

# How May I Help You?

## Using Neural Text Simplification to Improve Downstream NLP Tasks

Hoang Van, Zheng Tang, and Mihai Surdeanu

Department of Computer Science, University of Arizona  
{vnhh, zhenGTang, msurdeanu}@email.arizona.edu

### Abstract

The general goal of text simplification (TS) is to reduce text complexity for human consumption. In this paper, we investigate another potential use of neural TS: assisting machines performing natural language processing (NLP) tasks. We evaluate the use of neural TS in two ways: simplifying input texts at prediction time and augmenting data to provide machines with additional information during training. We demonstrate that the latter scenario provides positive effects on machine performance on two separate datasets. In particular, the latter use of TS significantly improves the performances of LSTM (1.82–1.98%) and SpanBERT (0.7–1.3%) extractors on TACRED, a complex, large-scale, real-world relation extraction task. Further, the same setting yields significant improvements of up to 0.65% matched and 0.62% mismatched accuracies for a BERT text classifier on MNLI, a practical natural language inference dataset.

### 1 Introduction

The goal of text simplification (TS) is to reduce text complexity (while preserving meaning) such that the corresponding text becomes more accessible to human readers. Previous works explored how TS can assist children (Kajiwara et al., 2013), non-native speakers (Pellow and Eskenazi, 2014), and people with disabilities (Rello et al., 2013). While this can be achieved in a variety of approaches (Sikka et al., 2020), most TS research has focused on two major approaches: rule-based and neural sequence-to-sequence (seq2seq). Since 2017, there is a significant increase of neural seq2seq TS methods (Zhang and Lapata, 2017; Zhao et al., 2018; Kriz et al., 2019; Maddela et al., 2020; Jiang et al., 2020).

In this paper, we analyze another potential use of the latter TS direction: assisting machines performing natural language processing (NLP) tasks.

To this end, we investigate two possible directions: (a) using TS to simplify input texts at prediction time, and (b) using TS to augment training data for the respective NLP tasks. We empirically analyze these two directions using two neural TS systems (Martin et al., 2019; Nisioi et al., 2017), and two NLP tasks: relation extraction using the TACRED dataset (Zhang et al., 2017), and multi-genre natural language inference (MNLI) (Williams et al., 2017). Further, within these two tasks, we explore three methods: two relation extraction approaches, one based on LSTMs (Hochreiter and Schmidhuber, 1997) and another based on transformer networks, SpanBERT (Joshi et al., 2020), and one method for MNLI also based on transformer networks, BERT (Devlin et al., 2018).

Our analysis shows that simplifying texts at prediction times does not improve results, but using TS to augment training data consistently helps in all configurations. In particular, after augmented data is added, all approaches outperform their respective configurations without augmented data on both TACRED (0.7–1.98% in F1) and MNLI (0.50–0.65% in accuracies) tasks. The reproducibility checklist and the software are available at this link: <https://github.com/vanh17/TextSim>.

### 2 Related Work

Recent work have effectively proven the practical application of neural networks and neural deep learning approaches to solving machine learning problems (Ghosh et al., 2021; Blalock et al., 2020; Yin et al., 2017).

With respect to input simplification, several works have utilized TS as a pre-processing step for downstream NLP tasks such as information extraction (Miwa et al., 2010; Schmidek and Barbosa, 2014; Niklaus et al., 2017), parsing (Chandrasekar et al., 1996), semantic role labeling (Vickrey and Koller, 2008), and machine translation (Štajner and Popović, 2016). However, most of them focus on

|   |               | ACCESS          | NTS             |
|---|---------------|-----------------|-----------------|
| 1 | Training Data | $0.67 \pm 0.16$ | $0.89 \pm 0.22$ |
| 2 | Dev Data      | $0.68 \pm 0.15$ | $0.92 \pm 0.18$ |

Table 1: The empirical differences in BLEU scores (Papineni et al., 2002) between original and simplified text generated by two TS systems, ACCESS and NTS, in TACRED training and dev datasets.

the use of rule-based TS methods. In contrast, we investigate the potential use of domain-agnostic neural TS systems in simplifying inputs for downstream tasks. We show that, despite the complexity of the tasks investigated and the domain agnosticity of the TS approaches, TS improves both tasks when used for training data augmentation, but not when used to simplify evaluation texts.

