Linguistic Corpus Annotation for Automatic Text Simplification Evaluation

Rémi Cardon, **Adrien Bibal**, **Rodrigo Wilkens**, **David Alfter**,
Magali Norré, **Adeline Müller**, **Patrick Watrin**, **Thomas François**
CENTAL, IL&C, University of Louvain, Belgium
{remi.cardon,adrien.bibal,rodrigo.wilkens,david.alfter
magali.norre,adeline.muller,patrick.watrin,thomas.francois}
@uclouvain.be

Abstract

Evaluating automatic text simplification (ATS) systems is a difficult task that is either performed by automatic metrics or user-based evaluations. However, from a linguistic point-of-view, it is not always clear on what bases these evaluations operate. In this paper, we propose annotations of the ASSET corpus that can be used to shed more light on ATS evaluation. In addition to contributing with this resource, we show how it can be used to analyze SARI's behavior and to re-evaluate existing ATS systems. We present our insights as a step to improve ATS evaluation protocols in the future.

1 Introduction

Automatic text simplification (ATS) is a task that consists in automatically adapting a text to make it more accessible for readers. It is the focus of more and more attention from researchers, with a growing number of publications every year and some surveys published recently (Saggion, 2017; Al-Thanyyan and Azmi, 2021; Štajner, 2021). As it is the case in many domains, ATS tasks are currently mostly explored using deep learning (Nisioi et al., 2017; Alva-Manchego et al., 2020b; Cooper and Shardlow, 2020), although rule-based systems are still being explored as well (Saggion, 2017; Evans and Orasan, 2019; Wilkens et al., 2020).

One serious hurdle that the field is currently facing is how to evaluate those systems (Grabar and Saggion, 2022). There are two common approaches: human judgment and automatic evaluation metrics. Human judgment is collected by asking people to rate the output of a system on a Likert scale using three criteria: grammaticality, meaning preservation and simplicity. This method is certainly encompassing, but requires lots of time and effort. In contrast, automatic evaluation metrics represent a reproducible and quick way to measure the performance of ATS systems. The most common metrics currently used are: BLEU (Papineni et al., 2002), a metric created to evaluate the performance of machine translation systems; SARI (Xu et al., 2016a), a metric specifically designed for ATS; and the Flesch-Kincaid readability formula (Kincaid et al., 1975). Although these metrics have serious shortcomings (Sulem et al., 2018a; Alva-Manchego et al., 2021; Tanprasert and Kauchak, 2021), their use is widespread in the field due to their ease of use. BLEU and SARI require reference data (human made simplifications) and are known to be more reliable when more references are available. In other words, their behavior is a function of the dataset used as the reference. Since different audiences have different needs when it comes to simplification (Rennes et al., 2022), making sure that a dataset will reflect those specific needs is important. Two other automatic metrics have been used in text simplification research but are not broadly used: BertScore (Zhang et al., 2020) – a metric designed for language generation evaluation – and SAMSA (Sulem et al., 2018b) – specifically designed for ATS but not easy to use as it requires semantic annotation.

In English, mostly three corpora are used for evaluation: TurkCorpus (Xu et al., 2016b), ASSET (Alva-Manchego et al., 2020a) and Newsela (Xu et al., 2015). The first two share the same source sentences, with different crowdsourced simplifications. For other languages, with less attention from the community, systems are evaluated against ad-hoc corpora (Kodaira et al., 2016; Cardon and Grabar, 2020; Anees and Abdul Rauf, 2021; Spring et al., 2021). Little is known about the way automatic metrics are related to the different types of simplification operations. Recent work (Vásquez-Rodríguez et al., 2021) investigated the relation of those metrics to basic computational operations such as add, delete or insert, but no work focuses on the linguistic simplifications listed in existing typologies (Brunato et al., 2022; Gala et al., 2020; Amancio and Specia, 2014).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1842 - 1866

In this paper, we aim to investigate the following hypothesis: evaluation metrics react differently depending on the type of linguistic operations applied to produce the reference(s) and the output of a system. To verify this hypothesis, we have annotated the ASSET corpus with a broad set of linguistic operations. The annotated resource is the first contribution of the paper and we hope it can spur research on new automated evaluation metrics. A second contribution is the annotation process itself (see Section 3), as we provide an annotation guide that has been validated with 9 annotators and could be reused to annotate other corpora in a similar way. In Section 4, we study SARI's behavior according to the operations we annotated and we calculate the SARI and BLEU scores of published systems on various subsets of ASSET (Section 4.2). This new angle of observations on automatic ATS evaluation, in relation with our hypothesis, is the third contribution of this paper.

2 Literature Review

Historically, before statistical and then neural methods became the main focus of ATS research, typologies were a requirement for building ATS systems, as they were the conceptual basis of rule-based methods. As documented by Siddharthan (2014), authors of ATS systems were mostly concerned with syntax (Chandrasekar et al., 1996; Dras, 1999; Brouwers et al., 2014) and produced typologies of the syntactic operations they aimed at performing on sentence structures. In more recent works, we could identify two main axes that were used to build typologies of simplification operations: one based on linguistic descriptions and the other one based on string edits.

2.1 Linguistically Based Operation Descriptions

The first type of typologies aims at identifying the linguistic operations at work during the simplification process. They have been studied for a variety of languages: Spanish (Bott and Saggion, 2014), Italian (Brunato et al., 2014, 2022), French (Koptient et al., 2019; Gala et al., 2020), Brazilian Portuguese (Caseli et al., 2009), Basque (Gonzalez-Dios et al., 2018) and English (Amancio and Specia, 2014). While they share some operations, for instance the transition from the verbal passive voice to the active voice, those typologies also have specific categories such as *proximization* (Bott and Saggion, 2014) – changing a sentence in order to address the reader directly – or *specification* (Koptient et al., 2019) – keeping a difficult term but adding an explanation next to it. Interestingly, these two examples depend on the genre of the text: it is unlikely to use proximization when simplifying encyclopedia articles for instance, and it is expected that specification would occur in text genres that involve technical jargon. Those typologies have been used to give more information about simplification corpora, and to observe what humans do when they simplify texts. It should be noted that none of the linguistically-based typologies have been integrated into an evaluation protocol for ATS systems, which is something that we hope to enable with this work.

2.2 String Edits Based Operation Description

This axis pertains to operations where sentences are seen as strings of tokens, that are edited during simplification. To the best of our knowledge, it has been explored almost exclusively for English (Coster and Kauchak, 2011; Alva-Manchego et al., 2017, 2020a; Vásquez-Rodríguez et al., 2021), with one recent exception for Italian (Brunato et al., 2022). As with linguistic typologies, the operations also differ from one work to the other, but the approach always consists in observing what happens to the tokens during simplification. There have been multiple uses for this king of typology. Like the linguistically-based one, it has been used to perform corpus analysis as a goal in itself (Alva-Manchego et al., 2020a). It has also been used to study the relation between string distances and the scores given by automatic evaluation metrics (Vásquez-Rodríguez et al., 2021). Some ATS systems incorporate such operations in their architecture (Alva-Manchego et al., 2017; Dong et al., 2019; Agrawal et al., 2021). The evaluation metric SARI integrates such operations as sub-components in its computation: *keep*, *add* and *delete* (See section 4.1 for more details about this).

3 Annotation

This section presents the dataset (Section 3.1) and the typology of the operations we use for annotation (Section 3.2). We then make a statement on what we chose not to annotate but could have been expected given the data and the current practices in ATS evaluation (Section 3.3). We describe the annotation process (Section 3.5) and we analyze

the annotated set that served for the inter-annotator agreement (Section 3.5), then we describe the resulting resource (Section 3.6. We finally compare our result with other similar works (Section 3.7.

3.1 Data

Two freely available corpora are currently widely used for evaluation in works on ATS for English: TurkCorpus (Xu et al., 2016b) and ASSET (Alva-Manchego et al., 2020a). We retained ASSET, which has been independently described as an improvement over TurkCorpus (Vásquez-Rodríguez et al., 2021). The following section introduces the operations with which we annotate the 3,590 references in ASSET's test set.

3.2 Typology

The works presented in Section 2.1 propose typologies built upon corpus observations. We build our own, relying on those typologies. In consequence, no new operations are introduced. The main goal is to have a typology that can be used as the core on which to build to analyze any corpus in any of the languages on which our source material is made. We did not take into account the genre-specific operations, such as the ones mentioned in Section 2.1 (*proximization* and *specification*). For convenience, we do not use a fine-grained distinction between the grammatical functions of the constituents that we annotate, thus we merged all the different part-of-speech modifications into a single operation.

Below we present the resulting set of operations. Each operation is shown along its short name that we use in the rest of the article. Some of them are self-explanatory and some are clarified with a short comment. For more details and examples, see Section 3 of the annotation guide in Appendix A.

We introduce the operations in two distinct sets: the first one contains operations that can be mapped to computational operations. The mapping of those operations to computational operations is the following: *insert* (also referred to as *add* in the literature), *delete* and *move* are already words that are used for computational operations. All the other categories are replacements/substitutions. It should be noted that the computational operation called *replace* is sometimes considered as a combination of *add* and *delete*, as it is the case in SARI for instance.

