Simple and Effective Text Simplification Using Semantic and Neural Methods

Elior Sulem, Omri Abend, Ari Rappoport
Department of Computer Science, The Hebrew University of Jerusalem
{eliors|oabend|arir}@cs.huji.ac.il

Abstract

Sentence splitting is a major simplification operator. Here we present a simple and efficient splitting algorithm based on an automatic semantic parser. After splitting, the text is amenable for further fine-tuned simplification operations. In particular, we show that neural Machine Translation can be effectively used in this situation. Previous application of Machine Translation for simplification suffers from a considerable disadvantage in that they are over-conservative, often failing to modify the source in any way. Splitting based on semantic parsing, as proposed here, alleviates this issue. Extensive automatic and human evaluation shows that the proposed method compares favorably to the state-of-the-art in combined lexical and structural simplification.

1 Introduction

Text Simplification (TS) is generally defined as the conversion of a sentence into one or more simpler sentences. It has been shown useful both as a preprocessing step for tasks such as Machine Translation (MT; Mishra et al., 2014; Štajner and Popović, 2016) and relation extraction (Niklaus et al., 2016), as well as for developing reading aids, e.g. for people with dyslexia (Rello et al., 2013) or non-native speakers (Siddharthan, 2002).

TS includes both structural and lexical operations. The main structural simplification operation is sentence splitting, namely rewriting a single sentence into multiple sentences while preserving its meaning. While recent improvement in TS has been achieved by the use of neural MT (NMT) approaches (Nisioi et al., 2017; Zhang et al., 2017; Zhang and Lapata, 2017), where TS is considered a case of monolingual translation, the sentence splitting operation has not been addressed by these systems, potentially due to the rareness of this operation in the training corpora (Narayan and Gardent, 2014; Xu et al., 2015).

We show that the explicit integration of sentence splitting in the simplification system could also reduce conservatism, which is a grave limitation of NMT-based TS systems (Alva-Manchego et al., 2017). Indeed, experimenting with a state-of-the-art neural system (Nisioi et al., 2017), we find that 66% of the input sentences remain unchanged, while none of the corresponding references is identical to the source. Human and automatic evaluation of the references (against other references), confirm that the references are indeed simpler than the source, indicating that the observed conservatism is excessive. Our methods for performing sentence splitting as pre-processing allows the TS system to perform other structural (e.g. deletions) and lexical (e.g. word substitutions) operations, thus increasing both structural and lexical simplicity.

For combining linguistically informed sentence splitting with data-driven TS, two main methods have been proposed. The first involves handcrafted syntactic rules, whose compilation and validation are laborious (Shardlow, 2014). For example, Siddharthan and Angrosh (2014) used 111 rules for relative clauses, appositions, subordination and coordination. Moreover, syntactic splitting rules, which form a substantial part of the rules, are usually language specific, requiring the development of new rules when ported to other languages (Aluísio and Gasperin, 2010; Seretan, 2012; Hung et al., 2012; Barlacchi and Tonelli, 2013, for Portuguese, French, Vietnamese, and Italian respectively). The second method uses linguistic information for detecting potential splitting points, while splitting probabilities are learned us-

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 162–173
Melbourne, Australia, July 15 - 20, 2018.

ing a parallel corpus. For example, in the system of Narayan and Gardent (2014) (henceforth, HYBRID), the state-of-the-art for joint structural and lexical TS, potential splitting points are determined by event boundaries.

In this work, which is the first to combine structural semantics and neural methods for TS, we propose an intermediate way for performing sentence splitting, presenting Direct Semantic Splitting (DSS), a simple and efficient algorithm based on a semantic parser which supports the direct decomposition of the sentence into its main semantic constituents. After splitting, NMT-based simplification is performed, using the NTS system. We show that the resulting system outperforms HYBRID in both automatic and human evaluation.

We use the UCCA scheme for semantic representation (Abend and Rappoport, 2013), where the semantic units are anchored in the text, which simplifies the splitting operation. We further leverage the explicit distinction in UCCA between types of Scenes (events), applying a specific rule for each of the cases. Nevertheless, the DSS approach can be adapted to other semantic schemes, like AMR (Banarescu et al., 2013).

We collect human judgments for multiple variants of our system, its sub-components, HYBRID and similar systems that use phrase-based MT. This results in a sizable human evaluation benchmark, which includes 28 systems, totaling at 1960 complex-simple sentence pairs, each annotated by three annotators using four criteria.[1] This benchmark will support the future analysis of TS systems, and evaluation practices.

Previous work is discussed in §2, the semantic and NMT components we use in §3 and §4 respectively. The experimental setup is detailed in §5. Our main results are presented in §6, while §7 presents a more detailed analysis of the system's sub-components and related settings.

2 Related Work

MT-based sentence simplification. Phrase-based Machine Translation (PBMT; Koehn et al., 2003) was first used for TS by Specia (2010), who showed good performance on lexical simplification and simple rewriting, but under-prediction of other operations. Štajner et al. (2015) took a similar approach, finding that it is beneficial to use training data where the source side is highly similar to the target. Other PBMT for TS systems include the work of Coster and Kauchak (2011b), which uses Moses (Koehn et al., 2007), the work of Coster and Kauchak (2011a), where the model is extended to include deletion, and PBMT-R (Wubben et al., 2012), where Levenshtein distance to the source is used for re-ranking to overcome conservatism.

The NTS NMT-based system (Nisioi et al., 2017) (henceforth, N17) reported superior performance over PBMT in terms of BLEU and human evaluation scores, and serves as a component in our system (see Section 4). Zhang et al. (2017) took a similar approach, adding lexical constraints to an NMT model. Zhang and Lapata (2017) combined NMT with reinforcement learning, using SARI (Xu et al., 2016), BLEU, and cosine similarity to the source as the reward. None of these models explicitly addresses sentence splitting.