On data augmentation for natural language processing downstream tasks, previous work show significant benefits of introducing noisy data on the machine performance (Van et al., 2021; Kobayashi, 2018). Previous efforts used TS approaches, e.g. lexical substitution, to augment training data for downstream tasks such as text classification (Zhang et al., 2015; Wei and Zou, 2019). However, these methods focused on replacing words with thesaurus-based synonyms, and did not emphasize other important lexical and syntactic simplification. Here, we use two out-of-the-box neural TS systems that apply both lexical and syntactic sentence simplification for data augmentation, and show that our data augmentation consistently leads to better performances. Note that we do not use rule-based TS systems because they have been proven to perform worse than their neural counterparts (Zhang and Lapata, 2017; Nisioi et al., 2017). Further, rule-based TS systems are harder to build in a domain-independent way due to the many linguistic/syntactic variations across domains.

### 3 Approach

We investigate the impact of text simplification on downstream NLP tasks in two ways: (a) simplifying input texts at prediction time, and (b) augmenting training data for the respective NLP tasks. We discuss the settings of these experiments next.

#### 3.1 Input Simplification at Prediction Time

We pose the run-time input simplification problem as a transparent data pre-processing problem. That is, given an input data point, we simplify the text

|                       |            | ACCESS          | NTS             |
|-----------------------|------------|-----------------|-----------------|
| <i>Train</i>          |            |                 |                 |
| 1                     | Premise    | $0.62 \pm 0.24$ | $0.76 \pm 0.25$ |
| 2                     | Hypothesis | $0.62 \pm 0.30$ | $0.80 \pm 0.17$ |
| <i>Dev mismatched</i> |            |                 |                 |
| 3                     | Premise    | $0.62 \pm 0.28$ | $0.80 \pm 0.22$ |
| 4                     | Hypothesis | $0.65 \pm 0.23$ | $0.81 \pm 0.17$ |
| <i>Dev matched</i>    |            |                 |                 |
| 5                     | Premise    | $0.62 \pm 0.30$ | $0.75 \pm 0.26$ |
| 6                     | Hypothesis | $0.60 \pm 0.25$ | $0.80 \pm 0.17$ |

Table 2: The empirical differences in BLEU scores (Papineni et al., 2002) between original and simplified text generated by two TS systems, ACCESS and NTS, in MNLI training and dev datasets.

while keeping the native format of the task, and then feed the modified input to the actual NLP task. For example, for the TACRED sentence “*the CFO Douglas Flint will become chairman, succeeding Stephen Green is leaving for a government job.*”, which contains a per:title relation between the two entities *Douglas Flint* and *chairman*, our approach will first simplify the text to “*the CFO Douglas Flint will become chairman, and Stephen Green is leaving to take a government job.*”. Then we generate a relation prediction for the simplified text using existing relation extraction classifiers.

#### 3.2 Data Augmentation for Training

Here, we augment training data by simplifying the text of some original training examples, and appending it to the original training dataset. First, we sample which examples should be used for augmentation with probability  $p$ . Second, once an example is selected for augmentation, we generate an additional example with the text portion simplified using TS. For example, for the data in section 3.1, we generate an additional training data with the corresponding simplified text.  $p$  is a hyper parameter that we tuned for each task (see next section).

### 4 Experimental Setup

**NLP tasks and methods:** We evaluate the impact of TS on two NLP tasks: (a) relation extraction (RE) using the TACRED dataset (Zhang et al., 2017), and (b) natural language inference (NLI) on the MNLI dataset (Williams et al., 2017).

TACRED is a large-scale RE dataset with 106,264 examples built on newswire and web text

|                        |                            | Simplified   | Original     | Simplified + Complement |
|------------------------|----------------------------|--------------|--------------|-------------------------|
| <b>Train Data Sets</b> |                            |              |              |                         |
| <b>LSTM</b>            |                            |              |              |                         |
| 1                      | Original                   | 59.95        | 62.70        | 61.32                   |
| 2                      | Simplified + Complement    | 62.34        | 62.59        | 62.12                   |
| 3                      | Simplified + Original (AD) | <b>62.64</b> | <b>64.52</b> | <b>64.08</b>            |
| <b>SpanBERT</b>        |                            |              |              |                         |
| 4                      | Original                   | 62.42        | 66.70        | 64.12                   |
| 5                      | Simplified + Complement    | 64.12        | 65.45        | 64.92                   |
| 6                      | Simplified + Original (AD) | <b>65.14</b> | <b>68.00</b> | <b>66.43</b>            |

