Iterative Edit-Based Unsupervised Sentence Simplification

Dhruv Kumar,[1] **Lili Mou,**[2] **Lukasz Golab,**[1] **Olga Vechtomova**[1]
[1]University of Waterloo
[2]Department of Computing Science, University of Alberta
Alberta Machine Intelligence Institute (Amii)
{d35kumar,lgolab,ovechtomova}@uwaterloo.ca
doublepower.mou@gmail.com

Abstract

We present a novel iterative, edit-based approach to unsupervised sentence simplification. Our model is guided by a scoring function involving fluency, simplicity, and meaning preservation. Then, we iteratively perform word and phrase-level edits on the complex sentence. Compared with previous approaches, our model does not require a parallel training set, but is more controllable and interpretable. Experiments on Newsela and WikiLarge datasets show that our approach is nearly as effective as state-of-the-art supervised approaches.[1]

1 Introduction

Sentence simplification is the task of rewriting text to make it easier to read, while preserving its main meaning and important information. Sentence simplification is relevant in various real-world and downstream applications. For instance, it can benefit people with autism (Evans et al., 2014), dyslexia (Rello et al., 2013), and low-literacy skills (Watanabe et al., 2009). It can also serve as a preprocessing step to improve parsers (Chandrasekar et al., 1996) and summarization systems (Klebanov et al., 2004).

Recent efforts in sentence simplification have been influenced by the success of machine translation. In fact, the simplification task is often treated as monolingual translation, where a complex sentence is translated to a simple one. Such simplification systems are typically trained in a supervised way by either phrase-based machine translation (PBMT, Wubben et al., 2012; Narayan and Gardent, 2014; Xu et al., 2016) or neural machine translation (NMT, Zhang and Lapata, 2017; Guo et al., 2018; Kriz et al., 2019). Recently, sequence-to-sequence (Seq2Seq)-based NMT systems are shown to be more successful and serve as the state of the art.

However, supervised Seq2Seq models have two shortcomings. First, they give little insight into the simplification operations, and provide little control or adaptability to different aspects of simplification (e.g., lexical vs. syntactical simplification). Second, they require a large number of complex-simple aligned sentence pairs, which in turn require considerable human effort to obtain.

In previous work, researchers have addressed some of the above issues. For example, Alva-Manchego et al. (2017) and Dong et al. (2019) explicitly model simplification operators such as word insertion and deletion. Although these approaches are more controllable and interpretable than standard Seq2Seq models, they still require large volumes of aligned data to learn these operations. To deal with the second issue, Surya et al. (2019) recently proposed an unsupervised neural text simplification approach based on the paradigm of style transfer. However, their model is hard to interpret and control, like other neural network-based models. Narayan and Gardent (2016) attempted to address both issues using a pipeline of lexical substitution, sentence splitting, and word/phrase deletion. However, these operations can only be executed in a fixed order.

In this paper, we propose an iterative, edit-based unsupervised sentence simplification approach, motivated by the shortcomings of existing work. We first design a scoring function that measures the quality of a candidate sentence based on the key characteristics of the simplification task, namely, fluency, simplicity, and meaning preservation. Then, we generate simplified candidate sentences by iteratively editing the given complex sentence using three simplification operations (lexical simplification, phrase extraction, deletion and reordering). Our model seeks the best simplified

[1]Code is released at https://github.com/ddhruvkr/Edit-Unsup-TS

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7918–7928

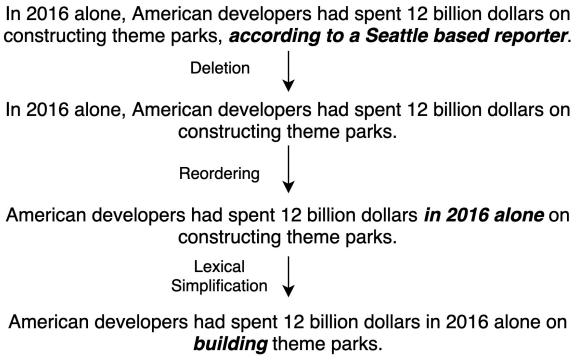

Figure 1: An example of three edit operations on a given sentence. Note that dropping clauses or phrases is common in text simplification datasets.

candidate sentence according to the scoring function. Compared with Narayan and Gardent (2016), the order of our simplification operations is not fixed and is decided by the model.

Figure 1 illustrates an example in which our model first chooses to delete a sentence fragment, followed by reordering the remaining fragments and replacing a word with a simpler synonym.

We evaluate our approach on the Newsela (Xu et al., 2015) and WikiLarge (Zhang and Lapata, 2017) corpora. Experiments show that our approach outperforms previous unsupervised methods and even performs competitively with state-of-the-art supervised ones, in both automatic metrics and human evaluations. We also demonstrate the interpretability and controllability of our approach, even without parallel training data.

