LLM-based German discourse parsing, leveraging structured outputs

Philipp Sandhaas
University of Potsdam / Student-ID: 814801
philipp.sandhaas@uni-potsdam.de

Abstract

This project explores the ability of Large Language Models (LLMs) to perform German Rhetorical Structure Theory (RST) discourse parsing using structured outputs without fine-tuning. A zero-shot parser based on GPT-4.1 with grammar-constrained decoding is evaluated against two neural baseline parsers (DMRST and DPLP). Experiments manipulate the presence of linebreaks to test the impact of surface cues on segmentation. Results show near-perfect segmentation performance (F1=0.948) with linebreaks but significantly lower performance on structural tasks (span F1 $\leq$ 0.132). The findings confirm that current LLMs rely heavily on non-lexical surface features rather than linguistic understanding, highlighting the continued need for specialized neural models in complex discourse parsing for German.

1 Introduction

1.1 Motivation and Background

Discourse parsing identifies the hierarchical structure underlying coherent text, critical for applications such as summarization and information extraction. Automatically parsing discourse remains challenging however, especially for German, where resources and evaluations lag behind English.

Advances in LLMs offer potential for zero-shot discourse parsing, which might be facilitated by structured output generation methods like grammar-constrained decoding (Geng et al., 2023). However, discourse parsing requires recursive reasoning and understanding of semantic relations, complicating the application of LLMs without task-specific training.

1.2 Research Questions

This project investigates the zero-shot performance of GPT-4.1 for German RST parsing using structured outputs. It explores (1) whether LLMs can segment text accurately without fine-tuning, (2) their ability to construct hierarchical discourse structures, (3) dependence on surface formatting cues, and (4) patterns in relation assignment compared to neural parsers.

2 Related Work

2.1 Rhetorical Structure Theory and Discourse Parsing

RST (Mann and Thompson, 1988) formalizes discourse coherence via hierarchical trees with EDUs and rhetorical relations. Computational parsing evolved from rule-based methods to neural architectures (Marcu, 2000; Ji and Eisenstein, 2014) and cross-lingual approaches have since demonstrated transfer potential across languages (Braud et al., 2017; Liu et al., 2021).

2.2 German Discourse Parsing Resources

German discourse parsing has been limited by scarce annotated corpora–mainly the Potsdam Commentary Corpus (Stede, 2004)–although recent corpora like APA (Hewett, 2023) and PARADISE (Seemann et al., 2023) add genre diversity. Improved parsers and harmonized annotations have advanced German parsing (Shahmohammadi and Stede, 2024), though challenges remain due to dataset sizes.

2.3 Large Language Models and Structured Output

LLMs have enabled zero-shot NLP but struggle with structured tasks requiring constrained outputs. Grammar-constrained decoding addresses structured generation challenges (Geng et al., 2023). Little works exist on LLM-based discourse parsing but studies indicate that surface features strongly influence model output, suggesting a need for fine-tuning (Maekawa et al., 2024).

3 Data and Annotation Schema

3.1 Corpora

Nine documents were selected: three each from the APA (Hewett, 2023), PARADISE (Seemann et al., 2023), and PCC (Stede, 2004) corpora. APA and PCC contain German newspaper texts—news reports and commentaries respectively—while PARADISE consists of business and pop-science blog posts, limited here to blog texts only. Document identifiers are retained where available, except in APA where none are provided.

Document	# Tokens	Distinct Relations	# Segments
apa_1	1457	10	21
apa_2	1174	8	16
apa_3	1261	8	16
CRE210_Blog	808	5	8
CRE219_Blog	799	6	8
FG041_Blog	842	7	8
maz-10374	535	8	12
maz-13946	471	7	11
maz-6918	211	6	11
Average	*840*	*7.2*	*12.3*

Table 1: Document statistics (excluding disjunct segments)

Documents were chosen for varied lengths and discourse relation diversity, under the assumption that longer texts with larger rhetorical trees more strictly challenge LLM parsing capabilities (Table 1).

3.2 Relation Set, Annotation, and Segmentation Guidelines

Following Shahmohammadi and Stede (2024), 92 fine-grained relations from the baselines were mapped to 15 coarse mononuclear and 3 multinuclear labels (see Appendix A). Eight multinuclear fine-grained relations were consolidated into mononuclear classes for consistency.

Annotation guidelines for the LLM primarily adopt the German-language schemas of Stede (2016) for consistency and structure. Where unavailable, English guidelines from Carlson and Marcu (2001) were translated and adapted. Segmentation instructions, including procedural explanations and examples, also derive from Stede (2016) (see Appendix D).

4 Methods

4.1 Parsing Approaches

4.1.1 Baseline Parsers

Two document-level German discourse parsers serve as baselines: the DMRST parser (Liu et al., 2021) and a fine-tuned DPLP parser (Shahmohammadi and Stede, 2024) based on Ji and Eisenstein (2014). Both operate by segmenting raw text into EDUs before constructing rhetorical structure trees.

The pretrained DMRST outputs boundary indices and custom top-down constituency trees (see Figure 6 in Appendix B for an example), converted here to normalized `.rs3` files via a Python API and Docker container[1]. An analogous Docker container[2] is used for the DPLP parser.

4.1.2 LLM-based Parser

An LLM agent orchestrated using LangGraph [3] parses German texts using grammar-constrained structured outputs (Geng et al., 2023)[4]. This choice is made to ensure that the LLM's generations adhere to the formal requirements for constructing well-formed rhetorical structure trees. The pipeline comprises optional segmentation (except in Experiment 3), root node creation, and a recursive parsing loop with steps for splitting spans, assigning nuclearity, and labeling relations, culminating in `.rs3` output. The structured workflow illustrated in Figure 1 is closely inspired by the top-down approach described in Maekawa et al. (2024) and consists of the following steps:

1. **Segmentation** (optional): given an input document and segmentation guidelines as system prompt, an ordered, non-overlapping, sequentially indexed list of EDUs is generated. (This stage is ommitted in experiment 3 and instead, the pre-segmented text is passed into the next stage.)

2. Create root (programmatic): given an ordered list of EDUs, a root node spanning EDUs 1 :

[1] https://hub.docker.com/r/psandhaas/dmrst-parser

[2] https://hub.docker.com/r/mohamadisara20/dplp-env

[3] https://langchain-ai.github.io/langgraph/reference/

[4] see also https://openai.com/index/introducing-structured-outputs-in-the-api/ & https://github.com/anthropics/anthropic-sdk-python?tab=readme-ov-file#tool-helpers

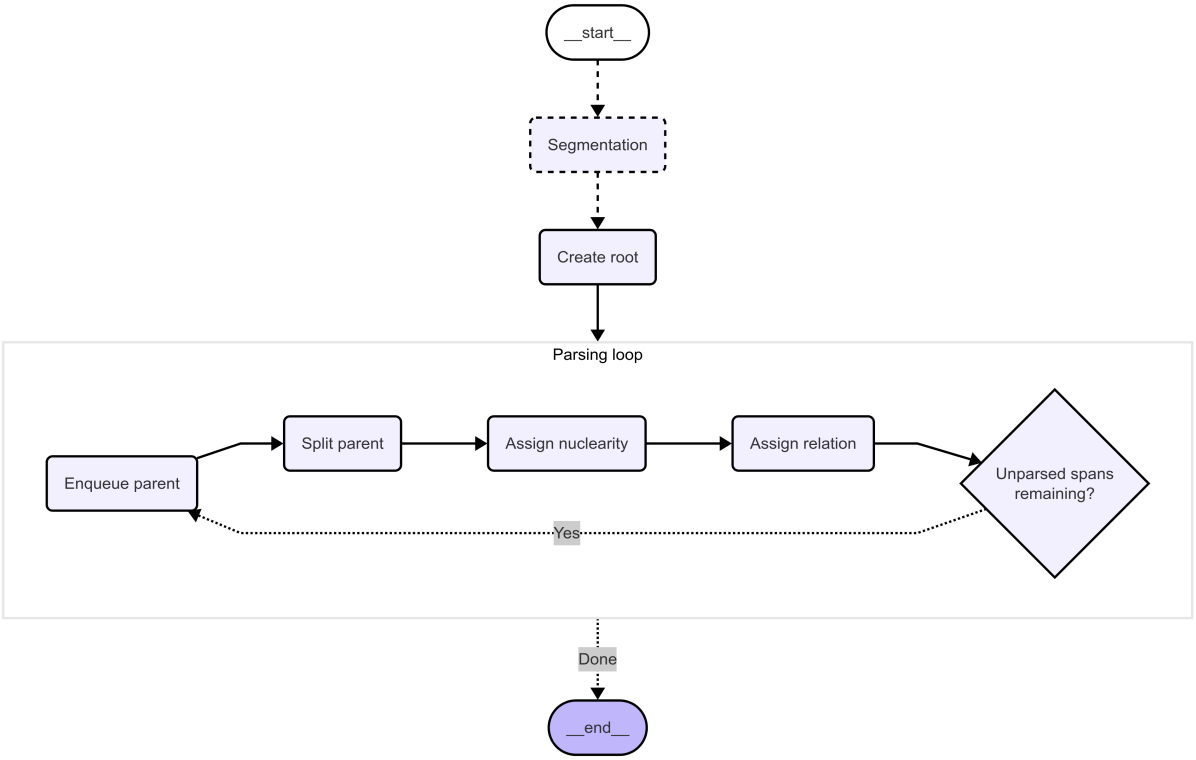


Figure 1: LangGraph workflow for top-down construction of a rhetorical structure tree

n (where n is the total number of segments) is created and added to the queue.

3. **Parsing loop**: applied recursively until all spans have been divided into EDUs (i.e. nodes with spans $i : j$, where $i = j$)

   3.1 Enqueue parent (programmatic): select the largest unprocessed span, assign a unique ID and create a new node.

   3.2 **Split parent**: given the corresponding text segments of an unprocessed node that spans EDUs $i : j$, an index k is determined such that $i \leq k \leq j$, in order to split the parent span into two unprocessed sibling nodes.

   3.3 **Assign nuclearity**: given the text segments of two unprocessed sibling nodes (i.e. left & right children of the current parent node), the nuclearity of both siblings is determined. Based on the assigned nuclearity, parent IDs of the child nodes are set programmatically[5].

