Clustering and Cleaning of Word Usage Graphs

Anonymous ACL submission

Abstract

Word Usage Graphs (WUGs) represent human judgments about semantic proximity between word uses as a weighted undirected graph. WUGs pose specific challenges to clustering algorithms such as incompleteness and noise. We are the first to systematically compare multiple graph clustering algorithms for WUGs and find that the Weighted Stochastic Block Model is comparable to or outperforms the current state-of-the-art. We further test various graph cleaning strategies to improve the quality of remaining cluster assignments while minimizing data loss. With better clustering and cleaning methods we hope to help researchers help other researchers improve the quality of their WUGs without additional manual annotation. We publish clustered and cleaned graphs for further research.

1 Introduction

In recent years, a new annotation paradigm for word senses has emerged under the name of **Word Usage Graphs** (WUGs, Schlechtweg et al., 2020, 2021b). In this paradigm, **humans** provide semantic proximity judgements for pairs of word uses (also known as *Word-in-Context* annotations), which are then represented as a weighted graph and clustered with a graph clustering algorithm, as displayed in Figure 1. This way, clusters representing word senses can be inferred from simple judgments about pairs of word uses, avoiding the need for word sense definitions. While, up to now, this approach has been applied mainly within the field of Lexical Semantic Change Detection (LSCD) (e.g. Schlechtweg et al., 2021b; Kurtyigit et al., 2021; Zamora-Reina et al., 2022), it can be applied generally in a Word Sense Induction (WSI) setting (Aksenova et al., 2022) or for Word Sense Disambiguation (WSD) when combined with a sense labelling procedure for word sense clusters (cf. Giulianelli et al., 2023; Kutuzov et al., 2024).

Being a rather recently developed annotation approach, the WUG paradigm brings many open questions. In this paper, we approach two important problems: (i) WUGs are undirected graphs with ordinal edge weights. They are often sparsely observed (annotated), contain considerable annotation noise and disagreements and have subgraphs annotated by different annotators. Node **clustering** under these conditions is challenging as e.g. many standard clustering algorithms such as Agglomerative Clustering (Ward Jr, 1963) need a complete adjacency matrix not provided by incomplete WUGs. The current state-of-the-art approach is Correlation Clustering (Bansal et al., 2004) as first applied to this problem by Schlechtweg et al. (2020), but mainly for lack of a systematic comparison. We thus test multiple graph clustering algorithms on a WUG dataset that provides an independent word sense annotation for evaluation. (ii) As a result of the above-described challenges, clusterings obtained on WUGs often show considerable error. Researchers may want to clean out unreliable cluster assignments before using them as ground truth for model evaluation (e.g. Schlechtweg et al., 2020; Aksenova et al., 2022; Zamora-Reina et al., 2022) or further modelling (Giulianelli et al., 2023; Kutuzov et al., 2024). Hence, we test several post hoc **cleaning** strategies and evaluate the results in terms of lost data and correspondence to the independent word sense annotation.

Our contributions can be summarized as follows:

- We are the first to systematically evaluate graph clustering algorithms on manually annotated WUGs.
- We considerably improve the clustering performance over the previous state-of-the-art.
- We are the first to empirically validate the clustering model proposed in Peixoto (2017) and slightly adjusted in Schlechtweg et al. (2021a).

	DWUG DE	DWUG DE Sense
n	50	24
N/V/A	34/14/2	16/7/1
|U|	≤100+≤100	25+25
AN	8	3
|J|	1.7	2.9
KRI	.67	.87
STYLE	use-use	use-sense

Table 1: Statistics for the latest version (2.3.0) of DWUG DE and the new DWUG DE Sense dataset. n = no. of target words, N/V/A = no. of nouns/verbs/adjectives, |U| = no. uses per word (t_1+t_2), AN = no. of annotators, |J| = avg. no. judgments per annotation instance, KRI = Krippendorff's α, STYLE = annotation style.

- We are the first to formulate and systematically evaluate cleaning procedures for WUGs, which will be crucial for further research building on this type of data.
- We publish improved clusterings and cleaned graphs for further research.[1]

2 Related Work

There are a number of recent WUG datasets for multiple languages (Schlechtweg et al., 2021b; Kurtyigit et al., 2021; Baldissin et al., 2022; Zamora-Reina et al., 2022; Kutuzov et al., 2022; Aksenova et al., 2022; Chen et al., 2023). Most of them are diachronic, meaning that the underlying word uses were sampled from different time periods. A few studies investigate the clustering and/or edge sampling procedures (Schlechtweg et al., 2021a; Tunc, 2021; Kotchourko, 2021). See also Schlechtweg (2023, pp. 54–67) for an in-depth analysis of cluster errors, and robustness of clusterings and semantic change scores derived from them. However, studies on clustering WUGs either do not evaluate the quality of obtained clusters against a realistic (i.e., empirically observed) gold standard or do not compare their models against others. Hence, we currently do not know which clustering algorithm should be preferred on WUGs in practice. Moreover, there are no previous results on cleaning WUGs. More recent work uses WUG clusters as a data source for generation of sense glosses (Giulianelli et al., 2023; Kutuzov et al., 2024) or for edge induction (Noble et al., 2024).

[1] Link will be put for publication version.

3 Datasets

For our experiments, we use the DWUG DE dataset (Schlechtweg et al., 2021b) and the DWUG DE Sense dataset (Schlechtweg, 2023, pp. 57–58) derived from it. Table 1 provides basic statistics for both of them.

3.1 DWUG DE

DWUG DE contains pairs of German word uses from two time periods annotated with judgements about relatedness of word meanings in those pairs collected from multiple annotators. For each target word, authors sampled pairs of uses such as (1) and (2) from two historical corpora (1800–1899, 1946–1990) and asked annotators to rate them on a ordinal relatedness scale from 1 (unrelated) to 4 (identical), as detailed in Table 3 in Appendix A.

(1) Im **Ohrwurm** ist der obere Magenmund inwendig mit einigen Zähnen in zwey Reihen besetzt.
*'In the **earworm** the upper stomach mouth is occupied inside with some teeth in two rows.'*

(2) Werden die Lieder **Ohrwürmer**, klingelt auch die Kinokasse.
*'If the songs become **catchy tunes**, the cinema cash register also rings.'*

The annotated pairs were represented as a weighted graph with the median of annotator judgments as edge weights and clustered with Correlation Clustering. All uses sharing a cluster were then interpreted as having the same sense and the semantic change for each word was measured based on these clusterings. We use version 2.3.0 for our experiments.[2]

3.1.1 DWUG DE Sense

Schlechtweg (2023) randomly chose 24 target words (out of 50) from the DWUG DE dataset, randomly sampled uses for each target word (25 per time period from at most 100 in the original dataset) and asked three annotators to label each use with a sense definition from a predefined inventory best describing meaning of the target word in this use. The data is then cleaned and aggregated. For our experiments, we use the "maj3" aggregation, meaning that all uses were identically annotated by all three annotators leaving 826 uses of 24 target words for evaluation. We use version

[2] https://zenodo.org/records/7441645.

