

# Transforming Complex Sentences into a Semantic Hierarchy

Christina Niklaus<sup>1,3</sup>, Matthias Cetto<sup>1</sup>, André Freitas<sup>2</sup>, and Siegfried Handschuh<sup>1,3</sup>

<sup>1</sup> University of St.Gallen

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

<sup>2</sup> University of Manchester

andre.freitas@manchester.ac.uk

<sup>3</sup> University of Passau

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

## Abstract

We present an approach for recursively splitting and rephrasing complex English sentences into a novel semantic hierarchy of simplified sentences, with each of them presenting a more regular structure that may facilitate a wide variety of artificial intelligence tasks, such as machine translation (MT) or information extraction (IE). Using a set of hand-crafted transformation rules, input sentences are recursively transformed into a two-layered hierarchical representation in the form of core sentences and accompanying contexts that are linked via rhetorical relations. In this way, the semantic relationship of the decomposed constituents is preserved in the output, maintaining its interpretability for downstream applications. Both a thorough manual analysis and automatic evaluation across three datasets from two different domains demonstrate that the proposed syntactic simplification approach outperforms the state of the art in structural text simplification. Moreover, an extrinsic evaluation shows that when applying our framework as a preprocessing step the performance of state-of-the-art Open IE systems can be improved by up to 346% in precision and 52% in recall. To enable reproducible research, all code is provided online.

## 1 Introduction

Text Simplification (TS) is defined as the process of reducing the linguistic complexity of natural language (NL) text by utilizing a more readily accessible vocabulary and sentence structure. Its goal is to improve the readability of a text, making information easier to comprehend for people with reduced literacy, such as non-native speakers (Paetzold and Specia, 2016), aphasics (Carroll et al., 1998), dyslexics (Rello et al., 2013) or deaf persons (Inui et al., 2003). However, not only human readers may benefit from TS. Previous work

has established that applying TS as a preprocessing step can improve the performance of a variety of natural language processing (NLP) tasks, such as Open IE (Saha and Mausam, 2018; Cetto et al., 2018), MT (Štajner and Popovic, 2016, 2018), Relation Extraction (Miwa et al., 2010), Semantic Role Labeling (Vickrey and Koller, 2008), Text Summarization (Siddharthan et al., 2004; Bouayad-Agha et al., 2009), Question Generation (Heilman and Smith, 2010; Bernhard et al., 2012), or Parsing (Chandrasekar et al., 1996; Jonnalagadda et al., 2009).

Linguistic complexity stems from the use of either a difficult vocabulary or sentence structure. Therefore, TS is classified into two categories: *lexical simplification* and *syntactic simplification*. Through substituting a difficult word or phrase with a more comprehensible synonym, the former primarily addresses a human audience. Most NLP systems, on the contrary, derive greater benefit from syntactic simplification, which focuses on identifying grammatical complexities in a sentence and converting these structures into simpler ones, using a set of text-to-text rewriting operations. *Sentence splitting* plays a major role here: it divides a sentence into several shorter components, with each of them presenting a simpler and more regular structure that is easier to process for downstream applications.

Many different methods for addressing the task of TS have been presented so far. As noted in Štajner and Glavaš (2017), data-driven approaches outperform rule-based systems in the area of lexical simplification (Glavaš and Štajner, 2015; Paetzold and Specia, 2016; Nisioi et al., 2017; Zhang and Lapata, 2017). In contrast, the state-of-the-art syntactic simplification approaches are rule-based (Siddharthan and Mandya, 2014; Ferrés et al., 2016; Saggion et al., 2015), providing more grammatical output and covering a wider range of syn-

tactic transformation operations, however, at the cost of being very conservative, often to the extent of not making any changes at all. Acknowledging that existing TS corpora (Zhu et al., 2010; Coster and Kauchak, 2011; Xu et al., 2015) are inappropriate for learning to decompose sentences into shorter, syntactically simplified components, as they contain only a small number of split examples, Narayan et al. (2017) lately compiled the first TS dataset that explicitly addresses the task of sentence splitting. Using this corpus, several encoder-decoder models (Bahdanau et al., 2014) are proposed for breaking down a complex source into a set of sentences with a simplified structure. Aharoni and Goldberg (2018) further explore this idea, augmenting the presented neural models with a copy mechanism (Gu et al., 2016; See et al., 2017).

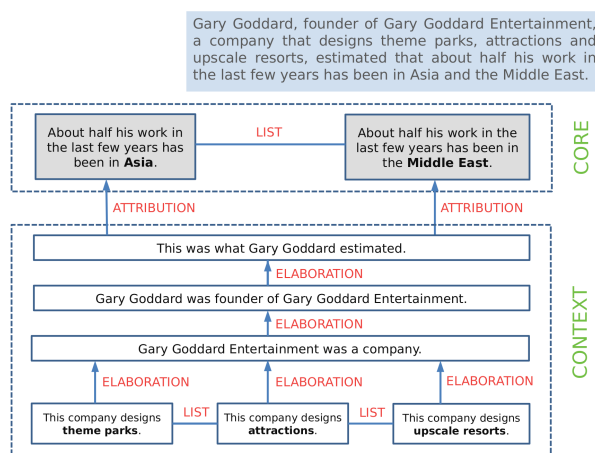


Figure 1: Example of the output that is generated by our proposed TS approach. A complex input sentence is transformed into a semantic hierarchy of simplified sentences in the form of minimal, self-contained propositions that are linked via rhetorical relations.

In contrast to above-mentioned end-to-end neural approaches, we followed a more systematic approach. First, we performed an in-depth study of the literature on syntactic sentence simplification, followed by a thorough linguistic analysis of the syntactic phenomena that need to be tackled in the sentence splitting task. Next, we materialized our findings into a small set of 35 hand-crafted transformation rules that decompose sentences with a complex linguistic structure into shorter constituents that present a simpler and grammatically sound structure, leveraging downstream semantic applications whose predictive quality deteriorates with sentence length and complexity.