   3.4 **Assign relation**: given the corresponding nuclearity, EDU-spans & text segments of two sibling nodes and a set of defined discourse relations as system prompt, a relation label is assigned to the relevant child node(s)[6]. The relation-type of the parent node is set programmatically, depending on whether a mono- or multinuclear relation was assigned to the child nodes. To limit the number of prompt-tokens[7] while simultaneously keeping the instructions manageable for the LLM, the output schema for this step only includes the coarse relation labels (see Listing 4 in Appendix D).

   3.5 Update nodes & queue (programmatic): parent node and leaf nodes (i.e. children with spans $i : j$ where $i = j$) are marked as processed, non-terminal child nodes are added to the queue.

[5] In case of mononuclear pairs, the parent node serves only as the parent of the Nucleus-child, which is itself the parent of the Satellite-child (effectively creating a unary subtree). In the multinuclear case, the parent node's ID is used for both children.

[6] In case of mononuclear relations, the generated relation label is assigned to the Satellite-child and a `"span"` label is assigned to the Nucleus-child. Otherwise, the same label is assigned to both child nodes.

[7] Totaling almost 10k, prompt-tokens outnumber the longest input document roughly by a factor of 10.

4. Convert outputs (programmatic): as a final step, processed outputs are converted into `.rs3`-XML-format.

The workflow was tested using OpenAI's GPT-4.1, GPT-4o, GPT-o4mini, and Anthropic's Claude Sonnet 4 models but only GPT-4.1 reliably followed output schemas.

4.2 Data Conversion and Tree Representation

To evaluate and visualize, all parser outputs were converted to `.rs3` format using a recursive `Node` class representing binary RST trees. Nodes correspond either to single EDUs (leaves) or spans (non-terminals), with nuclearity determining node type (mononuclear or multinuclear). Relations are encoded as node attributes following RST conventions.

The decision to represent trees as binary while lacking an explicit binarization procedure, induces incompatibilities with authentic n-ary structures present in some documents (e.g., multiple nuclei or satellites), resulting in malformed `.rs3` files and occasional failures in RST-Web rendering. This limitation uniformly affects gold and predicted trees, preserving evaluation comparability.

4.3 Experimental Settings

Segmentation and tree-construction tasks were evaluated across three experiments differing in surface cue availability:

- **Experiment 1:** Non-lexical segmentation cues (linebreaks) removed by replacing with whitespace, supplying only lexical cues.
- **Experiment 2:** Original text with linebreaks preserved, allowing implicit segmentation cues.
- **Experiment 3:** Pre-segmented input passed directly to tree construction, bypassing segmentation.

Tree construction followed the same LLM workflow throughout, as variations proved challenging.

4.4 Evaluation Protocol

To compare outputs differing in segmentation, all trees were standardized into connected `Nodes` aligned through a canonicalized tokenization process addressing punctuation, hyphenation, and quotations. Token-span alignment enabled fair scoring of:

- **EDUs**: Leaf nodes aligned to gold segments assess segmentation accuracy.
- **Spans**: Non-terminal nodes measure structural similarity.
- **Nuclearity**: Agreement on relative importance in siblings.
- **Relations**: Agreement on rhetorical role labels.

Fine-grained baseline relations were mapped to the LLM's coarse label set for uniform evaluation.

5 Results

5.1 Quantitative Results

		EDUs			Spans		
Parser	Exp.	Rec.	Prec.	F1	Rec.	Prec.	F1
DMRST		0.849	0.860	0.852	0.756	0.712	**0.214**
DPLP		0.541	0.545	0.542	0.206	0.222	0.213
LLM	1	0.679	0.720	0.690	0.102	0.085	0.087
LLM	2	0.962	0.937	**0.948**	0.153	0.122	0.132
LLM	3	(1.0)	(1.0)	(1.0)	0.133	0.117	0.121

		Nuclearity			Relations		
Parser	Exp.	Rec.	Prec.	F1	Rec.	Prec.	F1
DMRST		0.231	0.222	0.507	0.382	0.368	**0.372**
DPLP		0.542	0.523	**0.528**	0.334	0.329	0.329
LLM	1	0.390	0.413	0.388	0.141	0.147	0.138
LLM	2	0.510	0.463	0.483	0.188	0.172	0.178
LLM	3	0.467	0.449	0.456	0.192	0.190	0.190

Table 2: Comparison of parsing performances

Table 2 shows labeled recall, precision and micro-averaged F1-scores computed for EDUs, Spans, Nuclearity and Relations across all documents (see Tables 4 through 8 in Appendix C for parser-level performance).

The results reveal a clear performance hierarchy across parsing tasks. The DMRST parser demonstrates the strongest overall performance with balanced results across all metrics, achieving the highest span F1-score (0.214) and relation F1-score (0.372). The DPLP parser excels specifically at nuclearity prediction (F1=0.528) despite lower segmentation performance (F1=0.542).

The LLM experiments confirm the hypothesis regarding surface feature dependency. In Experiment 2, with linebreaks preserved, the LLM achieved near-perfect segmentation agreement (F1=0.948), demonstrating effective utilization of non-lexical segmentation cues. This contrasts sharply with Experiment 1 (F1=0.690 without linebreaks), providing strong evidence for the LLM approaches

heavy reliance on surface formatting rather than deep linguistic understanding. The pre-segmented condition (Experiment 3) yielded perfect segmentation by design but showed only marginal improvements in relation assignment while fairing marginally worse in Nuclearity & Spans assignment, highlighting fundamental limitations in hierarchical discourse construction.

Notably, segmentation errors propagate through the parsing pipeline (Da Cunha and Iruskieta (2010), van der Vliet (2010)). However, the DPLP parser's relatively low segmentation agreement (0.542) does not necessarily translate to proportionally lower performance in span construction, nuclearity assignment, or relation classification, suggesting some robustness to segmentation errors at higher structural levels.

5.2 Qualitative Results

The confusion matrices in Figure 3 reveal systematic biases in LLM relation assignment. Only the DMRST (3a) and DPLP parsers (3b) demonstrate balanced assignment of non-JOINT labels aligned with gold annotations. The LLM shows a strong preference for high-frequency relations, particularly JOINT, which accounts for 23% of gold labels but 68%, 55%, and 63% of LLM predictions in Experiments 1, 2, and 3 respectively.

This distributional bias is complemented by systematic under-assignment of the SPAN label, which represents 39% of gold annotations but only 14%, 15%, and 16% of LLM predictions across experiments. This pattern reflects the LLM's tendency to favor binary multinuclear relations over hierarchical span structures, suggesting reliance on distributional shortcuts rather than semantic understanding of discourse relations.

The LLM achieved limited success in predicting specific discourse relations across all experiments. In Experiments 2 and 3, it correctly identified single instances each of ATTRIBUTION and BACKGROUND relations, with parsing trees available in Appendix B. However, the overall pattern indicates that the model defaults to high-frequency relation types rather than engaging in systematic discourse analysis.

5.3 Error Analysis and Observations

Manual inspection of parsing outputs revealed several systematic issues that may deflate computed performance metrics. An implicit assumption made by the `Node` class is that segments in `.rs3`-files maintain sequential numbering, which does not always hold (e.g., document `maz-13946`). In all cases affected by this issue, this caused misalignments between token-spans and occasionally even EDUs to be dropped entirely (see Table 11 vis-a-vis Figure 7 in Appendix C for an example). It also led to malformed `.rs3`-files after conversion, which in turn caused the rendering of parsed trees to fail in RSTWeb (Zeldes, 2016).

The way in which the LLM is tasked with assigning a relation label, might further skew the results. Specifically, in step 3.4 **Assign relation** of the LangGraph workflow, the LLM is presented with the nuclearity, the EDU-span and the concatenated text segments (corresponding to the EDU-span) of a left and a right child. Although this approach doesn't lead to errors per se, it fails to capture the structural context of the partial tree that has been constructed so far, thus omitting potentially valuable information.

Finally, while the evaluation methodology's binary tree representation constraint enables consistent comparison across all parsers, it fails to capture the full complexity of authentic discourse structures, particularly multinuclear relations with more than two nuclei. This limitation affects all systems equally.

6 Discussion

6.1 Interpretation of Results

The findings of this project provide insights into the capabilities and limitations of current Large Language Models for German discourse parsing. The LLM-based approach achieved mixed results, demonstrating both promising capabilities and fundamental constraints that illuminate the current state of zero-shot discourse analysis.

The most significant finding concerns the LLM's heavy reliance on surface-level cues for segmentation. The dramatic performance difference between Experiment 1 (F1=0.690 without linebreaks) and Experiment 2 (F1=0.948 with linebreaks) confirms the hypothesis that current LLMs leverage non-lexical formatting information rather than engaging in deep linguistic analysis. This aligns with recent findings by Maekawa et al. (2024), who reported similar surface-feature dependency in English RST parsing with LLMs, which these findings seem to extend to German.

Compared to the baseline parsers, the LLM demonstrates competitive segmentation perfor-