2 Related Work

Early work used handcrafted rules for text simplification, at both the syntactic level (Siddharthan, 2002) and the lexicon level (Carroll et al., 1999). Later, researchers adopted machine learning methods for text simplification, modeling it as monolingual phrase-based machine translation (Wubben et al., 2012; Xu et al., 2016). Further, syntactic information was also considered in the PBMT framework, for example, constituency trees (Zhu et al., 2010) and dependency trees (Bingel and Søgaard, 2016). Narayan and Gardent (2014) performed probabilistic sentence splitting and deletion, followed by MT-based paraphrasing.

Nisioi et al. (2017) employed neural machine translation (NMT) for text simplification, using a sequence-to-sequence (Seq2Seq) model (Sutskever et al., 2014). Zhang and Lapata (2017) used reinforcement learning to optimize a reward based on simplicity, fluency, and relevance. Zhao et al. (2018a) integrated the transformer architecture and paraphrasing rules to guide simplification learning. Kriz et al. (2019) produced diverse simplifications by generating and re-ranking candidates by fluency, adequacy, and simplicity. Guo et al. (2018) showed that simplification benefits from multi-task learning with paraphrase and entailment generation. Martin et al. (2019) enhanced the transformer architecture with conditioning parameters such as length, lexical and syntactic complexity.

Recently, edit-based techniques have been developed for text simplification. Alva-Manchego et al. (2017) trained a model to predict three simplification operators (keep, replace, and delete) from aligned pairs. Dong et al. (2019) employed a similar approach but in an end-to-end trainable manner with neural networks. However, these approaches are supervised and require large volumes of parallel training data; also, their edits are only at the word level. By contrast, our method works at both word and phrase levels in an unsupervised manner.

For unsupervised sentence simplification, Surya et al. (2019) adopted style-transfer techniques, using adversarial and denoising auxiliary losses for content reduction and lexical simplification. However, their model is based on a Seq2Seq network, which is less interpretable and controllable. They cannot perform syntactic simplification since syntax typically does not change in style-transfer tasks. Narayan and Gardent (2016) built a pipeline-based unsupervised framework with lexical simplification, sentence splitting, and phrase deletion. However, these operations are separate components in the pipeline, and can only be executed in a fixed order.

Unsupervised edit-based approaches have recently been explored for natural language generation tasks, such as style transfer, paraphrasing, and sentence error correction. Li et al. (2018) proposed edit-based style transfer without parallel supervision. They replaced style-specific phrases with those in the target style, which are retrieved from the training corpus. Miao et al. (2019) used Metropolis–Hastings sampling for constrained sentence generation. In this paper, we model text generation as a search algorithm, and design search objective and search actions specifically for text simplification. Concurrent work further shows the success of search-based unsupervised text generation for paraphrasing (Liu et al., 2020) and summa-

rization (Schumann et al., 2020).

3 Model

In this section, we first provide an overview of our approach, followed by a detailed description of each component, namely, the scoring function, the edit operations, and the stopping criteria.

3.1 Overview

We first define a scoring function as our search objective. It allows us to impose both hard and soft constraints, balancing the fluency, simplicity, and adequacy of candidate simplified sentences (Section 3.2).

Our approach iteratively generates multiple candidate sentences by performing a sequence of lexical and syntactic operations. It starts from the input sentence; in each iteration, it performs phrase and word edits to generate simplified candidate sentences (Section 3.3).

Then, a candidate sentence is selected according to certain criteria. This process is repeated until none of the candidates improve the score of the source sentence by a threshold value. The last candidate is returned as the simplified sentence (Section 3.4).

3.2 Scoring Function

Our scoring function is the product of several individual scores that evaluate various aspects of a candidate simplified sentence. This is also known as the product-of-experts model (Hinton, 2002).

SLOR score from a syntax-aware language model (f_{eslor}). This measures the language fluency and structural simplicity of a candidate sentence.

A probabilistic language model (LM) is often used as an estimate of sentence fluency (Miao et al., 2019). In our work, we make two important modifications to a plain LM.

First, we replace an LM's estimated sentence probability with the syntactic log-odds ratio (SLOR, Pauls and Klein, 2012), to better measure fluency and human acceptability. According to Lau et al. (2017), SLOR shows the best correlation to human acceptability of a sentence, among many sentence probability-based scoring functions. SLOR was also shown to be effective in unsupervised text compression (Kann et al., 2018).

Given a trained language model (LM) and a sentence s, SLOR is defined as

$$\text{SLOR}(s) = \frac{1}{|s|}(\ln(P_{\text{LM}}(s)) - \ln(P_{\text{U}}(s))) \quad (1)$$

where P_{LM} is the sentence probability given by the language model, $P_{\text{U}}(s) = \prod_{w \in s} P(w)$ is the product of the unigram probability of a word w in the sentence, and $|s|$ is the sentence length.