One of our major goals was to *overcome the*

*conservatism* exhibited by state-of-the-art syntactic TS approaches, i.e. their tendency to retain the input sentence rather than transforming it. For this purpose, we decompose each source sentence into minimal semantic units and turn them into self-contained propositions. In that way, we provide a fine-grained output that is easy to process for subsequently applied NLP tools. Another major drawback of the structural TS approaches described so far is that they do not *preserve the semantic links between the individual split components*, resulting in a set of incoherent utterances. Consequently, important contextual information is lost, impeding the interpretability of the output for downstream semantic tasks. To prevent this, we establish a contextual hierarchy between the split components and identify the semantic relationship that holds between them. An example of the resulting output is displayed in Figure 1.

## 2 Related Work

To date, three main classes of techniques for syntactic TS with a focus on the task of sentence splitting have been proposed. The first uses a set of syntax-based hand-crafted transformation rules to perform structural simplification operations, while the second exploits machine learning (ML) techniques where the model learns simplification rewrites automatically from examples of aligned complex source and simplified target sentences. In addition, approaches based on the idea of decomposing a sentence into its main semantic constituents using a semantic parser were described.

### 2.1 Syntax-driven Rule-based Approaches

The line of work on structural TS starts with Chandrasekar et al. (1996), who manually defines a set of rules to detect points where sentences may be split, such as relative pronouns or conjunctions, based on chunking and dependency parse representations. Siddharthan (2002) presents a pipelined architecture for a simplification framework that extracts a variety of clausal and phrasal components from a source sentence and transforms them into stand-alone sentences using a set of hand-written grammar rules based on shallow syntactic features.

More recently, Siddharthan and Mandya (2014) propose RegenT, a hybrid TS approach that combines an extensive set of 136 hand-written gram-

mar rules defined over dependency tree structures for tackling 7 types of linguistic constructs with a much larger set of automatically acquired rules for lexical simplification. Taking a similar approach, Ferrés et al. (2016) describe a linguistically-motivated rule-based TS approach called YATS, which relies on part-of-speech tags and syntactic dependency information to simplify a similar set of linguistic constructs, using a set of only 76 hand-crafted transformation patterns in total. These two state-of-the-art rule-based structural TS approaches primarily target reader populations with reading difficulties, such as people suffering from dyslexia, aphasia or deafness. According to Siddharthan (2014), those groups most notably benefit from splitting long sentences that contain clausal constructions. Consequently, simplifying clausal components is the main focus of the proposed TS systems of this category.

Finally, Štajner and Glavaš (2017) present LEXEV and EVLEX, which combine a syntactic simplification approach that uses an even smaller set of 11 hand-written rules to perform sentence splitting and deletion of irrelevant sentences or sentence parts with an unsupervised lexical simplifier based on word embeddings (Glavaš and Štajner, 2015).

## 2.2 Approaches based on Semantic Parsing

While the TS approaches described above are based on syntactic information, there are a variety of methods that use semantic structures for sentence splitting. These include the work of Narayan and Gardent (2014) and Narayan and Gardent (2016), who propose a framework that takes semantically-shared elements as the basis for splitting and rephrasing a sentence. It first generates a semantic representation of the input to identify splitting points in the sentence. In a second step, the split components are then rephrased by completing them with missing elements in order to reconstruct grammatically sound sentences. Lately, with DSS, Sulem et al. (2018c) describe another semantic-based structural simplification framework that follows a similar approach.

## 2.3 Data-driven Approaches

More recently, data-driven approaches for the task of sentence splitting emerged. Narayan et al. (2017) propose a set of sequence-to-sequence models trained on the WebSplit corpus, a dataset of over one million tuples that map a single com-

plex sentence to a sequence of structurally simplified sentences. Aharoni and Goldberg (2018) further explore this idea, augmenting the presented neural models with a copy mechanism. Though outperforming the models used in Narayan et al. (2017), they still perform poorly compared to previous state-of-the-art rule-based syntactic simplification approaches. In addition, Botha et al. (2018) observed that the sentences from the WebSplit corpus contain fairly unnatural linguistic expressions using only a small vocabulary. To overcome this limitation, they present a scalable, language-agnostic method for mining training data from Wikipedia edit histories, providing a rich and varied vocabulary over naturally expressed sentences and their extracted splits. When training the best-performing model of Aharoni and Goldberg (2018) on this new split-and-rephrase dataset, they achieve a strong improvement over prior best results from Aharoni and Goldberg (2018). However, due to the uniform use of a single split per source sentence in the training set, each input sentence is broken down into two output sentences only. Consequently, the resulting simplified sentences are still comparatively long and complex.

## 3 Recursive Sentence Splitting

We present DISSIM, a recursive sentence splitting approach that creates a semantic hierarchy of simplified sentences.<sup>1</sup> The goal of our approach is to **generate an intermediate representation that presents a simple and more regular structure which is easier to process for downstream semantic applications and may support a faster generalization in ML tasks**. For this purpose, we *cover a wider range of syntactic constructs* (10 in total) than state-of-the-art rule-based syntactic frameworks. In particular, our approach is not limited to breaking up clausal components, but also splits and rephrases a variety of phrasal elements, resulting in a much more fine-grained output where each proposition represents a minimal semantic unit that is typically composed of a simple subject-predicate-object structure. Though tackling a larger set of linguistic constructs, our framework operates on a *much smaller set of only 35 manually defined rules* as compared to existing syntax-driven rule-based approaches.

<sup>1</sup>The source code of our framework is available under <https://github.com/Lambda-3/DiscourseSimplification>.

With the help of the transformation patterns that we specified, source sentences that present a complex linguistic form are transformed into clean, compact structures by disembedding clausal and phrasal components that contain only supplementary information. These elements are then transformed into independent sentences. In that way, the source sentence is reduced to its key information (“*core sentence*”) and augmented with a number of associated *contextual sentences* that disclose additional information about it, resulting in a novel hierarchical representation in the form of core sentences and accompanying contexts. Moreover, we identify the rhetorical relations by which core sentences and their associated contexts are connected in order to preserve their semantic relationship. The resulting representation of the source text, which we will call a “*discourse tree*” in the following, can then be used to facilitate a variety of artificial intelligence tasks, such as text summarization, MT, IE or opinion mining, among other.

