ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200342

Deciding *When*, *How* and *for Whom* to Simplify

Carolina Scarton[1], Pranava Madhyastha[2], Lucia Specia[3]

Abstract.

Current Automatic Text Simplification (TS) work relies on sequence-to-sequence neural models that learn simplification operations from parallel complex-simple corpora. In this paper we address three open challenges in these approaches: (i) avoiding unnecessary transformations, (ii) determining which operations to perform, and (iii) generating simplifications that are suitable for a given target audience. For (i), we propose joint and two-stage approaches where instances are marked or classified as simple or complex. For (ii) and (iii), we propose fusion-based approaches to incorporate information on the target grade level as well as the types of operation to perform in the models. While grade-level information is provided as metadata, we devise predictors for the type of operation. We study different representations for this information as well as different ways in which it is used in the models. Our approach outperforms previous work on neural TS, with our best model following the two-stage approach and using the information about grade level and type of operation to initialise the encoder and the decoder, respectively.

1 Introduction

Text Simplification (TS) is a text-to-text transformation task where the goal is to generate a simpler version of an original text by applying several operations to it. Such operations include word changes (e.g. a complex word being replaced by a simpler synonym) and/or syntactic transformations (e.g. a long sentence is split into two). Strategies for TS differ depending on either a given target audience or an application. Previous work has explored lexical simplification for specific target audiences, such as non-native speakers [14], and for improving other Natural Language Processing applications, such as machine translation (MT) [10].

Modern TS systems rely on parallel data with original-simplified pairs in order to learn simplification operations. The operations learnt can perform both lexical and syntactic operations together, in a single pass. Inspired by work on MT, state-of-the-art approaches use sequence-to-sequence (s2s) neural models with attention [13, 31, 7, 28, 21, 25, 32].

Two large-scale corpora exist for TS in English: Wikipedia-Simple Wikipedia (W-SW) [33] and the Newsela Corpus.[4] W-SW has pairs of original and simplified Wikipedia articles, with simplified versions generated by volunteers, without targeting a specific audience. Newsela contains original articles from the news domain and their professionally simplified versions. Each simplified version targets a specific US grade level. Previous work has shown that Newsela is a more reliable corpus than W-SW [29, 20]. Both W-SW and Newsela are aligned at article level, while data-driven approaches rely on sentence-level alignments.[5] Different approaches have been proposed to automatically align instances from this dataset [15, 27].

Given instance-level automatic alignments, four general types of operations can be defined:

- Identical: The simplified instance is an exact copy of the original instance.
- Elaboration: The simplified instance has the same number of sentences as the original, but with different content.
- One-to-many (split): The simplified instance has more sentences than the original.
- Many-to-one (merge): The simplified instance has fewer sentences than the original.

Data-driven models built using the Newsela dataset disregard useful information provided with the simplifications. The only exception is presented by Scarton and Specia [21], which uses the grade-level information as well as the type of simplification operation as artificial tokens to build s2s neural models (see Section 2). They show that the use of artificial tokens improves over a baseline system.

In this paper we further explore both given (meta) and inferred information from the corpus, while focusing on how to accurately *predict* such information, how to *represent* it and how to *integrate* it in the s2s models. In addition, we attempt to identify instances that do not require simplification in order to avoid unnecessary (spurious) transformations and further improve the performance. More specifically, our approaches address the following challenges in TS:

Learning *when* to simplify: We devise classifiers to filter out cases that should not be simplified and propose a fully automated two-stage approach.

Learning *how* to simplify: We propose methods to predict, represent and integrate information on operations into s2s architectures using fusion techniques.

Learning *for whom* to simplify: We propose different ways to fuse grade-level meta-information into the models.

We describe previous work that uses neural s2s approaches to TS in Section 2. Section 3 details the Newsela data, its automatic alignments and the neural models used in our experiments. In Section 4 we introduce our approach to determine when to simplify. Section 5 presents our operation classifiers and TS models that use inferred operation types and grade level information. In Section 6 we describe a two-stage approach guided by binary decisions on when to simplify.

[1] University, of Sheffield, UK; email: c.scarton@sheffield.ac.uk
[2] Imperial College London, UK; email: pranava@imperial.ac.uk
[3] Imperial College London, UK; email: l.specia@imperial.ac.uk
[4] https://newsela.com/data, v.2016-01-29.

[5] Contrary to MT, where sentence alignments are generally only one-to-one, other types of alignments naturally exist in TS, namely one-to-many (split) and many-to-one (merge). Therefore, we refer to the unit of simplification as *instances*.

2 Related work

Nisioi et al. [13] build an attention-based s2s model for TS using the W-SW dataset. BLEU [16] or SARI [30] (see Section 3) are used to find the best beam size to perform beam search. Their best system outperforms previous approaches according to BLEU. Zhang and Lapata [31] also propose an s2s model with reinforcement learning and evaluate their system in both W-SW and Newsela. SARI, BLEU and cosine similarity were used as reward policies. For the Newsela corpus they outperform previous work, while they are behind Narayan and Gardent [12] for W-SW.

