Controllable Text Simplification with Explicit Paraphrasing

Mounica Maddela[1], Fernando Alva-Manchego[2], Wei Xu[1]
[1]School of Interactive Computing, Georgia Institute of Technology
[2]Department of Computer Science, University of Sheffield
{mounica.maddela, wei.xu}@cc.gatech.edu f.alva@sheffield.ac.uk

Abstract

Text Simplification improves the readability of sentences through several rewriting transformations, such as lexical paraphrasing, deletion, and splitting. Current simplification systems are predominantly sequence-to-sequence models that are trained end-to-end to perform all these operations simultaneously. However, such systems limit themselves to mostly deleting words and cannot easily adapt to the requirements of different target audiences. In this paper, we propose a novel hybrid approach that leverages linguistically-motivated rules for splitting and deletion, and couples them with a neural paraphrasing model to produce varied rewriting styles. We introduce a new data augmentation method to improve the paraphrasing capability of our model. Through automatic and manual evaluations, we show that our proposed model establishes a new state-of-the art for the task, paraphrasing more often than the existing systems, and can control the degree of each simplification operation applied to the input texts.[1]

1 Introduction

Text Simplification aims to improve the readability of texts with simpler grammar and word choices while preserving meaning (Saggion, 2017). It provides reading assistance to children (Kajiwara et al., 2013), non-native speakers (Petersen and Ostendorf, 2007; Pellow and Eskenazi, 2014; Paetzold, 2016), and people with reading disabilities (Rello et al., 2013). It also helps with downstream natural language processing tasks, such as parsing (Chandrasekar et al., 1996), semantic role labelling (Vickrey and Koller, 2008), information extraction (Miwa et al., 2010), and machine translation (MT, Chen et al., 2012; Štajner and Popovic, 2016).

Since 2016, nearly all text simplification systems have been sequence-to-sequence (seq2seq) models trained end-to-end, which have greatly increased the fluency of the outputs (Zhang and Lapata, 2017; Nisioi et al., 2017; Zhao et al., 2018; Kriz et al., 2019; Dong et al., 2019; Jiang et al., 2020). However, these systems mostly rely on deletion and tend to generate very short outputs at the cost of meaning preservation (Alva-Manchego et al., 2017). Table 1 shows that they neither split sentences nor paraphrase well as reflected by the low percentage of splits ($<1\%$) and new words introduced ($<11.2\%$). While deleting words is a viable (and the simplest) way to reduce the complexity of sentences, it is suboptimal and unsatisfying. Professional editors are known to use a sophisticated combination of deletion, paraphrasing, and sentence splitting to simplify texts (Xu et al., 2015).

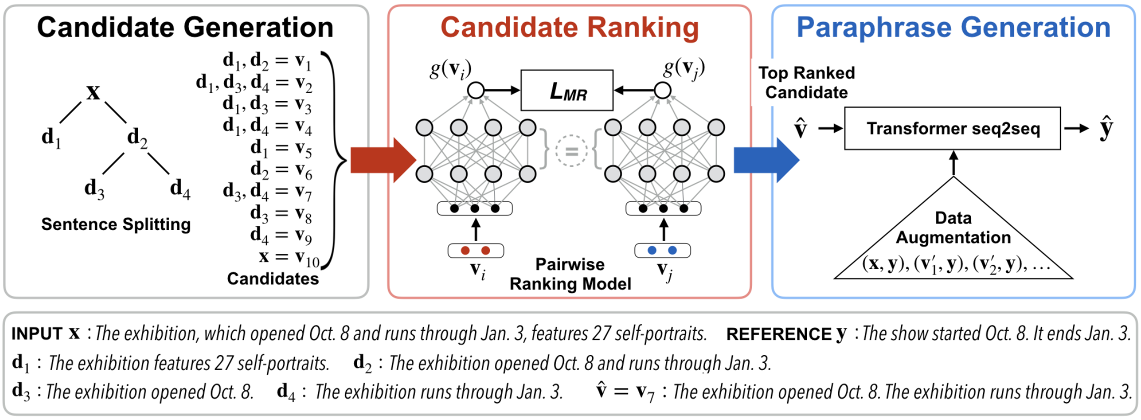
	OLen	**%new**	**%eq**	**%split**
Complex (input)	20.7	0.0	100.0	0.0
Narayan and Gardent (2014)†	10.4	0.7	0.8	0.4
Zhang and Lapata (2017)†	13.8	8.1	16.8	0.0
Dong et al. (2019)†	10.9	8.4	4.6	0.0
Kriz et al. (2019)†	10.8	11.2	1.2	0.0
LSTM	17.0	6.1	28.4	1.2
Our Model	17.1	17.0	3.0	31.8
Simple (reference)	17.9	29.0	0.0	30.0

Table 1: Output statistics of 500 random sentences from the Newsela test set. Existing systems rely on deletion and do not paraphrase well. **OLen**, **%new**, **%eq** and **%split** denote the average output length, percentage of new words added, percentage of system outputs that are identical to the inputs, and percentage of sentence splits, respectively. †We used the system outputs shared by their authors.

Another drawback of these end-to-end neural systems is the lack of controllability. Simplification is highly audience dependant, and what constitutes simplified text for one group of users may not be acceptable for other groups (Xu et al., 2015; Lee and Yeung, 2018). An ideal simplification system should be able to generate text with varied characteristics, such as different lengths, readability levels, and number of split sentences, which can be difficult to control in end-to-end systems.

[1] Our code and data are available at https://github.com/mounicam/controllable_simplification.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3536–3553
June 6–11, 2021. ©2021 Association for Computational Linguistics



INPUT $\mathbf{x}$: *The exhibition, which opened Oct. 8 and runs through Jan. 3, features 27 self-portraits.* **REFERENCE** $\mathbf{y}$: *The show started Oct. 8. It ends Jan. 3.*
$\mathbf{d}_1$: *The exhibition features 27 self-portraits.* $\mathbf{d}_2$: *The exhibition opened Oct. 8 and runs through Jan. 3.*
$\mathbf{d}_3$: *The exhibition opened Oct. 8.* $\mathbf{d}_4$: *The exhibition runs through Jan. 3.* $\hat{\mathbf{v}} = \mathbf{v}_7$: *The exhibition opened Oct. 8. The exhibition runs through Jan. 3.*

Figure 1: Overview of our proposed model for text simplification, which can perform a controlled combination of sentence splitting, deletion, and paraphrasing.

To address these issues, we propose a novel hybrid approach that combines linguistically-motivated syntactic rules with data-driven neural models to improve the diversity and controllability of the simplifications. We hypothesize that the seq2seq generation model will learn lexical and structural paraphrases more efficiently from the parallel corpus, when we offload some of the burden of sentence splitting (e.g., split at comma) and deletion (e.g., remove trailing preposition phrases) decisions to a separate component. Previous hybrid approaches for simplification (Narayan and Gardent, 2014; Siddharthan and Mandya, 2014; Sulem et al., 2018c) used splitting and deletion rules in a deterministic step before applying an MT-based paraphrasing model. In contrast, our approach provides a more flexible and dynamic integration of linguistic rules with the neural models through ranking and data augmentation (Figure 1).

We compare our method to several state-of-the-art systems in both automatic and human evaluations. Our model achieves overall better performance measured by SARI (Xu et al., 2016) and other metrics, showing that the generated outputs are more similar to those written by human editors. We also demonstrate that our model can control the extent of each simplification operation by: (1) imposing a soft constraint on the percentage of words to be copied from the input in the seq2seq model, thus limiting lexical paraphrasing; and (2) selecting candidates that underwent a desired amount of splitting and/or deletion. Finally, we create a new test dataset with multiple human references for Newsela (Xu et al., 2015), the widely used text simplification corpus, to specifically evaluate lexical paraphrasing.

2 Our Approach

Figure 1 shows an overview of our hybrid approach. We combine linguistic rules with data-driven neural models to improve the controllability and diversity of the outputs. Given an input complex sentence $\mathbf{x}$, we first generate a set of intermediate simplifications $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ that have undergone splitting and deletion (§2.1). These intermediate sentences are then used for two purposes: (1) Selected by a pairwise neural ranking model (§2.2) based on the simplification quality and then rewritten by the paraphrasing component; (2) Used for data augmentation to improve the diversity of the paraphrasing model (§2.3).

2.1 Splitting and Deletion

We leverage the state-of-the-art system for structural simplification, called **DisSim** (Niklaus et al., 2019), to generate candidate simplifications that focus on splitting and deletion.[2] The English version of DisSim applies 35 hand-crafted grammar rules to break down a complex sentence into a set of hierarchically organized sub-sentences (see Figure 1 for an example). We choose a rule-based approach for sentence splitting because it works really well. In our pilot experiments, DisSim successfully split 92% of 100 complex sentences from the training data with more than 20 words, and introduced errors for only 6.8% of these splits. We consider these sub-sentences as candidate simplifications for the later steps, except those that are extremely short or long (compression ratio $\notin$ [0.5, 1.5]). The compression ratio is calculated as the number of

[2] `https://github.com/Lambda-3/DiscourseSimplification`

words in a candidate simplification $\mathbf{v}^i$ (which may contain one or more sub-sentences) divided by that of the original sentence $\mathbf{x}$.