- **Move** (move)
- **Insert/Delete proposition** (inprop, delprop)
- **Insert/Delete modifier** (inmod, delmod). The definition of modifier we use is quite loose. This covers both word-level modifiers (e.g., a qualifying adjective modifying a noun) and sentence-level modifiers (e.g., adverbial phrases).
- **Insert/Delete for consistency** (incst, delcst). Any insertion or deletion required for the sentence to remain grammatical after another operation is performed.
- **Insert/Delete Other** (inoth, deloth). Any insertion or deletion that does not fit in the other insert categories.
- **Replace with synonym** (synonym)
- **Replace with hyperonym** (hyperonym)
- **Replace with hyponym** (hyponym)
- **Replace singular with plural** (s2p)
- **Replace plural with singular** (p2s)
- **Replace segment with a pronoun** (pron)
- **Replace pronoun with its antecedent** (fromPron)
- **Modify verbal features** (verbf). Any change of modality or tense on a verb.

The second set contains operations that can be performed with various computational operation combinations, or that are too complex to be consistently mapped to computational operations.

- **Active to passive** (a2p)
- **Passive to active** (p2a)
- **Part-of-speech change** (POSchange)
- **Split** (split)
- **Merge** (merge)
- **To impersonal form** (toImp)
- **To personal form** (fromImp)
- **Affirmation to negation** (a2n)
- **Negation to affirmation** (n2a)

We also added a label named **Erroneous simplification** (err). While we do not assess simplicity, this label lets the annotators signal manifest errors either in adequacy or grammaticality that make the simplification irrelevant as a reference in the evaluation of an ATS system.

3.3 Adequacy, Fluency and Simplicity

After describing the data we use and the typology that we propose, we would like to mention the aspects that we chose not to annotate. Usually, manual evaluation of simplifications is performed on three criteria: adequacy or meaning preservation, fluency or grammaticality, and simplicity. As the simplifications in the corpus were made by humans, we do not use the usual 5-point Likert scale and we simply chose to mark sentences that had obvious issues in one of those aspects as erroneous. We chose not to assess simplicity at all for several reasons.

First, in the literature the methods for doing so are not consistent from one work to the other; there is a need for standardization (Stodden, 2021). We consider that this is out of the scope of this paper. Second, we believe that a judgment on simplicity should be made by members of the target audience of the corpus or the system that is evaluated. ASSET was not made for an identified target audience, and, as an NLP research team, we do not represent a typical demographic target. Plus, Alva-Manchego et al. (2020a) collected human judgments of simplicity on 100 sentences from ASSET via crowdsourcing (with no specific demographic target), and obtained an inter-annotator agreement (Cohen, 1968) of 0.628. That observation corroborates the claim that assessing simplicity as a textual property, without a target audience in mind, is not optimal (Gooding, 2022).

3.4 Process

The annotation was performed using YAWAT (Germann, 2008), which has been used for the same purpose previously (Koptient et al., 2019). YAWAT lets the user create groups of tokens to annotate within sentence pairs, either belonging to both sentences or to one of the two (typically, insertions and deletions occur only on one side, while other operations such as synonymy involve tokens from both sides).[1] The creation of the typology preceded the start of the annotation process and was the basis for writing the annotation guide. Four persons (all NLP researchers: two PhD students and two postdocs) annotated the same 50 sentence pairs from ASSET. Three work directly in text simplification, one in adjacent domains such as lexical complexity or readability. That step was used (1) for assessing the clarity of the annotation guide, (2) to train the annotators on the annotation tool and (3) to discuss how to improve the annotation guide. It was the occasion to discuss difficult cases and how to address them[2] in order to reach consensus, and to tune priority rules. The typology itself was not modified as a result of those discussions.After this, we estimated that the annotation requires 10 hours per 100 sentence pairs as an upper bound. We reproduced that step twice, with 25 sentence pairs each time.

Once the final version of the guide was created, we hired five Master's students in linguistics (all enrolled in an NLP track) for the annotation[3]. They went through the same last two steps described above. The students were paid[4] to annotate. They could perform the task at their convenience.

The final step before the whole annotation began was to have everyone – the four researchers and the five students – annotate the same 50 new sentence pairs from ASSET. We calculated the inter-annotator agreement with the Davies & Fleiss agreement (Davies and Fleiss, 1982) using each token's label. The score was computed separately for the source side and the target side. This resulted in an agreement of 0.61 for the source side and 0.68 for the target side. We also calculated the agreement with all the different insertions merged into one category, and all the deletions merged into one category. This increased the agreement scores to 0.74 for the source side and 0.72 for the target side. This indicates that there is a tradeoff between the granularity of the labels and the agreement that can be expected. The first author manually reviewed the nine resulting sets of 50 sentence pairs in order to produce a single dataset. We did not proceed automatically in order to avoid any inconsistencies coming from disagreements. This aggregated dataset is referred to as the gold dataset in this paper, and the fully annotated dataset is referred to as ASSET_{ann}.

3.5 Analysis of the Annotation

In this section, we compare the annotations of each annotator versus the reference, on the gold dataset. First, the average κ score of each annotator with

[1] A more recent and more user-friendly tool for ATS corpora annotation exists, TS–ANNO (Stodden and Kallmeyer, 2022) but was not available yet during our annotation.

[2] Most of those cases can be found in the annotation guide (Appendix A) and serve as examples.

[3] While no native English speaker participated in the annotation, all participants have a proven strong B2 / C1 CEFR level of English.

[4] Hourly rate 25% above the national minimum wage.

the gold is 0.73 for the original sentences and 0.74 for the simplified sentences, which corresponds to a substantial agreement. Moreover, the standard deviation of the 9 annotators is rather low (σ = .04 for the source and σ = .05 for target), with κ values ranging from 0.65 to 0.79. This confirms that all annotators were able to annotate with a rather similar level of reliability.

In order to give a view on the reliability of the annotation per operation, we computed recall and precision by comparing the annotations of each annotator to the gold, per label. We averaged those values over the 9 annotators to get a global view of the categories that are difficult to annotate (see Table 8 in Appendix C for the complete results).

For most categories, the recall and precision values seem satisfactory, being superior to 0.6, which is in line with the global robust κ. On source sentences, annotators were prone to forget about the *fromPron* category, whereas they had trouble to correctly identify hyponyms and POS changes. A close investigation of the confusion matrices of each annotator reveals that most errors regarding hyponyms and hyperonyms are due to their annotation as synonyms. On simplifications, annotators tended to miss the two categories *fromPron* (confused with *incst* about 50% of the time) and *toImp*. In addition, they had trouble to correctly use the following categories: *inoth* (often confused with another insert operation such as *incst* and *inmod*) and *inprop* (which is also mixed with other insert operations).

3.6 Resource Description

This section describes the resource, called ASSET$_{ann}$, resulting from the annotation process. The ASSET test set contains 3,590 pairs of original and simplified sentences (359 sentences with 10 simplifications each). During the annotation process, we observed 19 pairs of identical sentences, and 227 erroneous simplifications across 157 different source sentences.[5] ASSET$_{ann}$ contains **3,323** annotated pairs of original and simplified sentences. In total, **12,827** operations were identified. *Synonym* is the most common operation observed (14% of the total number of operations). In fact, seven operations (*synonym*, *delcst*, *deloth*, *incst*, *delmod*, *move* and *delprop*) account for 70% of the operations in both the gold and ASSET$_{ann}$.

[5] 4 source sentences with 4 errors, 10 source sentences with 3 errors, 32 sentences with 2 errors and 105 with 1 error.

The number of operations in both datasets can be seen in Table 1. Notably, 17 operations were observed less than 10 times in the gold corpus.In the whole dataset, we identified that 31 annotations with delete labels (out of 5 073) were in the simplified side and 52 annotations with insert labels (out of 2 547) were in the original side. As those are errors, we automatically changed those labels into their adequate counterpart (e.g. changing all delcst labels found on the simplified side into incst operations). Table 1 and the results reported in Section 4 take this adjustment into account. We did not apply this to the data used for the analysis in Section 3.5.

Label	#anno gold	#anno
synonym	21	1793
delcst	29	1736
deloth	16	1549
incst	25	1391
delmod	22	1359
move	20	920
split	12	688
inoth	4	697
hyperonym	12	613
delprop	13	450
verbal features	6	428
POS change	1	278
inmod	4	243
inprop	3	195
hyponym	1	85
s2p	1	81
p2a	2	65
pron	3	64
fromPron	2	47
p2s	2	36
toImp	1	35
a2p	1	33
pos2neg	0	24
fromImp	0	23
neg2pos	0	8
merge	0	2

Table 1: Occurrence of the annotations

In order to see whether the number of operations in a sentence can be affected by the sentence length, we analyzed the relation between those two aspects (see in Appendix B for details). We found that while the longest sentences show the highest operation count, there is no significant correlation between the number of operations and the length

of a sentence.

Another important perspective is the number of operations per sentence. Given the nature of text simplification, different levels of linguistic operations are expected (e.g., lexical and syntactical operations). In this sense, we observed that 13% of the simplified sentences are the result of a single operation. Moreover, 50% of the sentences in the corpus result from up to 3 operations, and 88% of the sentences have up to 6 operations (see Table 7 in Appendix B).

