Keep it Simple: Unsupervised Simplification of Multi-Paragraph Text

Philippe Laban
UC Berkeley

Tobias Schnabel
Microsoft

Paul N. Bennett
Microsoft

Marti A. Hearst
UC Berkeley*

Abstract

This work presents Keep it Simple (KiS), a new approach to unsupervised text simplification which learns to balance a reward across three properties: fluency, salience and simplicity. We train the model with a novel algorithm to optimize the reward (k-SCST), in which the model proposes several candidate simplifications, computes each candidate's reward, and encourages candidates that outperform the mean reward. Finally, we propose a realistic text comprehension task as an evaluation method for text simplification. When tested on the English news domain, the KiS model outperforms strong supervised baselines by more than 4 SARI points, and can help people complete a comprehension task an average of 18% faster while retaining accuracy, when compared to the original text.

1 Introduction

The main objective of text simplification is to make a complex text accessible to a wide audience by increasing its readability. In contrast with text summarization – in which key content is selected to remain in the summary and other content is elided – in text simplification, ideally all relevant content is preserved.

We propose that text simplification algorithms need to balance three properties: (1) **fluency**: the simplified text should use well-formed English sentences, (2) **salience**: the simplified text should relay the same information as the original, and (3) **simplicity**: the simplified text should be syntactically and lexically simpler than the original.

Figure 1 provides intuition for the necessity of each of the three properties. It shows the original text and the output of the full proposed model compared to three reduced versions:

Original: NASA's Curiosity rover just celebrated a major milestone — 3,000 days on the surface of Mars. To mark the occasion, the space agency has released a stunning new panorama of the red planet, captured by the rover.

Model Full: NASA's Curiosity rover has now passed 3,000 days of travel on the surface of Mars. To mark the milestone, the space agency released a huge panorama of Mars, as seen by the rover.

Model No Fluency: NASA's Curiosity rover. celebrated. A major milestone — 3,000 days on. The of.. To mark. The space agency has. a stunning new panorama.. red planet. captured by. The rover. However

Model No Salience: NASA's Curiosity rover just celebrated a major milestone. The space agency has released a stunning new panoramic of the red planet, captured by the team. It was by the rover's panoramic camera.

Model No Simplicity: NASA's Curiosity rover has celebrated a major milestone, 3,000 days on the ground of Mars. To mark the occasion, the space agency has unveiled a stunning new panoramic view of the red planet, captured by the rover.

Figure 1: **Motivating example for the KiS method, based on a CBS article (Lewis, 2021).** We optimize a three-component reward: fluency, salience and simplicity. We show model outputs when trained with all three components, and with a missing component.

Without Fluency, the generator has no incentive to generate full sentences, and learns it can boost the simplicity score by generating short phrases with excessive punctuation.

Without Salience, the generator does not gain by covering facts in the original text, and can improve the simplicity score by learning to remove facts (e.g., not mentioning planet Mars by name).

Without Simplicity, the generator is not guided to favor syntactically and lexically simpler rewrites. In Figure 1, *Model No Simplicity* is in fact more complex than the original according to readability measures.

As we show in the related work section (Section 2), there are no high-quality, large datasets publicly released for text simplification. In this work, we build on recent progress of reinforcement learning (RL)-based training of text generators: we

* Author emails: {phillab,hearst}@berkeley.edu, {Tobias.Schnabel,Paul.N.Bennett}@microsoft.com

formulate a reference-free reward for text simplification and directly optimize it, circumventing the need for aligned data.

Our main contribution is the Keep it Simple (KiS) procedure, a novel unsupervised method for text simplification. Applied to the English news domain, KiS outperforms several supervised models on common simplification metrics such as SARI (Xu et al., 2016) and the Flesch-Kincaid Grade Level (Kincaid et al., 1975).

A second contribution is a new algorithm for RL-based training of text generators, k-SCST, which is an extension of Self-Critical Sequence Training (Rennie et al., 2017). For each input, we generate k sampled outputs (vs. 2 in SCST), and use the mean population reward as a baseline. We show in Section 4 that in our domain, k-SCST outperforms models trained with SCST.

A third contribution is a novel evaluation method for text simplification. Based on the assumption that simplified text should enable faster reading with better understanding, we propose a realistic Text Comprehension task. We show that people reading texts simplified by KiS are able to complete comprehension tasks faster than comparison texts.

Another departure from previous work is that we work with *paragraphs* as units of text. Most work in text simplification is done at the sentence level, despite work such as Zhong et al. (2020) showing that common simplification phenomena occur at the level of the paragraph, (e.g., the deletion, insertion or re-ordering of full sentences). Specifically, we train our models to simplify full paragraphs, and evaluate our models in a human evaluation on short *documents* (i.e., 3-4 paragraphs).

Through rigorous empirical evaluation, we demonstrate the strong performance of our approach; automated results show that this unsupervised approach is able to outperform strong supervised models by 4 SARI points or more. We publicly released the code and model checkpoints[1].

2 Related Work

Simplification Datasets. Early datasets were first based on Simple Wikipedia[2]: WikiSmall (Zhu et al., 2010), later expanded into WikiLarge (Zhang and Lapata, 2017). Xu et al. (2015) show there are quality concerns with Simple Wikipedia datasets, and propose Newsela[3] as a replacement. Newsela is a project led by educators re-writing news articles targeting different school grade levels. We view Newsela as the gold-standard for our work, and use the public Newsela release of 1,911 groups of articles to design and evaluate our work. Using a coarse paragraph alignment algorithm, we extract 40,000 paired simple/complex paragraphs targeting a separation of 4 grade levels. We call this dataset the *paired Newsela dataset*, which we use for analysis and baseline training.

Seq2Seq for Simplification. Text simplification is most commonly framed as a sequence-to-sequence (seq2seq) task, leveraging model architectures of other seq2seq tasks, such as natural machine translation (Zhu et al., 2010; Wubben et al., 2012). Martin et al. (2020) introduce ACCESS, a finetuned Transformer model that achieves state-of-the-art performance on WikiLarge. ACCESS can customize simplifications on parameters such as compression rate and paraphrase amount. We directly compare our approach to ACCESS.

