

Context-Preserving Text Simplification

Christina Niklaus¹³, Matthias Cetto¹, André Freitas², and Siegfried Handschuh¹³

¹ University of St.Gallen

{christina.niklaus, matthias.cetto, siegfried.handschuh}@unisg.ch

² University of Manchester

andre.freitas@manchester.ac.uk

³ University of Passau

{christina.niklaus, siegfried.handschuh}@uni-passau.de

Abstract

We present a context-preserving text simplification (TS) approach that recursively splits and rephrases complex English sentences into a semantic hierarchy of simplified sentences. Using a set of linguistically principled transformation patterns, input sentences are converted into a hierarchical representation in the form of core sentences and accompanying contexts that are linked via rhetorical relations. Hence, as opposed to previously proposed sentence splitting approaches, which commonly do not take into account discourse-level aspects, our TS approach preserves the semantic relationship of the decomposed constituents in the output. A comparative analysis with the annotations contained in the RST-DT shows that we are able to capture the contextual hierarchy between the split sentences with a precision of 89% and reach an average precision of 69% for the classification of the rhetorical relations that hold between them.

1 Introduction

Sentences that present a complex linguistic structure can be *hard to comprehend by human readers*, as well as *difficult to analyze by semantic applications*. Identifying grammatical complexities in a sentence and transforming them into simpler structures is the goal of syntactic TS. One of the major types of operations that are used to perform this rewriting step is *sentence splitting*: it divides a sentence into several shorter components, with each of them presenting a simpler and more regular structure that is easier to process by both humans (Siddharthan and Mandya, 2014; Saggion et al., 2015; Ferrés et al., 2016) and machines (Štajner and Popovic, 2016; Štajner and Popović, 2018; Saha and Mausam, 2018).

We propose a sentence splitting approach that can be used as a preprocessing step to generate

an intermediate representation. The objective is to facilitate and improve the performance of downstream tasks whose predictive quality deteriorates with sentence length and complexity. Our approach aims to **break down a complex sentence into a set of minimal propositions**, i.e. a sequence of sound, self-contained utterances with a simple and regular structure. Each of them presents a minimal unit of coherent information and cannot be further decomposed into meaningful propositions.

However, any sound and coherent text is not simply a loose arrangement of self-contained units, but rather a logical structure of utterances that are semantically connected (Siddharthan, 2014). Consequently, when carrying out syntactic TS operations without considering discourse implications, the rewriting may easily result in a disconnected sequence of simplified sentences, making the text harder to interpret. The vast majority of existing structural TS approaches though do not take into account discourse-level aspects. Therefore, they are prone to producing a set of incoherent utterances where important contextual information is lost. Thus, in order to **preserve the coherence structure** of the input we propose a context-preserving TS approach. It establishes a semantic hierarchy between the split components by (1) setting up a contextual hierarchy and (2) classifying the semantic relationship that holds between them (see Figure 1).

To the best of our knowledge, this is the first time that syntactically complex sentences are *split and rephrased within the semantic context* in which they occur. Our framework differs from previously proposed approaches by using a linguistically grounded transformation stage that applies clausal and phrasal disembedding mechanisms to transform sentences into shorter utterances with a more regular structure. By using a recursive top-down approach, it generates a novel *hierarchical*

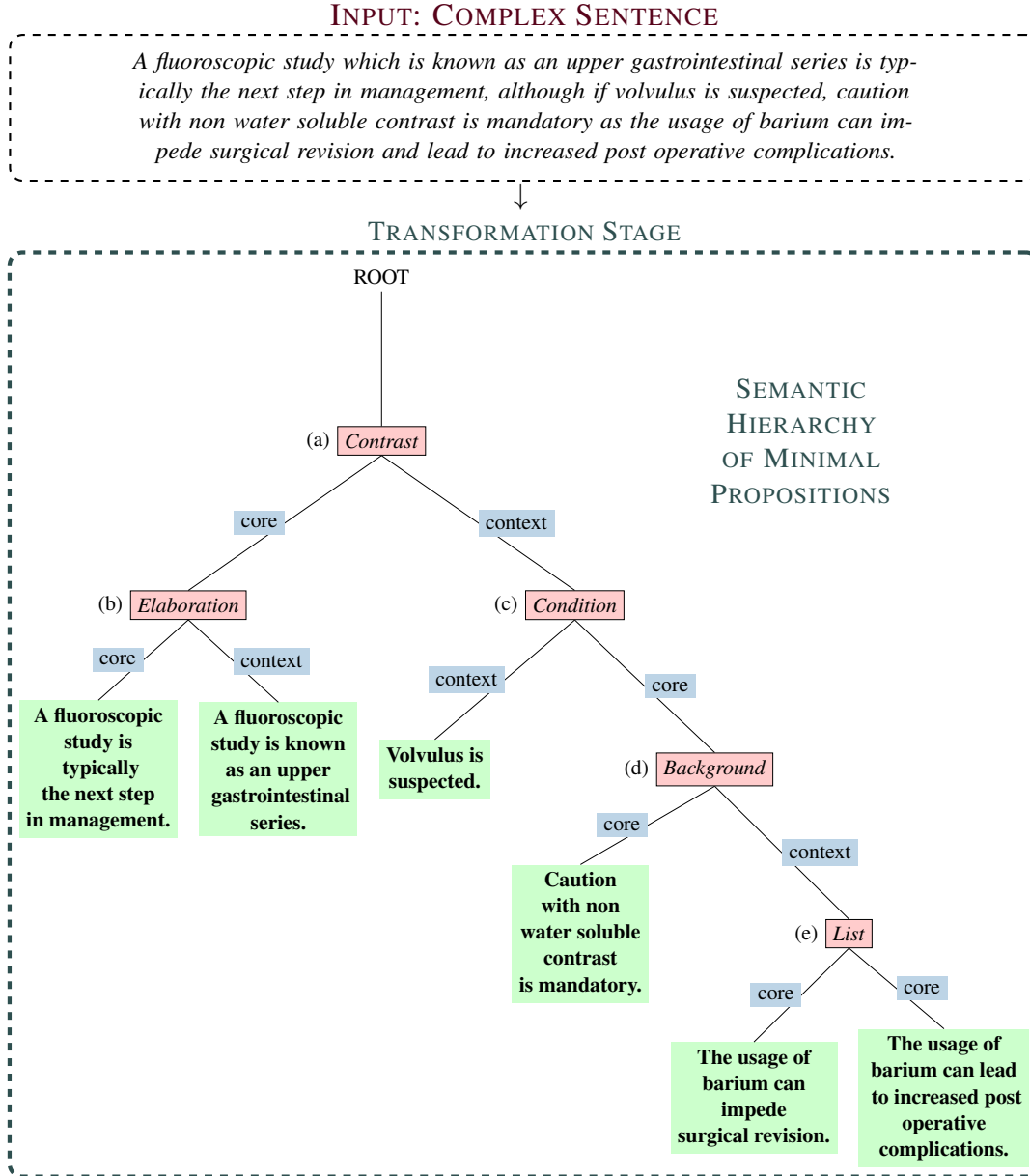


Figure 1: A complex sentence is transformed into a semantic hierarchy of simplified sentences in the form of minimal, self-contained propositions that are linked to each other via rhetorical relations. The output presents a regular, fine-grained structure that preserves the context of the input in the form of hierarchically ordered and semantically interconnected sentences.

representation between those units, capturing both their semantic context and relations to other units in the form of rhetorical relations. By taking advantage of the resulting fine-grained representation, the complexity of downstream tasks may be reduced, thus improving their performance. In addition, by incorporating the semantic context of the source sentences, our proposed representation preserves important contextual information that is needed to maintain the coherence structure of the input, allowing for a proper interpretation of complex assertions.

