Extending and Exploiting the Entity Graph for Analysis, Classification and Visualization of German Texts

Julia Suter and **Michael Strube**
Heidelberg Institute for Theoretical Studies
Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany
(julia.suter|michael.strube)@h-its.org

Abstract

We show that coherence information represented in an entity graph is valuable for NLP tasks as well as for the analysis of literary work. We present an extension of the entity graph for German by adding a syntactic role category for possession modifiers and reducing the weights for entities found in embedded structures. Furthermore, we extract a set of graph metrics and show that they are valuable features in text analysis of literary works and author classification.

1 Introduction

The entity graph introduced by Guinaudeau and Strube (2013) represents the relations between sentences and entities in a text and thus computes local coherence. Previous work has proven it a suitable method for measuring text coherence when applied to tasks such as sentence reordering, summary coherence rating and readability assessment (Guinaudeau and Strube, 2013; Mesgar and Strube, 2014; Mesgar and Strube, 2015). However, the entity graph has not yet been exploited to its fullest. First of all, it can still be improved by careful selection of entities and adjustment of the edges' weights. We show that adding a new syntactic category improves performance on the reordering task, and we introduce the idea of reducing weights for syntactically embedded entities.

Secondly, from the large set of possible graph measures, only the average out-degree has been used so far, specifically as a measure of coherence. We inspect other graph measures and use them for analyzing and visualizing literary work[1], crossing over into the field of digital humanities. While Stoddard (1991) manually extracted and depicted cohesion patterns in order to measure and investigate text cohesion, we can now automatically compute entity graphs to analyze coherence and other properties of texts. There is increasing demand for NLP methods in the humanities (see SIGHUM workshop[2]) and we suggest that the entity graph is a suitable representation of text with possible applications in both NLP and literary studies.

2 Entity Graph

In the entity graph $G = (V, E)$, the node set V contains all sentences and entities in a text, while E represents the set of all edges between sentences and entities (Guinaudeau and Strube, 2013). Function $w(s_i, e_j)$ indicates the weight of an edge which connects sentence s_i and entity e_j, where $w(s_i, e_j) = 1$ means that there is a mention of e_j in s_i. Following the insight of Grosz et al. (1995) that certain syntactic roles are more important than others, the syntactic role of e_j in s_i can be mapped to different edge weights:

$$w(s_i, e_j) = \begin{cases} 3 & \text{if } e_j \text{ is subject in } s_i \\ 2 & \text{if } e_j \text{ is object in } s_i \\ 1 & \text{otherwise} \end{cases}$$

The three different types of one-mode projections P_U, P_W and P_{Acc} capture the relations between sentences. P_U creates an edge between two sentences if they share at least one entity (binary weights). P_W follows the assumption that the connection between two sentences is stronger the more entities they have in common, so the edge weights represent the number of entities shared by two sentences. P_{Acc} integrates the syntactic information about the entities into the edge weights. The local coherence of a text can be measured by computing the average out-degree of the projection graph.

[1] Code and dataset are available at https://github.com/nlpAThits/entity-graph

[2] https://sighum.wordpress.com/events/latech-clfl-2018/

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

2.1 Extending the Entity Graph

We implemented the entity graph for German and extended the set of syntactic roles by one category representing two types of possession modifiers: possessive determiners and genitive modifiers. Possessive determiners can never be head of a noun phrase and thus are originally not considered in the entity grid and entity graph. However, they may contain relevant coherence information not captured by other entities: they create a link between the possessed entity and the possessor, which may not be explicitly mentioned in the current sentence. Example 1 demonstrates the issue: without adding the possession modifier category, the entity *John* would not be represented in the second sentence and thus the two sentences would not be connected by an edge in the entity graph.

(1) **John** fell down the stairs. Now **his** leg is broken.

Genitive modifiers on the other hand state an explicit possessor so they are considered in the original entity graph. They receive the default weight (1), even when dependent on a subject or object phrase as they do not represent the actual subject or object entity. In government and binding theory (Chomsky, 1981), however, the possessor is assumed to function as subject to the NP. Thus, we adjust the weight of possession modifiers by setting it equal to the weight of the entity they refer to.