Alva-Manchego et al. (2017) proposed to reduce conservatism, observed in PBMT and NMT systems, by first identifying simplification operations in a parallel corpus and then using sequence-labeling to perform the simplification. However, they did not address common structural operations, such as sentence splitting, and claimed that their method is not applicable to them.

Xu et al. (2016) used Syntax-based Machine Translation (SBMT) for sentence simplification, using a large scale paraphrase dataset (Ganitketitch et al., 2013) for training. While it does not target structural simplification, we include it in our evaluation for completeness.

Structural sentence simplification. Syntactic hand-crafted sentence splitting rules were proposed by Chandrasekar et al. (1996), Siddharthan (2002), Siddhathan (2011) in the context of rule-based TS. The rules separate relative clauses and coordinated clauses and un-embed appositives. In our method, the use of semantic distinctions instead of syntactic ones reduces the number of rules. For example, relative clauses and appositives can correspond to the same semantic category. In syntax-based splitting, a generation module is sometimes added after the split (Siddharthan, 2004), addressing issues such as reordering and determiner selection. In our model, no explicit regeneration is applied to the split sentences, which are fed directly to an NMT system.

Glavaš and Štajner (2013) used a rule-based system conditioned on event extraction and syntax

[1] The benchmark can be found in https://github.com/eliorsulem/simplification-acl2018.

for defining two simplification models. The event-wise simplification one, which separates events to separate output sentences, is similar to our semantic component. Differences are in that we use a single semantic representation for defining the rules (rather than a combination of semantic and syntactic criteria), and avoid the need for complex rules for retaining grammaticality by using a subsequent neural component.

Combined structural and lexical TS. Earlier TS models used syntactic information for splitting. Zhu et al. (2010) used syntactic information on the source side, based on the SBMT model of Yamada and Knight (2001). Syntactic structures were used on both sides in the model of Woodsend and Lapata (2011), based on a quasi-synchronous grammar (Smith and Eisner, 2006), which resulted in 438 learned splitting rules.

The model of Siddharthan and Angrosh (2014) is similar to ours in that it combines linguistic rules for structural simplification and statistical methods for lexical simplification. However, we use 2 semantic splitting rules instead of their 26 syntactic rules for relative clauses and appositions, and 85 syntactic rules for subordination and coordination.

Narayan and Gardent (2014) argued that syntactic structures do not always capture the semantic arguments of a frame, which may result in wrong splitting boundaries. Consequently, they proposed a supervised system (HYBRID) that uses semantic structures (Discourse Semantic Representations, (Kamp, 1981)) for sentence splitting and deletion. Splitting candidates are pairs of event variables associated with at least one core thematic role (e.g., agent or patient). Semantic annotation is used on the source side in both training and test. Lexical simplification is performed using the Moses system. HYBRID is the most similar system to ours architecturally, in that it uses a combination of a semantic structural component and an MT component. Narayan and Gardent (2016) proposed instead an unsupervised pipeline, where sentences are split based on a probabilistic model trained on the semantic structures of Simple Wikipedia as well as a language model trained on the same corpus. Lexical simplification is there performed using the unsupervised model of Biran et al. (2011). As their BLEU and adequacy scores are lower than HYBRID's, we use the latter for comparison.

Štajner and Glavaš (2017) combined rule-based simplification conditioned on event extraction, together with an unsupervised lexical simplifier. They tackle a different setting, and aim to simplify texts (rather than sentences), by allowing the deletion of entire input sentences.

Split and Rephrase. Narayan et al. (2017) recently proposed the Split and Rephrase task, focusing on sentence splitting. For this purpose they presented a specialized parallel corpus, derived from the WebNLG dataset (Gardent et al., 2017). The latter is obtained from the DBPedia knowledge base (Mendes et al., 2012) using content selection and crowdsourcing, and is annotated with semantic triplets of subject-relation-object, obtained semi-automatically. They experimented with five systems, including one similar to HYBRID, as well as sequence-to-sequence methods for generating sentences from the source text and its semantic forms.

The present paper tackles both structural and lexical simplification, and examines the effect of sentence splitting on the subsequent application of a neural system, in terms of its tendency to perform other simplification operations. For this purpose, we adopt a semantic corpus-independent approach for sentence splitting that can be easily integrated in any simplification system. Another difference is that the semantic forms in Split and Rephrase are derived semi-automatically (during corpus compilation), while we automatically extract the semantic form, using a UCCA parser.

3 Direct Semantic Splitting

3.1 Semantic Representation

UCCA (Universal Cognitive Conceptual Annotation; Abend and Rappoport, 2013) is a semantic annotation scheme rooted in typological and cognitive linguistic theory (Dixon, 2010b,a, 2012; Langacker, 2008). It aims to represent the main semantic phenomena in the text, abstracting away from syntactic forms. UCCA has been shown to be preserved remarkably well across translations (Sulem et al., 2015) and has also been successfully used for the evaluation of machine translation (Birch et al., 2016) and, recently, for the evaluation of TS (Sulem et al., 2018) and grammatical error correction (Choshen and Abend, 2018).

Formally, UCCA structures are directed acyclic graphs whose nodes (or *units*) correspond either to the leaves of the graph or to several elements viewed as a single entity according to some semantic or cognitive consideration.