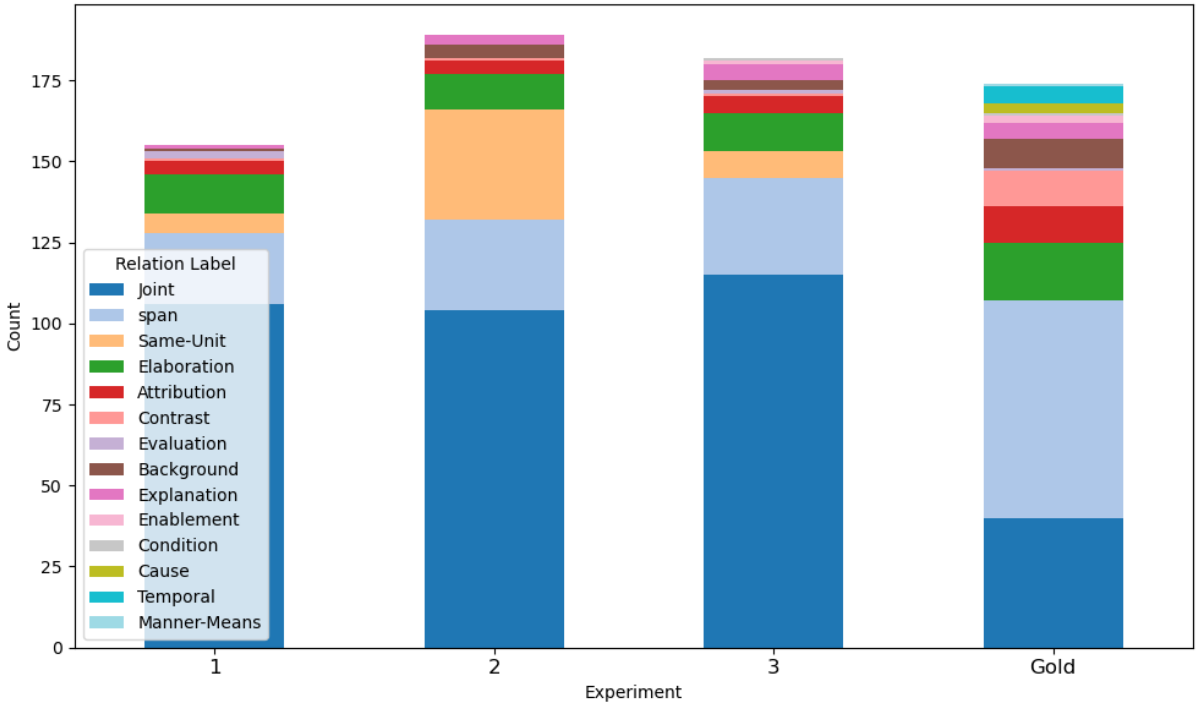


Figure 2: Distribution of assigned vs. gold relation labels across documents in LLM-experiments

mance when linebreaks are preserved but falls significantly short in structural tasks. The DMRST parser's balanced performance (span F1: 0.214, relation F1: 0.372) and the DPLP parser's strong nuclearity prediction (F1: 0.528) highlight the continued relevance of specialized neural architectures for discourse parsing tasks.

Regarding structural parsing tasks, the consistently low performance across all LLM experiments (span F1 $\leq$ 0.132) reveals fundamental limitations in hierarchical discourse construction. Unlike established neural parsers that achieve modest but consistent structural accuracy (DMRST: 0.214, DPLP: 0.213), the LLM struggles with the recursive nature of rhetorical tree building. This finding corroborates recent work by Geng et al. (2023), which suggests that grammar-constrained decoding alone may be insufficient for complex structural tasks without specialized training.

6.2 Limitations

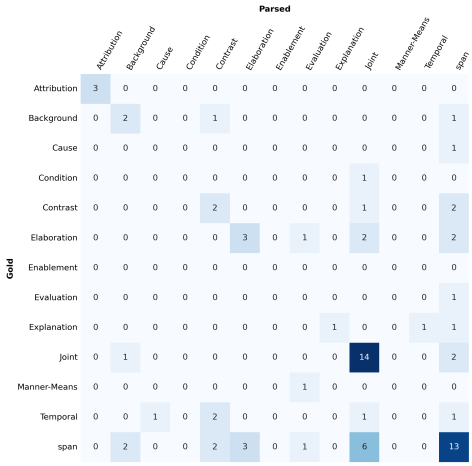
Several limitations of this project must be acknowledged. First, the sample size of nine documents, while sufficient for proof-of-concept evaluation, limits the generalizability of findings across different text types and domains. The documents were selected based on length and relation diversity, potentially introducing selection bias that may not reflect real-world discourse parsing scenarios.

The binary tree representation constraint imposed by the evaluation framework fails to capture authentic n-ary discourse structures. While this limitation affects all parsers equally, it underestimates the true complexity of German discourse patterns, particularly for multinuclear relations with multiple nuclei.

Technical implementation issues further limit the reliability of the above findings. The sequential EDU numbering assumption in the evaluation protocol led to segment misalignments in several documents (e.g., `maz-13946`), potentially deflating performance metrics artificially. Additionally, the LLM's workflow provides limited structural context during relation assignment, as each relation decision is made in isolation rather than considering the partially constructed tree structure.

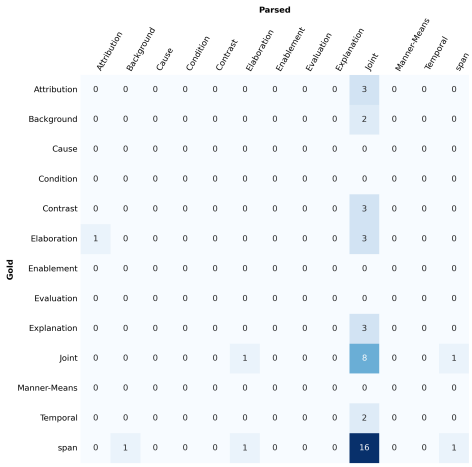
The choice of GPT-4.1 as the sole LLM introduces model-specific limitations. While other models (GPT-4o, Claude Sonnet 4) were tested, only GPT-4.1 successfully followed the structured output schema, limiting the ability to generalize findings across different LLM architectures.

Finally, the relation mappings from English may not transfer directly to German. The reliance on manually translated guidelines from English RST frameworks to achieve a shared label set, introduces potential inconsistencies that could affect the

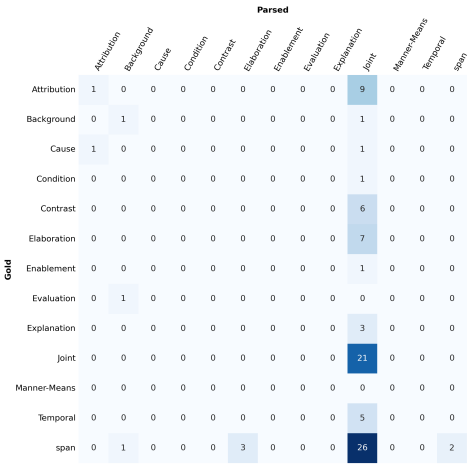


(a) DMRST

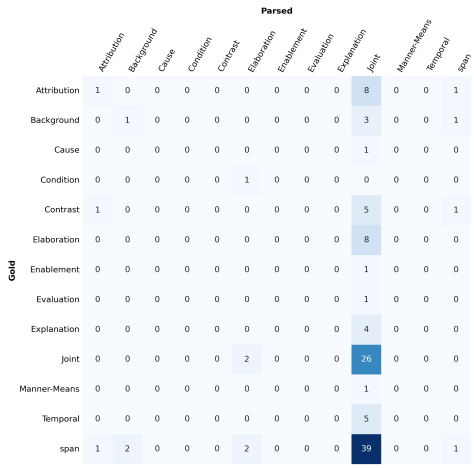
(b) DPLP



(c) LLM: Experiment 1



(d) LLM: Experiment 2



(e) LLM: Experiment 3

Figure 3: Confusion matrices of (coarse) relation labels

quality of the annotation prompt.

6.3 Implications and Future Work

The demonstrated surface-feature dependency suggests that improving LLM-based discourse parsing requires moving beyond localized prompt engineering toward more sophisticated approaches that incorporate semantic & structural representations.

Methodological improvements could address several identified limitations. Future work should

explore multi-step reasoning approaches that provide structural context during relation assignment.

Hybrid approaches combining LLMs with specialized neural architectures show particular promise. The LLM's strong segmentation capabilities (when linebreaks are preserved) could be combined with traditional neural parsers' structural parsing strengths, potentially achieving better overall performance than either approach alone.

Evaluation methodology improvements should prioritize more realistic tree representations that accommodate n-ary discourse structures, while being robust enough to handle differences in segmentation and tokenization. The evaluation procedure proposed by Morey et al. (2017) seems like a promising alternative.

7 Conclusion

This project investigated the capabilities of Large Language Models for German rhetorical structure parsing using a structured output approach without fine-tuning. Through controlled experiments comparing segmentation performance under different surface cue conditions, a heavy reliance on non-lexical formatting information rather than deep linguistic understanding for discourse segmentation could be demonstrated.

The key findings reveal a clear performance hierarchy: while LLMs can achieve near-perfect segmentation when linebreaks are preserved (F1=0.948), they consistently underperform established neural parsers on structural tasks (span F1 $\leq$ 0.132 vs. 0.213-0.214 for baselines). This pattern, combined with systematic biases toward high-frequency discourse relations, suggests that the described LLM approach engages in distributional pattern matching rather than genuine discourse analysis.

While LLMs show promise for certain subtasks like segmentation, the complex hierarchical reasoning required for rhetorical structure construction remains challenging. Future improvements will likely require hybrid approaches that combine the complementary strengths of LLMs and specialized neural architectures.

References

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual rst discourse parsing. *arXiv preprint arXiv:1701.02946*.

Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, 54(2001):56.

Iria Da Cunha and Mikel Iruskieta. 2010. Comparing rhetorical structures in different languages: The influence of translation strategies. *Discourse Studies*, 12(5):563–598.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. *arXiv preprint arXiv:2305.13971*.

Freya Hewett. 2023. Apa-rst: A text simplification corpus with rst annotations. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 173–179.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 13–24.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021. DMRST: A joint framework for document-level multilingual RST discourse segmentation and parsing. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 154–164, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Aru Maekawa, Tsutomu Hirao, Hidetaka Kamigaito, and Manabu Okumura. 2024. Can we obtain significant success in rst discourse parsing by using large language models? *arXiv preprint arXiv:2403.05065*.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT press.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2017. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324, Copenhagen, Denmark. Association for Computational Linguistics.

Hannah J Seemann, Sara Shahmohammadi, Tatjana Scheffler, and Manfred Stede. 2023. Building a parallel discourse-annotated multimedia corpus. *14–15 September 2023, University of Mannheim, Germany*, 8(3):17.

Sara Shahmohammadi and Manfred Stede. 2024. Discourse parsing for german with new rst corpora. In *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*, pages 65–74.

Manfred Stede. 2004. The potsdam commentary corpus. In *Proceedings of the Workshop on Discourse Annotation*, pages 96–102.

Manfred Stede. 2016. *Handbuch Textannotation: Potsdamer Kommentarkorpus 2.0*, volume 8. Universitätsverlag Potsdam.

Nynke van der Vliet. 2010. Inter annotator agreement in discourse analysis.