SLOR essentially penalizes a plain LM's probability by unigram likelihood and the length. It ensures that the fluency score of a sentence is not penalized by the presence of rare words. Consider two sentences, "*I went to England for vacation*" and "*I went to Senegal for vacation*." Even though both sentences are equally fluent, a standard LM will give a higher score to the former, since the word "England" is more likely to occur than "Senegal." In simplification, SLOR is preferred for preserving rare words such as named entities.[2]

Second, we use a syntax-aware LM, i.e., in addition to words, we use part-of-speech (POS) and dependency tags as inputs to the LM (Zhao et al., 2018b). For a word w_i, the input to the syntax-aware LM is $[\boldsymbol{e}(w_i); \boldsymbol{p}(w_i); \boldsymbol{d}(w_i)]$, where $\boldsymbol{e}(w_i)$ is the word embedding, $\boldsymbol{p}(w_i)$ is the POS tag embedding, and $\boldsymbol{d}(w_i)$ is the dependency tag embedding.

Note that our LM is trained on simple sentences. Thus, the syntax-aware LM prefers a syntactically simple sentence. It also helps to identify sentences that are structurally ungrammatical.

Cosine Similarity ($f_{\cos}$). Cosine similarity is an important measure of meaning preservation. We compute the cosine value between sentence embeddings of the original complex sentence (c) and the generated candidate sentence (s), where our sentence embeddings are calculated as the idf weighted average of individual word embeddings. Our sentence similarity measure acts as a hard filter, i.e., $f_{\cos}(s) = 1$ if $\cos(\boldsymbol{c}, \boldsymbol{s}) > \tau$, or $f_{\cos}(s) = 0$ otherwise, for some threshold τ.

Entity Score (f_{entity}). Entities help identify the key information of a sentence and therefore are also useful in measuring meaning preservation. Thus, we count the number of entities in the sentence as part of the scoring function, where entities are detected by a third-party tagger.

Length (f_{len}). This score is proportional to the inverse of the sentence length. It forces the model to generate shorter and simpler sentences. However, we reject sentences shorter than a specified length ($\leq$6 tokens) to prevent over-shortening.

[2]Note that we do not use SLOR to evaluate lexicon simplicity, which will later be evaluated by the Flesch reading ease (FRE) score. The SLOR score, in fact, preserves rare words, so that we can better design dictionary-based word substitution for lexical simplification (Section 3.3).

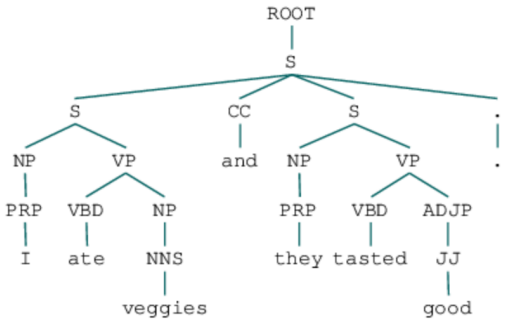

Figure 2: Constituency parse tree is used for detecting phrases.

FRE (f_{fre}). The Flesch Reading Ease (FRE) score (Kincaid et al., 1975) measures the ease of readability in text. It is based on text features such as the average sentence length and the average number of syllables per word. A higher scores indicate that the text is *simpler* to read.

We compute the overall scoring function as the product of individual scores.

$$f(s) = f_{\text{eslor}}(s)^{\alpha} \cdot f_{\text{fre}}(s)^{\beta} \cdot (1/f_{\text{len}}(s))^{\gamma} \cdot f_{\text{entity}}(s)^{\delta} \cdot f_{\cos}(s) \quad (2)$$

where the weights α, β, γ, and δ balance the relative importance of the different scores. Recall that the cosine similarity measure does not require a weight since it is a hard indicator function.

In Section 4.5, we will experimentally show that the weights defined for different scores affect different characteristics of simplification and thus provide more adaptability and controllability.

3.3 Generating Candidate Sentences

We generate candidate sentences by editing words and phrases. We use a third-party parser to obtain the constituency tree of a source sentence. Each clause- and phrase-level constituent (e.g., S, VP, and NP) is considered as a phrase. Since a constituent can occur at any depth in the parse tree, we can deal with both long and short phrases at different granularities. In Figure 2, for example, both "*good*" (ADJP) and "*tasted good*" (VP) are constituents and thus considered as phrases, whereas "*tasted*" is considered as a single word. For each phrase, we generate a candidate sentence using the edit operations explained below, with Figure 1 being a running example.

