Complexity-Weighted Loss and Diverse Reranking for Sentence Simplification

Reno Kriz*, João Sedoc*, Marianna Apidianaki△, Carolina Zheng*, Gaurav Kumar*, Eleni Miltsakaki†, and Chris Callison-Burch*

* Computer and Information Science Department, University of Pennsylvania
△ LIMSI, CNRS, Université Paris-Saclay, 91403 Orsay & LLF, Univ. Paris Diderot
† Choosito, Inc.
{rekriz,joao,gauku,carzheng,ccb}@seas.upenn.edu,
marianna@limsi.fr, eleni@choosito.com

Abstract

Sentence simplification is the task of rewriting texts so they are easier to understand. Recent research has applied sequence-to-sequence (Seq2Seq) models to this task, focusing largely on training-time improvements via reinforcement learning and memory augmentation. One of the main problems with applying generic Seq2Seq models for simplification is that these models tend to copy directly from the original sentence, resulting in outputs that are relatively long and complex. We aim to alleviate this issue through the use of two main techniques. First, we incorporate content word complexities, as predicted with a leveled word complexity model, into our loss function during training. Second, we generate a large set of diverse candidate simplifications at test time, and rerank these to promote fluency, adequacy, and simplicity. Here, we measure simplicity through a novel sentence complexity model. These extensions allow our models to perform competitively with state-of-the-art systems while generating simpler sentences. We report standard automatic and human evaluation metrics.[1]

1 Introduction

Automatic text simplification aims to reduce the complexity of texts and preserve their meaning, making their content more accessible to a broader audience (Saggion, 2017). This process can benefit people with reading disabilities, foreign language learners and young children, and can assist non-experts exploring a new field. Text simplification has gained wide interest in recent years due to its relevance for NLP tasks. Simplifying text during preprocessing can improve the performance of syntactic parsers (Chandrasekar et al., 1996) and

[1]Our code is available in our fork of Sockeye (Hieber et al., 2017) at https://github.com/rekriz11/sockeye-recipes.

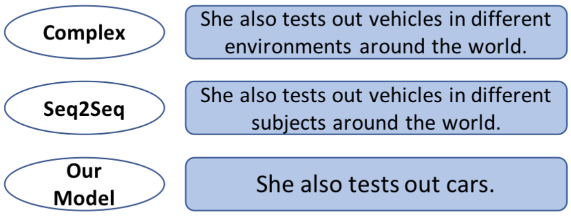

Figure 1: Example comparison of a simplification generated by a standard Seq2Seq model vs. our model.

semantic role labelers (Vickrey and Koller, 2008; Woodsend and Lapata, 2014), and can improve the grammaticality (fluency) and meaning preservation (adequacy) of translation output (Štajner and Popovic, 2016).

Most text simplification work has approached the task as a monolingual machine translation problem (Woodsend and Lapata, 2011; Narayan and Gardent, 2014). Once viewed as such, a natural approach is to use sequence-to-sequence (Seq2Seq) models, which have shown state-of-the-art performance on a variety of NLP tasks, including machine translation (Vaswani et al., 2017) and dialogue systems (Vinyals and Le, 2015).

One of the main limitations in applying standard Seq2Seq models to simplification is that these models tend to copy directly from the original complex sentence too often, as this is the most common operation in simplification. Several recent efforts have attempted to alleviate this problem using reinforcement learning (Zhang and Lapata, 2017) and memory augmentation (Zhao et al., 2018), but these systems often still produce outputs that are longer than the reference sentences. To avoid this problem, we propose to extend the generic Seq2Seq framework at both training and inference time by encouraging the model to choose simpler content words, and by effectively choosing an output based on a large set of candidate simplifications. The main contributions of this paper can be summarized as follows:

- We propose a custom loss function to replace standard cross entropy probabilities during training, which takes into account the complexity of content words.
- We include a similarity penalty at inference time to generate more diverse simplifications, and we further cluster similar sentences together to remove highly similar candidates.
- We develop methods to rerank candidate simplifications to promote fluency, adequacy, and simplicity, helping the model choose the best option from a diverse set of sentences.

An analysis of each individual components reveals that of the three contributions, reranking simplifications at post-decoding stage brings about the largest benefit for the simplification system. We compare our model to several state-of-the-art systems in both an automatic and human evaluation settings, and show that the generated simple sentences are shorter and simpler, while remaining competitive with respect to fluency and adequacy. We also include a detailed error analysis to explain where the model currently falls short and provide suggestions for addressing these issues.