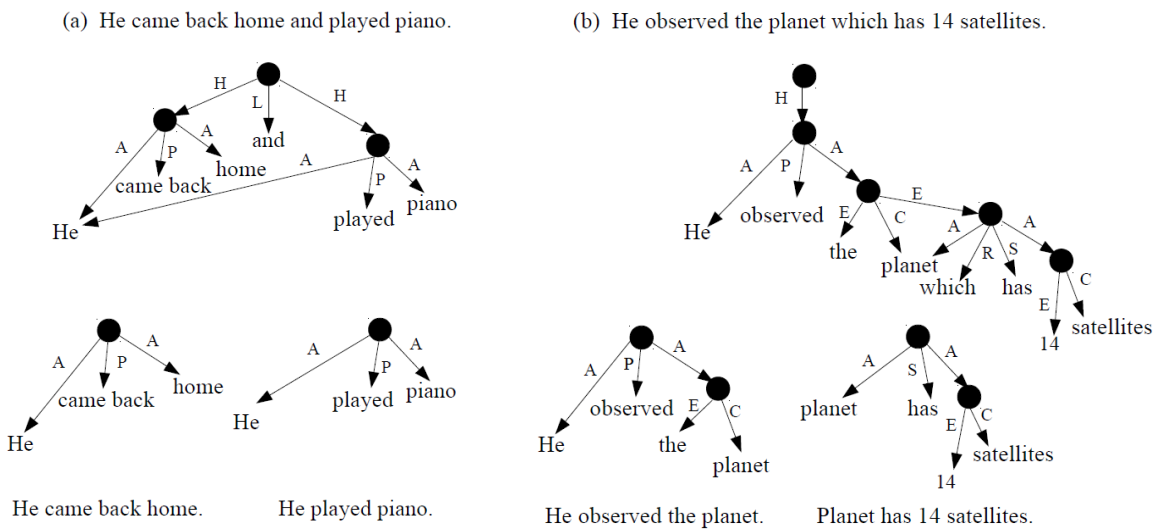

Figure 1: Example applications of rules 1 (Figure 1a) and 2 (Figure 1b). In both cases, the original sentence, the semantic parse, the extracted Scenes with the required modifications, and the output of the rules are presented top to bottom. The UCCA categories used are: Parallel Scene (H), Linker (L), Participant (A), Process/State (P/S), Center (C), Elaborator (E), Relator (R).

A *Scene* is UCCA's notion of an event or a frame, and is a unit that corresponds to a movement, an action or a state which persists in time. Every Scene contains one main relation, which can be either a Process or a State. Scenes contain one or more Participants, interpreted in a broad sense to include locations and destinations. For example, the sentence "He went to school" has a single Scene whose Process is "went". The two Participants are "He" and "to school".

Scenes can have several roles in the text. First, they can provide additional information about an established entity (Elaborator Scenes), commonly participles or relative clauses. For example, "(child) who went to school" is an Elaborator Scene in "The child who went to school is John" ("child" serves both as an argument in the Elaborator Scene and as the Center). A Scene may also be a Participant in another Scene. For example, "John went to school" in the sentence: "He said John went to school". In other cases, Scenes are annotated as Parallel Scenes (H), which are flat structures and may include a Linker (L), as in: "When$_L$ [he arrives]$_H$, [he will call them]$_H$".

With respect to units which are not Scenes, the category Center denotes the semantic head. For example, "dogs" is the Center of the expression "big brown dogs", and "box" is the center of "in the box". There could be more than one Center in a unit, for example in the case of coordination, where all conjuncts are Centers. We define the minimal center of a UCCA unit u to be the UCCA graph's leaf reached by starting from u and iteratively selecting the child tagged as Center.

For generating UCCA's structures we use TUPA, a transition-based parser (Hershcovich et al., 2017) (specifically, the TUPA$_{BiLSTM}$ model). TUPA uses an expressive set of transitions, able to support all structural properties required by the UCCA scheme. Its transition classifier is based on an MLP that receives a BiLSTM encoding of elements in the parser state (buffer, stack and intermediate graph), given word embeddings and other features.

3.2 The Semantic Rules

For performing DSS, we define two simple splitting rules, conditioned on UCCA's categories. We currently only consider Parallel Scenes and Elaborator Scenes, not separating Participant Scenes, in order to avoid splitting in cases of nominalizations or indirect speech. For example, the sentence "His arrival surprised everyone", which has, in addition to the Scene evoked by "surprised", a Participant Scene evoked by "arrival", is not split here.

Rule #1. Parallel Scenes of a given sentence are extracted, separated in different sentences, and concatenated according to the order of appearance. More formally, given a decomposition of a sentence S into parallel Scenes $Sc_1, Sc_2, \cdots Sc_n$ (indexed by the order of the first token), we obtain the

following rule, where "|" is the sentence delimiter:

$$S \longrightarrow Sc_1|Sc_2|\cdots|Sc_n$$

As UCCA allows argument sharing between Scenes, the rule may duplicate the same sub-span of S across sentences. For example, the rule will convert "He came back home and played piano" into "He came back home"|"He played piano."

Rule #2. Given a sentence S, the second rule extracts Elaborator Scenes and corresponding minimal centers. Elaborator Scenes are then concatenated to the original sentence, where the Elaborator Scenes, except for the minimal center they elaborate, are removed. Pronouns such as "who", "which" and "that" are also removed.

Formally, if $\{(Sc_1, C_1)\cdots(Sc_n, C_n)\}$ are the Elaborator Scenes of S and their corresponding minimal centers, the rewrite is:

$$S \longrightarrow S - \bigcup_{i=1}^{n}(Sc_i - C_i)|Sc_1|\cdots|Sc_n$$

where $S-A$ is S without the unit A. For example, this rule converts the sentence "He observed the planet which has 14 known satellites" to "He observed the planet| Planet has 14 known satellites.". Article regeneration is not covered by the rule, as its output is directly fed into the NMT component.