To further increase the variety of generated candidates, we supplement DisSim with a **Neural Deletion and Split** module trained on the text simplification corpus (§3.1). We use a Transformer seq2seq model with the same configuration as the base model for paraphrasing (§2.3). Given the input sentence $\mathbf{x}$, we constrain the beam search to generate 10 outputs with splitting and another 10 outputs without splitting. Then, we select the outputs that do not deviate substantially from $\mathbf{x}$ (i.e., Jaccard similarity $>$ 0.5). We add outputs from the two systems to the candidate pool V.

2.2 Candidate Ranking

We design a neural ranking model to score all the candidates that underwent splitting and deletion, $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, then feed the top-ranked one to the lexical paraphrasing model for the final output. We train the model on a standard text simplification corpus consisting of pairs of complex sentence $\mathbf{x}$ and manually simplified reference $\mathbf{y}$.

Scoring Function. To assess the "goodness" of each candidate $\mathbf{v}_i$ during training, we define the *gold* scoring function g^* as a length-penalized BERTscore:

$$g^*(\mathbf{v}_i, \mathbf{y}) = e^{-\lambda||\phi_{\mathbf{v}_i} - \phi_{\mathbf{y}}||} \times BERTScore(\mathbf{v}_i, \mathbf{y}) \quad (1)$$

BERTScore (Zhang et al., 2020b) is a text similarity metric that uses BERT (Devlin et al., 2019) embeddings to find soft matches between word pieces (Wu et al., 2016) instead of exact string matching. We introduce a length penalty to favor the candidates that are of similar length to the human reference $\mathbf{y}$ and penalize those that deviate from the target compression ratio $\phi_{\mathbf{y}}$. λ defines the extent of penalization and is set to 1 in our experiments. $\phi_{\mathbf{v}_i}$ represents the compression ratios of $\mathbf{v}_i$ compared to the input $\mathbf{x}$. In principle, other similarity metrics can also be used for scoring.

Pairwise Ranking Model. We train the ranking model in a pairwise setup since BERTScore is sensitive to the relative rather than absolute similarity, when comparing multiple candidates with the same reference. We transform the gold ranking of V ($|V| = n$) into n^2 pairwise comparisons for every candidate pair, and learn to minimize the pairwise ranking violations using hinge loss:

$$\begin{aligned} L_{MR} &= \frac{1}{m}\sum_{k=1}^{m}\frac{1}{n_k^2}\sum_{i=1}^{n_k}\sum_{j=1,i\neq j}^{n_k}\max(0, 1 - l_{ij}^k d_{ij}^k) \\ d_{ij}^k &= g(\mathbf{v}_i^k) - g(\mathbf{v}_j^k) \\ l_{ij}^k &= sign\left(g^*(\mathbf{v}_i^k, \mathbf{y}^k) - g^*(\mathbf{v}_j^k, \mathbf{y}^k)\right) \end{aligned} \quad (2)$$

where $g(.)$ is a feedforward neural network, m is the number of training complex-simple sentence pairs, k is the index of training examples, and n_k represents the number of generated candidates (§2.1). On average, n_k is about 14.5 for a sentence of 30 words, and can be larger for longer sentences. We consider 10 randomly sampled candidates for each complex sentence during training.

Features. For the feedforward network $g(.)$, we use the following features: number of words in $\mathbf{v}_i$ and $\mathbf{x}$, compression ratio of $\mathbf{v}_i$ with respect to $\mathbf{x}$, Jaccard similarity between $\mathbf{v}_i$ and $\mathbf{x}$, the rules applied on $\mathbf{x}$ to obtain $\mathbf{v}_i$, and the number of rule applications. We vectorize all the real-valued features using Gaussian binning (Maddela and Xu, 2018), which has shown to help neural models trained on numerical features (Liu et al., 2016; Sil et al., 2017; Zhong et al., 2020). We concatenate these vectors before feeding them to the ranking model. We score each candidate $\mathbf{v}_i$ separately and rank them in the decreasing order of $g(\mathbf{v}_i)$. We provide implementation details in Appendix A.

2.3 Paraphrase Generation

We then paraphrase the top-ranked candidate $\hat{\mathbf{v}} \in V$ to generate the final simplification output $\hat{\mathbf{y}}$. Our paraphrase generation model can explicitly control the extent of lexical paraphrasing by specifying the percentage of words to be copied from the input sentence as a soft constraint. We also introduce a data augmentation method to encourage our model to generate more diverse outputs.

Base Model. Our base generation model is a Transformer encoder-decoder initialized by the BERT checkpoint (**?**), which achieved the best reported performance on text simplification in the recent work (Jiang et al., 2020). We enhance this model with an attention-based copy mechanism to encourage lexical paraphrasing, while remaining faithful to the input.

Copy Control. Given the input candidate $\hat{\mathbf{v}} = (\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_l)$ of l words and the percentage of copying $cp \in (0, 1]$, our goal is to paraphrase the rest of $(1 - cp) \times l$ words in $\hat{\mathbf{v}}$ to a simpler version. To achieve this, we convert cp into a vector of the same dimension as BERT embeddings using Gaussian binning (Maddela and Xu, 2018) and add it to the beginning of the input sequence $\hat{\mathbf{v}}$. The Transformer encoder then produces a sequence of context-aware hidden states $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2 \ldots \mathbf{h}_l)$, where $\mathbf{h}_i$ corresponds to the hidden state of $\hat{v}_i$. Each $\mathbf{h}_i$ is fed into the copy network which predicts the probability p_i that word $\hat{v}_i$ should be copied to output. We create a new hidden state $\bar{\mathbf{h}}_i$ by adding $\mathbf{h}_i$ to a vector $\mathbf{u}$ scaled according to p_i. In other words, the scaled version of $\mathbf{u}$ informs the decoder whether the word should be copied. A single vector $\mathbf{u}$ is used across all sentences and hidden states, and is randomly initialized then updated during training. More formally, the encoding process can be described as follows:

$$\begin{aligned}(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_l) &= encoder([cp; \hat{v}_1, \hat{v}_2, \ldots, \hat{v}_l]) \\ \bar{\mathbf{h}}_i &= \mathbf{h_i} + p_i \cdot \mathbf{u}, \\ \bar{\mathbf{H}} &= (\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \ldots, \bar{\mathbf{h}}_l) \qquad (3)\end{aligned}$$

The Transformer decoder generates the output sequence from $\bar{\mathbf{H}}$. Our copy mechanism is incorporated into the encoder rather than copying the input words during the decoding steps (Gu et al., 2016; See et al., 2017). Unless otherwise specified, we use the average copy ratio of the training dataset, 0.7, for our experiments.

Multi-task Training. We train the paraphrasing model and the copy network in a multi-task learning setup, where predicting whether a word should be copied serves as an auxiliary task. The gold labels for this task are obtained by checking if each word in the input sentence also appears in the human reference. When a word occurs multiple times in the input, we rely on the monolingual word alignment results from JacanaAlign (Yao et al., 2013) to determine which occurrence is the one that gets copied. We train the Transformer model and the copy network jointly by minimizing the cross-entropy loss for both decoder generation and binary word classification. We provide implementation and training details in Appendix A.

Data Augmentation. The sentence pairs in the training corpus often exhibit a variable mix of splitting and deletion operations along with paraphrasing (see Figure 1 for an example), which makes it difficult for the encoder-decoder models to learn paraphrases. Utilizing DisSim, we create additional training data that focuses on lexical paraphrasing

For each sentence pair $\langle \mathbf{x}, \mathbf{y} \rangle$, we first generate a set of candidates $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ by applying DisSim to $\mathbf{x}$, as described in §2.1. Then, we select a a subset of V, called $V' = \{\mathbf{v}'_1, \mathbf{v}'_2, \ldots, \mathbf{v}'_{n'}\}$ $(V' \in V)$ that are fairly close to the reference $\mathbf{y}$, but have only undergone splitting and deletion. We score each candidate $\mathbf{v}_i$ using the length-penalized BERTScore $g^*(\mathbf{v}_i, \mathbf{y})$ in Eq. (1), and discard those with scores lower than 0.5. While calculating g^*, we set $\phi_{\mathbf{y}}$ and λ to 1 and 2 respectively to favor candidates of similar length to the reference $\mathbf{y}$. We also discard the candidates that have different number of split sentences with respect to the reference. Finally, we train our model on the filtered candidate-reference sentence pairs $\langle \mathbf{v}'_1, \mathbf{y} \rangle$, $\langle \mathbf{v}'_2, \mathbf{y} \rangle$, $\ldots$, $\langle \mathbf{v}'_{n'}, \mathbf{y} \rangle$, which focus on lexical paraphrasing, in addition to $\langle \mathbf{x}, \mathbf{y} \rangle$.