Finally, we identified the number of tokens added and deleted by each operation (see Table 6). This enables to verify the mapping between linguistic and computational operations. As expected, all deletions tend to remove all the annotated words. However, in some cases, some words remain, or other words are inserted, in order to keep the sentence correct. Similarly, *fromImp* and *pron* operations tend to remove tokens in the sentence. In the opposite direction, content insertion is most remarkable in *a2p*, *fromPron*, *neg2pos*, and *pos2neg* operations, and the operations specific for insertions (*incst*, *inmod*, *inoth* and *inprop*). The other operations do not modify the number of tokens. Comparing the token count in both sides of the corpus in relation to each operation, we observed that all operations could be arranged as *add*, *del* and *replace* following the guide.[6] This analysis allows us to indirectly measure the quality of the annotation, and it also allows us to observe that 25% of the operations in ASSET are insertions, 40% are deletions and 35% are replacements.

3.7 Comparison

Some works proposing typologies mentioned in Section 2.1 report proportions for the operations they annotated in their respective corpora. In this section, we compare those observations. It is important to note that all the compared works vary regarding the corpora they use, namely along characteristics such as context of creation, language, size, text genre, domain and target audience. The typologies that were used are all different as well. As overcoming all those gaps for a comprehensive analysis is out of the scope of this paper, we propose a simple overview of the results.

[6]We compared the distribution using a T-test (alpha of 0.05). We could not observe a statistical different in the number of tokens after and before the *deloth* and *delmod* due to their high variance. However, when observing their usage, it is clear that they removed the tokens.

Table 2 shows the outcome of our comparison. We include information about the context of the corpus creation in the *Method* column. Indeed, some source corpora were created as part of a work on ATS (S), and others were not (A). This might have an influence on the simplification process. For example, ASSET's crowdworkers were given examples of original sentences and simplified versions before performing the task. As one of the goals of the creators of ASSET was to produce a parallel corpus with "multiple rewriting operations" – in opposition to TurkCorpus which contained lexical operations – the approach may explain why the proportion of lexical operations we found is the lowest. The highest proportion of lexical operations was found in the CLEAR corpus. As CLEAR was built by aligning sentences from comparable corpora in the medical domain, this may be explained by the amount of specialized terminology in the complex side. The CBST corpus displays two methods of corpus creation, called "Intuitive" and "Stuctural". In both cases, the approach was to simplify scientific texts so that they could be understood by children. The first one consisted in asking a teacher to rewrite texts following only their knowledge of the target audience, the other one consisted in asking a court translator to simplify using easy-to-read guidelines. While the proportion of syntactic operations is similar in both cases, the other types vary.

Those surface observations indicate that there might not exist a universal way of simplifying a text, even within a given language. The factors that we could identify are numerous: language, target audience, domain, genre, profile of the person(s) performing the simplification, context of the simplification task (for human readers or for research), and type of instructions given. We believe that those criteria should be taken into account when working on ATS systems during the design of the system, the choices made in the selection and preparation of the data, the evaluation protocol and the comparison with other works.

4 Experiments

We propose two experiments that show how ASSET$_{ann}$ can help studying evaluation protocols. First, we analyze the behavior of SARI, with respect to ASSET$_{ann}$. Second, we re-evaluate existing ATS systems from the literature, to have a view on how they handle linguistic operations.

Corpus	Language	Size	Method	Source or Domain	Lexical	Syntactic	Split	Merge
ASSET$_{ann}$ (this work)	en	3 223 sp	S	Wikipedia	19.41%	37.22%	5.36%	0.01%
CBST Intuitive (Gonzalez-Dios et al., 2018)	eu	454 sp	S	Science	24.92%	39.54%	23.55%	0.40%
CBST Structural (Gonzalez-Dios et al., 2018)	eu	454 sp	S	Science	33.62%	39.39%	12.30%	0.22%
CLEAR (Koptient et al., 2019)	fr	663 sp	A	Medical	69%	23%	-	-
Parallel143 (Amancio and Specia, 2014)	en	143 sp	A	Wikipedia	39.8%	39.47%	7.02%	-
PorSimples (Caseli et al., 2009)	pt	104 dp	S	News	46.31%	15.6%	34.17%	0.24%
Simplext (Bott and Saggion, 2014)	es	145 sp	A	News	39.02%	37.4%	12.20%	0.81%
Terence (Brunato et al., 2014)	it	1 036 sp	A	Children's stories	40.01%	41,26%	1.75%	0.57%

Table 2: Comparison of reported simplification corpora contents. In the *Method* column, A = aligned from a pre-existing corpus (either comparable or manually simplified), S = manually simplified within a work related to ATS. In the *Size* column, sp = sentence pairs, dp = document pairs. The reported size corresponds to the sample used for analysis, which may differ from the actual size of the corpus with the same name. A cell containing "-" means either that the value is 0 or that the information was not reported for this operation type.

4.1 Analysis of SARI's Behavior

We use ASSET$_{ann}$ to analyze SARI's and its sub-components' behavior in relation to simplification operations. SARI is composed of three sub-components that are averaged to obtain SARI's final score. These sub-components are *keep*, *add* and *del*, respectively taking into account *n*-grams that are kept, added or deleted from the original sentence to the simplified one, taking reference(s) into account. More precisely, for each sub-component (*sc*) *keep*, *add* and *del*, the F1-score is computed for each *n*-gram size *n*[7]:

$$F1(n, sc) = \frac{2 * prec_{sc}(n) * recall_{sc}(n)}{prec_{sc}(n) + recall_{sc}(n)}$$

$$\text{SARI} = \frac{1}{3} \sum_{sc \in \{keep, add, del\}} \frac{1}{k} \sum_{n=1}^{k} F1(n, sc).$$

We created a tabular dataset in which the presence of operations is represented by the number of occurrences in the sentence pair annotation. For instance, "move = 2" and "synonym = 0" means that two *move* operations were applied to the sentence to obtain the simplification, while no *synonym* operation was found. By observing the correlation between the presence of specific operations and the global SARI scores, we find little correlation between the two. This means that no specific operation seems to correspond to the global SARI score. What is surprising is that operations mapped to SARI's sub-components (insertions and deletions) are not correlated with SARI scores (see Table 9 in Appendix D), despite being correlated with SARI's sub-components (see Table 10 in Appendix D).

In order to go further, we analyze how well combining operations can predict SARI's score. To evaluate our models, we use an average R^2 using a 10-fold cross-validation, with 9 folds for training and 1 fold for testing. The 157 source sentences for which we found at least one erroneous simplification were removed from the experiment. This is to make sure that all original sentences contain 10 non-erroneous simplifications for the 10-fold cross validation procedure. A Lasso regression model (Tibshirani, 1996) with optimized hyper-parameters barely obtains a R^2 of 0 when predicting the final SARI score. This indicates that the model cannot predict better than by simply using the mean. We found the same result with regression trees (Breiman et al., 1984), random forests (Breiman, 2001) and multi-layer perceptrons (Hinton, 1989)[8]. However, predicting SARI's sub-components is at least possible using Lasso with an average R^2 of 0.24, 0.03 and 0.23 for *keep*, *add* and *del* respectively. Table 3 presents the coefficients of a Lasso model trained on the whole corpus to predict SARI's sub-components. These coefficients were obtained with an R^2 of 0.25, 0.05 and 0.24 for *keep*, *add* and *del* respectively. The main finding at this level is that a large amount of operations have 0 as coefficient, meaning those have no effect on SARI's sub-components.

We can state that while SARI has a degree of relation to linguistic operations, averaging the score of all its sub-components hides this piece of information. This highlights two issues with SARI. First, SARI has a very low variance and is not very sensitive to the differences between system outputs. This makes predictions using only SARI's mean really efficient, as averaging three very different sub-components always leads to roughly the same SARI score. Second, as SARI relies on references

[7] This description of SARI is the one implemented in EASSE (Alva-Manchego et al., 2019), which is the implementation we used throughout this paper.

[8] All experiments of this section have been performed using Scikit-learn (Pedregosa et al., 2011)

	keep	*add*	*del*
inoth	0	0.1	0
split	0	0.73	0
deloth	-3.91	-0.19	3.05
delcst	-2.44	0	1.52
move	-1.55	0.14	1.5
delprop	-3.97	-0.74	3.66
incst	0	0.95	0
hyperonym	0	0.14	1.58
synonym	-1.68	0.61	3.63
delmod	-3.39	0	3.02

Table 3: Coefficients of Lasso regression models predicting SARI's sub-components. Operations with non-zeros coefficients for *add* and *del* indeed involve adding and deleting tokens. The negative coefficients for predicting *keep* indicates that the operation reduces the score of the *keep* sub-component of SARI. Absent operations have all coefficient values at 0.

to compute its score, the average on several references (in our case 9 references) reinforces the low variance of SARI. This last issue is also present for the three sub-components of SARI, which explain the somewhat low R^2 scores. Indeed, in all cases, predicting with the mean already makes it possible to obtain good results. It is therefore difficult to find the particular operations that explain more than what the mean already explains. This is particularly true for the *add* sub-component. This interpretation also accounts for the reason why the Lasso coefficients in Table 3 make sense, despite their associated low R^2: it is very difficult to explain what the mean does not already explain.