Vu et al. [28] experiment with neural semantic encoders [11] for TS. This approach creates a memory matrix for each encoding time step. This gives the model unrestricted access to the entire source instance, which allows the encoder to attend to other relevant words when encoding a given word. Their models do not outperform previous work according to SARI, but they show better results on human judgements of fluency, adequacy and simplicity. Guo et al. [7] propose a model that uses both attention and pointer-copy mechanisms [22]. They also experiment with multi-task learning, with entailment and paraphrasing generation as auxiliary tasks. Their models show considerable gains in SARI when compared to previous work on Newsela and W-SW datasets.

Sulem et al. [25] use semantic information from UCCA (Universal Cognitive Conceptual Annotation) [1] to build a system to perform instance splitting. These instances are then fed into an s2s model for further simplification. Although they show improvements for perceived simplicity over the NTS system, surprisingly, they do not experiment with corpora in which splitting appears in the gold simplifications.

Zhao et al. [32] integrate the Transformer s2s architecture [26] with the Simple PPDB [17] paraphrase dataset in two ways: (i) by adding a loss function that applies the rules present in the PPDB, (ii) by using an augmented dynamic memory that stores key-value pairs for each PPDB rule, where keys are a weighted average of encoder hidden states and current decoder hidden states and values are the ouptut vectors. Their approaches marginally outperform previous work on W-SW and Newsela datasets.

These previous research neither targets simplifications to a specific audience, nor uses explicit information about the type of simplification operations. The only exception is the work of Scarton and Specia [21], where they experiment with attention-based s2s models with information about the grade level and simplification operations added as artificial tokens to the source (complex) instances (following Johnson et al. [8]). Models using artificial tokens perform better, only when using oracle simplification operations. However, their approach to integrate metadata and inferred information is very different from ours. Tokens representing grade levels and tokens representing type of operations are simply put together as a single token and used as the first token in the source instances. This makes the data sparse and does not allow the exploitation of information in different parts of the s2s architecture. In addition, the performance of their operation classifiers is considerably low, which means that they harm the performance of the TS models when used.

In this paper: (i) we explore different ways of adding metadata (grade level) and/or inferred (simplification operations) information in the s2s architecture, (ii) we propose more robust classifiers for the task of inferring simplification operations, and (iii) we analyse models trained only with instance pairs where original and simplified versions are different from each other. This allows us to simulate a scenario where the user has control of deciding the instances (s)he wants to simplify. We also experiment with predicting whether or not an instance should be simplified before sending it to the TS model.

3 Data and neural models

Data The Newsela dataset (version 2016-01-09) contains $1,911$ original articles with up to five simplified versions each. These simplified versions are produced by professionals and target a specific grade level that can vary from 2 to 12 (lower values mean simpler articles). These articles need to be segmented into smaller units, and instance-level alignments need to be created, in order to use them into s2s models.

For our experiments, instance alignments are generated using the vicinity-driven search approach by Paetzold and Specia [15]. This is an unsupervised approach which allows for long-distance alignment (e.g. skipping sentences) and captures 1-to-n and n-to-1 alignments.

Alignments are generated from the original for each simplified article version and among all simplified versions. Therefore, if an original article 0 is aligned with up to four simplified versions (1, 2, 3 and 4), the alignments are extracted between 0-$\{$1,2,3,4$\}$, 1-$\{$2,3,4$\}$, 2-$\{$3,4$\}$ and 3-4. This method generates $550,644$ instance pairs (11M original tokens and 10M target tokens). Using these alignments, four types of simplification operation can be identified (as presented in Section 1). Table 1 shows the number of instances per type of operation in the Newsela dataset. From these instance pairs we randomly select training ($440,516$ instances), development ($2,000$ instances) and test ($55,064$ instances) sets (`FULL` dataset). For the development set, instances are selected to keep the four operation classes balanced. Therefore, this set has 500 instances pairs of each type.

Operation	Count	Proportion
Identical	146,151	26.54%
Elaboration	266,870	48.47%
Split	119,241	21.65%
Merge	18,382	3.34%

Table 1. Number and proportion of instances per type of operation in the Newsela corpus

Architecture Our models are based on s2s neural models with attention [4] and conditional gated recurrent units (CGRUs) [6] as decoder. As encoder we use a bi-directional recurrent GRU, followed by the decoder: a CGRU which is initialised with a non-linear transformation of the mean of encoder states. As attention mechanism, we use a simple feedforward network. Our implementation is based on the NMTpy toolkit [5].[6] We use the Adam optimiser with a learning rate of 0.0004 and batch size of 32. Encoder and decoder embedding dimensionality is 500, the maximum target instance length is 250, dropout $= 0.5$, with a fixed seed for all models. Model selection is performed according to SARI on the validation set.