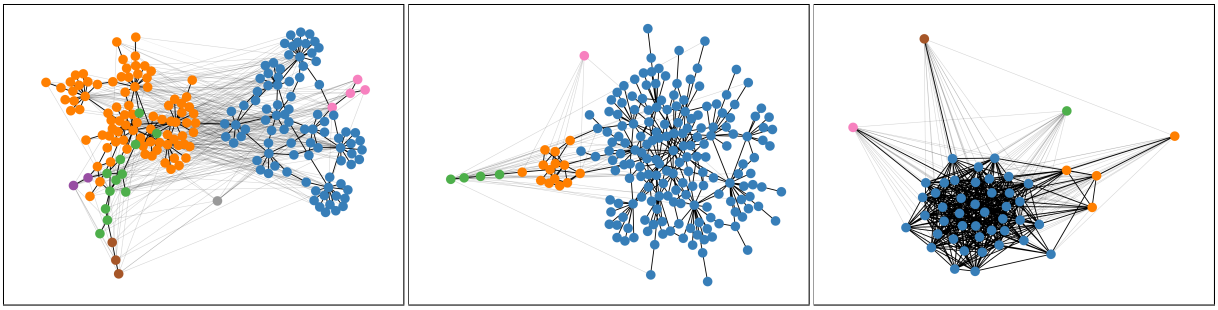


Figure 1: WUGs from different datasets (Schlechtweg et al., 2021b; Kurtyigit et al., 2021): English *plane* (left), Swedish *färg* (middle) and German *anpflanzen* (right). Noisy nodes and isolates were removed.

1.0.0 for our experiments.[3] DWUG DE Sense is a subset of DWUG DE in terms of target words and uses. However, it provides word sense annotations following the more established use-sense approach widely used in WSD (e.g. Langone et al., 2004; Hovy et al., 2006). It was cleaned on the use level. Hence, it can serve as a reliable gold standard to evaluate clusterings on the DWUG DE graphs.

4 Tasks

A WUG $\mathbf{G} = (\mathbf{U}, \mathbf{E}, \mathbf{W})$ is a weighted, undirected graph, where nodes $u \in U$ represent word uses and weights $w \in W$ represent the human-annotated semantic proximity of a pair of uses (an edge) $(u_1, u_2) \in E$ (McCarthy et al., 2016; Schlechtweg et al., 2020). We approach two tasks: Given an incomplete and noisy WUG,

1. cluster the G based on the edge weights such that uses with the same sense are in the same cluster,
2. remove nodes from the G which were clustered incorrectly while removing as few nodes as possible.

Note that the first task is basically WSI under specific conditions. Our main quality metrics are the Rand Index and the Adjusted Rand Index (RI and ARI, Hubert and Arabie, 1985) against gold clusters, and for the second task we additionally report the amount of nodes, senses, clusters and whole lemmas removed.

5 Models

In this section, we describe models solving two tasks described above, i.e., clustering and cleaning models.

[3] https://zenodo.org/records/8197553.

5.1 Graph preprocessing

First, note that the pre-processing techniques described below are not tested with WSBM as it shifts edge weights from discrete to dense values requiring dense distribution estimation during model fitting, which we found to suffer from a bug in graph-tool version 2.58 installed with Anaconda.

We define two graph preprocessing parameters:

- `t`: threshold for shifting all edge weights and
- `dwn`: downscales the influence of annotation noise.

We shift each edge weight $w = W(e)$, $e \in E$ to $w' = w - t$ where t is the `threshold` parameter. For both, CW and CC, this parameter decreases the influence of edge weights that are close to the threshold on cluster comparison scores during clustering (the sum of edge weights for Chinese Whispers and the cluster loss for Correlation Clustering). We test the values $t \in \{0.0, 1.8, 1.9 \ldots 3.2\}$ where the the value 0.0 corresponds to the original edge weights.

In their manual analysis, Schlechtweg (2023, p. 60) observed disagreements stemming from ambiguity of some word uses to be the major factor determining clustering errors. Hence, we introduce the `dwn` parameter scaling each shifted edge weight $w'(e) := w'(e)(1 - \sigma)$ where σ is the standard deviation of all judgements on edge e. This lowers the absolute values of questionable edge weights, which should decrease their influence on the clustering with Chinese Whispers and Correlation Clustering. We test models with and without downscaling.

5.2 Clustering

For model choice, we rely on the results from previous studies (Schlechtweg et al., 2021a; Tunc, 2021; Kotchourko, 2021). Below, we assume that cluster-

ing algorithms operate on the preprocessed graphs from Section 5.1.

Chinese Whispers Chinese Whispers (CW) is an efficient, randomized clustering algorithm with a time complexity linear with respect to the number of edges (Biemann, 2006). The algorithm first assigns all nodes to different clusters. Then the nodes are processed in randomized order for a small number of iterations (we set this hyperparameter to 20) and are assigned to the strongest cluster in the local neighborhood, i.e., the cluster whose sum of edge weights to the current node is maximal, i.e., given the currently processed node u and $N(u)$ being the set of all neighbouring nodes, u's new cluster assignment will be given by

$$C(u) := \arg\max_{c} \sum_{n \in (c \cap N(u))} weight((u,n))$$

where `weight` is a hyperparameter, which takes three different values:

1. *lin*: This calculates the weight of an edge between two nodes in a graph using linear weighting, which is the edge weight divided by the degree of the destination node: $weight((u,n)) = W(u,n)/d(n)$.
2. *log*: This computes the weight of an edge between two nodes in a graph using logarithm weighting, which is the edge weight divided by the logarithm of the degree of the destination node (shifted by one to avoid zero division): $weight((u,n)) = W(u,n)/log_2(d(n)+1)$.
3. *top*: This keeps edge weight as is: $weight((u,n)) = W(u,n)$.

We use the implementation provided by Ustalov et al. (2019).[4]

Correlation Clustering We use a variation of Correlation Clustering (CC) (Bansal et al., 2004), a graph clustering technique which minimizes the sum of cluster disagreements, i.e., the sum of negative edge weights within a cluster and the positive edge weights across clusters (Schlechtweg et al., 2020). CC has been used extensively in the LSCD context to cluster human annotations (Schlechtweg et al., 2021b; Kurtyigit et al., 2021; Kutuzov et al., 2022; Baldissin et al., 2022; Aksenova et al., 2022; Zamora-Reina et al., 2022; Chen et al., 2023). Those edges e with a weight $W(e) \geq 0$ are referred to as **positive** edges P_E while edges with weights $W(e) < 0$ are called **negative** edges N_E. Let further $C : U \mapsto L$ be some clustering on U, $\phi_{E,C}$ be the set of positive (high) edges **across** any of the clusters in clustering C and $\psi_{E,C}$ the set of negative (low) edges **within** any of the clusters. The algorithm then searches for a clustering C that minimizes the sum of weighted cluster disagreements:

$$SWD(C) = \sum_{e \in \phi_{E,C}} W(e) + \sum_{e \in \psi_{E,C}} |W(e)| \,.$$

That is, the sum of positive edge weights between clusters and (absolute) negative edge weights within clusters is minimized. Minimizing SWD is a discrete optimization problem which is NP-hard (Bansal et al., 2004). We use the implementation of Schlechtweg et al. (2021b).[5] The implementation approximates the global optimum with Simulated Annealing (Pincus, 1970), a standard discrete optimization algorithm. In order to reduce the search space, the implementation iterates over different values for the maximum number of clusters. It also iterates over randomly as well as heuristically chosen initial clustering states. The implementation has the following hyperparameters:

- $\texttt{t}_{\texttt{CC}}$: a threshold for shifting and splitting edge weights into positive and negative.
- $\texttt{max}_{\texttt{clusters}}$: the maximum number of clusters allowed in the search space,
- $\texttt{max}_{\texttt{atm}}$ and $\texttt{max}_{\texttt{iter}}$: the maximum attempts and maximum iterations for simulated annealing and
- `rep`: the number of repetitions of the clustering.