Table 3: F1 on TACRED test set of the LSTM and SpanBERT approaches using ACCESS (Martin et al., 2019) as the TS method. The different rows indicate the different data augmentation strategies applied on the training data, while the columns indicate the type of simplification applied at runtime on the test data. We investigated the following configurations: *Original*: unmodified dataset; *Simplified + Complement*: consists of simplified data that preserves critical information combined with original data when simplification fails to preserve important information; *Simplified + Original*: consists of all original data augmented with additional simplified data that preserves critical information. (AD) annotates models using data augmented by neural TS systems during training.

with an average sentence length of 36.4 words. Each sentence contains two entities in focus (called subject and object) and a relation that holds between them. We selected this task because the nature of RE requires critical information preservation, which is challenging for neural TS methods (Van et al., 2020). That is, the simplified sentences must contain the subject and object entities.

The MNLI corpus is a crowd-sourced collection of 433K sentence pairs annotated for NLI. The average sentence length in this dataset is 22.3 words. Each data point contains a premise-hypothesis pair and one of the three labels: contradiction, entailment, and neutral. We selected MNLI as the second task to further understand the effects of TS on machine performance on tasks that rely on long text, which is a challenge for TS methods (Shardlow, 2014; Xu et al., 2015).

We train three approaches for these two tasks. First, for TACRED, we use a classifier based on

|                        |                            | Simplified   | Original     | Simplified + Complement |
|------------------------|----------------------------|--------------|--------------|-------------------------|
| <b>Train Data Sets</b> |                            |              |              |                         |
| <b>LSTM</b>            |                            |              |              |                         |
| 1                      | Original                   | 60.47        | 62.70        | 61.03                   |
| 2                      | Simplified + Complement    | <b>63.40</b> | 62.96        | 62.28                   |
| 3                      | Simplified + Original (AD) | 62.91        | <b>64.68</b> | <b>64.35</b>            |
| <b>SpanBERT</b>        |                            |              |              |                         |
| 4                      | Original                   | 62.20        | 66.70        | 63.90                   |
| 5                      | Simplified + Complement    | 64.12        | 65.32        | 63.92                   |
| 6                      | Simplified + Original (AD) | <b>65.32</b> | <b>67.40</b> | <b>65.47</b>            |

Table 4: F1 on TACRED test set of the LSTM and SpanBERT approaches using NTS (Nisioi et al., 2017) as the TS method. This table follows the same format as Table 3.

LSTMs<sup>1</sup> (Hochreiter and Schmidhuber, 1997), and a second based on SpanBERT<sup>2</sup> (Joshi et al., 2020). For MNLI, we trained a BERT-based classifier<sup>3</sup> (Devlin et al., 2018). For reproducibility, we use the default settings and general hyper parameters recommended by the task and creators of the transformer networks (Zhang et al., 2017; Joshi et al., 2020; Devlin et al., 2018). Through this, we aim to separate potential improvements of our approaches from those coming from improved configurations.

**Text simplification methods:** For TS, we use two out-of-the-box neural seq2seq TS approaches: ACCESS (Martin et al., 2019), and NTS (Nisioi et al., 2017). Tables 1 and 2 show the BLEU scores (Papineni et al., 2002) between original and simplified text generated by these two TS systems for the two tasks. The tables highlight that both systems change the input texts, with ACCESS being more aggressive.

**Evaluation measures:** We directly followed the evaluation measures proposed by the original task organizers (Zhang et al., 2017; Williams et al., 2017). Specifically, we used these main metrics: (a) F1 on TACRED relation extraction, and (b)