Fusion techniques and representations Our work is related to work on multimodal MT [23], which shows that an encoder or decoder initialised with external information (image-related, in their case) helps the s2s model provide better informed translations. We test the following configurations to inform s2s models on the type of simplification operations or grade level:

[6] `https://github.com/lium-lst/nmtpytorch`

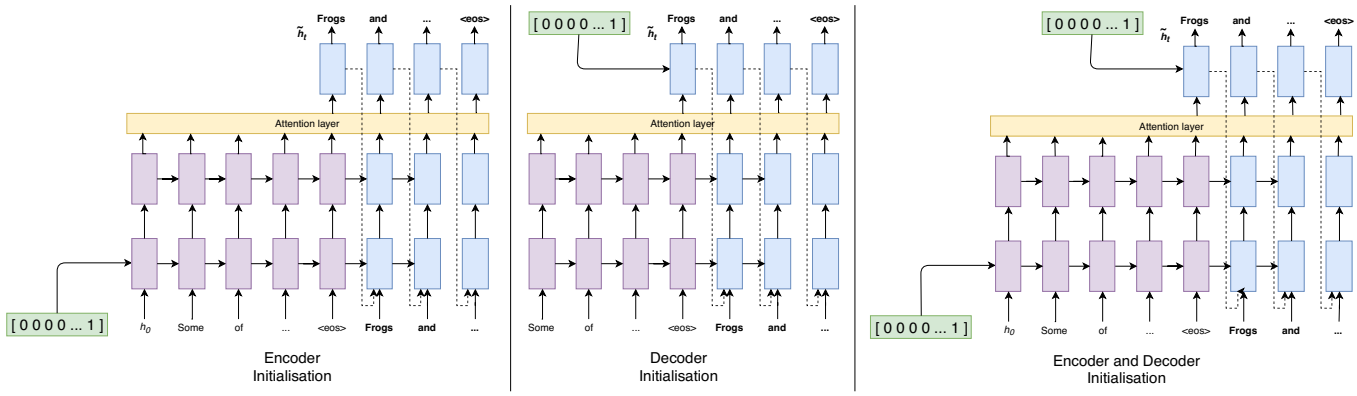

Figure 1. Neural model architecture illustrating encoder and decoder initialisation

- informing the encoder,
- informing the decoder,
- informing both encoder and decoder.

We investigate two ways of incorporating the metadata and/or inferred information into our models: (i) adding an artificial token to the beginning of the original instances, and (ii) using one-hot vectors with the metadata or inferred information to initialise the encoder and/or decoder. In (i), the NMT architecture is not modified, i.e. only the data needs manipulation, while in (ii), the architecture is modified to allow for the initialisation of the components by external vectors (see Figure 1).

Evaluation We assess the ability of the s2s approach to exploit the simplification operations by evaluating the output of the models using SARI. SARI is an n-gram-based metric that compares a system output with references and the original text. As it correlates highly with human scores for simplicity, we use it as a proxy for this aspect of simplification quality [30]. Although it is common to report BLEU scores, we believe this metric is misleading: BLEU score can increase if target instances repeat various words from the original instance, but not necessarily on the expected order of the reference simplification [20]. Similar observations are made in [24]. Therefore, *identical* instances output by simplification systems can show higher BLEU scores, even though the reference simplification is identical to the original(a discussion about using BLEU for TS is presented in the Supplementary Material)..

4 Deciding *when* to simplify

As shown in Table 1, not all instances need simplification. In fact, over one quarter of the Newsela instances are not simplified between two given levels. While in principle the s2s models can learn *how* to make this decision, we observed that the models trained with the entire dataset are more prone to keep instances intact than to transform them. In addition, having such instances at test time may lead to unnecessary (*potentially spurious*) transformations. In this section we study the performance of models with and without cases of identical instances. At this stage we assume a practical scenario where deciding when to simplify is performed by a user (e.g. the user selects complex instances). In Section 6 we also show our experiments with binary classifiers built to automate this decision.

Our experiments compare two baseline models built using either the entire dataset (FULL) or only instances that should be simplified according to our oracle (SIMP-ONLY). SIMP-ONLY has 323,790 instance pairs for training, 1,500 pairs for development and 40,309 pairs for test. The baseline models are a re-implementation of two models from [21]:

- nmt: baseline model trained without artificial tokens,
- nmt+<gl>: baseline model trained with an artificial token for grade level, their best model.

	FULL	SIMP-ONLY
nmt	35.71	42.91
nmt+<gl>	39.94	**46.48**

Table 2. SARI of simplification models when using either FULL or SIMP-ONLY – best results in bold

Models trained using the SIMP-ONLY subset show substantially higher SARI, confirming our hypothesis.