2 Related Work

Text simplification has often been addressed as a monolingual translation process, which generates a simplified version of a complex text. Zhu et al. (2010) employ a tree-based translation model and consider sentence splitting, deletion, reordering, and substitution. Coster and Kauchak (2011) use a Phrase-Based Machine Translation (PBMT) system with support for deleting phrases, while Wubben et al. (2012) extend a PBMT system with a reranking heuristic (PBMT-R). Woodsend and Lapata (2011) propose a model based on a quasi-synchronous grammar, a formalism able to capture structural mismatches and complex rewrite operations. Narayan and Gardent (2014) combine a sentence splitting and deletion model with PBMT-R. This model has been shown to perform competitively with neural models on automatic metrics, though it is outperformed using human judgments (Zhang and Lapata, 2017).

In recent work, Seq2Seq models are widely used for sequence transduction tasks such as machine translation (Sutskever et al., 2014; Luong et al., 2015), conversation agents (Vinyals and Le, 2015), summarization (Nallapati et al., 2016), etc. Initial Seq2Seq models consisted of a Recurrent Neural Network (RNN) that encodes the source sentence $\mathbf{x}$ to a hidden vector of a fixed dimension, followed by another RNN that uses this hidden representation to generate the target sentence $\mathbf{y}$. The two RNNs are then trained jointly to maximize the conditional probability of the target sentence given the source sentence, i.e. $P(\mathbf{y}|\mathbf{x})$. Other works have since extended this framework to include attention mechanisms (Luong et al., 2015) and transformer networks (Vaswani et al., 2017).[2] Nisioi et al. (2017) was the first major application of Seq2Seq models to text simplification, applying a standard encoder-decoder approach with attention and beam search. Vu et al. (2018) extended this framework to incorporate memory augmentation, which simultaneously performs lexical and syntactic simplification, allowing them to outperform standard Seq2Seq models.

There are two main Seq2Seq models we will compare to in this work, along with the statistical model from Narayan and Gardent (2014). Zhang and Lapata (2017) proposed DRESS (Deep REinforcement Sentence Simplification), a Seq2Seq model that uses a reinforcement learning framework at training time to reward the model for producing sentences that score high on fluency, adequacy, and simplicity. This work showed state-of-the-art results on human evaluation. However, the sentences generated by this model are in general longer than the reference simplifications. Zhao et al. (2018) proposed DMASS (Deep Memory Augmented Sentence Simplification), a multi-layer, multi-head attention transformer architecture which also integrates simplification rules. This work has been shown to get state-of-the-art results in an automatic evaluation, training on the WikiLarge dataset introduced by Zhang and Lapata (2017). Zhao et al. (2018), however, does not perform a human evaluation, and restricting evaluation to automatic metrics is generally insufficient for comparing simplification models. Our model, in comparison, is able to generate shorter and simpler sentences according to Flesch-Kincaid grade level (Kincaid et al., 1975) and human judgments, and provide a comprehensive analysis using human evaluation and a qualitative error analysis.

[2]For a detailed description of Seq2Seq models, please see (Sutskever et al., 2014).

3 Seq2Seq Approach

3.1 Complexity-Weighted Loss Function

Standard Seq2Seq models use cross entropy as the loss function at training time. This only takes into account how similar our generated tokens are to those in the reference simple sentence, and not the complexity of said tokens. Therefore, we first develop a model to predict word complexities, and incorporate these into a custom loss function.

3.1.1 Word Complexity Prediction

Extending the complex word identification model of Kriz et al. (2018), we train a linear regression model using length, number of syllables, and word frequency; we also include Word2Vec embeddings (Mikolov et al., 2013). To collect data for this task, we consider the Newsela corpus, a collection of 1,840 news articles written by professional editors at 5 reading levels (Xu et al., 2015).[3] We extract word counts in each of the five levels; in this dataset, we denote 4 as the original complex document, 3 as the least simplified re-write, and 0 as the most simplified re-write. We propose using Algorithm 1 to obtain the complexity label for each word w, where l_w represents the level given to the word, and c_{w_i} represents the number of times that word occurs in level i.