### 3.1 Transformation Stage

The structural TS framework that we propose takes a sentence as input and performs a recursive transformation stage that is based upon 35 hand-crafted grammar rules. Each rule defines how to *split up and rephrase* the input into structurally simplified sentences (subtask 1), establish a *contextual hierarchy* between the split components (subtask 2) and identify the *semantic relationship* that holds between those elements (subtask 3).

The transformation patterns are based on syntactic and lexical features that can be derived from a sentence’s phrase structure. They were heuristically determined in a rule engineering process whose main goal was to provide a best-effort set of patterns, targeting the challenge of being applied in a recursive fashion and to overcome biased or incorrectly structured parse trees. We empirically determined a fixed execution order of the rules by examining which sequence achieved the best simplification results in a manual qualitative analysis conducted on a development test set of 100 randomly sampled Wikipedia sentences. The grammar rules are applied recursively in a top-down fashion on the source sentence, until no more simplification pattern matches. In that way, the input is turned into a discourse tree, consisting of a **set of hierarchically ordered and semantically**

**interconnected sentences that present a simplified syntax.** Table 2 displays some examples of our transformation patterns,<sup>2</sup> which are specified in terms of Tregex patterns.<sup>3</sup>

	CLAUSAL/PHRASAL TYPE	# RULES
<b>Clausal disembedding</b>		
1	Coordinate clauses	1
2	Adverbial clauses	6
3a	Relative clauses (non-defining)	8
3b	Relative clauses (defining)	5
4	Reported speech	4
<b>Phrasal disembedding</b>		
5	Coordinate verb phrases (VPs)	1
6	Coordinate noun phrases (NPs)	2
7a	Appositions (non-restrictive)	1
7b	Appositions (restrictive)	1
8	Prepositional phrases (PPs)	3
9	Adjectival and adverbial phrases	2
10	Lead NPs	1
	Total	35

Table 1: Linguistic constructs addressed by DISSIM.

**Subtask 1: Sentence Splitting and Rephrasing.** Each transformation rule takes a sentence’s phrasal parse tree<sup>4</sup> as input and encodes a pattern that, in case of a match, will extract textual parts from the tree. The decomposed text spans, as well as the remaining text span are then transformed into new stand-alone sentences. In order to ensure that the resulting simplified output is grammatically sound, some of the extracted text spans are combined with their corresponding referents from the main sentence or appended to a simple phrase (e.g. “*This is*”). In that way, the simplification rules encode both the splitting points and rephrasing procedure for reconstructing proper sentences. Both coordinate and subordinate clauses, as well as various types of phrasal elements are addressed by our TS approach. Table 1 provides an overview of the linguistic constructs that are tackled, including the number of transformation patterns that were specified for the respective syntactic phenomenon.

For a better understanding of the splitting and rephrasing procedure, Figure 2 visualizes the application of the first grammar rule that matches the given input sentence. The upper part of the box represents the complex input, which is matched against the simplification pattern. The lower part

<sup>2</sup>For reproducibility purposes, the complete set of transformation patterns is available under [https://github.com/Lambda-3/DiscourseSimplification/tree/master/supplemental\\_material](https://github.com/Lambda-3/DiscourseSimplification/tree/master/supplemental_material).

<sup>3</sup>See Levy and Andrew (2006) for details on the rule syntax.

<sup>4</sup>generated by Stanford’s pre-trained lexicalized parser (Socher et al., 2013)

RULE	TREGEX PATTERN	EXTRACTED SENTENCE
<b>SharedNPPostCoordinationExtractor</b> (for coordinate verb phrases)	ROOT <<: (S < (NP \$. (VP < +(VP) (VP > VP \$. VP ))))	NP + VP .
<b>SubordinationPreExtractor</b> (for adverbial clauses with pre-posed subordinative clauses)	ROOT <<: (S < (SBAR < ( S < (NP \$. VP) ) \$. (NP \$. VP)))	S < (NP \$. VP) .

Table 2: A selection of transformation rule patterns. A boxed pattern represents the part that is extracted from the input sentence. An underlined pattern designates its referent. A pattern in bold will be deleted from the remaining part of the input.

then depicts the result of the transformation operation.

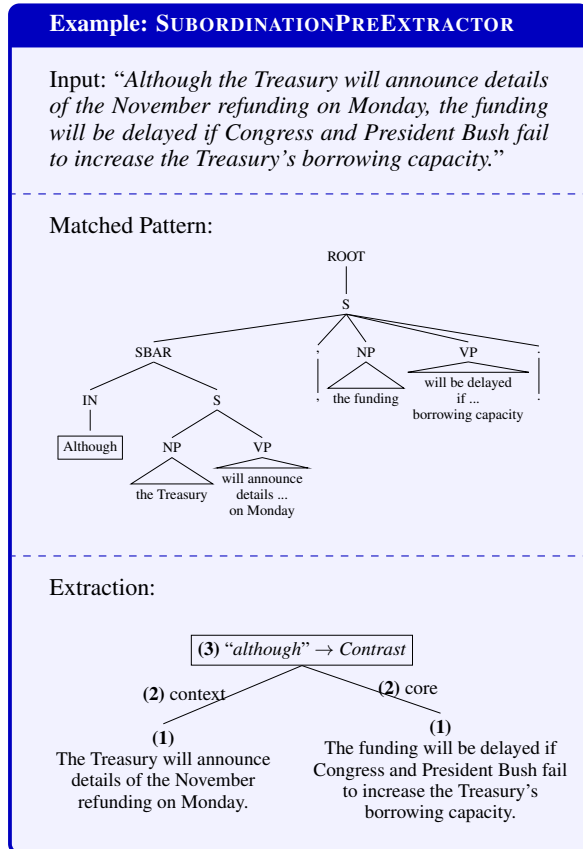


Figure 2: **(Subtask 1)** The source sentence is split up and rephrased into a set of syntactically simplified sentences. **(Subtask 2)** Then, the split sentences are connected with information about their constituency type to establish a contextual hierarchy between them. **(Subtask 3)** Finally, by identifying and classifying the rhetorical relations that hold between the simplified sentences, their semantic relationship is restored which can be used to inform downstream applications.