In summary, we make the following contributions: (i) We conduct a systematic analysis of syntactic phenomena involved in complex sentence constructions whose findings are materialized in a small set of 35 hand-crafted transformation rules. (ii) We propose a context-preserving syntactic TS approach which transforms complex sentences into a semantic hierarchy of minimal propositions. (iii) The proposed method is linguistically grounded and does not require any training data. (iv) As a proof of concept, we develop a reference implementation. (v) We perform a comprehensive empirical

evaluation, demonstrating that we reach state-of-the-art performance in the classification of both the hierarchical order and the semantic relationship that hold between the split sentences.

2 Context-Preserving Sentence Splitting

We present DISSIM, a context-preserving TS approach that creates a semantic hierarchy of simplified sentences.¹ It takes a sentence as input and performs a recursive transformation stage that is based upon a small set of 35 hand-crafted rules.

2.1 Transformation Patterns

In the development of the transformation patterns, we followed a principled and systematic procedure, with the goal of eliciting a universal set of transformation rules. They were heuristically determined in a rule-engineering process that was carried out on the basis of an in-depth study of the literature on syntactic sentence simplification, e.g. (Siddharthan, 2006, 2014, 2002; Siddharthan and Mandya, 2014; Evans and Orăsan, 2019; Heilman and Smith, 2010; Mitkov and Saggion, 2018; Mallinson and Lapata, 2019; Ferrés et al., 2016). Next, we performed a thorough linguistic analysis of the syntactic phenomena that need to be tackled in the sentence splitting task. Details on the underlying linguistic principles, supporting the systemacity and universality of the developed transformation patterns, can be found in Section A in the appendix.

The transformation patterns encode syntactic and lexical features that can be derived from a sentence’s phrase structure. Each rule specifies (1) how to *split up and rephrase* the input into structurally simplified sentences and (2) how to *set up a contextual hierarchy* between the split components and how to *identify the semantic relationship* that holds between those elements. An example of a transformation rule is provided in Table 1.²

2.2 Data Model: Linked Proposition Tree

The transformation algorithm takes a complex sentence as input and recursively transforms it into a semantic hierarchy of minimal propositions. The output is represented as a linked proposition tree.

¹The source code of our framework is publicly available for download under the following link: <https://github.com/Lambda-3/DiscourseSimplification>.

²For reproducibility purposes, the full set of patterns can be found online: https://github.com/Lambda-3/DiscourseSimplification/tree/master/supplemental_material.

Its basic structure is depicted in Figure 2. A linked proposition tree is a labeled binary tree $LPT = (V, E)$.

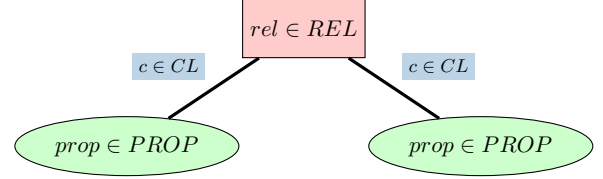


Figure 2: Basic structure of a linked proposition tree LPT . It represents the data model of the semantic hierarchy of minimal propositions.

Let $V \in \{REL, PROP\}$ be the set of nodes, where $PROP$ is the set of leaf nodes denoting the set of **minimal propositions**. A $prop \in PROP$ is a triple $(s, v, o) \in CT$, where $CT = \{SV, SVA, SVC, SVO, SVOO, SVOA, SVOC\}$ represents the set of clause types (Del Corro and Gemulla, 2013). Hence, $s \in S$ denotes a subject, $v \in V$ a verb and $o \in \{O, A, C, OO, OA, OA, \emptyset\}$ a direct or indirect object, adverbial or complement (or a combination thereof). Accordingly, a minimal proposition $prop \in PROP$ is a simple sentence³ that is reduced to its clause type.⁴ Thus, it represents a minimal unit of coherent information where all optional constituents are discarded, resulting in an utterance that expresses a single event consisting of a predicate and its core arguments.

Furthermore, let $REL = \{Contrast, List, Disjunction, Cause, Result, Temporal, Background, Condition, Elaboration, Explanation, Spatial, Attribution, Unknown\}$ be the set of **rhetorical relations**, comprising the set of inner nodes. A $rel \in REL$ represents the semantic relationship that holds between its child nodes. It reflects the semantic context of the associated propositions $prop \in PROP$. In that way, the coherence structure of the input is preserved.

Finally, let $E \in CL$, with $CL \in \{core, context\}$, be the set of **constituency labels**. A $c \in CL$ represents a labeled edge that connects two nodes $V \in LPT$. It enables the distinction between core information and less relevant contextual information. In that way, hierarchical structures between the split propositions $prop \in PROP$ are captured. Figure 1 shows a concrete example of the linked

³A simple sentence comprises exactly one clause.

⁴In addition, a specified set of phrasal elements were extracted. The interested reader may refer to Section A in the appendix for more details.

ROOT <<: (S < (NP \$. (VP < +(VP) (**SBAR** <, (IN \$+ (S < (NP \$. VP))))))))

Table 1: Example of a transformation pattern (for decomposing adverbial clauses). They are specified in terms of Tregex patterns (Levy and Andrew, 2006). A boxed pattern represents the part of a sentence that is extracted from the input and transformed into a new stand-alone sentence. A pattern in bold is deleted from the source. The underlined part is labelled as a context sentence, while the remaining part represents core information. The italic pattern is used as a cue phrase for identifying the rhetorical relation that holds between the decomposed spans.

proposition tree that is generated by our TS approach on an example sentence.

2.3 Transformation Algorithm

The transformation algorithm of our approach is given in Algorithm 1. It takes a sentence as input and applies the transformation patterns to recursively transform it into a semantic hierarchy of minimal propositions, represented as an *LPT*.

Algorithm 1 Transform into Semantic Hierarchy

Input: complex source sentence *str*

Output: linked proposition tree *tree*

```

1: function INITIALIZE(str)
2:   new_leaves ← source sentence str
3:   new_node ← create a new parent node for new_leaves
4:   new_node.labels ← None
5:   new_node.rel ← ROOT
6:   linked proposition tree tree ← initialize with new_node
7:   return tree
8: end function

9: procedure TRAVERSE TREE(tree)
10:  ▷ Process leaves (i.e. propositions) from left to right
11:  for leaf in tree.leaves do
12:    ▷ Check transformation rules in fixed order
13:    for rule in TRANSFORM_RULES do
14:      if match then
15:        ▷ (a) Sentence splitting
16:        simplified_propositions ← decompose leaf into a
17:        set of simplified propositions
18:        new_leaves ← convert simplified_propositions
19:        into leaf nodes
20:        ▷ (b) Constituency Type Classification
21:        new_node ← create a new parent node for new_leaves
22:        new_node.labels ← link each leaf in new_leaves to
23:        new_node and label each edge with the leaf's constituency
24:        type c ∈ CL
25:        ▷ (c) Rhetorical Relation Identification
26:        cue_phrase ← extract cue phrase from leaf.parse_tree
27:        new_node.rel ∈ REL ← match cue_phrase against a
28:        predefined set of rhetorical cue words
29:        ▷ Update Tree
30:        tree.replace(leaf, new_node)
31:        ▷ Recursion
32:        TRAVERSE TREE(tree)
33:      end if
34:    end for
35:  end for
36:  return tree
37: end procedure

```

Initialization In the initialization step (1-7), the linked proposition tree *LPT* is instantiated. At this

point, it consists of the source sentence, which is represented as a single leaf node, and an unlabeled edge to the root node.

