mathematics

MDPI

Article

Text Simplification to Specific Readability Levels

Wejdan Alkaldi [1,*] and Diana Inkpen [2,*]

[1] Department of Information Technology, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

[2] School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward, Ottawa, ON K1N 6N5, Canada

* Correspondence: walkaldi@ksu.edu.sa (W.A.); diana.inkpen@uottawa.ca (D.I.); Tel.: +96-650-620-8504 (W.A.); +1-613-56258000 (ext. 6711) (D.I.)

Abstract: The ability to read a document depends on the reader's skills and the text's readability level. In this paper, we propose a system that uses deep learning techniques to simplify texts in order to match a reader's level. We use a novel approach with a reinforcement learning loop that contains a readability classifier. The classifier's output is used to decide if more simplification is needed, until the desired readability level is reached. The simplification models are trained on data annotated with readability levels from the Newsela corpus. Our simplification models perform at sentence level, to simplify each sentence to meet the specified readability level. We use a version of the Newsela corpus aligned at the sentence level. We also produce an augmented dataset by automatically annotating more pairs of sentences using a readability-level classifier. Our text simplification models achieve better performance than state-of-the-art techniques for this task.

Keywords: text simplification; deep learning; reinforcement learning; readability level; data augmentation

MSC: 68T50

check for updates

Citation: Alkaldi, W.; Inkpen, D. Text Simplification to Specific Readability Levels. *Mathematics* **2023**, *11*, 2063. https://doi.org/10.3390/math11092063

Academic Editor: Catalin Stoean

Received: 23 March 2023
Revised: 16 April 2023
Accepted: 19 April 2023
Published: 26 April 2023

1. Introduction

The ultimate goal of writing a text is to communicate. Any written text must be readable and understandable to its targeted audience. However, readers might have a low level of reading skills and cannot understand a given text. The organization of the text and the vocabulary used affects the text readability level. Manipulating these features could increase the readability of the text to a certain level that allows poor literacy readers or children to read and understand the written text.

Text Simplification (TS) techniques available now do not use the readability level as a required feature for the output text. Instead, they typically simplify the given text to whatever readability level it can reach. For instance, consider readability levels from 1 to 4 (as used in Newsela dataset to classify documents to their readability level), where level 1 represents a very complex text to read and level 4 represents a very simple text to read. If a reader with reading level 3wants to read a text with readability level 1, the text must be simplified to the reader's level at least, i.e., level 3 or 4. However, using the available simplification techniques, the original text could be simplified to a simple text with a readability level that cannot be controlled. In the example, if the output text readability is at level 2, then the text is still difficult for the reader to grasp and comprehend, despite being simplified from its original state. So, the original text must be re-simplified to represent the readability level of at least 3. Unfortunately, this scenario cannot be executed with the available techniques since the readability level of a text does not play a role in the present simplification models. To fill this gap, we create a novel state-of-the-art simplification model that is trained over aligned sentences from the Newsela dataset (https://newsela.com/data/ accessed on 6 November 2019) [1].

Also, we produce additional data in an automatic way, to improve the performance of the simplification. The model takes a complex text with a low readability level, and produces a simplified version of the text that considers the required readability level. This will ensure that every simplified text will be readable and understandable by its targeted audience.

We start with related works in Section 2, where we express simplification projects available in non-English languages, followed by deep learning techniques used in current TS projects. Then, we explain the framework of our simplification in Section 3. We start with the datasets used, the simplification models, and the evaluation measures applied. After that, we discuss the experiments in Section 4, including training and testing setup, examining samples of generated simplified sentences, and presenting the results for the experiments. Section 5 compares and analyses the performance of all the trained models on the same test set followed by the limitations we faced in Section 6. Finally, in Section 7, we conclude our paper and present directions for future work.

2. Related Work

2.1. Natural Language Simplification

In Natural Language Processing (NLP) applications, early Text Simplification (TS) systems are built based on statistical machine translation models like PBMT-R [2] and Hybrid [3]. While most TS researches are done for the English language, TS is also applied across many other languages. Every language has its own specific characteristics. It is non-trivial to re-implement existing TS techniques into other languages. Every language has different characteristics that need to be handled differently. Languages like Latin and Swedish, use complex verb conjugations; e.g. specific forms of verbs express passive voice sentences. While Mandarin Chinese, have unchangeable verb forms when expressing passive voice sentences. This means their verbs do not have any tenses. Several projects focus on re-implementing existing TS techniques and adapting them to their own language. TS is a major challenge in all languages. We found many projects and tools in TS for different languages. Most of them were developed to assist people with disabilities or learning difficulties.