Removal. For each phrase detected by the parser, this operation generates a new candidate sentence by removing that phrase from the source sentence. In Figure 1, our algorithm can drop the phrase "*according to a Seattle based reporter*," which is not the main clause of the sentence. The removal operation allows us to remove peripheral information in a sentence for content reduction.

Extraction. This operation simply extracts a selected phrase (including a clause) as the candidate sentence. This allows us to select the main clause in a sentence and remove remaining peripheral information.

Reordering. For each phrase in a sentence, we generate candidate sentences by moving the phrase before or after another phrase (identified by clause- and phrase-level constituent tags). In the running example, the phrase "*In 2016 alone*" is moved between the phrases "*12 billion dollars*" and "*on constructing theme parks*." As seen, the reordering operation is able to perform syntactic simplification.

Substitution. In each phrase, we identify the most complex word as the rarest one according to the idf score. For the selected complex word, we generate possible substitutes using a two-step strategy.

First, we obtain candidate synonyms by taking the union of the WordNet synonym set (Miller, 1995) and the closest words from GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) embeddings (where embedding closeness is measured by Euclidean distance). Second, a candidate synonym is determined to be an appropriate simple substitute if it satisfies the following conditions: a) it has a lower idf score than the complex word, where the scores are computed from the target simple sentences, b) it is not a morphological inflection of the complex word, c) its word embedding exceeds a cosine similarity threshold to the complex word, and, d) it is has the same part-of-speech and dependency tags in the sentence as the complex word. We then generate candidate sentences by replacing the complex word with all qualified lexical substitutes. Notably, we do not replace entity words identified by entity taggers.

In our example sentence, consider the phrase "*constructing theme parks*." The word "constructing" is chosen as the word to be simplified, and is replaced with "building." As seen, this operation performs lexical simplification.

3.4 The Iterative Algorithm

Given an input complex sentence, our algorithm iteratively performs edits to search for a higher-

scoring candidate.

In each iteration, we consider all the operations (i.e., removal, extraction, reordering, and substitution). Each operation may generate multiple candidates (e.g., multiple words for substitution); we filter out a candidate sentence if the improvement does not pass an operation-specific threshold. We choose the highest-scoring sentence from those that are not filtered out. Our algorithm terminates if no edit passes the threshold, and the final candidate is our generated simplified sentence.

Our algorithm includes a filtering step for each operation. We only keep a candidate sentence if it is better than the previous one by a multiplicative factor, i.e.,

$$f(c)/f(s) > r_{\text{op}} \tag{3}$$

where s is the sentence given by the previous iteration, and c is a candidate generated by operator op from s.

Notably, we allow different thresholds for each operation. This provides control over different aspects of simplification, namely, lexicon simplification, syntactic simplification, and content reduction. A lower threshold for substitution, for example, encourages the model to perform more lexical simplification.

4 Experiments

4.1 Data

We use the Newsela (Xu et al., 2015) and the WikiLarge datasets (Zhang and Lapata, 2017) for evaluating our model.

Newsela is a collection of 1,840 news articles written by professional editors at 5 reading levels for children. We use the standard split and exclude simple-complex sentence pairs that are one reading level apart, following Zhang and Lapata (2017). This gives 95,208 training, 1,129 validation, and 1,077 test sentences.

The WikiLarge dataset is currently the largest text simplification corpus. It contains 296,402, 2,000, and 359 complex-simple sentence pairs for training, validation, and testing, respectively. The training set of WikiLarge consists of automatically aligned sentence pairs from the normal and simple Wikipedia versions. The validation and test sets contain multiple human-written references, against which we evaluate our algorithm.

For each corpus, we only use its training set to learn a language model of simplified sentences. For the WikiLarge dataset, we also train a Word2Vec embedding model from scratch on its source and target training sentences. These embeddings are used to obtain candidate synonyms in the substitution operation.

4.2 Training Details

For the LM, we use a two-layer, 256-dimensional recurrent neural network (RNN) with the gated recurrent unit (GRU, Chung et al., 2014). We initialize word embeddings using 300-dimensional GloVe (Pennington et al., 2014); out-of-vocabulary words are treated as `UNK`, initialized uniformly in the range of ± 0.05. Embeddings for POS tags and dependency tags are 150-dimensional, also initialized randomly. We fine-tune all embeddings during training.

We use the Averaged Stochastic Gradient Descent (ASGD) algorithm (Polyak and Juditsky, 1992) to train the LM, with 0.4 as the dropout and 32 as the batch size. For the Newsela dataset, the thresholds r_{op} in the scoring function are set to 1.25 for all the edit operations. All the weights in our scoring function $(\alpha, \beta, \gamma, \delta)$ are set to 1. For the WikiLarge dataset, the thresholds are set as 1.25 for the removal and reordering operations, 0.8 for substitution, and 5.0 for extraction. The weights in the scoring function $(\alpha, \beta, \gamma, \delta)$ are set to $0.5, 1.0, 0.25$ and 1.0, respectively.

