Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer

Kim Cheng Sheang, Horacio Saggion
LaSTUS, TALN, Universitat Pompeu Fabra
C/Roc Boronat, 138, 08018 Barcelona, Spain
{kimcheng.sheang,horacio.saggion}@upf.edu

Abstract

Recently, a large pre-trained language model called T5 (A Unified Text-to-Text Transfer Transformer) has achieved state-of-the-art performance in many NLP tasks. However, no study has been found using this pre-trained model on Text Simplification. Therefore in this paper, we explore the use of T5 fine-tuning on Text Simplification combining with a controllable mechanism to regulate the system outputs that can help generate adapted text for different target audiences. Our experiments show that our model achieves remarkable results with gains of between +0.69 and +1.41 over the current state-of-the-art (BART+ACCESS). We argue that using a pre-trained model such as T5, trained on several tasks with large amounts of data, can help improve Text Simplification.[1]

1 Introduction

Text Simplification (TS) can be regarded as a natural language generation task where the generated text has a reduced language complexity in both vocabulary and sentence structure while preserving its original information and meaning (Saggion, 2017). Its applications can be used as reading assessment tools for people with low-literacy skills such as children (Watanabe et al., 2009), and non-native speakers (Paetzold and Specia, 2016), or people with cognitive disabilities such as autism (Barbu et al., 2015), aphasia (Carroll et al., 1999), and dyslexia (Rello et al., 2013a; Matausch and Peböck, 2010). In addition, TS can also be used as a preprocessing step to improve the results of many NLP tasks, e.g., Parsing (Chandrasekar et al., 1996), Information Extraction (Evans, 2011; Jonnalagadda and Gonzalez, 2010), Question Generation (Bernhard et al., 2012), Text Summarization (Siddharthan et al., 2004), and Machine Translation (Štajner and Popović, 2016, 2019).

In recent years, research in TS has been mostly focused on developing models based on deep neural networks (Vu et al., 2018; Zhao et al., 2018b; Martin et al., 2020b). However, and to the best of our knowledge, very few studies of transfer learning –where a model is first pre-trained on a data-rich task and then fine-tuned on downstream tasks– have been explored in TS.

In this paper, we propose a transfer learning and controllable sentence simplification model that harnesses the power of the Unified Text-to-Text Transfer Transformer (T5) pre-trained model (Raffel et al., 2020), combining it with control tokens to provide a way to generate output that adapts to different target users. Such a model can be adjusted to fit the need of different users without having to build everything from the ground up.

We make the following contributions:

- We introduce a transfer learning approach combined with a controllable mechanism for sentence simplification task.
- We make an improvement to the performance of the sentence simplification system.
- We introduce a new control token #words to help the model generate sentences by replacing long complex words with shorter alternatives.
- We conduct an evaluation and comparison between different sizes of pre-trained models and a detailed analysis on the effect of each control token.
- We show that by choosing the right control token values and pre-trained model, the model achieves the state-of-the-art performance in two well-known benchmarking datasets.

[1]The code and data are available at https://github.com/KimChengSHEANG/TS_T5

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 341–352,
Aberdeen, Scotland, UK, 20-24 September 2021.

2 Related Work

2.1 Sentence Simplification

It is often regarded as a monolingual translation problem (Zhu et al., 2010; Coster and Kauchak, 2011; Wubben et al., 2012), where the models are trained on parallel complex-simple sentences extracted from English Wikipedia and Simple English Wikipedia (SEW) (Zhu et al., 2010).

There are many approaches based on statistical Machine Translation (SMT), including phrase-based MT (PBMT) (Štajner et al., 2015), and syntax-based MT (SBMT) (Xu et al., 2016). Nisioi et al. (2017) introduced Neural Text Simplification (NTS), a Neural-Machine-Translation-based system (NMT) which performs better than SMT. Zhang and Lapata (2017) took a similar approach adding lexical constraints combining the NMT model with reinforcement learning. After the release of Transformer (Vaswani et al., 2017), Zhao et al. (2018a) introduced a Transformer-based approach and integrated it with a paraphrase database for simplification called Simple PPDB (Pavlick and Callison-Burch, 2016a). The model outperforms all previous state-of-the-art models in sentence simplification.

Our proposed model is also a sequence-to-sequence Transformer-based model, but instead of using the original Transformer by Vaswani et al. (2017), we use T5 (Raffel et al., 2020).

2.2 Controllable Sentence Simplification

In recent years, there has been increased interest in conditional training with sequence-to-sequence models. It has been applied to some NLP tasks such as controlling the length and content of summaries (Kikuchi et al., 2016; Fan et al., 2017), politeness in machine translation (Sennrich et al., 2016), and linguistic style in text generation (Ficler and Goldberg, 2017). Scarton and Specia (2018) introduced the controllable TS model by embedding grade level token <grade> into the sequence-to-sequence model. Martin et al. (2020b) took a similar approach adding 4 tokens into source sentences to control different aspects of the output such as length, paraphrasing, lexical complexity, and syntactic complexity. Kariuk and Karamshuk (2020) took the idea of using control tokens from Martin et al. (2020b) and used it in unsupervised approach by integrating those control tokens into the back translation algorithm, which allows the model to self-supervise the process of learning inter-relations between a control sequence and the complexity of the outputs. The results of Scarton and Specia (2018), Martin et al. (2020b), and Kariuk and Karamshuk (2020) have shown that adding control tokens does help improve the performance of sentence simplification models quite significantly.