2.4 Controllable Generation

We can control our model to concentrate on specific operations. For split- or delete-focused simplification, we select candidates with desirable length or number of splits during the candidate generation step. We perform only the paraphrase generation step for paraphrase-focused simplification. The paraphrasing model is designed specifically to paraphrase with minimal deletion and without splitting. It retains the length and the number of split sentences in the output, thus preserving the extent of deletion and splitting controlled in the previous steps. We control the degree of paraphrasing by changing the copy ratio.

3 Experiments

In this section, we compare our approach to various sentence simplification models using both automatic and manual evaluations. We show that our model achieves a new state-of-the-art and can adapt easily to different simplification styles, such as paraphrasing and splitting without deletion.

3.1 Data and Experiment Setup

We train and evaluate our models on Newsela (Xu et al., 2015)[3] and Wikipedia copora (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011). Newsela consists of 1,882 news

[3] https://newsela.com/data/

Models	SARI	add	keep	del	FK	SLen	OLen	CR	%split	s-BL	%new	%eq
Complex (input)	15.9	0.0	47.6	0.0	12.0	23.7	23.8	1.0	0.0	100.0	0.0	100.0
Simple (reference)	90.5	86.8	86.6	98.2	7.4	14.4	19.0	0.83	28.0	35.5	33.0	0.0
LSTM	35.0	1.6	**45.5**	57.8	8.9	17.6	17.9	0.8	1.9	66.5	5.0	20.2
Hybrid-NG	35.8	1.9	41.8	63.7	9.9	21.2	23.7	1.0	11.6	59.7	8.8	5.1
Transformer$_{bert}$	37.0	3.1	43.6	64.4	8.1	**15.6**	20.2	0.87	**24.1**	58.8	12.8	10.2
EditNTS	38.1	1.6	45.8	66.5	8.5	16.0	21.4	0.92	32.0	71.4	8.3	**0.2**
Our Model	**38.7**	**3.3**	42.9	**70.0**	**7.9**	15.8	**20.1**	**0.86**	23.9	**48.7**	**16.2**	0.4

Table 2: Automatic evaluation results on NEWSELA-AUTO test set. We report **SARI, the main automatic metric** for simplification, and its three edit scores namely precision for delete (**del**) and F1 scores for **add** and **keep** operations. We also report FKGL (**FK**), average sentence length (**SLen**), output length (**OLen**), compression ratio (**CR**), self-BLEU (**s-BL**), percentage of sentence splits (**%split**), average percentage of new words added to the output (**%new**), and percentage of sentences identical to the input (**%eq**). **Bold** typeface denotes the best performances (i.e., closest to the reference).

articles with each article rewritten by professional editors for students in different grades. We used the complex-simple sentence pairs automatically aligned by Jiang et al. (2020), called the NEWSELA-AUTO dataset. To capture sentence splitting, we joined the adjacent sentences in the simple article that are aligned to the same sentence in the complex article. Following Štajner et al. (2015), we removed the sentence pairs with high (>0.9) and low (<0.1) BLEU (Papineni et al., 2002) scores, which mostly correspond to the near identical and semantically divergent sentence pairs respectively. The final dataset consists of 259,778 train, 32,689 validation and 33,391 test complex-simple sentence pairs, where ~30% of pairs involve sentence splitting. Besides Newsela, we also provide the details of experiments on Wikipedia corpus in Appendix F, which show similar trends.

To demonstrate that our model can be controlled to generate diverse simplifications, we evaluate under the following settings: (i) Standard evaluation on the NEWSELA-AUTO test set similar to the methodology in the recent literature (Jiang et al., 2020; Dong et al., 2019; Zhang and Lapata, 2017), and (ii) Evaluation on different subsets of the NEWSELA-AUTO test set that concentrate on a specific operation. We selected 9,356 sentence pairs with sentence splits for split-focused evaluation. Similarly, we chose 9,511 sentence pairs with compression ratio < 0.7 and without sentences splits to evaluate delete-focused simplification. We created a new dataset, called NEWSELA-TURK, to evaluate lexical paraphrasing.[4] Similar to the WIKIPEDIA-TURK benchmark corpus (Xu et al., 2016), NEWSELA-TURK consists of human-written references focused on lexical paraphrasing. We first selected sentence pairs from the NEWSELA-AUTO test set of roughly similar length (compression ratio between 0.8 and 1.2) and no sentence splits because they more likely involve paraphrasing. Then, we asked Amazon Mechanical Turk workers to simplify the complex sentence without any loss in meaning.[5] To ensure the quality of simplifications, we manually selected the workers using the qualification test proposed in Alva-Manchego et al. (2020), during which the workers were asked to simplify three sentences. We selected top 35% of the 300 workers that participated in the test. We periodically checked the submissions and removed the bad workers. In the end, we collected 500 sentences with 4 references for each sentence.

3.2 Existing Methods

We use the following simplification approaches as baselines: (i) **BERT-Initialized Transfomer** (**?**), where the encoder is initialized with BERT$_{base}$ checkpoint and the decoder is randomly initialized. It is the current state-of-the-art for text simplification (Jiang et al., 2020). (ii) **EditNTS** (Dong et al., 2019),[6] another state-of-the-art model that uses a neural programmer-interpreter (Reed and de Freitas, 2016) to predict the edit operation on each word, and then generates the simplified sentence. (iii) **LSTM baseline**, a vanilla encoder-decoder model used in Zhang and Lapata (2017). (iv) **Hybrid-NG** (Narayan and Gardent, 2014),[7] one of the best existing hybrid systems that performs splitting and deletion using a probabilistic model and lexical substitution with a phrase-based machine translation system. We retrained all the models on the NEWSLA-AUTO dataset.

[4] We also provide results on 8,371 sentence pairs of NEWSELA-AUTO test set with compression ratio > 0.9 and no splits in Appendix D, which show similar trends.

[5] We provide instructions in Appendix B

[6] https://github.com/yuedongP/EditNTS

[7] https://github.com/shashiongithub/Sentence-Simplification-ACL14

Models	SARI	add	keep	del	FK	SLen	OLen	CR	%split	s-BL	%new	%eq
Complex (input)	22.3	0.0	67.0	0.0	12.8	23.3	23.5	1.0	0.0	100.0	0.0	100.0
Simple (reference)	62.3	44.8	68.3	73.9	11.1	23.8	23.5	1.01	0.0	48.5	24.1	0.0
Hybrid-NG	38.2	2.8	57.0	54.8	10.7	21.6	23.1	0.98	7.0	57.2	9.1	1.4
Transformer$_{bert}$	36.0	3.3	54.9	49.8	8.9	16.1	20.2	0.87	23.0	58.7	13.3	7.6
EditNTS	37.4	1.6	61.0	49.6	9.5	16.9	21.9	0.94	0.0	73.1	5.8	**0.0**
Our Model	38.1	**3.9**	55.1	55.5	8.8	16.6	20.2	0.86	19.6	**50.4**	15.7	**0.0**
Our Model (no split; *cp* = 0.6)	39.0	3.8	57.7	55.6	**11.2**	22.1	22.9	0.98	0.2	55.9	**18.0**	1.0
Our Model (no split; *cp* = 0.7)	**41.0**	3.4	63.1	**56.6**	11.5	22.2	22.9	0.98	**0.0**	69.4	10.4	4.2
Our Model (no split; *cp* = 0.8)	40.6	2.9	**65.0**	54.0	11.8	**22.4**	**23.2**	**0.99**	**0.0**	77.7	6.6	10.8

Table 3: Automatic evaluation results on NEWSELA-TURK that focuses on paraphrasing (500 complex sentences with 4 human written paraphrases). We control the extent of paraphrasing of our models by specifying the percentage of words to be copied (cp) from the input as a soft constraint.

Models	SARI	add	keep	del	FK	SLen	OLen	CR	%split	s-BL	%new	%eq
Complex (input)	17.0	0.0	51.1	0.0	14.6	30.0	30.2	1.0	0.0	100.0	0.0	100.0
Simple (reference)	93.0	89.9	91.6	97.5	7.0	13.4	28.6	0.98	100.0	36.8	29.7	0.0
Hybrid-NG	37.1	2.2	44.9	64.1	11.6	25.5	**30.1**	**1.0**	17.3	57.7	8.7	1.6
Transformer$_{bert}$	39.5	4.2	47.3	67.0	8.8	17.1	25.3	0.85	39.7	57.7	11.9	5.2
EditNTS	38.9	1.5	49.1	66.2	9.1	16.9	26.2	0.88	50.3	71.2	7.2	0.2
Our Model	39.4	4.0	46.6	67.6	8.7	17.5	25.5	0.85	40.6	**48.3**	**15.6**	**0.1**
Our Model (w/ split)	**42.1**	**5.6**	**50.6**	**70.1**	**8.1**	**15.3**	30.3	1.02	**93.5**	60.7	12.4	1.1

Table 4: Automatic evaluation results on a splitting-focused subset of the NEWSELA-AUTO test set (9,356 sentence pairs with splitting). Our model chooses only candidates that have undergone splitting during the ranking step.