Using ASSET_{ann}, we highlighted two elements of SARI. First, we presented in Table 3 the operations that best match the sub-components of SARI. This favors the use of SARI's sub-components for evaluation, instead of the global score. Second, the difficulty to make better predictions than the mean highlights the issue of SARI's low variance.

4.2 Re-evaluation of ATS Systems

This section shows how the annotation we propose can be useful for the ATS research community. First, as we identified 2 % of erroneous simplifications – which might affect the current perception of existing systems –, we rescored the systems proposed by Zhang and Lapata (2017); Dong et al. (2019); Martin et al. (2020); Nisioi et al. (2017); Wubben et al. (2012) (see the *orig* and *new* columns of Table 4[9]). Overall, BLEU scores decrease by 1.4 on average. seq2seq_{wiki} obtains the best BLEU score (from 76.3 to 85.9), outperforming Dress-Ls, which decreased by 4 points (from 86.4 to 82.4). For SARI, removing errors decreased all scores. In particular, MUSS systems have the strongest decrease (average of 1.3 points).

We also analyzed the effect of linguistically motivated groups of operations. For that, we created a subset of ASSET by keeping only references that were annotated with at least one lexical operation (i.e., *synonym*, *hyponym*, and *hyperonym* – 2,138 references) and another subset with references having at least one syntactic operation (i.e., *inprop*, *inmod*, *inoth*, *delprop*, *delmod*, *deloth*, *a2p*, *p2a*, *split*, *toImp*, *fromImp*, *a2n*, *n2a*, – 2,799 references). These results are presented in columns synt and lex of Table 4. In general, SARI scores decreased for all models. MUSS_{wiki}♠ (Martin et al., 2020) remains the highest scoring system for lexical and syntactic simplification. In general, we observed that the decrease of SARI is on average 1.3 with the syntactic subset, and an impressive decrease average of 3.5 points with the lexical subset.

Finally, we explore the behavior of computational operations in relation to our mapping (see Section 3.2) – also observed in the number of added, deleted, and replaced tokens (see Section 3.6) – in order to observe the systems' performance. We create 3 subsets: one with references containing at least one *add* operation (2,040 references), another one with *del* operations (2,866 references) and one with *replace* operations (2,719 references). Comparing the SARI scores, MUSS systems tend to present similar scores for the three subsets, but the difference between the scores in the entire dataset (column ORIG) and each subset is remarkably different: $\text{MUSS}_{newsela}$♠ is the most stable system with a decrease of 2.7, 0.5 and 1.4 respectively for *add*, *del* and *replace* subsets. On the other hand, MUSS_{wiki} is the system with the greatest loss of performance 4.3, 1.5 and 2.0 respectively for *add*, *del* and *replace* subsets. In general, the systems kept a similar SARI score with *del* (average decrease of 1.3) and *replace* (average decrease of 1.8) subsets. Using the *add* subset, SARI decreased on average by 4 points on all systems. It is noticeable that systems with better SARI scores have a large proportion of *add*.

[9]The *orig* and *new* columns indicate, respectively, the values for the original ASSET and the values after removing the

	SARI							BLEU		Proportion	
	orig	new	lex	synt	mapping			orig	new	add	del
					add	del	repl				
Dress (Zhang and Lapata, 2017)	37.1	36.3	33.4	34.6	32.0	34.6	34.7	84.2	80.0	0.04	0.31
Dress-Ls (Zhang and Lapata, 2017)	36.6	35.8	32.7	34.0	31.3	34.1	34.1	**86.4**	82.2	0.04	0.30
EditNTS (Dong et al., 2019)	34.9	34.0	30.3	32.3	29.4	32.3	31.9	86.2	83.3	0.04	0.23
NTS-SARI (Nisioi et al., 2017)	34.0	33.4	29.5	31.7	28.7	31.6	31.1	84.2	82.0	0.06	0.21
PBMT-R (Wubben et al., 2012)	34.6	33.7	29.2	32.1	29.2	32.2	31.3	79.4	77.6	0.09	0.15
MUSS♠ (Martin et al., 2020)	42.7	41.4	38.3	40.3	37.9	40.3	39.9	66.2	64.6	0.16	0.28
MUSS$_{newsela}$ (Martin et al., 2020)	42.9	41.6	37.7	40.4	38.0	40.5	39.6	71.4	70.1	0.20	0.30
MUSS$_{newsela}$♠ (Martin et al., 2020)	41.4	40.3	37.5	39.8	37.6	39.8	38.9	78.4	75.7	0.11	0.37
MUSS$_{wiki}$ (Martin et al., 2020)	43.6	42.3	38.5	40.8	38.1	40.9	40.3	76.3	74.8	0.19	0.26
MUSS$_{wiki}$♠ (Martin et al., 2020)	**44.2**	**42.8**	**39.6**	**41.6**	**39.2**	**41.6**	**41.3**	72.9	71.5	0.19	0.29
MUSS♣♠ (Martin et al., 2020)	41.1	40.0	37.2	39.5	37.1	39.5	38.7	77.2	74.0	0.06	0.35
seq2seq♠ (Martin et al., 2020)	38.0	37.0	33.5	36.1	33.7	36.2	35.2	61.8	60.4	0.17	0.23
seq2seq$_{wiki}$ (Martin et al., 2020)	32.7	32.0	27.7	30.1	27.4	30.2	29.9	76.3	**85.9**	0.03	0.18

Table 4: SARI and BLEU scores of systems considering the original and annotated ASSET and grouping the sentences regarding the operation type. Proportion is the proportion of *add* and *del* operations as calculated by EASSE. Unsupervised MUSS are indicated with ♠ and MUSS with mBart is indicated with ♣.

5 Discussion

We performed the annotation of ASSET with the goal of shedding light on automatic ATS evaluation using linguistic information. This enabled us to make several insightful contributions.

Regarding the analysis of current automatic evaluation, we have shown that SARI's sub-components can inform on the linguistic operations present in references that appear in a system output, whereas this is lost when averaging the three sub-components into one single score. This insight is an argument in favor of reporting the sub-components' scores while performing evaluation, as some works started to do (Zhao et al., 2020; Tanprasert and Kauchak, 2021). The analyses we performed also make the case for further exploration in bridging linguistic operations and computational operations. We believe that ASSET$_{ann}$ and our experiments can serve to pave the way for future ATS evaluation practices that would go further in that direction.

We selected subsets of references to use for the evaluation according to predetermined sets of operations. Creating such subsets allows to focus on relevant operations for a given target audience's specific needs in the evaluation framework. ASSET$_{ann}$ is a first step in that direction.

As discussed in Section 3.3, we did not evaluate simplicity. Though it can be time consuming, we recommend to assess simplicity based on the precise definition of the task. During the resource creation, annotators told us they thought that some simplifications were, in fact, obviously not simpler. Working with members of the target audience of a given task in order to identify references that should not be taken into account would allow for a more accurate evaluation protocol.

Finally, as a future work we would like to leverage the annotated dataset to automatically reproduce the annotation.

6 Conclusion

In this work, we produced an annotated version of the ASSET test set. The resource was annotated by 9 annotators, following a typology that we proposed based on existing similar typologies. Based on the annotation, we could clean the test set by removing 227 sentence pairs with manifest errors and produce a curated version of ASSET. We re-evaluated systems for which the outputs were available and provided the community with updated performance results using SARI and BLEU. The different resources described in the paper and the code used for our analyses are available online.[10]

In addition, we performed an extensive analysis of SARI's sub-components and found links between those and linguistic operations. We see these results as a promising direction to improve automatic ATS evaluation by exploring the relationship between computational and linguistic operations.

2% of the sentences identified as erroneous simplification.

[10] https://github.com/remicardon/assetann

7 Acknowledgements

We would like to express our gratitude to Nils Bouckaert, Elena Cao, Angela Kasparian, Melanie Johanns and Luca Matarelli, who helped us annotate the dataset. We also thank Damien de Meyere and Hubert Naets for their support with YAWAT.

We also thank the anonymous reviewers for their suggestions and comments that helped improve the quality of the paper.

Rémi Cardon is supported by the FSR Incoming Postdoc Fellowship program of the FSR - Université Catholique de Louvain. Adrien Bibal is supported by the Walloon region with a Win2Wal funding. Rodrigo Wilkens is supported by a research convention with France Education International (FEI). David Alfter is supported by the Fonds de la Recherche Scientifique de Belgique (F.R.S-FNRS) under the grant MIS/PGY F.4518.21.

8 Limitations

The limitations of our work are the following:

- We do not comment upon simplicity in the corpus. The reason for this is explained in Section 3.3;
- We should mention that the typology proposed, as it aims to be as generic as possible, covers a large amount of simplification transformations, but might need some adaptations or additions to be applied to specific contexts or languages;
- We identified that some operations were less consensual to annotate than others and that granularity makes the task more difficult (Section 3.5). That said, this does not invalidate our analyses and we intend our typology to serve as a basis for producing task-specific sets of operations, based on the translation of users' needs into concrete simplification operations;
- The annotation was performed only on one evaluation dataset. We discuss this in Sections 1 and 3.1. We found linguistic operations that do not influence the score of SARI's sub-components (Section 4.1), but as we only annotated ASSET's test set, more observations need to be performed to confirm or refute this conclusion.