Building upon Martin et al. (2020b), we fine-tune T5 with all control tokens as defined in Martin et al. (2020b) to control different aspects of the output sentences. Moreover, we add one more control token (number of words ratio) in order to be able to generate new sentences with a similar length as the source but shorter in word length as we believe that the number characters ratio alone is not enough for the model to generate shorter words.

3 Model

In this work, we fine-tune T5 pre-trained model with the controllable mechanism on Text Simplification. T5 (A Unified Text-to-Text Transfer Transformer) (Raffel et al., 2019) is pre-trained on a number of supervised and unsupervised tasks such as machine translation, document summarization, question answering, classification tasks, and reading comprehension, as well as BERT-style token and span masking (Devlin et al., 2019). There are five different variants of T5 pre-trained models: T5-small (5 attention modules, 60 million parameters), and T5-base (12 attention modules, 220 million parameters). Due to the limited resources of Colab Pro, we are able to train only T5-small and T5-base.

3.1 Control Tokens

We use control tokens to control different aspects of simplification such as compression ratio (#Chars), paraphrasing (Levenshtein similarity), lexical complexity (word rank), and syntactic complexity (the depth of dependency tree) as defined in (Martin et al., 2020b). Then, we add another control token word ratio (#Words) to control word length. We argue that word ratio is another important control token because normally word frequency correlates well with familiarity, and word length can be an additional factor as long words tend to be hard to read (Rello et al., 2013b). Moreover, corpus studies of original and simplified texts show that simple texts contain shorter and more frequent words (Drndarević and Saggion, 2012). Therefore, we add word ratio to help the model generate sim-

plified sentences with a similar amount of words and shorter in word length, whereas #Chars alone could help the model regulate sentence length but not word length.

- **#Chars (C):** character length ratio between source sentence and target sentence. The number of characters in target divided by that of the source.
- **LevSim (L):** normalized character-level Levenshtein similarity (Levenshtein, 1966) between the source and target.
- **WordRank (WR):** inverse frequency order of all words in the target divided by that of the source.
- **DepTreeDepth (DTD):** maximum depth of the dependency tree of the target divided by that of the source.
- **#Words (W):** number of words ratio between source sentence and target sentence. The number of words in target divided by that of the source.

Table 1 shows an example of a sentence embedded with control tokens for training.

Source
simplify: W_0.58 C_0.52 L_0.67 WR_0.92 DTD_0.71 In architectural decoration Small pieces of colored and iridescent shell have been used to create mosaics and inlays, which have been used to decorate walls, furniture and boxes.
Target
Small pieces of colored and shiny shell has been used to decorate walls, furniture and boxes.

Table 1: This table shows how control tokens are embedded into the source sentence for training. The keyword **simplify** is added at the beginning of each source sentence to mark it as a simplification task.

4 Experiments

Our model is developed using the Huggingface Transformers library (Wolf et al., 2019)[2] with PyTorch[3] and Pytorch lightning[4].

[2] https://huggingface.co/transformers/model_doc/t5.html

[3] https://pytorch.org

[4] https://pytorchlightning.ai

4.1 Datasets

We use the WikiLarge dataset (Zhang and Lapata, 2017) for training. It is the largest and most commonly used text simplification dataset containing 296,402 sentence pairs from automatically aligned complex-simple sentence pairs English Wikipedia and Simple English Wikipedia which is compiled from (Zhu et al., 2010; Woodsend and Lapata, 2011; Kauchak, 2013).

For validation and testing, we use TurkCorpus (Xu et al., 2016), which has 2000 samples for validation and 359 samples for testing, and each complex sentence has 8 human simplifications. We also use a newly created dataset called ASSET (Alva-Manchego et al., 2020) for testing, which contains 2000/359 samples (validation/test) with 10 simplifications per source sentence.

4.2 Evaluation Metrics

Following previous research (Zhang and Lapata, 2017; Martin et al., 2020a), we use automatic evaluation metrics widely used in text simplification task.

SARI (Xu et al., 2016) compares system outputs with the references and the source sentence. It measures the performance of text simplification on a lexical level by explicitly measuring the goodness of words that are added, deleted and kept. So far, it is the most commonly adopted metric and we use it as an overall score.

BLEU (Papineni et al., 2002) is originally designed for Machine Translation and is commonly used previously. BLEU has lost its popularity on Text Simplification due to the fact that it correlates poorly with human judgments and often penalizes simpler sentences (Sulem et al., 2018). We keep using it so that we can compare our system with previous systems.