$\texttt{t}_{\texttt{CC}}$ has an equivalent effect as the `threshold` parameter described above. Hence, it will not be varied. $\texttt{max}_{\texttt{clusters}}$ is set to 20 based on the assumption that most words have less than 20 senses. We set $\texttt{max}_{\texttt{atm}}$ and $\texttt{max}_{\texttt{iter}}$ to 2000 and 50000 respectively, and `rep` to 5. These have shown near to optimal performance on DWUG DE in Tunc (2021).

Weighted Stochastic Block Model We use a Bayesian formulation of the Weighted Stochastic Block Model (WSBM), a generative model for random graphs popular in biology, physics and social sciences (Aicher et al., 2014; Peixoto, 2017).

[4] https://github.com/nlpub/chinese-whispers

[5] https://github.com/Garrafao/WUGs

The model has been first applied on WUGs by Schlechtweg et al. (2021a) and subsequently by Kotchourko (2021) and Noble et al. (2024). The basic assumption of the WSBM is that nodes belong to latent blocks (clusters), and that nodes in the same block are stochastically equivalent (i.e., they have edges drawn from the same distribution). Fitting the model is equivalent to determining the optimal latent block structure providing a clustering of word uses.

The inference of the latent block structure is driven by both edge existence and edge weights. This is achieved by treating edge weights as covariates that are sampled from some distribution (e.g. binomial) conditioned on the vertex partition (Peixoto, 2014a), i.e.,

$$P(A, x|\theta, \gamma, b) = P(x|A, \gamma, b)P(A|\theta, b)$$

with the covariates being sampled only on existing edges, and where $\boldsymbol{\gamma}_{rs}$ is a set of parameters that govern the sampling of the weights between groups r and s. The posterior partition distribution is then

$$P(b|A, x) = \frac{P(x|A, b)P(A|b)P(b)}{P(A, x)},$$

omitting the parameters θ, γ as in the non-parametric WSBM through the use of marginal likelihoods (Peixoto, 2017). In our experiments we use the non-parametric, micro-canonical implementation of the WSBM which avoids explicitly encoding distribution parameters for edge weights by replacing them with hard quantities (Peixoto, 2014c). The non-parametric model avoids over-fitting, and micro-canonical distributions are easier to compute while approaching their canonical counterparts asymptotically (Peixoto, 2017). Finding the maximum of the posterior distribution of the WSBM is NP-hard (Peixoto, 2015). Hence, we infer the optimal partitioning of vertices $P(b|x)$ asymptotically with multilevel agglomerative Markov chain Monte Carlo (MCMC) Peixoto (2014b). All experiments were done with Peixoto (2017)'s implementation.[6] We keep all hyperparameters (e.g. the temperature parameter for MCMC, β) at their default values, except for the following ones:

- `dist`: the distribution fitted to the observed edge weights within and between blocks (clusters),
- `mrg`: whether to marginalize out edge probabilities,
- `dgr`: whether to use the degree-corrected model version,
- $\texttt{B}_{\texttt{max}}$: the maximum number of clusters to consider during search and
- `niter`: the number of sweeps performed in multilevel agglomerative acceptance-rejection MCMC search.

We test all discrete distributions available in the implementation: poisson, binomial, and geometric. We test the model with and without marginalizing out edge probabilities (Schlechtweg et al., 2021a), with and without degree-correction (Karrer and Newman, 2011). $\texttt{B}_{\texttt{max}}$ and `niter` are set to 30 and 100 respectively. These choices for manipulation are driven by Schlechtweg et al. (2021a)'s findings and examples in Peixoto (2014a).

5.3 Postprocessing

We define the following cluster postprocessing parameters:

- $\texttt{t}_{\texttt{clps}}$: threshold for collapsing clusters.

We apply a cluster postprocessing step merging clusters with the average between-cluster edge weights above $\texttt{t}_{\texttt{clps}}$. This parameter follows the idea that nodes from two different clusters should really correspond to two different senses, thus, the between-cluster edges should have judgments from the lower end of the annotation scale. Hence, cluster with high judgments on the between-cluster edges likely correspond to the same sense and should be merged.

5.4 Cleaning

There are no previous studies on cleaning strategies for WUGs. Hence, we derive a number of postprocessing (cleaning) heuristics based on the insights obtained from a manual analysis of clustering errors (Schlechtweg, 2023, pp. 59–61):

- $\texttt{t}_{\texttt{stdnode}}$: remove nodes with average standard deviation on its edges above the threshold,
- $\texttt{t}_{\texttt{dgrnode}}$: remove nodes with degree (number of edges) below the threshold,
- $\texttt{t}_{\texttt{sizecluster}}$: remove clusters with a size below the threshold,
- $\texttt{t}_{\texttt{cntcluster}}$: remove poorly connected clusters. We calculate the percentage of annotated edges for each cluster pair and then average these percentages per cluster. We then remove clusters with an average connectedness below the threshold.

[6] https://graph-tool.skewed.de/

These strategies are motivated by the hypothesis that clustering errors mainly stem from noise in the graph (e.g. through ambiguity), edge sparsity or their combination. We test the effect of each of the hyperparameters individually, reducing the size of the grid. We also test the effect of the above-described cluster postprocessing parameter t_{clps} in more detail as part of the cleaning experiments, which was not feasible above given the large hyperparameter grid in the clustering experiments. For each cleaning run, we first remove nodes and edges with a high number of 0-judgments before applying the cleaning strategy. We further remove all isolates.For each threshold hyperparameter, we calculate the grid by first gathering all observed values of the underlying variable (e.g. all degrees of nodes for $t_{dgrnode}$) over all graphs in DWUG DE with published opt clusterings. We then divide these scores in 50 percentiles and select all unique percentiles for the respective hyperparameter grid. This way we avoid testing many values having little effect on the graphs.

6 Experiments

We now apply the clustering and cleaning models described above to the annotated graphs from DWUG DE varying their hyperparameters. Cluster quality is measured as correspondence to DWUG DE Sense clusters with the Adjusted Rand Index (ARI, Hubert and Arabie, 1985).

6.1 Clustering

Evaluation Setup. To compare different methods, we need some labelled data to select optimal hyperparameters for each of them, and also separate labelled data to calculate the unbiased estimates of clustering quality. Since we have only 24 words in our dataset, we decided to employ leave-one-out cross-validation. Specifically, following the idea proposed in (Cawley and Talbot, 2010) when calculating ARI of a method for a particular test word, we first select the hyperparameters that maximise the average ARI on all other words, then calculate ARI for the test word. This helps avoid over-fitting during hyperparameter selection and obtain unbiased estimates of ARI for each method on each test word.

This evaluation setup helps to get unbiased estimates of the quality of a whole pipeline consisting of a particular clustering method, pre- and post-processing steps, and also the hyperparameter selection approach. However, it potentially select entirely different hyperparameter configurations for each fold. Table 2 shows the number of folds each configuration was selected for. Evidently, for each clustering method there exists a winning configuration selected for 70-80% of folds. This configuration can be recommended as the default one when running our pipelines on new data.