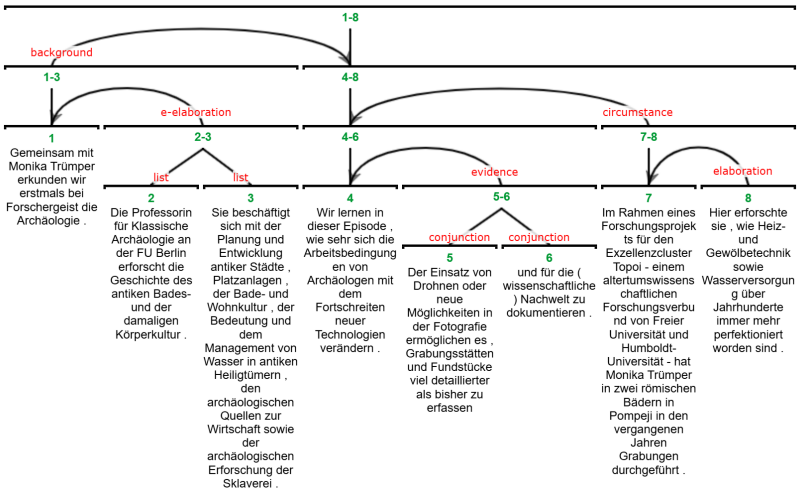
Amir Zeldes. 2016. rstweb-a browser-based annotation interface for rhetorical structure theory and discourse relations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–5.

A Relation set

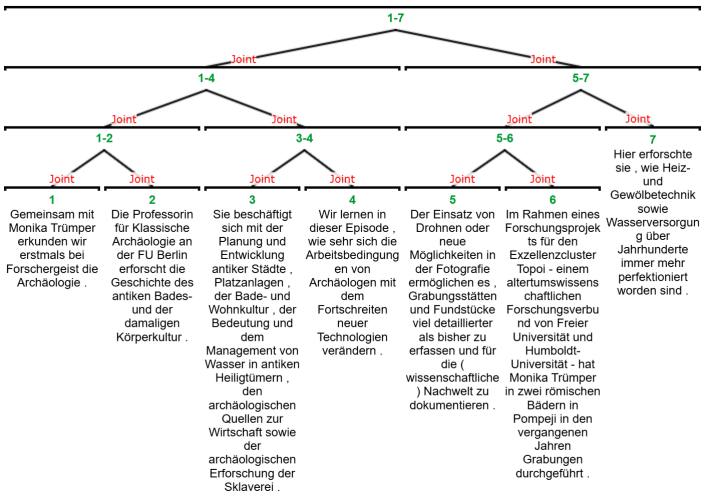
Coarse label	Fine labels
Mononuclear relations	(†*fine-grained labels of multinuc. relations that were mapped to mononuc. coarse labels*)
ATTRIBUTION	Attribution, Attribution-negative
BACKGROUND	Background, Circumstance, Preparation
CAUSE	Cause, Cause-result, Causemult†, Consequence, Nonvolitional-cause, Nonvolitional-result, Result, Volitional-cause, Volitional-result
COMPARISON	Analogy, Comparison, Comparisonmult†, Preference, Proportion
CONDITION	Condition, Conditionmult†, Contingency, Hypothetical, Otherwise, Unconditional, Unless
CONTRAST	Antithesis, Concession, Contrast, Contrastmult†
ELABORATION	Definition, E-Elaboration, Elaboration, Elaboration-additional, Elaboration-general-specific, Elaboration-object-attribute, Elaboration-part-whole, Elaboration-process-step, Elaboration-set-member, Example, Parenthetical
ENABLEMENT	Enablement, Purpose
EVALUATION	Comment, Conclusion, Evaluation, Evaluationmult†, Interpretation
EXPLANATION	Evidence, Explanation, Explanation-argumentative, Explanationmult†, Justify, Motivation, Reason
MANNER-MEANS	Manner, Manner-Means, Means
SUMMARY	Restatement, Restatement-mn†, Summary
TEXTUALORGANIZATION	TextualOrganization†
TOPIC-CHANGE	Topic-drift, Topic-shift, Topic-Change, Topichangemult†, Topidriftmult†
TOPIC-COMMENT	Comment-topic, Problem-solution, Question, Question-answer, Rhetorical-question, Solutionhood, Statement-response, Topic-Comment, Topicommentmult†
Multinuclear relations	
JOINT	Conjunction, Disjunction, Joint, List
SAME-UNIT	Same-Unit
TEMPORAL	Inverted-sequence, Sequence, Temporal, Temporal-after, Temporal-before, Temporal-same-time

Table 3: Mapping of fine- to coarse-grained relations following Braud et al. (2017) & Carlson and Marcu (2001)