The KURA project [4] is a Japanese project and one of the earliest works found in TS. It aims to simplify Japanese language text for deaf students by developing a lexico-structural paraphrasing engine. KURA introduced the concept of phrase-based simplification which identifies then simplifies complex terms [5]. SIMPLIFICA [6] is another tool for producing simplified texts in Portuguese. It helps authors write simple texts for poor literate readers. The author writes a text and receives a simplified version. SIMPLIFICA uses lexical and syntactic simplification features to assist the readability of the text targeting Brazilian Portuguese. The tool performs simplification on the sentence level. Similarly, the PorSimples project [7] developed text adaptation tools for Brazilian Portuguese. The tools developed serve both people with poor literacy levels and authors who produce texts for this audience. It is one of the largest TS projects with three main systems and many types of simplification techniques investigated in [8]. Its main purpose is to increase the comprehension of written texts through the simplification of their linguistic structure. It replaces uncommon words with more usual words. It also changes the sentence syntactic structure to an easier form to avoid ambiguity. The Simplext project [9] develops tools that produce a simplified text for the Spanish language. It has a particular focus on producing applications of TS for dyslexic readers [8].

Another work [10] developed a pioneering TS model that can control the sentence level. It trained a TS model on a corpus of sentences with tags referring to 11 grade levels (2–12) [11]. The trained model generates sentences of a desired level specified by a tag attached to the input. This model controls the syntactic complexity but often produces difficult words for the target grade level [12]. It uses the Naive Bayes classifier from scikit-learn toolkit [13] with extra few features which could be improved. To enhance this TS work, an Auto-Regressive Transformers (AR) model is proposed [12] that controls the lexical complexity using weights. The model is trained on a dataset with weights added to training loss according to the levels of words from [10]. Therefore, it generates only the words with the desired level. Both [10,12] use only Sequence-to-Sequence (Seq2Seq) model as the main TS component.

Later, EDITOR was proposed [14] which is a Non Auto-Regressive transformer (NAR) where the decoder layer is used to apply a sequence of edits on the initial input sequence. The sequence can be empty or has repositioning and insertion commands. The model never learns to delete tokens from the source, instead learning to delete tokens inserted by the model. An enhanced version of this work is found [15] that identifies complex words from the source that are too complex for the target grade. These words are deleted from the initial sequence before getting refined by EDITOR. All these models [10,12,14,15] focus on grades "2–12" as the main levels to simplify to. Focusing on only 4 simplified versions gives more balanced dataset to train on.

There are other TS works that are developed for a specific domain. One of these domains is medical and biomedical fields using TS across many languages like English, Spanish, and French [16–19]. Another domain is the legal field. TS can be used to simplify legal documents for individuals to help in understand and comprehend any required legal text [20–23].

2.2. *Deep Learning in Text Simplification*

Deep Learning (DL) is the state-of-the-art approach for solving many NLP problems. It uses neural networks as the central component to process and analyze written text, then produce the output results. There are only few tools that we found for TS using DL techniques. DRESS [24] is one of the few NLP systems that provides a reinforcement learning-based TS model. It allows only one level of simplification instead of several simplified levels of a given text, as we do in our task.

Another state-of-the-art sentence simplification system that uses DL methods is EditNTS [25]. Its model learns explicit edit operations (ADD, DELETE, and KEEP) via a neural programmer-interpreter approach. It is trained to predict a series of edit operations for each word of the original complex sentence. Then, using this series of operations, it generates the simplified sentence. EditNTS favors generating short sentences with big semantic deviation [26]. It produces only one level of simplification, as all other simplification systems except the one we are proposing in this paper. However, we are able to train EditNTS on our data for multiple levels for comparison purposes.

3. Simplification Framework

3.1. *Dataset*

We use the Newsela Corpus that contains 10,786 documents with readability levels varying from 0 to 4 that targets students of grades between 2 and 12. The corpus contains 2154 original complex documents labeled with Level 0 which means that they are not simplified and they are difficult to read. For every complex document, it provides four simplified versions written by expert editors. Each version represents a readability level that varies from Level 1 (representing the first level of simplification) to Level 4 (the most readable version of the document). The higher the readability level number, the simpler the document text.

We used sentence alignment on Newsela dataset as found in [27], which uses a neural CRF model. The aligned pairs of sentences are labeled with the readability level of the target sentence. We excluded pairs that had non-English words or consisted less than three words in a sentence (not a proper complete sentence) and obtained 464,555 pairs of Newsela Aligned Sentences (hereafter, the NAS dataset).

We also classified more sentences to enrich our dataset. Several works were put together to help determine the text readability level [6,28–37]. However, we decided to use a DL classifier that classifies text into five readability levels (0–4) found in [38]. We modified the document classification features from that system by removing paragraph features in order to be able to classify the simplicity level of a text at the sentence level. Then we trained and tested the modified sentence classifiers on the NAS dataset (split into 80% for training and 20% for test) to find the best classification model. Table 1 shows the classification results on the sentence level. Similar to the document classification results, the best sentence classification model was using CNN classifier with an accuracy of 85.52%. Using the trained classifier against Wikipedia Corpus and Mechanical Turk Corpus, we produced 238,019 pairs of automatically Classified Simplified Sentences (hereafter CSS). We used CSS to augment the NAS dataset and obtained 702,574 pairs of sentences as our Augmented Simplification Dataset (hereafter ASD), in order to be able to provide more training data for our models. All three datasets are divided into four categories (level 1 to level 4) based on the readability level of the target sentences (simple sentences). For every category, we split the datasets into 90% for training (10% of it for validation) and 10% for test.