FKGL (Kincaid et al., 1975) In addition to SARI and BLEU, we use FKGL to measure readability; however, it does not take into account grammaticality and meaning preservation.

We compute SARI, BLEU, and FKGL using EASSE (Alva-Manchego et al., 2019)[5], a simplification evaluation library.

4.3 Training Details

We performed hyperparameters search using Optuna (Akiba et al., 2019) with T5-small and reduced

[5] https://github.com/feralvam/easse

size dataset to speed up the process. All models are trained with the same hyperparameters such as a batch size of 6 for T5-base and 12 for T5-small, maximum token of 256, learning rate of 3e-4, weight decay of 0.1, Adam epsilon of 1e-8, 5 warm up steps, 5 epochs, and the rest of the parameters are left with default values from Transformers library. Also, the seed is set to 12 for reproducibility. For the generation, we use beam size of 8. Our models are trained and evaluated using Google Colab Pro, which has a random GPU T4 or P100. Both have 16GB of memory, up to 25GB of RAM, and a time limit of 24h maximum for the execution of cells. Training of T5-base model for 5 epochs usually takes around 20 hours.

4.4 Choosing Control Token Values at Inference

In this experiment, we want to search for control token values that make the model generate the best possible simplifications. Thus, we select the values that achieve the best SARI on the validation set using the same tool that we use for hyperparameters tuning, Optuna (Akiba et al., 2019), and keep those values fixed for sentences in the test set. We repeat the same process for each evaluation dataset.

4.5 Baselines

We benchmark our model against several well-known state-of-the-art systems:

YATS (Ferrés et al., 2016)[6] Rule-based system with linguistically motivated rule-based syntactic analysis and corpus-based lexical simplifier which generates sentences based on part-of-speech tags and dependency information.

PBMT-R (Wubben et al., 2012) Phrase-based MT system trained on a monolingual parallel corpus with candidate re-ranking based on dissimilarity using Levenshtein distance.

UNTS (Surya et al., 2019) Unsupervised Neural Text Simplification is based on the encode-attend-decode style architecture (Bahdanau et al., 2014) with a shared encoder and two decoders and trained on unlabeled data extracted from English Wikipedia dump.

Dress-LS (Zhang and Lapata, 2017) A Seq2Seq model trained with deep reinforcement learning combined with a lexical simplification model to improve complex word substitutions.

DMASS+DCSS (Zhao et al., 2018b) A Seq2Seq model trained with the original Transformer architecture (Vaswani et al., 2017) combined with the simple paraphrase database for simplification PPDB. (Pavlick and Callison-Burch, 2016b).

ACCESS (Martin et al., 2020b) Seq2Seq system trained with four control tokens attached to source sentence: character length ratio, Levenshtein similarity ratio, word rank ratio, and dependency tree depth ratio between source and target sentence.

BART+ACCESS (Martin et al., 2020a) The system fine-tunes BART (Lewis et al., 2020) and adds the simplification control tokens from ACCESS.

4.6 Results

We evaluate our models automatically on two different datasets TurkCorpus and ASSET. In addition, we also perform a human evaluation on one of our models, which is described in Section 5. Table 2 reports the results of automatic evaluation of our models compared with other state-of-the-art systems. Our model **T5-base**+#chars+WordRank+LevSim+DepTreeDepth performs best on TurkCorpus with the SARI score of 43.31, while the other model **T5-base**+All Tokens performs best on ASSET with SARI score of 45.04 compared to the current state-of-the-art BART+ACCESS with the SARI score of 42.62 on TurkCorpus and 43.63 on ASSET. Following these results, our models out-perform all the state-of-the-art models in the literature in all approaches: rule-based, supervised and unsupervised approach even without using any additional resources.

5 Human Evaluation

In addition to automatic evaluation, we performed a human evaluation on the outputs of different systems. Following recent works (Alva-Manchego et al., 2017; Dong et al., 2019; Zhao et al., 2020), we run our evaluation on Amazon Mechanical Turk by asking five workers to rate using 5-point likert scale on three aspects: (1) Fluency (or Grammaticality): is it grammatically correct and well-formed?, (2) Simplicity: is it simpler than the original sentence?, and (3) Adequacy (or Meaning preservation): does it preserve meaning of the original sentence? More detailed instructions can be found in Appendix A. For this evaluation, we

[6] http://able2include.taln.upf.edu

Model	Data	ASSET			TurkCorpus		
		SARI↑	BLEU↑	FKGL↓	SARI↑	BLEU↑	FKGL↓
YATS	Rule-based	34.4	72.07	7.65	37.39	74.87	7.67
PBMT-R	PWKP (Wikipedia)	34.63	79.39	8.85	38.04	82.49	8.85
UNTS	Unsup. Data	35.19	76.14	7.60	36.29	76.44	7.60
Dress-LS	WikiLarge	36.59	86.39	7.66	36.97	81.08	7.66
DMASS+DCSS	WikiLarge	38.67	71.44	7.73	39.92	73.29	7.73
ACCESS	WikiLarge	40.13	75.99	7.29	41.38	76.36	7.29
BART+ACCESS	WikiLarge	43.63	76.28	6.25	42.62	78.28	6.98
T5-base+#Chars+WordRank							
+LevSim+DepTreeDepth	WikiLarge	44.91	71.96	6.32	**43.31**	66.23	6.17
T5-base+All Tokens	WikiLarge	**45.04**	71.21	5.88	43.00	64.42	5.63