Result Overview. The cross-validated ARI for each clustering method is reported in Table 2. WSBM outperforms two other methods, and CW demonstrates poor performance. Additionally, Appendix B compares these methods on each word individually.

Since the test set contains 24 words only, it is important to check for statistical significance of the differences between performance of our clustering methods. We set the confidence level of 5% and employ the Wilcoxon signed-rank test with one-sided alternative and Pratt method (Pratt, 1959) to account for zero differences.The differences between WSBM and CW, and also between CC and CW are statistically significant, while the difference between WSBM and CC is not. We can reliably conclude that CW is worse than two other methods. The size of the test set does not allow to draw reliable conclusions regarding the comparison of WSBM and CC though.

Previous Models. Another way to select hyperparameter configurations for testing is by previous results or theoretical argument. This way, we select two model configurations to compare: (i) The WSBM+t_{clps}=False+dist=binomial+mrg=True +dgr=False and (ii) CC+t=2.5+dwn=False +t_{clps}=False. The first is suggested by the rather superficial evaluation of Schlechtweg et al. (2021a) and has a theoretical motivation as statistically sound model (Peixoto, 2017). The second model has a theoretical motivation based on the interpretation of the annotation scale (Schlechtweg et al., 2020) and was used to create the published clusterings for most WUG datasets (e.g. Schlechtweg et al., 2021b; Kurtyigit et al., 2021; Chen et al., 2023). We now test whether their performances are significantly different: The two models have an average ARI of .81 and .75 respectively. The Wilcoxon test shows that the difference is not statistically significant though.[7]

[7]However, p-value is 0.0516, which is only a bit higher than our critical value of 0.05.

method	ARI	t	dwn	t_{clps}	dist	mrg	dgr	weight	#folds
		-	**-**	**False**	**binomial**	**True**	**True**	**-**	**20**
WSBM	.76	-	-	2.4	binomial	True	False	-	2
		-	-	False	binomial	True	False	-	1
		-	-	2.3	binomial	True	True	-	1
		2.5	**True**	**2.3**	**-**	**-**	**-**	**-**	**18**
		2.4	True	2.4	-	-	-	-	2
CC	.72	2.5	True	2.4	-	-	-	-	1
		2.6	False	2.3	-	-	-	-	1
		2.9	True	2.3	-	-	-	-	1
		2.6	True	2.3	-	-	-	-	1
		3.0	**True**	**2.4**	**-**	**-**	**-**	**top**	**17**
		2.0	True	2.4	-	-	-	top	2
CW	.55	2.8	False	2.4	-	-	-	top	2
		2.2	True	2.3	-	-	-	top	1
		2.2	True	2.4	-	-	-	top	1
		2.9	True	2.3	-	-	-	top	1

Table 2: The configurations of hyperparameters selected for each method in at least one CV fold. The configuration selected for the majority of folds is in **bold**. "-" marks non-applicable parameters for the respective method.

Shifting and collapsing thresholds. CC and CW profit from post-clustering collapsing of clusters with t_{clps}=2.3/2.4. For CC, the shifting threshold t=2.5 seems optimal, however with additional post-collapsing at 2.3. For CW, a higher shifting thresholds of t=3.0 seems optimal, with additional collapsing at 2.4. Collapsing has no pronounced effect for the WSBM.

Ambiguity downscaling. Table 2 indicates that dwn=True has a positive effect on CC and CW as it is part of most optimal configurations. For CC, dwn=True for 23/24 folds while for CW this is true for 22/24 folds.

Edge marginalization, degree correction and distribution for WSBM. The performance difference for WSBM+t_{clps}=False+dist=binomial with and without edge marginalization (mrg=False/True) is .175/.176 vs. .813/.815 (dgr=False/True), the difference is statistically significant and also quite large. This trend is opposite for other distributions (poisson, geometric), but the performance of these models is always lower than 0.34. This confirms the observations of Schlechtweg et al., but leaves an open question why the trend is inconsistent across distributions. Furthermore, 21/24 folds have dgr=True suggesting that degree-correction has a positive effect. The binomial distribution is part of the selected model on all folds confirming previous results by Schlechtweg et al. (2021a).

Weight parameter for CW. The weight parameter is weight=top across all selected models suggesting that degree weighting has no positive effect for CW.

6.2 Cleaning

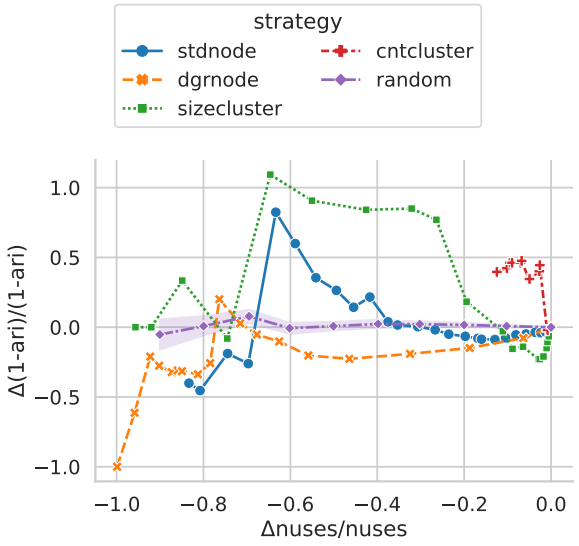
As indicated above, cleaning experiments were only performed for the published DWUG DE opt clusterings, which were obtained with CC at a threshold of 2.5 after removing nodes with a high number of 0-judgments and any nan edges and isolates, without further preprocessing or postprocessing. This choice is driven by the fact that this clustering approach is widely used for other datasets and thus our results can more easily be assumed to generalize to these datasets.

Evaluation Setup. To compare different cleaning methods we have to compare the trade-offs they offer between the amount of information removed from a graph and the clustering quality of the remaining part of this graph, next we explain how we quantify these trade-offs. Better cleaning methods result in higher quality for the same proportion of removed uses. In addition, it is important to consider the number of senses that have all of their uses removed after cleaning because, generally, we would prefer a method that preserves all or almost all word senses even if it removes more uses e.g. due to heavier filtering of uses of the most frequent senses.

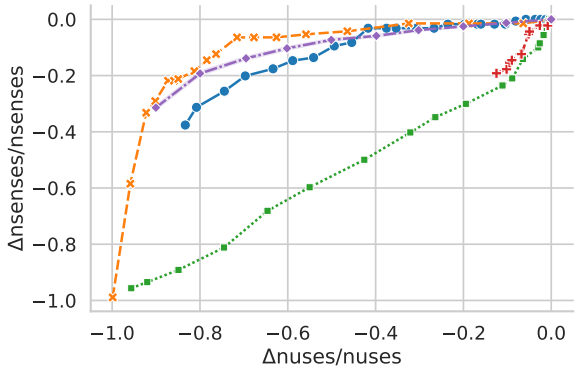
As a measure of change in the number of uses we employ the relative change averaged across all target words:

$$\frac{\Delta uses}{uses} = \frac{1}{N}\sum_{i=1}^{N}\frac{uses'_i - uses_i}{uses_i}$$

where $uses_i$ and $uses'_i$ are the number of uses of the i-th target word before and after cleaning,



(a) the complement of ARI (↓)



(b) the number of senses (↑)

Figure 2: Comparison of cleaning methods. Relative changes are plotted, arrows show if we prefer higher or lower values. For the random baseline we show the mean and the 95% CI for the mean based on 100 runs for each X value.