Algorithm 1 Word Complexity Data Collection

procedure DATA COLLECTION
  $l_w \leftarrow 4$
  for $i \in \{3, 0\}$ **do**
    if $c_{w_i} \geq 0.7 * c_{w_{i+1}}$ **then**
      if $c_{w_i} \geq 0.4 * c_{w_4}$ **then**
        $l_w \leftarrow i$
  return l_w

Here, we initially label the word with the most complex level, 4. If at least 70% of the instances of this word is preserved in level 3, we reassign the label as level 3; if the label was changed, we then do this again for progressively simpler levels.

As examples, Algorithm 1 labels “pray”, “sign”, and “ends” with complexity level 0, and “proliferation”, “consensus”, and “emboldened” with complexity level 4. We split the data extracted from Algorithm 1 into Train, Validation and Test sets (90%, 5% and 5%, respectively, and use them for

Model	Correlation	MSE
Frequency Baseline	-0.031	1.90
Length Baseline	0.344	1.51
LinReg	**0.659**	**0.92**

Table 1: Pearson Correlation and Overall Mean Squared Error (MSE) of the word-level complexity prediction model (LinReg). Comparison to length-based and frequency-based baselines.

training and evaluating the complexity prediction model. [4]

We report the Mean Squared Error (MSE) and Pearson correlation on our test set in Table 1.[5] We compare our model to two baselines, which predict complexity using log Google n-grams frequency (Brants and Franz, 2006) and word length, respectively. For these baselines, we calculate the minimum and maximum values for words in the training set, and then normalize the values for words in the test set.

3.1.2 Loss Function

We propose a metric that modifies cross entropy loss to upweight simple words while downweighting more complex words. More formally, the probabilities of our simplified loss function can be generated by the process described in Algorithm 2. Since our word complexities are originally from 0 to 4, with 4 being the most complex, we need to reverse this ordering and add one, so that more complex words and non-content words are not given zero probability. In this algorithm, we denote the original probability vector as **CE**, our vocabulary as **V**, the predicted word complexity of a word v as $score_v$, the resulting weight for a word as w_v, and our resulting weights as **SCE**, which we then normalize and convert back to logits.

Here, α is a parameter we can tune during experimentation. Note that we only upweight simple content words, not stopwords or entities.

3.2 Diverse Candidate Simplifications

To increase the diversity of our candidate simplifications, we apply a beam search scoring modification proposed in Li et al. (2016). In standard

[3]Newsela is an education company that provides reading materials for students in elementary through high school. The Newsela corpus can be requested at https://newsela.com/data/

[4]Note that we also tried continuous rather than discrete labels for words by averaging frequencies, but found that this increased the noise in the data. For example, “the” and “dog” were incorrectly labeled as level 2 instead of 0, since these words are seen frequently across all levels.

[5]We report MSE results by level in the appendix.

Algorithm 2 Simplified Loss Function

procedure SIMPLIFIED LOSS
  CE $\leftarrow$ softmax($logits_{CE}$)
  for $v \in \mathbf{V}$ **do**
    $score_v \leftarrow WordComplexity(v)$
    if v is a content word **then**
      $w_v \leftarrow (4 - s_v) + 1$
    else
      $w_v \leftarrow 1$
  $w_v \leftarrow \left(\frac{w_v}{\sum_{v \in V} w_v}\right)^{\alpha}$ for $v \in \mathbf{V}$
  SCE $\leftarrow$ **CE** $\cdot$ **w**
  return SCE

beam search with a beam width of b, given the b hypotheses at time $t-1$, the next set of hypotheses is generated by first selecting the top b candidate expansions from each hypothesis. These $b \times b$ hypotheses are then ranked by the joint probabilities of their sequence of output tokens, and the top b according to this ranking are chosen.

We observe that candidate expansions from a single parent hypothesis tend to dominate the search space over time, even with a large beam. To increase diversity, we apply a penalty term based on the rank of a generated token among the b candidate tokens from its parent hypothesis.