References

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769, Online. Association for Computational Linguistics.

Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020a. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020b. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un) suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Marcelo Amancio and Lucia Specia. 2014. An analysis of crowdsourced text simplifications. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 123–130, Gothenburg, Sweden. Association for Computational Linguistics.

Yusra Anees and Sadaf Abdul Rauf. 2021. Automatic sentence simplification in low resource settings for Urdu. In *Proceedings of the 1st Workshop on NLP for Positive Impact*, pages 60–70, Online. Association for Computational Linguistics.

Stefan Bott and Horacio Saggion. 2014. Text simplification resources for Spanish. *Language Resources and Evaluation*, 48:93–120.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Laetitia Brouwers, Delphine Bernhard, Anne-Laure Ligozat, and Thomas François. 2014. Syntactic sentence simplification for french. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)@ EACL 2014*, pages 47–56.

Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2022. Linguistically-based comparison of different approaches to building corpora for text simplification: A case study on italian. *Frontiers in Psychology*, 13.

Dominique Brunato, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2014. Defining an annotation scheme with a view to automatic text simplification. In *Proceedings of the Italian Conference on Computational Linguistics and of the International Workshop EVALITA*, pages 87–92.

Rémi Cardon and Natalia Grabar. 2020. French biomedical text simplification: When small and precise helps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 710–716, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Helena M Caseli, Tiago F Pereira, Lucia Specia, Thiago AS Pardo, Caroline Gasperin, and Sandra Maria Aluísio. 2009. Building a brazilian portuguese parallel corpus of original and simplified texts. *Advances in Computational Linguistics, Research in Computer Science*, 41:59–70.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *The 16th International Conference on Computational Linguistics*.

J. Cohen. 1968. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–20.

Michael Cooper and Matthew Shardlow. 2020. CombiNMT: An exploration into neural text simplification models. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5588–5594, Marseille, France. European Language Resources Association.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.

M. Davies and Joseph L. Fleiss. 1982. Measuring agreement for multinomial data. *Biometrics*, 38:1047.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Mark Dras. 1999. *Tree adjoining grammar and the reluctant paraphrasing of text*. Macquarie University Sydney.

Richard Evans and Constantin Orasan. 2019. Sentence simplification for semantic role labelling and information extraction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

Núria Gala, Amalia Todirascu, Delphine Bernhard, Rodrigo Wilkens, and Jean-Paul Meyer. 2020. Transformations syntaxiques pour une aide à l'apprentissage de la lecture : typologie, adéquation et corpus adaptés. *SHS Web of Conferences*, 78:14006.

Ulrich Germann. 2008. Yawat: Yet Another Word Alignment Tool. In *Proceedings of the ACL: HLT Demo Session*, pages 20–23, Columbus, Ohio. Association for Computational Linguistics.

Itziar Gonzalez-Dios, María Jesús Aranzabe, and Arantza Díaz de Ilarraza. 2018. The corpus of Basque simplified texts (CBST). *Language Resources and Evaluation*, 52(1):217–247.

Sian Gooding. 2022. On the ethical considerations of text simplification. In *9th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT-2022)*, pages 50–57, Dublin, Ireland. Association for Computational Linguistics.

Natalia Grabar and Horacio Saggion. 2022. Evaluation of automatic text simplification: Where are we now, where should we go from here. In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 453–463, Avignon, France. ATALA.

Geoffrey E Hinton. 1989. Connectionist learning procedures. *Artificial Intelligence*, 40(1):185–234.

J. Kincaid, R.P. Fishburne, R. Rodgers, and B. Chissom. 1975. *Derivation of new readability formulas for navy enlisted personnel*. Technical report, n°8-75, Research Branch Report.

Tomonori Kodaira, Tomoyuki Kajiwara, and Mamoru Komachi. 2016. Controlled and balanced dataset for Japanese lexical simplification. In *Proceedings of the ACL Student Research Workshop*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Anaïs Koptient, Rémi Cardon, and Natalia Grabar. 2019. Simplification-induced transformations: typology and some characteristics. In *Proceedings of the BioNLP Workshop and Shared Task*, pages 309–318.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020. MUSS: multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*.

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)*, pages 85–91.

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 of the Association for Computational Linguistics*, pages 311–318.

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

Evelina Rennes, Marina Santini, and Arne Jonsson. 2022. The swedish simplification toolkit: – designed with target audiences in mind. In *Proceedings of The 2nd Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) within the 13th Language Resources and Evaluation Conference*, pages 31–38, Marseille, France. European Language Resources Association.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

Nicolas Spring, Annette Rios, and Sarah Ebling. 2021. Exploring German multi-level text simplification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 1339–1349, Held Online. INCOMA Ltd.

Sanja Štajner. 2021. Automatic text simplification for social good: Progress and challenges. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 2637–2652.

Regina Stodden. 2021. When the scale is unclear–analysis of the interpretation of rating scales in human evaluation of text simplification. *Proceedings of the 1st Workshop on Current Trends in Text Simplification (CTTS 2021, co-located with SEPLN 2021)*.

Regina Stodden and Laura Kallmeyer. 2022. TS-ANNO: An annotation tool to build, annotate and evaluate text simplification corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 145–155, Dublin, Ireland. 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 Conference on Empirical Methods in Natural Language Processing*, pages 738–744.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Teerapaun Tanprasert and David Kauchak. 2021. Flesch-kincaid is not a text simplification evaluation metric. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14, Online. Association for Computational Linguistics.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Laura Vásquez-Rodríguez, Matthew Shardlow, Piotr Przybyła, and Sophia Ananiadou. 2021. Investigating text simplification evaluation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 876–882, Online. Association for Computational Linguistics.

Laura Vásquez-Rodríguez, Matthew Shardlow, Piotr Przybyla, and Sophia Ananiadou. 2021. The role of text simplification operations in evaluation. In *Proceedings of the SEPLN Workshop on Current Trends in Text Simplification*, pages 57–69.

Rodrigo Wilkens, Bruno Oberle, and Amalia Todirascu. 2020. Coreference-based text simplification. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 93–100.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024.

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. 2016a. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016b. Optimizing Statistical Machine Translation for Text Simplification.

Transactions of the Association for Computational Linguistics, 4:401–415.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 584–594.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9668–9675.

A Annotation Guide

1 Connection to Yawat

Yawat is the annotation tool used. It is a web interface. As Yawat presents security risks, access is done in two steps.

First, navigate to REDACTED URL.

You will be asked for a login and a password:

- login: REDACTED
- password: REDACTED

Normally, you can save this information in the browser so that you don't have to copy the password each time.

You then reach the Yawat connection interface where you will use the login and password communicated to you.

2 Using Yawat

Once connected, you will see a list of files to annotate. Click on a file name to go to the annotation interface proper.

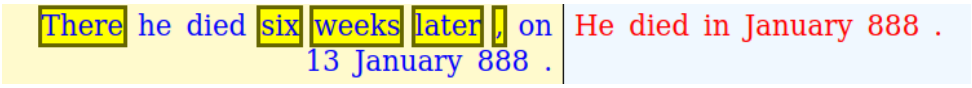
In the annotation interface, sentences are grouped: the original sentence on the left and the simplified sentence on the right.

It it possible to switch to a different view where the sentences are shown one above the other by clicking on the logo with the two rectangles in the top right corner of the page.

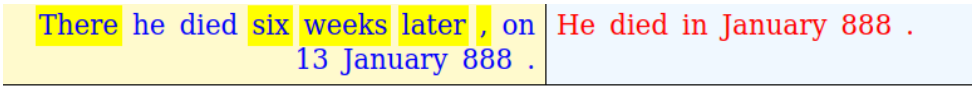
On first opening of an annotation file, check a "done" case, then uncheck it, then reload the page. Without this step, the interface will not be responsive.

Annotating a segment of text, either on one side only (e.g., for insertions or deletions), or by aligning segments of the two sentences (e.g., for lexical substitutions) is done in three steps:

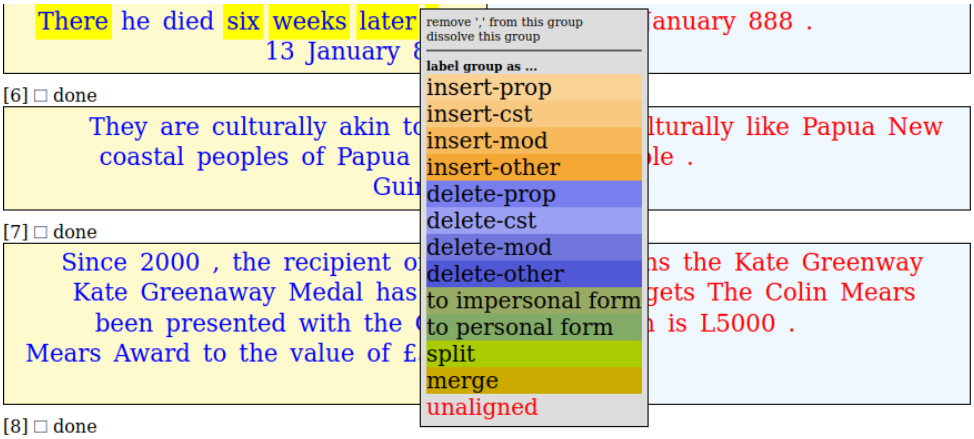
- Click on each token of the group to annotate



- Hold down Shift and click on one of the highlighted words, then release Shift. This step validates the group.