N is the number of target words. Similarly, the relative change in the number of senses or clusters is calculated first for each target word, and then averaged across all target words. If all uses of a particular target word are filtered, out we naturally define the number of uses, senses and clusters for this word as 0.

As the main measure of change in the clustering quality we rely on the relative change of the complement of ARI, Appendix C explains why this metric is selected. First, we calculate the relative change for each word:

$$\frac{\Delta(1-ari_i)}{(1-ari_i)} = \frac{(1-ari_i') - (1-ari_i)}{1-ari_i}$$

where ari_i and ari_i' measure the quality of the original clustering of all uses and clustering of uses that survived after cleaning. For a target word with no uses, the quality of clustering cannot be naturally defined. Thus, the metrics of clustering quality are averaged only across those words that have some uses survived after filtering. In Appendix D we compare the results using both the original clustering metrics ARI and RI, and also their complements, all of them are calculated the same way.

Result Overview. Figure 2 compares the cleaning methods including a random baseline, which randomly removes the given proportion of nodes. The first plot shows the relative change of the complement of ARI averaged across those target words that have at least one use after cleaning, the values below 0 mean an increase of clustering quality after cleaning. The second plot shows the average relative change of the number of senses for all target words, the higher values meaning more senses survived are preferable.

The only method that consistently improves the clustering quality while preserving almost all senses is *dgrnode*. It also leaves more senses compared to all other methods when the same proportion of uses is removed.The best clustering quality is obtained when roughly half of the uses are removed. Despite the proportion of removed uses is large in this case, very few senses are fully filtered out making this filtering configuration practically useful. *Sizecluster* gives a comparable improvement with much fewer uses removed, but at the same time with much more senses fully lost due to removing whole clusters. *Stdnode* is competitive when we allow removing only 5-10% of nodes, but cannot give the same improvement as *dgrnode* unless 70-80% of nodes are filtered out and 30-40% of senses are lost which seems hardly acceptable for practical use. Finally, *cntcluster* results in a large loss of senses and no improvement in clustering quality.

7 Conclusion

We systematically evaluated graph clustering algorithms and cleaning strategies on manually annotated WUGs. We were able to show that the Weighted Stochastic Block Model outperforms the previous state-of-the-art model, Correlation Clustering. However, the difference was not statistically significant. Further, we identified the removal of nodes by their degree as effective cleaning strategy. We publish the improved clusterings and cleaned graphs for further research.

8 Limitations

The main limitation of our work is the size of the gold dataset containing only 24 words. This is due to the scarcity of data annotated both with the WUG and the traditional approach. Further, the DWUG DE WUGs have been annotated with a special annotation approach in rounds involving many edge sampling heuristics. We do not know whether our results generalize to WUGs annotated in a different way, e.g. with random sampling of edges.

Some interesting experiments are missing in our work: We did not test the dense version of the WSBM. We also only tested the cleaning strategies on the published clusterings and did not test combinations of cleaning strategies. We only applied cleaning as a post-processing step, but it could be applied as a pre-processing before clustering.

Our work relies on the assumptions that semantic proximity judgments between pairs of uses reflect the structure of traditional sense definition judgments as we use the former to reproduce the latter through clustering. This assumption sometimes does not hold, also because annotators may disagree in the interpretation of the word uses.

References

Christopher Aicher, Abigail Z. Jacobs, and Aaron Clauset. 2014. Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221—-248.

Anna Aksenova, Ekaterina Gavrishina, Elisei Rykov, and Andrey Kutuzov. 2022. RuDSI: Graph-based word sense induction dataset for Russian. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 77–88, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Gioia Baldissin, Dominik Schlechtweg, and Sabine Schulte im Walde. 2022. DiaWUG: A Dataset for Diatopic Lexical Semantic Variation in Spanish. In *Proceedings of the 13th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine Learning*, 56(1-3):89–113.

Chris Biemann. 2006. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, page 73–80, USA. Association for Computational Linguistics.

Gavin C. Cawley and Nicola L.C. Talbot. 2010. On overfitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, 11:2079–2107.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023. ChiWUG: A graph-based evaluation dataset for Chinese lexical semantic change detection. In *Proceedings of the 4th International Workshop on Computational Approaches to Historical Language Change*, Singapore. Association for Computational Linguistics.

Mario Giulianelli, Iris Luden, Raquel Fernandez, and Andrey Kutuzov. 2023. Interpretable word sense representations via definition generation: The case of semantic change analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57—-60, USA. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2:193–218.

Brian Karrer and M. E. J. Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107.

Serge Kotchourko. 2021. Optimizing human annotation of word usage graphs in a realistic simulation environment. Bachelor thesis, University of Stuttgart.

Sinan Kurtyigit, Maike Park, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Lexical Semantic Change Discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online. Association for Computational Linguistics.

Andrey Kutuzov, Mariia Fedorova, Dominik Schlechtweg, and Nikolay Arefyev. 2024. Enriching word usage graphs with cluster definitions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6189–6198, Torino, Italia. ELRA and ICCL.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022. NorDiaChange: Diachronic semantic change dataset for

Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Helen Langone, Benjamin R. Haskell, and George A. Miller. 2004. Annotating wordnet. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL*, Boston, MA, USA.

Diana McCarthy, Marianna Apidianaki, and Katrin Erk. 2016. Word sense clustering and clusterability. *Computational Linguistics*, 42(2):245–275.

Bill Noble, Francesco Periti, and Nina Tahmasebi. 2024. Improving word usage graphs with edge induction. In *Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change*, pages 92–107, Bangkok, Thailand. Association for Computational Linguistics.

Tiago P. Peixoto. 2014a. Documentation of the graph-tool python library. (last checked july 17, 2020).

Tiago P. Peixoto. 2014b. Efficient monte carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E*, 89(1).

Tiago P. Peixoto. 2014c. The graph-tool python library. *figshare*.

Tiago P. Peixoto. 2015. Model selection and hypothesis testing for large-scale network models with overlapping groups. *Physical Review X*, 5:011033.

Tiago P. Peixoto. 2017. Nonparametric weighted stochastic block models. *Physical Review E*, 97.

Martin Pincus. 1970. A monte carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, 18(6):1225–1228.

John W. Pratt. 1959. Remarks on zeros and ties in the wilcoxon signed rank procedures. *Journal of the American Statistical Association*, 54(287):655–667.

Dominik Schlechtweg. 2023. *Human and Computational Measurement of Lexical Semantic Change*. Ph.D. thesis, University of Stuttgart, Stuttgart, Germany.

Dominik Schlechtweg, Enrique Castaneda, Jonas Kuhn, and Sabine Schulte im Walde. 2021a. Modeling sense structure in word usage graphs with the weighted stochastic block model. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 241–251, Online. Association for Computational Linguistics.

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

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

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021b. DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Tunc. 2021. Optimierung von Clustering von Wortverwendungsgraphen. Bachelor thesis, University of Stuttgart.

Dmitry Ustalov, Alexander Panchenko, Chris Biemann, and Simone Paolo Ponzetto. 2019. Watset: Local-Global Graph Clustering with Applications in Sense and Frame Induction. *Computational Linguistics*, 45(3):423–479.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022. LSCDiscovery: A shared task on semantic change discovery and detection in Spanish. In *Proceedings of the 3rd International Workshop on Computational Approaches to Historical Language Change*, Dublin, Ireland. Association for Computational Linguistics.