Data availability remains one of the main limitations to seq2seq-based text simplification. We side-step this issue entirely by working with unsupervised data, only requiring a small dataset with coarse-level alignments for calibration.

Lexical Simplification focuses on the substitution of single words or phrases with simpler equivalents, with diverse approaches using lexical databases such as WordNet (Thomas and Anderson, 2012), to using contextualized word vectors (Qiang et al., 2020). These methods tend to be limited, as they do not consider syntactic complexity, and have no direct way of modeling deletions and insertions. We incorporate a lexical score (L_{Score}) as one of the rewards in our simplicity component.

Text-edit for Simplification. Recent work (Dong et al., 2019; Stahlberg and Kumar, 2020) has modeled text simplification as a *text-edit* task, learning sequences of word-edits that transform the input into the output. Text editing offers explainability, at the cost of added model complexity. We find that without explicitly representing edits, the KiS model easily learns to copy (using attention heads) and deviate from the original text. Outputs can be post-processed into edits, if desired.

Unsupervised Simplification has mostly been limited to lexical simplification. Recently Surya et al. (2019) (Unsup NTS) proposed a system that

[1] https://github.com/tingofurro/keep_it_simple

[2] https://simple.wikipedia.org/

[3] https://newsela.com/

can perform both lexical and syntactic simplification, with a joint encoder, and two decoders (simple and complex). We directly compare our unsupervised approach to Unsup NTS.

RL for Simplification. Prior work (Zhang and Lapata, 2017; Guo et al., 2018) used Reinforcement Learning (RL)-based simplification. However, in both cases, components of the reward or training procedure involved reference simplifications, requiring an aligned dataset. By designing a reference-free reward, we are able to train our model with RL without supervision.

Evaluation of Simplification. This usually falls into two categories: automatic offline evaluation, and human evaluation. Automatic evaluations usually involve using n-gram overlap calculations such as BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016)). SARI was shown to correlate better with human judgements of simplicity than BLEU, and it has since become a standard (Zhang and Lapata, 2017; Surya et al., 2019; Martin et al., 2020). In our experiments, we report both SARI and BLEU.

Human evaluation is typically done in an *intrinsic* way – e.g., by directly rating factors like fluency, simplicity and relevance of model outputs (Surya et al., 2019; Wubben et al., 2012). In this work, we propose an extrinsic, task-based protocol. In our comprehension study, we directly measure how much simplified texts can help a human reader answer questions more efficiently. The closest to our evaluation design is that of Angrosh et al. (2014) with the important difference that we require participants to resubmit after erroneous answers. In pilot studies, we found this step to be crucial for high-quality responses.

3 KiS Components

In KiS, we approach unsupervised simplification as a (non-differentiable) reward maximization problem. As shown in Figure 2, there are four components to the reward: simplicity, fluency, salience and guardrails which are jointly optimized. This is essential to avoid trivial solutions that only consider subsets. We therefore use the product of all components as the total reward, because the product is sensitive to the sharp decrease of a single component. For example, the triggering of a single guardrail leads to the zeroing of the total reward. Each component is normalized to the $[0, 1]$ range.

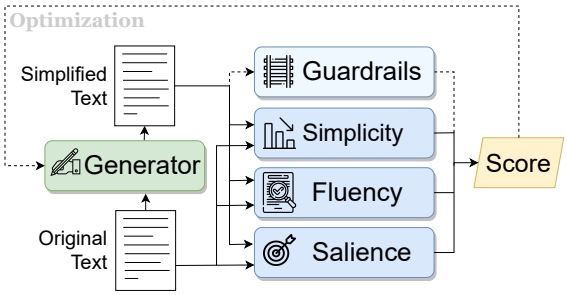

Figure 2: **Keep it Simple is an unsupervised training procedure for text simplification.** The text generator (GPT-2) produces candidate simplifications, scored according to *fluency*, *simplicity*, *salience*. *Guardrails* enforce the model does not learn high-scoring shortcuts.

```
def S_Score(original,simple):
    Fstart = fkgl(original)
    tgt = target_delta(Fstart)
    Fend   = fkgl(simple)
    D = Fend-Fstart
    return clip(1-((D-tgt)/tgt),0,1)
def target_delta(Fstart):
    # Line-fitted from analysis
    if Fstart < 4.0:
        return 0.1
    if Fstart < 12:
        return 0.5*Fstart-1.9
    return 0.8*Fstart-5.6
```

Figure 3: S_{Score} **algorithm.** `fkgl` computes the Flesch-Kincaid grade level.

3.1 Simplicity

The simplicity score should establish whether the generator's output uses simpler language than the original text. We follow prior work (Ferrés et al., 2016) and organize our score into a syntactic score S_{Score}, and a lexical score L_{Score}. Syntactic simplification focuses on reducing the complexity of a sentence, for example by reducing the number of words in a clause, or reducing distant dependencies. In lexical simplification, the objective is to replace complex phrases with simpler synonyms. To produce a single simplicity score, we take the product of S_{Score} and L_{Score} (both in $[0, 1]$).

3.1.1 Syntactic Simplicity: S_{Score}

We measure syntactic complexity via the Flesch-Kincaid grade level (FKGL) as it is easy to compute and maps to a grade-level which also corresponds to the scale used by Newsela. Other readability metrics such as Dale-Chall formula (Dale and Chall, 1948), or the Gunning-Fog index (Gunning, 1969) could be used, and future work could examine the effect of choosing one readability metric over the

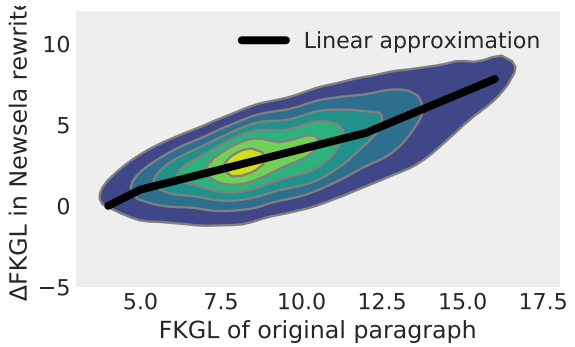

Figure 4: **Analysis (Kernel Density Estimate plot) of change in Flesch-Kincaid Grade Level in the paired Newsela dataset.** Most simple paragraphs have lower FKGL than the original paragraphs (positive $\Delta FKGL$). When the original paragraph's FKGL is higher (x-axis), the change in FKGL tends to be larger (y-axis). We fit a linear approximation, which we use to compute the $Sscore$.

other. Another viable option is the Lexile score (Smith et al., 2016), however, because its implementation is not publicly released, we cannot use it during training and we report it only for evaluation (done manually on the Lexile Hub[4]).