We use CoreNLP (Manning et al., 2014) to construct the constituency tree and Spacy[3] to generate part-of-speech and dependency tags.

4.3 Competing Methods

We first consider the reference to obtain an upper-bound for a given evaluation metric. We also consider the complex sentence itself as a trivial baseline, denoted by `Complex`.

Next, we develop a simple heuristic that removes rare words occurring ≤ 250 times in the simple sentences of the training corpus, denoted by `Reduce-250`. As discussed in Section 4.4, this simple heuristic demonstrates the importance of balancing different automatic evaluation metrics.

For unsupervised competing methods, we compare with Surya et al. (2019), which is inspired by unsupervised neural machine translation. They proposed two variants, `UNMT` and `UNTS`, but their results are only available for WikiLarge.

[3] https://spacy.io/

Method	SARI↑	Add↑	Delete↑	Keep↑	BLEU↑	GM↑	FKGL↓	Len
Reference	70.13	-	-	-	100	83.74	3.20	12.75
Baselines								
Complex	2.82	-	-	-	21.30	7.75	8.62	23.06
Reduce-250	28.39	-	-	-	11.79	18.29	-0.23	14.48
Supervised Methods								
PBMT-R	15.77	3.07	38.34	5.90	18.1	16.89	7.59	23.06
Hybrid	28.61*	0.95*	78.86*	6.01*	14.46	20.34	4.03	12.41
EncDecA	24.12	2.73	62.66	6.98	21.68	22.87	5.11	16.96
Dress	27.37	**3.08**	71.61	7.43	23.2	25.2	4.11	14.2
Dress-Ls	26.63	3.21	69.28	7.4	**24.25**	25.41	4.21	14.37
DMass	31.06	1.25	84.12	7.82	11.92	19.24	3.60	15.07
S2S-All-FA	30.73	2.64	81.6	**7.97**	19.55	24.51	2.60	10.81
Edit-NTS	30.27*	2.71*	80.34*	7.76*	19.85	24.51	3.41	10.92
EncDecP	28.31	-	-	-	23.72	**25.91**	-	-
EntPar	**33.22**	2.42	**89.32**	7.92	11.14	19.24	1.34	7.88
Unsupervised Methods (Ours)								
RM+EX	26.07	2.35	68.35	7.5	**27.22**	26.64	2.95	12.9
RM+EX+LS	26.26	2.28	68.94	7.57	27.17	**26.71**	2.93	12.88
RM+EX+RO	26.99	**2.47**	70.88	7.63	26.31	26.64	3.14	12.81
RM+EX+LS+RO	27.11	2.40	71.26	**7.67**	26.21	26.66	3.12	12.81
RM+EX+LS+RO†	**30.44**	2.05	**81.77**	7.49	17.36	22.99	2.24	9.61

Table 1: Results on the Newsela dataset. † denotes the model with parameters tuned by SARI; other variants are tuned by the geometric mean (GM). ↑The higher, the better. ↓The lower, the better. * indicates a number that is different from that reported in the original paper. This is due to a mistreatment of capitalization in the previous work (confirmed by personal correspondence).

We also compare our model with supervised methods. First, we consider non-neural phrase-based machine translation (PBMT) methods: `PBMT-R` (Wubben et al., 2012), which re-ranks sentences generated by PBMT for diverse simplifications; `SBMT-SARI` (Xu et al., 2016), which uses an external paraphrasing database; and `Hybrid` (Narayan and Gardent, 2014), which uses a combination of PBMT and discourse representation structures. Next, we compare our method with neural machine translation (NMT) systems: `EncDecA`, which is a vanilla Seq2Seq model with attention (Nisioi et al., 2017); `Dress` and `Dress-Ls`, which are based on deep reinforcement learning (Zhang and Lapata, 2017); `DMass` (Zhao et al., 2018a), which is a transformer-based model with external simplification rules; `EncDecP`, which is an encoder-decoder model with a pointer-mechanism; `EntPar`, which is based on multi-task learning (Guo et al., 2018); `S2S-All-FA`, which a reranking based model focussing on lexical simplification (Kriz et al., 2019); and `Access`, which is based on the transformer architecture (Martin et al., 2019). Finally, we compare with a supervised edit-based neural model, `Edit-NTS` (Dong et al., 2019).

We evaluate our model with a different subset of operations, i.e., removal (`RM`), extraction (`EX`), reordering (`RO`), and lexical substitution (`LS`). In our experiments, we test the following variants: `RM+EX`, `RM+EX+LS`, `RM+EX+RO`, and `RM+EX+LS+RO`.

4.4 Automatic Evaluation

Tables 1 and 2 present the results of the automatic evaluation on the Newsela and WikiLarge datasets, respectively.