4: Identical
3: Closely Related
2: Distantly Related
1: Unrelated

Table 3: The DURel relatedness scale (Schlechtweg et al., 2018).

A Annotation scale

Table 3 shows the ordinal annotation scale for use pairs used in the WUG paradigm.

B Cross-validation results per word

Figure 3 compares ARI for the three clustering methods for each test word individually. For 18 out of 24 words WSBM gives one of the best clusterings, and on 8 of those words it strictly outperforms two other methods. For comparison, the second best performing method CC returns one of the best clusterings for 16 out of 24 words, and for 5 of those words it is strictly better than the others. For 7 words WSBM returns the clustering identical to the ground truth one, getting the ARI of 1.0. For the words *Titel* and *Seminar* no method shows reasonable ARI.

C Metrics for the cleaning experiments

We considered different metrics as the main quality metrics, and finally selected the relative change in the complements of RI and ARI averaged across the survived target words. As RI is the pairwise accuracy, i.e. the proportion of pairs of uses that are correctly put into the same or different clusters (depending on their gold labels), and ARI is its shifted and scaled version, their complements 1-RI and 1-ARI quantify the pairwise error rate. We argue that averaging the relative changes in the error rate better reflect our intuition that equivalent absolute improvements in RI for the words that are already clustered almost perfectly and those with very bad clustering are not comparable.

As an example, consider two lemmas with RI for the first decreased from 0.9 to 0.8 and for the second increased from 0.2 to 0.3 after cleaning. Intuitively, the second change is much smaller than the first one, and the overall performance is now worse. But after averaging the absolute RIs we will conclude that nothing changed: $\overline{\Delta ri} = \frac{(0.8-0.9)+(0.3-0.2)}{2} = 0$. The relative change of 1-ri will reveal that we have 2x more pairwise errors (the relative increase in error rate of 1.0) for the first lemma and a small decrease for the second one, so the average will be significantly larger: $\overline{\Delta(1-ri)/(1-ri)} = \frac{(0.2-0.1)/0.1+(0.7-0.8)/0.8}{2} = \frac{1-1/8}{2} = 0.4375$. This can be interpreted as an increase in the number of errors by 43.75%.

D Extended comparison of cleaning methods

Figure 4 shows how the number of clusters and the number of senses change as usages are removed. Clearly we want as few senses to be removed as possible, but removing some clusters may be desirable if they poorly correspond to senses. The most conservative method is *dgrnode*, for the same proportion of removed nodes it removes the smallest number of senses and clusters. For the methods removing whole clusters, i.e. *cntcluster* and *sizecluster*, we see that both the number of clusters and the number of senses reduce rapidly, but the number of senses decrease a bit more slowly in the beginning. This is probably due to some senses appearing in small removed clusters also appear in larger clusters.

Figure 5 extends figure 2 with additional metrics. However, they show a similar overall picture. Compared to the relative changes, the absolute value of ARI similarly shows a bit less articulated but consistent superiority of *dgrnode* over other methods.

For the methods removing whole clusters, i.e. *sizecluster* and *cntcluster*, when comparing the relative change of RI and ARI the results are contradictory. This is likely related to RI of a random assignment of uses to clusters becoming better as the number of clusters decreases, which is taken into account by ARI.

Figures 6 and 7 show how ARI changes after cleaning for each target word individually. There is no single method that outperforms all other methods or at least the random baseline on all words.

E Collapsing threshold

Figure 8 shows the effect of collapsing with different thresholds on ARI. The left plot shows the average ARI across all target words, seemingly there is a significant increase if the threshold is properly selected. However, from the right plot we see that if the word *artikulieren* is excluded

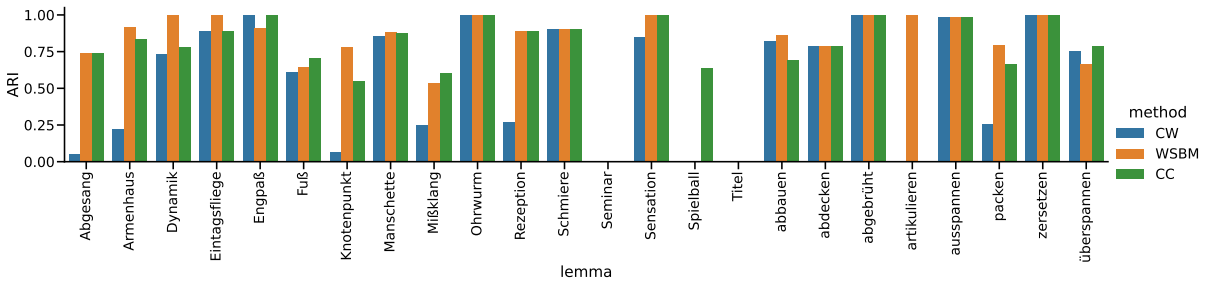


Figure 3: Comparison of clustering quality for each test word. The unbiased estimates of ARI obtained from cross-validation are shown.

the positive effect from collapsing becomes very small. Figure 9 explores the effect of collapsing for each target word individually. For most words collapsing cannot help, but can significantly hurt if the threshold is too small, i.e. too few clusters remain. For a few words (*Engpaß, Rezeption, packen, überspannen*) collapsing gives a small but consistent improvement. For the word *artikulieren* the improvement in ARI is huge, from 0 to 1, but this is related to this word having only a single sense, so any clustering except for merging all uses together will give the ARI of 0.