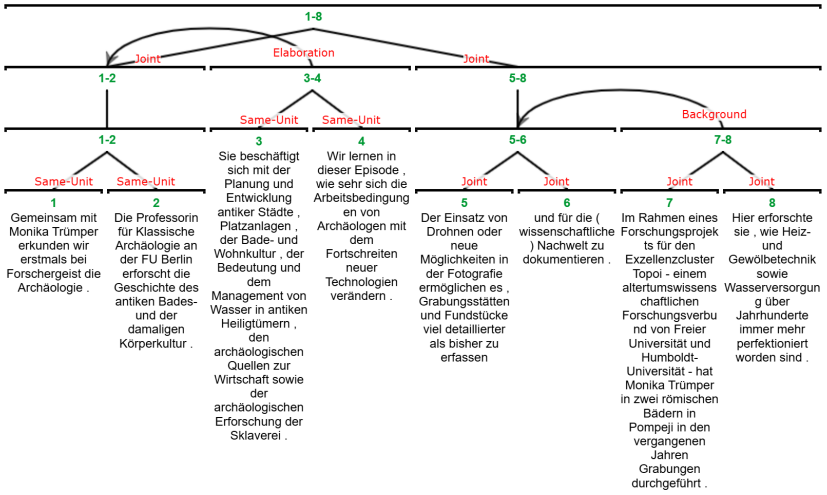
B Example outputs



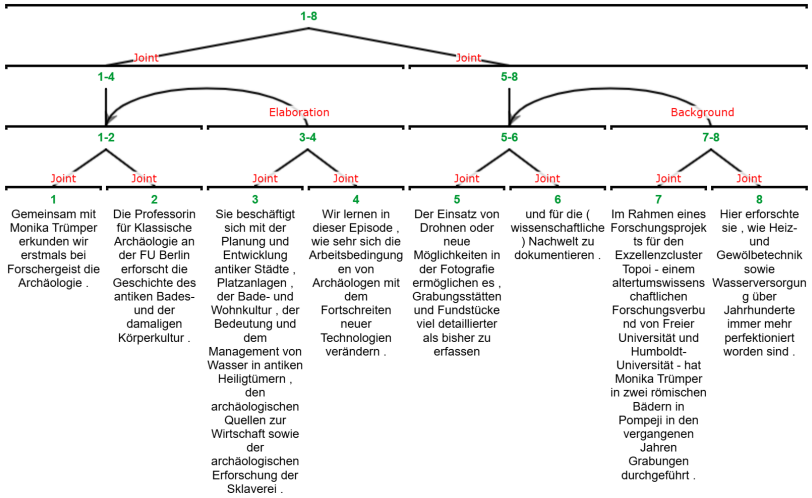
(a) Gold



(b) Experiment 1

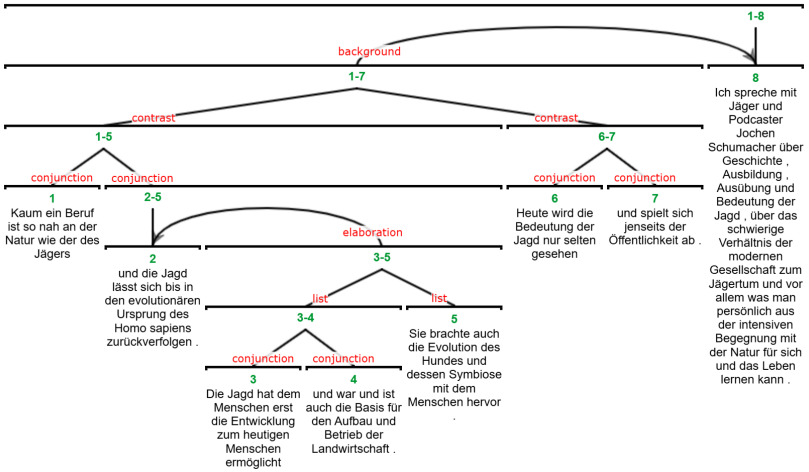


(c) Experiment 2

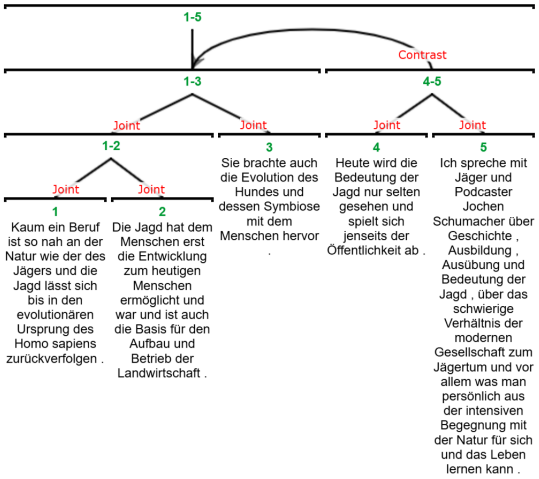


(d) Experiment 3

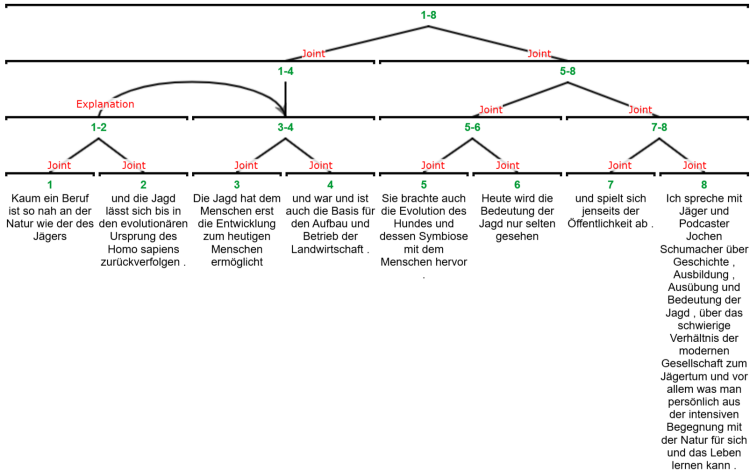
Figure 4: LLM outputs for `FG041_Blog`



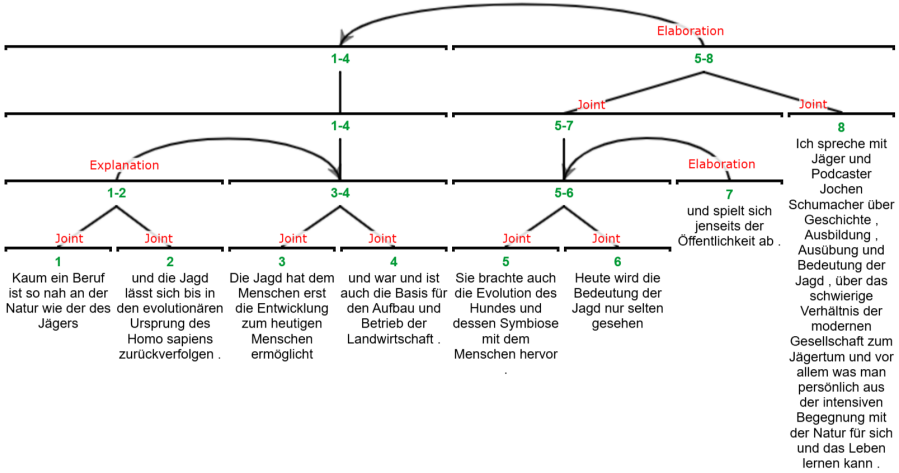
(a) Gold



(b) Experiment 1



(c) Experiment 2



(d) Experiment 3

Figure 5: LLM outputs for `CRE210_Blog`

Although the report, || which has released || before the stock market opened, || didn't trigger the 190.58 point drop in the Dow Jones Industrial Average, || analysts said || it did play a role in the market's decline.||

(a) Example input, annotated with EDU breaks

```
[4, 7, 13, 26, 28, 37]*
```

*Indices refer to boundary tokens

(b) EDU Breaks

```
(1*:Satellite=Contrast:4,5:Nucleus=span:6)
(1:Nucleus=Same-Unit:3,4:Nucleus=Same-Unit:4)
(5:Satellite=Attribution:5,6:Nucleus=span:6)
(1:Satellite=span:1,2:Nucleus=Elaboration:3)
(2:Nucleus=span:2,3:Satellite=Temporal:3)
```

*Indices refer to (inclusive) spans of EDUs

(c) Rhetorical tree in top-down constituency format

Figure 6: Example output of the DMRST-parser

C Supplemental material

	EDUs			**Spans**			**Nuclearity**			**Relations**		
Document	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
apa_1	0.762	0.889	0.821	0.214	0.231	0.222	0.400	0.452	0.424	0.314	0.355	0.333
apa_2	0.750	0.857	0.800	0.500	0.556	0.526	0.538	0.609	0.571	0.385	0.435	0.408
apa_3	0.875	0.737	0.800	0.667	0.571	0.615	0.750	0.636	0.689	0.571	0.485	0.525
blogposts_CRE210_Blog	1.000	1.000	**1.000**	0.667	0.800	**0.727**	0.857	0.923	**0.889**	0.714	0.769	**0.741**
blogposts_CRE219_Blog	0.500	0.400	0.444	0.500	0.333	0.400	0.417	0.312	0.357	0.333	0.250	0.286
blogposts_FG041_Blog	0.750	0.857	0.800	0.400	0.500	0.444	0.385	0.455	0.417	0.308	0.364	0.333
pcc_maz-10374	1.000	1.000	**1.000**	0.250	0.111	0.154	0.500	0.381	0.432	0.062	0.048	0.054
pcc_maz-13946	1.000	1.000	**1.000**	0.000	0.000	0.000	0.571	0.421	0.485	0.500	0.368	0.424
pcc_maz-6918	1.000	1.000	**1.000**	0.600	0.500	0.545	0.312	0.294	0.303	0.250	0.235	0.242
Average	0.849	0.860	0.852	0.422	0.400	0.404	0.526	0.498	0.507	0.382	0.368	0.372

Table 4: Performance of DMRST parser

	EDUs			**Spans**			**Nuclearity**			**Relations**		
Document	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
apa_1	0.048	0.062	0.054	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
apa_2	0.250	0.333	0.286	0.000	0.000	0.000	0.115	0.143	0.128	0.038	0.048	0.043
apa_3	0.688	0.733	0.710	0.250	0.250	0.250	0.357	0.370	0.364	0.179	0.185	0.182
blogposts_CRE210_Blog	1.000	1.000	**1.000**	1.000	1.000	**1.000**	1.000	1.000	**1.000**	1.000	1.000	**1.000**
blogposts_CRE219_Blog	0.250	0.222	0.235	0.000	0.000	0.000	0.167	0.133	0.148	0.000	0.000	0.000
blogposts_FG041_Blog	1.000	1.000	**1.000**	0.600	0.750	0.667	0.769	0.833	0.800	0.462	0.500	0.480
pcc_maz-10374	1.000	1.000	**1.000**	0.000	0.000	0.000	0.625	0.526	0.571	0.125	0.105	0.114
pcc_maz-13946	0.182	0.167	0.174	0.000	0.000	0.000	0.071	0.050	0.059	0.071	0.050	0.059
pcc_maz-6918	0.455	0.385	0.417	0.000	0.000	0.000	0.188	0.143	0.162	0.125	0.095	0.108
Average	0.541	0.545	0.542	0.206	0.222	0.213	0.366	0.355	0.359	0.222	0.220	0.221

Table 5: Performance of DPLP parser

Document	EDUs			Spans			Nuclearity			Relations		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
apa_1	0.286	0.500	0.364	0.071	0.111	0.087	0.171	0.286	0.214	0.000	0.000	0.000
apa_2	0.250	0.444	0.320	0.100	0.167	0.125	0.269	0.467	0.341	0.115	0.200	0.146
apa_3	1.000	1.000	**1.000**	0.167	0.182	0.174	0.429	0.444	0.436	0.107	0.111	0.109
blogposts_CRE210_Blog	0.250	0.400	0.308	0.000	0.000	0.000	0.286	0.571	0.381	0.143	0.286	0.190
blogposts_CRE219_Blog	0.750	0.600	0.667	0.250	0.167	**0.200**	0.583	0.438	**0.500**	0.417	0.312	**0.357**
blogposts_FG041_Blog	0.750	0.857	0.800	0.000	0.000	0.000	0.385	0.417	0.400	0.154	0.167	0.160
pcc_maz-10374	0.917	0.846	0.880	0.000	0.000	0.000	0.500	0.381	0.432	0.188	0.143	0.162
pcc_maz-13946	0.909	0.833	0.870	0.333	0.143	**0.200**	0.571	0.421	0.485	0.143	0.105	0.121
pcc_maz-6918	1.000	1.000	**1.000**	0.000	0.000	0.000	0.312	0.294	0.303	0.000	0.000	0.000
Average	0.679	0.720	0.690	0.102	0.085	0.087	0.390	0.413	0.388	0.141	0.147	0.138

Table 6: Performance of LLM (Experiment 1)

Document	EDUs			Spans			Nuclearity			Relations		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
apa_1	1.000	1.000	**1.000**	0.143	0.133	0.138	0.486	0.472	0.479	0.114	0.111	0.113
apa_2	1.000	1.000	**1.000**	0.200	0.182	0.190	0.385	0.370	0.377	0.077	0.074	0.075
apa_3	1.000	1.000	**1.000**	0.250	0.273	**0.261**	0.429	0.444	0.436	0.107	0.111	0.109
blogposts_CRE210_Blog	1.000	1.000	**1.000**	0.000	0.000	0.000	0.571	0.615	0.593	0.429	0.462	**0.444**
blogposts_CRE219_Blog	0.750	0.600	0.667	0.250	0.167	0.200	0.583	0.438	0.500	0.250	0.188	0.214
blogposts_FG041_Blog	1.000	1.000	**1.000**	0.200	0.200	0.200	0.615	0.615	**0.615**	0.231	0.231	0.231
pcc_maz-10374	1.000	1.000	**1.000**	0.000	0.000	0.000	0.562	0.474	0.514	0.125	0.105	0.114
pcc_maz-13946	0.909	0.833	0.870	0.333	0.143	0.200	0.643	0.474	0.545	0.357	0.263	0.303
pcc_maz-6918	1.000	1.000	**1.000**	0.000	0.000	0.000	0.312	0.263	0.286	0.000	0.000	0.000
Average	0.962	0.937	0.948	0.153	0.122	0.132	0.510	0.463	0.483	0.188	0.172	0.178

Table 7: Performance of LLM (Experiment 2)

Document	EDUs			Spans			Nuclearity			Relations		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
apa_1	1.000	1.000	**1.000**	0.214	0.214	0.214	0.429	0.429	0.429	0.143	0.143	0.143
apa_2	1.000	1.000	**1.000**	0.200	0.200	0.200	0.385	0.385	0.385	0.077	0.077	0.077
apa_3	1.000	1.000	**1.000**	0.250	0.250	**0.250**	0.393	0.393	0.393	0.107	0.107	0.107
blogposts_CRE210_Blog	1.000	1.000	**1.000**	0.000	0.000	0.000	0.500	0.583	0.538	0.357	0.417	0.385
blogposts_CRE219_Blog	1.000	1.000	**1.000**	0.000	0.000	0.000	0.500	0.462	0.480	0.250	0.231	0.240
blogposts_FG041_Blog	1.000	1.000	**1.000**	0.200	0.250	0.222	0.615	0.667	**0.640**	0.385	0.417	**0.400**
pcc_maz-10374	1.000	1.000	**1.000**	0.000	0.000	0.000	0.500	0.400	0.444	0.125	0.100	0.111
pcc_maz-13946	1.000	1.000	**1.000**	0.333	0.143	0.200	0.571	0.444	0.500	0.286	0.222	0.250
pcc_maz-6918	1.000	1.000	**1.000**	0.000	0.000	0.000	0.312	0.278	0.294	0.000	0.000	0.000
Average	1.000	1.000	**1.000**	0.133	0.117	0.121	0.467	0.449	0.456	0.192	0.190	0.190

Table 8: Performance of LLM (Experiment 3)

