Sentence Simplification with Memory-Augmented Neural Networks

Tu Vu[1], **Baotian Hu**[2], **Tsendsuren Munkhdalai**[3] **and Hong Yu**[1,2]
[1]University of Massachusetts Amherst, Amherst, MA 01003, USA
tuvu@cs.umass.edu
[2]University of Massachusetts Medical School, Worcester, MA 01655, USA
{baotian.hu,hong.yu}@umassmed.edu
[3]Microsoft Research, Montréal, QC H3A 3H3, Canada
tsendsuren.munkhdalai@microsoft.com

Abstract

Sentence simplification aims to simplify the content and structure of complex sentences, and thus make them easier to interpret for human readers, and easier to process for downstream NLP applications. Recent advances in neural machine translation have paved the way for novel approaches to the task. In this paper, we adapt an architecture with augmented memory capacities called Neural Semantic Encoders (Munkhdalai and Yu, 2017) for sentence simplification. Our experiments demonstrate the effectiveness of our approach on different simplification datasets, both in terms of automatic evaluation measures and human judgments.

1 Introduction

The goal of sentence simplification is to compose complex sentences into simpler ones so that they are more comprehensible and accessible, while still retaining the original information content and meaning. Sentence simplification has a number of practical applications. On one hand, it provides reading aids for people with limited language proficiency (Watanabe et al., 2009; Siddharthan, 2003), or for patients with linguistic and cognitive disabilities (Carroll et al., 1999). On the other hand, it can improve the performance of other NLP tasks (Chandrasekar et al., 1996; Knight and Marcu, 2000; Beigman Klebanov et al., 2004).

Prior work has explored monolingual machine translation (MT) approaches, utilizing corpora of simplified texts, e.g., Simple English Wikipedia (SEW), and making use of statistical MT models, such as phrase-based MT (PBMT) (Štajner et al., 2015; Coster and Kauchak, 2011; Wubben et al., 2012), tree-based MT (TBMT) (Zhu et al., 2010; Woodsend and Lapata, 2011), or syntax-based MT (SBMT) (Xu et al., 2016).

Inspired by the success of neural MT (Sutskever et al., 2014; Cho et al., 2014), recent work has started exploring neural simplification with sequence to sequence (Seq2seq) models, also referred to as encoder-decoder models. Nisioi et al. (2017) implemented a standard LSTM-based Seq2seq model and found that they outperform PBMT, SBMT, and unsupervised lexical simplification approaches. Zhang and Lapata (Zhang and Lapata, 2017) viewed the encoder-decoder model as an agent and employed a deep reinforcement learning framework in which the reward has three components capturing key aspects of the target output: simplicity, relevance, and fluency.

The common practice for Seq2seq models is to use recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU, Cho et al., 2014) for the encoder and decoder (Nisioi et al., 2017; Zhang and Lapata, 2017). These architectures were designed to be capable of memorizing long-term dependencies across sequences. Nevertheless, their memory is typically small and might not be enough for the simplification task, where one is confronted with long and complicated sentences.

In this study, we go beyond the conventional LSTM/GRU-based Seq2seq models and propose to use a memory-augmented RNN architecture called Neural Semantic Encoders (NSE). This architecture has been shown to be effective in a wide range of NLP tasks (Munkhdalai and Yu, 2017). The contribution of this paper is twofold:

(1) First, we present a novel simplification model which is, to the best of our knowledge, the first model that use memory-augmented RNN for the task. We investigate the effectiveness of neural Seq2seq models when different neural architectures for the encoder are considered. Our experiments reveal that the NSELSTM model that uses an

Proceedings of NAACL-HLT 2018, pages 79–85
New Orleans, Louisiana, June 1 - 6, 2018.

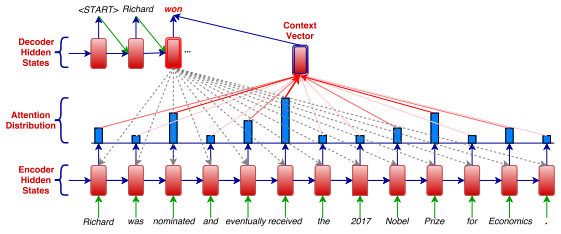

Figure 1: Attention-based encoder-decoder model. The model may attend to relevant source information while decoding the simplification, e.g., to generate the target word *won* the model may attend to the source words *received*, *nominated* and *Prize*.