Table 2: We report SARI, BLEU and FKGL evaluation results of our model compared with others on TurkCorpus and ASSET test set (SARI and BLEU higher the better, FKGL lower the better). BLEU and FKGL scores are not quite relevant for sentence simplification, and we keep them just to compare with the previous models. All the results of the literature are taken from Martin et al. (2020a), except YATS which is generated using its web interface.

randomly select 100 sentences from different simplification systems trained on WikiLarge dataset, except YATS which is rule-based. Table 3 reports the results in averaged values.

Model	Fluency	Simplicity	Adequacy
YATS	4.03*	3.62*	3.92*
DMASS+DCSS	3.84*	3.70*	3.48*
BART+ACCESS	**4.41**	**4.02**	4.13
Our Model	4.30	3.99	**4.18**

Table 3: Results of human evaluation on 100 random sentences selected from TurkCorpus test set. Best results are marked in bold, and results marked with an '*' are significantly lower than our model according to pared t-test with $p<0.01$. Our model in use here is **T5-base+All Tokens**.

The results have shown that our model performs lower in fluency and about the same in simplicity, and better in adequacy compared to BART+ACCESS. Based on our observation, there are two reasons that humans rated our model lower on fluency: (1) our model generates incorrect text format (without spaces) in some sentences (examples in Table 4). The problem can be easily spotted by human, but it does not affect the automatic evaluation as EASSE uses a tokenizer which can split the whole sentence correctly. (2) Our model tends to produce longer sentences than BART+ACCESS and in some cases, the subject is repeated twice when the sentence is split into two (e.g., relative clause). The repetition is also considered as one of the key features of simplification as it makes text easier to understand, but for native or fluent language speakers, repetition and the longer sentence make the fluency worse. Moreover, due to these problems, the evaluators also tend to lower the simplicity score as they consider it harder to read.

Sentence
So far the'celebrity'episodes have included Vic Reeves, Nancy Sorrell, and Gaby Roslin.
New South Wales'biggest city and capital is Sydney.

Table 4: Examples of incorrect text format generated by our model.

6 Ablation Study

In this section, we investigate the contribution of each token and different T5 pre-trained models to the performance of the system. Table 5 reports the scores of models trained on WikiLarge and evaluated with TurkCorpus and ASSET test set. Table 6 shows all control token values used for all

Model	ASSET			TurkCorpus		
	SARI↑	BLEU↑	FKGL↓	SARI↑	BLEU↑	FKGL↓
T5-small (No tokens)	29.85	90.39	8.94	34.50	94.16	9.44
T5-small + All Tokens	39.12	86.08	6.99	40.83	85.12	6.78
T5-base (No tokens)	34.15	88.97	8.94	37.56	90.96	8.81
T5-base:						
+#Words	38.51	84.02	7.45	38.86	89.10	8.61
+#Chars	39.58	79.22	6.06	38.95	84.81	7.76
+LevSim	41.58	82.52	6.53	40.90	85.45	7.55
+WordRank	41.40	76.75	5.85	41.44	85.46	7.67
+DepTreeDepth	40.08	81.94	6.56	39.18	87.60	7.81
T5-base:						
+WordRank+LevSim	42.85	80.38	4.47	41.75	83.90	7.42
+#Chars+WordRank+LevSim	44.89	56.76	5.93	42.91	67.09	6.53
+#Words+#Chars+WordRank+LevSim	44.65	58.52	5.52	43.03	68.11	5.96
+#Chars+WordRank+LevSim+DepTreeDepth	44.91	71.96	6.32	**43.31**	66.23	6.17
+All Tokens	**45.04**	71.21	5.88	43.00	64.42	5.63

Table 5: Ablation study on different T5 models and different control token values. Each model is trained and evaluated independently. We report SARI, BLEU and FKGL on TurkCorpus and ASSET test set. Control token values corresponded to each model are listed in the Table 6