We observe the same trend as in [21]: the model using grade levels as artificial tokens outperforms the baseline model (nmt) by a large margin.[7]

Table 3 shows examples of instances that were either over or under-simplified by nmt+<gl> models using the FULL dataset, but were correctly simplified by the same model built with SIMP-ONLY. The first example was marked to be kept as is (*identical*), however, the system built with FULL over-simplified it (the highlighted in red in the column FULL). The second example was marked to be split and, while the system built with SIMP-ONLY performs the correct operation (highlighted in blue in the column SIMP-ONLY), the system built with FULL copies the original instance.

Table 4 shows an analysis on the FULL models assessed in the FULL test set in order to identify the number of instances affected by having identical instances in the training data. Around 30% of the instances were oversimplified (instances that should be kept without simplification, but were instead simplified by the models) and around 60% of the instances were under-simplified (instances that should

[7] We tried our best to use a configuration of NMTpy as close as possible to the OpenNMT-based configuration in [21]. Data split, as well as architectural and implementation differences may be the reason behind the deviations in performance – e.g. [21] use LSTMs while we use GRU.

Operation	Original	FULL	SIMP-ONLY
identical	Patty Murray said **she was ready for questions** about the cost of her summer hunger program.	Patty Murray said **she was ready for the cost of her summer hunger program.**	Patty Murray said **she was ready for questions** about the cost of her summer hunger program.
split	He also sits on an aerospace workforce training committee **and** said that most other Washington state suppliers in his industry have been seeing the same problem.	He also sits on an aerospace workforce training committee **and** said that most other Washington state suppliers in his industry have been seeing the same problem.	He also is on an aerospace training committee**. He** said that most other Washington state suppliers in his industry have been seeing the same problem.

Table 3. Examples of instances simplified by different models trained either on FULL or on SIMP-ONLY data. The fist example (*identical*) the original instance should be kept as is, however, the FULL model attempted to simplify and removed important information. On the second example, a *split* operation should be applied, however, the FULL model outputted the original instance. On the other hand, SIMP-ONLY performs the correct simplification

be simplified but were kept the same as the original). In summary, out of 55,064 instances in the FULL test set, models performed the incorrect operation due to the identical instances in 49.42% of the cases for nmt and 55.76% of the cases for nmt+<gl>.

	Oversimplified	Under-simplified
nmt	4,031 (27.32%)	23,184 (57.52%)
nmt+<gl>	3,839 (26.00%)	26,873 (66.67%)

Table 4. Analysis of oversimplification and under-simplification for the models evaluated on the FULL test set

In the following sections we explore other ways of deciding whether or not an instance should be simplified without pre-filtering. Since one of the simplification operations is *identical*, we propose models that include this information as one of the possible operations.

5 Learning *how* to simplify

As mentioned previously, grade level is given at both training and test time, while the types of operation would have to be predicted at test time. Here we experiment with different classifiers for predicting such operations (Section 5.1) and different ways of adding metadata and inferred information in the neural models (Section 5.2).

5.1 Classification of operations

Classifiers are trained using our Newsela training set evaluated in the test set with both the FULL dataset and SIMP-ONLY. We experiment with two different approaches for building *type of operation* classifiers:

Instance embeddings Following Arora et al. [3], instance embeddings are extracted as follows and used as features:

1. importance weights for words are computed by first obtaining word frequency over our corpus as:

$$weight_{w_i} = \frac{0.001}{0.001 + (\frac{freq(w_i)}{\sum_0^n freq(w_n)})},$$

   where n is the total number of words,
2. for each word, the corresponding pre-trained word embeddings is multiplied by its importance weight,
3. for each instance, an embedding is computed as the weighted average of word embeddings for all words in the instance, and
4. the first principle component is removed from the space.

From these representation, we build classifiers with the random forest (RF) algorithm, using the scikit-learn toolkit [18].[8]

BCN+ELMo Following the work of Peters et al. [19], we build a classifier using the ELMo vectors. We adapted the biattentive classification network (BCN) model [9] using the AllenNLP toolkit[9] and trained a classification model with a learning rate of 0.0001.

5.1.1 *Classifiers trained on FULL*

Table 5 (top part) shows the performance of our operation type classifiers for the four-class scenario in our development set. Results for a baseline (random classifier – RAND) are also shown. The best performing classifier is BCN+ELMo with instance embeddings as features, which is considerably higher than the RAND baseline. Figure 2(a) shows the normalised confusion matrix for our four-class classifiers. RF is better at predicting *elaboration* (E) and *identical* (I) operations than the other two classifiers, while BNC+ELMo is better at predicting *split* and *merge*.