(Coarse) **Relation label**	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Gold**
ATTRIBUTION	2.58	2.12	2.75	6.32
BACKGROUND	0.65	2.12	1.65	5.17
CAUSE	0.00	0.00	0.00	1.72
CONDITION	0.00	0.00	0.55	0.57
CONTRAST	0.65	0.53	0.55	6.32
ELABORATION	7.74	5.82	6.59	10.34
ENABLEMENT	0.00	0.00	0.55	1.15
EVALUATION	1.29	0.00	0.55	0.57
EXPLANATION	0.65	1.59	2.75	2.87
JOINT	68.39	55.03	63.19	22.99
MANNER-MEANS	0.00	0.00	0.00	0.57
SAME-UNIT	3.87	17.99	4.40	0.00
TEMPORAL	0.00	0.00	0.00	2.87
SPAN	14.19	14.81	16.48	38.51

Table 9: Percentages of label distributions across LLM-experiments compared to gold

Token-span	**EDUs**		**Spans**		**Nuclearity**		**Relations** (Fine)		**Relations** (Coarse)	
	gold	parsed	gold	parsed	gold	parsed	gold	parsed	gold	parsed
0, 12	✓	✓			N	N	Conjunction	Joint	Joint	Joint
0, 26				✓		S		Explanation		Explanation
0, 51				✓		N		Joint		Joint
0, 65			✓		N		Contrast		Contrast	
0, 80			✓		S		Background		Background	
12, 26	✓	✓			N	N	Span	Joint	Span	Joint
12, 65			✓		N		Conjunction		Joint	
26, 37	✓	✓			N	N	Conjunction	Joint	Joint	Joint
26, 51			✓		N		List		Joint	
26, 65			✓		S		Elaboration		Elaboration	
37, 51	✓	✓			N	N	Conjunction	Joint	Joint	Joint
51, 65	✓	✓			N	N	List	Joint	Joint	Joint
51, 73				✓		N		Joint		Joint
51, 128				✓		N		Joint		Joint
65, 73	✓	✓			N	N	Conjunction	Joint	Joint	Joint
65, 80			✓		N		Contrast		Contrast	
73, 80	✓	✓			N	N	Conjunction	Joint	Joint	Joint
73, 128				✓		N		Joint		Joint
80, 128	✓	✓			N	N	Span	Joint	Span	Joint

Table 10: `blogposts_CRE210_Blog` after alignment; Experiment: LLM (Experiment 2)

Token-span	EDUs		Spans		Nuclearity		Relations (Fine)		Relations (Coarse)	
	gold	parsed	gold	parsed	gold	parsed	gold	parsed	gold	parsed
0, 13	✓	✓			N	N	Span	Joint	Span	Joint
0, 42				✓		N		Span		Span
0, 84				✓		N		Span		Span
0, 166			✓		N		Span		Span	
14, 22	✓	✓			S	N	Concession	Joint	Contrast	Joint
14, 166			✓		S		Evidence		Explanation	
23, 42	✓	✓			N	S	List	Elaboration	Joint	Elaboration
43, 58	✓	✓			N	N	List	Span	Joint	Span
43, 84				✓		S		Elaboration		Elaboration
59, 75	✓	✓			N	N	List	Joint	Joint	Joint
59, 84				✓		S		Evaluation		Evaluation
76, 84	✓	✓			N	N	List	Joint	Joint	Joint
85, 94	✓	✓			N	N	Span	Joint	Span	Joint
85, 109				✓		N		Joint		Joint
85, 170				✓		S		Contrast		Contrast
85, 171			✓		S		Interpretation		Evaluation	
95, 109	✓	✓			N	N	Span	Joint	Span	Joint
110, 128		✓				N		Joint		Joint
110, 151	✓				S		Background		Background	
129, 150		✓				N		Joint		Joint
129, 170				✓		S		Background		Background
151, 165		✓				N		Joint		Joint
152, 166	✓				N		Span		Span	
166, 170		✓				S		Elaboration		Elaboration
167, 171	✓				S		Means		Manner-means	

Table 11: pcc_maz-13946 after alignment; Experiment: LLM (Experiment 1)
Note the missing token-span between 171-195.

(0, 47): [Die Oranienburger Mittelstadt kann sich glücklich schätzen , dass sie zu den 15 ausgewählten Stadtteilen im Land Brandenburg gehört , die in den nächsten fünf Jahren vom 220 Millionen Mark schweren EU-Förderprogramm " Zukunft im Stadtteil " (ZiS) profitieren werden .]
(47, 62): [Eine erste Bestandsaufnahme der Situation in der Mittelstadt brachte nun ein überraschend negatives Bild .]
(62, 72): [Bisher wurde ihr Zustand zwar als nicht unproblematisch eingeschätzt ,]
(72, 88): [einen Vergleich mit schwer gebeutelten Neubaugebieten , wie etwa Velten-Süd , hätten Kommunalpolitiker aber abgelehnt .]
(88, 109): [Eine Bürgerbefragung ergab jedoch , dass mehr als die Hälfte der Bewohner der Mittelstadt schon über den Wegzug nachgedacht hat .]
(109, 134): [Eine bedeutende Minderheit schätzt die Wohnqualität als " schlecht " oder " sehr schlecht " ein .]
(134, 152): [Als deutliches Zeichen für den beginnenen Abstieg wird der Einbau von Sperrgittern an Durchgängen zu Wohnanlagen gewertet .]
(152, 162): [Einwohner sprechen von ständigem Vandalismus im Umfeld der Schulen .]
(162, 173): [Dass es nach den Notsignalen wieder Zeichen der Hoffnung gibt ,]
(173, 179): [dafür könnte das ZiS-Programm sorgen .]
(179, 195): [Das aber wird ohne ein Mitwirken der Mieter und Vermieter in der Mittelstadt nicht funktionieren .’

Figure 7: Gold segmentation of maz-13946, indexed by Token-spans

D Prompts & output schemas

List of output schemas


```
class Segmentation(pydantic.BaseModel):
    """
    Ziel der Segmentierung ist es, den Text in eine lineare Folge von EDUs zu partitionieren.

    ## Grundregeln der Segmentierung
    Die Segmentierung geschieht auf Basis der folgenden Grundregeln:
    - Eine EDU entspricht einer erkennbaren, selbstständigen Sprechhandlung (Illokution). Diese muss
    ↪  aber nicht im engen Sinne strukturell „vollständig“ sein: Etwaige Elisionen sind bei der
    ↪  Beurteilung aufzufüllen, anaphorische/ kataphorische Verweise durch ihre Antezedenten zu
    ↪  ersetzen.
        - Ausnahme: Parenthetische Einschübe, die in der Mitte eines Segments stehen, werden auch dann
        ↪  nicht als EDU abgegrenzt, wenn sie eigentlich eine Illokution darstellen.
    - Eine EDU muss ein Verb enthalten.
        - Ausnahme: Präpositionalphrasen werden als eigene EDU abgegrenzt, wenn folgende Bedingungen
        ↪  erfüllt sind:
            - Die eingebettete Nominalphrase ein Substantiv enthält, mit dem auf einen Sachverhalt
            ↪  referiert wird (z. B., aber nicht unbedingt, ein nominalisiertes Verb).
            - Die Präpositionalphrase steht in einer klar erkennbaren Kohärenzrelation zum umgebenden
            ↪  Haupt- bzw. Nebensatz.
    - EDUs dürfen sich nicht überschneiden.
    - Eine EDU erstreckt sich über einen vollständigen Satz oder einen satzwertigen/ phrasalen Teil
    ↪  eines Satzes. D.h. EDUs erstrecken sich nicht über Satzgrenzen hinweg.

    ## Unterscheidung nach strukturellen Typen
    ### Strukturelle Typen
    - Hauptsatz
        - vollständig (HS)
        - unvollständig (HSF)
    - Nebensatz
        - Satzgliederweiterung:
            - Subjektsatz (SUB)
            - Objektsatz (OBJ)
            - Adverbialsatz (ADV)
            - Prädikativsatz (PRD)
        - Attributsatz:
            - restr. Relativsatz (ARR)
            - nichtrestr. Relativsatz (ANR)
            - Partizipialkonstruktion (AKP)
            - sonstige (ATT)
        - weiterführender Nebensatz (WEI): nimmt Bezug nicht auf ein einzelnes Element des
        ↪  vorangehenden (Teil-)Satzes, sondern auf dessen gesamte Aussage.
    - Fragment
        - einleitend (FRE)
        - beendend (FRB)

    Die Grundregeln werden für die unterschiedlichen Segmenttypen wie folgt umgesetzt:
        - Hauptsätze (HS) und -fragmente (HSF) bilden stets eine eigenständige EDU. (HSF stellen dabei
        ↪  eine gewisse „Grauzone“ für die Grundregel dar, da hier das Problem der Elision auftritt.)
        - Fragmente: FRE & FRB sind auf den Status einer selbstständigen Illokution hin zu überprüfen
        ↪  und dann entweder als eigene EDU abzugrenzen oder dem benachbarten HS/HSF anzufügen.
        - Nebensätze:
            - SUB, OBJ, PRD bilden nur dann eine eigenständige EDU, wenn der Autor darin eine
            ↪  eingebettete Proposition persönlich bewertet bzw. beurteilt
            - im Fall der Reihung zweier/ mehrerer koordinierte SUB, OBR oder PRD (mit oder ohne
            ↪  Konjunktion) müssen auch zwei/ mehrere EDUs geschaffen werden
            - ARR und ATT dienen der näheren Charakterisierung eines Diskursreferenten, übermitteln
            ↪  also keine eigenständige Informationseinheit und bilden daher keine EDU.
            - ANR und WEI kommunizieren selbstständige  Informationseinheiten und bilden eine EDU.
```

```
            - ADV stehen in einer durch einen Konnektor markierten inhaltlichen Relation zum
            ↪  übergeordneten HS/HSF, damit eine eigenständige Informationseinheit und bilden eine
            ↪  EDU.
            - AKP müssen im Einzelfall anhand der Grundregel beurteilt werden.

    ## Beispiel-Segmentierung
    Die Nummern im nachfolgenden Text kennzeichnen EDUs.

    (1) Die Lausitzer Braunkohle AG ( Laubag ) hat im abgelaufenen Geschäftsjahr mit der Förderung und
    ↪  dem Verkauf von Braunkohle erstmals Verluste gemacht. (2) Damit wird erneut deutlich, dass
    ↪  eine Neuordnung der Energiewirtschaft in Ostdeutschland überfällig ist. (3) Denn die Laubag
    ↪  und ihr wichtigster Kunde, die Veag, hängen wie siamesische Zwillinge voneinander ab. (4) Als
    ↪  die Veag wegen der Liberalisierung des Strommarktes unter Druck geriet und ihre Strompreise
    ↪  senken musste, (5) hielt sie sich bei ihrem Lieferanten schadlos. (6) Die Laubag musste
    ↪  Preiszugeständnisse machen, (7) die sie nun selbst in Bedrängnis bringen. (8) Einziger Weg
    ↪  aus dem Dilemma ist die Bildung eines neuen Energiekonzerns aus Rohstofflieferanten,
    ↪  Stromerzeugern und Endversorgern, (9) in dem Risiken besser verteilt werden können. (10) Die
    ↪  Idee, auch einen Gasversorger mit ins Boot zu holen, ist im Prinzip nicht schlecht. (11)
    ↪  Allerdings gilt auch hier, dass zu viele Köche den Brei  verderben können. (12) Zumindest muss
    ↪  klar sein, wer im neuen Konzern das Sagen hat.

    # WICHTIG
    Der Text des Dokuments darf **nicht** verändert werden! D.h. der ursprüngliche Text muss sich
    ↪  lückenlos aus der Liste der generierten EDUs wieder zusammensetzen lassen. Z.B.:
        document = '''Dies ist ein Beispieltext, der der Illustration dient. Dies ist wichtig, um
        ↪  Missverständnisse zu vermeiden.'''
        edus = [
            EDU(text="Dies ist ein Beispieltext,", start=0, end=6),
            EDU(text="der der Illustration dient.", start=6, end=11),
            EDU(text="Dies ist wichtig,", start=11, end=15),
            EDU(text="um Missverständnisse zu vermeiden.", start=15, end=20)
        ]
        assert document == ''.join(edu.text for edu in edus)
    """
    document: Union[str, None] = pydantic.Field(
        description="Der unsegmentierte Text, der in EDUs partitioniert wurde.",
        exclude=True,
        default=None
    )
    edus_: List[EDU] = pydantic.Field(
        description="Die EDU-Objekte, die die Segmentierung des Dokuments in EDUs repräsentieren. Die
        ↪  EDUs sind in der Reihenfolge ihres Auftretens im Text sortiert.",
        exclude=True
    )

    @pydantic.computed_field
    @property
    def edus(self) -> Dict[int, EDU]:
        """Segmentierte EDUs, indiziert nach ihrer Reihenfolge im Text."""
        return {i + 1: edu for i, edu in enumerate(self.edus_)}
```