A further adjustment of the entity graph involves the reduction of weights for entities that appear in a syntactically embedded structure of the sentence, such as nested prepositional phrases, relative or subjunctive clauses, or participle clauses. Barzilay and Lapata (2008) and Guinaudeau and Strube (2013) do not specify how to assign weights to such entities. In order to consider both original syntactic role and lower coherence status of embedded entities, we reduce the weight by a factor (i.e. 0.75). In nested constructions, multiple reductions for the same entity are possible.

2.2 Experiments

We compare the entity graph with the entity and weight adjustments described above to the original normalized entity graph. We test the two versions on the sentence reordering task, evaluating how well they can distinguish between correct and incorrect sentence order. Discrimination accuracy measures how often the correct sentence order is chosen over the scrambled order. The insertion accuracy measures how often a sentence is re-inserted at the correct position. The positional insertion metric measures how close the guessed position is to the correct one, so higher scores indicate better performance (Elsner and Charniak, 2007).

	Disc. Acc.	**Pos. Ins.**	**Ins. Acc.**
Random	0.5	-0.02	0.04
Entity Graph, G&S			
P_U	0.881	0.124	0.102
P_W	0.879	0.134	0.106
P_{Acc}	0.880	0.142	0.114
Entity Graph, adjusted version			
P_U	0.889**	0.132**	0.105**
P_W	0.887**	0.142**	0.110**
P_{Acc}	0.900**	0.156**	0.119**
P_{Acc} w/o *poss*	0.892*	0.141	0.114
P_{Acc} w/o *red*	0.899**	0.151**	0.119**

Table 1: Sentence reordering, normalized entity graph vs. adjusted version.

We retrieved 1593 documents of a length of 10 to 60 sentences from the German Tüba-DZ news corpus, which provides gold annotations for syntactic parsing and coreference resolution (Telljohann et al., 2003). We test statistical significance with the Wilcoxon signed-rank test (Wilcoxon, 1945) to check whether the adapted version performs significantly[3] better than the original version (cf. Table 1). We compare pairwise for P_U, P_W and P_{Acc} .

In general, the adjustments to the entity graph improve performance. The P_U system is least affected by the change since the possession modifier category only helps when there is otherwise no connecting entity between two sentences. A stronger improvement can be observed for P_W, as the additional category captures many entities that would otherwise not be represented. A similar improvement can also be observed for P_{Acc}. For this system, we also examined the effects of the new category and the weight reduction individually. The bottom two rows of Table 1 indicate that the increase in performance is mainly due to the new category for possession modifiers. The weight reduction only leads to a very minor, although still weakly significant improvement for the discrimination task. In future work, the weight reduction method has to be investigated further and the reduction cases defined more clearly, possibly with regard to text type.

[3] * indicates $p < 0.05$, ** indicates $p < 0.01$

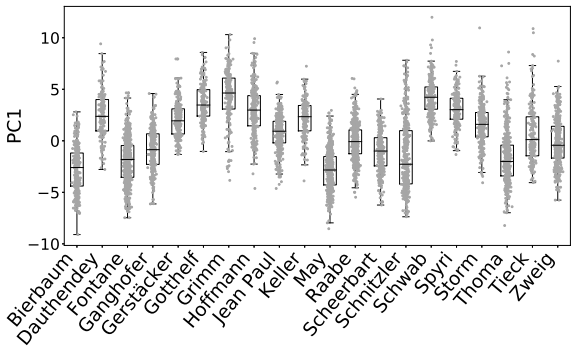

Figure 1: Boxplot of first principal component.

3 Text Analysis Metrics

The average out-degree of the entity graph has proven a valuable measure for local coherence. However, no other entity graph metrics have been examined as to whether they are suitable for representing text information. We computed several graph metrics and used them as features to analyze literary works. For our experiment, we extracted 616 German texts from 20 different authors from the Project Gutenberg, truncating longer texts to 2500 sentences. For syntactic parsing and coreference resolution, we employed the hybrid dependency parser *ParZu* (Sennrich et al., 2009) and the rule-based incremental entity-mention system *CorZu* (Klenner and Tuggener, 2011). We split the texts into chunks of 50 sentences, resulting in 7357 graphs with 50 nodes each.

We rendered circular visualizations of the graphs using *NetworkX* (Hagberg et al., 2008). Nodes represent sentences and are ordered in counterclockwise orientation. Edge widths encode the weights (cf. Figures 2-6). This type of visualization is powerful for analysis of individual samples. However, investigating larger text corpora in this way is very time-consuming and thus is better addressed using automated graph characterization.