Models	SARI	add	keep	del	FK	SLen	OLen	CR	%split	s-BL	%new	%eq
Complex (input)	9.6	0.0	28.8	0.0	12.9	25.8	26.0	1.0	0.0	100.0	0.0	100.0
Simple (reference)	85.7	82.7	76.0	98.6	6.7	12.6	12.6	0.5	0.0	19.6	32.6	0.0
Hybrid-NG	35.8	1.4	27.0	79.1	10.6	22.7	25.9	1.0	13.3	58.9	8.7	3.6
Transformer$_{bert}$	36.8	2.2	29.6	78.7	8.4	**16.2**	21.7	0.85	27.7	57.9	12.3	8.2
EditNTS	37.1	1.0	29.7	80.7	8.8	16.6	23.1	0.91	36.6	71.8	7.8	0.6
Our Model	**39.2**	**2.4**	**29.8**	**85.3**	**8.2**	16.4	21.9	0.85	29.1	48.8	**15.6**	0.4
Our Model (no split; CR<0.7)	38.2	2.0	28.5	84.1	8.6	16.8	**17.5**	**0.68**	**0.1**	**42.0**	12.5	**0.2**

Table 5: Automatic evaluation results on a deletion-focused subset of the NEWSELA-AUTO test set (9,511 sentence pairs with compression ratio $<$ 0.7 and no sentence splits). Our model selects only candidates with similar compression ratio and no splits during ranking.

3.3 Automatic Evaluation

Metrics. We report **SARI** (Xu et al., 2016), which averages the F1/precision of n-grams ($n \in \{1, 2, 3, 4\}$) inserted, deleted and kept when compared to human references. More specifically, it computes the F1 score for the n-grams that are added (**add**),[8] which is an important indicator if a model is good at paraphrasing. The model's deletion capability is measured by the F1 score for n-grams that are kept (**keep**) and precision for those deleted (**del**).[9] To evaluate a model's paraphrasing capability and diversity, we calculate the BLEU score with respect to the input (**s-BL**), the percentage of new words (**%new**) added, and the percentage of system outputs identical to the input (**%eq**). Low s-BL, %eq, or high %new indicate that the system is less conservative. We also report Flesch-Kincaid (**FK**) grade level readability (Kincaid and Chissom, 1975), average sentence length (**SLen**), the percentage of splits (**%split**), compression ratio (**CR**), and average output length (**OLen**). We do not report BLEU because it often does not correlate with simplicity (Sulem et al., 2018a,b; Xu et al., 2016).

Results. Table 2 shows the results on NEWSELA-AUTO test set. Our model outperforms the state-of-the-art Transformer$_{bert}$ and EditNTS models with respect to SARI.[10] EditNTS and LSTM focus on

[8] We slightly improved the SARI implementation by Xu et al. (2016) to exclude the spurious ngrams while calculating the F1 score for **add**. For example, if the input contains the phrase "*is very beautiful*", the phrase "*is beautiful*" is treated as a new phrase in the original implementation even though it is caused by the delete operation.

[9] SARI score of a reference with itself may not always be 100 as it considers 0 divided by 0 as 0, instead of 1, when calculating n-gram precision and recall. This avoids the inflation of **del** scores when the input is same as the output.

[10] According to Jiang et al. (2020), a BERT-initialized Transformer performs better than EditNTS. We see a different behavior here because we retained sentence splits from 0-1, 1-2,

Model	Overall Simplification				Structural Simplification			
	Fluency	Adequacy	Simplicity	Average	Fluency	Adequacy	Has Split	Correct Split
Hybrid-NG	3.23*	2.96*	3.40*	3.19*	3.25*	3.53*	42%	15%
EditNTS	3.88*	**3.70**	3.67	3.75	4.08	3.81*	41%	18%
Transformer$_{bert}$	3.91	3.63	3.65*	3.73	4.15	3.65*	53%	49%
Our Model	**4.02**	3.65	**3.77**	**3.81**	4.19	4.13	97%	90%
Simple (reference)	4.12	3.64	3.97	3.84	4.41	4.10	100%	100%

Table 6: Human evaluation of 100 random simplifications from the NEWSELA-AUTO test set and the split-focused subset of the same test set. **Has Split** and **Correct Split** denote the percentage of the output sentences that have undergone splitting and the percentage of coherent splits respectively. * denotes that our model is significantly better than the corresponding baseline (according to a t-test with $p < 0.05$).

deletion as they show high self-BLEU (>66.5) and FK (>8.8) scores despite having compression ratios similar to other systems. Transformer model alone is rather conservative and copies 10.2% of the sentences directly to the output. Although Hybrid-NG makes more changes than any other baselines, its SARI and add scores are 3.7 and 1.7 points lower than our model indicating that it generates more errors. Our model achieves the lowest self-BLEU (48.7), FK (7.9), and percentage of sentences identical to the input (0.4), and the highest add (3.3) score and percentage of new words (16.2%). In other words, our system is the least conservative, generates more good paraphrases, and mimics the human references better. We provide examples of system outputs in Table 9 and Appendix C.

Tables 3, 4, and 5 show the results on NEWSELA-TURK, split-focused, and delete-focused subsets of NEWSELA-AUTO test set respectively. For these experiments, we configure our model to focus on specific operations (details in §2.4). Our model again outperforms the existing systems according to SARI, add score, and percentage of new words, which means that our model is performing more meaningful paraphrasing. We show that we can control the extent of paraphrasing by varying the copy ratio (cp). Our model splits 93.5% of the sentences, which is substantially better than the other models.

3.4 Human Evaluation

We performed two human evaluations: one to measure the overall simplification quality and the other to specifically capture sentence splitting.[11] For the first one, we asked five Amazon Mechanical Turk workers to evaluate fluency, adequacy and simplicity of 100 random simplifications from the NEWSELA-AUTO test set. We supplemented the fluency and adequacy ratings with binary questions described in Zhang et al. (2020a) for the second evaluation over another 100 simplifications from the NEWSELA-AUTO split-focused test set. We asked if the output sentence exhibits spitting and if the splitting occurs at the correct place. While fluency measures the grammaticality of the output, adequacy captures the extent of meaning preserved when compared to the input. Simplicity evaluates if the output is simpler than the input. Each sentence was rated on a 5-point Likert scale and we averaged the ratings from the five workers. We chose the majority value for the binary ratings. We used the output of our model that is tailored for sentence splitting for the second evaluation.

Table 6 demonstrates that our model achieves the best fluency, simplicity, and overall ratings. The adequacy rating is also very close to that of Transformer$_{bert}$ and EditNTS even though our model is performing more paraphrasing (Table 2), which verifies that the changes made by our system are meaningful. Our model achieves the most number of correct sentence splits (90%), and the highest fluency (4.19) for syntactic simplification, showing that it can generate more number of coherent sentence splits when compared to other models.

4 Model Analysis

In this section, we analyze the contribution of each model component and examine the system errors.

4.1 System Ablations

We evaluate our key design choices, namely candidate ranking that is based on length-penalized BERTScore and paraphrase generation that uses data augmentation and copy attention. Table 8 summarizes the results. Our pairwise ranking model (BERTScore$_{len}$) achieves an increase of 3.2 points in SARI when compared to choosing a random (Random) candidate. Randomly selecting a candidate also performs fairly well, indicating that the

2-3 readability levels in NEWSELA-AUTO, which contained more lexical overlaps and inflated the scores for EditNTS.

[11] We provide instructions in Appendix E.

	Examples
Good (49%)	
Complex	*The Seattle kids petitioned Washington state last year to adopt stricter science-based regulations to protect them against climate change.*
Simple	*The eight kids from Seattle who petitioned Washington state last year are fighting for the state to adopt stricter laws to protect their future against climate change.*
Our Model	*The Seattle kids **asked** the Washington state last year **to take tougher rules against** climate change.*
Hallucinations (13%)	
Complex	*"It's more of a family than living outside," said Jessica Konczal, 33, whose husband is Sgt. Matthew Konczal .*
Simple	*Jessica Konczal, age 33, says the base is its own type of family .*
Our Model	*"It's more of a family than living outside," said Jessica Konczal, **one of the protesters**.*
Fluency Errors (14%)	
Complex	*Parental feedback on the menu additions so far, from some of the early adopter markets, has been "phenomenal," Leverton said.*
Simple	*Leverton said parents love the new food.*
Our Model	*So far, **parents parents have feedback** on the menu **changes** has been **a great deal**, **he added**.*
Bad Substitution (15%)	
Complex	*One of the device's inventors explained to the president that the machine was a prototype.*
Simple	*One of the girls explained it was just a prototype.*
Our Model	*One of the inventors **told** the president that the **first design was a design**.*
Anaphora Resolution / Entailment (12%)	
Complex	*Sea slugs dubbed sacoglossans are some of the most remarkable biological burglars on the planet.*
Simple	*The sea slugs known as sacoglossans are some of the most interesting burglars on the planet.*
Our Model	*Scientists say **these** are some of the most **interesting creatures** on the planet.*
Human Reference Errors (4%)	
Complex	*They float in and out of formations that combine the underwater world with the stage.*
Simple	*In the dance on a round stage on the seabed, the dancers swim together and apart.*
Our Model	*They float in and out of **places** that combine stage with the underwater.*

Table 7: Error analysis of 200 simplifications from NEWSELA-AUTO test set generated by our model. **Red** marks the errors; **blue** marks good paraphrases.