Table 1. Sentence classifiers results using aligned sentences.

Dataset	Classifier Model	Accuracy
Training (xval)	CNN	**85.69%**
	SVM	81.02%
	Random Forest	85.64%
Test	CNN	**85.52%**
	SVM	80.68%
	Random Forest	85.48%

3.2. *Simplification Models*

3.2.1. Seq2seq Model with Attention

We use the model Seq2Seq with Attention layer (S2SA) as a base for our work. Seq2seq models are used in solving most of text-to-text generation problems, including TS. The model takes a sequence of items (words) as an input, and generates another sequence of items as an output. The model consists at least two Recurrent Neural Networks (RNNs), an Encoder, and a Decoder [39]. A simple illustration of S2SA model we used in this work is shown in Figure 1 with a simplification sentence example. Our model uses Gated Recurrent Units (GRU) as RNN units, since GRU requires less memory units than Long Short Term Memory (LSTM); thus, it trains faster. Besides, according to [40], when using long text and a small dataset, GRU performance surpassed that of LSTM. Therefore, using GRU is more appropriate for our work. Both the encoder and the decoder have an embedding layer with 256 dimensions, 256 hidden states, GRUs unites with dropout equals to 10%, and a linear layer to pass the output through. To enhance the performance when dealing with long sentences, we added an attention layer [41] to the decoder to find where to focus for better-predicted outputs. The layer contains two linear layers with 256 hidden dimensions. With this layer, our S2SA model can deal with all sentences of any length without forgetting the source input.

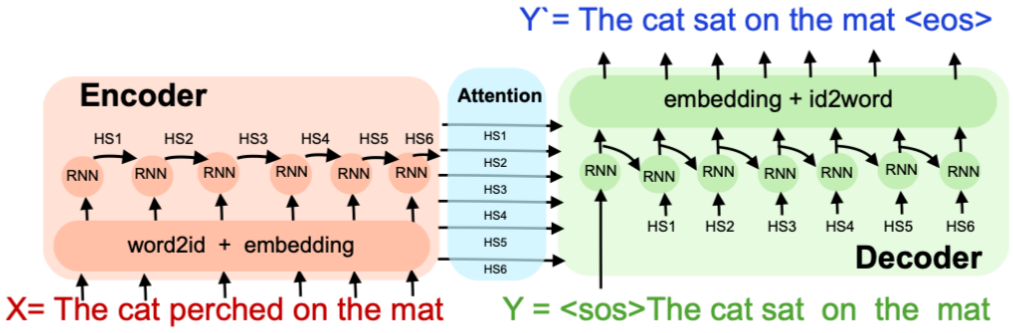
Y`= The cat sat on the mat <eos>
Encoder
Attention
embedding + id2word
HS1
HS2
HS3
HS4
HS5
HS6
RNN
word2id + embedding
Decoder
X= The cat perched on the mat
Y = <sos>The cat sat on the mat

Figure 1. Illustration of S2SA model with a simple simplification example.

3.2.2. Reinforcement Learning

Reinforcement Learning (RL) is the state-of-the-art technology in DL for TS. To further boost our simplification model results, we used the S2SA model with Reinforcement Learning loop (S2SARL), Figure 2 shows a simple illustration of our RL model with an example. RL is a machine learning technique that enables an agent to learn in an interactive environment by trial and error using rewards earned from its own actions [42]. The main components of a RL system are the environment and the agent. We start the model by creating the vocabulary dictionary table using the words found in the dataset. Then for the agent, we set up our S2SA model introduced earlier in Section 3.2.1 to produce set of actions (words) using the dictionary table created. We initialized the reward, status, total loss, and the vocabulary dictionary table to zeros. Then we built a step function that uses the environment tools to perform a simplification for a given sentence (sequence of input words).

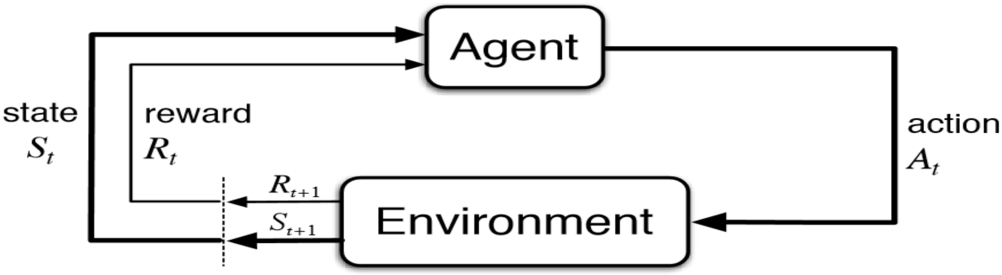

Figure 2. Simple Reinforcement Learning model.