Model	ASSET	TurkCorpus
T5-small (No tokens)		
T5-small + All Tokens	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$	$W_{1.05}$ $C_{0.95}$ $WR_{0.85}$ $L_{0.85}$ $DTD_{0.85}$
T5-base (No tokens)		
T5-base:		
+#Words	$W_{0.75}$	$W_{0.85}$
+#Chars	$C_{0.5}$	$C_{0.75}$
+LevSim	$L_{0.75}$	$L_{0.85}$
+WordRank	$WR_{0.25}$	$WR_{0.85}$
+DepTreeDepth	$DTD_{0.5}$	$DTD_{0.75}$
T5-base:		
+WordRank+LevSim	$W_{0.75}$ $L_{0.75}$	$W_{0.85}$ $L_{0.85}$
+#Chars+WordRank+LevSim	$C_{0.95}$ $WR_{0.75}$ $LevSim_{0.75}$	$C_{0.95}$ $WR_{0.85}$ $L_{0.85}$
+#Words+#Chars+WordRank+LevSim	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$
+#Chars+WordRank+LevSim+DepTreeDepth	$C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$	$C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$
+All Tokens	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$	$W_{1.05}$ $C_{0.95}$ $WR_{0.85}$ $L_{0.85}$ $DTD_{0.85}$

Table 6: These are the control token values used for the ablation study in Table 5. Each model is trained and evaluated independently. The values are selected using the hyperparameters search tool mentioned in Section 4.4.

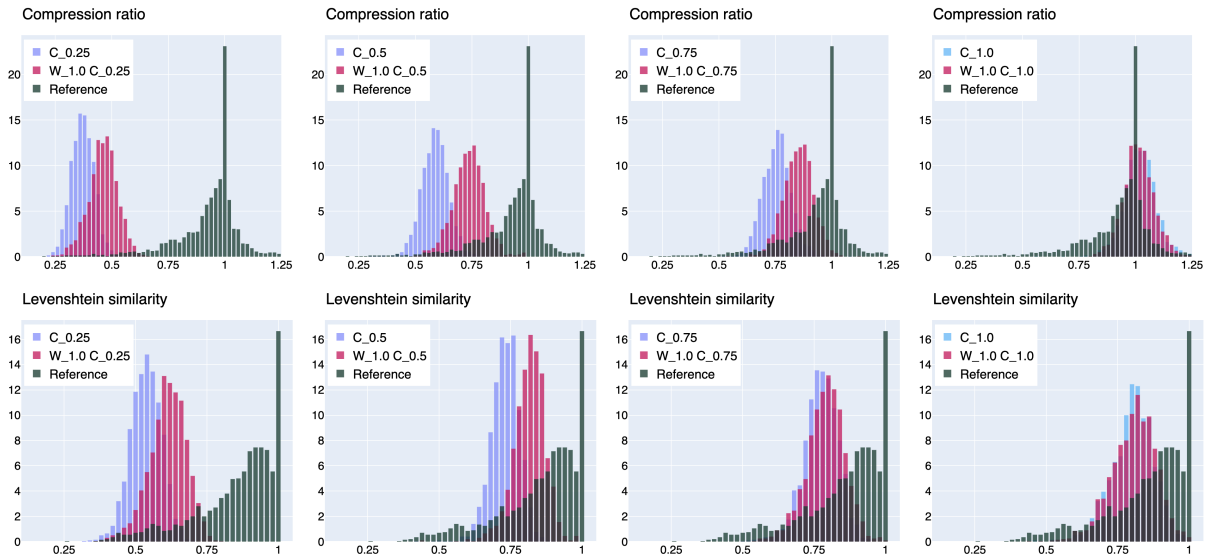


Figure 1: Influence of #Words and #Chars control tokens on the simplification outputs. Red represents the outputs of the model trained with four tokens, without #Words control token. Blue represents the outputs of the model trained with all five tokens. Green is the reference taken from TurkCorpus. The first row shows the compression ratio (number of chars ratio between system outputs and source sentences), and second row is the Levenshtein similarity (words similarity between system outputs and source sentences) of each model. We plot the results of the 2000 validation sentences from TurkCorpus. Other control token values used here are set to 0.75, the example in Table 7.

the models in Table 5 which are selected using the same process and tool as mentioned in Section 4.4.

Based on the results, the larger model (T5-base) performs better than the smaller one (T5-small) on both datasets (+3.06 on TurkCorpus, +4.3 on ASSET). It is due to the fact that larger model has more information which could generate better and more coherent text. Moreover, when added control tokens, the performance increases significantly. With only one token, WordRank performs best on TurkCorpus (+3.88 over T5-base) and LevSim on ASSET (+7.43 over T5-base).

Using pre-trained model alone does not gain much improvement, only when combined with control tokens, the results improve by a big margin (+3.06 and +9.28 for T5-small with and without tokens), and (+5.75 and +10.89 for T5-base with and without tokens).

6.1 Analysis on the effect of #Words

Our goal of using #Words control token is to make the model learn to generate shorter words whereas #Chars alone could help the model regulate the sentence length but not word length, so here we investigate how #Words and #Chars control tokens affect the outputs.

For the model with #Words token to work, it has to be incorporated with #Chars as #Words determines the number of words and #Chars limits the number of characters in the sentence. In our examples Table 7, we set #Words to 1.0, which means the number of words in the simplified sentence has to be similar to the original sentence, and #Chars is set to 0.5 and 0.75, which means keeping the same amount of words but reduces 50% or 25% of characters.