<sup>1</sup><https://github.com/yuhaozhang/tacred-relation>

<sup>2</sup><https://huggingface.co/SpanBERT/spanbert-large-cased>

<sup>3</sup><https://huggingface.co/bert-base-cased>

|                        |   | Simplified<br>m/mm acc        | Original<br>m/mm acc                         |
|------------------------|---|-------------------------------|--|
| <i>Train Data Sets</i> |   |                               |  |
| <b>ACCESS</b>          |   |                               |  |
| 1                      | Original                                | 71.25/71.43                   | 82.89/83.10                                  |
| 2                      | Original Swapped<br>with 10% Simplified | 71.76 ± 0.13/<br>72.12 ± 0.08 | 83.00 ± 0.03/<br>83.25 ± 0.05                |
| 3                      | Original Swapped<br>with 20% Simplified | 72.74 ± 0.10/<br>73.10 ± 0.12 | 82.66 ± 0.07/<br>82.88 ± 0.09                |
| 4                      | 5% Simplified<br>+ Original (AD)        | 71.30 ± 0.15/<br>71.52 ± 0.10 | <b>83.47 ± 0.04</b><br><b>83.61 ± 0.08</b>   |
| 5                      | 10% Simplified<br>+ Original (AD)       | 71.81 ± 0.07/<br>71.99 ± 0.08 | 82.81 ± 0.05/<br>83.05 ± 0.09                |
| 6                      | 15% Simplified<br>+ Original (AD)       | 71.87 ± 0.11/<br>72.10 ± 0.07 | 82.92 ± 0.05/<br>83.13 ± 0.06                |
| <b>NTS</b>             |   |                               |  |
| 7                      | Original                                | 33.36/33.53                   | 82.89/83.10                                  |
| 8                      | Original Swapped<br>with 10% Simplified | 33.39 ± 0.10/<br>33.46 ± 0.08 | 83.28 ± 0.07/<br>83.50 ± 0.11                |
| 9                      | Original Swapped<br>with 20% Simplified | 33.71 ± 0.08/<br>33.90 ± 0.11 | 82.60 ± 0.14/<br>82.79 ± 0.09                |
| 10                     | 5% Simplified<br>+ Original (AD)        | 33.35 ± 0.10/<br>33.50 ± 0.09 | 83.20 ± 0.09/<br>83.41 ± 0.10                |
| 11                     | 10% Simplified<br>+ Original (AD)       | 33.50 ± 0.07/<br>33.80 ± 0.09 | <b>83.51 ± 0.05</b> /<br><b>83.70 ± 0.07</b> |
| 12                     | 15% Simplified<br>+ Original (AD)       | 33.65 ± 0.04/<br>33.79 ± 0.10 | 83.09 ± 0.05<br>83.25 ± 0.07                 |

Table 5: Matched (m) and mismatched (mm) accuracies on MNLI development using text simplified/augmented by ACCESS (top half) and NTS (bottom half). *Original Swapped with x% Simplified* consists of original data with x% of data points replaced with their simplified form. *x% Simplified + Original* consists of the original data augmented with an additional x% of simplified data. (AD) annotates models using data augmented by neural TS systems during training. Note that our results in the original configuration differ slightly from those in (Devlin et al., 2018). This is likely due to the different hardware and library versions used (Belz et al., 2021).

matched/mismatched accuracies on MNLI.

**Hyper parameter tuning:** We tuned the only hyper parameter for data augmentation, the percentage of augmented data points,  $p$ , for MNLI. On this task we augmented 5, 10, and 15% of sentence pairs from training data, and found 5 and 10% of training data as the best thresholds for ACCESS and NTS respectively. For TACRED, we did not use this hyper parameter. Instead, we used all simplifications that preserve critical information for data augmentation. That is, we added all simplified sentences that preserve the subject and object entities necessary for the underlying relation. We found that 66% of training data sentences could

|                        |                                   | Simplified<br>m/mm acc | Original<br>m/mm acc |
|------------------------|-----------------------------------|------------------------|----------------------|
| <i>Train Data Sets</i> |                                   |                        |                      |
| <b>ACCESS</b>          |                                   |                        |                      |
| 1                      | Original                          | 71.10/71.30            | 82.78/83.00          |
| 2                      | 5% Simplified +<br>Original (AD)  | <b>71.21/71.40</b>     | <b>83.37/83.50</b>   |
| <b>NTS</b>             |                                   |                        |                      |
| 3                      | Original                          | 33.25/33.45            | 82.78/83.00          |
| 4                      | 10% Simplified +<br>Original (AD) | <b>33.39/33.61</b>     | <b>83.43/83.62</b>   |

Table 6: Matched (m) and mismatched (mm) accuracies on MNLI test, using the best configurations from development.

be simplified while preserving this information by ACCESS, and 72% by NTS.

## 5 Results and Discussion

Tables 3 and 4 summarize our results on TACRED for the two distinct TS methods. Because we tuned the hyper parameter  $p$  for MNLI, we report results on both development and test for this task (Tables 5 and 6, respectively). Further, for MNLI we also report average performance (and standard deviation) for 3 runs, where we select a different sample to be simplified in each run. This is not necessary for TACRED; for this task we simplified *all* data points that preserved critical information i.e., the two entities participating in the relation.<sup>4</sup>

**Input simplification at prediction time:** Tables 3 and 4 show that simplifying inputs at test time does not yield improvements (compare the *Original* column with the third one). There are absolute decreases in performance of 1.38–2.58% and 1.67–2.80% in F1 on TACRED for ACCESS and NTS systems, respectively (subtract column 3 from column 2 in rows 1 and 4).