### Subtask 2: Constituency Type Classification.

Each split will create two or more sentences with a simplified syntax. In order to establish a contextual hierarchy between them, we connect them with information about their constituency type. According to Fay (1990), clauses can be related

to one another in two ways: First, there are parallel clauses that are linked by coordinating conjunctions, and second, clauses may be embedded inside another, introduced by subordinating conjunctions. The same applies to phrasal elements. Since the latter commonly express minor information, we denote them *context sentences*. In contrast, the former are of equal status and typically depict the key information contained in the input. Therefore, they are called *core sentences* in our approach. To differentiate between those two types of constituents, the transformation patterns encode a simple syntax-based approach where subordinate clauses and phrasal elements are classified as *context sentences*, while coordinate clauses/phrases are labelled as *core*.<sup>5</sup>

### Subtask 3: Rhetorical Relation Identification.

Finally, we aim to determine intra-sentential semantic relationships in order to restore semantic relations between the disembedded components. For this purpose, we identify and classify the rhetorical relations that hold between the simplified sentences, making use of both syntactic and lexical features which are encoded in the transformation patterns. While syntactic features are manifested in the phrasal composition of a sentence’s parse tree, lexical features are extracted from the parse tree in the form of *cue phrases*. The determination of potential cue words and their positions in specific syntactic environments is based on the work of Knott and Dale (1994). The extracted cue phrases are then used to infer the type of rhetorical relation. For this task we utilize a predefined list of rhetorical cue words adapted from the work of Taboada and Das (2013), which assigns them to the relation that they most likely trigger. For example, the transformation rule in Figure 2 spec-

<sup>5</sup>This approach roughly relates to the concept of nuclearity in Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), which specifies each text span as either a *nucleus* or a *satellite*. The nucleus span embodies the central piece of information, whereas the role of the satellite is to further specify the nucleus.

ifies that “although” is the cue word here, which is mapped to a “Contrast” relationship.

### 3.2 Final Discourse Tree

The leaf nodes resulting from the first simplification pass are recursively simplified in a top-down approach. When no more transformation rule matches, the algorithm stops. The final discourse tree for the example sentence of Figure 2 is shown in Figure 3.

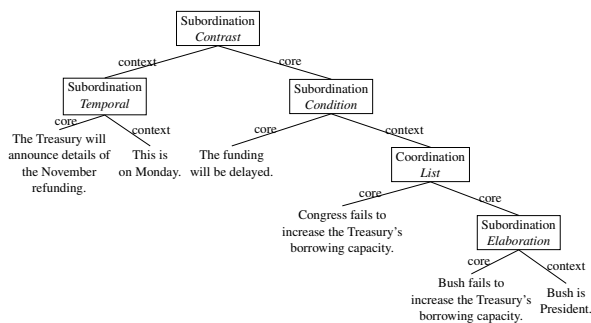


Figure 3: Final discourse tree of the example sentence.

## 4 Experimental Setup

To compare the performance of our TS approach with state-of-the-art syntactic simplification systems, we evaluated DISSIM with respect to the *sentence splitting task* (subtask 1). The evaluation of the rhetorical structures (subtasks 2 and 3) will be subject of future work.

**Corpora.** We conducted experiments on three commonly used simplification corpora from two different domains. The first dataset we used was Wikilarge, which consists of 359 sentences from the PWKP corpus (Xu et al., 2016). Moreover, to demonstrate domain independence, we compared the output generated by our TS approach with that of the various baseline systems on the Newsela corpus (Xu et al., 2015), which is composed of 1077 sentences from newswire articles. In addition, we assessed the performance of our simplification system using the 5000 test sentences from the WikiSplit benchmark (Botha et al., 2018), which was mined from Wikipedia edit histories.

**Baselines.** We compared our DISSIM approach against several state-of-the-art baseline systems that have a strong focus on syntactic transformations through explicitly modeling splitting operations. For Wikilarge, these include (i) DSS; (ii) SENTS (Sulem et al., 2018c), which is an extension of DSS that runs the split sentences through

the NTS system (Nisioi et al., 2017); (iii) HYBRID (Narayan and Gardent, 2014); (iv) YATS; and (v) RegenT. In addition, we report evaluation scores for the complex input sentences, which allows for a better judgment of system conservatism, and the corresponding simple reference sentences. With respect to the Newsela dataset, we considered the same baseline systems, with the exceptions of DSS and SENTS, whose outputs were not available. Finally, regarding the WikiSplit corpus, we restricted the comparison to the best-performing system in Botha et al. (2018), Copy512, which is a sequence-to-sequence neural model augmented with a copy mechanism and trained over the WikiSplit dataset.

**Automatic Evaluation.** The automatic metrics that were calculated in the evaluation procedure comprise a number of basic statistics, including (i) the average sentence length of the simplified sentences in terms of the average number of tokens per output sentence (#T/S); (ii) the average number of simplified output sentences per complex input (#S/C); (iii) the percentage of sentences that are copied from the source without performing any simplification operation (%SAME), serving as an indicator for system conservatism; and (iv) the averaged Levenshtein distance from the input ( $LD_{SC}$ ), which provides further evidence for a system’s conservatism. Furthermore, in accordance with prior work on TS, we report average BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016) scores for the rephrasings of each system.<sup>6</sup> Finally, we computed the SAMSA and SAMSA<sub>abl</sub> score of each system, which are the first metrics that explicitly target syntactic aspects of TS (Sulem et al., 2018b).

**Manual Analysis.** Human evaluation is carried out on a subset of 50 randomly sampled sentences per corpus by 2 non-native, but fluent English speakers who rated each input-output pair according to three parameters: grammaticality (G), meaning preservation (M) and structural simplicity (S) (see Section A of the appendix).