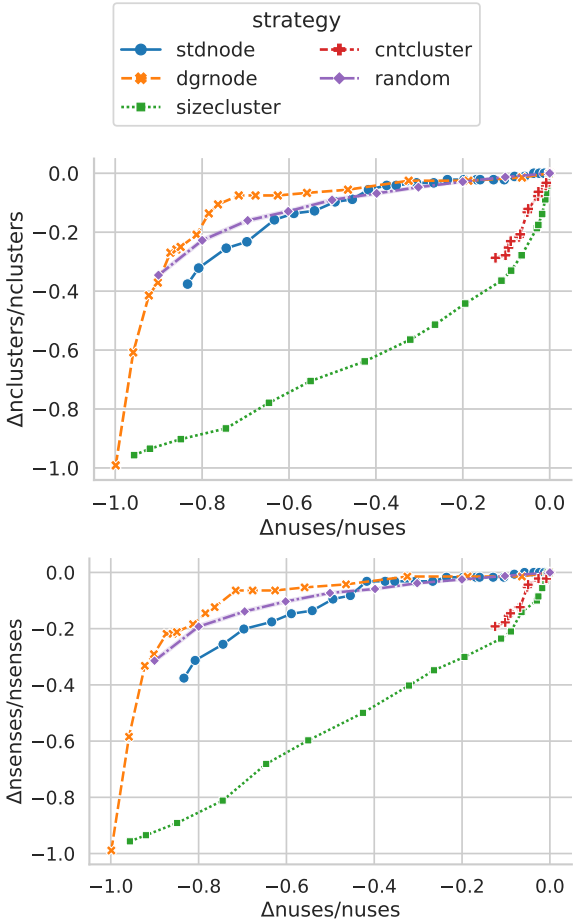
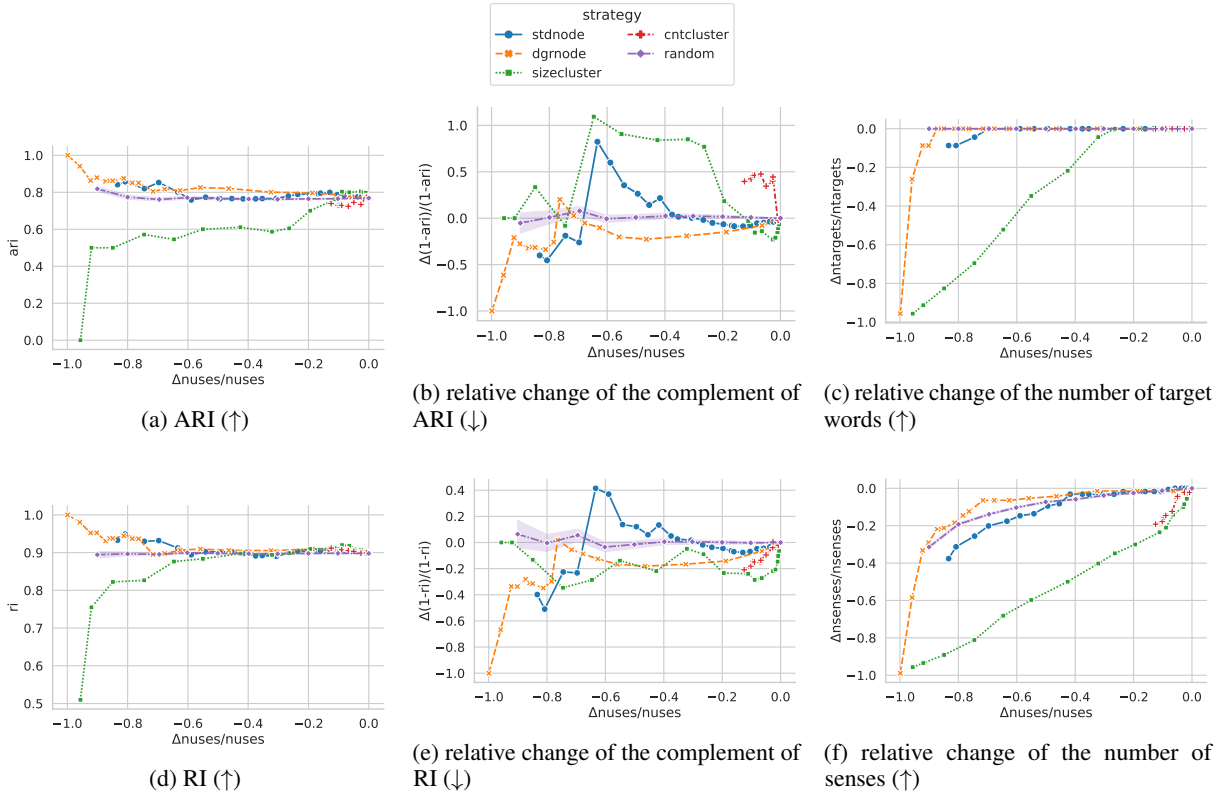


Figure 4: Relative change of the number of clusters and senses after cleaning averaged across target words.



(a) ARI (↑)

(b) relative change of the complement of ARI (↓)

(c) relative change of the number of target words (↑)

(d) RI (↑)

(e) relative change of the complement of RI (↓)

(f) relative change of the number of senses (↑)

Figure 5: Comparison of cleaning methods, the extended version.

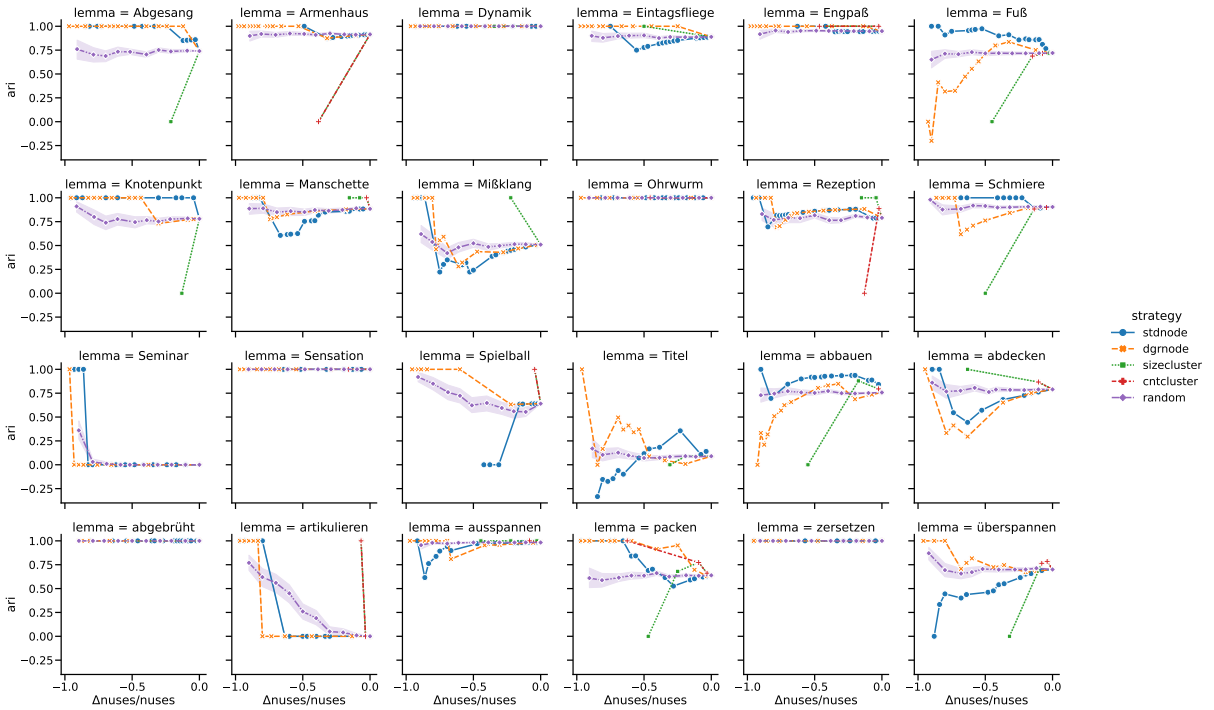


Figure 6: ARI individually for each target word.