	Accuracy	F1	Precision	Recall
Four-class classifiers				
RF	0.482	0.513	0.648	0.482
BCN+ELMo	**0.706**	**0.703**	**0.710**	**0.706**
RAND	0.258	0.258	0.258	0.258
Three-class classifiers				
RF	0.511	0.576	0.775	0.511
BCN+ELMo	**0.809**	**0.812**	**0.832**	**0.809**
RAND	0.336	0.336	0.336	0.336
Binary classifiers				
RF	0.822	0.886	0.922	0.852
BCN+ELMo	**0.833**	**0.893**	**0.927**	**0.861**
RAND	0.495	0.528	0.623	0.495

Table 5. Performance of our four-class, three-class and binary classifiers evaluated on the development set

[8] We also experimented with other algorithms and vector representations, but RF with the instance embeddings described above was the best performing classifier.

[9] https://allennlp.org/elmo

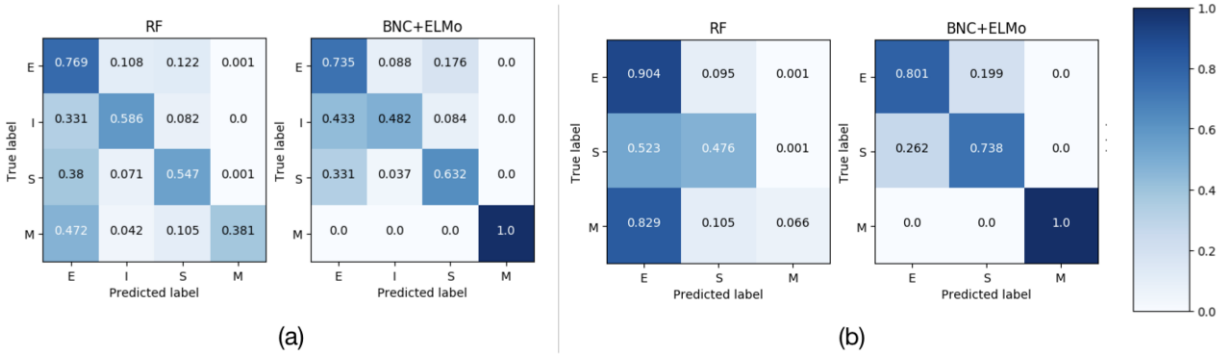

Figure 2. Normalised confusion matrix of (a) the four-class classifiers and (b) the three-class classifiers (evaluated on the test set)

5.1.2 *Classifiers trained on* SIMP−ONLY

Table 5 (middle part) shows the performance of our operation type classifiers for the three-class scenario (all operations but *identical*. In this case, the best classifier is BCN+ELMo. Figure 2(b) shows the normalised confusion matrix for the three-class scenario. The BCN+ELMo model performs better in this case, mainly because the *identical* operation, i.e. the operation that this classifier could not predict well, is not present. RF is not reliable at classifying *split* and *merge* (M) instances, even though it is still the best at classifying *elaboration* instances. However, as show in Figure 2(a), RF's good performance on the four-class classifier is mainly due to its performance on the *identical* class, which is not present here.

5.2 Guiding TS models

In this section we show different ways of adding metadata and inferred information in our TS models.

Baselines Baseline models are [21]:

- nmt and nmt+<gl> (see Section 4),
- nmt+<gl+op>: model trained with artificial tokens for grade level and operations and tested with either oracle or predicted operations (RF or BCN+ELMo).

Fusion systems One-hot vectors are generated from the output of each classifier to represent the predicted operations. Each position in these vectors corresponds to one specific class. For instance, to represent the elaboration class, the vector would be $[1, 0, 0, 0]$, while to represent the identical class the vector would be $[0, 1, 0, 0]$. The following models are proposed:[10]

- nmt+<gl>+Dec(op): model with grade-level artificial tokens and decoder initialisation using type of operation vectors (oracle or predicted),
- nmt+Enc(gl)Dec(op): model with encoder and decoder initialisation. The encoder is initialised using the grade level vectors, while the decoder is initialised using simplification operation vectors (oracle or predicted).

In all operation-informed models, the oracle operations are used for training and development.

Table 6 shows the results according to SARI for three subsets of the data: (i) FULL evaluated on the entire test set, (ii) FULL evaluated on the SIMP−ONLY test set and (iii) SIMP−ONLY (training and test). Models built with encoder initialisation for grade level and decoder initialisation for type of operations perform the best for all data splits. As expected, in all cases the oracle information about types of operation leads to the best results. For the SIMP−ONLY scenario, models with encoder and decoder initialisation outperform the baseline even when tested on predicted operations. Initialising both the encoder and the decoder with one-hot vectors leads to better results than using a hybrid approach (nmt+gl+Dec(op)).[11]

The best model overall is the nmt+Enc(gl)+Dec(op) built using SIMP−ONLY data. In this case according to SARI, the difference between using the oracle operations and the BCN+ELMo predicted operations is marginal (48.04 vs. 48.69 in SARI), which indicates that BCN+ELMo would be the best classifier to be used in practice.