Tree Traversal Next, the linked proposition tree is recursively traversed, splitting up the input in a top-down approach (9). Starting from the root node, the leaves are processed in depth-first order. For every leaf (11), we check if its phrasal parse tree matches one of the transformation patterns (13). The rules are applied in a fixed order that was empirically determined.⁵ The first pattern that matches the proposition's parse tree will be employed (14). For instance, the first rule that matches the source sentence from Figure 1 is the pattern depicted in Table 1.

(a) Sentence Splitting In a first step, the current proposition is decomposed into a set of shorter utterances that present a more regular structure (15, 17). This is achieved through disembedding clausal or phrasal components and converting them into stand-alone sentences. Accordingly, the transformation rule encodes both the split point and the rephrasing procedure for reconstructing grammatically sound sentences. Table 2 provides an overview of the linguistic constructs that are tackled by our approach.⁶ Each split will create two sentences with a simpler syntax. They are represented as leaf nodes in the linked proposition tree (18, 19) (see subtask a in Fig. 3). To establish a semantic hierarchy between the split spans, two further subtasks are carried out.

(b) Constituency Type Classification To set up a contextual hierarchy between the split sentences, the transformation rule determines the constituency type *c* ∈ *CL* of the leaf nodes that were created in the previous step (22-24). To differentiate between *core* sentences that contain the key message of the input and *contextual* sentences that

⁵See Section B in the appendix.

⁶Note that this subtask is presented in detail in Niklaus et al. (2019b) and Niklaus et al. (2019a). Therefore, we focus on subtasks b and c in this work.

	CLAUSAL/PHRASAL TYPE	HIERARCHY	# RULES
Clausal disembedding			
1	Coordinate clauses	coordinate	1
2	Adverbial clauses	subordinate	6
3a	Relative clauses (non-restrictive)	subordinate	5
3b	Relative clauses (restrictive)	subordinate	4
4	Reported speech	subordinate	4
Phrasal disembedding			
5	Coordinate verb phrases	coordinate	1
6	Coordinate noun phrases	coordinate	2
6	Participial phrases	subordinate	4
8a	Appositions (non-restrictive)	subordinate	1
8b	Appositions (restrictive)	subordinate	1
9	Prepositional phrases	subordinate	3
10	Adjectival and adverbial phrases	subordinate	2
11	Lead NPs	subordinate	1
	Total		35

Table 2: Linguistic constructs addressed by our proposed context-preserving TS approach DISSIM.

provide additional information about it, the transformation pattern encodes a simple syntax-based method. **Based on the assumption that subordinations commonly express background information, simplified propositions resulting from subordinate clausal or phrasal elements are classified as context sentences, while those emerging from their superordinate counterparts are labelled as core sentences. Coordinations, too, are flagged as core sentences, since they are of equal status and typically depict the main information of the input** (see subtask b in Fig. 3).⁷

(c) Rhetorical Relation Identification To preserve the semantic relationship between the simplified propositions, we classify the rhetorical relation $rel \in REL$ that holds between them. For this purpose, we utilize a predefined list of rhetorical cue words adapted from the work of [Taboada and Das \(2013\)](#).⁸ To infer the type of rhetorical relation, the transformation pattern first extracts the cue phrase of the given sentence (26). It is then used as a lexical feature for classifying the type of rhetorical relation that connects the split propositions (27, 28). For example, the rule in Table 1 specifies that the phrase “although” is the cue word in the source sentence of Fig. 1, which is mapped to a “Contrast” relationship according to the findings in [Taboada and Das \(2013\)](#) (see subtask c in Figure 3).

Recursion Next, the linked proposition tree is updated by replacing the leaf node that was processed in this run with the newly generated subtree

⁷This approach relates to the concept of nuclearity in RST. For details, see Section C in the appendix.

⁸The full list of cue phrases that serve as lexical features for the identification of rhetorical relations is provided in the appendix in Section D.

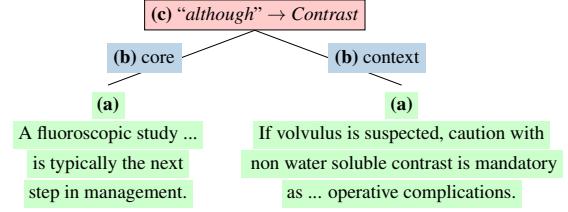


Figure 3: Semantic hierarchy after the first transformation pass. **(Subtask a)** The source sentence is split up and rephrased into a set of syntactically simplified sentences. **(Subtask b)** Then, the split sentences are connected with information about their constituency type to establish a contextual hierarchy between them. **(Subtask c)** Finally, by identifying and classifying the rhetorical relation that holds between the simplified sentences, their semantic relationship is preserved.

(30). It is composed of the simplified propositions, their semantic relationship $rel \in REL$ and constituency labels $c \in CL$. Fig. 3 depicts the result of the first transformation pass on the example sentence from Fig. 1. The resulting leaf nodes are then recursively simplified in a top-down fashion (32).

Termination The algorithm terminates when no more rule matches the set of simplified propositions $prop \in PROP$ in the leaf nodes. It outputs the source sentence’s linked proposition tree LPT (36), representing its semantic hierarchy of minimal semantic units. In that way, the input is transformed into a set of hierarchically ordered and semantically interconnected sentences that present a simplified syntax. Figure 1 shows the final linked proposition tree of our example sentence.

3 Evaluation

3.1 Experimental Setup⁹

3.1.1 Automatic Metrics

We evaluate the constituency type classification and rhetorical relation identification steps by mapping the simplified sentences that were generated in the sentence splitting subtask to the Elementary Discourse Units (EDUs) of the RST-DT corpus.¹⁰ This dataset is a collection of 385 Wall Street Journal articles annotated with rhetorical relations based on the RST framework ([Mann and Thompson, 1988](#)). For matching simplified sentences generated by

⁹To enable reproducible research, all code / datasets used in the experiments are provided online: https://github.com/Lambda-3/DiscourseSimplification/tree/master/supplemental_material.

¹⁰<https://t1p.de/n6t9>

our TS approach to the annotations of the RST-DT corpus, we compare each split sentence to all the EDUs of the corresponding input sentence. For each pair, we search for the longest contiguous matching subsequence. Next, based on the size of the matched sequences, a similarity score between the two input strings is calculated. Each pair whose similarity score surpasses an empirically determined threshold of 0.65 is considered a match.

Constituency Type Classification To determine whether the hierarchical relationship that is assigned by our TS framework between a pair of simplified sentences is correct, we check if the hierarchy of its contextual layers corresponds to the nuclearity of the aligned text fragments of the RST-DT. For this purpose, we make use of the nuclearity status encoded in the annotations of this dataset. In addition, we compare the performance of our TS approach with that of a set of widely used sentence-level discourse parsers on this task.