NSE as the encoder and an LSTM as the decoder performed the best among these models, improving over strong simplification systems. (2) Second, we perform an extensive evaluation of various approaches proposed in the literature on different datasets. Results of both automatic and human evaluation show that our approach is remarkably effective for the task, significantly reducing the reading difficulty of the input, while preserving grammaticality and the original meaning. We further discuss some advantages and disadvantages of these approaches.

2 Neural Sequence to Sequence Models

2.1 Attention-based Encoder-Decoder Model

Our approach is based on an attention-based Seq2seq model (Bahdanau et al., 2015) (Figure 1). Given a complex source sentence $\mathcal{X} = x_{1:T_x}$, the model learns to generate its simplified version $\mathcal{Y} = y_{1:T_y}$. The encoder reads through $\mathcal{X}$ and computes a sequence of hidden states $h_{1:T_x}$:

$$h_t = \mathcal{F}^{enc}(h_{t-1}, x_t),$$

where $\mathcal{F}^{enc}$ is a non-linear activation function (e.g., LSTM), h_t is the hidden state at time t. Each time the model generates a target word y_t, the decoder looks at a set of positions in the source sentence where the most relevant information is located. Specifically, another non-linear activation function F^{dec} is used for the decoder where the hidden state s_t at time t is computed by:

$$s_t = \mathcal{F}^{dec}(s_{t-1}, y_{t-1}, c_t).$$

Here, the context vector c_t is computed as a weighted sum of the hidden vectors $h_{1:T_x}$:

$$c_t = \sum_{i=1}^{T_x} \alpha_{ti} h_i, \qquad \alpha_{ti} = \frac{\exp(s_{t-1} \odot h_i)}{\sum_{j=1}^{T_x} \exp(s_{t-1} \odot h_j)},$$

where $\odot$ is the dot product of two vectors. Generation is conditioned on c_t and all the previously generated target words $y_{1:t-1}$:

$$P(\mathcal{Y}|\mathcal{X}) = \prod_{t=1}^{T_y} P(y_t|\{y_{1:t-1}\}, c_t),$$

$$P(y_t|\{y_{1:t-1}\}, c_t) = \mathcal{G}(y_{t-1}, s_t, c_t),$$

where $\mathcal{G}$ is some non-linear function. The training objective is to minimize the cross-entropy loss of the training source-target pairs.

2.2 Neural Semantic Encoders

An RNN allows us to compute a hidden state h_t of each word summarizing the preceding words $x_{1:t}$, but not considering the following words $x_{t+1:T_x}$ that might also be useful for simplification. An alternative approach is to use a bidirectional-RNN (Schuster and Paliwal, 1997). Here, we propose to use Neural Semantic Encoders (NSE, Munkhdalai and Yu, 2017). During each encoding time step t, we compute a memory matrix $M_t \in \mathbb{R}^{T_x \times D}$ where D is the dimensionality of the word vectors. This matrix is initialized with the word vectors and is refined over time through NSE's functions to gain a better understanding of the input sequence. Concretely, NSE sequentially reads the tokens $x_{1:T_x}$ with its *read* function:

$$r_t = \mathcal{F}^{enc}_{read}(r_{t-1}, x_t),$$

where $\mathcal{F}^{enc}_{read}$ is an LSTM, $r_t \in \mathbb{R}^D$ is the hidden state at time t. Then, a *compose* function is used to compose r_t with relevant information retrieved from the memory at the previous time step, M_{t-1}:

$$c_t = \mathcal{F}^{enc}_{compose}(r_t, m_t),$$

where $\mathcal{F}^{enc}_{compose}$ is a multi-layer perceptron with one hidden layer, $c_t \in \mathbb{R}^{2D}$ is the output vector, and $m_t \in \mathbb{R}^D$ is a linear combination of the memory slots of M_{t-1}, weighted by $\sigma_{ti} \in \mathbb{R}$:

$$m_t = \sum_{i=1}^{T_x} \sigma_{ti} M_{t-1,i}, \qquad \sigma_{ti} = \frac{\exp(r_t \odot M_{t-1,i})}{\sum_{j=1}^{T_x} \exp(r_t \odot M_{t-1,j})}.$$