- Hold down Shift and click on one of the words in the group and the annotation menu will appear. The list changes based on whether the words of the groups appear in the two sentences or just in one of the two sentences.



After annotating a sentence pair, you can check the "done" case. This updates your progress on the welcome page, saves the annotation of the sentence and advances the view so that the next unchecked (i.e., sentence to annotate) appears next on screen.

To make sure that your progress is saved, click on the "save" button in the top right corner of the page.

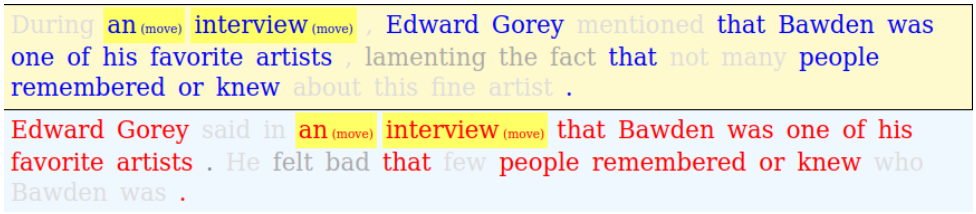
3 Annotation

This section describes each element of the typology.

3.1 Move

This label is used to indicate words that change positions but are not otherwise modified. This label is to be annotated on the level of the constituent that is moved. **Transformation operations take precedence over this label.**

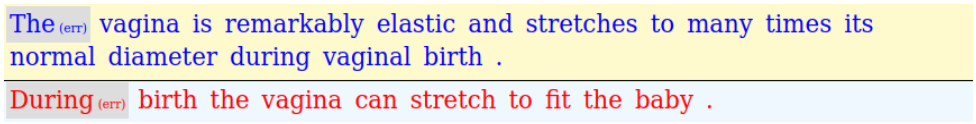
Example:



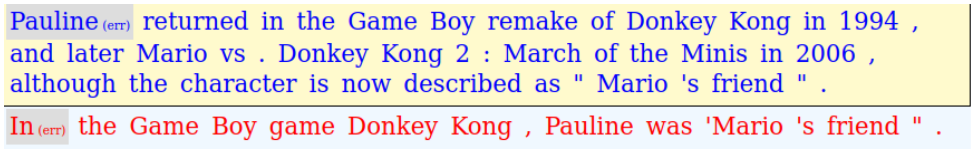
3.2 Erroneous simplification

This label is to be used with parsimony: the label is used to indicate cases where the simplification is contradicting the original sentence or where the simplification is clearly unrelated to the original sentence. In this case, no other label than this one is to be used; all other transformations are ignored. The label is applied to the first word of each sentence.

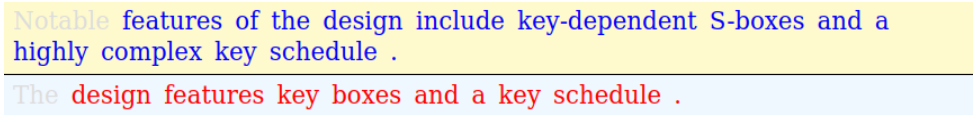
Examples:



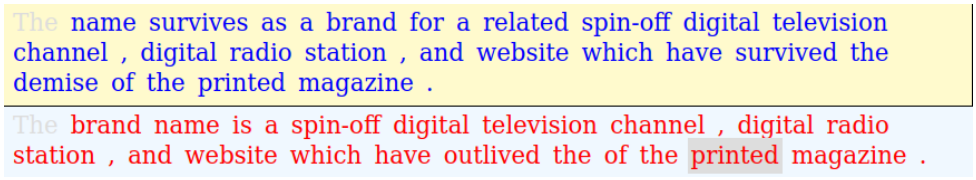
The segment *to fit the baby* doesn't correspond to any information in the original sentence.



The two sentences are contradictory.



Deleting the affixes *-dependent* and *S-* distorts the meaning of the original sentence.



While it is possible to reconstruct the meaning of the sentence, there's a missing word between *the* and *of* in the simplified sentence.

3.3 Insert/Delete

3.3.1 -cst

cst stands for *consistency*. This label is used for transformations **imposed by the context**, i.e. transformations that become necessary for the sentence to remain grammatical due to other transformations. For example, if *eats* becomes *eaten by* in a pair of sentences, *by* is annotated as *insert-cst*. Example (insertion):

In the example, the annotation indicates that *She* and *was* were added due to another transformation (the sentence splitting).

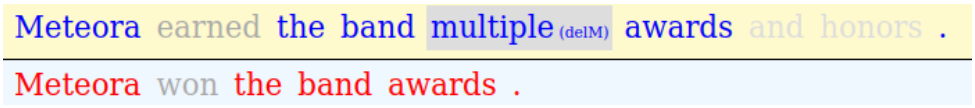
These labels will be used abundantly, including:

- All insertions and deletions of puncutation (except sentence splitting, see section 3.5.5)
- All insertions and deletions of possessive markers (*'s*)
- In cases where a preposition or determiner could be annotated with the synonymy label (*on March 9, 2000* → *in 2000*; *a* → *the*)

3.3.2 -mod

mod stands for *modifier*. All added or deleted modifiers are to be annotated with this label. This covers both word-level modifiers (for example a qualifying adjective modifying a noun) and sentence-level modifiers (for example adverbial phrases).

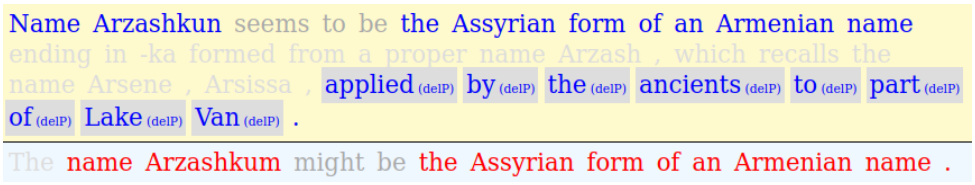
Example (deletion):



3.3.3 -prop

prop stands for *proposition*. All added or deleted propositions are to be annotated with this label.

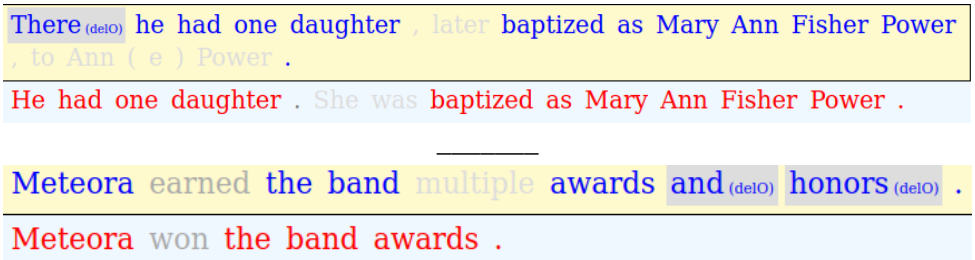
Example (deletion):



3.3.4 -other

This label is used for all insertions and deletions not covered by the aforementioned cases.

Examples (deletion):

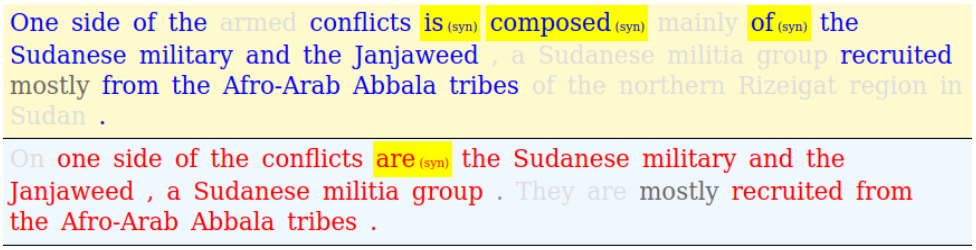


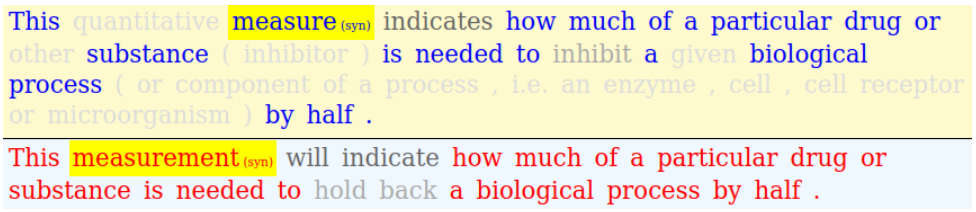
3.4 Lexical transformations

3.4.1 Without part-of-speech change

- The label *synonym* is used in the broadest sense; verbal paraphrases are also covered by this definition, as shown in the first example.