Figure 1 shows the differences in density distribution (first row) and similarity (second row) between model 1 in red without #Words token, model 2 in blue with #Words tokens, and the one in green is the reference. The first column #Chars is set to 0.25, second column #Chars=0.5, third column #Chars=0.75, fourth #Chars=1.0, and in all cases #words is set to 1.0. From the plots, we can see that model 1 does more compression than model 2, which means model 2 preserve more words than model 1.

Table 7 shows some example sentences comparing models with #Chars_0.75 and #Chars_0.5. When #Chars is set to 0.75, we do not see much difference between the two models, but when #Chars is set to 0.5, the two models have differences in terms of sentence length and word length. For example, the word **mathematics** in the example number one is replaced with the word **math** in model 2 (with #Words) and removed by model 1

Tokens	Model 1: **#Chars_0.5** WordRank_0.75 LevSim_0.75 DepTreeDepth_0.75 Model 2: **#Words_1.0 #Chars_0.5** WordRank_0.75 LevSim_0.75 DepTreeDepth_0.75
Source:	In order to accomplish their objective, surveyors use elements of geometry, engineering, trigonometry, **mathematics**, physics, and law.
Model 1:	In order to accomplish their objective, surveyors use geometry, engineering, and law.
Model 2:	In order to do this, surveyors use geometry, engineering, trigonometry, **math**, physics, and law.
Source:	The **municipality** has about 5700 inhabitants.
Model 1:	The municipality has 5700.
Model 2:	The **town** has about 5700.
Source:	A hunting dog refers to any dog who **assists** humans in hunting.
Model 1:	A hunting dog is any dog who hunts.
Model 2:	A hunting dog is a dog who **helps** humans in hunting.
Tokens	Model 1: **#Chars_0.75** WordRank_0.75 LevSim_0.75 DepTreeDepth_0.75 Model 2: **#Words_1.0 #Chars_0.75** WordRank_0.75 LevSim_0.75 DepTreeDepth_0.75
Source:	The park has become a traditional **location** for **mass demonstrations**.
Model 1:	The park has become a popular **place** for **demonstrations**.
Model 2:	The park has become a **place** for **people to show things**.
Source:	Frances was later **absorbed** by an **extratropical** cyclone on November 21.
Model 1:	Frances was later **taken** in by an extratropical cyclone.
Model 2:	Frances was later **taken** over by a cyclone on November 21.
Source:	There are claims that thousands of people were **impaled** at a single time.
Model 1:	There are claims that thousands of people were **killed**.
Model 2:	There are also stories that thousands of people were **killed** at a time.

Table 7: Examples showing the differences between the model with number of words ratio versus the one without. Model 1 trained with four tokens, without #Words control token, and model 2 trained with all five control tokens. All control token values used to generate the outputs are listed in the rows Tokens. We use bold to highlight the differences.

(without #Words). Second example, the word **municipality** is replaced by the word **town** by model 2, and model 1 simply keeps the word and crops the sentence (the same problem with the third example). In addition, the fourth example, the word **location** is replaced by both models with the word **place**, the phrase **mass demonstration** is reduced to **demonstration** by the model 1 whereas model 2 changes to four shorter words **people to show things**.

There are many cases where model 1 and model 2 generate the same substitutions, but very often model 1 tends to crop the end of the sentence or drops some words to fulfill the length constraint. Whereas model 2 tends to generate longer sentences than model 1, less crop, and very often replaces long complex words with shorter ones. Even though, based on the results from Table 2, adding the #Words control token does not significantly improve the SARI score and sometimes even lowers the score, it certainly holds its purpose.

7 Conclusion

In this paper, we propose a method which leverages a big pre-trained model (T5) fine-tuning it for the Controllable Sentence Simplification task. The experiments have shown good results of 43.31 SARI on TurkCorpus evaluation set and of 45.04 on ASSET evaluation set, outperforming the current state-of-the-art model. Also, we have shown that adding the control token #Words is useful for generating substitutions with a shorter lengths.

Acknowledgments

We acknowledge support from the project Context-aware Multilingual Text Simplifi-

cation (ConMuTeS) PID2019-109066GB-I00/AEI/10.13039/501100011033 awarded by Ministerio de Ciencia, Innovación y Universidades (MCIU) and by Agencia Estatal de Investigación (AEI) of Spain. Also, we would like to thank the three anonymous reviewers for their insightful suggestions.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

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 (Volume 1: Long Papers)*, pages 295–305.

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 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. 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): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Eduard Barbu, M Teresa Martín-Valdivia, Eugenio Martínez-Cámara, and L Alfonso Ureña-López. 2015. Language technologies applied to document simplification for helping autistic people. *Expert Systems with Applications*, 42(12):5076–5086.

Delphine Bernhard, Louis De Viron, Véronique Moriceau, and Xavier Tannier. 2012. Question generation for french: collating parsers and paraphrasing questions. *Dialogue & Discourse*, 3(2):43–74.