Here, $M_{t-1,i}$ is the i^{th} row of the memory matrix at time $t-1$, M_{t-1}. Next, a *write* function is used to map c_t to the encoder output space:

$$w_t = \mathcal{F}^{enc}_{write}(w_{t-1}, c_t),$$

where $\mathcal{F}^{enc}_{write}$ is an LSTM, $w_t \in \mathbb{R}^D$ is the hidden state at time t. Finally, the memory is updated accordingly. The retrieved memory content pointed by σ_{ti} is erased and the new content is added:

$$M_{t,i} = (1 - \sigma_{ti}) M_{t-1,i} + \sigma_{ti} w_t.$$

NSE gives us unrestricted access to the entire source sequence stored in the memory. As such, the encoder may attend to relevant words when encoding each word. The sequence $w_{1:T_x}$ is then used as the sequence $h_{1:T_x}$ in Section 2.1.

2.3 Decoding

We differ from the approach of Zhang et al. (2017) in the sense that we implement both a greedy strategy and a beam-search strategy to generate the target sentence. Whereas the greedy decoder always chooses the simplification candidate with the highest log-probability, the beam-search decoder keeps a fixed number (beam) of the highest scoring candidates at each time step. We report the best simplification among the outputs based on automatic evaluation measures.

3 Experimental Setup

3.1 Datasets

Following (Zhang and Lapata, 2017), we experiment on three simplification datasets, namely: (1) *Newsela* (Xu et al., 2015), a high-quality simplification corpus of news articles composed by Newsela[1] professional editors. We used the split of the data in (Zhang and Lapata, 2017), i.e., 94,208/1,129/1,077 pairs for train/dev/test. (2) *WikiSmall* (Zhu et al., 2010), which contains aligned complex-simple sentence pairs from English Wikipedia (EW) and SEW. The dataset has 88,837/205/100 pairs for train/dev/test. (3) *WikiLarge* (Zhang and Lapata, 2017), a larger corpus in which the training set is a mixture of three Wikipedia datasets in (Zhu et al., 2010; Woodsend and Lapata, 2011; Kauchak, 2013), and the development and test sests are complex sentences taken from *WikiSmall*, each has 8 simplifications written by Amazon Mechanical Turk workers (Xu et al., 2016). The dataset has 296,402/2,000/359 pairs for train/dev/test. Table 1 provides statistics on the training sets.

Dataset	vocab size		#tokens/sent	
	src	tgt	src	tgt
Newsela	41,066	30,193	25.94	15.89
WikiSmall	113,368	93,835	24.26	20.33
WikiLarge	201,841	168,962	25.17	18.51

Table 1: Statistics for the training sets: the vocabulary size (vocab size), and the average number of tokens per sentence (#tokens/sent) of the source (src) and target (tgt) language.

3.2 Models and Training Details

We implemented two attention-based Seq2seq models, namely: (1) LSTMLSTM: the encoder is implemented by two LSTM layers; (2) NSELSTM: the encoder is implemented by NSE. The decoder in both cases is implemented by two LSTM layers. For all experiments, our models have 300-dimensional hidden states and 300-dimensional word embeddings. Parameters were initialized from a uniform distribution [-0.1, 0.1). We used the same hyperparameters across all datasets. Word embeddings were initialized either randomly or with Glove vectors (Pennington et al., 2014) pre-trained on Common Crawl data (840B tokens), and fine-tuned during training. We used a vocabulary size of 20K for Newsela, and 30K for WikiSmall and WikiLarge. Our models were trained with a maximum number of 40 epochs using Adam optimizer (Kingma and Ba, 2015) with step size $\alpha = 0.001$ for LSTMLSTM, and 0.0003 for NSELSTM, the exponential decay rates $\beta_1 = 0.9, \beta_2 = 0.999$. The batch size is set to 32. We used dropout (Srivastava et al., 2014) for regularization with a dropout rate of 0.3. For beam search, we experimented with beam sizes of 5 and 10. Following (Jean et al., 2015), we replaced each out-of-vocabulary token $\langle unk \rangle$ with the source word x_k with the highest alignment score α_{ti}, i.e., $k = \underset{i}{\text{argmax}}(\alpha_{ti})$.