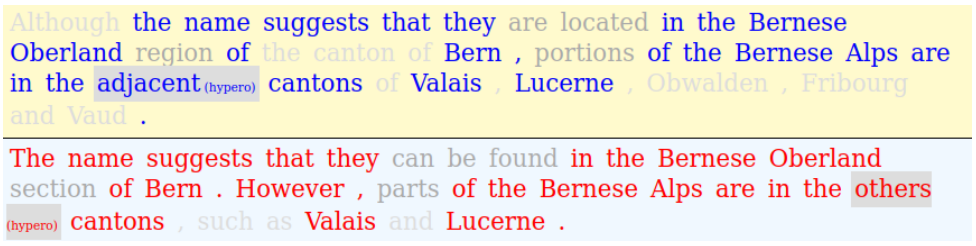
  Examples :





  The labels *hyperonym* and *hyponym* are rather self-explanatory. The simplified term is characterized with regards to the original term. For example, if the original sentence contains *cat* and the simplified sentence *animal*, the label will be *hyperonym*. These labels do not only cover what could be found in a lexical network but have to be evaluated in the context of the sentence, as shown in the following example.

  Example (hyperonymy):



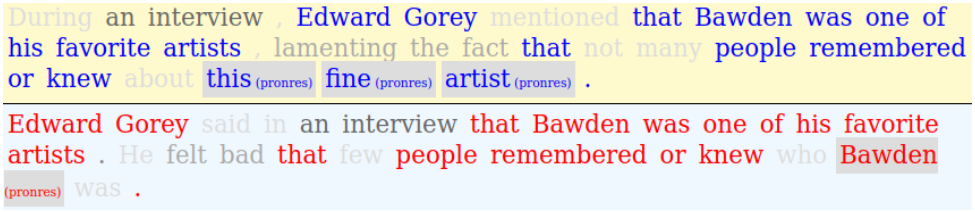
- *singular to plural* and *plural to singular* are rather self-explanatory. The simplified term is characterized with regards to the original term. If, for example, the original sentence contains *cats* and the simplified sentence *cat*, the label will be *plural to singular*. This label is to be used solely if the transformation is a change in number. **Lexical substitution labels take precedence over this label.**

  No examples encountered during the testing phase.

3.4.2 Part-of-speech changes

There are two types of part-of-speech changes:

- Transformations related to coreference: *pronominalization* and *pronoun resolution*. The simplified term is characterized with regards to the original term. The notion of *pronoun* is used in the sense of *anaphoric expression*, as illustrated in the following example:



- Part-of-speech changes: *POS change*. This label concerns words that are in a relationship of derivation. For example, this label is used for transformations such as *advertisement* → *advertise*.

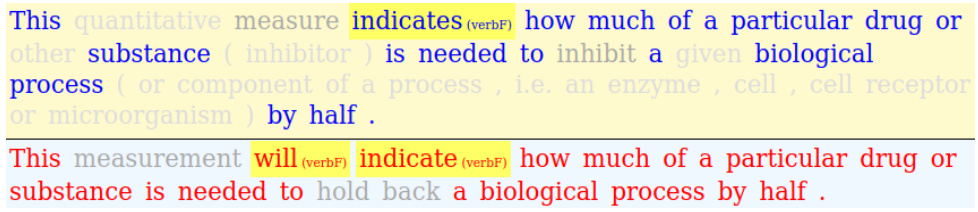
  No examples encountered during the testing phase.

3.5 Syntactic transformations

3.5.1 Verbal features

This label indicates changes in tense or modality of the verb. For example, if the original sentence contains *does* and the simplified sentence *will do*, this change will be annotated as *verbal features*. **Lexical substitution and grammatical voice take precedence over this label.**

Example:

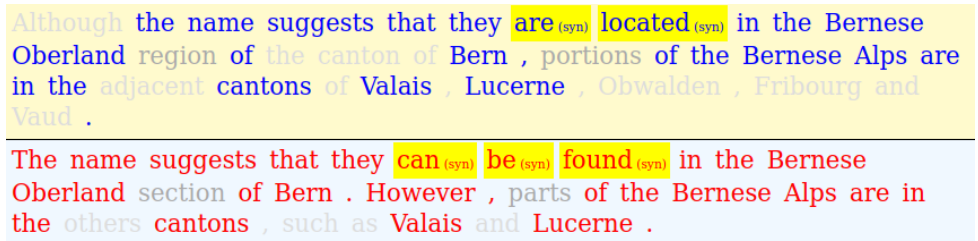


3.5.2 Active to passive / Passive to active

These labels indicate a change in voice. Only the verb is annotated with this label; changes in syntactic function of the agent or patient are not annotated. **If the subject of the verb changes, this label takes priority. If the subject of the verb does not change, lexical transformations take precedence over this label.**

Examples:

The complement "the town charter" becomes the subject in the simplification, thus this label is applied.



Since the subject is the same in both sentences, the label *synonym* is used instead.

3.5.3 To impersonal form / To personal form

This label indicates transformations such as *It is our house that the cat is in* → *The cat is in our house*. In this example, *It, is, that* in the original sentence are annotated with the label *To personal form*. In the opposite case, the same words (in the simplified sentence) would be annotated with *To impersonal form*.

Example (to impersonal form):

3.5.4 Affirmation to negation / Negation to affirmation

These labels indicate the change from an affirmative sentence to a negated sentence and vice versa. Only negation markers are annotated with this label. Negation markers appearing in the simplification are annotated with *Affirmation to negation*, while negation markers disappearing from the original sentence are annotated with *Negation to affirmation*.

Example :

During an interview , Edward Gorey mentioned that Bawden was one of his favorite artists , lamenting the fact that not (n2a) many (n2a) people remembered or knew about this fine artist .

Edward Gorey said in an interview that Bawden was one of his favorite artists . He felt bad that few (n2a) people remembered or knew who Bawden was .

3.5.5 Merge / Split

Split indicates that the original sentence is split into multiple sentences in the simplified version. For example, if the original sentence is *The cat is tall and also blue* and the simplified sentence *The cat is tall. It is also blue.*, the full stop between the two simplified sentences is annotated with *Split*. The segment *It is* in *It is also blue.* is annotated as *insert-cst* (see examples). The label *merge* – two sentences or more grouped into one sentence – is present for reasons of coherence. However, there normally is no case where this label is applicable, as sentences are simplified one by one in ASSET.

Examples:

During an interview , Edward Gorey mentioned that Bawden was one of his favorite artists , lamenting the fact that not many people remembered or knew about this fine artist .

Edward Gorey said in an interview that Bawden was one of his favorite artists . (split) He felt bad that few people remembered or knew who Bawden was .

During an interview , Edward Gorey mentioned that Bawden was one of his favorite artists , lamenting the fact that not many people remembered or knew about this fine artist .

Edward Gorey said in an interview that Bawden was one of his favorite artists . He (insertC) felt bad that few people remembered or knew who Bawden was .

3.6 Note

It can happen that certain transformations can be annotated in more than one way, with no rules of priority taking absolute precedence. As it is impossible to cover all possibilities, an arbitrary rule is to **annotate as close to the token-level as possible**.

For example, the the following sentence:

Schuschnigg immediately responded publicly that reports of riots were false .

Schuschnigg immediately informed the public of the false riot reports .

The transformation *reports of riots* → *riot reports* can be annotated in two ways:

- Annotate the two groups with the label *synonym*
- Annotate (i) *riots* and *riot* as *plural to singular*, (ii) *reports* and *reports* as *move*, and (iii) *of* in the original sentence as *delete-cst*.

Both approaches are valid, but in order to minimize disagreement, we apply the second choice in this case.

4 Table of abbreviations

The following table summarizes the abbreviations used in the annotation interface and their corresponding labels as described in this document. The table is organized alphabetically by the labels shown in the interface, except for mirrored operations.

Abbreviation	Operation
a2n	affirmation to negation
n2a	negation to affirmation
a2p	active to passive
p2a	passive to active
delC	delete-cst
delM	delete-mod
delO	delete-other
delP	delete-prop
err	erroneous simplification
hypero	hyperonym
hypo	hyponym
insertC	insert-cst
insertM	insert-mod
insertO	insert-other
insertP	insert-prop
merge	merge
move	move
POSC	POS change
pronom	pronominalization
pronres	pronoun resolution
p2s	plural to singular
s2p	singular to plural
split	split
syn	synonym
toImp	to impersonal form
toPers	to personal form
verbF	verbal features

Table 5: Glossary of abbreviations used in the screenshots throughout the annotation guide.