If Y_{t-1}^{j} is the j^{th} top hypothesis at time $t-1$, $j \in [1..b]$, and $y_t^{j,j'}$ is a candidate token generated from Y_{t-1}^{j}, where $j' \in [1..b]$ represents the rank of this particular token among its siblings, then our modified scoring function is as follows (here, δ is a parameter we can tune during experimentation):

$$S(Y_{t-1}^{j}, y_t^{j,j'}) = \log p(y_1^j, \ldots, y_{t-1}^j, y_t^{j,j'} | x) - j' * \delta \quad (1)$$

Extending the work of Li et al. (2016), to further increase the distance between candidate simplifications, we can cluster similar sentences after decoding. To do this, we convert each candidate into a document embedding using Paragraph Vector (Le and Mikolov, 2014), cluster the vector representations using k-means, and select the sentence nearest to the centroids. This allows us to group similar sentences together, and only consider candidates that are relatively more different.

3.3 Reranking Diverse Candidates

Generating diverse sentences is helpful only if we are able to effectively rerank them in a way that promotes simpler sentences while preserving fluency and adequacy. To do this, we propose three

Model	Correlation	MSE
Length Baseline	0.503	3.72
CNN (ours)	**0.650**	**1.13**

Table 2: Pearson Correlation and Overall Mean Squared Error (MSE) for the sentence-level complexity prediction model (CNN), compared to a length-based baseline.

ranking metrics for each sentence i:

- **Fluency** (f_i): We calculate the perplexity based on a 5-gram language model trained on English Gigaword v.5 (Parker et al., 2011) using KenLM (Heafield, 2011).
- **Adequacy** (a_i): We generate Paragraph Vector representations (Le and Mikolov, 2014) for the input sentence and each candidate and calculate the cosine similarity.
- **Simplicity** (s_i): We develop a sentence complexity prediction model to predict the overall complexity of each sentence we generate.

To calculate sentence complexity, we modify a Convolutional Neural Network (CNN) for sentence classification (Kim, 2014) to make continuous predictions. We use aligned sentences from the Newsela corpus (Xu et al., 2015) as training data, labeling each with the complexity level from which it came.[6] As with the word complexity prediction model, we report MSE and Pearson correlation on a held-out test set in Table 2.[7]

We normalize each individual score between 0 and 1, and calculate a final score as follows:

$$score_i = \beta_f f_i + \beta_a a_i + \beta_s s_i \quad (2)$$

We tune these weights (β) on our validation data during experimentation to find the most appropriate combinations of reranking metrics. Examples of improvements resulting from the including each of our contributions are shown in Table 3.

4 Experiments

4.1 Data

We train our models on the Newsela Corpus. In previous work, models were mainly trained on the parallel Wikipedia corpus (PWKP) consisting of paired sentences from English Wikipedia

[6] We respect the train/test splits described in Section 4.1.
[7] We report MSE results by level in the appendix.

and Simple Wikipedia (Zhu et al., 2010), or the extended WikiLarge corpus (Zhang and Lapata, 2017). We choose to instead use Newsela, because it was found that 50% of the sentences in Simple Wikipedia are either not simpler or not aligned correctly, while Newsela has higher-quality simplifications (Xu et al., 2015).

As in Zhang and Lapata (2017), we exclude sentence pairs corresponding to levels 4-3, 3-2, 2-1, and 1-0, where the simple and complex sentences are just one level apart, as these are too close in complexity. After this filtering, we are left with 94,208 training, 1,129 validation, and 1,077 test sentence pairs; these splits are the same as Zhang and Lapata (2017). We preprocess our data by tokenizing and replacing named entities using CoreNLP (Manning et al., 2014).

4.2 Training Details

For our experiments, we use Sockeye, an open source Seq2Seq framework built on Apache MXNet (Hieber et al., 2017; Chen et al., 2015). In this model, we use LSTMs with attention for both our encoder and decoder models with 256 hidden units, and two hidden layers. We attempt to match the hyperparameters described in Zhang and Lapata (2017) as closely as possible; as such, we use 300-dimensional pretrained GloVe word embeddings (Pennington et al., 2014), and Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. We ran our models for 30 epochs.[8]

During training, we use our complexity-weighted loss function, with $\alpha = 2$; for our baseline models, we use cross-entropy loss. At inference time, where appropriate, we set the beam size $b = 100$, and the similarity penalty $\delta = 1.0$. After inference, we set the number of clusters to 20, and we compare two separate reranking weightings: one which uses fluency, adequacy, and simplicity (FAS), where $\beta_f = \beta_a = \beta_s = \frac{1}{3}$; and one which uses only fluency and adequacy (FA), where $\beta_f = \beta_a = \frac{1}{2}$ and $\beta_s = 0$.