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

Raman Chandrasekar, Christine Doran, and Srinivas Bangalore. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th 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: Human Language Technologies*, pages 665–669.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. *arXiv preprint arXiv:1906.08104*.

Biljana Drndarević and Horacio Saggion. 2012. Towards automatic lexical simplification in spanish: an empirical study. In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 8–16.

Richard J Evans. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and linguistic computing*, 26(4):371–388.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*.

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.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Siddhartha Jonnalagadda and Graciela Gonzalez. 2010. Biosimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction. In *AMIA Annual Symposium Proceedings*, volume 2010, page 351. American Medical Informatics Association.

Oleg Kariuk and Dima Karamshuk. 2020. Cut: Controllable unsupervised text simplification. *arXiv preprint arXiv:2012.01936*.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. *arXiv preprint arXiv:1609.09552*.

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.

Reno Kriz, Joao Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-weighted loss and diverse reranking for sentence simplification. *arXiv preprint arXiv:1904.02767*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020a. Multilingual unsupervised sentence simplification. *arXiv preprint arXiv:2005.00352*.

Louis Martin, Éric Villemonte de La Clergerie, Benoît Sagot, and Antoine Bordes. 2020b. Controllable Sentence Simplification. In *LREC 2020 - 12th Language Resources and Evaluation Conference*, Marseille, France. Due to COVID19 pandemic, the 12th edition is cancelled. The LREC 2020 Proceedings are available at http://www.lrec-conf.org/proceedings/lrec2020/index.html.

Kerstin Matausch and Birgit Peböck. 2010. Easyweb–a study how people with specific learning difficulties can be supported on using the internet. In *International Conference on Computers for Handicapped Persons*, pages 641–648. Springer.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91.

Gustavo H Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3761–3767.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ellie Pavlick and Chris Callison-Burch. 2016a. Simple ppdb: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148.

Ellie Pavlick and Chris Callison-Burch. 2016b. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Luz Rello, Susana Bautista, Ricardo Baeza-Yates, Pablo Gervás, Raquel Hervás, and Horacio Saggion. 2013b. One half or 50%? an eye-tracking study of number representation readability. In *IFIP Conference on Human-Computer Interaction*, pages 229–245. Springer.

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

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 (Volume 2: Short Papers)*, pages 712–718.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of*

the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 35–40.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 896–902, Geneva, Switzerland. COLING.

Sanja Štajner, Iacer Calixto, and Horacio Saggion. 2015. Automatic text simplification for spanish: Comparative evaluation of various simplification strategies. In *Proceedings of the international conference recent advances in natural language processing*, pages 618–626.

Sanja Štajner and Maja Popović. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.

Sanja Štajner and Maja Popović. 2019. Automated text simplification as a preprocessing step for machine translation into an under-resourced language. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1141–1150.

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

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, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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, Volume 2 (Short Papers)*, pages 79–85, New Orleans, Louisiana. Association for Computational Linguistics.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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*, pages 409–420.

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, Jeju Island, Korea. 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*, 4:401–415.