	SARI	**FK**	**CR**	**%split**	**%new**
Complex (input)	15.9	12.2	1.0	0.0	0.0
Simple (reference)	90.5	7.5	0.83	28.9	32.8
Random Candidate	33.7	8.1	0.81	34.4	5.5
BERTScore_{len}	36.9	9.0	0.87	25.9	5.9
Our Model	**38.6**	8.4	0.88	26.1	**18.9**
− augmentation	37.6	7.9	0.86	**29.5**	12.7
− copy attn	36.0	8.1	0.87	26.2	15.9
− only Transformer	37.9	**7.7**	0.78	26.3	16.5
− only DisSim	37.2	8.3	**0.84**	27.1	18.0

Table 8: Model ablation study on dev set

sentence splitting and deletion models we chose are of good quality.

Compared to our final model (Our Model), its variants without data augmentation (− augmentation) and copy mechanism (− copy attn) suffer a drop of 1.0 and 2.6 points in SARI respectively and a decrease of at least 3.0% of new words, which demonstrates that these components encourage the system to paraphrase. Our model trained on only DisSim (− only DisSim) and Transformer (− only Transformer) candidates performs close to our best model (Our Model) in terms of SARI.

4.2 Error Analysis

To understand the errors generated by our model, we manually classified 200 simplifications from the NEWSELA-AUTO test set into the following categories: (a) **Good**, where the model generated meaningful simplifications, (b) **Hallucinations**, where the model introduced information not in the input, (c) **Fluency Errors**, where the model generated ungrammatical output, (d) **Anaphora Resolution**, where it was difficult to resolve pronouns in the output. (e) **Bad substitution**, where the model inserted an incorrect simpler phrase, and (e) **Human Reference Errors**, where the reference does not reflect the source sentence. Note that a simplification can belong to multiple error categories. Table 7 shows the examples of each category.

5 Related Work

Before the advent of neural networks, text simplification approaches performed each operation separately in a pipeline manner using either handcrafted rules (Carroll et al., 1999; Siddharthan, 2002; Siddharthan et al., 2004) or data-driven methods based on parallel corpora (Zhu et al., 2010; Woodsend and Lapata, 2011; Narayan and Gardent, 2014). Following neural machine translation, the trend changed to performing all the operations together end-to-end (Zhang and Lapata, 2017; Nisioi et al., 2017; Zhao et al., 2018; Alva-Manchego et al., 2017; Vu

	System Outputs
Complex	*Since 2010, project researchers have uncovered documents in Portugal that have revealed who owned the ship.*
Simple	*Since 2010, experts have been figuring out who owned the ship.*
Hybrid-NG	*since 2010, the project* ***scientists*** *have uncovered documents in portugal that have* ***about*** *who* ***owns*** *the ship.*
LSTM	*since 2010,* ***scientists*** *have uncovered documents in portugal that have revealed who owned the ship.*
Transformer$_{bert}$	***they discovered*** *that the ship* ***had been important****.*
EditNTS	*since 2010, project researchers have uncovered documents in portugal****.*** ***have*** *revealed who owned the ship.*
Our Model (cp = 0.6)	***scientists*** *have* ***found*** *a* ***secret deal. they have discovered*** *who owned the ship.*
Our Model (cp = 0.7)	***scientists*** *have* ***found*** *documents in portugal****. they have also found out*** *who owned the ship.*
Our Model (cp = 0.8)	***scientists*** *have* ***found*** *documents in portugal****. they have discovered*** *who owned the ship.*
Complex	*Experts say China's air pollution exacts a tremendous toll on human health.*
Simple	*China's air pollution is very unhealthy.*
Hybrid-NG	*experts say the* ***government's*** *air pollution exacts a toll on human health.*
LSTM	*experts say china's air pollution exacts a tremendous toll on human health.*
Transformer$_{bert}$	*experts say china's pollution has a tremendous* ***effect*** *on human health.*
EditNTS	*experts say china's air pollution* ***can cause*** *human health.*
Our Model (cp = 0.6)	*experts say china's air pollution* ***is a big problem*** *for human health.*
Our Model (cp = 0.7)	*experts say china 's air pollution* ***can cause a lot of damage*** *on human health.*
Our Model (cp = 0.8)	*experts say china 's air pollution* ***is*** *a* ***huge*** *toll on human health.*

Table 9: Examples of system outputs. **Red** marks the errors; **blue** marks good paraphrases. cp is a soft constraint that denotes the percentage of words that can be copied from the input.

et al., 2018; Kriz et al., 2019; Dong et al., 2019; Jiang et al., 2020) at the cost of controllability and performance as shown in this paper.

Controllable text simplification has been attempted before, but only with limited capability. Scarton and Specia (2018) and Martin et al. (2020) added additional tokens to the input representing grade level, length, lexical, and structural complexity constraints. Nishihara et al. (2019) proposed a loss which controls word complexity, while Mallinson and Lapata (2019) concatenated constraints to each word embedding. Kumar et al. (2020) proposed a linguistic scoring function to control the edits to the input.

Another long body of research focuses on a single simplification operation and can be broadly divided into three categories: (1) Lexical Simplification (Specia et al., 2012; Horn et al., 2014; Glavaš and Štajner, 2015; Paetzold and Specia, 2017, 2015; Maddela and Xu, 2018; Qiang et al., 2020), where complex words are substituted with simpler words. (2) Syntactic Simplification (Siddharthan, 2006; Aharoni and Goldberg, 2018; Botha et al., 2018; Niklaus et al., 2019), which deals exclusively with sentence splitting, and (3) Sentence Compression (Filippova et al., 2015; Rush et al., 2015; Nallapati et al., 2016; See et al., 2017; Baziotis et al., 2019), where the goal is to shorten the input sentence by removing its irrelevant content.

6 Conclusion

We proposed a novel hybrid approach for sentence simplification that performs better and produces more diverse outputs than the existing systems. We designed a new data augmentation method to encourage the model to paraphrase. We created a new dataset, NEWSELA-TURK, to evaluate paraphrasing-focused simplifications. We showed that our model can control various attributes of the simplified text, such as number of sentence splits, length, and number of words copied from the input.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. We thank Newsela for sharing the data and NVIDIA for providing GPU computing resources. This research is supported in part by the NSF award IIS-1822754, ODNI and IARPA via the BETTER program contract 19051600004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

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*.

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

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQ^3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the ACL*.

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

Han-Bin Chen, Hen-Hsen Huang, Hsin-Hsi Chen, and Ching-Ting Tan. 2012. A simplification-translation-restoration framework for cross-domain SMT applications. In *Proceedings of the 24th International Conference on Computational Linguistics*.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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*.

Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF sentence alignment model for text simplification. In *Proceedings of the Association for Computational Linguistics*.

Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing*.

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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*.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Vechtomova Olga. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the Association for Computational Linguistics*.

John Lee and Chak Yan Yeung. 2018. Personalizing lexical simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Dan Liu, Wei Lin, Shiliang Zhang, Si Wei, and Hui Jiang. 2016. Neural networks models for entity discovery and linking. *CoRR*, abs/1611.03558.

Mounica Maddela and Wei Xu. 2018. A word-complexity lexicon and a neural readability ranking model for lexical simplification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jonathan Mallinson and Mirella Lapata. 2019. Controllable sentence simplification: Employing syntactic and lexical constraints.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

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*.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. Transforming complex sentences into a semantic hierarchy. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*.

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*.

Gustavo Paetzold and Lucia Specia. 2015. LEXenstein: A framework for lexical simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*.

Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Gustavo Henrique Paetzold. 2016. *Lexical Simplification for Non-Native English Speakers*. Ph.D. thesis, University of Sheffield.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.

David Pellow and Maxine Eskenazi. 2014. An open corpus of everyday documents for simplification tasks. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations*.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: A corpus analysis. In *Proceedings of Workshop on Speech and Language Technology for Education*.

Jipeng Qiang, Yun Li, Zhu Yi, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. *Association for the Advancement of Artificial Intelligence*.

Scott E. Reed and Nando de Freitas. 2016. Neural programmer-interpreters. In *4th International Conference on Learning Representations*.

Luz Rello, Ricardo Baeza-Yates, and Horacio Saggion. 2013. The impact of lexical simplification by verbal paraphrases for people with and without dyslexia. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Proceedings of the Language Engineering Conference*.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*.