4.3 Baselines and Models

We compare our models to the following baselines:

- **Hybrid** performs sentence splitting and deletion before simplifying with a phrase-based machine translation system (Narayan and Gardent, 2014).
- **DRESS** is a Seq2Seq model trained with reinforcement learning which integrates lexical simplifications (Zhang and Lapata, 2017).[9]
- **DMASS** is a Seq2Seq model which integrates the transformer architecture and additional simplifying paraphrase rules (Zhao et al., 2018).[10]

We also present results on several variations of our models, to isolate the effect of each individual improvement. **S2S** is a standard sequence-to-sequence model with attention and greedy search. **S2S-Loss** is trained using our complexity-weighted loss function and greedy search. **S2S-FA** uses beam search, where we rerank all sentences using fluency and adequacy (FA weights). **S2S-Cluster-FA** clusters the sentences before reranking using FA weights. **S2S-Diverse-FA** uses diversified beam search, reranking using FA weights. **S2S-All-FAS** uses all contributions, reranking using fluency, adequacy, and simplicity (FAS weights). Finally, **S2S-All-FA** integrates all modifications we propose, and reranks using FA weights.

5 Results

In this section, we compare the baseline models and various configurations of our model with both standard automatic simplification metrics and a human evaluation. We show qualitative examples where each of our contributions improves the generated simplification in Table 3.

5.1 Automatic Evaluation

Following previous work (Zhang and Lapata, 2017; Zhao et al., 2018), we use SARI as our main automatic metric for evaluation (Xu et al., 2016).[11] Specifically, SARI calculates how often a generated sentence correctly keeps, inserts, and deletes n-grams from the complex sentence, using the reference simple standard as the gold-standard, where $1 \leq n \leq 4$. Note that we do not use

[8] All non-default hyperparameters can be found in the Appendix.

[9] For Hybrid and DRESS, we use the generated outputs provided in Zhang and Lapata (2017). We made a significant effort to rerun the code for DRESS, but were unable to do so.

[10] For DMASS, we ran the authors' code on our data splits from Newsela, in collaboration with the first author to ensure an accurate comparison.

[11] To calculate SARI, we use the original script provided by (Xu et al., 2016).

Complex Sentence	Model 1	Model 1 Sentence	Model 2	Model 2 Sentence
Mary travels between two offices.	S2S	Mary is a professor at the park.	S2S-Loss	Mary goes between two offices.
Their fatigue changes their voices, but they're still on the freedom highway.	S2S	Their condition changes their voices, but they're still on the freedom highway.	S2S-FA	Their fatigue changes their voices.
Just until recently, the education system had banned Islamic headscarves in schools and made schoolchildren recite a pledge of allegiance.	S2S-FA	The education system had banned Islamic law.	S2S-Cluster-FA	Only until recently , the education system had banned Islamic hijab in schools.
Police used tear gas, dogs and clubs on the unarmed protesters.	S2S-FA	Police used tear gas and dogs on the unarmed protesters.	S2S-Diverse-FA	They used tear gas and dogs.

Table 3: Example sentences where each component of our model improved the output sentence, compared to a model that does not use that component.

Model	SARI	Oracle
Hybrid	33.27	–
DRESS	36.00	–
DMASS	34.35	–
S2S	36.32	–
S2S-Loss	36.03	–
S2S-FA	36.47	54.01
S2S-Cluster-FA	**37.22**	50.36
S2S-Diverse-FA	35.36	52.65
S2S-All-FAS	36.30	50.40
S2S-All-FA	**37.11**	50.40

Table 4: Comparison of our models to baselines and state-of-the-art models using SARI. We also include oracle SARI scores (Oracle), given a perfect reranker. S2S-All-FA is significantly better than the DMASS and Hybrid baselines using a student t-test ($p < 0.05$).