B Measurements of the annotated resource

	Whole dataset				Gold			
	#tks (sent. orig)		#tks (sent. simpl)		#tks (sent. orig)		#tks (sent. simpl)	
Tag	Avg	Std	Avg	Std	Avg	Std	Avg	Std
a2p	1,182	0,625	2,000	0,492	2,000	NA	2,000	NA
delcst	1,204	0,523	NA	NA	1,263	0,440	NA	NA
delmod	1,956	1,955	0,004	0,091	1,759	1,164	NA	NA
deloth	2,749	2,713	NA	NA	2,955	1,846	NA	NA
delprop	5,912	5,339	NA	NA	6,188	3,486	NA	NA
fromImp	2,000	0,978	0,130	0,337	NA	NA	NA	NA
fromPron	1,163	0,421	2,020	0,958	1,000	NA	1,500	0,500
hyperonym	1,625	2,328	1,165	0,571	1,000	NA	1,000	NA
hyponym	1,078	0,307	1,333	0,667	1,500	1,225	1,000	NA
incst	NA	NA	1,359	0,788	NA	NA	1,340	0,723
inmod	NA	NA	1,389	0,83	NA	NA	1,500	0,866
inoth	NA	NA	1,723	1,32	NA	NA	1,400	0,490
inprop	NA	NA	2,873	1,862	NA	NA	3,000	1,414
merge	1,000	NA	NA	NA	NA	NA	NA	NA
move	2,294	1,922	2,295	1,926	2,917	3,128	2,938	3,165
neg2pos	2,750	1,714	1,250	0,433	NA	NA	NA	NA
p2a	1,970	0,758	1,152	0,435	1,500	0,500	1,000	NA
p2s	1,026	0,160	1,000	NA	1,000	NA	1,000	NA
pos2neg	1,600	0,849	2,440	1,061	NA	NA	NA	NA
POSchange	1,009	0,125	1,035	0,215	1,000	NA	1,000	NA
pron	3,313	4,183	1,060	0,237	1,333	0,471	1,000	NA
s2p	1,085	0,453	1,000	NA	1,000	NA	1,000	NA
split	NA	NA	1,000	NA	NA	NA	1,000	NA
synonym	1,320	0,726	1,371	0,784	1,361	0,713	1,528	0,897
toImp	NA	NA	2,143	0,723	NA	NA	2,000	NA
verbf	1,233	0,523	1,107	0,335	1,000	NA	1,000	NA

Table 6: Average (and standard deviation) number of tokens in the original and simplified sentences per tag, in the whole dataset and in the gold.

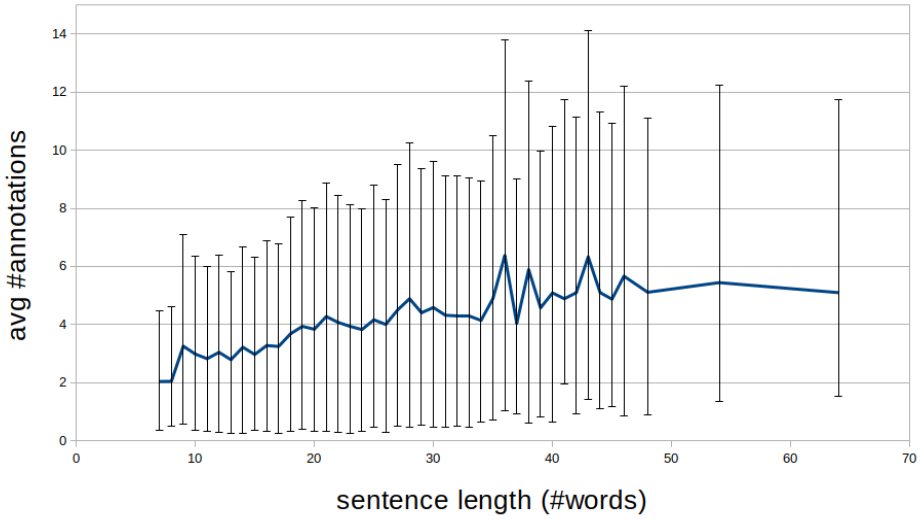


Figure 1: Number of annotations w.r.t. the size of the sentence (error bar indicates a confidence interval of 0.05).

#anno	#pairs of sent
1	444
2	533
3	594
4	548
5	464
6	334
7	191
8	106
9	57
10	24
11	7
12	2
14	1

Table 7: Number of annotations per sentence in ASSET$_{ann}$.

C Average recall and precision values per type of simplification operations over the 9 annotators

		Simplified		Original	
Operation	**Total Count**	**Recall**	**Precision**	**Recall**	**Precision**
a2p	1	0.89	0.79	0.83	0.79
delcst	30	NA	NA	0.69	0.71
delmod	22	NA	NA	0.68	0.63
deloth	16	NA	NA	0.61	0.52
delprop	13	NA	NA	0.57	0.77
err	5	0.58	0.75	0.58	0.75
fromPron	2	0.33	0.92	0.39	0.92
hyperonym	12	0.63	0.65	0.6	0.69
hyponym	1	0.56	0.32	0.56	0.38
incst	24	0.75	0.75	NA	NA
inmod	4	0.59	0.52	NA	NA
inoth	4	0.51	0.2	NA	NA
inprop	3	0.53	0.34	NA	NA
move	20	0.81	0.89	0.81	0.89
none	NA	**0.94**	**0.96**	**0.93**	**0.93**
p2a	2	0.83	0.88	0.85	0.76
p2s	2	0.78	0.92	0.78	0.92
POSchange	1	1	0.49	1	0.48
pron	3	0.7	0.77	0.69	0.8
s2p	1	0.78	0.65	0.78	0.65
split	12	0.93	0.93	NA	NA
synonym	12	0.66	0.69	0.66	0.66
toImp	1	0.44	0.67	NA	NA
verbf	6	0.81	0.79	0.81	0.82

Table 8: Number of annotations per operation found in the gold corpus, and average recall and precision at the token level for the 9 annotators, per operation (simplified sentences and original sentences). NA indicates operations not seen in the respective data. *None* has a count of NA because it is only applied when an annotator does not label a token.

D Correlations Linguistic Operations – SARI and Linguistic Operations – SARI's sub-components

	Spearman correlation	p-value
inoth	0.0161	0.3349
split	0.0915	0
deloth	-0.0137	0.4119
p2s	-0.017	0.3084
delcst	0.0413	0.0134
verbf	0.0519	0.0018
pron	-0.035	0.036
move	0.012	0.4731
inprop	0.0295	0.0775
delprop	-0.0381	0.0223
incst	0.0718	0
s2p	-0.0338	0.0428
fromPron	-0.0038	0.8194
merge	-0.0033	0.8421
p2a	0.0119	0.4752
pos2neg	0.0028	0.8648
neg2pos	-0.018	0.2806
hyponym	-0.0191	0.253
toImp	-0.0283	0.09
fromImp	-0.0349	0.0367
POSchange	0.0177	0.2895
hyperonym	0.0374	0.025
a2p	-0.018	0.2814
synonym	0.163	0
delmod	-0.0025	0.8792
inmod	0.0194	0.2451

Table 9: Spearman correlation between the occurrence of operations in sentences and their SARI score. p-values of 0 mean that the values are lower than 0.0001. Large p-values come from the fact that some operations do not sufficiently occur in the corpus.

	keep		add		del	
Transformation	Spearman	p-value	Spearman	p-value	Spearman	p-value
inoth	-0.0752	0	0.0658	0.0001	0.073	0
split	0.0251	0.1328	0.1925	0	-0.0063	0.7052
deloth	-0.2214	0	0.0008	0.9614	0.2015	0
p2s	-0.0478	0.0042	0.0144	0.3899	0.0378	0.0235
delcst	-0.2267	0	0.1482	0	0.2279	0
verbf	-0.0827	0	0.089	0	0.1441	0
pron	-0.0827	0	0.011	0.5084	0.0321	0.0547
move	-0.1764	0	0.0828	0	0.1486	0
inprop	-0.0699	0	0.069	0	0.0947	0
delprop	-0.1694	0	-0.0636	0.0001	0.1809	0
incst	-0.1269	0	0.2198	0	0.1362	0
s2p	-0.108	0	0.0258	0.1217	0.073	0
fromPron	-0.0458	0.006	0.0114	0.4931	0.04	0.0165
merge	-0.0257	0.123	-0.0056	0.736	0.0269	0.107
p2a	-0.0304	0.0684	0.0123	0.4616	0.0421	0.0116
pos2neg	-0.038	0.0229	0.0186	0.2663	0.0346	0.038
neg2pos	-0.038	0.0226	-0.0147	0.3777	0.0378	0.0235
hyponym	-0.0373	0.0256	0.0263	0.1145	0.0068	0.6847
toImp	-0.0826	0	0.0148	0.3767	0.049	0.0033
fromImp	-0.0355	0.0333	-0.0278	0.0957	-0.0055	0.7408
POSchange	-0.1317	0	0.0655	0.0001	0.1538	0
hyperonym	-0.0303	0.0699	0.0836	0	0.0649	0.0001
a2p	-0.0307	0.0658	-0.0281	0.0926	0.0155	0.353
synonym	-0.0607	0.0003	0.2135	0	0.2116	0
delmod	-0.1857	0	0.0195	0.2416	0.2071	0
inmod	-0.0472	0.0047	0.0332	0.0466	0.0725	0

Table 10: Spearman correlations (and their respective p-value) between each annotated transformation and the sub-components of SARI (keep, add, del). p-values of 0 means that the value is lower than 0.0001.