We use the SARI metric (Xu et al., 2016) to measure the simplicity of the generated sentences. SARI computes the arithmetic mean of the n-gram F1 scores of three rewrite operations: adding, deleting, and keeping. The individual F1-scores of these operations are reported in the columns "Add," "Delete," and "Keep."

We also compute the BLEU score (Papineni et al., 2002) to measure the closeness between a candidate and a reference. Xu et al. (2016) and Sulem et al. (2018) show that BLEU correlates with human judgement on fluency and meaning preservation for text simplification.[4]

[4]This does not hold when sentence splitting is involved. In our datasets, however, sentence splitting is rare, for example, 0.18% in the Newsela validation set).

Method	SARI↑	Add↑	Delete↑	Keep↑	BLEU↑	FKGL↓	Len
Baselines							
Complex	27.87	-	-	-	99.39	-	22.61
Supervised Methods							
PBMT-R	38.56	5.73	36.93	73.02	81.09	8.33	22.35
Hybrid	31.40	1.84	45.48	46.87	48.67	**4.56**	13.38
EncDecA	35.66	2.99	28.96	**75.02**	**89.03**	8.42	21.26
Dress	37.08	2.94	43.15	65.15	77.41	6.59	16.14
Dress-Ls	37.27	2.81	42.22	66.77	80.44	6.62	16.39
Edit-NTS	38.23	3.36	39.15	72.13	86.69	7.30	18.87
EntPar	37.45	-	-	-	81.49	7.41	-
Access	**41.87**	**7.28**	**45.79**	72.53	75.46	7.22	22.27
Models using external knowledge base							
SBMT-SARI	39.96	5.96	41.42	72.52	73.03	7.29	23.44
DMass	40.45	5.72	42.23	73.41	-	7.79	-
Unsupervised Methods							
UNMT	35.89	1.94	37.68	68.04	70.61	8.23	21.85
UNTS	37.20	1.50	41.27	68.81	74.02	7.84	19.05
RM+EX	36.46	1.68	35.17	**72.54**	**88.90**	6.47	18.62
RM+EX+LS	**37.85**	**2.31**	43.65	67.59	73.62	**6.30**	18.45
RM+EX+RO	36.54	1.73	36.10	71.79	85.07	6.89	19.24
RM+EX+LS+RO	37.58	2.30	**43.97**	66.46	70.15	6.69	19.54

Table 2: Results on the WikiLarge dataset. ↑The higher, the better. ↓The lower, the better.

In addition, we include a few intrinsic measures (without reference) to evaluate the quality of a candidate sentence: the Flesch–Kincaid grade level (FKGL) evaluating the ease of reading, as well as the average length of the sentence.

A few recent text simplification studies (Dong et al., 2019; Kriz et al., 2019) did not use BLEU for evaluation, noticing that the complex sentence itself achieves a high BLEU score (albeit a low SARI score), since the complex sentence is indeed fluent and preserves meaning. This is also shown by our `Complex` baseline.

For the Newsela dataset, however, we notice that the major contribution to the SARI score is from the deletion operation. By analyzing previous work such as `EntPar`, we find that it reduces the sentence length to a large extent, and achieves high SARI due to the extremely high F1 score of "Delete." However, its BLEU score is low, showing the lack of fluency and meaning. This is also seen from the high SARI of (`Reduce-250`) in Table 1. Ideally, we want both high SARI and high BLEU, and thus, we calculate the geometric mean (GM) of them as the main evaluation metric for the Newsela dataset.

On the other hand, this is not the case for WikiLarge, since none of the models can achieve high SARI by using only one operation among "Add," "Delete," and "Keep." Moreover, the complex sentence itself yields an almost perfect BLEU score (partially due to the multi-reference nature of WikiLarge). Thus, we do not use GM, and for this dataset, SARI is our main evaluation metric.

Overall results on Newsela. Table 1 shows the results on Newsela. By default (without †), validation is performed using the GM score. Still, our unsupervised text simplification achieves a SARI score around 26–27, outperforming quite a few supervised methods. Further, we experiment with SARI-based validation (denoted by †), following the setting of most previous work (Dong et al., 2019; Guo et al., 2018). We achieve 30.44 SARI, which is competitive with state-of-the-art supervised methods.

Our model also achieves high BLEU scores. As seen, all our variants, if validated by GM (without †), outperform competing methods in BLEU. One of the reasons is that our model performs text simplification by making edits on the original sentence instead of rewriting it from scratch.

In terms of the geometric mean (GM), our unsupervised approach outperforms all previous work, showing a good balance between simplicity and content preservation. The readability of our generated sentences is further confirmed by the intrinsic FKGL score.

