Improving Word Usage Graphs With Edge Induction

Bill Noble
bill.noble@gu.se
University of Gothenburg

Francesco Periti
francesco.periti@unimi.it
University of Milan

Nina Tahmasebi
nina.tahmasebi@gu.se
University of Gothenburg

A **Word Usage Graph** (WUG) is a set of usages for a particular word, along with a number of relatedness scores between those usages.

1 = unrelated; 2 = closely related; 3 = distantly related; 4 = identical

Graph clustering can be used to infer infer a word sense for each usage in a WUG without relying on traditional (costly) word sense annotation. WUGs can be used to measure lexical semantic change via interpretable time-bound word sense frequency distributions (Schlechtweg et al., 2020; Periti & Tahmasebi, 2024).

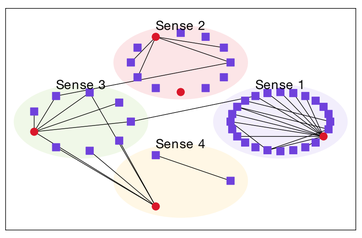
Human-annotated WUGs are typically sparsely annotated since annotating every usage-usage pair would be resource intensive. Can edge induction models act as **computational annotators** of incomplete WUGs?

Research Questions

RQ 1: Can we improve the annotation-efficiency by inferring missing graph edges before clustering the graph?

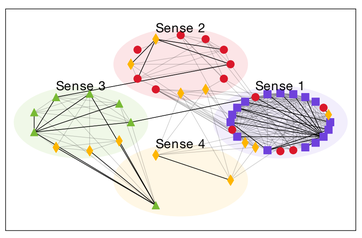
RQ 2: If so, what is the relative predictive contribution of **structural features** (from existing WUG edges) and **contextual features** (from distributional features of the usage texts)?

Example: *Ausspannen*

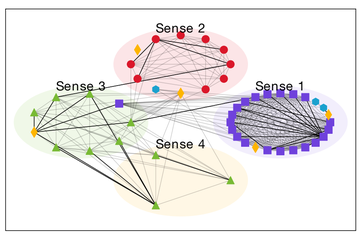


Usage clusters for the word *ausspannen* inferred from sparse human judgments.

Node color indicates inferred cluster of the usage and background color indicates the ground truth human-annotated sense.



Usage clusters after adding edges inferred with graph-structural features (**triangle evidence**).



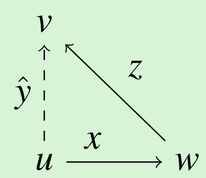
Usage clusters after adding edges inferred with both graph-structural and textual features (**XL-Lexeme similarity**).

Model features

all experiments use logistic regression models

log-triangle

Intuition: If we know the value of edges ***x*** and ***z***, this should tell us something about the unknown edge ***y***. More generally, we can count up the values along all the length-two paths from ***u*** to ***v***.



$$\mathbf{x}^{\text{tri}}_{\langle u,v\rangle}[i] = \sum_{w_j}\{1 \mid \langle W(u,w_j), W(w_j,v)\rangle = p_i\}$$

where ***i*** indexes each of the possible length-two paths. I.e., {(1,2), (1,3), (1,4), (2,1), ...}

Each additional path of the same value adds less information, so we take the log of the count vector

$$\mathbf{x}^{\text{log-tri}}_{\langle u,v\rangle}[i] = \log(\mathbf{x}^{\text{tri}}_{\langle u,v\rangle}[i] + 1)$$

xl-lexeme-cos

Intuition: XL-LEXEME (Cassotti et al., 2023) is a state-of-the-art Word in Context (WIC) model; given a pair of contexts containing the same target word, it predicts if their in-context meaning is the same or different. The cosine distance between XL-LEXEME embeddings should also be useful for the WUG edge annotation task (classifying relatedess scores 1 to 4).

$$x^{\text{xl-lex}}_{\langle u,v\rangle} = \delta^{\cos}(\mathbf{u}, \mathbf{v})$$

where ***u*** and ***v*** are the XL-Lexeme embeddings of the two in-context targets

xl-lexme-cos + log-triangle

Intuition: Two features are better than one.

$$\mathbf{x}^{\text{log-tri+xl-lex}}_{\langle u,v\rangle} = \mathbf{x}^{\text{log-tri}}_{\langle u,v\rangle} \oplus [x^{\text{xl-lex}}_{\langle u,v\rangle}]$$

Modeling choices

We can choose how to ***stratify*** the data.

word-level: A separate classifier is trained for each lexeme. It's trained on edges in the train set and tested on a set of held-out edges.

language-level: A classifier is trained for each language. Training examples are shared across lexemes.

cross-lingual: Just one classifier is trained, sharing data across all three languages.

When using the triangle-based features, we can do ***iterated inference***. After infering new edges, we update the features based on the new graph and run the inference again. The original ground-truth edges are always preserved.