Listing 1: Output schema for step 1 **Segmentation** (adapted from Stede (2016))

```
class SplitArgs(pydantic.BaseModel):
    k: int = pydantic.Field(
        description="Der EDU-Index, an dem der Span geteilt werden soll. Muss im Bereich (span.start=i
        ↪  <= k < span.end=j) liegen. Der erste resultierende Span inkludiert das EDU mit Index k und
        ↪  der zweite Span beginnt mit k+1. Beispiel: span=(1,3), k=2 -> resultierende Spans: (1,2)
        ↪  & (3,3)",
        gt=0
    )
```

Listing 2: Output schema for step 3.2 **Split parent**

```
class NuclearityArgs(pydantic.BaseModel):
    nuclearity: Literal["N-N", "N-S", "S-N"] = pydantic.Field(
```

```
        description="Die Nuklearität, die den beiden Spans zugewiesen werden soll. 'N' steht für
        ↪  Nukleus, 'S' für Satellit. Die Reihenfolge der Nuklearität entspricht der Reihenfolge der
        ↪  Spans."
    )
```

Listing 3: Output schema for step 3.3 **Assign nuclearity**

```
class RelationArgs(pydantic.BaseModel):
    """
    Zur Prüfung, ob für zwei Textsegmente eine bestimmte Relation anwendbar ist, sollte folgendermaßen
    ↪  vorgegangen werden:
        • Gibt es einen Hinweis durch einen Konnektor, der die Relation anzeigt oder die Menge der
        ↪  möglichen Relationen zumindest einschränkt? (In manchen Fällen ist allerdings die an der
        ↪  Oberfläche signalisierte Relation nicht die pragmatisch „wichtige“.)
        • Interpunktionszeichen liefern zwar keine sehr klaren Hinweise auf bestimmte Relationen, doch
        ↪  es gibt Tendenzen, wie z.B. der Zusammenhang zwischen dem Semikolon und kontrastiven
        ↪  Relationen.
        • Möchte der Autor durch die Juxtaposition der Segmente den jeweils genannten Effekt beim
        ↪  Leser erreichen? Dies ist eine notwendige Bedingung für die Anwendung einer Relation.
        • Wenn die Relationsdefinition Beschränkungen für den Typ oder die Funktion von Nukleus,
        ↪  Satellit, oder ihrer Kombination nennt, sind diese erfüllt? Dies sind (soweit vorhanden)
        ↪  ebenfalls notwendige Bedingungen.
        • Wird die mit der Relation verbundene Nukleus/Satellit-Verbindung der Rolle beider Segmente
        ↪  für die Textfunktion gerecht? Dies ist ein weniger striktes Kriterium als die beiden
        ↪  vorgenannten, kann aber oft die Entscheidung erleichtern, wenn mehrere Relationen
        ↪  anwendbar erscheinen.

    ## Relations-Definitionen
        • N: Charakterisierung des Typs und/oder der Funktion des Nukleus (als Beschreibung der
        ↪  Haltung des Autors, nicht des Ausdrucks im Text)
        • S: Charakterisierung des Typs und/oder der Funktion des Satellits (als Beschreibung der
        ↪  Haltung des Autors, nicht des Ausdrucks im Text)
        • N/S: Charakterisierung der Funktion der Nukleus/Satellit Kombination. Wenn es Beschränkungen
        ↪  oder Tendenzen für die textuelle Abfolge von N und S gibt, sind sie hier ebenfalls
        ↪  genannt.
        • Effekt: Charakterisierung des mit der Verwendung der Relation vom Autor intendierten
        ↪  Effekts, formuliert als „vorher–nachher“ Veränderung.
        • Typische Konnektoren
        • Beispiel: N und S sind jeweils markiert. Wenn vorab ein Kontext charakterisiert wird,
        ↪  geschieht das in Kursivschrift.
        • Bemerkung: (optional)
    Die Felder N, S und N/S bleiben frei, wenn es für eine Relation keine entsprechenden
    ↪  Beschränkungen gibt. Bei multinuklearen Relationen entfallen die Felder S und N/S.

    ### Mononukleare Relationen
    #### Attribution
    • N: Inhalt der berichteten Nachricht (muss in einem separaten Satzteil stehen)
    • S: Quelle der Attribution (ein Satzteil mit einem berichtenden Verb oder eine Phrase, die z.B.
    ↪  mit "entsprechend" oder "gemäß" beginnt)
    • N/S: Um einen Satz in Attributionsquelle und Inhalt zu segmentieren, müssen zwei Bedingungen
    ↪  erfüllt sein:
        1) Es muss eine explizite Quelle für die Attribution vorhanden sein. Wenn der Satz, der das
        ↪  berichtende Verb enthält, die Quelle der Zuschreibung nicht angibt und die Quelle auch
        ↪  nicht an anderer Stelle im Satz oder im näheren Kontext identifiziert werden kann,
        ↪  besteht keine Attribution-Relation, und der berichtende und der berichtete Satz werden
        ↪  als eine Einheit behandelt. (Dies kommt häufig bei Passivkonstruktionen oder generischen
        ↪  Ausdrücken vor.)
        2) Der Nebensatz darf kein Infinitivkomplement sein.
    • Effekt: Der Leser erkennt, dass eine Quelle in S über den Inhalt in N berichtet und diesen einer
    ↪  anderen Quelle zuschreibt.
    • Beispiel: [Analysten schätzten]S [dass die Umsätze in US-amerikanischen Geschäften im Quartal
    ↪  ebenfalls zurückgingen.]N
    • Bemerkung: Die Relation wird auch mit kognitiven Prädikaten verwendet, um Gefühle, Gedanken,
    ↪  Hoffnungen usw. einzuschließen. Sie gilt auch im Fall der negativen Formulierung (z.B. "er
    ↪  bestritt, dass...").
    #### Background
```

```
   • N/S: Das Verstehen von S erleichtert dem Leser das Verständnis für den Inhalt von N; S enthält
   ↪  orientierende Hintergrundinformation, ohne die N nicht oder nur schwer verständlich wäre. Im
   ↪  Text geht S meist dem N voraus, aber nicht immer. Ein Background-Satellit am Anfang eines
   ↪  Textes hat oftmals auch die Funktion, das Thema kurz einzuführen.
   • Effekt: Die Fähigkeit des Lesers, den Inhalt von N zu verstehen, wird verbessert.
   • Typische Konnektoren: (selten durch Konnektoren angezeigt)
   • Beispiel: [Burkina Faso hieß bis 1984 noch Obervolta.]S [Nach einer EMNID-Umfrage glauben viele
   ↪  Europäer bis heute, dass es sich um zwei verschiedene Länder handelt.]N
   • Bemerkung: Die Relation besteht eher selten zwischen EDUs, sondern in der Regel zwischen größeren
   ↪  Segmenten. Viele Kommentare sind so strukturiert, dass ein Background-Satellit den Textanfang
   ↪  bildet, also den inhaltlichen Ausgangspunkt für die nachfolgende Kommentierung darstellt.
   #### Cause
   • N: ein realer Sachverhalt in der Welt.
   • S: ein realer Sachverhalt in der Welt.
   • N/S: der in N beschriebene Sachverhalt wird durch den in S beschriebenen Sachverhalt verursacht.
   • Effekt: Leser erkennt den Kausalzusammenhang in der Welt.
   • Typische Konnektoren: weil; da; deshalb; ...
   • Beispiel: [Überrascht reagierte auch Bürgermeister Jochen Wagner.]N [Schließlich gaben die
   ↪  Stadtverordneten erst Montagabend grünes Licht für die weitere Erschließung des neuen
   ↪  Ortsteils Diepensee.]S
   #### Comparison
   • N: Entität/Sachverhalt/Bereich, der als Vergleichsobjekt dient.
   • S: Entität/Sachverhalt/Bereich, der hinsichtlich eines sich unterscheidenden Aspekts mit N
   ↪  verglichen wird.
   • N/S: N & S gleichen einander in einer oder mehr Dimensionen und unterscheiden sich in einem oder
   ↪  mehreren Aspekten. Die Bereiche/Entitäten/usw. stehen nicht im Gegensatz zueinander.
   • Effekt: Leser erkennt den Vergleichscharakter der Relation.
   • Bemerkung: Comparison vergleicht zwei Textbereiche anhand einer Dimension, die abstrakt sein
   ↪  kann. Die Relation kann vermitteln, dass einige abstrakte Entitäten, die sich auf die
   ↪  Vergleichsrelation beziehen, ähnlich, unterschiedlich, größer als, kleiner als usw. sind.
   #### Condition
   • N: Eine hypothetische, künftige oder anderweitig irreale Situation.
   • S: Eine hypothetische, künftige oder anderweitig irreale Situation.
   • N/S: S beeinflusst die Realisierung von N: N wird nur dann (nicht) realisiert, wenn S (nicht)
   ↪  realisiert wird.
   • Effekt: Leser erkennt die Abhängigkeit der (Nicht-)Realisierung von N von der