Using the graph-tool library (Peixoto, 2014), we computed for each graph the number of edges, diameter, largest component, largest eigenvalue, maximum flow, minimum cut, cut partition, global clustering coefficient and assortativity coefficient. In addition, we computed the mean over all 50 nodes for each of the following node measures: pagerank, katz centrality, eigenvector, author and hub centrality, hits, edge weights, degrees, vertex and edge betweenness, local clustering coefficient and k-core decomposition (Newman, 2010). This sums up to a feature set of 24 features.

3.1 Author Analysis by Graph Metrics

We began our analysis of the extracted graph features by creating boxplots grouped by author for each feature. We found that the distributions of a number of features, including the average degree, show a similar pattern across authors, which corresponds to the pattern of the first principal component (PC) in a PCA (cf. Figure 1). These features are all tightly connected to the total number of edges. They show particularly high values for texts written by Schwab or the Brothers Grimm, which are mostly myths from antiquity and fairy tales, and therefore tend to be shorter and focused on fewer entities within a more compact storyline (cf. Figure 2 for an exemplary graph visualization). The max flow metric supports this observation: while most graphs show a max flow of zero, indicating that there is no uninterrupted path from beginning to end, texts from Brothers Grimm and Schwab present exceptions to this. As a contrast, long narratives such as those written by May show low PC1 values and zero max flow as they contain fewer sentence connecting entities (cf. Figure 3).

This observation prompted us to investigate the relation of PC1 and overall text length. We find that the two are moderately correlated (r=-0.518, p-value=$1.33e^{-43}$), which is surprising given that all samples consisted of exactly 50 sentences. This finding highlights that a relationship exists between local entity graph properties and the total length of a literary work.

We were unable to find a similarly clear-cut explanation for the second PC. However, the third PC, which is exemplified by measures such as assortativity and the mean of weights, appears to be mainly related to sub-graph structure. The highest values for this PC are found for the works of Schnitzler, indicating that these graphs contain tightly connected sub-graphs despite not having a high overall degree (cf. Figure 4). Low PC3 values, on the other hand, are found for texts from Jean Paul, suggesting that nodes are connected more uniformly across the graph instead of forming tightly connected components (cf. Figure 5).

3.2 Author Classification

We used the extracted graph measures described in Section 3 as features for training Support Vector Classifiers with radial basis function (RBF) kernels that predict author and fine-grained genre, respectively. We employed the scikit-learn library for im-

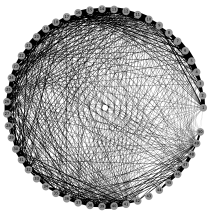

Fig. 2: Grimm.

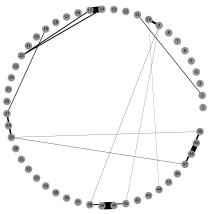

Fig. 3: May.

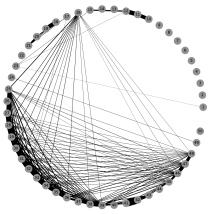

Fig. 4: Schnitzler.

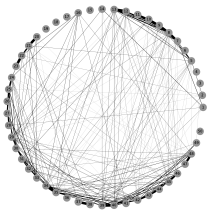

Fig. 5: Jean Paul.

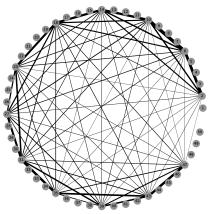

Fig. 6: Tieck.

	Acc.	**Prec.**	**Recall**	**F**
Random	0.182	0.018	0.1	0.030
EG P_U	0.454	0.450	0.431	0.433
EG P_W	0.478	0.473	0.457	0.460
EG P_{Acc}	0.470	0.464	0.450	0.451
Syntactic	0.740	0.739	0.730	0.731
+ EG P_{Acc}	0.787	0.779	0.773	0.772

Table 2: Author classification, entity graph features, syntactic features, and combined feature set.