After the extraction of Parallel Scenes and Elaborator Scenes, the resulting simplified Parallel Scenes are placed before the Elaborator Scenes. See Figure 1.

4 Neural Component

The split sentences are run through the NTS state-of-the-art neural TS system (Nisioi et al., 2017), built using the OpenNMT neural machine translation framework (Klein et al., 2017). The architecture includes two LSTM layers, with hidden states of 500 units in each, as well as global attention combined with input feeding (Luong et al., 2015). Training is done with a 0.3 dropout probability (Srivastava et al., 2014). This model uses alignment probabilities between the predictions and the original sentences, rather than character-based models, to retrieve the original words.

We here consider the w2v initialization for NTS (N17), where word2vec embeddings of size 300 are trained on Google News (Mikolov et al., 2013a) and local embeddings of size 200 are trained on the training simplification corpus (Řehůřek and Sojka, 2010; Mikolov et al., 2013b). Local embeddings for the encoder are trained on the source side of the training corpus, while those for the decoder are trained on the simplified side.

For sampling multiple outputs from the system, beam search is performed during decoding by generating the first 5 hypotheses at each step ordered by the log-likelihood of the target sentence given the input sentence. We here explore both the highest (h1) and fourth-ranked (h4) hypotheses, which we show to increase the SARI score and to be much less conservative.[2] We thus experiment with two variants of the neural component, denoted by NTS-h1 and NTS-h4. The pipeline application of the rules and the neural system results in two corresponding models: SENTS-h1 and SENTS-h4.

5 Experimental Setup

Corpus All systems are tested on the test corpus of Xu et al. (2016),[3] comprising 359 sentences from the PWKP corpus (Zhu et al., 2010) with 8 references collected by crowdsourcing for each of the sentences.

Semantic component. The TUPA parser[4] is trained on the UCCA-annotated *Wiki* corpus.[5]

Neural component. We use the NTS-w2v model[6] provided by N17, obtained by training on the corpus of Hwang et al. (2015) and tuning on the corpus of Xu et al. (2016). The training set is based on manual and automatic alignments between standard English Wikipedia and Simple English Wikipedia, including both good matches and partial matches whose similarity score is above the 0.45 scale threshold (Hwang et al., 2015). The total size of the training set is about 280K aligned sentences, of which 150K sentences are full matches and 130K are partial matches.[7]

Comparison systems. We compare our findings to HYBRID, which is the state of the art for joint structural and lexical simplification, imple-

[2] Similarly, N17 considered the first two hypotheses and showed that h2 has an higher SARI score and is less conservative than h1.

[3] https://github.com/cocoxu/simplification (This also includes SARI tools and the SBMT-SARI system.)

[4] https://github.com/danielhers/tupa

[5] http://www.cs.huji.ac.il/~oabend/ucca.html

[6] https://github.com/senisioi/NeuralTextSimplification

[7] We also considered the default initialization for the neural component, using the NTS model without word embeddings. Experimenting on the tuning set, the w2v approach got higher BLEU and SARI scores (for h1 and h4 respectively) than the default approach.

mented by Zhang and Lapata (2017).[8] We use the released output of HYBRID, trained on a corpus extracted from Wikipedia, which includes the aligned sentence pairs from Kauchak (2013), the aligned revision sentence pairs in Woodsend and Lapata (2011), and the PWKP corpus, totaling about 296K sentence pairs. The tuning set is the same as for the above systems.

In order to isolate the effect of NMT, we also implement SEMoses, where the neural-based component is replaced by the phrase-based MT system Moses,[9] which is also used in HYBRID. The training, tuning and test sets are the same as in the case of SENTS. MGIZA[10] is used for word alignment. The KenLM language model is trained using the target side of the training corpus.

Additional baselines. We report human and automatic evaluation scores for Identity (where the output is identical to the input), for Simple Wikipedia where the output is the corresponding aligned sentence in the PWKP corpus, and for the SBMT-SARI system, tuned against SARI (Xu et al., 2016), which maximized the SARI score on this test set in previous works (Nisioi et al., 2017; Zhang and Lapata, 2017).

Automatic evaluation. The automatic metrics used for the evaluation are: (1) BLEU (Papineni et al., 2002) (2) SARI (System output Against References and against the Input sentence; Xu et al., 2016), which compares the n-grams of the system output with those of the input and the human references, separately evaluating the quality of words that are added, deleted and kept by the systems. (3) F_{add}: the addition component of the SARI score (F-score); (4) F_{keep}: the keeping component of the SARI score (F-score); (5) P_{del}: the deletion component of the SARI score (precision).[11] Each metric is computed against the 8 available references. We also assess system conservatism, reporting the percentage of sentences copied from the input (%Same), the averaged Levenshtein distance from the source (LD_{SC}, which considers additions, deletions, and substitutions), and the number of source sentences that are split (#Split).[12]

Human evaluation. Human evaluation is carried out by 3 in-house native English annotators, who rated the different input-output pairs for the different systems according to 4 parameters: Grammaticality (G), Meaning preservation (M), Simplicity (S) and Structural Simplicity (StS). Each input-output pair is rated by all 3 annotators. Elicitation questions are given in Table 1.

As the selection process of the input-output pairs in the test corpus of Xu et al. (2016), as well as their crowdsourced references, are explicitly biased towards lexical simplification, the use of human evaluation permits us to evaluate the structural aspects of the system outputs, even where structural operations are not attested in the references. Indeed, we show that system outputs may receive considerably higher structural simplicity scores than the source, in spite of the sample selection bias.