Figure 3 shows the S_{Score} algorithm. We compute the original paragraph's FKGL (`FStart`), used to compute a target FKGL (`tgt`). The score is a linear ramp measuring how close the achieved FKGL (`Fend`) is to the target, clipped to $[0, 1]$.

In the initial design, the target drop was a constant: 4 grade levels, independent of `FStart`. However, analysis on the paired Newsela corpus revealed that the target FKGL should depend on the initial FKGL. This makes sense intuitively: an already syntactically simple paragraph should not require further simplification, while more complex paragraphs require more simplification. Figure 4 shows the positive correlation between the original paragraph's FKGL and the drop of FKGL in the simplified text. We fit a piece-wise linear function to calculate the target FKGL drop from the initial paragraph.

3.1.2 Lexical Simplicity: L_{Score}

Lexical simplicity focuses on whether words in the input paragraph (W_1) are more complex than ones in the output paragraph (W_2). We rely on the observation that word frequency and difficulty are correlated (Breland, 1996), and use word frequency in a large corpus of text (Brysbaert and New, 2009) to determine simplicity.

[4]https://hub.lexile.com

Because word frequency follows a Zipf power law, we use Speer et al. (2018)'s log normalization, adjusting the frequency on a $[0, 8]$ range, with words at 0 being non-existent in the corpus, and 8 for most common words. As an example, the word *vigorous* has a frequency of 3.54, while its more common synonym *strong* obtains 5.23.

We compute the average Zipf frequency of the set of inserted words ($Z(W_2 - W_1)$), and the set of deleted words ($Z(W_1 - W_2)$). The difference

$$\Delta Z(W_1, W_2) = Z(W_2 - W_1) - Z(W_1 - W_2) \quad (1)$$

should be positive. Analysis of the *paired Newsela corpus* reveals that 91% of pairs have a positive $\Delta Z(W_1, W_2)$, with a median value of 0.4. We use this median as the target Zipf shift in the L_{Score}, and use a ramp shape similar to the S_{Score}, clipped between 0 and 1 (denoted as $[\cdot]^+$):

$$L_{Score}(W_1, W_2) = \left[1 - \frac{|\Delta Z(W_1,W_2) - 0.4|}{0.4}\right]^+ \quad (2)$$

3.2 Fluency

We use two sub-components for the fluency component: a pre-trained language-model, and a discriminator trained dynamically with the generator.

3.2.1 Language-Model Fluency

Language models assign a probability to a sequence of words. This probability is often used to measure fluency of generated text (Kann et al., 2018; Salazar et al., 2020). The KiS fluency score is based on a language model in a way similar way to Laban et al. (2020). The language model is used to obtain a likelihood of the original paragraph ($LM(p)$) and of the generated output $LM(q)$. We use average log-likelihood, for numerical stability. The language model fluency score is then:

$$LM_{Score}(p, q) = \left[1 - \frac{LM(p) - LM(q)}{\lambda}\right]^+ \quad (3)$$

λ is a tunable hyper-parameter. If the $LM(q)$ is lower than $LM(p)$ by λ or more, $LM_{Score}(p, q) = 0$. If $LM(q)$ is above or equal to $LM(p)$, then $LM_{Score}(p, q) = 1$, and otherwise, it is a linear interpolation.

We set $\lambda = 1.3$ as it is the value for which the *paired Newsela dataset* achieves an average LM_{Score} of 0.9.

3.2.2 Discriminator Fluency

The LM_{Score} is static and deterministic, which can be limiting, as the generator can learn during training how to adapt and exploit flaws in the language-model (e.g., learning to alter capitalization).

Inspired from the Generative Adversarial Network (GAN) framework (Goodfellow et al., 2014), we create a dynamic discriminator, trained in conjunction with the generator, dynamically adapting the fluency score during training.

Specifically, we use a RoBERTa model (Liu et al., 2019) as the basis for the discriminator, a classifier with two labels: 1 for authentic paragraphs, and 0 for generator outputs.

As the generator produces outputs, they are assigned a label of 0 and added to a *training buffer*, while the original paragraphs are assigned a label of 1 and added to the training buffer as well.

Once the training buffer reaches a size of 2,000 samples, the discriminator is trained, using 90% of the training buffer. We train the discriminator for 5 epochs (details of training are in Appendix A.1). At the end of each epoch, we checkpoint the discriminator model. We compare the 5 checkpoints in terms of F-1 performance on the remaining 10% of the training buffer, and keep the best checkpoint as the new discriminator.

The discriminator's probability that a paragraph (q) is authentic is the discriminator score:

$$D_{Score}(q) = p_{disc}(Y = 1|X = q) \qquad (4)$$

As with GANs, there is an equilibrium between the generator attempting to maximize the probability of generating real outputs ("fooling" the discriminator), and the discriminator succeeding at distinguishing generated and authentic texts.

3.3 Salience

For the salience component, we use the *coverage model* introduced in the summary loop (Laban et al., 2020) for the domain of text summarization, and adapt it to the simplification domain.

The coverage model is a Transformer-based model trained to look at generated text and answer fill-in-the-blank questions about the original text. The score is based on model accuracy at filling in the blanks: the more is filled in, the more relevant the generated content is, and the higher the score.

A key element of the coverage model is its masking procedure, which decides which words to mask. In the summary loop, a limited number of extracted keywords (up to 15 words) are masked. By contrast, for simplification, we mask all non-stop words, amounting to a masking rate of about 40%.