In order to get further insights into the quality of our implemented simplification patterns, we performed an extensive *qualitative analysis of the 35 hand-crafted transformation rules*, comprising a

<sup>6</sup>For the computation of the BLEU and SARI scores we used the implementation of Nisioi et al. (2017) which is available under <https://github.com/senisioi/NeuralTextSimplification>.

manual recall-based analysis of the simplification patterns, and a detailed error analysis.

**Usefulness.** Since the DISSIM framework that we propose is aimed at serving downstream semantic applications, we measure if an improvement in the performance of NLP tools is achieved when using our TS approach as a preprocessing step. For this purpose, we chose the task of Open IE (Banko et al., 2007) and determine whether such systems benefit from the sentence splitting approach presented in this work.

## 5 Results and Discussion

**Automatic Evaluation.** The upper part of Table 3 reports the results that were achieved on the 359 sentences from the Wikilarge corpus, using a set of automatic metrics. Transforming each sentence of the dataset, our DISSIM approach reaches the highest splitting rate among the TS systems under consideration, together with HYBRID, DSS and SENTS. With 2.82 split sentences per input on average, our framework outputs by a large margin the highest number of structurally simplified sentences per source. Moreover, consisting of 11.01 tokens on average, the DISSIM approach returns the shortest sentences of all systems. The relatively high word-based Levenshtein distance of 11.90 confirms previous findings.

With regard to SARI, our DISSIM framework (35.05) again outperforms the baseline systems. However, it is among the systems with the lowest BLEU score (63.03). Though, Sulem et al. (2018a) recently demonstrated that BLEU is inappropriate for the evaluation of TS approaches when sentence splitting is involved, since it negatively correlates with structural simplicity, thus penalizing sentences that present a simplified syntax, and presents no correlation with the grammaticality and meaning preservation dimensions. For this reason, we only report these scores for the sake of completeness and to match past work. According to Sulem et al. (2018b), the recently proposed SAMSA and SAMSA<sub>abl</sub> scores are better suited for the evaluation of the sentence splitting task. With a score of 0.67, the DISSIM framework shows the best performance for SAMSA, while its score of 0.84 for SAMSA<sub>abl</sub> is just below the one obtained by the RegenT system (0.85).<sup>7</sup>

<sup>7</sup>According to Sulem et al. (2018b), SAMSA highly correlates with human judgments for S and G, while SAMSA<sub>abl</sub>

The results on the Newsela dataset, depicted in the middle part of Table 3, support our findings on the Wikilarge corpus, indicating that our TS approach can be applied in a domain independent manner. The lower part of Table 3 illustrates the numbers achieved on the WikiSplit dataset. Though the Copy512 system beats our approach in terms of BLEU and SARI, the remaining scores are clearly in favour of the DISSIM system.

**Manual Analysis.** The results of the human evaluation are displayed in Table 4. The inter-annotator agreement was calculated using Cohen’s  $\kappa$ , resulting in rates of 0.72 (G), 0.74 (M) and 0.60 (S). The assigned scores demonstrate that our DISSIM approach outperforms all other TS systems in the S dimension. With a score of 1.30 on the Wikilarge sample sentences, it is far ahead of the baseline approaches, with HYBRID (0.86) coming closest. However, this system receives the lowest scores for G and M. RegenT obtains the highest score for G (4.64), while YATS is the best-performing approach in terms of M (4.60). However, with a rate of only 0.22, it achieves a low score for S, indicating that the high score in the M dimension is due to the conservative approach taken by YATS, resulting in only a small number of simplification operations. This explanation also holds true for RegenT’s high mark for G. Still, our DISSIM approach follows closely, with a score of 4.50 for M and 4.36 for G, suggesting that it obtains its goal of returning fine-grained simplified sentences that achieve a high level of grammaticality and preserve the meaning of the input. Considering the average scores of all systems under consideration, our approach is the best-performing system (3.39), followed by RegenT (3.16). The human evaluation ratings on the Newsela and WikiSplit sentences show similar results, again supporting the domain independence of our proposed approach.

The results of the recall-based qualitative analysis of the transformation patterns, together with the findings of the error analysis are illustrated in Section B of the appendix in Tables 9 and 10. Concerning the quality of the implemented simplification rules, the percentage of sentences that were correctly split was approaching 100% for coordinate and adverbial clauses, and exceeded 80% on average.

achieves the highest correlation for M.

	#T/S	#S/C	%SAME	LD <sub>SC</sub>	BLEU	SARI	SAMSA	SAMSA <sub>abl</sub>
<b>359 test sentences from the Wikilarge corpus</b>								
Complex	22.06	1.03	100	0.00	94.25	32.53	0.59	0.96
Simple reference	20.19	1.14	0.00	7.14	99.48	43.09	0.48	0.78
DISSIM	<b>11.01</b>	<b>2.82</b>	<b>0.00</b>	11.90	63.03	<b>35.05</b>	<b>0.67</b>	0.84
DSS	12.91	1.87	<b>0.00</b>	8.14	74.42	34.32	0.64	0.75
SENTS	14.17	1.09	<b>0.00</b>	<b>13.79</b>	54.37	29.76	0.40	0.58
HYBRID	13.44	1.03	<b>0.00</b>	13.04	48.97	26.19	0.47	0.76
YATS	18.83	1.40	18.66	4.44	73.07	33.03	0.56	0.80
RegenT	18.20	1.45	41.50	3.77	<b>82.49</b>	32.41	0.61	<b>0.85</b>
<b>1077 test sentences from the Newsela corpus</b>								
Complex	23.34	1.01	100	0.00	20.91	9.84	0.49	0.96
Simple reference	12.81	1.01	0.00	16.25	100	91.13	0.25	0.46
DISSIM	<b>11.20</b>	<b>2.96</b>	<b>0.00</b>	13.00	14.54	<b>49.00</b>	<b>0.57</b>	0.84
HYBRID	12.49	1.02	<b>0.00</b>	<b>13.46</b>	14.42	40.34	0.38	0.74
YATS	18.71	1.42	16.16	5.03	73.07	36.88	0.50	0.83
RegenT	16.74	1.61	33.33	5.03	<b>18.96</b>	32.83	0.55	<b>0.85</b>
<b>5000 test sentences from the WikiSplit corpus</b>								
Complex	32.01	1.10	100	0.00	74.28	29.91	0.37	0.95
Simple reference	18.14	2.08	0.00	7.48	100	94.71	0.49	0.75
DISSIM	<b>11.91</b>	<b>4.09</b>	<b>0.76</b>	<b>19.10</b>	51.96	39.33	<b>0.54</b>	<b>0.84</b>
Copy512	16.55	2.08	13.30	2.39	<b>76.42</b>	<b>61.51</b>	0.51	0.78

Table 3: Automatic evaluation results.