Following previous work (e.g., Narayan and Gardent, 2014; Xu et al., 2016; Nisioi et al., 2017), Grammaticality (G) and Meaning preservation (M) are measured using a 1 to 5 scale. Note that in the first question, the input sentence is not taken into account. The grammaticality of the input is assessed by evaluating the Identity transformation (see Table 2), providing a baseline for the grammaticality scores of the other systems.

Following N17, a -2 to +2 scale is used for measuring simplicity, where a 0 score indicates that the input and the output are equally complex. This scale, compared to the standard 1 to 5 scale, permits a better differentiation between cases where simplicity is hurt (the output is more complex than the original) and between cases where the output is as simple as the original, for example in the case of the identity transformation. Structural simplicity is also evaluated with a -2 to +2 scale. The question for eliciting StS is accompanied with a negative example, showing a case of lexical simplification, where a complex word is replaced by a simple one (the other questions appear without examples). A positive example is not included so as not to bias the annotators by revealing the nature of the operations we focus on (splitting and deletion). We follow N17 in applying human evaluation on the first 70 sentences of the test corpus.[13]

The resulting corpus, totaling 1960 sentence pairs, each annotated by 3 annotators, also include

[8] https://github.com/XingxingZhang/dress

[9] http://www.statmt.org/moses/

[10] https://github.com/moses-smt/mgiza

[11] Uniform tokenization and truecasing styles for all systems are obtained using the Moses toolkit.

[12] We used the NLTK package (Loper and Bird, 2002) for these computations.

[13] We do not exclude system outputs identical to the source, as done by N17.

the additional experiments described in Section 7 as well as the outputs of the NTS and SENTS systems used with the default initialization.

The inter-annotator agreement, using Cohen's quadratic weighted κ (Cohen, 1968), is computed as the average agreement of the 3 annotator pairs. The obtained rates are 0.56, 0.75, 0.47 and 0.48 for G, M, S and StS respectively.

System scores are computed by averaging over the 3 annotators and the 70 sentences.

G	Is the output fluent and grammatical?
M	Does the output preserve the meaning of the input?
S	Is the output simpler than the input?
StS	Is the output simpler than the input, ignoring the complexity of the words?

Table 1: Questions for the human evaluation.

	G	M	S	StS
Identity	4.80	5.00	0.00	0.00
Simple Wikipedia	4.60	4.21	0.83	0.38
Only MT-Based Simplification				
SBMT-SARI	3.71	3.96	0.14	-0.15
NTS-h1	4.56	**4.48**	0.22	0.15
NTS-h4	4.29	3.90	0.31	0.19
Only Structural Simplification				
DSS	3.42	4.15	0.16	0.16
Structural+MT-based Simplification				
Hybrid	2.96	2.46	0.43	0.43
SEMoses	3.27	3.98	0.16	0.13
SENTS-h1	3.98	3.33	**0.68**	**0.63**
SENTS-h4	3.54	2.98	0.50	0.36

Table 2: Human evaluation of the different NMT-based systems. Grammaticality (G) and Meaning preservation (M) are measured using a 1 to 5 scale. A -2 to +2 scale is used for measuring simplicity (S) and structural simplicity (StS) of the output relative to the input sentence. The highest score in each column appears in bold. Structural simplification systems are those that explicitly model structural operations.

6 Results

Human evaluation. Results are presented in Table 2. First, we can see that the two SENTS systems outperform HYBRID in terms of G, M, and S. SENTS-h1 is the best scoring system, under all human measures.

In comparison to NTS, SENTS scores markedly higher on the simplicity judgments. Meaning preservation and grammaticality are lower for SENTS, which is likely due to the more conservative nature of NTS. Interestingly, the application of the splitting rules by themselves does not yield a considerably simpler sentence. This likely stems from the rules not necessarily yielding grammatical sentences (NTS often serves as a grammatical error corrector over it), and from the incorporation of deletions, which are also structural operations, and are performed by the neural system.

An example of high structural simplicity scores for SENTS resulting from deletions is presented in Table 5, together with the outputs of the other systems and the corresponding human evaluation scores. NTS here performs lexical simplification, replacing the word "incursions" by "raids" or "attacks"'. On the other hand, the high StS scores obtained by DSS and SEMoses are due to sentence splittings.

Automatic evaluation. Results are presented in Table 3. Identity obtains much higher BLEU scores than any other system, suggesting that BLEU may not be informative in this setting. SARI seems more informative, and assigns the lowest score to Identity and the second highest to the reference.

Both SENTS systems outperform HYBRID in terms of SARI and all its 3 sub-components. The h4 setting (hypothesis #4 in the beam) is generally best, both with and without the splitting rules.

Comparing SENTS to using NTS alone (without splitting), we see that SENTS obtains higher SARI scores when hypothesis #1 is used and that NTS obtains higher scores when hypothesis #4 is used. This may result from NTS being more conservative than SENTS (and HYBRID), which is rewarded by SARI (conservatism is indicated by the %Same column). Indeed for h1, %Same is reduced from around 66% for NTS, to around 7% for SENTS. Conservatism further decreases when h4 is used (for both NTS and SENTS). Examining SARI's components, we find that SENTS outperforms NTS on F_{add}, and is comparable (or even superior for $h1$ setting) to NTS on P_{del}. The superior SARI score of NTS over SENTS is thus entirely a result of a superior F_{keep}, which is easier for a conservative system to maximize.