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

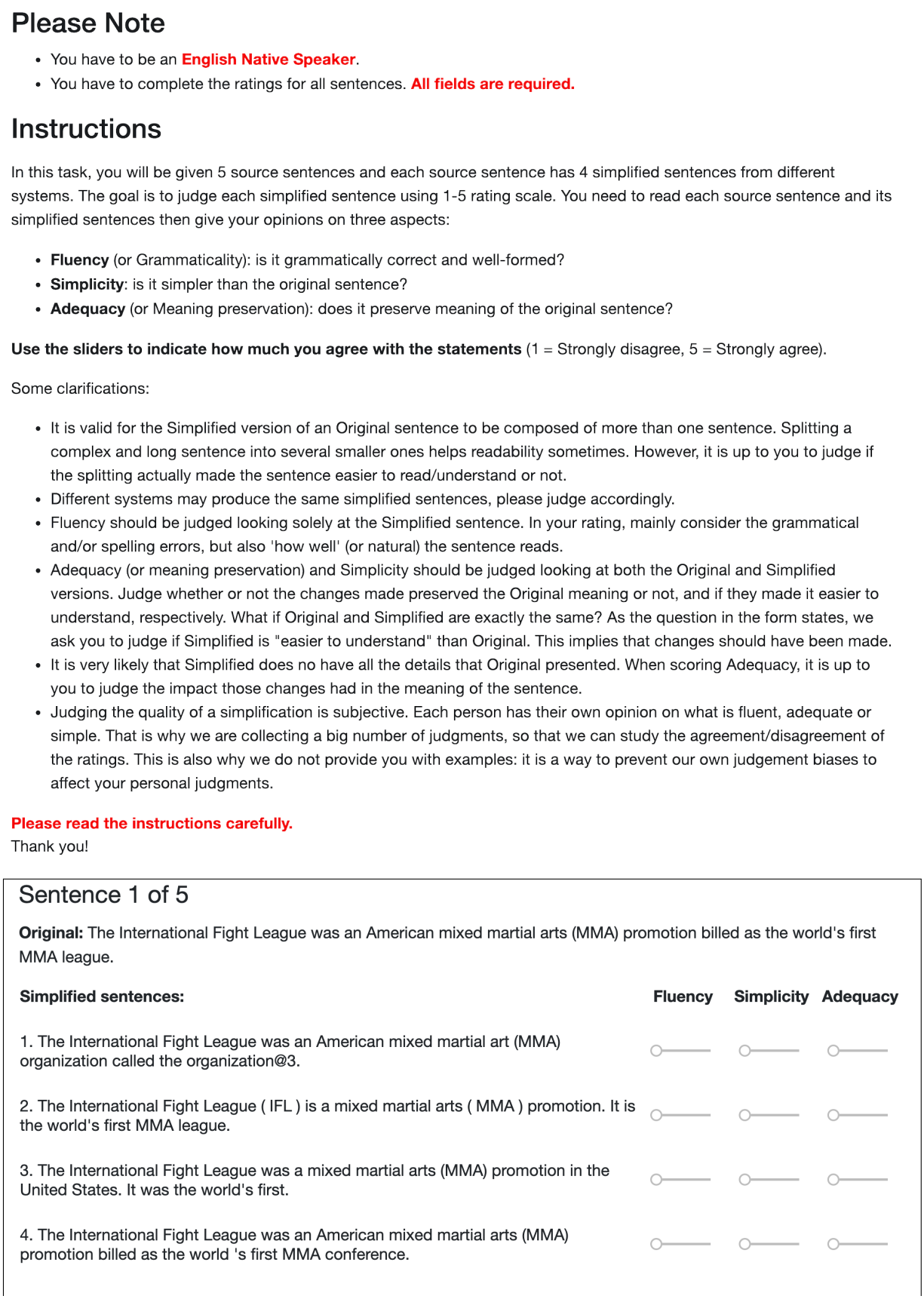
Sanqiang Zhao, Rui Meng, Daqing He, Saptono Andi, and Parmanto Bambang. 2018a. Integrating transformer and paraphrase rules for sentence simplification. *arXiv preprint arXiv:1810.11193*.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018b. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

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

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.

A Human Evaluation Interface



Please Note

- You have to be an **English Native Speaker**.
- You have to complete the ratings for all sentences. **All fields are required.**

Instructions

In this task, you will be given 5 source sentences and each source sentence has 4 simplified sentences from different systems. The goal is to judge each simplified sentence using 1-5 rating scale. You need to read each source sentence and its simplified sentences then give your opinions on three aspects:

- **Fluency** (or Grammaticality): is it grammatically correct and well-formed?
- **Simplicity**: is it simpler than the original sentence?
- **Adequacy** (or Meaning preservation): does it preserve meaning of the original sentence?

Use the sliders to indicate how much you agree with the statements (1 = Strongly disagree, 5 = Strongly agree).

Some clarifications:

- It is valid for the Simplified version of an Original sentence to be composed of more than one sentence. Splitting a complex and long sentence into several smaller ones helps readability sometimes. However, it is up to you to judge if the splitting actually made the sentence easier to read/understand or not.
- Different systems may produce the same simplified sentences, please judge accordingly.
- Fluency should be judged looking solely at the Simplified sentence. In your rating, mainly consider the grammatical and/or spelling errors, but also 'how well' (or natural) the sentence reads.
- Adequacy (or meaning preservation) and Simplicity should be judged looking at both the Original and Simplified versions. Judge whether or not the changes made preserved the Original meaning or not, and if they made it easier to understand, respectively. What if Original and Simplified are exactly the same? As the question in the form states, we ask you to judge if Simplified is "easier to understand" than Original. This implies that changes should have been made.
- It is very likely that Simplified does no have all the details that Original presented. When scoring Adequacy, it is up to you to judge the impact those changes had in the meaning of the sentence.
- Judging the quality of a simplification is subjective. Each person has their own opinion on what is fluent, adequate or simple. That is why we are collecting a big number of judgments, so that we can study the agreement/disagreement of the ratings. This is also why we do not provide you with examples: it is a way to prevent our own judgement biases to affect your personal judgments.

Please read the instructions carefully.
Thank you!

Sentence 1 of 5

Original: The International Fight League was an American mixed martial arts (MMA) promotion billed as the world's first MMA league.

Simplified sentences:	**Fluency**	**Simplicity**	**Adequacy**
1. The International Fight League was an American mixed martial art (MMA) organization called the organization@3.			
2. The International Fight League (IFL) is a mixed martial arts (MMA) promotion. It is the world's first MMA league.			
3. The International Fight League was a mixed martial arts (MMA) promotion in the United States. It was the world's first.			
4. The International Fight League was an American mixed martial arts (MMA) promotion billed as the world 's first MMA conference.			

Figure 2: Our interface is based on the one proposed by Kriz et al. (2019), and the consent form based on Alva-Manchego et al. (2020).