After performing every step, the agent updates the reward, status, loss, and the vocabulary dictionary values with new values based on the predicted simplified sentence (sequence of actions).

To prepare the environment, we set up the *Target* Level number (1 to 4) and provide tools to help the agent during training like: observe current status, get all possible outputs for an action (predicted word), and give appropriate rewards based on a set of chosen actions (predicted simplified sentence). The reward value is determined by the readability level of the predicted sentence ($PrdS$). For every ($PrdS_t$), we use the adapted readability level classifier ($Rclf$) found in [38] to classify the $PrdS_t$ sentence into its readability level. Then we calculate the reward R_t as follows:

$$R_t = \begin{cases} -0.5 & \text{if } Rclf(PrdS_t) < Target \\ +2.0 & \text{if } Rclf(PrdS_t) == Target \\ +1.0 & \text{if } Rclf(PrdS_t) > Target \end{cases}$$

Using the reward function, if the predicted sentence readability level is less than the Targeted Level, the environment gives −0.5 as a penalty. This encourages the agent to predict simpler sentences for their next step. If the predicted sentence readability level matches the Targeted Level, the reward will be +2.0 to encourage the agent to keep this level of simplicity. However, if the output is too simple, i.e., the readability level is more than the Targeted Level, the reward will be only +1.0. Penalizing the agent with negative rewards for exceeding the Targeted Level did not improve the output. Yet giving a smaller reward like +1.0, improved the results.

Figure 3 shows the structure of our S2SARL model, with a simple simplification example. The RL loop aims to maximize the reward given to the agent at every step during training stage. Therefore, the agent chooses the actions that influence the environment to produce higher rewards. Our RL loop is different from the one in the DRESS system. It is designed specifically for our task of simplifying a sentence to a specified readability level.

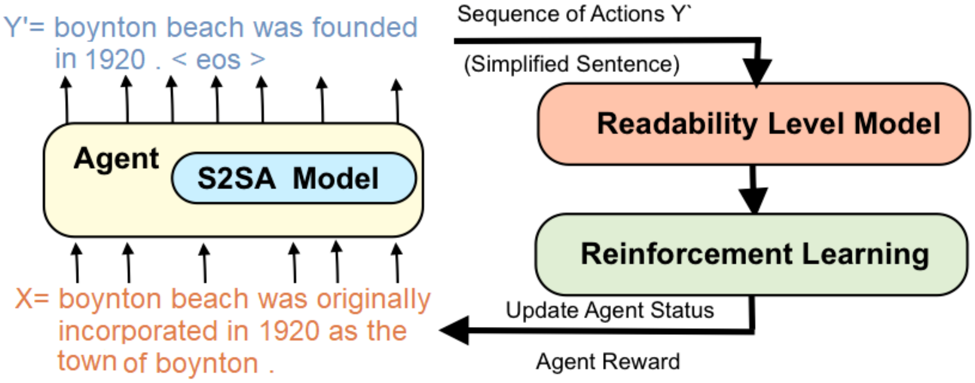

Figure 3. Illustration of S2SARL model with a simple simplification example.

3.3. Evaluation Method

To evaluate our work, we use EditNTS [25] as a notable simplification model to compare our work with. EditNTS uses DL to produce a series of edit operations (delete, keep, and add) to operate on the original sentence. The evaluation will consider 12 trained versions of each model: EditNTS, S2SA, and S2SARL. Each model will be trained against the datasets NAS, CSS, and ASD including the categories from Level 1 to Level 4 for each dataset.

After training each model, we report the results using System output Against References and against the Input sentence (SARI) and BiLingual Evaluation Understudy (BLEU) scores since they are popularly used in measuring the quality of TS models. SARI measures the simplicity of a sentence by focusing on the words added, deleted and kept [43]. While BLEU score is more related to the meaning preservation as shown in [44]. Then, we apply the 36 resulted trained models against one common test data. We choose the test part of the ASD dataset, Level 1 to Level 4, since they are not automatically classified and rather assigned by professional editors as mentioned in Section 3.1. We then compare the reported scores.

4. Experiments

4.1. Training and Testing

We train our simplification models S2SA and S2SARL along with EditNTS against every readability category, labeled from level 1 to level 4, from the training parts of NAS and CSS datasets. We also train them against ASD categories, which includes both NAS and CSS datasets as an augmented simplification dataset. To avoid memory problems due to the vocabulary dictionary size for each dataset, we use a batch size of 128 for training the models to level 1 and 4, and batch size of 64 for training the models to level 2 and 3. The number of epochs are set to 20 for training all the models over all four categories. We record SARI and BLEU scores for all the experiments to measure the simplification models' performance on the set aside test sets.

4.2. Examples of Generated Sentences

Examples of simplified sentences using the S2SARL model that was trained against NAS with targeted readability level 3 are shown in Table 2. Generated and target sentences could have the same words but with different word spelling, e.g., honour and honor. This is due to the available spelling found in the dictionary table during training phase. Also, some words are annotated as <unk> which means that the word was not present during training in the dictionary table.