This change reflects a difference in expectation between summarization and simplification: in summarization, only key components are expected to be recovered from a summary, whereas in simplification most of the original paragraph should be recoverable. Coverage ranges in $[0, 1]$, and reference simplifications in the *paired Newsela corpus* obtain an average score of 0.76, confirming that manual simplification can achieve high coverage.

3.4 Guardrails

We use *guardrails* as simple pattern-based scores to avoid common pathological generation problems that we observed. Unlike the main components, guardrails are binary, giving a score of 1 (pass) unless they trigger (score of 0). We use two guardrails: brevity and inaccuracy.

3.4.1 Brevity guardrail

The brevity guardrail ensures the length of generated paragraph (L_2) falls in a range around the original paragraph's length (L_1). We compute a compression ratio: $C = L_2/L_1$. If $C_{min} \leq C \leq C_{max}$, the guardrail passes, otherwise it triggers.

We set $[C_{min}, C_{max}] = [0.6, 1.5]$, because these values ensure the guardrail is not triggered on 98% of the paired Newsela dataset; this can be adapted depending on the application.

3.4.2 Inaccuracy guardrail

Modern text generation models are known to *hallucinate* facts (Huang et al., 2020), which has led the community to create models to detect and correct hallucinations (Cao et al., 2020; Zhang et al., 2020; Wang et al., 2020).

We propose a light-weight inaccuracy detector as a guardrail. We use a Named Entity Recognition (NER) model (Honnibal et al., 2020) to extract entities present in the original paragraph (E_1) and the model's output (E_2). We trigger the guardrail if an entity present in E_2 is not in E_1.

Even though human writers can successfully introduce new entities without creating inaccuracies (e.g., replacing the city *La Paz* with the country *Bolivia*), we find that text generators predominantly introduce inaccuracies with novel entities. This simple heuristic can eventually be replaced once inaccuracy detection technology matures.

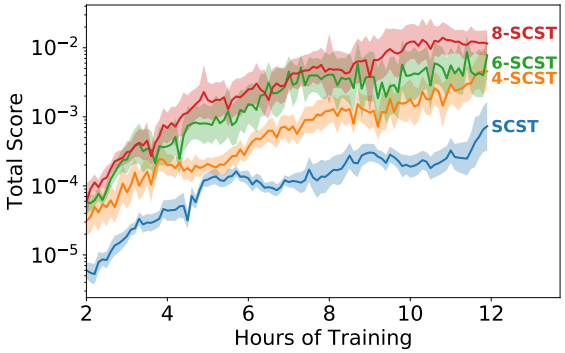

Figure 5: **Training KiS models comparing SCST with k-SCST.** We try 4, 6 and 8 as values for k. Increasing k improves performance and stability.

4 KiS Training

Rennie et al. (2017) introduced Self-Critical Sequence Training (SCST) as an effective algorithm for reward-based training of text generators, successfully applying it to image captioning. The efficacy of SCST was later confirmed on other text generation tasks such as question generation (Zhang and Bansal, 2019), and summarization (Celikyilmaz et al., 2018; Laban et al., 2020). In SCST, a probabilistic model is used to generate two distinct candidates: C^S, a candidate constructed by sampling the word distribution at each step, and $\hat{C}$, by taking the argmax of the word distribution at each step. Each candidate is scored, obtaining rewards of R^S and $\hat{R}$, respectively, and the loss is:

$$L = (\hat{R} - R^S) \sum_{i=0}^{N} \log p(w_i^S | w_1^S ... w_{i-1}^S, P) \quad (5)$$

where $p(w_i^S|...)$ represents the probability of the i-th word conditioned on previously generated sampled sequence according to the model, P is the input paragraph, and N the number of words in the generated sequence. Intuitively, minimizing this loss increases the likelihood of the sampled sequence if $R^S > \hat{R}$, and decreases it otherwise, both increasing the expected total reward.

One limitation in SCST occurs when the two sequences achieve comparable rewards ($R^S \simeq \hat{R}$): the loss nears zero, and the model has little to learn, wasting a training sample. In our experiments with SCST, this can occur with 30% of samples.

We propose an extension of SCST, which we call k-SCST. We generate k sampled candidates ($k > 2$), compute the rewards of each candidate $R^{S1}, ..., R^{Sk}$, as well as the mean reward achieved by this sampled population: $\bar{R^S} = (R^{S1} + ... + R^{Sk})/k$, which we use as the baseline, instead of $\hat{R}$. The loss L becomes:

$$L = \sum_{j=1}^{k} (\bar{R^S} - R^{Sj}) \sum_{i=0}^{N} \log p(w_i^{Sj} | w_1^{Sj} ... w_{i-1}^{Sj}, P) \quad (6)$$

We use a GPT2-medium for the generator, initialized with the released pre-trained checkpoint. Experimental details such as data and optimizer used are provided in Appendix A.1.

In Figure 5, we show results of a direct comparison of SCST ($k = 2$) with k-SCST varying k in $\{4, 6, 8\}$, while keeping other components of the training fixed. Because of the variance involved in RL training, we recorded six independent training runs for each setting (for a total of 24 runs), and plot the average reward across runs of a setting, as well as the standard error of the mean (SEM).

We observe that increasing k leads to higher average reward, and less variation in the reward. In our setting, k-SCST boosts performance and stabilizes training. We use $k = 8$ in all final models, as increasing k further is impractical due to GPU memory limitations.

We believe k-SCST's advantage stems from two factors: first, obtaining a better estimate of the distribution of rewards by sampling more outputs, second, by using the mean reward as the baseline, saving on computation of a separate baseline generation. We believe k-SCST can also improve learning in other text generation applications and plan to pursue this in future work.

5 Experiments

We present results experimentally validating the KiS procedure for text simplification. We give results based on automatic metrics, on a novel human comprehension task, and from an ablation study.

5.1 Models Compared

We compare the **KiS Model** to three strong supervised models, and an unsupervised approach.

ACCESS from (Martin et al., 2020), is a state-of-the-art Transformer model trained on WikiLarge (300,000 pairs of complex/simple sentences). This model uses default parameters ($NBChar$=0.95, $LevSim$=0.75).

ACCESS90 is identical to **ACCESS**, with different parameters ($NBChar$=0.90, $LevSim$=0.75), reducing target compression from 95% to 90%, matching the average compression rate in Newsela.