Model	Len	FKGL	TER	Ins	Edit
Complex	23.1	11.14	0	0	–
Hybrid	12.4	7.82	0.49	0.01	–
DRESS	14.4	7.60	0.44	0.07	–
DMASS	15.1	7.40	0.59	0.28	–
S2S	16.1	7.91	0.41	0.23	–
S2S-Loss	16.4	8.11	0.40	0.31	–
S2S-FA	7.6	6.42	0.73	0.01	7.28
S2S-Cluster-FA	9.1	6.49	0.68	0.05	7.55
S2S-Diverse-FA	7.5	5.97	0.78	0.07	8.22
S2S-All-FAS	9.1	5.37	0.68	0.05	7.56
S2S-All-FA	10.8	6.42	0.61	0.07	7.56
Reference	12.8	6.90	0.67	0.42	–

Table 5: Average sentence length, FKGL, TER score compared to input, and number of insertions. We also calculate average edit distance (Edit) between candidate sentences for applicable models.

BLEU (Papineni et al., 2002) for evaluation; even though it correlates better with fluency than SARI, Sulem et al. (2018) recently showed that BLEU often negatively correlates with simplicity on the task of sentence splitting. We also calculate oracle SARI, where appropriate, to show the score we could achieve if we had a perfect reranking model. Our results are reported in Table 4.

Our best models outperform previous state-of-the-art systems, as measured by SARI. Table 4 also shows that, when used separately, reranking and clustering result in improvements on this metric. Our loss and diverse beam search methods have more ambiguous effects, especially when combined with the former two; note however that including diversity before clustering does slightly improve the oracle SARI score.

We calculate several descriptive statistics on the generated sentences and report the results in Table 5. We observe that our models produce sentences that are much shorter and lower reading level, according to Flesch-Kincaid grade level (FKGL) (Kincaid et al., 1975), while making more changes to the original sentence, according to Translation Error Rate (TER) (Snover et al., 2006). In addition, we see that the customized loss function increases the number of insertions made, while both the diversified beam search and clustering techniques individually increase the distance between sentence candidates.

5.2 Human Evaluation

While SARI has been shown to correlate with human judgments on simplicity, it only weakly cor-

Model	Fluency	Adequacy	Simplicity	All
Hybrid	2.79*	2.76	2.88*	2.81*
DRESS	**3.50**	**3.11***	3.03	**3.21***
DMASS	2.59*	2.15*	2.50*	2.41*
S2S-All-FAS	3.35	2.50*	**3.11**	2.99
S2S-All-FA	3.38	2.66	**3.08**	3.04
Reference	3.82*	3.23*	3.29*	3.45*

Table 6: Average ratings of crowdsourced human judgments on fluency, adequacy and complexity. Ratings significantly different from S2S-All-FA are marked with * ($p < 0.05$); statistical significance tests were calculated using a student t-test. We provide 95% confidence intervals for each rating in the appendix.

relates with judgments on fluency and adequacy (Xu et al., 2016). Furthermore, SARI only considers simplifications at the word level, while we believe that a simplification metric should also take into account sentence structure complexity. We plan to investigate this further in future work.

Due to the current perceived limitations of automatic metrics, we also choose to elicit human judgments on 200 randomly selected sentences to determine the relative overall quality of our simplifications. For our first evaluation, we ask native English speakers on Amazon Mechanical Turk to evaluate the fluency, adequacy, and simplicity of sentences generated by our systems and the baselines, similar to Zhang and Lapata (2017). Each annotator rated these aspects on a 5-point Likert Scale. These results are found in Table 6.[12]

As we can see, our best models substantially outperform the Hybrid and DMASS systems. Note that DMASS performs the worst, potentially because the transformer model is a more complex model that requires more training data to work properly. Comparing to DRESS, our models generate simpler sentences, but DRESS better preserves the meaning of the original sentence.

To further investigate why this is the case, we know from Table 5 that sentences generated by our model are overall shorter than other models, which also corresponds to higher TER scores. Napoles et al. (2011) notes that on sentence compression, longer sentences are perceived by human annotators to preserve more meaning than shorter sentences, controlling for quality. Thus, the drop in human-judged adequacy may be related to our sentences' relatively short lengths.

To test that this observation also holds true for simplicity, we took the candidates generated by

[12]We present the instructions for all of our human evaluations in the appendix.