6 A two-stage approach

In Section 4 we show the advantages of training TS models without identical alignments. In this section, we propose a fully automated two-stage approach: instances to be simplified are selected and then sent to a TS model built with SIMP−ONLY data. For instance selection we devise binary classifiers (*simplify* vs. *do not simplify*), using the same features and methods presented in Section 5.1. Table 5 (bottom part) shows the accuracy, F1 score, precision and recall for these classifiers. Binary prediction appears to be a simpler problem, since the performance of our classifiers increases consistently.

We experiment with both BCN+ELMo and RF binary classifiers to perform instance selection for our two-stage approach. The RF model predicts $43,389$ instances to be simplified, whilst BCN+ELMo predicts $42,379$ instances to be simplified (out of $55,064$ of the test set). We then gave the selected instances to our best three-class classifier (also BCN+ELMo) to get a prediction on the type of simplification to be performed in each instance. Finally, the test instances with information on grade level and type of operation are fed into the best model built with SIMP−ONLY data (nmt+Enc(gl)+Dec(BCN+ELMo) – Table 6). The results, shown in Table 7, are not as good as those with oracle filtering

[10] Due to space constraints we do not show results for models built using grade-level artificial tokens and encoder initialisation with types of operation vectors. Such models did not outperform nmt+Enc(gl)Dec(op) models.

[11] Examples of the outputs of our TS systems can be found in the Supplementary Material.

		FULL FULL test	FULL SIMP-ONLY test	SIMP-ONLY
Baselines	nmt	34.59	37.51	42.91
	nmt+gl	37.76	39.94	46.48
Artificial tokens	nmt+gl+oracle	41.66	45.16	45.42
	nmt+gl+RF	39.71	42.80	43.96
	nmt+gl+BCN+ELMo	40.10	43.41	45.87
Artificial tokens and decoder initialisation	nmt+gl+Dec(oracle)	40.70	43.85	44.52
	nmt+gl+Dec(RF)	35.27	36.82	40.67
	nmt+gl+Dec(BCN+ELMo)	35.76	37.43	46.00
Encoder and Decoder initialisation	nmt+Enc(gl)+Dec(oracle)	**42.17**	**45.85**	**48.69**
	nmt+Enc(gl)+Dec(RF)	35.47	37.22	47.17
	nmt+Enc(gl)+Dec(BCN+ELMo)	36.00	37.80	<u>48.04</u>

Table 6. SARI of TS models: best models are in bold (the best model with predicted operation types is underlined)

(nmt+Enc(gl)+Dec(oracle) SARI = 48.69 in Table 6). Nevertheless, this is still a very competitive approach, especially given it is fully automated, mainly when compared to the baselines nmt and nmt+gl. In BCN+ELMo setting, when comparing to nmt+gl, the best model with predicted operations improves over 3 SARI points, whilst the improvements seem with RF setting is over 6 SARI points.

This can be explained because the BCN+ELMo only showed minor improvements over the RF classifier in the binary setting (bottom of Table 5). In addition, when looking for results in the test set, the RF classifier (Accuracy = 0.829 and F1 = 0.887) outperforms BCN+ELMo (Accuracy = 0.805 and F1 = 0.871), which can explain the performance in Table 7.

BCN+ELMo	
nmt	35.65
nmt+gl	39.40
nmt+Enc(gl)+Dec(BCN+ELMo)	**42.65**
RF	
nmt	35.44
nmt+gl	39.16
nmt+Enc(gl)+Dec(BCN+ELMo)	**45.50**

Table 7. SARI of the two-stage TS approach

- nmt+Dec(gl): only initiliases the decoder with grade-level vectors,
- nmt+Enc(oracle)+Dec(gl): initialises the encoder using oracle type of operation vectors and the decoder using grade level vectors.

Table 8 shows SARI results for the above systems and some of the previously presented models. Grade-level vectors for initialising either the encoder or the decoder perform better than using artificial tokens for both test sets. There seems to be no significant difference between initialising the encoder or the decoder with grade level vectors, given the performance of nmt+Enc(gl) and nmt+Dec(gl) are virtually the same. However, when also considering the type of operation vectors, adding grade-level vectors into the encoder and type of operation vectors into the decoder results in the best models.

	FULL test	SIMP-ONLY test
nmt+<gl>	37.76	39.94
nmt+Enc(gl)	38.17	40.52
nmt+Dec(gl)	38.31	40.69
nmt+Enc(gl)+Dec(oracle)	**42.17**	**45.85**
nmt+Enc(oracle)+Dec(gl)	41.43	44.85

Table 8. SARI of FULL models adding grade-level information in different ways (best models are in bold)

7 *For whom* to simplify

Section 5.2 shows results for models built using grade-level information to guide the encoder by either adding an artificial token to the original instance or by initialising the encoder with vector representations. In this section we further examine the best way of using the grade-level information.