Method	SARI↑	Add↑	Delete↑	Keep↑	BLEU↑	GM↑	FKGL↓	Len
RM+EX+LS+RO	27.11	2.40	71.26	7.67	26.21	26.66	3.12	12.81
− SLOR	27.63	2.22	73.20	7.49	24.14	25.83	2.61	12.37
− syntax-awareness	26.91	2.16	71.19	7.39	24.98	25.93	3.65	12.76

Table 3: Ablation test of the SLOR score based on syntax-aware language modeling.

Value	SARI↑	BLEU↑	GM↑	FRE↑	Len
Effect of threshold r_{op}					
1.0	29.20	21.69	25.17	83.75	11.75
1.1	28.38	23.59	25.87	82.83	12.17
1.2	27.45	25.54	26.48	81.98	12.62
1.3	26.60	**26.47**	26.53	81.47	13.07
Effect of weight α for f_{eslor}					
0.75	27.04	25.75	26.39	83.46	12.46
1.25	26.91	25.96	26.43	81.26	12.96
1.50	26.74	25.20	25.96	80.94	13.06
2.0	26.83	24.29	25.53	80.11	13.15
Effect of weight β for f_{fre}					
0.5	26.42	25.53	25.97	78.61	13.20
1.5	27.38	26.04	**26.70**	84.31	12.58
2.0	27.83	25.27	26.52	87.03	12.26
3.0	28.29	23.69	26.52	**90.34**	11.91
Effect of weight γ for $1/f_{\text{len}}$					
0.5	24.54	25.06	24.80	80.49	14.55
2.0	29.00	21.65	25.06	82.69	10.93
3.0	29.93	19.05	23.88	82.20	10.09
4.0	**30.44**	17.36	22.99	80.86	9.61
Effect of weight δ for f_{entity}					
0.5	27.81	24.68	26.20	83.6	12.01
2.0	25.44	24.63	25.03	79.36	14.28

Table 4: Analysis of the threshold value of the stopping criteria and relative weights in the scoring function.

Overall results on WikiLarge. For the Wikilarge experiments in Table 2, we perform validation on SARI, which is the main metric in this experiment. Our model outperforms existing unsupervised methods, and is also competitive with state-of-the-art supervised methods.

We observe that lexical simplification (LS) is important in this dataset, as its improvement is large compared with the Newsela experiment in Table 1. Additionally, reordering (RO) does not improve performance, as it is known that WikiLarge does not focus on syntactic simplification (Xu et al., 2016). The best performance for this experiment is obtained by the RM+EX+LS model.

4.5 Controllability

We now perform a detailed analysis of the scoring function described in Section 3.2 to understand the effect on different aspects of simplification. We use the RM+EX+LS+RO variant and the Newsela corpus as the testbed.

The SLOR score with syntax-aware LM. We analyze our syntax-aware SLOR score in the search objective. First, we remove the SLOR score and use the standard sentence probability. We observe that SLOR helps preserve rare words, which may be entities. As a result, the readability score (FKGL) becomes better (i.e., lower), but the BLEU score decreases. We then evaluate the importance of using a structural LM instead of a standard LM. We see a decrease in both SARI and BLEU scores. In both cases, the GM score decreases.

Threshold values and relative weights. Table 4 analyzes the effect of the hyperparameters of our model, namely, the threshold in the stopping criteria and the relative weights in the scoring function.

As discussed in Section 3.4, we use a threshold as the stopping criteria for our iterative search algorithm. For each operation, we require that a new candidate should be better than the previous iteration by a multiplicative threshold r_{op} in Equation (3). In this analysis, we set the same threshold for all operations for simplicity. As seen in Table 4, increasing the threshold leads to better meaning preservation since the model is more conservative (making fewer edits). This is shown by the higher BLEU and lower SARI scores.

Regarding the weights for each individual scoring function, we find that increasing the weight β for the FRE readability score makes sentences shorter, more readable, and thus simpler. This is also indicated by higher SARI values. When sentences are rewarded for being short (with large γ), SARI increases but BLEU decreases, showing less meaning preservation. The readability scores initially increase with the reduction in length, but then decrease. Finally, if we increase the weight δ for the entity score, the sentences become longer and more complex since the model is penalized more for deleting entities.

In summary, the above analysis shows the controllability of our approach in terms of different simplification aspects, such as simplicity, meaning preservation, and readability.

4.6 Human Evaluation

We conducted a human evaluation on the Newsela dataset since automated metrics may be insufficient for evaluating text generation. We chose 30 sentences from the test set for annotation and considered a subset of baselines. For our model variants, we chose RM+EX+LS+RO, considering both validation settings (GM and SARI).