   ↪  (Nicht-)Realisierung von S.
   • Typische Konnektoren: es sei denn; ...
   • Beispiel: [Morgen wird der Satellit in den Pazifik stürzen.]N [Es sei denn, er verglüht doch
   ↪  noch vollständig in der Erdatmosphäre.]S
   #### Contrast
   • N: Ein „wichtigerer“ Inhalt, der mit S vergleichbar aber nicht identisch ist.
   • S: Ein weniger „wichtiger“ Inhalt, der mit N vergleichbar aber nicht identisch ist.
   • N/S: Die Inhalte sind einander ähnlich, miteinander vergleichbar; sie sind aber nicht identisch,
   ↪  sondern unterscheiden sich in für den Autor wichtigen Aspekten.
   • Effekt: Leser erkennt die Vergleichbarkeit von N & S und die Betonung des Unterschieds.
   • Typische Konnektoren: demgegenüber; hingegen; während; aber; ...
   • Beispiel: [Mein erstes Auto war ein Kleinwagen.]N/S [Das zweite hingegen ein ausgewachsener
   ↪  Kombi.]S/N
   #### Elaboration
   • N/S: S liefert genauer Information bzw. Details zum Inhalt von N. N geht S im Text voraus.
   ↪  Typische Zusammenhänge zwischen N und S sind Menge::Element, Ganzes::Teil,
   ↪  Abstraktion::Instanz, Vorgang::Einzelschritt.
   • Effekt: Leser erkennt, dass S genauere Information zu N liefert.
   • Typische Konnektoren: besonders; beispielsweise; ...
   • Beispiel: [Diepensee siedelt um.]N [Ohne Wenn und Aber.]S
   #### Enablement
   • N: Eine vom Leser auszuführende Tätigkeit.
   • N/S: Das Verstehen von S erleichtert dem Leser, die von N beschriebene Tätigkeit auszuführen.
   • Effekt: Die Fähigkeit des Lesers, die Tätigkeit in N auszuführen, wird gesteigert.
   • Typische Konnektoren: damit; ...
   • Beispiel: [Wechseln Sie die Zündkerzen aus.]N [Ein Vierkantschlüssel befindet sich unter der
   ↪  Abdeckung.]S
   #### Evaluation
   • N: Beschreibung eines Sachverhalts, oder eine subjektive Aussage (allerdings nicht aus
   ↪  Perspektive des Autors)
   • S: Eine subjektive Bewertung (positiv/negativ, erstrebenswert/nicht erstrebenswert) aus
   ↪  Perspektive des Autors
   • N/S: S bewertet N
   • Effekt: Leser erkennt die Bewertungsrelation zwischen S und N
```

```
• Typische Konnektoren: (selten durch Konnektoren angezeigt)
• Beispiel: [Seine Vergangenheit schien wie ein Fluch über dem Hotelkomplex zu liegen.]S
↪  [Jahrelang hatte das Amtsgericht Potsdam umsonst versucht, es an den Mann zu bringen.]N
• Bemerkung: Meist folgt das evaluierende Segment auf das evaluierte; manchmal ist es jedoch
↪  umgekehrt, wie im obigen Beispiel.
#### Explanation
• N: Eine Aussage/Einschätzung/These, die der Leser möglicherweise nicht akzeptiert oder als nicht
↪  genügend wichtig oder positiv einschätzt.
• S: Eine Aussage, die der Leser wohl akzeptieren wird; in der Regel die „objektive“ Beschreibung
↪  eines Faktums.
• N/S: Durch das Verstehen von S akzeptiert der Leser die Aussage von N leichter, bzw. teilt die
↪  damit verbundene Einschätzung des Autors.
• Effekt: Leser glaubt eher, dass die in N getroffene Aussage zutrifft.
• Typische Konnektoren: (kausale Konnektoren)
• Beispiel: [Und nun scheint sogar unsere Landesregierung entschlossen, diese scheinbare
↪  Gleichbehandlung der beiden Fächer zu beseitigen.]N [Stolpe, Reiche und Co. sagen zwar Ja zu
↪  einem möglichen Kompromissangebot aus Karlsruhe, dekretieren aber: Einen Wahlpflichtbereich
↪  LER/Religion kann es nicht geben.]S
• Bemerkung: Explanation verbindet oft ein längeres Satellit-Segment mit einem kürzeren Nukleus
↪  (der These).
#### Manner-Means
• N: Eine Handlung/Aktivität
• N/S: S gibt Informationen, die die Realisierung/Ausführung von N wahrscheinlicher/einfacher
↪  machen (z. B. ein Instrument)
• Effekt: Leser erkennt den Zusammenhang der höheren Wahrscheinlichkeit oder der Vereinfachung der
↪  Handlungsausführung
• Typische Konnektoren: dazu; damit; ...
• Beispiele: [Berliner fahren im August immer gern nach Lichtenrade.]N [Dazu nehmen sie meistens
↪  die S25.]S
#### Summary
• N: N umfasst mehr als nur eine EDU.
• N/S: S folgt im Text auf N und wiederholt die Information von N, ist jedoch kürzer.
• Effekt: Leser erkennt die zusammenfassende Funktion von S.
• Typische Konnektoren: in Kürze; ...
#### TextualOrganization
• N/S: S verknüpft N mit einem funktional-strukturell übergeordneten Element. Weder S noch N haben
↪  eine rhetorische Beziehung zueinander, sondern dienen der Organisation des Texts.
• Effekt: Leser erkennt die zusammenfassende Funktion von S.
• Bemerkung: TextualOrganization verknüpft funktional-strukturelle Elemente, wie z. B. Titel,
↪  Autor oder Signaturblock.
#### Topic-Change
• N: Das Thema von N ist das bedeutendere Element, zu dem der Fokus wechselt.
• S: Das Thema von S ist das weniger bedeutende Element, das aus dem Fokus rückt.
• Bemerkung: Topic-Change wird verwendet, um größere Textabschnitte zu verknüpfen, wenn der Fokus
↪  von einem Abschnitt zum anderen wechselt.
#### Topic-Comment
• S: Der Inhalt von S kann als „Problem“ aufgefasst werden.
• N/S: Der Inhalt von N kann als Lösung des in S dargestellten Problems aufgefasst werden. N geht
↪  in der Regel S im Text voraus.
• Effekt: Leser erkennt N als Lösung des Problems in S.
• Typische Konnektoren: (selten durch Konnektoren angezeigt)
• Beispiel: [Mit der Verabschiedung des Nichtraucherschutzgesetzes sitzen viele Kneipen in der
↪  Falle.]S [Es empfiehlt sich, früh genug auf die Einrichtung abtrennbarer Räume zu achten.]N

### Multinukleare Relationen
#### Joint
• N: Die nicht unbedingt typgleichen Nuklei geben separate Informationen, stehen aber in keiner
↪  klar identifizierbaren semantischen oder pragmatischen Relation zueinander, haben gemeinsam
↪  auch nicht den Charakter einer Aufzählung. Nichtsdestotrotz besteht eine kohärente Verbindung,
↪  weil sie der übergeordneten Textfunktion gleichermaßen dienen.
• Effekt: Leser erkennt, dass jeder Nukleus „eine eigene Botschaft“ hat, die aber jeweils
↪  derselben Textfunktion dienlich sind.
• Typische Konnektoren: additive Konnektoren wie "zudem", "auch"
• Bemerkung: Joint ist zu verwenden, wenn eine multinukleare Relation gesucht wird und keine der
↪  übrigen passt.
#### Same-Unit
• N: Die in bestimmtem Sinne typgleichen Nuklei geben Informationen, die als zusammengehörig,
↪  aufzählend erkennbar sind, mithin eine gemeinsame Rolle für die Textfunktion spielen.
• Effekt: Leser erkennt die gemeinsame Funktion der Nuklei.
• Typische Konnektoren: Komma, Nummerierungen, "und", "oder", "je A, desto B", ...
```

```
    • Beispiel: Was ich gestern getan habe? [Essen kochen,]N [Kinder versorgen,]N [Bad putzen.]N
    • Bemerkung: Same-Unit findet vor allem Anwendung, um zwei, z.B. durch Relativsätze oder
    ↪  Parenthesen unterbrochene, Teile des gleichen EDUs zu verbinden.
    #### Temporal
    • N: Die Nuklei beschreiben Sachverhalte der Welt, die in einer bestimmten zeitlichen Abfolge
    ↪  stattfinden.
    • Effekt: Leser erkennt die temporale Relation zwischen den Nuklei.
    • Typische Konnektoren: "dann"; "anschließend"; "und"; "zuvor", ...
    • Beispiel: [Um neun betrat die Lehrerin den Klassenraum.]N [Fünf Minuten später verkündete sie,
    ↪  dass ein Test geschrieben wird.]N
    • Bemerkung: Die Nennung der Ereignisse kann der zeitlichen Abfolge entsprechen ("dann") oder
    ↪  gegenäufig sein ("zuvor").
    """
    relation: Union[Mononuclear, Multinuclear] = pydantic.Field(
        description="Die RST-Relation, die die beiden Spans verbindet. Falls beide Spans Nuklei sind,
        ↪  muss die Relation eine Multinuclear-Relation sein. Falls einer der beiden Spans ein
        ↪  Satellit ist, muss die Relation eine Mononuclear-Relation sein."
    )
```

Listing 4: Output schema for step 3.4 **Assign relation** (adapted from Stede (2016) & Carlson and Marcu (2001))