Table 2. Simplified sentences using S2SARL trained against NAS and readability level set to 3.

Sentence	Text	Readability Level
Source Target Predicted	volterra is a town in the tuscany region of italy. volterra is a town in italy. volterra is a town in tuscany, <eos>	0 3 **3**
Source Target Predicted	he was appointed cbe in 1969. he was given the honour of cbe in 1969. he was given the honor of cbe in 1969. <eos>	0 3 **3**
Source Target Predicted	the seat of the district is the town of cossonay. the capital is the town of cossonay. the capital is the town of <unk>.<eos>	0 3 **4**
Source	punctuation, capitalization, and spacing are usually ignored, although some (such as "rats live on no evil star") include the spacing.	0
Target Predicted	rats live on no evil star. rats live on no evil star. <eos>	3 **3**

The S2SARL model aims to produce sentences with readability matching the target level. The table shows the readability level for the predicted sentences. Most of them are level 3 to match the target level as expected, but sometimes the sentence has higher readability level like level 4 in Table 2. That is due to the reward function in the RL loop introduced in Section 3.2.2. We did not penalize the agent for exceeding the target level, +2 for reaching target level and +1 when exceeding the level.

Comparing the performance of the two models S2SARL and S2SA, Table 3 shows the prediction of the two models using the CSS with readability level of 4. The table shows how S2SA sometimes produce sentences with lower readability level than we anticipated, which is level 4 in these sentences. This is because S2SA does not take into consideration the readability level when simplifying. Also, if we look at the Target sentences in Table 3 and compare it with the generated sentences, we see an improvement in the simplified sentences generated with S2SARL model compared with the ones generated with S2SA model.

Table 3. Simplification using S2SA and S2SARL with level 4 augmented data.

Sentence	Text	Readability Level
Source Target S2SA S2SARL	thank you for your contributions. thank you for your changes. thank you for your changes. <eos> thank you for your changes. <eos>	 4 4
Source Target S2SA S2SARL	the capital of the state is aracaju (pop 664,908). the state 's capital is aracaju. the capital of the state is . . <eos> the capital of the state is aracaju. <eos>	 3 4
Source Target S2SA S2SARL	the birthstone for july would be a red ruby. its birthstone is the ruby. july 's birthstone is the ruby. <eos> its birthstone is the ruby. <eos>	 3 4
Source Target S2SA S2SARL	boynton beach was originally incorporated in 1920 as the town of boynton. boynton beach was founded in 1920. boynton was part of the town of boynton. <eos> boynton beach was founded in 1920. <eos>	 4 4

4.3. Results

After training and validating the models, we apply them on the test data that was split from each dataset category. The results on the test data are shown in Tables 4–6. The tables show that S2SARL model always gives the best BLEU score compared with S2SA and EditNTS for all readability levels. However, when the dataset is small, like shown for level 1 and level 4 in Table 5, the S2SA model obtains better SARI scores. The model S2SARL gives better SARI results only when trained on a bigger dataset, and that is why we augmented the simplification dataset (to produce the ASD set). EditNTS prefers to generate short sentences with big semantic deviation. It usually deletes important information of the original sentences and generates shorter sentences, as discussed in [26]. This explains the low EditNTS scores in the tables.

Table 4. Test scores for TS models trained on Newsela Aligned Sentences (NAS) using NAS test data.

Dataset	Model	SARI	BLEU
To Level 1 5129 pairs	EditNTS S2SA S2SARL	26.48 **31.76** 31.57	65.23 65.61 **70.22**
To Level 2 9780 pairs	EditNTS S2SA S2SARL	20.62 27.18 **31.56**	46.81 53.95 **60.53**
To Level 3 13,922 pairs	EditNTS S2SA S2SARL	20.26 30.83 **32.27**	33.28 45.24 **53.85**
To Level 4 17,626 pairs	EditNTS S2SA S2SARL	23.21 31.69 **32.42**	23.97 42.60 **50.97**

Table 5. Test scores for TS models trained on Classified Simplified Sentences (CSS) using CSS test data.

Dataset	Model	SARI	BLEU
To Level 1 1350 pairs	EditNTS S2SA S2SARL	21.92 **28.12** 26.34	49.45 51.23 **67.67**
To Level 2 10,652 pairs	EditNTS S2SA S2SARL	21.89 30.97 **32.57**	49.30 65.79 **70.92**
To Level 3 9380 pairs	EditNTS S2SA S2SARL	17.12 31.56 **32.50**	35.13 59.26 **64.50**
To Level 4 2422 pairs	EditNTS S2SA S2SARL	21.60 **29.79** 29.36	27.35 65.78 **69.70**

Table 6. Test scores for TS models trained on Augmented Simplification Dataset (ASD) using ASD test data.