Model	SARI	BLEU	%FKGL	%Lexile	Comp.	Cov.
Newsela	-	-	87	**79**	.918	.754
Finetune Baseline	.470	**.719**	68	52	.903	**.894**
ACCESS Default	.666	.649	86	63	.958	.805
ACCESS 90	.674	.644	93	64	.921	.789
Unsup NTS	.677	.535	48	57	.753	.618
KiS Model	**.709**	.526	**100**	72	.852	.640

Table 1: **Automatic results on Newsela test-set.** *SARI* and *BLEU* are reference-based metrics. *%FKGL* and *%Lexile* are percentages of model outputs lowering the grade level. *Comp.* is the average compression ratio (# words), and *Cov.* the output's average coverage score.

Finetune Baseline is a GPT2-medium model finetuned on the *paired Newsela dataset*. Large pre-trained models often perform competitively in low-resource environments, making this a strong point of comparison.

Unsup NTS from (Surya et al., 2019) is an unsupervised approach based on successively encoding and denoising text using a GRU architecture.

Training details for the KiS Model and Finetune Baseline are in Appendix A.1.

5.2 Automatic Results

We put aside 500 samples from the *paired Newsela dataset* as a test set to compare models on automatic metrics. We compare models on SARI and BLEU, report the percentage when readability measures see an improvement in readability: %FKGL, and %Lexile and compute the average compression rate (Comp.), and coverage (Cov.). Results are summarized in Table 1.

The KiS model achieves the highest SARI score by a margin of 0.04, even though it is an unsupervised approach.

Finetune Baseline achieves the highest BLEU and salience scores, but lowest SARI score. We interpret this as showing the model takes the least risk: high salience, with little simplification.

We observe that all models are able to increase readability in terms of FKGL and Lexile compared to original paragraphs. We note that for almost all models, the percentage is lower for the Lexile measure than for FKGL, showing that an improvement in Lexile score is more difficult to achieve than FKGL. The KiS model achieves an increase in Lexile readability 72% of the time, the closest figure to 79% of the Newsela human-written reference.

We note that the perfect performance of KiS on %FKGL could be explained by the fact that FKGL is a part of a component being optimized (S_{Score}), however Lexile was not.

In terms of compression, the KiS model compresses the second most, most likely hurting its coverage. Adjusting the Brevity guardrail could encourage the model to compress less. ACCESS90 has the compression rate closest to Newsela references, but this only leads to a modest improvement in SARI when compared to ACCESS.

Overall, the Newsela references achieve the best percentage of Lexile readability improvement, while outperforming the KiS model at coverage: there is still a gap between human-written simplifications and model-generated ones.

5.3 Human Comprehension Study

We propose a human comprehension study to evaluate the usefulness of simplification results. Simplified text should be easier to read than the original text, while retaining accuracy and understanding. We design a task to evaluate how well both manual and automated simplifications achieve this objective. The main idea is to show readers a text and ask them to answer multiple-choice questions, evaluating the texts based on time and retries needed to select the correct answer.

5.3.1 Study Design

Five different versions of each document were generated as stimuli: the original document, the Newsela reference, and versions from the three best-performing methods from the last section: KiS, Finetune Baseline, and ACCESS. We did not include Unsup NTS in our analysis, because of its low performance on %FKGL and %Lexile metrics. Associated with each document are five manually generated multiple-choice questions, each with one or more correct answers and one to four distractors. The original and the Newsela texts were checked manually by experimenters to ensure that all allow for questions to be answered correctly. Crowdworkers were shown four documents in succession, in a between-participants design. Order of document and stimuli type were randomized. Figure 6 shows two stimuli of a document (original and KiS) along with the comprehension questions. (The entire set of five stimuli can be found in Figure A2 in the Appendix.)

After several rounds of pilot testing, we arrived at the following design choices:

Document theme. We chose recent news articles involving complex themes (e.g., trajectory of iceberg) as the source of documents. For news articles, recency seems to engage participants, and

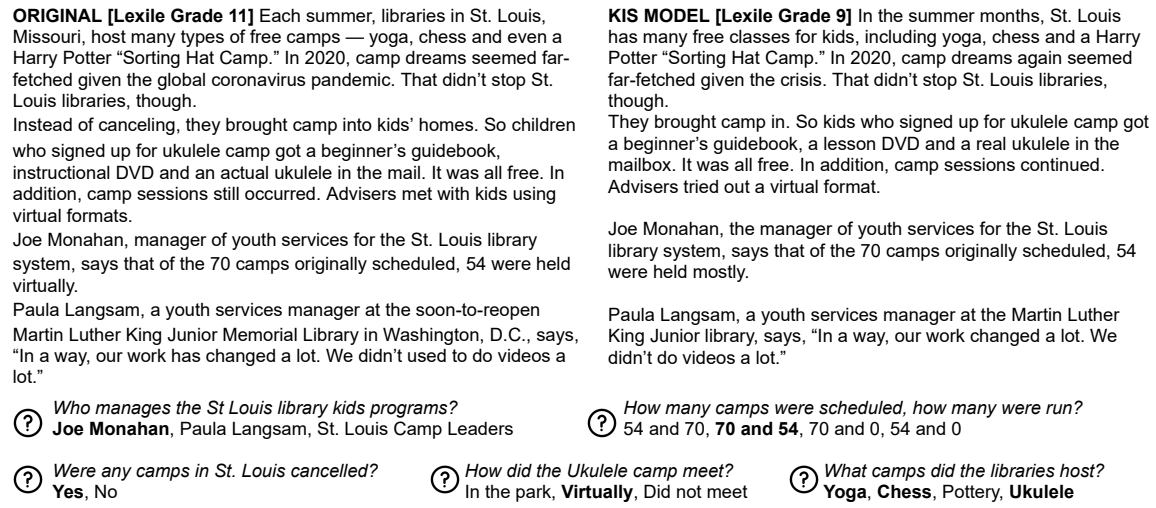

Figure 6: **Example Task (from a Washington Post article (Kelati, 2020)) for the Comprehension Study.** Shown are two of five stimuli: original document (left), and KiS model output (right). Participants read a text and answered comprehension questions (bottom). Average completion time was 160 seconds (original) and 136 seconds (KiS model output).

technical terms increase the impact of simplification.