Our hypothesis is that grade-level information is most useful to guide the encoder, whilst the type of operation is best in guiding the decoder. We test this hypothesis by analysing three new models together with some previously presented models:

- nmt+Enc(gl): only initialises the encoder with grade-level vectors,

8 Examples of TS systems output

Table 9 shows output examples for our best models: nmt+gl+oracle and nmt+Enc(gl)+Dec(oracle), comparing them to the baseline (nmt+gl) and the references on the SIMP-ONLY dataset.

9 Conclusions

In this paper we present an empirical study on ways to guide s2s TS approaches. Our experiments and results for each of the three challenges we identified led us to the following main findings:

Split	
Original	So I've had ample opportunity to interact with many teachers and know from first-hand experience they're not the problem.
Reference	So I've interacted with many teachers. I know from first-hand experience they're not the problem.
nmt+gl	**So I've had plenty to interact with many teachers and know from local experience they're not the problem.**
nmt+gl+oracle	So I've had plenty opportunity to interact with many teachers. I know from first-hand experience they're not the problem.
nmt+Enc(gl)+Dec(oracle)	So I've had plenty to talk with many teachers. I know that they're not the problem.
nmt+Enc(gl)+Dec(BCN+ELMo)	So I've had plenty to talk with many teachers. I know that they're not the problem.
Elaboration	
Original	Since then, Islamic State has demonstrated the capacity to adapt and innovate, combining the most effective terrorist practices honed over the last three decades.
Reference	It is combining the terror practices developed by extremist groups over the last 30 years.
nmt+gl	Since then, Islamic State has demonstrated the capacity to adapt and innovate**. It is** combining the most effective terrorist practices in the last 30 years.
nmt+gl+oracle	Since then Islamic State has demonstrated the capacity to adapt.
nmt+Enc(gl)+Dec(oracle)	Since then, Islamic State has demonstrated the capacity to adapt.
nmt+Enc(gl)+Dec(BCN+ELMo)	Since then, Islamic State has demonstrated the capacity to adapt.
Merge	
Original	All year, scientists have been forecasting an El Niño during which warm ocean water at the equator near South America can affect the weather dramatically. Now the water is only slightly warmer than normal at the equator and scientists say a mild El Niño is on the way, with less dramatic weather effects.
Reference	Now the water is only slightly warmer than normal at the equator, and scientists say a mild El Niño is on the way.
nmt+gl	**All year, scientists have been forecasting an El Niño.**
nmt+gl+oracle	The water is only slightly warmer than normal at the equator and scientists say a mild El Niño is on the way.
nmt+Enc(gl)+Dec(oracle)	The water **is** in the equator near South America can affect the weather dramatically, and scientists say a mild El Niño is on the way
nmt+Enc(gl)+Dec(BCN+ELMo)	The water **is** in the equator near South America can affect the weather dramatically, and scientists say a mild El Niño is on the way

Table 9. Examples of instances simplified by different models using the SIMP-ONLY dataset. Outputs that do not present the type of operation that appears in the reference or contain errors are marked in bold and red.

***When* to simplify** We observed that filtering complex instances that need simplification and then passing these instances on to an s2s model trained without *identical* instances always results in better performance than models trained on the dataset as a whole. This is the case for non-guided models as well as models guided by operations types. Filtering can be done as part of the model or in a two-stage approach, where instances are automatically marked for simplification then sent to a model trained only on *non-identical* instance pairs. Our two-stage approach shows competitive results to those obtained by oracle filtering followed by the same TS model. Although it can be argued that a two-stage approach would introduce noise to the model, it is worth noting that such an approach performs better than not having any filtering.

***How* to simplify** Informing models with simplification operations outperforms the baseline models even when using predicted types of operation, as long as predictions are sufficiently accurate. The best models are built with our proposed fusion approach, which is substantially better than simply adding artificial tokens to the source instance.

***For whom* to simplify** We confirm the findings in [21] about the importance of grade-level information and propose a new method that adds this information via one-hot vectors. Our new fusion approach outperforms the use of artificial tokens.

Future work include exploring more fine-grained classification of operations, such as in [2], and adding types of information, such as the splitting information learnt by from rules extracted by a robust semantic parser as shown in [25].

Acknowledgements

This work was supported by SIMPATICO (H2020-EURO-6-2015, number 692819) and MultiMT (H2020 ERC Starting Grant, number 678017) projects.

REFERENCES

[1] Omri Abend and Ari Rappoport, 'Universal conceptual cognitive annotation (ucca)', in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 228–238. Association for Computational Linguistics, (2013).

[2] Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia, '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)*, pp. 295–305, Taipei, Taiwan, (November 2017). Asian Federation of Natural Language Processing.

[3] Sanjeev Arora, Yingyu Liang, and Tengyu Ma, 'A Simple but Tough-to-Beat Baseline for Sentence Embeddings', in *Proceedings of the 5th International Conference on Learning Representations*, pp. 1–16, Toulon, France, (2017).