Dataset	Model	SARI	BLEU
To Level 1 6478 pairs	EditNTS S2SA S2SARL	25.32 **32.07** 30.75	61.25 69.18 **70.23**
To Level 2 20,432 pairs	EditNTS S2SA S2SARL	20.99 28.43 **32.30**	46.94 60.31 **65.22**
To Level 3 23,301 pairs	EditNTS S2SA S2SARL	19.89 30.67 **32.47**	33.77 50.24 **56.43**
To Level 4 20,048 pairs	EditNTS S2SA S2SARL	23.06 31.62 **32.62**	24.86 44.08 **51.10**

5. Comparison and Analysis

The TS models applied in this work (EditNTS, S2SA, and S2SARL) are trained on 12 different datasets: NAS (Level-1 to Level-4), CSS (Level-1 to Level-4), and ASD (Level-1 to Level-4). The experiments produced 36 trained models: 12 EditNTS, 12 S2SA, and 12 S2SARL models as shown in the Tables 4–6. To compare the performance of all those models, we test them on the same test data that should not include any automatically classified sentences as targets, i.e, CSS and ASD. Therefore, we tested all the models on the NAS test data (Level-1 to Level-4) since all its target sentences are classified and labeled by expert editors as explained in Section 3.1.

The test results are compared as shown in Table 7. Looking at the table, S2SARL model outperforms the other two simplification models across all readability levels. That is due to the involvement of the output sentence readability level during the training phase of the model (in the RL loop). As shown in Table 7, S2SARL models give the best BLEU scores across all four readability levels when trained with ASD since it is the largest simplification dataset (in term of the number of training sentence pairs) compared with NAS and CSS. However, for SARI scores, S2SARL models report the best scores throughout all four readability levels when trained against the CSS dataset. Although ASD is larger than CSS since it contains the CSS and the NAS datasets, training S2SARL model over ASD did not increase the SARI scores. This could be due to the alignment technique used for aligning Newsela sentences (NAS) in [27]. The alignment includes sentence splitting, merging, and paraphrasing with deletion which resulted in more meaningful sentences, while the sentences found in CSS do not include sentence splitting or merging.

To summarise the analysis, S2SARL gives better BLEU scores when trained with ASD (which includes CSS and NAS with sentence splitting, merging, and paraphrasing). That is because BLEU score focuses on grammar and meaning [18]. On the other hand, SARI score pays more attention to the lexical aspects of the sentences [43]. Therefore, S2SARL returns good SARI scores when trained against CSS only, where the lexical part is not changed as much compared with the NAS dataset.

Table 7. Testing 36 simplification models on ASD test data across all four readability levels. NAS: Newsela Aligned Sentences, CSS: Classified Simplified Sentences, and ASD: Augmented Simplification Dataset.

Test on NAS Level 1 (5129 Pairs)			
Trained on	**Model**	**SARI**	**BLEU**
NAS-Level1	EditNTS	26.48	65.23
	S2SA	31.76	65.61
	S2SARL	31.57	70.22
CSS-Level1	EditNTS	26.41	65.37
	S2SA	34.07	33.35
	S2SARL	**34.08**	36.25
ASD-Level1	EditNTS	26.81	65.70
	S2SA	31.36	73.11
	S2SARL	31.26	**76.47**
Test on NAS Level 2 (9780 pairs)			
Trained on	**Model**	**SARI**	**BLEU**
NAS-Level2	EditNTS	20.62	46.81
	S2SA	27.18	53.95
	S2SARL	31.56	60.53
CSS-Level2	EditNTS	15.66	46.15
	S2SA	31.36	35.07
	S2SARL	**32.51**	43.67
ASD-Level2	EditNTS	20.63	46.82
	S2SA	25.23	61.78
	S2SARL	31.73	**68.69**

Table 7. *Cont.*

Test on NAS Level 3 (13,922 pairs)			
Trained on	**Model**	**SARI**	**BLEU**
NAS-Leve3	EditNTS	20.26	33.28
	S2SA	30.83	45.24
	S2SARL	32.27	53.85
CSS-Leve3	EditNTS	15.72	32.36
	S2SA	33.23	23.30
	S2SARL	**33.24**	23.27
ASD-Leve3	EditNTS	20.52	33.77
	S2SA	32.21	52.96
	S2SARL	32.23	**61.88**

Test on NAS Level 4 (17,626 pairs)			
Trained on	**Model**	**SARI**	**BLEU**
NAS-Leve4	EditNTS	23.21	23.97
	S2SA	31.69	42.60
	S2SARL	32.42	50.97
CSS-Leve4	EditNTS	12.71	24.22
	S2SA	33.23	12.32
	S2SARL	**33.24**	12.41
ASD-Leve4	EditNTS	23.32	24.86
	S2SA	32.31	61.22
	S2SARL	32.32	**61.38**

6. Limitations

Working on a dataset that consists four levels of simplification was limited to the sentences available by Newsela dataset. Although we automatically augmented the dataset with more labeled simplified sentences, it would be more efficient if we work on a larger dataset labeled by expert users like Newsela. Also, applying reinforcement learning during training phase is time-consuming compared with a plain S2SA model. Therefore, we applied only one method to reward the agent using the output readability level.