Section length. We chose document length of 3-4 paragraphs (or 200 words), and five comprehension questions. Document length should not be too short (makes some questions trivial), or too long (adds a retrieval component to the task).

Selection of questions. Questions were generated via a GPT2 question generation model fine-tuned on the NewsQA dataset (Trischler et al., 2017). We select questions answerable by both the original and Newsela references, attempting to have both factoid (answer is entity) and reasoning questions.

Re-submission until correct. When submitting answers, participants received feedback on which were incorrect, and were required to re-submit until all answers were correct. This aligns the objective of the participant (i.e., finishing the task rapidly), with the task's objective (i.e., measuring participant's efficiency at understanding). This also gives a way to discourage participants from "brute-forcing" the task, re-submitting many combinations until one works.

We note that some components of the study such as the choice of document themes and the selection of comprehension questions are elements that create variability in the results. We release the models used in the study, as well all generated texts that were evaluated to enable follow-up research and to aid reproducibility.

Model	Time (sec)	# Subs.	Comp.	CASpeed
♭ Original	174.0	4.23	1.0	1.00
♮ Newsela	163.3	5.10	1.08	**1.15**
∀ ACCESS	188.5	6.69	0.96	0.88
∃ Finetune Baseline	161.0 ∀	4.70	0.97	1.04
∇ KiS Model	**142.6** ♭ ♮ ∀	4.10 ∀	0.87	**1.06**

Table 2: **Results of the Human Comprehension Study.** We measure average completion time (Time), number of submissions (#Subs.), compression ratio (Comp.) and a compression-accounted speed-up (CASpeed). Each text version is assigned a symbol used to indicate statistical significance ($p < 0.05$).

5.3.2 Study Results

We ran the study on Mechanical Turk, accepting crowd-workers with 1700+ completed tasks, and an acceptance rate of 97%+. The study was active for two weeks in December 2020, and remunerated participants completing all four sections at a rate of $10/hour. (Appendix A.2 shows crowd-worker instructions and the document/version distributions.) When removing "brute-forced" submissions (10+ re-submissions), we are left with 244 submissions, used for result analysis reported in Table 2, (A more detailed results table is included in Appendix A.4.)

We measure two outcomes: question completion time (in seconds), and number of submissions to correctness. We performed a Kruskal-Wallis test (Kruskal and Wallis, 1952) with a Dunn post-hoc test (Dunn, 1964) for statistical significance between pairs of conditions.

In line with study objectives, simplified texts

help participants complete the task faster than reading original texts, with three of the four simplified versions leading to improvements in completion times. Participants were fastest with KiS simplifications (18% faster). The KiS model led to a statistically significant speed-up compared to the originals, Newsela references, and ACCESS simplifications. ACCESS simplifications surprisingly led to a non-significant slow-down, which we attribute to a potential loss in fluency that might have confused participants.

One important factor we consider is that shorter passages (i.e., smaller compression) might lead to a speed-up regardless of simplicity. We confirm this by finding a small positive correlation between passage length and completion time of 0.09. We compute a *compression-adjusted speed-up* (*CASpeed*) ratio by: (1) computing the passage length of each simplified version, (2) linearly extrapolating the expected completion time for this passage length for original paragraphs, and (3) computing the ratio of the extrapolation to the observed completion time. If $CASpeed > 1$, participants were faster than expected for the passage length. Newsela reference paragraphs achieve the best *CASpeed*, followed by the KiS model. This suggests that good simplification can involve making texts longer.

5.4 Ablation Study

We train three ablated models, each missing a reward component to gain understanding in the value of each component of the KiS procedure.

Figure 1 gives a qualitative perspective on each ablation. Without fluency, the generator learns to generate incomplete sentences, without salience, it omits important information, and without simplicity, it can sometimes "complexify".

We computed complete automatic results for the ablated models, and find that each ablation leads to a decrease on an evaluation metric, confirming that all three components are necessary to generate high-quality simplifications (details in Appendix A.5).

6 Limitations and Future Work

Improved Accuracy Scoring. The current guardrail for inaccuracy is rudimentary; trained models still generate non-factual simplifications. Recent work in fact-checking for the summarization domain (Kryscinski et al., 2020; Li et al., 2018) could be adapted to the simplification domain to improve this.

Inclusion of Supervised Signal. In this work, we establish that text simplification can be approached in an unsupervised manner. In future work, Keep it Simple could be used as a pre-training strategy, or used jointly with supervised training.

Reproducibility of Human Evaluation. Even though we release the models, stimuli and comprehension questions used in the human evaluation, some elements of the procedure introduce randomness. Participating crowd-workers differ in literacy level which may have an effect on their performance at the task (Alonzo et al., 2021).

New Settings, Domains and Languages. We limited our experiments to the simplification of English news articles following prior work, but plan to pursue other languages in the future. Similarly, because Keep it Simple does not require labeled data, it can be applied to new settings (e.g., rewriting to inverse the effects of simplification), or to new domains (e.g., legal texts).

7 Conclusion

We have shown that text simplification can be approached in an unsupervised manner via KiS. By optimizing a reward comprised of simplicity, fluency and salience components, KiS is able to outperform strong supervised models on automatic metrics (+0.04 in SARI). We propose a human comprehension task to evaluate the usefulness of simplification and show that simplifications tend to lead to a measurable speed-up in task completion, with KiS texts producing the best speed-up of 18% on average. These are first steps for unsupervised text simplification, and we suggest that future work should focus on adapting the methodology to new domains (i.e., legal), non-English languages, and refining optimized rewards to take factuality into account.

Acknowledgments

We would like to thank Katie Stasaski, Dongyeop Kang, and the ACL reviewers for their helpful comments, as well as Newsela for providing a version of their simplified news corpus. This work was supported by a Microsoft BAIR Commons grant as well as a Microsoft Azure Sponsorship.