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 'Neural machine translation by jointly learning to align and translate', in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, (2015).

[5] Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault, 'Nmtpy: A flexible toolkit for advanced neural machine translation systems', *Prague Bull. Math. Linguistics*, **109**, 15–28, (2017).

[6] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio, 'Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism', in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 866–875, San Diego, CA, (2016). Association for Computational Linguistics.

[7] Han Guo, Ramakanth Pasunuru, and Mohit Bansal, 'Dynamic multi-level multi-task learning for sentence simplification', in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 462–476, Santa Fe, NM, (2018). Association for Computational Linguistics.

[8] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean, 'Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation', *Transactions of the Association for Computational Linguistics*, **5**, 339–351, (2017).

[9] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher, 'Learned in Translation: Contextualized Word Vectors', in *Proceedings of the 31st Conference on Neural Information Processing System*, pp. 6294–6305, Long Beach, CA, (2017). Curran Associates, Inc.

[10] Shachar Mirkin, Sriram Venkatapathy, and Marc Dymetman, 'Confidence-driven Rewriting for Improved Translation', in *In Proceedings of the Machine Translation Summit XIV*, pp. 257–264, Nice, France, (2013).

[11] Tsendsuren Munkhdalai and Hong Yu, 'Neural semantic encoders', in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 397–407, Valencia, Spain, (2017). Association for Computational Linguistics.

[12] Shashi Narayan and Claire Gardent, 'Hybrid simplification using deep semantics and machine translation', in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 435–445, Baltimore, MD, (June 2014). Association for Computational Linguistics.

[13] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu, 'Exploring neural text simplification models', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 85–91, Vancouver, Canada, (July 2017). Association for Computational Linguistics.

[14] Gustavo Henrique Paetzold, *Lexical Simplification for Non-Native English Speakers*, Ph.D. dissertation, University of Sheffield, Sheffield, UK, 2016.

[15] Gustavo Henrique Paetzold and Lucia Specia, 'Vicinity-driven paragraph and sentence alignment for comparable corpora', *CoRR*, **abs/1612.04113**, (2016).

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: A method for automatic evaluation of machine translation', in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318, Philadelphia, PA, (2002). Association for Computational Linguistics.

[17] Ellie Pavlick and Chris Callison-Burch, 'Simple PPDB: A Paraphrase Database for Simplification', in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 143–148, Berlin, Germany, (August 2016). Association for Computational Linguistics.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research*, **12**, 2825–2830, (2011).

[19] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, 'Deep contextualized word representations', in *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, New Orleans, LA, (June 2018). Association for Computational Linguistics.

[20] Carolina Scarton, Gustavo Henrique Paetzold, and Lucia Specia, 'Text Simplification from Professionally Produced Corpora', in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pp. 3504–3510, Miyazaki, Japan, (2018). European Language Resources Association (ELRA).

[21] Carolina Scarton and Lucia Specia, 'Learning Simplifications for Specific Target Audiences', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 712–718, Melbourne, Australia, (2018). Association for Computational Linguistics.

[22] Abigail See, Peter J. Liu, and Christopher D. Manning, 'Get to the point: Summarization with pointer-generator networks', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, (2017). Association for Computational Linguistics.

[23] Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott, 'A shared task on multimodal machine translation and crosslingual image description', in *Proceedings of the First Conference on Machine Translation*, pp. 543–553, Berlin, Germany, (August 2016). Association for Computational Linguistics.

[24] Elior Sulem, Omri Abend, and Ari Rappoport, 'BLEU is Not Suitable for the Evaluation of Text Simplification', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 738–744, Brussels, Belgium, (2018). Association for Computational Linguistics.

[25] Elior Sulem, Omri Abend, and Ari Rappoport, 'Simple and Effective Text Simplification Using Semantic and Neural Methods', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 162–173, Melbourne, Australia, (2018). Association for Computational Linguistics.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *Proceedings of the 31st Conference on Neural Information Processing Systems*, pp. 1–11, Long Beach, CA, (2017).

[27] Sanja Štajner, Marc Franco-Salvador, Simone Paolo Ponzetto, Paolo Rosso, and Heiner Stuckenschmidt, 'Sentence alignment methods for improving text simplification systems', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 97–102, Vancouver, Canada, (July 2017). Association for Computational Linguistics.

[28] Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu, '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)*, pp. 79–85, New Orleans, LA, (2018). Association for Computational Linguistics.

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

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

[31] Xingxing Zhang and Mirella Lapata, 'Sentence simplification with deep reinforcement learning', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 595–605, Copenhagen, Denmark, (September 2017). Association for Computational Linguistics.

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

[33] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych, 'A monolingual tree-based translation model for sentence simplification', in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1353–1361, Beijing, China, (2010). Association for Computational Linguistics.