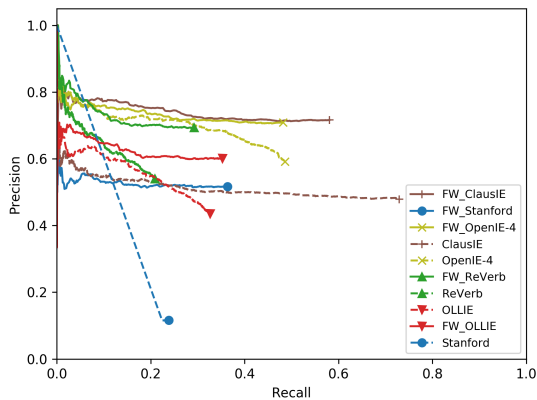


Figure 4: Performance of state-of-the-art Open IE systems *with* (solid lines) and *without* (dashed lines) sentence splitting as a preprocessing step.

System	Precision	Recall	AUC
Stanford Open IE	+ <b>346%</b>	+ <b>52%</b>	+ <b>597%</b>
REVERB	+ 28%	+ 40%	+ 57%
OLLIE	+ 38%	+ 8%	+ 20%
ClausIE	+ 50%	- 20%	+ 15%
OpenIE-4	+ 20%	- 1%	+ 3%

Table 5: Improvements when using DISSIM as a preprocessing step.

**Usefulness.** To investigate whether our proposed structural TS approach is able to improve the performance of downstream NLP tasks, we compare the performance of a number of state-of-the-art Open IE systems, including ClausIE (Del Corro and Gemulla, 2013), OpenIE-4 (Mausam, 2016), REVERB (Fader et al., 2011), OLLIE (Mausam et al., 2012) and Stanford Open

	G	M	S	avg.
<b>Wikilarge test set</b>				
Simple reference	4.70	4.56	-0.2	3.02
DISSIM	4.36	4.50	<b>1.30</b>	<b>3.39</b>
DSS	3.44	3.68	0.06	2.39
SENTS	3.48	2.70	-0.18	2.00
HYBRID	3.16	2.60	0.86	2.21
YATS	4.40	<b>4.60</b>	0.22	3.07
RegenT	<b>4.64</b>	4.56	0.28	3.16
<b>Newsela test set</b>				
Simple reference	4.92	2.94	0.46	2.77
DISSIM	4.44	4.60	<b>1.38</b>	<b>3.47</b>
HYBRID	2.97	2.35	0.93	2.08
YATS	4.26	4.42	0.32	3.00
RegenT	<b>4.54</b>	<b>4.70</b>	0.62	3.29
<b>WikiSplit test set</b>				
Simple reference	4.72	4.32	0.44	3.16
DISSIM	4.36	4.36	<b>1.66</b>	<b>3.46</b>
Copy512	<b>4.72</b>	<b>4.72</b>	0.92	3.45

Table 4: Human evaluation ratings on a random sample of 50 sentences from each dataset.

IE (Angeli et al., 2015), when directly operating on the raw input data with their performance when our DISSIM framework is applied as a preprocessing step. For this purpose, we made use of the Open IE benchmark framework proposed in Stanovsky and Dagan (2016).<sup>8</sup>

The results are displayed in Figure 4. The resulting improvements in overall precision, recall and area under the curve (AUC) are listed in Table 5. The numbers show that when using our DISSIM framework, all systems under consideration gain in AUC. The highest improvement in AUC was achieved by Stanford Open IE, yielding a 597% increase over the output produced when acting as a stand-alone system. AUC scores of REVERB and OLLIE improve by 57% and 20%. While REVERB primarily profits from a boost in recall (+40%), ClausIE, OLLIE and OpenIE-4 mainly improve in precision (+50%, +38% and +20%).

## 6 Comparative Analysis

In the following, we compare our TS framework with state-of-the-art rule-based syntactic TS approaches and discuss the strengths and weaknesses of each system.

**Sentence Splitting.** Table 6 compares the output generated by the TS systems RegenT and YATS

<sup>8</sup>In Cetto et al. (2018), we further present the performance of our system using the matching function that was originally described in Stanovsky and Dagan (2016), which uses a more fine-grained metric for the comparison of relational phrases and arguments.



on a sample sentence. As can be seen, RegenT and YATS break down the input into a sequence of sentences that present its message in a way that is easy to digest for human readers. However, the sentences are still rather long and present an irregular structure that mixes multiple semantically unrelated propositions, potentially causing problems for downstream tasks. On the contrary, our fairly aggressive simplification strategy that splits a source sentence into a large set of very short sentences<sup>9</sup> is rather inapt for a human audience and may in fact even hinder reading comprehension. Though, we were able to demonstrate that the transformation process we propose can improve the performance of downstream NLP applications.