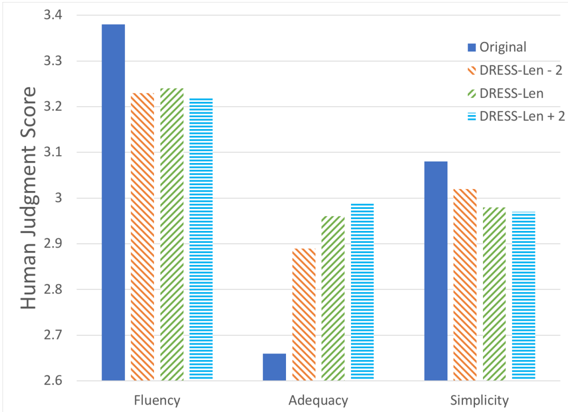

Figure 2: Effect of length on human judgments.

our best model, and after reranking them as before, we selected three sets of sentences:

- **MATCH-Dress0**: Highest ranked sentence with length closest to that of DRESS (DRESS-Len); average length is 14.10.
- **MATCH-Dress+2**: Highest ranked sentence with length closest to (DRESS-Len + 2); average length is 15.32.
- **MATCH-Dress-2**: Highest ranked sentence with length closest to (DRESS-Len - 2); average length is 12.61.

The average fluency, adequacy, and simplicity from human judgments on these new sentences are shown in Figure 2, along with those ranked highest by our best model (Original). As expected, meaning preservation does substantially increase as we increase the average sentence length, while simplicity decreases. Interestingly, fluency also decreases as sentence length increases; this is likely due to our higher-ranked sentences having greater fluency, as defined by language model perplexity.

6 Error Analysis

To gain insight in what aspects of the simplification process are challenging to our model, we present the most recurring types of errors from our test set.

6.1 Examples of Error Types

1. Long and complex sentences with multiple clauses

   (a) *Complex*: Turkey has long enshrined the secular ideals of founding father Mustafa Kemal Ataturk, particularly in an education system that until recently banned Islamic headscarves in schools and made schoolchildren begin the day reciting an oath of allegiance to Ataturk's legacy.
   Reference: Schools in Turkey had banned headscarves.
   Simple: They made schoolchildren to Ataturk's history.

   (b) *Complex*: And Wal-Mart, which imports more fruits and vegetables from Mexico than any other U.S. company, announced its effort to force improvements up and down its supply chain.
   Reference: Experts said Wal-Mart is an important company.
   Simple: Wal-Mart used more fruits and vegetables from the company.

2. Need for anaphora resolution

   (a) *Complex*: He is the creative director of Rethink Leisure & Entertainment , which is working on several projects in China and elsewhere in Asia .
   Reference: He is with Rethink Leisure & Entertainment.
   Simple: He is working on several projects in China.

   (b) *Complex*: Teachers there say Richie reads like a high school student.
   Reference: He reads like a high school student.
   Simple: Richie says he is a high school student.

3. Simplifying the wrong part of the sentence

   (a) *Complex*: Parks deliberately maintained her image as shy and proper, said Adrienne Cannon, an expert on African-American history.
   Reference: Adrienne Cannon studies African-American history.
   Simple: She is an expert on African-American history.

   (b) *Complex*: His father owned the home when the lava flowed slowly to the coast.
   Reference: His father still owned the home.
   Simple: The river cut slowly to the coast.

4. Poor substitution due to word embeddings proximity

   (a) *Complex*: Just until recently, the education system had banned Islamic **headscarves** in schools and made schoolchildren begin the day reciting a pledge of allegiance to Ataturk's legacy.
   Reference: Schools in Turkey had banned headscarves.
   Simple: Just until recently, the education officials had banned Islamic **flags**.

   (b) *Complex*: In Beijing kite circles, Fei is widely known as the elder **statesman**.
   Reference: In Beijing, Fei is widely known as an expert.
   Simple: In Beijing, Fei is considered a **doctor**.

5. Training confusion due to misalignments in training corpus

   (a) *Complex*: He heard the applause.
   Reference: The winning word was "magician."
   Simple: But he heard the song.

   (b) *Complex*: More than 300,000 students across the country signed up to participate.
   Reference: Rogers Middle School took part in the program.
   Simple: There were more than 300,000 students.

6. Acceptable simplifications that were not available in the training

   (a) *Complex*: He later tried to buy his freedom.
   Reference: Henson tried to buy his freedom later.
   Simple: The man tried to buy his freedom.

   (b) *Complex*: Middle school was a rough couple of years for Talmus' daughter, Lili Smith.
   Reference: She had a hard time in middle school.
   Simple: School was a rough couple of years.