Rhetorical Relation Identification To assess the performance of the rhetorical relation identification step, we determine the distribution of the relation types allocated by our TS approach when operating on the 7,284 input sentences of the RST-DT and compare it to the distribution of the manually annotated rhetorical relations of this corpus. Moreover, we examine for each matching sentence pair whether the rhetorical relation assigned by our TS framework equates the relation that connects the corresponding EDUs in the RST-DT dataset. For this purpose, we apply the more coarse-grained classification scheme from [Taboada and Das \(2013\)](#), who group the full set of 78 rhetorical relations that are used in the RST-DT corpus into 19 classes of relations that share rhetorical meaning. Finally, we analyze the performance of our framework on the relation labeling task in comparison to a number of discourse parser baselines.

3.1.2 Manual Analysis

To get a deeper insight into the accuracy of the semantic hierarchy established between the split components, the automatic evaluation described above is complemented by a manual analysis. Three human judges independently of each other assessed each decomposed sentence according to the following four criteria: (i) **Limitation to core information**: Is the simplified output limited to core information of the input sentence? (*yes - no - malformed*); (ii) **Soundness of the contextual propo-**

sition: Does the simplified sentence express a meaningful context fact? (*yes - no*); (iii) **Correctness of the context allocation**: Is the contextual sentence assigned to the parent sentence to which it refers? (*yes - no*); and (iv) **Properness of the identified semantic relationship**: Is the contextual sentence linked to its parent sentence via the correct semantic relation? (*yes - no - unspecified*).

The first three categories of our analysis address the correctness of the constituency type classification task, while the latter targets the rhetorical relation identification step. The annotation task was carried out on a random sample of 100 sentences from the OIE2016 Open IE benchmark ([Stanovsky and Dagan, 2016](#)).

3.2 Results

3.2.1 Automatic Metrics

Using the matching function described in Section 3.1.1, we obtained 1,827 matched sentence pairs, i.e. 11.74% of the pairs of simplified sentences were successfully mapped to a counterpart of EDUs from the RST-DT. The relatively low number of matches can be attributed to the fact that the text spans we compare have very different features.¹¹ As we are primarily interested in determining whether the constituency and relation labels that are assigned by our approach are correct, we will focus on precision in the following.¹²

Constituency Type Classification In 88.89% of the matched sentence pairs, the hierarchical relationship that is allocated between a pair of simplified sentences by our reference TS implementation DISSIM corresponds to the nuclearity status of the aligned EDUs from RST-DT, i.e. in case of a nucleus-nucleus relationship in RST-DT, both output sentences from DISSIM are assigned to the same context layer, while in case of a nucleus-satellite relationship the sentence mapped to the nucleus EDU is allocated to the context layer *cl*, whereas the sentence mapped to the satellite span is assigned to the subordinate context layer *cl+1*. The majority of the cases where our TS approach assigns a hierarchical relationship that differs from

¹¹For details, see Section F in the appendix.

¹²The fraction of labels that are successfully retrieved (i.e. recall) is of minor importance in our setting. In addition, this score might be biased, since a large proportion of EDUs from RST-DT is not mapped to a counterpart of simplified propositions in our experiments. Therefore, we refrain from reporting recall scores.

the nuclearity in the RST-DT corpus can be attributed to relative clauses (see Table 3).

Source	Mr. Volk, 55 years old, succeeds Duncan Dwight, who retired in September.	
	MAPPED SPAN 1	MAPPED SPAN 2
DISSIM	Volk succeeds Duncan Dwight.	Duncan Dwight retired in September.
Nuclearity Relation	context layer 0	context layer 1
	Elaboration	
RST-DT	Mr. Volk, 55 years old, succeeds Duncan Dwight, who retired in September.	
Nuclearity Relation	nucleus	nucleus
	Elaboration	

Table 3: Example.

	nuclearity	relation
DPLP (Ji and Eisenstein, 2014)	71.1	61.8
Feng and Hirst (2014)	71.0	58.2
2-Stage Parser (Wang et al., 2017)	72.4	59.7
Lin et al. (2019)	91.3	81.7
SPADE (Soricut and Marcu, 2003)	56.1	44.9
HILDA (Hernault et al., 2010)	59.7	48.2
PAR-s (Joty et al., 2015)	75.2	66.1
Lin et al. (2019)	(86.4)*	(77.5)*
DISSIM	88.9	69.5

Table 4: Precision of DISSIM and the discourse parser baselines, as reported by their authors. (*) In case of automatic discourse segmentation, for Lin et al. (2019) the F_1 -score is available only.

Table 4 displays the precision that the discourse parser baselines achieve on the 991 sentences of the RST-DT test set in distinguishing between nucleus and satellite spans (“nuclearity”). For the approaches in the upper part of the table, the authors report the systems’ performance when using gold EDU segmentation, while for those in the lower part the performance is indicated based on automatic segmentation, i.e. when they are fed the output of their respective discourse segmenter. Since our framework makes use of the simplified sentences that were generated in the previous step when setting up the semantic hierarchy, it is better comparable to the latter group. The figures show that in this case our approach outperforms all other systems in the constituency type classification task by a large margin of 13.7% at a minimum.¹³

Rhetorical Relation Identification Table 5 displays the frequency distribution of the 19 classes of rhetorical relations that were specified in Taboada and Das (2013). The ten most frequently occurring classes make up for 89.45% of the relations

that are present in the dataset. We decided to limit ourselves to these classes in the evaluation of the rhetorical relation identification step, with two exceptions. First, we did not take into account the “Topic-change” and “Same-unit” classes. Second, we merged the two highly related classes of “Cause” and “Explanation” into a single category.

RHET. RELATION	COUNT	PERCENT.	PRECISION
Elaboration	7,675	25.65%	0.5550
Joint	7,116	23.78%	0.6673
Attribution	2,984	9.97%	0.9601
Same-unit	2,788	9.32%	—
Contrast	1,522	5.09%	0.7421
Topic-change	1,315	4.39%	—
Explanation	966	3.21%	0.7037
Cause	754	2.52%	
Temporal	964	3.22%	0.7895
Background	897	2.30%	0.4459
			avg.: 0.6948

Evaluation (2.0%), Enablement (1.8%), Comparison (1.5%), Textual organization (1.2%), Condition (1.1%), Topic-comment (0.9%), Manner-means (0.7%), Summary (0.7%), Span (0.0%)

Table 5: Frequency distribution of the 19 classes of relations from Taboada and Das (2013) and the precision of DISSIM’s rhetorical relation identification step.

The right column in Table 5 displays the precision of our TS approach for each class of rhetorical relation when run over the sentences from RST-DT. The “Attribution” relation reaches by far the highest precision. The remaining relations, too, show decent scores, with a precision of around 70%. The only exception is “Background”. The difficulty with this type of relationship is that it signifies a very broad category that is not signalled by discourse markers and therefore hard to detect by our approach (Taboada and Das, 2013). With an average precision of 69.5% in the relation labeling task (see Table 4), our framework again surpasses all the discourse parser baselines under consideration when using automatic discourse segmentation.¹⁴

When comparing the distribution of the rhetorical relations that were identified by our TS approach on the source sentences from the RST-DT (see Figure 5 in Section F in the appendix) to that of the manually annotated gold relations displayed in Table 5, it turns out that there is a very high similarity between the two of them. However, it must be noted that in about a fifth of the cases, our TS

¹³A very recent approach to intra-sentential sentence parsing was proposed in Lin et al. (2019), achieving an F_1 -score of 86.4%. However, the authors do not report its precision.