SYSTEM	OUTPUT
Input	<i>The house was once part of a plantation and it was the home of Josiah Henson, a slave who escaped to Canada in 1830 and wrote the story of his life.</i>
RegenT	The house was once part of a plantation. And it was the home of Josiah Henson, a slave. This slave escaped to Canada in 1830 and wrote the story of his life.
YATS	The house was once part of a plantation. And it was the home of Josiah Henson. Josiah Henson was a slave who escaped to Canada in 1830 and wrote the story of his life.
DISSIM	#1 0 <b>The house was once part of a plantation.</b> L:LIST #2 #2 0 <b>It was the home of Josiah Henson.</b> L:ELABORATION #3 L:LIST #1 #3 1 <b>Josiah Henson was a slave.</b> L:ELABORATION #4 L:ELABORATION #6 #4 2 <b>This slave escaped to Canada.</b> L:TEMPORAL #5 L:LIST #6 #5 3 <b>This was in 1830.</b> #6 2 <b>This slave wrote the story of his life.</b> L:LIST #4

Table 6: Simplification example (from Newsela).

SYSTEM	OUTPUT
Input	<i>“The ambassador’s arrival has not been announced and he flew in complete secrecy,” the official said.</i>
LEXEV, EvLEX	He arrived in complete secrecy.
DISSIM	#1 0 <b>The ambassador’s arrival has not been announced.</b> L:LIST #2 L:ATTRIBUTION #3 #2 0 <b>He flew in complete secrecy.</b> L:LIST #1 L:ATTRIBUTION #3 #3 1 <b>This was what the official said.</b>

Table 7: Example (Štajner and Glavaš, 2017).

<sup>9</sup>In the output generated by DISSIM, contextual sentences are linked to their referring sentences and semantically classified by rhetorical relations. The number indicates the sentences’ context layer  $cl$ . Sentences with  $cl = 0$  carry the core information of the source, whereas sentences with a  $cl \geq 1$  provide contextual information about a sentence with a context layer of  $cl-1$ .

**Text Coherence.** The vast majority of syntactic simplification approaches do not take into account discourse-level aspects, producing a disconnected sequence of simplified sentences which results in a loss of cohesion that makes the text harder to interpret (Siddharthan, 2014). However, two notable exceptions have to be mentioned. Siddharthan (2006) was the first to use discourse-aware cues in one of RegenT’s predecessor systems, with the goal of generating a coherent output, e.g. by choosing appropriate determiners (“*This slave*” in Table 6). However, as opposed to our approach, where a semantic relationship is established for each output sentence, only a comparatively low number of sentences is linked by such cue words in Siddharthan (2006)’s framework (and its successors). EVLEX and LEXEV also operate on the discourse level. They are semantically motivated, eliminating irrelevant information from the input by maintaining only those parts of the input that belong to factual event mentions. Our approach, on the contrary, aims to preserve the full informational content of a source sentence, as illustrated in Table 7. By distinguishing core from contextual information, we are still able to extract only the key information given in the input.

## 7 Conclusion

We presented a recursive sentence splitting approach that transforms structurally complex sentences into a novel hierarchical representation in the form of core sentences and accompanying contexts that are semantically linked by rhetorical relations. In a comparative analysis, we demonstrated that our TS approach achieves the highest scores on all three simplification corpora with regard to SAMSA (0.67, 0.57, 0.54), and comes no later than a close second in terms of SAMSA<sub>abl</sub> (0.84, 0.84, 0.84), two recently proposed metrics targeted at automatically measuring the syntactic complexity of sentences. These findings are supported by the other scores of the automatic evaluation, as well as the manual analysis. In addition, the extrinsic evaluation that was carried out based on the task of Open IE verified that downstream semantic applications profit from making use of our proposed structural TS approach as a preprocessing step. In the future, we plan to investigate the constituency type classification and rhetorical relation identification steps and port this approach to languages other than English.

## References

- Roe Aharoni and Yoav Goldberg. 2018. [Split and rephrase: Better evaluation and stronger baselines](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724. 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.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- 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.
- Delphine Bernhard, Louis De Viron, Véronique Moriceau, and Xavier Tannier. 2012. Question generation for french: collating parsers and paraphrasing questions. *Dialogue & Discourse*, 3(2):43–74.
- Jan A. Botha, Manaal Faruqi, John Alex, Jason Baldridge, and Dipanjan Das. 2018. [Learning to split and rephrase from wikipedia edit history](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737. Association for Computational Linguistics.
- Nadjet Bouayad-Agha, Gerard Casamayor, Gabriela Ferraro, Simon Mille, Vanesa Vidal, and Leo Wanner. 2009. Improving the comprehension of legal documentation: the case of patent claims. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 78–87. ACM.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.
- Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018. [Graphene: Semantically-linked propositions in open information extraction](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2300–2311. Association for Computational Linguistics.
- R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. [Motivations and methods for text simplification](#). In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2, COLING '96*, pages 1041–1044, Stroudsburg, PA, USA. Association for Computational Linguistics.
- William Coster and David Kauchak. 2011. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, HLT '11*, pages 665–669, Stroudsburg, PA, 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*, pages 355–366, New York, NY, USA. ACM.
- 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.
- Daniel Ferrés, Montserrat Marimon, Horacio Saggion, and Ahmed AbuRa'ed. 2016. Yats: Yet another text simplifier. In *Natural Language Processing and Information Systems*, pages 335–342, Cham. Springer International Publishing.
- Goran Glavaš and Sanja Štajner. 2015. [Simplifying lexical simplification: Do we need simplified corpora?](#) In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68. Association for Computational Linguistics.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. [Incorporating copying mechanism in sequence-to-sequence learning](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640. Association for Computational Linguistics.
- Michael Heilman and Noah A Smith. 2010. Extracting simplified statements for factual question generation. In *Proceedings of QG2010: The Third Workshop on Question Generation*, volume 11.
- Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16, PARAPHRASE '03*, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

- Siddhartha Jonnalagadda, Luis Tari, Jörg Hakenberg, Chitta Baral, and Graciela Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 177–180. Association for Computational Linguistics.
- Alistair Knott and Robert Dale. 1994. Using linguistic phenomena to motivate a set of coherence relations. *Discourse processes*, 18(1):35–62.
- 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*, pages 2231–2234.
- 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.
- Mausam. 2016. [Open information extraction systems and downstream applications](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077.
- 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.
- Makoto Miwa, Rune Søtre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. [Entity-focused sentence simplification for relation extraction](#). In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796. Coling 2010 Organizing Committee.
- Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 435–445.
- Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference*, pages 111–120. Association for Computational Linguistics.
- Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616. Association for Computational Linguistics.
- Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 85–91.
- Gustavo H. Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16*, pages 3761–3767. AAAI Press.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.
- Luz Rello, Ricardo Baeza-Yates, and Horacio Saggion. 2013. The impact of lexical simplification by verbal paraphrases for people with and without dyslexia. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 501–512. Springer.
- Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. [Making it simplex: Implementation and evaluation of a text simplification system for spanish](#). *ACM Trans. Access. Comput.*, 6(4):14:1–14:36.
- Swarnadeep Saha and Mausam. 2018. [Open information extraction from conjunctive sentences](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299. Association for Computational Linguistics.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.
- Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71. IEEE.
- Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.
- Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.
- Advaith 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. Association for Computational Linguistics.