Our models were tuned on the development sets, either with BLEU (Papineni et al., 2002) that scores the output by counting n-gram matches with the reference, or SARI (Xu et al., 2016) that compares the output against both the reference and the input sentence. Both measures are commonly used to automatically evaluate the quality of simplification output. We noticed that SARI should be used with caution when tuning neural Seq2seq simplification models. Since SARI depends on the differences between a system's output and the input sentence, large differences may yield very good SARI even though the output is ungrammatical. Thus, when tuning with SARI, we ignored epochs in which the BLEU score of the output is too low, using a threshold ς. We set ς to 22 on Newsela, 33 on WikiSmall, and 77 on WikiLarge.

3.3 Comparing Systems

We compared our models, either tuned with BLEU (-B) or SARI (-S), against systems reported in (Zhang and Lapata, 2017), namely DRESS, a deep reinforcement learning model, DRESS-LS, a combination of DRESS and a lexical simplification model (Zhang and Lapata, 2017), PBMT-

[1] https://newsela.com

R, a PBMT model with dissimilarity-based reranking (Wubben et al., 2012), HYBRID, a hybrid semantic-based model that combines a simplification model and a monolingual MT model (Narayan and Gardent, 2014), and SBMT-SARI, a SBMT model with simplification-specific components. (Xu et al., 2016).

3.4 Evaluation

We measured BLEU, and SARI at corpus-level following (Zhang and Lapata, 2017). In addition, we also evaluated system output by eliciting human judgments. Specifically, we randomly selected 40 sentences from each test set, and included human reference simplifications and corresponding simplifications from the systems above[2]. We then asked three volunteers[3] to rate simplifications with respect to *Fluency* (the extent to which the output is grammatical English), *Adequacy* (the extent to which the output has the same meaning as the input sentence), and *Simplicity* (the extent to which the output is simpler than the input sentence) using a five point Likert scale.

4 Results and Discussions

4.1 Automatic Evaluation Measures

The results of the automatic evaluation are displayed in Table 2. We first discuss the results on Newsela that contains high-quality simplifications composed by professional editors. In terms of BLEU, all neural models achieved much higher scores than PBMT-R and HYBRID. NSELSTM-B scored highest with a BLEU score of 26.31. With regard to SARI, NSELSTM-S scored best among neural models (29.58) and came close to the performance of HYBRID (30.00). This indicates that NSE offers an effective means to better encode complex sentences for sentence simplification.

On WikiSmall, HYBRID – the current state-of-the-art – achieved best BLEU (53.94) and SARI (30.46) scores. Among neural models, NSELSTM-B yielded the highest BLEU score (53.42), while NSELSTM-S performed best on SARI (29.75). On WikiLarge[4], again, NSELSTM-B had the highest BLEU score of 92.02. SBMT-SARI – that was trained on a huge corpus of 106M sentence pairs and 2B words – scored highest on SARI with 39.96, followed by DRESS-LS (37.27), DRESS (37.08), and NSELSTM-S (36.88).

Model	Newsela		WikiSmall		WikiLarge	
	BLEU	**SARI**	**BLEU**	**SARI**	**BLEU**	**SARI**
PBMT-R	18.19	15.77	46.31	15.97	81.11	38.56
HYBRID	14.46	**30.00**	**53.94**	**30.46**	48.97	31.40
SBMT-SARI	NA		NA		73.08	**39.96**
DRESS	23.21	27.37	34.53	27.48	77.18	37.08
DRESS-LS	24.30	26.63	36.32	27.24	80.12	37.27
LSTMLSTM-B	24.38	27.66	50.53	17.67	88.81	34.22
NSELSTM-B	**26.31**	27.42	53.42	17.47	**92.02**	33.43
LSTMLSTM-S	23.50	28.67	31.32	28.04	81.95	35.45
NSELSTM-S	22.62	29.58	29.72	29.75	80.43	36.88

Table 2: Model performance using automatic evaluation measures (BLEU and SARI).

4.2 Human Judgments

The results of human judgments are displayed in Table 3. On Newsela, NSELSTM-B scored highest on Fluency. PBMT-R was significantly better than all other systems on Adequacy while LSTMLSTM-S performed best on Simplicity. NSELSTM-B did very well on both Adequacy and Simplicity, and was best in terms of Average. Example model outputs on Newsela are provided in Table 4.