¹⁴with the exception of Lin et al. (2019)’s parser, for which only the F_1 -score is reported by the authors, though. Hence, it is not directly comparable to the other approaches whose performance is analyzed based on their precision.

approach is not able to identify a rhetorical relation between a pair of decomposed sentences (“*Unknown*”). For the most part, this can be attributed to sentence pairs whose relation is not explicitly stated in the underlying source sentence. Since our approach is based on cue phrases, searching for discourse markers that explicitly signal rhetorical relations, it has difficulties in identifying relations that can merely be implied.

3.2.2 Manual Analysis

The results of the human evaluation are displayed in Table 6. The inter-annotator agreement was calculated using Fleiss’ κ (Fleiss, 1971). The figures indicate fair to substantial agreement between the three annotators, suggesting that the evaluation scores present a reliable result.

Category	Yes	No	Malf.	Unspec.	κ
Limitation to core information	68.2%	20.0%	11.9%	—	0.39
Soundness of the contextual proposition	83.1%	16.9%	—	—	0.51
Correctness of the context allocation	93.2%	6.8%	—	—	0.41
Properness of the semantic relationship	69.8%	7.0%	—	23.2%	0.69

Table 6: Results of the manual analysis.

In more than two out of three cases, the annotators marked the propositions that were classified as core sentences by our TS approach as correct, thus approving that they have a meaningful interpretation and that their content is truly restricted to core information of the underlying source sentence. Only about 12% of the simplified sentences are malformed according to our annotations. The remaining fifth of output core sentences was judged as being misclassified, i.e. they rather contribute less relevant background data than key information of the input. Regarding the soundness of the context propositions, only about 17% of the output propositions that were classified as context sentences were labelled as being inaccurate, while as many as 83% present proper contextual propositions, expressing a meaningful context fact that is asserted by the input and can be properly interpreted. Furthermore, 93% of the context sentences are assigned to their respective parent sentence, whereas only 6% of them are misallocated, according to the annotators’ labels. Finally, our evaluation revealed that our TS approach shows a decent performance for the rhetorical relation identification step, too. More than two-thirds of the sentence pairs are classified with the correct rhetorical relation, according to our

manual analysis. Only 7% of them are assigned an improper relation. However, in nearly a quarter of the cases, our TS approach was not able to identify a semantic relationship between the given pair of sentences. This can be explained by the fact that for this subtask, our framework follows a rather simplistic approach that is primarily based on cue phrases. Therefore, it fails to identify a semantic relationship whenever none of the specified keywords appears in the underlying input sentence. As a result, our approach provides very precise results. Covering only a small subset of rhetorical relations it lacks in completeness, though.

3.3 Extrinsic Evaluation

The fine-grained representation of complex sentences in the form of hierarchically ordered and semantically interconnected propositions may serve as an intermediate representation for downstream tasks. An application area that may benefit greatly from our approach as a preprocessing step is the task of Open IE (Banko et al., 2007).

Coverage and Accuracy We demonstrated that relational tuples in the form of predicate-argument structures can be extracted more easily from the simplified sentences (Cetto et al., 2018b,a). In fact, a comparative analysis with state-of-the-art Open IE systems (Fader et al., 2011; Mausam et al., 2012; Del Corro and Gemulla, 2013; Angeli et al., 2015; Stanovsky et al., 2016; Mausam, 2016; Gashteovski et al., 2017; Saha et al., 2017; Stanovsky et al., 2018) on a large benchmark corpus (Stanovsky and Dagan, 2016) revealed that their performance in terms of precision and recall is improved by up to 25% and 27%, respectively, when taking advantage of the split propositions instead of dealing with the complex source sentences (see Table 7).

System	Precision	Recall	F ₁	AUC
REVERB	-7.2%	+19.1%	+8.4%	+39.2%
OLLIE	+1.1%	-1.5%	-0.3%	-1.1%
ClausIE	+17.0%	-3.5%	+8.1%	+13.0%
Stanford	+25.0%	+27.2%	+25.5%	+35.5%
Open IE				
PropS	-6.1%	+16.9%	+4.5%	+12.4%
OpenIE-4	+10.0%	+8.6%	+9.4%	+19.6%
MinIE	+23.7%	-1.8%	+12.5%	+21.3%
OpenIE-5	+5.0%	+4.2%	+4.6%	+9.0%
RnnOIE	-15.0%	+0.9%	-8.3%	-14.1%

Table 7: Improvements in precision, recall, F₁ and AUC on the OIE2016 benchmark when using our reference implementation DISSIM as a pre-processing step.

Coherence Moreover, our TS approach enables existing Open IE systems to enrich their output with semantic information. By capturing intra-sentential rhetorical structures and hierarchical relationships between the extracted tuples, the shallow semantic representation of Open IE tuples in the form of isolated predicate-argument structures is extended. In that way, the semantic context of the individual tuples is preserved.¹⁵

4 Related Work

Discourse-level TS The vast majority of structural TS approaches do not take into account discourse-level aspects. However, two notable exceptions have to be mentioned. Siddharthan (2006) was the first to use discourse-aware cues in the simplification process. As opposed to our approach, though, where a semantic relationship is established for each simplified output sentence, only a comparatively low number of sentences is linked by such cue words. Another approach that operates on the level of discourse was proposed by Štajner and Glavaš (2017). It performs a semantically motivated content reduction by maintaining only those parts of a sentence that belong to factual event mentions. Our approach, on the contrary, aims to preserve all the information contained in the source.

Discourse Parsing The challenge of uncovering coherence structures in texts is pursued in the field of Discourse Parsing. It aims to identify discourse relations that hold between textual units in a document (Marcu, 1997). A well-established theory of text structure used in this area is RST. Here, textual coherence is explained by the existence of rhetorical relations that hold between adjacent text spans in a hierarchical structure. Approaches to detect rhetorical structure arrangements in texts range from early rule-based approaches (Marcu, 2000) to supervised data-driven models that were trained on annotated corpora such as the RST-DT (Feng and Hirst, 2014; Li et al., 2014; Lin et al., 2019).¹⁶

5 Conclusion

We presented a context-preserving TS approach that transforms structurally complex sentences into a hierarchical representation in the form of core

sentences and accompanying contexts that are semantically linked by rhetorical relations. In our experiments, we mapped the simplified sentences from our reference implementation DISSIM to the EDUs from RST-DT and showed that we obtain a very high precision of 89% for the constituency type classification and a decent score of 69% on average for the rhetorical relation identification. In the future, we plan to improve the latter step by extending our approach to also capture implicit relationships between the decomposed sentences.

References

- Omri Abend and Ari Rappoport. 2013. [Universal Conceptual Cognitive Annotation \(UCCA\)](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.
- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. [Leveraging linguistic structure for open domain information extraction](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.
- Nguyen Bach, Qin Gao, Stephan Vogel, and Alex Waibel. 2011. [TriS: A statistical sentence simplifier with log-linear models and margin-based discriminative training](#). In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 474–482, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Lynn Carlson and Daniel Marcu. 2001. [Discourse tagging reference manual](#). *ISI Technical Report ISI-TR-545*, 54:56.
- Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018a. [Graphene: a context-preserving open information extraction system](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 94–98, Santa Fe, New Mexico. Association for Computational Linguistics.
- Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018b. [Graphene](#):

¹⁵For details, see Section E in the appendix.

¹⁶Section G in the appendix elaborates on why it is not possible to simply use an RST parser for establishing the semantic hierarchy between the decomposed spans.