Advaith Siddharthan and Angrosh Mandya. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics*.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2017. Neural cross-lingual entity linking. In *Proceedings of the 30th Conference on Association for the Advancement of Artificial Intelligence*.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics: Proceedings of the Main Conference and the Shared Task and Proceedings of the Sixth International Workshop on Semantic Evaluation*.

Sanja Štajner, Hannah Béchara, and Horacio Saggion. 2015. A deeper exploration of the standard PB-SMT approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018c. Simple and effective text simplification using semantic and neural methods. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Sentence simplification with memory-augmented neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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*.

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*.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. A lightweight and high performance monolingual word aligner. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Li Zhang, Huaiyu Zhu, Siddhartha Brahma, and Yunyao Li. 2020a. Small but mighty: New benchmarks for split and rephrase.

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

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

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

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2020. Discourse level factors for sentence deletion in text simplification. In *Proceedings of the 2020 Conference on Association for the Advancement of Artificial Intelligence*.

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*.

A Implementation and Training Details

We implemented two separate Transformer models for neural deletion and split component (§2.1) and paraphrase generation (§2.3) using the Fairseq[12] toolkit. Both the encoder and decoder follow BERT$_{base}$[13] architecture, while the encoder is also initialized with BERT$_{base}$ checkpoint. For neural deletion and split component, we used a beam search of width 10 to generate candidates. The copy attention mechanism is a feedforward network containing 3 hidden layers, 1000 nodes in each layer with *tanh* activation, and a single linear output node with *sigmoid* activation. We used Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001, linear learning rate warmup of 40k steps, and 100k training steps. We used a batch size of 64. We used BERT WordPiece tokenizer. During inference, we constrained the beam-search to not repeat trigrams and emitted sentences that avoided aggressive deletion (compression ratio $\in [0.9, 1.2]$. We chose the best checkpoint based on the SARI score (Xu et al., 2016) on the dev set. We saved a checkpoint after every epoch. We did not perform any hyperparameter search and directly used the hyperparameters of the BERT-initialized Transformer described in **?**. The model takes 10 hours to train on 1 NVIDIA GeForce GPU.

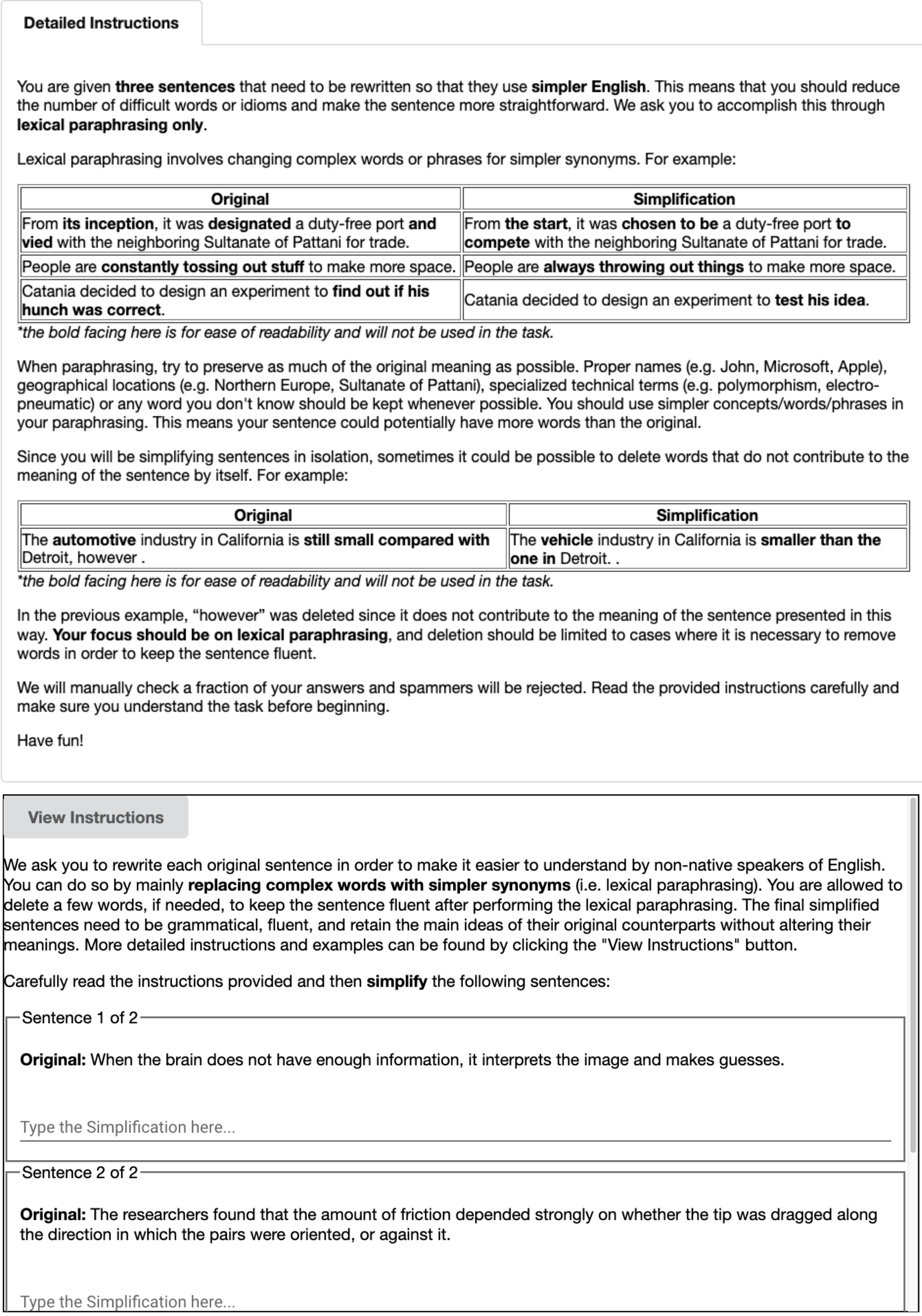
Our pairwise ranking model, implemented using the PyTorch framework, consists of 3 hidden layers, 100 nodes in each layer, $tanh$ activation, and a single linear output node. We used Adam optimizer with a learning rate of 0.01 and 10 epochs. We applied a dropout of 0.2. For Gaussian binning, we vectorized the numerical features into 10 dimensional vectors. The model takes half hour to train on 1 NVIDIA GeForce GPU. We do not perform any extensive hyperparameter tuning. We just examined few values for learning rate (0.001, 0.01 and 0.1) and chose the best based on the SARI score on the dev set. We used the original code for DisSim.[14]

[12] https://github.com/pytorch/fairseq

[13] https://github.com/google-research/bert

[14] https://github.com/Lambda-3/DiscourseSimplification

B Annotation Interface



You are given **three sentences** that need to be rewritten so that they use **simpler English**. This means that you should reduce the number of difficult words or idioms and make the sentence more straightforward. We ask you to accomplish this through **lexical paraphrasing only**.

Lexical paraphrasing involves changing complex words or phrases for simpler synonyms. For example:

Original	Simplification
From **its inception**, it was **designated** a duty-free port **and vied** with the neighboring Sultanate of Pattani for trade.	From **the start**, it was **chosen to be** a duty-free port **to compete** with the neighboring Sultanate of Pattani for trade.
People are **constantly tossing out stuff** to make more space.	People are **always throwing out things** to make more space.
Catania decided to design an experiment to **find out if his hunch was correct**.	Catania decided to design an experiment to **test his idea**.

**the bold facing here is for ease of readability and will not be used in the task.*

When paraphrasing, try to preserve as much of the original meaning as possible. Proper names (e.g. John, Microsoft, Apple), geographical locations (e.g. Northern Europe, Sultanate of Pattani), specialized technical terms (e.g. polymorphism, electro-pneumatic) or any word you don't know should be kept whenever possible. You should use simpler concepts/words/phrases in your paraphrasing. This means your sentence could potentially have more words than the original.

Since you will be simplifying sentences in isolation, sometimes it could be possible to delete words that do not contribute to the meaning of the sentence by itself. For example:

Original	Simplification
The **automotive** industry in California is **still small compared with** Detroit, however .	The **vehicle** industry in California is **smaller than the one in** Detroit. .

**the bold facing here is for ease of readability and will not be used in the task.*

In the previous example, "however" was deleted since it does not contribute to the meaning of the sentence presented in this way. **Your focus should be on lexical paraphrasing**, and deletion should be limited to cases where it is necessary to remove words in order to keep the sentence fluent.

We will manually check a fraction of your answers and spammers will be rejected. Read the provided instructions carefully and make sure you understand the task before beginning.

Have fun!

We ask you to rewrite each original sentence in order to make it easier to understand by non-native speakers of English. You can do so by mainly **replacing complex words with simpler synonyms** (i.e. lexical paraphrasing). You are allowed to delete a few words, if needed, to keep the sentence fluent after performing the lexical paraphrasing. The final simplified sentences need to be grammatical, fluent, and retain the main ideas of their original counterparts without altering their meanings. More detailed instructions and examples can be found by clicking the "View Instructions" button.