References

Oliver Alonzo, Jessica Trussell, Becca Dingman, and Matt Huenerfauth. 2021. Comparison of methods for evaluating complexity of simplified texts among deaf and hard-of-hearing adults at different literacy levels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Mandya Angrosh, Tadashi Nomoto, and Advaith Siddharthan. 2014. Lexico-syntactic text simplification and compression with typed dependencies. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1996–2006, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

H. Breland. 1996. Word frequency and word difficulty: A comparison of counts in four corpora. *Psychological Science*, 7:96 – 99.

M. Brysbaert and B. New. 2009. Moving beyond kucera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior Research Methods*, 41:977–990.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

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

Olive Jean Dunn. 1964. Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252.

Daniel Ferrés, Montserrat Marimon, Horacio Saggion, et al. 2016. Yats: yet another text simplifier. In *International Conference on Applications of Natural Language to Information Systems*, pages 335–342. Springer.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27.

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

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*, pages 462–476.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Doi.org/10.5281/zenodo.1212303.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469.

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*, pages 313–323.

Haben Kelati. 2020. Librarians find creative ways to serve kids when buildings are closed for browsing. *The Washington Post*.

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. Technical report, Naval Technical Training Command Millington TN Research Branch.

William H Kruskal and W Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5135–5150. Association for Computational Linguistics.

Sophie Lewis. 2021. Nasa curiosity rover celebrates 3,000th day on mars with stunning panorama of planet. *CBS News*.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Louis Martin, Éric Villemonte de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4689–4698.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8649–8656.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712.

Malbert Smith, J. Turner, Eleanor E. Sanford-Moore, and Heather H. Koons. 2016. The lexile framework for reading: An introduction to what it is and how to use it.

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. Luminosoinsight / wordfreq: v2.2. Doi.org/10.5281/zenodo.1443582.

Felix Stahlberg and Shankar Kumar. 2020. Seq2edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159.

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*, pages 2058–2068.

S Rebecca Thomas and Sven Anderson. 2012. Wordnet-based lexical simplification of a document. In *KONVENS*, pages 80–88.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020.

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

W. Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

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

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

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

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D Manning, and Curtis Langlotz. 2020. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5108–5120.

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2020. Discourse level factors for sentence deletion in text simplification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9709–9716.

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 2010)*, pages 1353–1361.

Ethical Considerations

We present a method for text simplification and verify its performance on text from the news domain in the English language. Even though we expect the method to be adaptable to other domains and languages, we have not verified this assumption experimentally and limit our claims to the English news domain.

When comparing to prior work (e.g., ACCESS model), we obtained implementations directly from the authors (through Github repositories) and produced results following the recommended setting, with an objective to present prior work as a strong comparison point.

For the human evaluation, we paid the annotators above the minimum wage, and did not collect any personal identifiable information. We selected topics to avoid sensitive or political subjects and had our protocols reviewed by the university's IRB committee (Protocol ID: 2018-07-11230). We relied on a third party (Amazon Mechanical Turk) to remunerate the crowd-workers.

A Appendices

A.1 Training Details

We detail the model architecture size, data, optimizer of the models we train in the paper. All models were trained using Pytorch and HuggingFace's Transformers library[5]. We use the Apex[6] library to enable half-precision training.

The KiS procedure was trained on a single GPU, either an Nvidia V-100 (16Gb memory) or a Quadro RTX 8000 (48 Gb memory). We ran a total of around 200 experiments, with an average run-time of one week.

Because the procedure is unsupervised, the model was trained using a large unreleased corpus of news articles, containing 7 million news articles in English.

KiS Model is initialized with a *GPT2-medium* model. We used the Adam optimizer, with a learning rate of 10^{-6}, a batch-size of 1, using k-SCST with $k = 8$.

Finetune Baseline is initialized with a *GPT2-medium* model. We train using using standard teacher forcing on the 40,000 samples in the *paired Newsela dataset*, reserving 2,000 samples for validation. We use the Adam optimizer, and use the validation set to choose a learning rate of 10^{-5}, and a batch-size of 8, and run for 3 epochs before seeing a plateau in the validation loss.

Discriminator Model is initialized with a **Roberta-base**, and retrained every time the training buffer reaches 2,000 samples. The discriminator is reset to the original *Roberta-base* each time the training buffer is full. We use a standard cross-entropy loss, the ADAM optimizer with a learning rate of 10^{-5} and a batch size of 8. Each time we retrain, we run for 5 epochs, and checkpoint one model after each epoch. The checkpoint that achieves the highest performance on a validation set becomes the new discriminator for the next round.

A.2 Human Evaluation Instructions

Figure A1 shows the instructions given to crowdworker participants for the manual evaluation.

- The entire HIT should take no more than 15 minutes:
(1) You will answer a pre-questionnaire.
(2) Read 4 short news stories and answer comprehension questions about each.
- If you believe the answer is not in the document, you can select the option "Answer not in document".
- There is no time limit for each individual document or question.
- You can leave at any point but will not complete the HIT.
- You can complete this task at most once.
- If you have a question/problem, contact us at *email*.

Figure A1: Instructions given to participants of the comprehension evaluation. Participants were recruited on Amazon Mechanical Turk (MTurk), on which jobs are named "HIT".

A.3 Full Example of Generated Texts

Figure A2 is a complement to Figure 6, with the five stimuli that were shown for the *Covid Libraries* document.

A.4 Detailed of Human Evaluation Results

Table A1 details the timing and number of participants for each combination of document and stimuli.

[5]https://github.com/huggingface/transformers
[6]https://github.com/nvidia/apex

ORIGINAL [Lexile Grade 11] Each summer, libraries in St. Louis, Missouri, host many types of free camps — yoga, chess and even a Harry Potter “Sorting Hat Camp.” In 2020, camp dreams seemed far-fetched given the global coronavirus pandemic. That didn’t stop St. Louis libraries, though.
Instead of canceling, they brought camp into kids’ homes. So children who signed up for ukulele camp got a beginner’s guidebook, instructional DVD and an actual ukulele in the mail. It was all free. In addition, camp sessions still occurred. Advisers met with kids using virtual formats.
Joe Monahan, manager of youth services for the St. Louis library system, says that of the 70 camps originally scheduled, 54 were held virtually.
Paula Langsam, a youth services manager at the soon-to-reopen Martin Luther King Junior Memorial Library in Washington, D.C., says, “In a way, our work has changed a lot. We didn’t used to do videos a lot.”