- Advait Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th international conference on Computational Linguistics*, page 896. Association for Computational Linguistics.
- Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing With Compositional Vector Grammars. In *ACL*.
- 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 Popovic. 2018. Improving machine translation of english relative clauses with automatic text simplification. In *Proceedings of the First Workshop on Automatic Text Adaptation (ATA)*.
- 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 (EMNLP)*, page (to appear), Austin, Texas. Association for Computational Linguistics.
- Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. [Bleu is not suitable for the evaluation of text simplification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744. Association for Computational Linguistics.
- Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. [Semantic structural evaluation for text simplification](#). 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 685–696. Association for Computational Linguistics.
- Elior Sulem, Omri Abend, and Ari Rappoport. 2018c. [Simple and effective text simplification using semantic and neural methods](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–173. 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.
- David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352. Association for Computational Linguistics.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.
- Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594. Association for Computational Linguistics.
- Zheming Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1353–1361. Association for Computational Linguistics.

## A Annotation Guidelines for the Manual Evaluation

Table 8 lists the questions for the human annotation. Since the focus of our work is on structural rather than lexical simplification, we follow the approach taken in [Sulem et al. \(2018c\)](#) in terms of SIMPLICITY and restrict our analysis to the syntactic complexity of the resulting sentences, which is measured on a scale that ranges from -2 to 2 in accordance with [Nisioi et al. \(2017\)](#), while neglecting the lexical simplicity of the output sentences. Regarding the GRAMMATICALITY and MEANING PRESERVATION dimensions, we adopted the guidelines from [Štajner and Glavaš \(2017\)](#), with some minor deviations to better reflect our goal of simplifying the *structure* of the input sentences, while *retaining their full informational content*.

PARAM.	QUESTION	SCALE
G	Is the output fluent and grammatical?	1 to 5
M	Does the output preserve the meaning of the input?	1 to 5
S	Is the output simpler than the input, ignoring the complexity of the words?	-2 to 2

Table 8: Questions for the human annotation.

## B Qualitative Analysis of the Transformation Patterns and Error Analysis

Tables 9 and 10 show the results of the recall-based qualitative analysis of the transformation patterns, together with the findings of the error analysis. These analyses were carried out on a dataset which we compiled.<sup>10</sup> It consists of 100 Wikipedia sentences per syntactic phenomenon tackled by our TS approach. In the construction of this corpus we ensured that the collected sentences exhibit a great syntactic variability to allow for a reliable predication about the coverage and accuracy of the specified simplification rules.

Note that we do not consider the rules for disembedding adjectival/adverbial phrases and lead NPs, since an examination of the frequency distribution of the syntactic constructs tackled by our approach over the Wikilarge, Newsela and Wiki-Split test sentences has shown that these types of constructs occur relatively rarely.

	freq.	%fired	%correct trans.
<b>Clausal disembedding</b>			
Coordinate clauses	113	93.8%	99.1%
Adverbial clauses	113	84.1%	96.8%
Relative clauses (non-def.)	108	88.9%	70.8%
Relative clauses (defining)	103	86.4%	75.3%
Reported speech	112	82.1%	75.0%
<b>Phrasal disembedding</b>			
Coordinate VPs	109	85.3%	89.2%
Coordinate NPs	115	48.7%	82.1%
Appositions (non-restrictive)	107	86.0%	83.7%
Appositions (restrictive)	122	87.7%	72.0%
PPs	163	68.1%	75.7%
Total	1165	81.1%	82.0%

Table 9: Recall-based qualitative analysis of the transformation rule patterns. This table presents the results of a manual analysis of the performance of the hand-crafted simplification patterns. The first column lists the syntactic phenomena under consideration, the second column indicates its frequency in the dataset, the third column displays the percentage of the grammar fired, and the fourth column reveals the percentage of sentences where the transformation operation results in a correct split.

<sup>10</sup>The dataset is available under [https://github.com/Lambda-3/DiscourseSimplification/tree/master/supplemental\\_material](https://github.com/Lambda-3/DiscourseSimplification/tree/master/supplemental_material).

	Err. 1	Err. 2	Err. 3	Err. 4	Err. 5	Err. 6
<b>Clausal disembedding</b>						
Coordinate clauses	1	0	0	0	0	0
Adverbial clauses	1	1	0	1	0	0
Relative clauses (non-def.)	5	8	0	0	14	1
Relative clauses (defining)	8	8	2	0	5	1
Reported speech	5	1	13	1	2	1
<b>Phrasal disembedding</b>						
Coordinate VPs	4	3	2	1	0	0
Coordinate NPs	3	3	0	3	1	0
Appositions (non-restrictive)	0	5	3	0	7	0
Appositions (restrictive)	1	21	3	0	0	0
PPs	3	11	4	6	4	0
Total	31 (19%)	61 (37%)	27 (16%)	12 (7%)	33 (20%)	3 (2%)

Table 10: Error analysis. This table shows the results of the error analysis conducted on the same dataset. Six types of errors were identified (Error 1: additional parts; Error 2: missing parts; Error 3: morphological errors; Error 4: wrong split point; Error 5: wrong referent; Error 6: wrong order of the syntactic elements).