Carefully read the instructions provided and then **simplify** the following sentences:

Sentence 1 of 2

Original: When the brain does not have enough information, it interprets the image and makes guesses.

Type the Simplification here...

Sentence 2 of 2

Original: The researchers found that the amount of friction depended strongly on whether the tip was dragged along the direction in which the pairs were oriented, or against it.

Type the Simplification here...

Figure 2: Annotation guidelines for our NEWSELA-TURK corpus along with example questions.

C System Outputs

	System Outputs
Complex	*This year, the FAA has approved dozens of permits for agricultural drone businesses.*
Simple	*This year, it approved dozens of permits for agricultural drone businesses.*
Hybrid-NG	*this year, the* ***government*** *has approved dozens of drone permits for agricultural businesses.*
LSTM	*this year, the faa has approved dozens of permits for agricultural drone businesses.*
Transformer$_{bert}$	*this year, the faa has approved dozens of permits for agricultural businesses.*
EditNTS	*this year, the* ***government*** *has approved dozens of permits for drone businesses* ***for no permission****.*
Our Model (*cp* = 0.6)	*this year, the faa* ***has allowed many businesses to use drones****.*
Our Model (*cp* = 0.7, 0.8)	*this year, the faa has approved dozens of permits for* ***drones****.*
Complex	*The room echoed with the sounds of song, the beat of drums, the voices of young men.*
Simple	*As she spoke, the building echoed with music and the beat of drums.*
Hybrid-NG	*echoed the room.*
LSTM	*the room echoed with the sounds of song, the voices of young men.*
Transformer$_{bert}$	*the room echoed with the sound of song, the beat of drums, the voices of young men.*
EditNTS	*the room echoed with the sounds of song, the beat of drums, the voices of young men* ***who are hungry and legs****.*
Our Model (*cp* = 0.6)	***the sound of the room was full of sounds*** *of young men and the voices of* ***cellos****.*
Our Model (*cp* = 0.7)	***the sound of the room sounded like a lot of music,*** *and the voices of young men.*
Our Model (*cp* = 0.8)	***the sound of the room sounded*** *like a song, the beat of drums, and the voices of young men.*

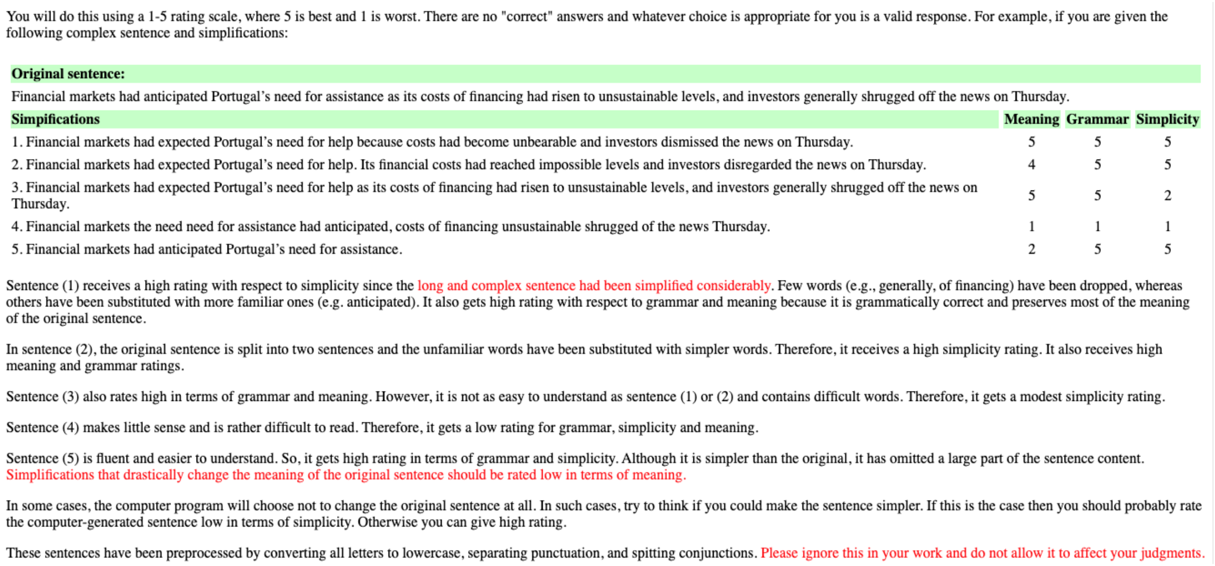
Table 10: Examples of system outputs by our paraphrase generation model and other baselines. Our model generates paraphrase-focused simplifications by skipping the splitting and deletion steps and running only the neural paraphrase generation component. (**red** marks the errors; **blue** marks good paraphrases). *cp* is a soft constraint that denotes the extent of paraphrasing in terms of number of words that can be copied from the input.

D Additional Evaluation on Newsela

Models	**SARI**	**add**	**keep**	**del**	**FK**	**SLen**	**OLen**	**CR**	**%split**	**s-BL**	**%new**	**%eq**
Complex (input)	20.6	0.0	61.7	0.0	9.2	16.9	17.0	1.0	0.0	100.0	0.0	100.0
Simple (reference)	94.6	93.6	91.4	98.8	8.7	17.9	17.9	1.06	0.0	48.0	29.7	0.0
Hybrid-NG	35.0	2.3	52.7	50.1	7.8	16.1	**17.0**	**1.0**	5.6	61.7	9.3	9.1
Transformer$_{bert}$	35.3	3.4	52.9	49.6	7.0	13.5	15.2	0.91	10.4	60.2	14.4	15.7
EditNTS	37.7	2.0	56.4	54.5	7.6	14.2	15.5	0.93	8.7	69.0	7.1	3.5
Our Model	37.9	**4.4**	51.3	58.0	6.7	13.6	15.3	0.92	9.7	**49.2**	19.2	**0.9**
Our Model (no split; *cp* = 0.6)	38.3	3.9	53.8	**57.3**	7.9	16.1	16.7	**1.0**	**0.0**	53.4	**20.8**	3.6
Our Model (no split; *cp* = 0.7)	**39.1**	3.7	58.5	55.2	8.3	16.2	16.8	**1.0**	**0.0**	67.6	12.4	11.0
Our Model (no split; *cp* = 0.8)	38.0	3.3	**60.3**	50.4	**8.5**	**16.4**	16.9	**1.0**	**0.0**	76.5	8.2	20.3

Table 11: Automatic evaluation results on a subset of Newsela test set that focuses on paraphrasing (8371 complex-simple sentence with compression ratio $>$ 0.9 and no splits). We control the extent of paraphrasing of our models by specifying the percentage of words to be copied (*cp*) from the input as a soft constraint.

E Human Evaluation Interface



You will do this using a 1-5 rating scale, where 5 is best and 1 is worst. There are no "correct" answers and whatever choice is appropriate for you is a valid response. For example, if you are given the following complex sentence and simplifications:

Original sentence:

Financial markets had anticipated Portugal's need for assistance as its costs of financing had risen to unsustainable levels, and investors generally shrugged off the news on Thursday.

Simpifications	**Meaning**	**Grammar**	**Simplicity**
1. Financial markets had expected Portugal's need for help because costs had become unbearable and investors dismissed the news on Thursday.	5	5	5
2. Financial markets had expected Portugal's need for help. Its financial costs had reached impossible levels and investors disregarded the news on Thursday.	4	5	5
3. Financial markets had expected Portugal's need for help as its costs of financing had risen to unsustainable levels, and investors generally shrugged off the news on Thursday.	5	5	2
4. Financial markets the need need for assistance had anticipated, costs of financing unsustainable shrugged of the news Thursday.	1	1	1
5. Financial markets had anticipated Portugal's need for assistance.	2	5	5

Sentence (1) receives a high rating with respect to simplicity since the long and complex sentence had been simplified considerably. Few words (e.g., generally, of financing) have been dropped, whereas others have been substituted with more familiar ones (e.g. anticipated). It also gets high rating with respect to grammar and meaning because it is grammatically correct and preserves most of the meaning of the original sentence.

In sentence (2), the original sentence is split into two sentences and the unfamiliar words have been substituted with simpler words. Therefore, it receives a high simplicity rating. It also receives high meaning and grammar ratings.

Sentence (3) also rates high in terms of grammar and meaning. However, it is not as easy to understand as sentence (1) or (2) and contains difficult words. Therefore, it gets a modest simplicity rating.