On WikiSmall, NSELSTM-B performed best on both Fluency and Adequacy. On WikiLarge, LSTMLSTM-B achieved the highest Fluency score while NSELSTM-B received the highest Adequacy score. In terms of Simplicity and Average, NSELSTM-S outperformed all other systems on both WikiSmall and WikiLarge.

As shown in Table 3, neural models often outperformed traditional systems (PBMT-R, HYBRID, SBMT-SARI) on Fluency. This is not surprising given the recent success of neural Seq2seq models in language modeling and neural machine translation (Zaremba et al., 2014; Jean et al., 2015). On the downside, our manual inspection reveals that neural models learn to perform copying very well in terms of rewrite operations (e.g., copying, deletion, reordering, substitution), often outputting the same or parts of the input sentence.

Finally, as can be seen in Table 3, REFERENCE scored lower on Adequacy compared to Fluency and Simplicity on Newsela. On Wikipedia-based datasets, REFERENCE obtained high Adequacy scores but much lower Simplicity scores compared to Newsela. This supports the assertion by previous work (Xu et al., 2015) that SEW has a large proportion of inadequate simplifications.

[2]The outputs of comparison systems are available at https://github.com/XingxingZhang/dress.

[3]two native English speakers and one non-native fluent English speaker

[4]Here, BLEU scores are much higher compared to Newsela and WikiSmall since there are 8 reference simplifications for each input sentence in the test set.

Model	Newsela				WikiSmall				WikiLarge			
	F	A	S	Avg.	F	A	S	Avg.	F	A	S	Avg.
REFERENCE	4.58	2.98	3.99	3.85	4.63	3.97	3.59	4.06	4.59	4.43	2.38	3.80
PBMT-R	3.73	**3.90**	1.98	3.20	4.07	4.11	2.28	3.49	4.22	4.09	2.31	3.54
HYBRID	2.77	2.56	2.41	2.58	3.21	3.62	2.56	3.13	2.63	2.48	2.26	2.46
SBMT-SARI	NA				NA				3.89	3.87	2.54	3.43
DRESS	3.98	2.84	2.93	3.25	4.35	3.33	3.49	3.72	4.56	3.66	2.63	3.62
DRESS-LS	3.99	2.90	2.98	3.29	4.43	3.33	3.56	3.77	4.68	3.88	2.63	3.73
LSTMLSTM-B	3.95	2.93	3.14	3.34	4.42	3.88	2.65	3.65	**4.80**	4.47	1.89	3.72
NSELSTM-B	**4.26**	3.13	3.39	**3.59**	**4.74**	**4.22**	2.49	3.82	4.73	**4.58**	1.94	3.75
LSTMLSTM-S	4.24	3.03	**3.45**	3.57	4.59	3.40	3.42	3.80	4.73	4.23	2.21	3.72
NSELSTM-S	3.83	2.78	3.01	3.21	4.57	3.28	**3.81**	**3.89**	4.65	3.95	**2.90**	**3.83**

Table 3: Average human ratings (Fluency (F), Adequacy (A), Simplicity (S), and Average (Avg.)).

COMPLEX: Another parent , Mike Munson , sits on the bench with a tablet and uses an app to track and analyze the team 's shots .
REFERENCE: **Basketball** parent Mike Munson sits on the bench with a tablet , **like an iPad** .
PBMT-R: Another parent , Mike Munson **is** on the bench with a tablet and uses an app to track and analyze the team 's shots .
HYBRID: another parent , mike munson sits uses an app to track and analyze shots .
DRESS: Another parent , Mike Munson , sits on the bench with a **computer** .
DRESS-LS: Another parent , Mike Munson , sits on the bench with a **computer** .
LSTMLSTM-B: **He starts** on the bench with a tablet and uses an app to track .
NSELSTM-B: Another parent , Mike Munson , sits on the bench with a tablet and uses an app to track .
LSTMLSTM-S: **She** sits on the bench with a tablet and uses an app to track and **study** the team 's shots .
NSELSTM-S: **He** sits on the bench with a tablet .
COMPLEX: Stowell believes that even documents about Lincoln 's death will give people a better understanding of the man who was assassinated 150 years ago this April .
REFERENCE: Stowell **thinks** that even **information** about Lincoln 's death will **help** people **understand him** .
PBMT-R: Stowell **thinks** that even documents about Lincoln 's death will give people a better understanding of the man who was killed 150 years ago this April .
HYBRID: documents **that** will give people a understanding the man was assassinated 150 years ago .
DRESS: Stowell **thinks** that even documents about Lincoln 's death will give people a better understanding of the man .
DRESS-LS: Stowell **thinks** that even documents about Lincoln 's death will give people a better understanding of the man .
LSTMLSTM-B: Stowell believes that **only** documents about Lincoln 's death will give people a better understanding .
NSELSTM-B: Stowell believes that **the discovery** about Lincoln 's death will give people a better understanding of the man .
LSTMLSTM-S: Stowell **thinks** that even documents about Lincoln 's death will give people a better understanding of the man .
NSELSTM-S: Stowell **thinks** that even **papers** about Lincoln 's death will give people a better understanding of the man .