Comparing HYBRID with SEMoses, both of which use Moses, we find that SEMoses obtains higher BLEU and SARI scores, as well as G and M human scores, and splits many more sentences. HYBRID scores higher on the human simplicity measures. We note, however, that applying NTS alone is inferior to HYBRID in terms of simplicity, and that both components are required to obtain high simplicity scores (with SENTS).

We also compare the sentence splitting component used in our systems (namely DSS) to that used in HYBRID, abstracting away from deletion-based and lexical simplification. We therefore apply DSS to the test set (554 sentences) of the

	BLEU	SARI	F_{add}	F_{keep}	P_{del}	% Same	LD_{SC}	#Split
Identity	94.93	25.44	0.00	76.31	0.00	100	0.00	0
Simple Wikipedia	69.58	39.50	8.46	61.71	48.32	0.00	33.34	0
Only MT-Based Simplification								
SBMT-SARI	74.44	**41.46**	**6.77**	69.92	**47.68**	4.18	23.31	0
NTS-h1	**88.67**	28.73	0.80	**70.95**	14.45	66.02	17.13	0
NTS-h4	79.88	36.55	2.59	65.93	41.13	2.79	24.18	1
Only Structural Simplification								
DSS	76.57	36.76	3.82	68.45	38.01	8.64	25.03	208
Structural+MT-Based Simplification								
HYBRID	52.82	27.40	2.41	43.09	36.69	1.39	61.53	3
SEMoses	74.45	36.68	3.77	67.66	38.62	7.52	27.44	208
SENTS-h1	58.94	30.27	3.01	51.52	36.28	6.69	59.18	0
SENTS-h4	57.71	31.90	3.95	51.86	39.90	0.28	54.47	17

Table 3: The left-hand side of the table presents BLEU and SARI scores for the combinations of NTS and DSS, as well as for the baselines. The highest score in each column appears in bold. The right hand side presents lexical and structural properties of the outputs. %Same: proportion of sentences copied from the input; LD_{SC}: Averaged Levenshtein distance from the source; #Split: number of split sentences. Structural simplification systems are those that explicitly model structural operations.

	BLEU	SARI	F_{add}	F_{keep}	P_{del}	% Same	LD_{SC}	#Split	G	M	S	StS
Moses	92.58	28.19	0.16	75.73	8.70	79.67	3.22	0	4.25	4.78	0	0.04
SEMoses	74.45	36.68	3.77	67.66	38.62	7.52	27.44	208	3.27	3.98	**0.16**	0.13
SETrain1-Moses	91.24	33.06	0.41	76.07	22.69	60.72	4.47	1	4.23	4.54	-0.12	-0.13
SETrain2-Moses	**94.31**	26.71	0.07	**76.20**	3.85	92.76	1.45	0	4.73	**4.99**	0.01	-0.005
Moses$_{\text{LM}}$	92.66	28.19	0.18	75.68	8.71	79.39	3.43	0	4.55	4.82	-0.01	-0.04
SEMoses$_{\text{LM}}$	74.49	**36.70**	**3.79**	67.67	**38.65**	7.52	27.45	208	3.32	4.08	0.15	**0.14**
SETrain1-Moses$_{\text{LM}}$	85.68	36.52	2.34	72.85	34.37	27.30	6.71	33	4.03	4.63	-0.11	-0.12
SETrain2-Moses$_{\text{LM}}$	94.22	26.66	0.10	76.19	3.69	92.20	1.43	0	**4.75**	**4.99**	0.01	-0.01

Table 4: Automatic and human evaluation for the different combinations of Moses and DSS. The automatic metrics as well as the lexical and structural properties reported (%Same: proportion of sentences copied from the input; LD_{SC}: Averaged Levenshtein distance from the source; #Split: number of split sentences) concern the 359 sentences of the test corpus. Human evaluation, with the G, M, S, and StS parameters, is applied to the first 70 sentences of the corpus. The highest score in each column appears in bold.

WEB-SPLIT corpus (Narayan et al., 2017) (See Section 2), which focuses on sentence splitting. We compare our results to those reported for a variant of HYBRID used without the deletion module, and trained on WEB-SPLIT (Narayan et al., 2017). DSS gets a higher BLEU score (46.45 vs. 39.97) and performs more splittings (number of output sentences per input sentence of 1.73 vs. 1.26).

7 Additional Experiments

Replacing the parser by manual annotation. In order to isolate the influence of the parser on the results, we implement a semi-automatic version of the semantic component, which uses manual UCCA annotation instead of the parser, focusing of the first 70 sentences of the test corpus. We employ a single expert UCCA annotator and use the UCCAApp annotation tool (Abend et al., 2017).

Results are presented in Table 6, for both SENTS and SEMoses. In the case of SEMoses, meaning preservation is improved when manual UCCA annotation is used. On the other hand, simplicity degrades, possibly due to the larger number of Scenes marked by the human annotator (TUPA tends to under-predict Scenes). This effect doesn't show with SENTS, where trends are similar to the automatic parses case, and high simplicity scores are obtained. This demonstrates that UCCA parsing technology is sufficiently mature to be used to carry out structural simplification.

We also directly evaluate the performance of the parser by computing F1, Recall and Precision DAG scores (Hershcovich et al., 2017), against the manual UCCA annotation.[14] We obtain for primary edges (i.e. edges that form a tree structure) scores of 68.9 %, 70.5%, and 67.4% for F1, Recall and Precision respectively. For remotes edges (i.e. additional edges, forming a DAG), the scores are 45.3%, 40.5%, and 51.5%. These results are comparable with the out-of-domain results reported by Hershcovich et al. (2017).