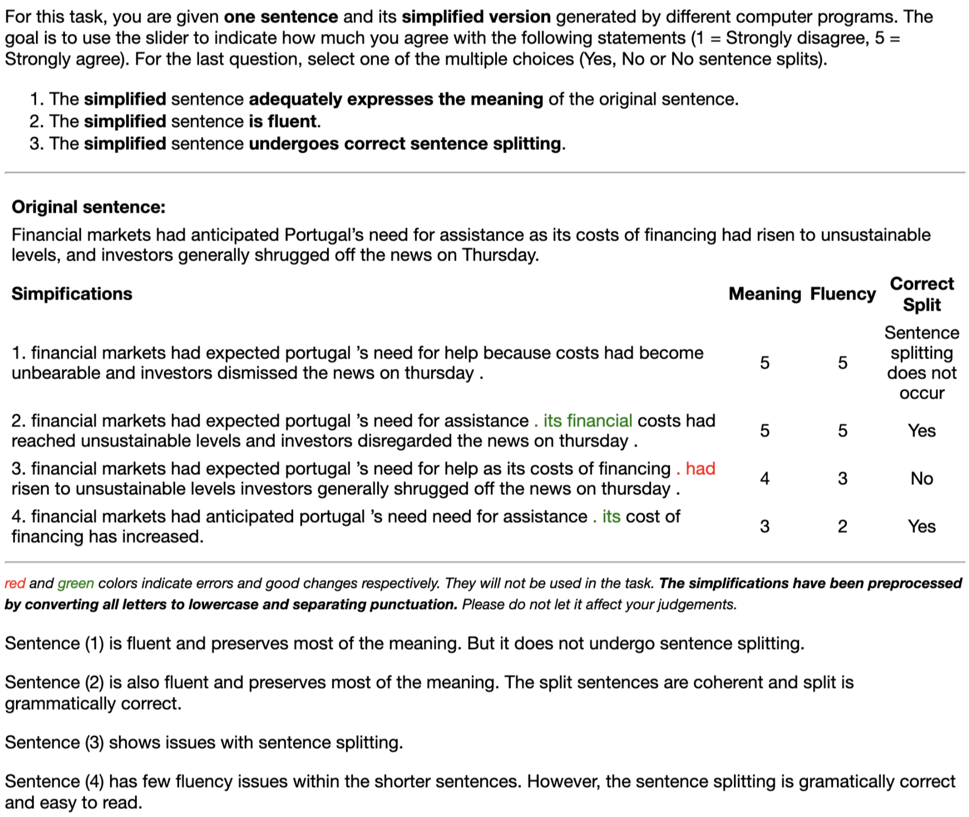
Sentence (4) makes little sense and is rather difficult to read. Therefore, it gets a low rating for grammar, simplicity and meaning.

Sentence (5) is fluent and easier to understand. So, it gets high rating in terms of grammar and simplicity. Although it is simpler than the original, it has omitted a large part of the sentence content. Simplifications that drastically change the meaning of the original sentence should be rated low in terms of meaning.

In some cases, the computer program will choose not to change the original sentence at all. In such cases, try to think if you could make the sentence simpler. If this is the case then you should probably rate the computer-generated sentence low in terms of simplicity. Otherwise you can give high rating.

These sentences have been preprocessed by converting all letters to lowercase, separating punctuation, and spitting conjunctions. Please ignore this in your work and do not allow it to affect your judgments.

Figure 3: Guidelines provided to the Amazon Mechanical Turk workers for evaluating simplified sentences. Our interface is based on the one proposed by Kriz et al. (2019).



For this task, you are given **one sentence** and its **simplified version** generated by different computer programs. The goal is to use the slider to indicate how much you agree with the following statements (1 = Strongly disagree, 5 = Strongly agree). For the last question, select one of the multiple choices (Yes, No or No sentence splits).

1. The **simplified** sentence **adequately expresses the meaning** of the original sentence.
2. The **simplified** sentence **is fluent**.
3. The **simplified** sentence **undergoes correct sentence splitting**.

Original sentence:

Financial markets had anticipated Portugal's need for assistance as its costs of financing had risen to unsustainable levels, and investors generally shrugged off the news on Thursday.

Simpifications	**Meaning**	**Fluency**	**Correct Split**
1. financial markets had expected portugal 's need for help because costs had become unbearable and investors dismissed the news on thursday .	5	5	Sentence splitting does not occur
2. financial markets had expected portugal 's need for assistance . its financial costs had reached unsustainable levels and investors disregarded the news on thursday .	5	5	Yes
3. financial markets had expected portugal 's need for help as its costs of financing . had risen to unsustainable levels investors generally shrugged off the news on thursday .	4	3	No
4. financial markets had anticipated portugal 's need need for assistance . its cost of financing has increased.	3	2	Yes

red and green colors indicate errors and good changes respectively. They will not be used in the task. ***The simplifications have been preprocessed by converting all letters to lowercase and separating punctuation.*** *Please do not let it affect your judgements.*

Sentence (1) is fluent and preserves most of the meaning. But it does not undergo sentence splitting.

Sentence (2) is also fluent and preserves most of the meaning. The split sentences are coherent and split is grammatically correct.

Sentence (3) shows issues with sentence splitting.

Sentence (4) has few fluency issues within the shorter sentences. However, the sentence splitting is gramatically correct and easy to read.

Figure 4: Guidelines provided to the Amazon Mechanical Turk workers for evaluating simplified sentences specifically for sentence splitting.

F Evaluation on Wikipedia

Models	SARI	add	keep	del	FK	SLen	OLen	CR	%split	s-BL	%new	%eq
Complex (input)	25.9	0.0	77.8	0.0	13.4	22.4	22.6	1.0	0.0	100.0	0.0	100.0
Simple (reference)	42.0	20.6	59.9	45.5	10.9	19.1	19.3	0.88	1.1	55.2	15.3	7.8
Hybrid-NG	25.4	0.1	42.7	33.5	9.0	13.3	13.4	0.6	**0.8**	38.2	1.4	3.1
LSTM	32.6	2.1	59.8	36.0	10.0	17.8	17.8	0.84	**0.8**	60.0	10.7	15.0
Transformer$_{bert}$	35.1	4.3	**61.8**	39.2	10.4	16.7	18.8	0.85	10.9	62.1	11.1	11.1
EditNTS	36.1	2.5	67.4	38.5	11.7	20.9	22.4	1.02	6.4	63.5	**13.5**	0.0
Our Model	35.9	4.7	63.6	39.6	9.2	14.7	**19.8**	0.9	33.7	63.2	12.9	9.2
Our Model (no split; cp = 0.6)	36.5	**4.9**	63.2	**41.4**	**10.8**	**18.6**	19.9	**0.89**	6.7	61.9	12.4	3.9
Our Model (no split; cp = 0.7)	**37.5**	4.3	68.8	39.4	11.2	19.1	20.9	0.94	8.9	72.6	8.6	12.3
Our Model (no split; cp = 0.8)	37.0	3.8	72.0	35.3	11.7	19.8	21.7	0.97	8.4	80.4	6.6	24.5

Table 12: Automatic evaluation results on TURK dataset (Xu et al., 2015) that focuses on lexical paraphrasing.

Models	SARI	add	keep	del	FK	SLen	OLen	CR	%split	s-BL	%new	%eq
Complex (input)	20.5	0.0	61.5	0.0	13.4	22.4	22.6	1.0	0.8	100.0	0.0	100.0
Simple (reference)	46.3	20.0	51.0	67.9	9.1	14.8	18.9	0.87	24.2	46.2	20.5	0.6
Hybrid-NG	29.8	0.1	37.0	52.2	9.0	13.3	13.4	0.6	0.8	**38.2**	1.4	3.1
LSTM	36.1	2.4	**51.8**	54.2	10.0	17.8	17.8	0.84	0.8	59.9	10.8	14.8
Transformer$_{bert}$	38.7	5.0	53.5	57.7	10.4	16.7	**18.8**	**0.85**	10.9	62.1	11.2	11.1
EditNTS	37.8	2.7	56.0	54.9	11.7	20.9	22.4	1.02	6.4	63.6	13.4	**0.0**
Our Model	**39.7**	**5.3**	55.1	**58.8**	**9.2**	**14.7**	19.8	0.9	**33.7**	63.1	**14.0**	8.9

Table 13: Automatic evaluation results on ASSET (Alva-Manchego et al., 2020) dataset that contains all the three simplification operations.

We use the complex-simple sentence pairs from WIKI-AUTO (Jiang et al., 2020), which contains 138,095 article pairs and 604k non-identical aligned and partially-aligned sentence pairs. To capture sentence splitting, we join the sentences in the simple article mapped to the same sentence in the complex article. Similar to Newsela, we remove the sentence pairs with high (>0.9) and low (<0.1) BLEU (Papineni et al., 2002) scores. For validation and testing purposes, we use the following two corpora: (i) TURK corpus (Xu et al., 2015) for lexical paraphrasing and (ii) ASSET corpus (Alva-Manchego et al., 2020) for multiple rewrite operations. While the former corpus has 8 human-written references for 2000 validation and 359 test sentences, the latter corpus provides 10 references for the same sentences. We remove the validation and test sentences from the training corpus. Tables 12 and 13 show the results on TURK and ASSET respectively.