Table 4: Example model outputs on Newsela. Substitutions are shown in bold.

4.3 Correlations

Table 5 shows the correlations between the scores assigned by humans and the automatic evaluation measures. There is a positive significant correlation between Fluency and Adequacy (0.69), but a negative significant correlation between Adequacy and Simplicity (-0.64). BLEU correlates well with Fluency (0.63) and Adequacy (0.90) while SARI correlates well with Simplicity (0.73). BLEU and SARI show a negative significant correlation (-0.54). The results reflect the challenge of managing the trade-off between Fluency, Adequacy and Simplicity in sentence simplification.

	Adequacy	Simplicity	BLEU	SARI
Fluency	0.69**	-0.03	0.63**	-0.48**
Adequacy		-0.64**	0.90**	-0.81**
Simplicity			-0.56**	0.73**
BLEU				-0.54**

Table 5: Pearson correlation between the scores assigned by humans and the automatic evaluation measures. Scores marked ** are significant at $p < 0.01$.

5 Conclusions

In this paper, we explore neural Seq2seq models for sentence simplification. We propose to use an architecture with augmented memory capacities which we believe is suitable for the task, where one is confronted with long and complex sentences. Results of both automatic and human evaluation on different datasets show that our model is capable of significantly reducing the reading difficulty of the input, while performing well in terms of grammaticality and meaning preservation.

6 Acknowledgements

We would like to thank Emily Druhl, Jesse Lingeman, and the UMass BioNLP team for their help with this work. We also thank Xingxing Zhang and Sergiu Nisioi for valuable discussions, and the anonymous reviewers for their thoughtful comments and suggestions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Curran Associates, Inc., San Diego, CA, USA, pages 3104–3112.

Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In *Proceed- ings of Ontologies, Dabases, and Applications of Semantics (ODBASE) International Conference, volume 3290 of Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pages 735–747.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics, Bergen, Norway, pages 269–270.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*. Stroudsburg, PA, USA, pages 1041–1044.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734.

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

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal, pages 134–140.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1537–1546.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Curran Associates, Inc., San Diego, CA, USA.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI) and Twelfth Conference on Innovative Applications of Artificial Intelligence (IAAI)*. AAAI Press, pages 703–710.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural semantic encoders. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics, Valencia, Spain, pages 397–407.

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 (ACL)*. Association for Computational Linguistics, Baltimore, Maryland, pages 435–445.

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 (ACL)*. Association for Computational Linguistics, Vancouver, Canada, pages 85–91.

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 (ACL)*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 311–318.

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)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Advaith Siddharthan. 2003. Syntactic simplification and text cohesion. Ph.D. Thesis, University of Cambridge, University of Cambridge.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27 (NIPS)*, Curran Associates, Inc., pages 3104–3112.

Sanja Štajner, Hannah Bechara, and Horacio Saggion. 2015. A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*. Association for Computational Linguistics, Beijing, China, pages 823–828.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: Reading assistance for low-literacy readers. In *Proceedings of the 27th ACM International Conference on Design of Communication (SIGDOC)*. ACM, New York, NY, USA, pages 29–36.

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 (EMNLP)*. Association for Computational Linguistics, Edinburgh, Scotland, UK., pages 409–420.

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

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 (TACL)* 3:283–297.

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

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .

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 (EMNLP)*. Association for Computational Linguistics, Copenhagen, Denmark, pages 595–605.

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)*. Coling 2010 Organizing Committee, Beijing, China, pages 1353–1361.