars indicate a decrease.

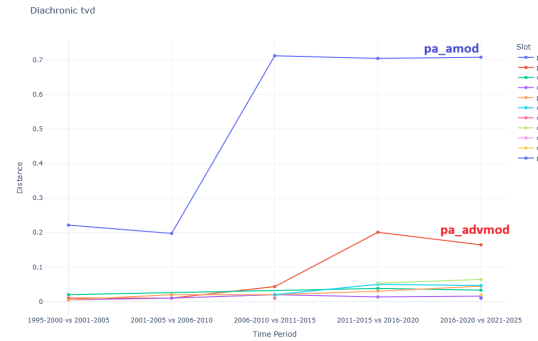


Figure 4: Weighted TVD trajectories for slot-filler distributions across dependency slots.

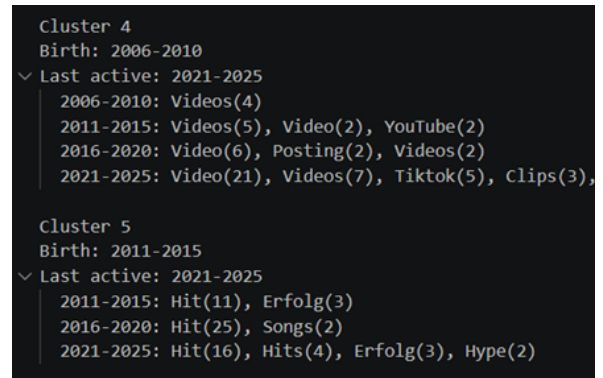


Figure 5: Incremental clustering results with social-media-related clusters growing rapidly from 2011.

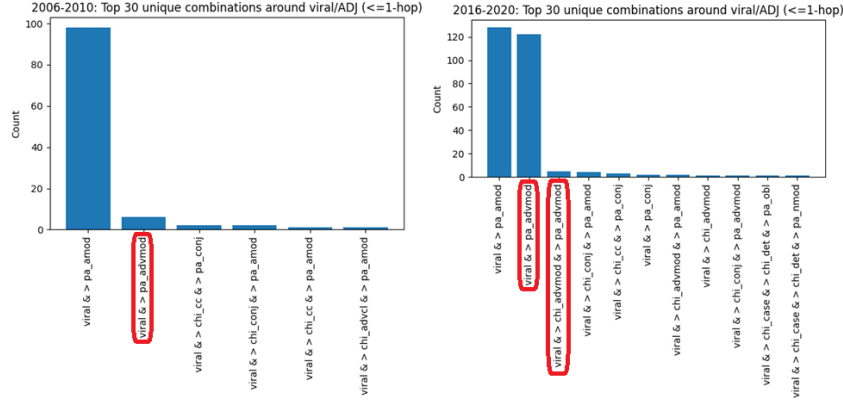


Figure 6: Frequencies of different constructions around *viral*

reveal that these changes are increasingly associated with fillers from online and social-media discourse, including *Hit*, *Marketing*, *Video*, *TikTok* in *pa_amod*, and *gehen* in *pa_advmod*. Slot-filler analysis therefore complements the structural signal by showing the lexical contexts accompanying the newer meaning of *viral*.

The same development is visible at the constructional level, showing that the rise of the adverbial *viral* is not limited to an isolated dependency relation, but also affects the larger combinations of grammatical relations in which the target occurs. From around the same period, configurations containing *pa_advmod* become increasingly frequent (Figure 6). For example, the configuration [*chi_advmod* + *pa_advmod*] occurs in expressions such as *gerade viral gehen* ‘to be going viral right now’, where *viral* is linked to *gehen* through *pa_advmod* and to *gerade* through *chi_advmod*.

At the morphological level, the frequency of Variant increases around the same period as *pa_advmod*. These adverbial uses are annotated as Variant=Short as they lack the normal adjectival inflection because they are used as an English loan. However, FEAT[Variant] contains only the value Short, meaning that the change is visible at the feature-type level, but not in the corresponding feature-value dimension. This contrasts with the dependency dimensions, where change is visible both at the broader structural level and within individual slot-filler distributions.