- Semantically-linked propositions in open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2300–2311, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Luciano Del Corro and Rainer Gemulla. 2013. [Clausie: Clause-based open information extraction](#). In *Proceedings of the 22nd International Conference on World Wide Web, WWW '13*, page 355–366, New York, NY, USA. Association for Computing Machinery.
- Richard Evans and Constantin Orăsan. 2019. [Identifying signs of syntactic complexity for rule-based sentence simplification](#). *Natural Language Engineering*, 25(1):69–119.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. [Identifying relations for open information extraction](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Richard Fay, editor. 1990. *Collins Cobuild English Grammar*. Collins.
- Vanessa Wei Feng and Graeme Hirst. 2014. [A linear-time bottom-up discourse parser with constraints and post-editing](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.
- Daniel Ferrés, Montserrat Marimon, Horacio Saggion, and Ahmed AbuRa'ed. 2016. [YATS: yet another text simplifier](#). In *Natural Language Processing and Information Systems - 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings*, volume 9612 of *Lecture Notes in Computer Science*, pages 335–342. Springer.
- J.L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.
- Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. [MinIE: Minimizing facts in open information extraction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.
- Michael Heilman and Noah A. Smith. 2010. Extracting simplified statements for factual question generation. In *Proceedings of the QG2010: The Third Workshop on Question Generation*, pages 11–20.
- Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. 2010. Hilda: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).
- 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, Baltimore, Maryland. Association for Computational Linguistics.
- Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2015. [CODRA: A novel discriminative framework for rhetorical analysis](#). *Computational Linguistics*, 41(3):385–435.
- Roger Levy and Galen Andrew. 2006. [Tregex and tsurgeon: tools for querying and manipulating tree data structures](#). In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).
- Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. [Recur-sive deep models for discourse parsing](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2061–2069, Doha, Qatar. Association for Computational Linguistics.
- Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. 2019. [A unified linear-time framework for sentence-level discourse parsing](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200, Florence, Italy. Association for Computational Linguistics.
- Jonathan Mallinson and Mirella Lapata. 2019. Controllable sentence simplification: Employing syntactic and lexical constraints. *arXiv preprint arXiv:1910.04387*.
- 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. 1997. [The rhetorical parsing of unrestricted natural language texts](#). In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–103, Madrid, Spain. Association for Computational Linguistics.
- Daniel Marcu. 2000. [The rhetorical parsing of unrestricted texts: a surface-based approach](#). *Computational Linguistics*, 26(3):395–448.
- Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. [Open language learning for information extraction](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

- Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, page 4074–4077. AAAI Press.
- Ruslan Mitkov and Horacio Saggin. 2018. [Text simplification](#).
- Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. [A survey on open information extraction](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019a. [DisSim: A discourse-aware syntactic text simplification framework for English and German](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 504–507, Tokyo, Japan. Association for Computational Linguistics.
- Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019b. [Transforming complex sentences into a semantic hierarchy](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3415–3427, Florence, Italy. Association for Computational Linguistics.
- Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, London.
- Horacio Saggin, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. [Making it simplext: Implementation and evaluation of a text simplification system for spanish](#). *ACM Transactions on Accessible Computing*, 6(4).
- Swarnadeep Saha and Mausam. 2018. [Open information extraction from conjunctive sentences](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Swarnadeep Saha, Harinder Pal, and Mausam. 2017. [Bootstrapping for numerical open IE](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics.
- Advait Siddharthan. 2002. An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71. IEEE.
- Advait Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.
- Advait Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.
- Advait Siddharthan and Angrosh Mandya. 2014. [Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules](#). In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731, Gothenburg, Sweden. Association for Computational Linguistics.
- Radu Soricut and Daniel Marcu. 2003. [Sentence level discourse parsing using syntactic and lexical information](#). In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 228–235.
- Sanja Štajner and Goran Glavaš. 2017. Leveraging event-based semantics for automated text simplification. *Expert systems with applications*, 82:383–395.
- Sanja Štajner and Maja Popovic. 2016. [Can text simplification help machine translation?](#) In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.
- Sanja Štajner and Maja Popović. 2018. [Improving machine translation of English relative clauses with automatic text simplification](#). In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 39–48, Tilburg, the Netherlands. Association for Computational Linguistics.
- Gabriel Stanovsky and Ido Dagan. 2016. [Creating a large benchmark for open information extraction](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.
- Gabriel Stanovsky, Jessica Fidler, Ido Dagan, and Yoav Goldberg. 2016. [Getting more out of syntax with props](#). *CoRR*, abs/1603.01648.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. [Supervised open information extraction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.
- Maite Taboada and Debopam Das. 2013. Annotation upon annotation: Adding signalling information to a corpus of discourse relations. *D&D*, 4(2):249–281.
- Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. [A two-stage parsing method for text-level discourse analysis](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188, Vancouver, Canada. Association for Computational Linguistics.

A Process of Development of the Transformation Patterns

One of the fundamental goals of our context-preserving TS approach is to decompose complex assertions into a set of self-contained minimal propositions. Intuitively, the simplified propositions are “*easy to read and understand, and arguably easily processed by computers*” (Bach et al., 2011). Below, we present a more formal specification of the minimality property we aim for in the resulting simplified sentences, both on a syntactic and a semantic level.

Minimality on the Syntactic Level In syntax, four types of sentence structures are distinguished: *simple sentences, compound sentences, complex sentences and compound-complex sentences* (Quirk et al., 1985). The goal of our approach is to transform a given source sentence into a set of *simple sentences*. A simple sentence is a sentence that comprises exactly one independent clause, i.e. a group of words that has both a subject (S) and a verb (V), and optionally an indirect or direct object (O), an adverbial (A) or a complement (C). It expresses a complete thought, i.e. some coherent piece of information, as the following example shows: “*I admire her reasoning.*”

Accordingly, the method we propose aims to split and rephrase complex multi-clause sentences into sequences of simple sentences that each contain exactly one independent clause. Fay (1990) distinguishes four central functional categories of clauses:

- **coordinate clauses** (1 rule),
- **adverbial clauses** (6 rules addressing pre- and postposed adverbial clauses as well as adverbial clauses of purpose introduced by the phrase “*to do*”),
- **relative clauses** (9 rules targeting both restrictive and non-restrictive relative clauses, lexicalized on the different types of relative pronouns) and
- **reported speech** (4 rules addressing both direct and reported speech with pre- and postposed attribution).

Thus, in order to divide complex assertions into single-clause elements, we specified, in a first step, a set of transformation patterns that target above-mentioned categories of clauses, converting complex and compound sentences into a set of simple

sentences.

However, simple sentences may still include optional constituents that render it overly complex. For instance, the two adverbials “*in Princeton*” and “*in 1955*” in the simple sentence “[*Albert Einstein*]_S [*died*]_V [*in Princeton*]_A [*in 1955*]_A.” specify additional contextual information that can be left out without producing ill-formed output. Rather, the remaining clause “*Albert Einstein died.*” still carries semantically meaningful information. Hence, in order to transform simple sentences into atomic semantic units, we need to further reduce them to their corresponding clause type. According to Quirk et al. (1985), clauses can be classified into seven different clause types based on the grammatical function of their constituents, as illustrated in Table 8 (Del Corro and Gemulla, 2013).