6.2 Error Discussion

Attempting to rewrite very long and complex sentences resulted to consistent errors, as shown in 1a and 1b. This observation in combination with the examples of mis-alignments in the training corpus (5a and 5b) indicate that we either need to improve the alignments such the model can capture that the simplification process involves in many cases splitting a sentence and then simplifying or train to learn when to split first and then attempt rewriting.

The next two types of errors show failure in capturing discourse level meaning: a) errors due to failed pronoun resolution, shown in 2a and 2b, and b) errors due to the most important part of the sentence being left out, shown in 3b and 3b. In these cases, the sentences were not bad, but the information was assigned to the wrong referent, or important meaning was left out. In 4a and 4b, the substitution is clearly semantically related to the target, but changes the meaning. Finally, there were examples of acceptable simplifications, as in 6a and 6b, that were classified as errors because they were not in the gold data. We provide additional examples for each error category in the appendix.

To improve the performance of future models, we see several options. We can improve the original alignments within the Newsela corpus, particularly in the case where sentences are split. Prior to simplification, we can use additional context around the sentences to perform anaphora resolution; at this point, we can also learn when to perform sentence splitting; this has been done in the Hybrid model (Narayan and Gardent, 2014), but has not yet been incorporated into neural models. Finally, we can use syntactic information to ensure the main clause of a sentence is not removed.

7 Conclusion

In this paper, we present a novel Seq2Seq framework for sentence simplification. We contribute three major improvements over generic Seq2Seq models: a complexity-weighted loss function to encourage the model to choose simpler words; a similarity penalty during inference and clustering post-inference, to generate candidate simplifications with significant differences; and a reranking system to select the simplification that promotes both fluency and adequacy. Our model outperforms previous state-of-the-art systems using SARI, the standard metric for simplification. More importantly, while other previous models generate relatively long sentences, our model is able to generate shorter and simpler sentences, while remaining competitive regarding human-evaluated fluency and adequacy. Finally, we provide a qualitative analysis of where our different contributions improve performance, the effect of length on human-evaluated meaning preservation, and the current shortcomings of our model as insights for future research.

Generating diverse outputs from Seq2Seq models could be used in a variety of NLP tasks, such as chatbots (Shao et al., 2017), image captioning (Vijayakumar et al., 2018), and story generation (Fan et al., 2018). In addition, the proposed techniques can also be extremely helpful in leveled and personalized text simplification, where the goal is to generate different sentences based on who is requesting the simplification.

8 Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback on this work. We would also like to thank Devanshu Jain, Shyam Upadhyay, and Dan Roth for their feedback on the post-decoding aspect of this work, as well as Anne Cocos and Daphne Ippolito for their insightful comments during proofreading.

This material is based in part on research sponsored by DARPA under grant number HR0011-15-C-0115 (the LORELEI program). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA and the U.S. Government.

The work has also been supported by the French National Research Agency under project ANR-16-CE33-0013. This research was partially supported by João Sedoc's Microsoft Research Dissertation Grant. Finally, we gratefully acknowledge the support of NSF-SBIR grant 1456186.

References

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1. In *LDC2006T13*, Philadelphia, Pennsylvania. Linguistic Data Consortium.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. *CoRR*, abs/1512.01274.

Will Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, UK.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

J. Peter Kincaid, Robert P. Fishburne, Richard E. 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 ; research branch report 8-75.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 207–217, New Orleans, Louisiana. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages 1188–1196. JMLR.org.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A Simple, Fast Diverse Decoding Algorithm for Neural Generation. *CoRR*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97, Portland, Oregon. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 435–445, Baltimore, Maryland. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

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

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition LDC2011T07. DVD. *Philadelphia: Linguistic Data Consortium*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Horacio Saggion. 2017. *Automatic Text Simplification*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

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

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Neural Information Processing Systems*, Montreal, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Neural Information Processing Systems*, Long Beach, CA.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, Columbus, Ohio. Association for Computational Linguistics.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Proceedings of the International Conference on Machine Learning, Deep Learning Workshop*.

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.

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, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2014. Text rewriting improves semantic role labeling. *Journal of Artificial Intelligence Research*, 51:133–164.

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, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3(1):283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4(1):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. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Saptono Andi, and Parmanto Bambang. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 EMNLP Conference*, pages 3164–3173, Brussels, Belgium.

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, Beijing, China.