To evaluate how much additional edges improve clustering, we need to choose a ***clustering algorithm***. The choice of algorithm may affect the results so we test three:

correlation: Possible partitions are scored with to the sum of edge weights within and across clusters. weights are adjusted so that 1 and 2 are considered negative

sbm-binomial: The Hierarchical Stochastic Block Model (Peixoto, 2014) is a generative model that assigns edges according to a hierarchy of block memberships (the inferred blocks are treated as clutsers). The binomial model draws edge weights from a binomial distribution

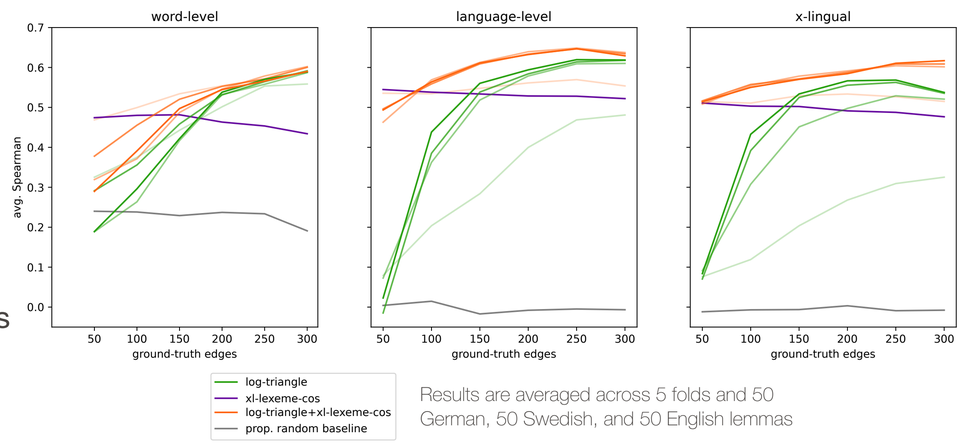
sbm-layers: The Layered SBM (Peixoto, 2015) treats each weight as a separate edge type and jointly infers a block structure for all types.

Experiment 1 edge induction performance

We test each model's classification performance on held-out annotated edges, with the rest of the edges used for training and computing structural features. Data: Schlechtweg et al., 2021

Performance is evaluated by Spearman correlation with human annotators. Inference iterations are shown with increasingly saturated lines (max 4).

Results: More edges are better, but sharing data across lexemes and languages can help. Both structural and contextual features are useful and interated inference is effective.

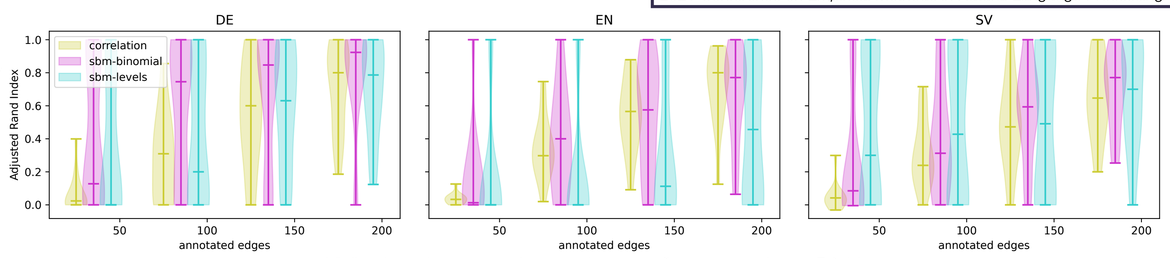


Results are averaged across 5 folds and 50 German, 50 Swedish, and 50 English lemmas

Experiment 2 clustering robustness

The goal of this experiment is to observe the stability of each clustering algorithm given different numbers of ground-truth edges. For this, we use a dataset of unusually densely annotated WUGs so that we can downsample edges to observe the effect. Robustness is measured with the Adjusted Rand Index (ARI) between clusters inferred from the downsampled WUG and those inferred from the WUG with 300 annotated edges.

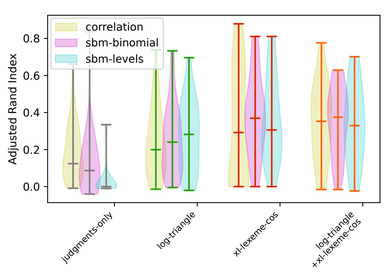
Results: The SBM-Binomial algorithm stabilizes the fastest, but there is still variation across words.



Violin plots show spread across lemmas, first averaged over 5 folds.

Experiment 3 realistic scenario

We chose 24 German lemmas for which there are both WUGs and direct human sense annotation. Starting with a median of 55 edges, we test how well the clusters correlate with sense annotation. The judgments-only baseline uses ground truth edges alone, while subsequent conditions also include induced edges as input to the clustering algorithm.



Results: Both the xl-lexeme-cos and log-triangle features help and SBM-binomial is the best-performing clustering algorithm in most cases.

Conclusions

1. Inferring missing edges before clustering can improve the resulting clusters, which increases annotation efficiency.
2. Both structural and contextual features contribute to the improvement in edge prediction and subsequent WUG clusters.

References

Pierluigi **Cassotti**, Lucia Siciliani, Marco DeGemmis, Giovanni Semeraro, and Pierpaolo Basile. **2023**. XL-LEXEME: WiC pretrained model for cross-lingual LEXical sEMantic changE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*

Tiago P. **Peixoto**. **2014**. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. *Physical Review*

Tiago P. **Peixoto**. **2015**. Inferring the mesoscale struc-Tture of layered, edge-valued, and time-varying net-works. *Physical Review*

Francesco **Periti** and Nina **Tahmasebi**. **2024**. A Systematic Comparison of Contextualized Word Embeddings for Lexical Semantic Change. In *Proceedings of the 2024 Conference of the North American Chapter of the 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 Fourteenth Workshop on Semantic Evaluation*

Dominik **Schlechtweg**, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. **2021**. 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*

Riksbankens Jubileumsfond grant M21-0021