The clause type conveys the *minimal unit of coherent information* in the clause. Accordingly, if a constituent of a clause that is also part of its type is removed, the resulting clause does not carry semantically meaningful information any more (Del Corro and Gemulla, 2013). Hence, while constituents that belong to the clause type are essential components of the corresponding simple sentence, all other constituents are optional and can be discarded without leading to an incoherent or semantically meaningless output. Therefore, we do not only perform clausal disembedding. Instead, we go down to the level of individual phrases, with the goal of extracting and converting optional phrasal constituents into stand-alone sentences.

So far, a minimal proposition can be defined as a simple sentence that is reduced to its clause type by omitting all optional constituents, i.e. all elements that do not appear in the type of the underlying clause. However, transforming complex source sentences into simple sentences and trimming them to their clause types may still result in over-specified propositions due to overly complex subclausal units. For instance, in the simple sentence “[*Bell, a telecommunication company,*]_S [*makes and distributes*]_V [*electrical goods, computers and building products*]_O.” (of clause type SVO), the phrasal elements in all three positions are unnecessarily complex. While the subject contains an appositive phrase that further specifies the noun to which it refers (“*Bell*”), both the verb and the object include coordinated conjunctions that can be decomposed into separate elements, resulting in a set of clauses that present a much simpler syntax.

	CLAUSE TYPE	EXAMPLE
T_1	SV	<i>[Albert Einstein]_S [died]_V.</i>
T_2	SVA	<i>[Albert Einstein]_S [remained]_V [in Princeton]_A.</i>
T_3	SVC	<i>[Albert Einstein]_S [is]_V [smart]_C.</i>
T_4	SVO	<i>[Albert Einstein]_S [has won]_V [the Nobel Prize]_O.</i>
T_5	SVOO	<i>[The Royal Swedish Academy of Sciences]_S [gave]_V [Albert Einstein]_O [the Nobel Prize]_O.</i>
T_6	SVOA	<i>[The doorman]_S [showed]_V [Albert Einstein]_O [to his office]_A.</i>
T_7	SVOC	<i>[Albert Einstein]_S [declared]_V [the meeting]_O [open]_C.</i>

Table 8: The seven types of clauses. S: Subject, V: Verb, C: Complement, O: Object, A: Adverbial.

Consequently, we further simplify such utterances by extracting phrasal elements from the input and transforming them into stand-alone sentences. In total, following Heilman and Smith (2010), our TS approach addresses seven types of phrasal constructs:

- **coordinate verb phrases** (1 rule),
- **coordinate noun phrase lists** (2 rules targeting lists of noun phrases in subject and object position),
- **participial phrases** (4 rules addressing both restrictive and non-restrictive embedded, pre- and postposed participial phrases),
- **appositive phrases** (2 rules addressing restrictive and non-restrictive appositions),
- **prepositional phrases** (3 rules addressing prepositional phrases that act as complements of verb phrases and prepositional phrases that are offset by commas),
- **adjectival and adverbial phrases** (2 rules targeting pre- and postposed adjectival and adverbial phrases) and
- **lead noun phrases** (1 rule).

To sum up, by the notion of a minimal proposition we understand a simple sentence that has been broken down to its essential constituents, i.e. the elements that are part of its clause type, while also extracting a specified set of phrasal units.

Minimality on the Semantic Level From a semantic point of view, a text can be seen as a collection of events (also called “frames” or “scenes”) (Abend and Rappoport, 2013) that describe some activity, state or property. From a semantic perspective, the goal of our TS approach is to split sentences into separate frames, where each of them presents a single event. An event typically consists of a predicate, a set of arguments and optionally secondary relations. While the predicate (typically a verb, but nominal or adjectival predicates are also

possible) is the main determinant of what the event is about, arguments describe its participants (e.g., who and where). They represent core elements of a frame, i.e. essential components that make it unique and different from other frames. Secondary relations, in contrast, represent non-core elements that introduce additional relations, describing further event properties (e.g., when and how). In order to avoid over-specified propositions that are difficult to handle for downstream applications, such non-core elements of a frame are extracted and transformed into stand-alone sentences in our proposed TS approach.

To sum up, our TS approach aims to split sentences that present a complex structure into its basic semantic building blocks in the form of events. Complex sentences that contain several events are decomposed into separate frames, while non-core elements of a frame are extracted and transformed into stand-alone sentences. Hence, from a semantic point of view, a minimal proposition can be seen as an utterance expressing a single event consisting of a predicate and its core arguments.

B Execution Order of the Patterns

The execution order of the transformation patterns was determined by examining which sequence achieved the best simplification results in a manual qualitative analysis that was conducted on a random sample of 300 sentences from Wikipedia.

The execution order is given in Table 9. The 35 rules could be grouped into 17 classes.

The execution order of the transformation patterns was balanced for various criteria, including the frequency of the rules, their complexity and specificity, as well as susceptibility to errors. In general, the least error-prone rules are executed first (e.g., in a manual analysis, we determined that 99.1% of the coordinate clauses (1) were correctly split). During the transformation process, we then work our way to the more error-prone ones, such

ORDER	RULE GROUP
1	coordinate clauses
2	non-restrictive relative clauses
3	appositive phrases
4	preposed adverbial clauses and participial phrases
5	coordinate verb phrases
6	postposed adverbial clauses and participial phrases
7	reported speech with preposed attribution
8	postposed adverbial clauses
9	reported speech with postposed attribution
10	embedded participial phrases
11	restrictive relative clauses
12	prepositional phrases that act as complements of verb phrases
13	postposed participial phrases
14	adjectival/adverbial phrases
15	lead noun phrases
16	prepositional phrases that are offset by commas
17	lists of noun phrases

Table 9: Execution order of the patterns.

as the rules for decomposing lists of noun phrases (17), which can easily be confused with appositive phrases. Beyond that, the patterns become more and more complex towards the end of the list, since we need to check for special cases and possibilities of confusion, e.g. (12) and (17). It also becomes clear that we first operate on coarse-grained level, dividing up clauses (1-11), before we go down to the phrasal level (10-17), resulting in much more fine-grained splits. Moreover, specific rules are carried out early in the process, such as the rules for disembedding relative clauses (lexicalized on the relative pronouns, 2) and appositive phrases (based on named entities, 3). More general rules, on the other hand, tend to be executed later. Finally, there is a rough orientation on the frequency of the rules, starting with the ones that are triggered more often, and working towards the ones that occur less frequently in the transformation process.

To sum up, the execution order of the transformation patterns is based on the following criteria, which have been balanced against each other:

- (1) **susceptibility to errors**: from less to more error-prone
- (2) **granularity**: from clausal to phrasal units
- (3) **complexity**: from simple to complex
- (4) **specificity**: from specific to general
- (5) **frequency**: from frequent to infrequent

C Constituency Type Classification

In RST, each text span is specified as either a nucleus or a satellite. The *nucleus* span embodies the central piece of information and is comparable to what we denote a core sentence, whereas the

role of the *satellite* is to further specify the nucleus, corresponding to a context sentence in our case.

D Mapping of Cue Phrases to Rhetorical Relations

Table 10 lists the full set of cue phrases that serve as lexical features for the identification of rhetorical relations when establishing the semantic hierarchy between a pair of split sentences. It further shows to which rhetorical relation each of them is mapped.

In addition, *Spatial* relations are identified on the basis of named entities. *Attribution* relations are detected using a pre-defined list of verbs of reported speech and cognition (Carlson and Marcu, 2001).