7. Conclusions and Future Work

The goal of our simplification method was to produce simple sentences at a certain readability level using DL models. We used aligned sentences from the Newsela dataset (NAS) and augmented the corpus with automatically classified sentences from the Wikipedia and the Mechanical Turk datasets (CSS), creating a novel augmented simplification dataset (ASD) that we used later for simplification. Then we created the simplification models, S2SA and S2SARL, where the S2SARL model employs the readability level as part of the simplification process using the reinforcement learning loop to produce simplified sentence to the desired readability level. We trained EditNTS and the created models with the same datasets NAS, CSS, and ASD, to compare their performance. We found that S2SARL always outperform the other two models for every dataset used. We also compared all the simplification models (S2SA, S2SARL, and EditNTS), that were trained on different datasets, by testing them on the same test data, the test part of NAS. The results of SARI and BLEU scores were compared and analysed.

Our work brings novelty in the area of TS in the way we train our deep leaning models using augmented data, and in the way we perform the reinforcement leaning loop using a readability classifier.

In future work, other evaluation measures could be incorporated in the RL loop as a part of the reward function, for example the SARI score to measure simplicity, or the cosine between the generated and and the target sentences vectors to measure their similarity, in addition to the readability level given by the classifier. Also, the simplification models could be trained on paragraph level using the Newsela aligned paragraphs. Another direction of future work is to develop a similar system for other languages, for specific level of simplification targeted.

Author Contributions: W.A. wrote the article and implemented and tested the system. D.I. helped with the design and with revisions to the article. All authors have read and agreed to the published version of the manuscript

Funding: The first author was funded by the Research Center of College of Computer and Information Sciences, Deanship of Scientific Research in King Saud University; and the second author was funded by the Natural Science and Engineering Research Council of Canada (NSERC).

Data Availability Statement: We are making our dataset available to other researchers, upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Newsela Inc. Newsela Dataset. 2019. Available online: http://newsela.com/data/ (accessed on 1 May 2020).
2. Wubben, S.; Van Den Bosch, A.; Krahmer, E. Sentence simplification by monolingual machine translation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jeju Island, Republic of Korea, 8–14 July 2012; pp. 1015–1024.
3. Narayan, S.; Gardent, C. Hybrid simplification using deep semantics and machine translation. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–24 June 2014; pp. 435–445.
4. Takahashi, T.; Iwakura, T.; Iida, R.; Fujita, A.; Inui, K. KURA: A transfer-based lexico-structural para-phrasing engine. In Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (NLPRS 2001) Workshop on Automatic Paraphrasing: Theories and Applications, Tokyo, Japan, 27–30 November 2001; pp. 37–46.
5. Inui, K.; Fujita, A.; Takahashi, T.; Iida, R.; Iwakura, T. Text Simplification for Reading Assistance: A Project Note. In Proceedings of the Second International Workshop on Paraphrasing-Volume 16, Sapporo, Japan, 11 July 2003; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; PARAPHRASE '03, pp. 9–16. [CrossRef]
6. Scarton, C.; de Oliveira, M.; Candido, A.; Gasperin, C.; Aluísio, S.M. SIMPLIFICA: A tool for authoring simplified texts in Brazilian Portuguese guided by readability assessments. In Proceedings of the NAACL HLT 2010 Demonstration Session, Los Angeles, CA, USA, 2 June 2010.
7. Aluisio, S.; Gasperin, C. PorSimples: Simplification of Portuguese Texts Fostering Digital Inclusion and Accessibility. In Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, Los Angeles, CA, USA, 6 June 2010.
8. Shardlow, M. A Survey of Automated Text Simplification. *Int. J. Adv. Comput. Sci. Appl.* **2014**, *4*, 58–70. [CrossRef]
9. Saggion, H.; Gómez-Martínez, E.; Etayo, E.; Anula, A.; Bourg, L. Text Simplification in Simplext. Making Text More Accessible. *Proces. Leng. Nat.* **2011**, *47*, 341–342.
10. Scarton, C.; Specia, L. Learning simplifications for specific target audiences. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 712–718.
11. Xu, W.; Callison-Burch, C.; Napoles, C. Problems in Current Text Simplification Research: New Data Can Help. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 283–297. [CrossRef]
12. Nishihara, D.; Kajiwara, T.; Arase, Y. Controllable text simplification with lexical constraint loss. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence, Italy, 28 July–2 August 2019; pp. 260–266.
13. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
14. Xu, W.; Carpuat, M. EDITOR: An edit-based transformer with repositioning for neural machine translation with soft lexical constraints. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 311–328. [CrossRef]
15. Agrawal, S.; Xu, W.; Carpuat, M. A non-autoregressive edit-based approach to controllable text simplification. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 3757–3769.
16. Štajner, S. Automatic text simplification for social good: Progress and challenges. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 2637–2652.
17. Van, H.; Kauchak, D.; Leroy, G. AutoMeTS: The autocomplete for medical text simplification. *arXiv* **2020**, arXiv:2010.10573.
18. Van den Bercken, L.; Sips, R.J.; Lofi, C. Evaluating neural text simplification in the medical domain. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3286–3292.
19. Cardon, R.; Grabar, N. French biomedical text simplification: When small and precise helps. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020.
20. Collantes, M.; Hipe, M.; Sorilla, J.L.; Tolentino, L.; Samson, B. Simpatico: A text simplification system for senate and house bills. In Proceedings of the 11th National Natural Language Processing Research Symposium, Manila, Philippines, 24–25 April 2015; pp. 26–32.
21. Bhatia, V.K. Simplification v. easification—The case of legal texts1. *Appl. Linguist.* **1983**, *4*, 42–54. [CrossRef]