Overall, the case of *viral* illustrates how one semantic development can leave complementary traces across several linguistic dimensions. Rather than collapsing these signals into a single score, SynFlow allows users to identify where change occurs and inspect the dimension-specific evidence.

Representation	Subtask 1	Subtask 2
Slot filler	4th /28	12th /28
Frame Semantics	9th /28	9th /28

Table 2: Ranking of previously evaluated SynFlow representations against systems reported for the English SemEval-2020 benchmark. Full results are reported in Tat et al. (2026b,a).

4.2 Experimental benchmark

We previously benchmarked SynFlow on SemEval-2020 Task 1 (Schlechtweg et al., 2020) using two linguistic representations: dependency-based slot fillers and Frame Semantics (Tat et al., 2026b,a). Both representations proved effective for lexical semantic change detection and ranked above many neural-network-based systems on the benchmark (Table 2), while retaining direct interpretability of the linguistic evidence underlying the change scores.

5 Conclusion and future work

We presented SynFlow, an open-source toolkit for multidimensional diachronic semantic analysis. It natively supports dependency-based, constructional, and morphological information while allowing other semantic patterns to be analysed through the same interface. The *viral* case study illustrates how a single semantic development can be traced across multiple dimensions, while previous benchmark results show competitive performance against embedding-based methods. Future work will extend SynFlow to additional linguistic dimensions and integrate further parsers and LLM-based annotation methods into the extraction pipeline.

References

- Yoav Benjamini and Daniel Yekutieli. 2001. [The control of the false discovery rate in multiple testing under dependency](#). *The Annals of Statistics*, 29(4).
- Lars Borin, Markus Forsberg, Martin Hammarstedt, Louise Holmer, and Arild Matsson. 2025. [Korp: Språkbanken’s word research platform](#). In Dana Danneélls, Kristian Blensénus, and Lars Borin, editors, *Sixty years of Swedish computational lexicography*, volume 3 of *Digital Linguistics*, pages 175–193.
- Dominique Brunato, Andrea Cimino, Felice Dell’Orletta, Giulia Venturi, and Simonetta Montemagni. 2020. [Profiling-UD: a tool for linguistic profiling of texts](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7145–7151, Marseille, France. European Language Resources Association.
- Christian Chiarcos and Anke Lüdeling. 2009. ANNIS: A Search Tool for Multi-Layer Annotated Corpora. In *Proceedings of the Corpus Linguistics Conference 2009 (CL2009)*, volume 358.
- Mark Davies. 2010. [The corpus of historical american english \(coha\)](#).
- Louis Estève and Kaja Dobrovoljc. 2026. [DELTA: A toolkit for measuring linguistic diversity in dependency-parsed corpora](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 75–85, Rabat, Morocco. Association for Computational Linguistics.
- Stefan Evert. 2008. Corpora and collocations. In *Corpus Linguistics. An International Handbook*.
- Dirk Geeraerts. 1993. [Vagueness’s puzzles, polysemy’s vagaries](#). *Cognitive Linguistics*, 4(3):223–272.
- Bruno Guillaume. 2021. [Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.
- 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 (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.
- Marlies Jansegers and Stefan Th. Gries. 2020. [Towards a dynamic behavioral profile: A diachronic study of polysemous sentir in Spanish](#). *Corpus Linguistics and Linguistic Theory*, 16(1):145–187.
- Bryan Jurish. 2015. DiaCollo: On the trail of diachronic collocations.
- Luka Krsnik and Kaja Dobrovoljc. 2025. [STARK: A toolkit for dependency \(sub\)tree extraction and analysis](#). In *Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025)*, pages 44–51, Ljubljana, Slovenia. Association for Computational Linguistics.
- Andrey Kutuzov, Lidia Pivovarova, and Mario Giulianelli. 2021. [Grammatical Profiling for Semantic Change Detection](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 423–434, Online. Association for Computational Linguistics.
- James Law. 2019. [Diachronic frame analysis](#). *Constructions and Frames*, 11(1):43–78. Type: Journal Article.
- Wortschatz Leipzig. 2026. [Downloads – German – Wortschatz Leipzig](#).
- Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllénsten, and Martina Miliani. 2022. [A comparative evaluation and analysis of three generations of Distributional Semantic Models](#). *Language Resources and Evaluation*, 56(4):1269–1313.
- M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. 1997. [The Jensen-Shannon divergence](#). *Journal of the Franklin Institute*, 334(2):307–318.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). *Preprint*, arXiv:1301.3781.
- Francesco Periti and Stefano Montanelli. 2024. [Lexical Semantic Change through Large Language Models: a Survey](#). *ACM Computing Surveys*, 56(11):1–38.
- Daniela Pettersson-Traba. 2016. [Analyzing the behavioral profiles of sets of near-synonyms in American English from a diachronic perspective](#).
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Anastasiia Ryzhova, Daria Ryzhova, and Ilya Sochenkov. 2021. [Detection of Semantic Changes in Russian Nouns with Distributional Models and Grammatical Features](#).
- 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 Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

- Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word embeddings. In *International Conference on Learning Representations (ICLR)*.
- Tamar Sovran. 2004. [Polysemy and Semantic Frames - A diachronic study of "Truth" in Hebrew](#). 26(1):89–100.
- Karen Sullivan. 2007. Metaphoric Extension and Invited Inferencing in Semantic Change. *Culture, Language and Representation*, Special Issue: Metaphor and Discourse:257–274.
- Eve Sweetser. 1990. *From Etymology to Pragmatics: Metaphorical and Cultural Aspects of Semantic Structure*. Cambridge Studies in Linguistics. Cambridge University Press, Cambridge.
- Nina Tahmasebi, Lars Borin, Adam Jatowt, Yang Xu, and Simon Hengchen. 2021. Computational approaches to semantic change.
- Nina Tahmasebi and Haim Dubossarsky. 2023. [Computational modeling of semantic change](#). *eprint*:2304.06337.
- Bach Phan Tat, Kris Heylen, Stefano De Pascale, Dirk Geeraerts, and Dirk Speelman. 2026a. [ReFRAME or remain: Unsupervised lexical semantic change detection with frame semantics](#). In *Proceedings of the 15th Joint Conference on Lexical and Computational Semantics (*SEM 2026)*, pages 83–97, San Diego, California, United States. Association for Computational Linguistics.
- Bach Phan Tat, Kris Heylen, Dirk Geeraerts, Stefano De Pascale, and Dirk Speelman. 2026b. [Transparent semantic change detection with dependency-based profiles](#). In *The Proceedings for the 6th International Workshop on Computational Approaches to Language Change (LChange'26)*, pages 97–109, Rabat, Morocco. Association for Computational Linguistics.
- Luka Terčon and Kaja Dobrovoljc. 2025. [ComparaTree: A multi-level comparative treebank analysis tool](#). In *Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025)*, pages 129–139, Ljubljana, Slovenia. Association for Computational Linguistics.
- Tiago Timponi Torrent, Ely Edison Da Silva Matos, Alexandre Diniz Da Costa, Maucha Andrade Gamonal, Simone Peron-Corrêa, and Vanessa Maria Ramos Lopes Paiva. 2025. [A flexible tool for a qualia-enriched FrameNet: the FrameNet Brasil WebTool](#). *Language Resources and Evaluation*, 59(4):4529–4557.
- Christine Marie Vais. 2022. *A Diachronic Application of Frame Semantics to Meaning Change in Prepositions: The Case of German mit*. Doctoral dissertation, University of California, Berkeley, Berkeley, CA.
- Word-Sketch-Difference. 2019. [Word sketch difference](#).