We followed the evaluation setup in Dong et al. (2019), and measure the adequacy (*How much meaning from the original sentence is preserved?*), simplicity (*Is the output simper than the original sentence?*), and fluency (*Is the output grammatical?*) on a five-point Likert scale. We recruited three volunteers, one native English speaker and two non-native fluent English speakers. Each of the volunteer was given 30 sentences from different models (and references) in a randomized order. Additionally, we asked the volunteers to measure the number of instances where models produce incorrect details or generate text that is not implied by the original sentence. We did this because neural models are known to hallucinate information (Rohrbach et al., 2018). We report the average count of false information per sentence, denoted as FI.

We observe that our model RM+EX+LS+RO (when validated by GM) performs better than Hybrid, a combination of PBMT and discourse representation structures, in all aspects. It also performs competitively with remaining supervised NMT models.

For adequacy and fluency, Dress-Ls performs the best since it produces relatively longer sentences. For simplicity, S2S-All-FA performs the best since it produces shorter sentences. Thus, a balance is needed between these three measures. As seen, RM+EX+LS+RO ranks second in terms of the average score in the list (reference excluded). The human evaluation confirms the effectiveness of our unsupervised text simplification, even when compared with supervised methods.

We also compare our model variants RM+EX+LS+RO (validated by GM) and RM+EX+LS+RO† (validated by SARI). As expected, the latter generates shorter sentences, performing better in simplicity but worse in adequacy and fluency.

Regarding false information (FI), we observe that previous neural models tend to generate more false information, possibly due to the vagueness in the continuous space. By contrast, our approach only uses neural networks in the scoring function, but performs discrete edits of words and phrases. Thus, we achieve high fidelity (low FI) similar to the non-neural Hybrid model, which also performs editing on discourse parsing structures with PBMT.

Method	A$^\uparrow$	S$^\uparrow$	F$^\uparrow$	Avg$^\uparrow$	FI$^\downarrow$
Hybrid	2.63	2.74	2.39	2.59	**0.03**
Dress-Ls	**3.29**	3.05	**4.11**	**3.48**	0.2
EntPar	1.92	2.97	3.16	2.68	0.47
S2S-All-FA	2.25	**3.24**	3.90	3.13	0.3
Edit-NTS	2.37	3.17	3.73	3.09	0.23
RM+EX+LS+RO	2.97	3.09	3.78	3.28	**0.03**
RM+EX+LS+RO†	2.58	3.21	3.33	3.04	0.07
Reference	2.91	3.49	4.46	3.62	0.77

Table 5: Human evaluation on Newsela, where we measure adequacy (A), simplicity (S), fluency (F), and their average score (Avg), based on 1–5 Likert scale. We also count average instances of false information per sentence (FI).

In summary, our model takes advantage of both neural networks (achieving high adequacy, simplicity, and fluency) and traditional phrase-based approaches (achieving high fidelity).

Interestingly, the reference of Newsela has a poor (high) FI score, because the editors wrote simplifications at the document level, rather than the sentence level.

5 Conclusion

We proposed an iterative, edit-based approach to text simplification. Our approach works in an unsupervised manner that does not require a parallel corpus for training. In future work, we plan to add paraphrase generation to generate diverse simple sentences.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), under grant Nos. RGPIN-2019-04897, RGPIN-2020-04465, and the Canada Research Chair Program. Lili Mou is also supported by AltaML, the Amii Fellow Program, and the Canadian CIFAR AI Chair Program. This research was supported in part by Compute Canada (www.computecanada.ca).

References

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 Eighth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 295–305.

Joachim Bingel and Anders Søgaard. 2016. Text simplification as tree labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 337–343.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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 (ACL)*, pages 3393–3402.

Richard Evans, Constantin Orăsan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 131–140.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 462–476.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, pages 313–323.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 735–747. Springer.

Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-weighted loss and diverse reranking for sentence simplification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3137–3147.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies NAACL-HLT*, pages 1865–1874.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 55–60.

Louis Martin, Benoît Sagot, Éric de la Clergerie, and Antoine Bordes. 2019. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. CGMH: Constrained sentence generation by Metropolis-Hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6834–6842.

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

George A. Miller. 1995. Wordnet: a lexical database for english.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 435–445.

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference (INLG)*, pages 111–120.

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 (ACL), 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 (ACL)*, pages 311–318.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 959–968.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Boris T Polyak and Anatoli B Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855.

Luz Rello, Clara Bayarri, Azuki Gòrriz, Ricardo Baeza-Yates, Saurabh Gupta, Gaurang Kanvinde, Horacio Saggion, Stefan Bott, Roberto Carlini, and Vasile Topac. 2013. Dyswebxia 2.0!: more accessible text for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, page 25.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4035–4045.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71. IEEE.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 738–744.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2058–2068.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 29–36.

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 (ACL)*, 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 (TACL)*, 3:283–297.

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

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

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018a. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3164–3173.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018b. A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 170–175.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 1353–1361.