NEWSELA [Lexile Grade 7] In St. Louis, Missouri, libraries hold many types of free camps in the summer. They have yoga, chess and even a Harry Potter camp. In 2020, camp ideas seemed unlikely due to the spread of COVID-19. That did not stop St. Louis libraries, though. They did not cancel their programs. They brought camp into kids' homes.
Children who signed up for ukulele camp got a beginner's book, instructional video and a ukulele in the mail. It was all free. In addition, camp sessions still went on. Camp leaders met with kids over video calls.
Joe Monahan runs the kids programs for the St. Louis libraries. He said 70 camps were scheduled at first. Some were dropped but 54 were still held online.
Paula Langsam works at the Martin Luther King Junior Memorial Library in Washington, D.C. It reopened on September 24. She said, "Our work has changed a lot. We didn't used to do videos a lot."

KIS MODEL [Lexile Grade 9] In the summer months, St. Louis has many free classes for kids, including yoga, chess and a Harry Potter “Sorting Hat Camp.” In 2020, camp dreams again seemed far-fetched given the crisis. That didn’t stop St. Louis libraries, though. They brought camp in. So kids who signed up for ukulele camp got a beginner’s guidebook, a lesson DVD and a real ukulele in the mailbox. It was all free. In addition, camp sessions continued. Advisers tried out a virtual format.
Joe Monahan, the manager of youth services for the St. Louis library system, says that of the 70 camps originally scheduled, 54 were held mostly.
Paula Langsam, a youth services manager at the Martin Luther King Junior library, says, “In a way, our work changed a lot. We didn’t do videos a lot.”

FINETUNE BASELINE [Lexile Grade 9] Each summer, St. Louis libraries host many types of free camps — yoga, chess and even a Harry Potter "Sorting Hat Camp." In 2020, camp dreams seemed far-fetched given the global cough . However , the future looked bright for libraries in St. Louis.
Instead of canceling, they brought camp into kids' homes. This is called ukulele camp. Kids signed up for it in the mail. They got a beginner's guidebook, video and an actual ukulele to play. It was all free. Advisers met with kids using virtual formats.
Joe Monahan, manager of youth services for the St. Louis library system, says that 70 camps were originally planned. Of those, 54 were held mostly.
Paula Langsam is a youth services manager who works at the Martin Luther King Junior Memorial Library in Washington, D.C. She says, "In a way, our work has changed a lot. We didn't used to do videos a lot."

ACCESS [Lexile Grade 11] Each summer, libraries in St. Louis, Missouri, has many different types of free camps that are yoga, chess and even a Harry Potter gang Sorting Hat Camp. In 2020, camp dreams seemed far-fetched that there was the global coronavirus pandemic. That did not stop St. Louis libraries, though.
Instead of being canceled, they brought camp into children's homes. So children who signed up for ukulele camp got a guidebook. They also had an actual ukulelele in the mail. It was all free. In addition, camp meetings still happened. Advisers met with new children using virtual formats.
Joe Monahan, also known as Joe Monahan, has youth services for the St. Louis library system says that of the 70 camps first started, 54 were held.
Paula Langsam, also known as Paula Langsam, is a youth services manager at the soon-to-reopen Martin Luther King Junior Library in Washington, D. We did not use to do many videos a lot.

Who manages the St Louis library kids programs?
Joe Monahan, Paula Langsam, St. Louis Camp Leaders

How many camps were scheduled, how many were run?
54 and 70, **70 and 54**, 70 and 0, 54 and 0

Were any camps in St. Louis cancelled?
Yes, No

How did the Ukulele camp meet?
In the park, **Virtually**, Did not meet

What camps did the libraries host?
Yoga, **Chess**, Pottery, **Ukulele**

Figure A2: **Complement to Figure 6.** Example Task for the Comprehension Study. Participants were assigned to one of five settings: original, Newsela, KiS, Finetune Baseline, and ACCESS. Participants were instructed to answer the five comprehension questions.

	Simplification Model				
Document Id	**Original**	**Newsela**	**Sup. Base.**	**ACCESS**	**KiS**
Marvel Show	152 (12)	209 (11)	140 (11)	209 (14)	126 (13)
Covid Libraries	167 (14)	180 (12)	182 (10)	190 (13)	171 (12)
Sustainable Food	163 (13)	144 (10)	181 (13)	242 (13)	154 (12)
Iceberg Collision	208 (14)	116 (11)	139 (12)	104 (12)	119 (12)
Version Aggregate	174 (53)	163 (44)	161 (46)	188 (52)	143 (49)

Table A1: **Average time taken and number of participants in each of the document/stimuli combinations.** Also shown are aggregates (mean time taken and total number of participants).

Model	SARI	BLEU	%FKGL	%Lexile	Comp.	Cov.
KiS Full	0.709	0.526	100	72	0.85	0.636
KiS No Fluency	0.718	0.611	99	95	1.02	0.901
KiS No Salience	0.695	0.591	100	65	1.01	0.701
KiS No Simplicity	0.672	0.617	51	23	0.92	0.809

Table A2: **Automatic results of the three ablation models.** *SARI* and *BLEU* are reference-based metrics. *% FKGL* and *% Lexile* are the percentage of simplified paragraphs with a lower FKGL and Lexile score than the original paragraph. *Comp.* is the average compression ratio (# of words), and *Cov.* is the average coverage score of the simplifications.

A.5 Detail of Ablation Study Results

Table A2 details the metric results of the three ablated models, an extension to Table 1. An example output of each ablated model, illustrating the limitation when a score component is missing, is given in Figure 1.

One surprising element is that the model trained without fluency achieves higher scores on almost all metrics, compared to the full model. This surprising fact is due to the fact that without fluency, the model does not learn to generate full sentences (see the example in Figure 1). Instead, the model learns to concatenate high-scoring phrases together, which can boost automatic metrics artificially. In fact, the strong performance of a model generating incomplete sentences reveals a limitation of current automatic metrics, such as BLEU and SARI.