22. Garimella, A.; Sancheti, A.; Aggarwal, V.; Ganesh, A.; Chhaya, N.; Kambhatla, N. Text Simplification for Legal Domain:{I} nsights and Challenges. In Proceedings of the Natural Legal Language Processing Workshop, Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 296–304.
23. Rubab, I. Investigating the Effect of Text Simplification to Speed the Justice in Pakistan. Ph.D. Thesis, Islamia University, Bahawalpu, Pakistan, 2018.
24. Zhang, X.; Lapata, M. Sentence Simplification with Deep Reinforcement Learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 584–594. [CrossRef]
25. Dong, Y.; Li, Z.; Rezagholizadeh, M.; Cheung, J.C.K. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. *arXiv* **2019**, arXiv:1906.08104.
26. Lin, X.W.Z.; Wan, X. Neural sentence simplification with semantic dependency information. In Proceedings of the AAAI Workshop on Deep Learning on Graphs: Methods and Applications, Virtual, 28 February 2021.
27. Jiang, C.; Maddela, M.; Lan, W.; Zhong, Y.; Xu, W. Neural CRF model for sentence alignment in text simplification. *arXiv* **2020**, arXiv:2005.02324.
28. Kincaid, J.P.; Fishburne, R.P.J.; Rogers, R.L.; Chissom, B.S. *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*; Technical Report; Institute for Simulation and Training, University of Central Florida: Millington, TN, USA, 1975.
29. Gunning, R. The Fog Index After Twenty Years. *J. Bus. Commun.* **1969**, *6*, 3–13.
30. Aluisio, S.; Specia, L.; Gasperin, C.; Scarton, C. Readability assessment for text simplification. In Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, Los Angeles, CA, USA, 5 June 2010; pp. 1–9.
31. Bessou, S.; Chenni, G. Efficient Measuring of Readability to Improve Documents Accessibility for Arabic Language Learners. *arXiv* **2021**, arXiv:2109.08648.
32. Marvin Imperial, J.; Ong, E. Under the Microscope: Interpreting Readability Assessment Models for Filipino. *arXiv* **2021**, arXiv:2110.00157.
33. Yeakel, K.; Tzeng, S. *Autograder: Classifying Documents to Grade School Level*; Stanford University: Stanford, CA, USA, 2019.
34. Štajner, S.; Ponzetto, S.P.; Stuckenschmidt, H. Automatic assessment of absolute sentence complexity. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; Volume 17, pp. 4096–4102.
35. Larson, R.R. Introduction to Information Retrieval. *J. Am. Soc. Inf. Sci. Technol.* **2010**, *61*, 852–853.
36. Giovanelli, C.; Liu, X.; Sierla, S.; Vyatkin, V.; Ichise, R. Towards an aggregator that exploits big data to bid on frequency containment reserve market. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 7514–7519. [CrossRef]
37. Li, H. Deep learning for natural language processing: Advantages and challenges. *Natl. Sci. Rev.* **2018**, *5*, 24–26. [CrossRef]
38. Alkaldi, W.; Inkpen, D. Classifying Documents to Multiple Readability levels. In Proceedings of the AAAI 2021 Spring Symposium on Artificial Intelligence for K-12 Education, Virtual, 22–24 March 2021.
39. Sojasingarayar, A. Seq2Seq AI Chatbot with Attention Mechanism. *arXiv* **2020**, arXiv:2006.02767. Available online: http://xxx.lanl.gov/abs/2006.02767 (accessed on 1 January 2021).
40. Yang, S.; Yu, X.; Zhou, Y. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. In Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 12–14 June 2020; pp. 98–101. [CrossRef]
41. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025.
42. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
43. Alva-Manchego, F.; Martin, L.; Scarton, C.; Specia, L. 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*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 49–54. [CrossRef]
44. Xu, W.; Napoles, C.; Pavlick, E.; Chen, Q.; Callison-Burch, C. Optimizing Statistical Machine Translation for Text Simplification. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 401–415. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.