E Merits for Downstream Applications

Coherence Previous work in the area of Open IE has mainly focused on the extraction of isolated relational tuples, ignoring the cohesive nature of texts where important contextual information is spread across clauses or sentences. Consequently, state-of-the-art Open IE approaches are prone to generating a loose arrangement of tuples that lack the expressiveness needed to infer the true meaning of complex assertions (Niklaus et al., 2018).

Our approach allows existing Open IE systems to enrich their output with semantic information. By leveraging the semantic hierarchy of minimal propositions generated by our discourse-aware TS approach, they are able to extract semantically typed relational tuples from complex source sentences. Thus, the shallow semantic representation of state-of-the-art Open IE approaches in the form

RHETORICAL RELATION	CUE PHRASES
Contrast	although, but, but now, despite, even though, even when, except when, however, instead, rather, still, though, thus, until recently, while, yet
List	and, in addition, in addition to, moreover
Disjunction	or
Cause	largely because, because, since
Result	as a result, as a result of
Temporal	after, and after, next, then, before, previously
Background	as, now, once, when, with, without
Condition	if, in case, unless, until
Elaboration	more provocatively, even before, for example, further, recently, since, since now, so, so far, where, whereby, whether
Explanation	simply because, because of, indeed, so, so that

Table 10: Mapping of cue phrases to rhetorical relations.

RnnOIE (stand-alone):			
(1)	(A fluoroscopic study;	known;	as an upper gastrointestinal series)
(2)	(caution with non water soluble contrast;	is;	mandatory as the usage of barium)
(3)	(as the usage;	of barium can impede;	surgical revision and lead)
(4)	(;	to increased;	post operative complications)}
RnnOIE (with pre-processing):			
(5)	#1 0 (A fluoroscopic study;	is;	typically, the next step in management)
(5a)	L:ELABORATION #2		
(5b)	L:CONTRAST #3		
(6)	#2 1 (This; fluoroscopic study is known;	as an upper gastrointestinal series)	
(7)	#3 0 (Caution with non water soluble;	is;	mandatory)
(7a)	L:CONTRAST #1		
(7b)	L:CONDITION #7		
(7c)	L:BACKGROUND #4		
(7d)	L:BACKGROUND #5		
(7e)	L:BACKGROUND #6		
(8)	#4 1 (The usage of barium;	can impede;	surgical revision)
(8a)	L:LIST #5		
(8b)	L:LIST #6		
(9)	#5 1 (The usage of barium;	can lead;	to increased post operative complications)
(9a)	L:LIST #4		
(9b)	L:LIST #6		
(10)	#6 1 (The usage of barium;	to increased;	post operative complications)
(10a)	L:LIST #4		
(10b)	L:LIST #5		
(11)	#7 1 (Volvulus;	is suspected;)

Figure 4: Comparison of the relational tuples extracted by RnnOIE (Stanovsky et al., 2018) *with* and *without* using our context-preserving TS approach as a preprocessing step.

of disconnected predicate-argument structures is extended, capturing intra-sentential rhetorical structures and hierarchical relationships between the relational tuples. In that way, the semantic context of the extracted tuples is preserved, allowing for a proper interpretation of complex assertions. See Figure 4 for an illustrative example.

F Evaluation

While the goal of our TS approach is to generate well-formed syntactically simplified sentences, the EDUs in the RST-DT are copied verbatim from the source, resulting in an output of varied length that is usually not grammatically sound. Moreover, in many cases, the EDUs mix multiple semantic units, whereas our approach aims to split the input into atomic components, with each of them expressing

a coherent and indivisible proposition.

G Discourse Parsing

The syntactic analysis we propose for establishing the semantic hierarchy between the decomposed spans is bound to the RST discourse markers. However, it is not possible to simply use an RST parser for this task. As illustrated in Figure 6, such a parser does not return grammatically sound sentences. Instead, it segments the input into basic textual units, so-called elementary discourse units (EDUs), which are copied verbatim from the source. In order to reconstruct proper sentences, rephrasing is required. For this purpose, amongst others, referring expressions have to be identified, and phrases have to be rearranged and inflected. Moreover, the textual units resulting from the segmentation pro-

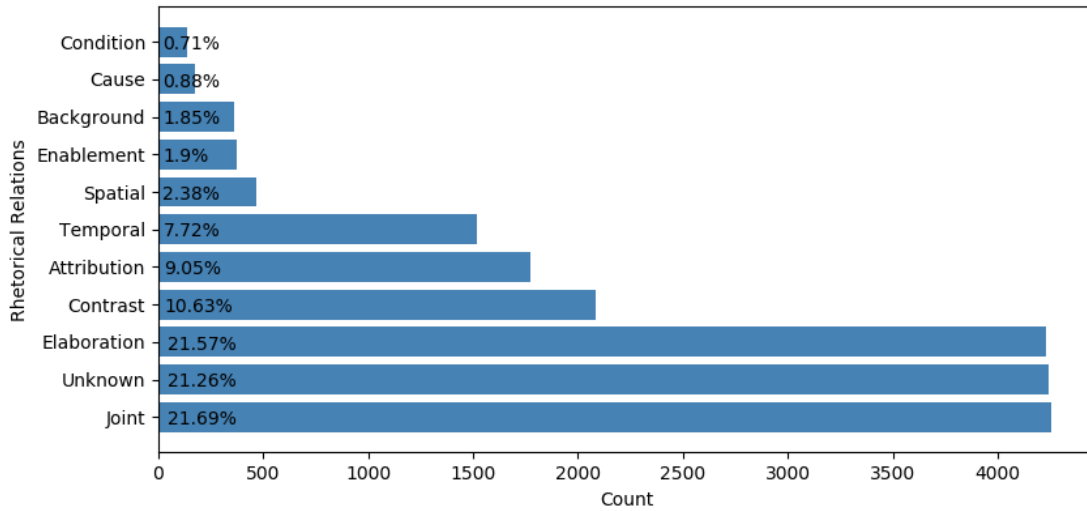


Figure 5: Distribution of the rhetorical relations.

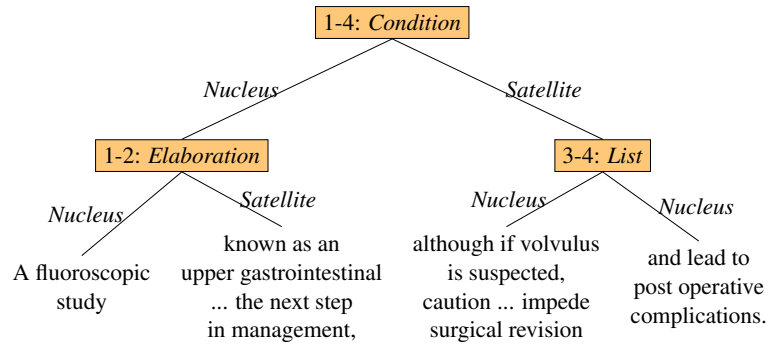


Figure 6: Rhetorical structure tree of our example sentence, generated using the RST parser proposed in Ji and Eisenstein (2014). The leaves correspond to *EDUs*, while each node is characterized by its *nuclearity* and a *rhetorical relation* between adjacent text spans.

cess are too coarse-grained for our purpose, since RST parsers mostly operate on clausal level. The goal of our approach, though, is to split the input into minimal semantic units, which requires to go down to the phrasal level in order to produce a much more fine-grained output in the form of minimal propositions.