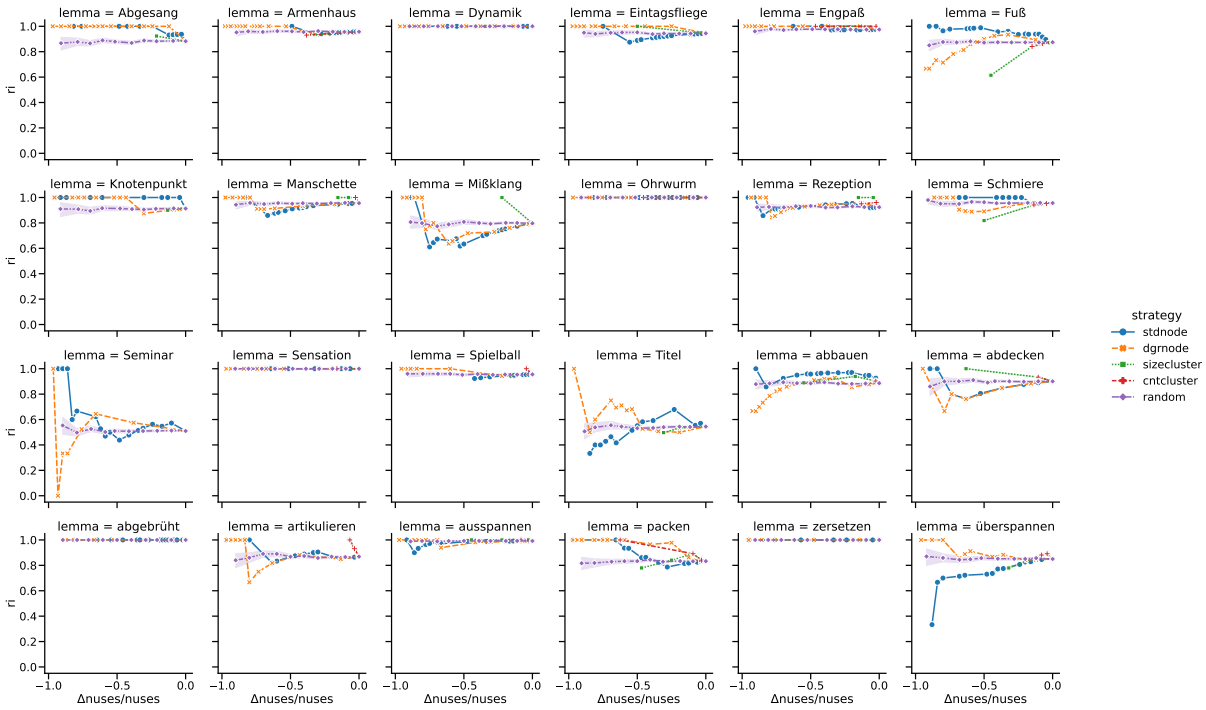


Figure 7: RI individually for each target word.

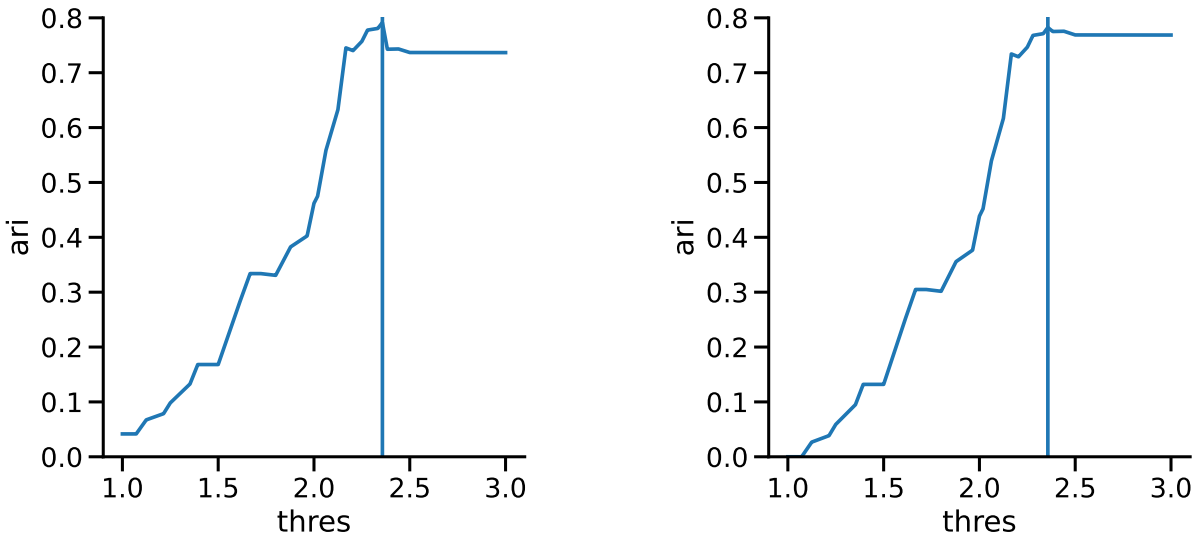


Figure 8: ARI w.r.t. the collapsing threshold. ARI is averaged across all words (left) and all words excluding *artikulieren* (right). The vertical line denotes the optimal threshold, which is the same in both cases.

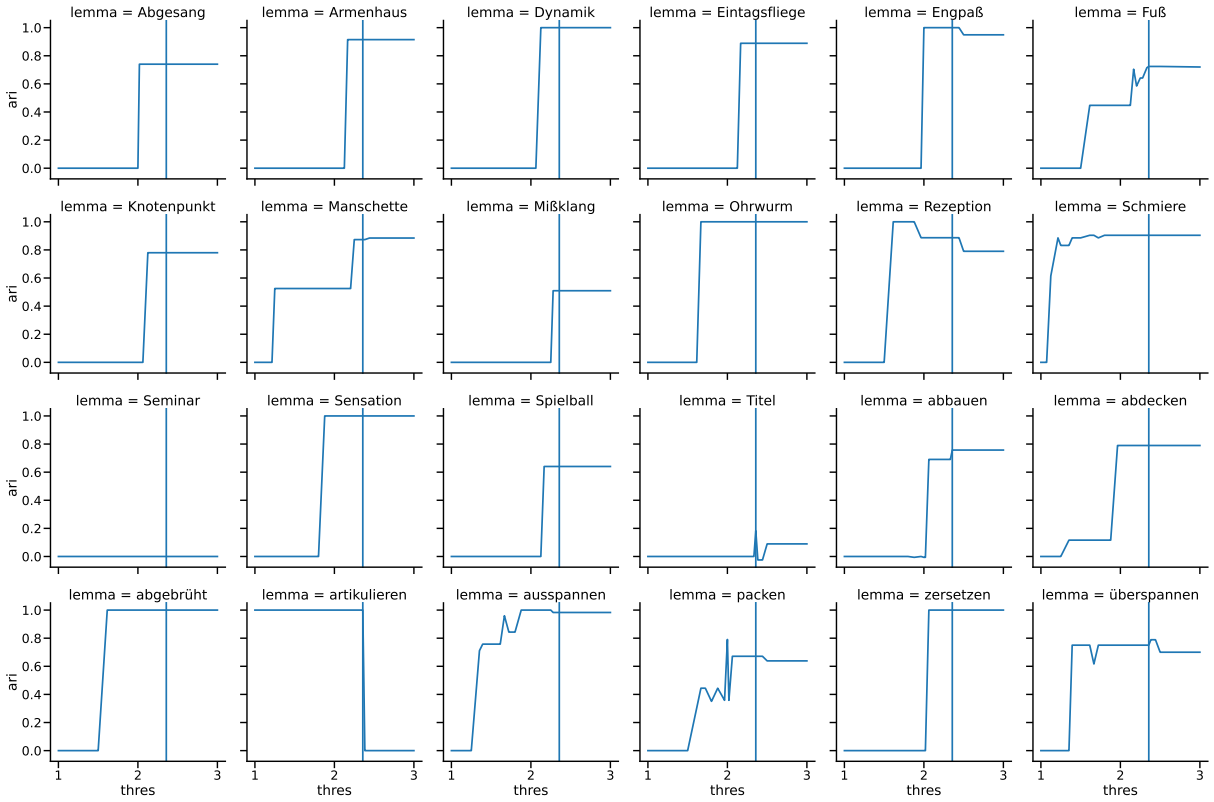


Figure 9: ARI individually for each target word w.r.t. the collapsing threshold. The vertical line denotes the threshold maximizing the average ARI across all target words.