	Acc.	**Prec.**	**Recall**	**F**
Random	0.153	0.015	0.1	0.026
EG P_U	0.407	0.271	0.302	0.267
EG P_W	0.412	0.321	0.309	0.288
EG P_{Acc}	0.416	0.328	0.319	0.298
Syntactic	0.572	0.555	0.552	0.541
+ EG P_{Acc}	0.615	0.605	0.561	0.560

Table 3: Genre classification, entity graph features, syntactic features, and combined feature set.

plementation (Pedregosa et al., 2011). We compare the system with features derived from the entity graph to a system with 31 purely syntactic features, an adaption from the lexico-syntactic system described in Feng (2015, p. 113). We discarded the lexical features since it has been shown that they often overfit the training data (Moosavi and Strube, 2017) and they are considered a less reliable authorial fingerprint than syntactic features as they are content-dependent (Stamatatos, 2009). In order to evaluate whether the entity graph features contain information that is not encoded in the syntactic features, we also evaluate the combined system with both entity graph and syntactic features.

For author classification, we extracted 5081 samples of 50 sentences each from 10 different authors (Jean Paul, Schnitzler, May, Hoffmann, Bierbaum, Zweig, Storm, Thoma, Fontane, Raabe). We report accuracy, macro average precision, recall and F1 score in Table 2. The 24 entity graph features alone classify roughly 47% of the samples correctly. The combined system with both entity graph and syntactic features significantly outperforms the syntactic feature system, implying that the entity graph features capture author information that is not represented by the syntactic features. We conclude that although the entity graph features on their own are not sufficient for author attribution, they contain valuable information that may improve other systems.

3.3 Genre Classification

For the genre classification task (Kessler et al., 1997; Webber, 2009), we extracted 1953 samples from 30 authors, each labeled with one genre (comedy, drama, fairy tale, fiction, legend, narrative, novel, novelette, preface, or tragedy). In general, the scores (shown in Table 3) are lower than for author classification, most likely because of the smaller dataset and the overlapping genres. Fiction, for instance, has a wide scope that may encompass some of the more fine-grained genres. Combining the entity graph features with the syntactic ones improves performance, albeit not significantly. Given this ambiguous first result, further work will be required to determine the true extent of genre information encoded in entity graph features.

4 Conclusions

We have improved the entity graph by adding a category for possession modifiers and reducing weights for syntactically embedded entities. Beyond its use as a local coherence measure, we have shown that visualization and graph metrics analysis of the entity graph can reveal meaningful properties of text, as exemplified by the literary work analysis and the contribution of new information to author classification. We expect that this approach will be suitable for many applications both in NLP and digital humanities, ranging from more precise coherence measures to text structure analysis.

Acknowledgments

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Noam Chomsky. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Micha Elsner and Eugene Charniak. 2007. A generative discourse-new model for text coherence. Technical report, Technical Report CS-07-04, Brown University.

Wei Vanessa Feng. 2015. *RST-style discourse parsing and its applications in discourse analysis*. Ph.D. thesis, University of Toronto (Canada).

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 93–103.

Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

Brett Kessler, Geoffrey Numberg, and Hinrich Schütze. 1997. Automatic detection of text genre. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 32–38. Association for Computational Linguistics.

Manfred Klenner and Don Tuggener. 2011. An incremental entity-mention model for coreference resolution with restrictive antecedent accessibility. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 178–185.

Mohsen Mesgar and Michael Strube. 2014. Normalized entity graph for computing local coherence. In *Proceedings of TextGraphs-9: The Workshop on Graph-based Methods for Natural Language Processing*, pages 1–5.

Mohsen Mesgar and Michael Strube. 2015. Graph-based coherence modeling for assessing readability. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 309–318.

Nafise Sadat Moosavi and Michael Strube. 2017. Lexical features in coreference resolution: To be used with caution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 14–19.

Mark E.J. Newman. 2010. *Networks: An Introduction*. Oxford University Press, New York, N.Y.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Tiago P. Peixoto. 2014. The graph-tool Python library. `https://figshare.com/articles/graph_tool/1164194`. Online. Last accessed May 31, 2018.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for German. *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 115–124.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Sally Stoddard. 1991. *Text and Texture: Patterns of Cohesion*. Ablex, Norwood, N.J.

Heike Telljohann, Erhard W. Hinrichs, and Sandra Kübler. 2003. Stylebook for the Tübingen treebank of written German (TüBa-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany.

Bonnie Webber. 2009. Genre distinctions for discourse in the Penn TreeBank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 674–682. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.