Experiments on Moses. We test other variants of SEMoses, where phrase-based MT is used instead of NMT. Specifically, we incorporate semantic information in a different manner by implementing two additional models: (1) SETrain1-Moses, where a new training corpus is obtained by applying the splitting rules to the target side of the

[14] We use the evaluation tools provided in https://github.com/danielhers/ucca, ignoring 9 sentences for which different tokenizations of proper nouns are used in the automatic and manual parsing.

		G	M	S	StS
Identity	In return, Rollo swore fealty to Charles, converted to Christianity, and undertook to defend the northern region of France against the incursions of other Viking groups.	5.00	5.00	0.00	0.00
Simple Wikipedia	In return, Rollo swore fealty to Charles, converted to Christianity, and swore to defend the northern region of France against raids by other Viking groups.	4.67	5.00	1.00	0.00
SBMT-SARI	In return, Rollo swore fealty to Charles, converted to Christianity, and set out to defend the north of France from the raids of other viking groups.	4.67	4.67	0.67	0.00
NTS-h1	In return, Rollo swore fealty to Charles, converted to Christianity, and undertook to defend the northern region of France against the raids of other Viking groups.	5.00	5.00	1.00	0.00
NTS-h4	In return, Rollo swore fealty to Charles, converted to Christianity, and undertook to defend the northern region of France against the attacks of other Viking groups.	4.67	5.00	1.00	0.00
DSS	Rollo swore fealty to Charles. Rollo converted to Christianity. Rollo undertook to defend the northern region of France against the incursions of other viking groups.	4.00	4.33	1.33	1.33
HYBRID	In return Rollo swore, and undertook to defend the region of France., Charles, converted	2.33	2.00	0.33	0.33
SEMoses	Rollo swore put his seal to Charles. Rollo converted to Christianity. Rollo undertook to defend the northern region of France against the incursions of other viking groups.	3.33	4.00	1.33	1.33
SENTS-h1	Rollo swore fealty to Charles.	5.00	2.00	2.00	2.00
SENTS-h4	Rollo swore fealty to Charles and converted to Christianity.	5.00	2.67	1.33	1.33

Table 5: System outputs for one of the test sentences with the corresponding human evaluation scores (averaged over the 3 annotators). Grammaticality (G) and Meaning preservation (M) are measured using a 1 to 5 scale. A -2 to +2 scale is used for measuring simplicity (S) and structural simplicity (StS) of the output relative to the input sentence.

	G	M	S	StS
DSS^m	3.38	3.91	-0.16	-0.16
SENTS^m-h1	4.12	3.34	0.61	0.58
SENTS^m-h4	3.60	3.24	0.26	0.12
SEMoses^m	3.32	4.27	-0.25	-0.25
SEMoses$^m_{LM}$	3.43	4.28	-0.18	-0.19

Table 6: Human evaluation using manual UCCA annotation. Grammaticality (G) and Meaning preservation (M) are measured using a 1 to 5 scale. A -2 to +2 scale is used for measuring simplicity (S) and structural simplicity (StS) of the output relative to the input sentence. X^m refers to the semi-automatic version of the system X.

training corpus; (2) SETrain2-Moses, where the rules are applied to the source side. The resulting parallel corpus is concatenated to the original training corpus. We also examine whether training a language model (LM) on split sentences has a positive effect, and train the LM on the split target side. For each system X, the version with the LM trained on split sentences is denoted by X_{LM}.

We repeat the same human and automatic evaluation protocol as in §6, presenting results in Table 4. Simplicity scores are much higher in the case of SENTS (that uses NMT), than with Moses. The two best systems according to SARI are SEMoses and SEMoses$_{LM}$ which use DSS. In fact, they resemble the performance of DSS applied alone (Tables 2 and 3), which confirms the high degree of conservatism observed by Moses in simplification (Alva-Manchego et al., 2017). Indeed, all Moses-based systems that don't apply DSS as preprocessing are conservative, obtaining high scores for BLEU, grammaticality and meaning preservation, but low scores for simplicity. Training the LM on split sentences shows little improvement.

8 Conclusion

We presented the first simplification system combining semantic structures and neural machine translation, showing that it outperforms existing lexical and structural systems. The proposed approach addresses the over-conservatism of MT-based systems for TS, which often fail to modify the source in any way. The semantic component performs sentence splitting without relying on a specialized corpus, but only an off-the-shelf semantic parser. The consideration of sentence splitting as a decomposition of a sentence into its Scenes is further supported by recent work on structural TS evaluation (Sulem et al., 2018), which proposes the SAMSA metric. The two works, which apply this assumption to different ends (TS system construction, and TS evaluation), confirm its validity. Future work will leverage UCCA's cross-linguistic applicability to support multi-lingual TS and TS pre-processing for MT.

Acknowledgments

We would like to thank Shashi Narayan for sharing his data and the annotators for participating in our evaluation and UCCA annotation experiments. We also thank Daniel Hershcovich and the anonymous reviewers for their helpful advices. This work was partially supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and by the Israel Science Foundation (grant No. 929/17), as well as by the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proc. of ACL-13*, pages 228–238.

Omri Abend, Shai Yerushalmi, and Ari Rappoport. 2017. UCCAApp: Web-application for syntactic and semantic phrase-based annotation. In *Proc. of ACL'17, System Demonstrations*, pages 109–114.

Sandra Maria Aluísio and Caroline Gasperin. 2010. Foestering disgital inclusion and accessibility: The PorSimples project for simplification of Portuguese texts. In *Proc. of NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, pages 46–53.

Fernando Alva-Manchego, Joachim Bingel, Gustavo H. Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proc. of IJCNLP'17*, pages 295–305.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Rrepresentation for sembanking. *Proc. of Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Gianni Barlacchi and Sara Tonelli. 2013. ERNESTA: A sentence simplification tool for children's stories in Italian. In *Proc. of CICLing'13*, pages 476–487.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proc. of ACL'11*, pages 465–501.

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. 2016. HUME: Human UCCA-based evaluation of machine translation. In *Proc. of EMNLP'16*, pages 1264–1274.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for sentence simplification. In *Proc. of COLING'96*, pages 1041–1044.

Leshem Choshen and Omri Abend. 2018. Reference-less measure of faithfulness for grammatical error correction. In *Proc. of NAACL'18 (Short papers)*. To appear.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

William Coster and David Kauchak. 2011a. Learning to simplify sentences using Wikipedia. In *Proc. of ACL, Short Papers*, pages 1–9.

William Coster and David Kauchak. 2011b. Simple English Wikipedia: A new text simplification task. In *Proc. of ACL'11*, pages 665–669.

Robert M.W. Dixon. 2010a. *Basic Linguistic Theory: Grammatical Topics*, volume 2. Oxford University Press.

Robert M.W. Dixon. 2010b. *Basic Linguistic Theory: Methodology*, volume 1. Oxford University Press.

Robert M.W. Dixon. 2012. *Basic Linguistic Theory: Further Grammatical Topics*, volume 3. Oxford University Press.

Jury Ganitketitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proc. of NAACL-HLT'13*, pages 758–764.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planning. In *Proc. of ACL'17*, pages 179–188.

Goran Glavaš and Sanja Štajner. 2013. Event-centered simplification of news stories. In *Proc. of the Student Research Workshop associated with RANLP 2013*, pages 71–78.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL'17*, pages 1127–1138.

Bui Thanh Hung, Nguyen Le Minh, and Akira Shimazu. 2012. Sentence splitting for Vietnamese-English machine translation. In *Knowledge and Systems Engineering, 2012 Fourth International Conference*, pages 156–160.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from Standard Wikipedia to Simple Wikipedia. In *Proc. of NAACL'15*, pages 211–217.

Hans Kamp. 1981. A theory of truth and semantic representation. In *Formal methods in the study of language*. Mathematisch Centrum. Number pt.1 in Mathematical Centre tracts.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proc. of ACL'13*, pages 1537–1546.

Guillam Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Open NMT: Open-source toolkit for neural machine translation. ArXiv:1701.02810 [cs:CL].

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Buch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proc. of ACL'07 on interactive poster and demonstration sessions*, pages 177–180.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL'03*, pages 48–54.

Ronald W. Langacker. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford University Press, USA.

Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. In *Proc. of EMNLP'02*, pages 63–70.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. of EMNLP'15*, pages 1412–1421.

Pablo N Mendes, Max Jakob, and Christian Bizer. 2012. DBpedia: A multilingual cross-domain knowledge base. In *Proc. of LREC'12*, pages 1813–1817.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proc. of Workshop at International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Kshitij Mishra, Ankush Soni, Rahul Sharma, and Dipti Misra Sharma. 2014. Exploring the effects of sentence simplification on Hindi to English Machine Translation systems. In *Proc. of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society*, pages 21–29.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proc. of ACL'14*, pages 435–445.

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proc. of INLG'16*, pages 111–120.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proc. of EMNLP'17*, pages 617–627.

Christina Niklaus, Bernahard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proc. of COLING'16*.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proc. of ACL'17 (Short paper)*, pages 85–91.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL'02*, pages 311–318.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proc. of LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help?: text simplification strategies for people with dyslexia. In *Proc. of the 10th International Cross-Disciplinary Conference on Web Accesibility*, pages 15:1 – 15:10.

Violeta Seretan. 2012. Acquisition of syntactic simplification rules for French. In *Proc. of LREC'12*, pages 4019–4026.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Proc. of LEC*, pages 64–71.

Advaith Siddharthan. 2004. Syntactic simplification and text cohesion. Technical Report 597, University of Cambridge.

Advaith Siddharthan and M. A. Angrosh. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proc. of EACL'14*, pages 722–731.

Advaith Siddhathan. 2011. Text simplification using typed dependencies: A comparison of the robustness of different generation strategies. In *Proc. of the 13th European Workshop on Natural Language Generation*, pages 2–11. Association of Computational Linguistics.

David A. Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proc. of the 1st Workshop in Statistical Machine Translation*, pages 23–30.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Proc. of the 9th International Conference on Computational Processing of the Portuguese Language*, pages 30–39.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Sanja Štajner, Hannah Bechara, and Horacio Saggion. 2015. A deeper exploration of the standard PB-SMT approach to text simplification and its evaluation. In *Proc. of ACL'15 (Short papers)*, pages 823–828.

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 Popović. 2016. Can text simplification help machine translation. *Baltic J. Modern Computing*, 4:230–242.

Elior Sulem, Omri Abend, and Ari Rappoport. 2015. Conceptual annotations preserve structure across translations. In *Proc. of 1st Workshop on Semantics-Driven Statistical Machine Translation (S2Mt 2015)*, pages 11–22.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proc. of NAACL'18*. To appear.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proc. of EMNLP'11*, pages 409–420.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proc. of ACL'12*, pages 1015–1024.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: new data can help. *TACL*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *TACL*, 4:401–415.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc.of ACL'01*, pages 523–530.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proc. of EMNLP'17*, pages 595–605.

Yaoyuan Zhang, Zhenxu Ye, Dongyan Zhao, and Rui Yan. 2017. A constrained sequence-to-sequence neural model for sentence simplification. ArXiv:1704.02312 [cs.CL].

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proc. of COLING'10*, pages 1353–1361.