Similarly, on MNLI, the performance on simplified inputs is lower than the classifier tested on the original data. The performance drops on MNLI are more severe (11.68–49.53% and 11.70–49.55% in matched and mismatched accuracies) (subtract column 1 from column 2 in row 1 and row 3 in Table 6 pairwise). We hypothesize that this is due to the quality of simplifications in MNLI being lower than those in TACRED. In the latter situation

<sup>4</sup>This is not possible for MNLI, where it is unclear which part of the text is critical for the task.

|   | Gold Data  | Our Approach   | Baseline                   |
|---|--|--|----------------------------|
| 1 | <b>P:</b> In the apt description of one witness, it drops below the radar screen ... you don't know where it is. <b>H:</b> It is hard for one to realize what just happened. <b>Gold Label:</b> Entailment                                 | <b>P:</b> It drops below the radar screen ... you don't know where it is. <b>H:</b> It is hard for one to find what just happened. <b>Predict:</b> Entailment  | <b>Predict:</b> Neutral    |
| 2 | <b>P:</b> The tourist industry continued to expand, and though it the top two income earners in Spain, was ... consequences. <b>H:</b> Tourism is not very big in Spain. <b>Gold Label:</b> Contradict                                     | <b>P:</b> The tourist industry continued to expand, and ... top two income earners in Spain. ... consequences. <b>H:</b> Tourism is very big in Spain. <b>Predict:</b> Entailment  | <b>Predict:</b> Contradict |
| 3 | <b>P:</b> This site includes a list of all award winners and a searchable database of Government Executive articles. <b>H:</b> The Government Executive articles housed on the website are not able to be searched. <b>Ans:</b> Contradict | <b>P:</b> This site includes a list of all award winners and a searchable database of Government Executive articles. <b>H:</b> The Government Executive articles are not able to find the website to be searched. <b>Ans:</b> Contradict | <b>Ans:</b> Neutral        |

Table 7: Qualitative comparison of the outputs from our approach (text simplification by ACCESS) and the respective BERT baseline on the original MNLI data. *P*, *H* indicates premise and hypothesis.

we could apply a form of quality control, i.e., by accepting only the simplifications that preserve the subject and object of the relation. To illustrate the benefits/dangers of text simplification, we show a few examples where simplification improves/hurts MNLI output in Table 7.

**Augmenting training data:** As shown in row 3 and 6 in Table 3 and 4, all methods trained on augmented data yield consistent performance improvements, regardless of the RE classifier used (LSTM or SpanBERT) or TS method used (ACCESS or NTS). There are absolute increases of 1.30–1.82% F1 for ACCESS and 0.70–1.98% F1 for NTS on (substract row 1 from row 3 and row 4 from row 6 for ACCESS and NTS respectively). The best configuration is when the original training data is augmented with all data points that could be simplified while preserving the subject and object of the relation (rows 4 and 8 in the two tables). These results confirm that TS systems can provide additional, useful training information for RE methods.

Similarly, on MNLI, the classifier trained using augmented data outperforms the BERT classifier that is trained only on the original MNLI data. For two TS systems, ACCESS and NTS, we observe performance increases of 0.59–0.65% matched accuracy, and 0.50–0.62% mismatched accuracy (compare rows 1 vs. 2, and row 3 vs. 4 in Table 6). This confirms that TS as data augmentation is also useful for NLI.

All in all, our experiments suggest that our data augmentation approach using TS is fairly general. It does not depend on the actual TS method used, and it improves three different methods from two different NLP tasks. Further, our results indicate that our augmentation approach is more beneficial for tasks with lower resources (e.g., TACRED),

but its impact decreases as more training data is available (e.g., MNLI).

## 6 Conclusion

We investigated the effects of neural TS systems on downstream NLP tasks using two strategies: (a) simplifying input texts at prediction time, and (b) augmenting data to provide machines with additional information during training. Our experiments indicate that the latter strategy consistently helps multiple NLP tasks, regardless of the underlying method used to address the task, or the neural approach used for TS.

## Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the World Modelers and HABITUS programs. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee, and is managed in accordance with its conflict of interest policies.

## References

- Anja Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. A systematic review of reproductibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393.
- Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Gutttag. 2020. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*.
- R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. *Motivations and methods for text simplifica-*

- tion. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Mithun Ghosh, Sahil Hassan, and Prama Debnath. 2021. Ensemble based neural network for the classification of mura dataset. *Journal of Nature*, 4:1–5.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. *arXiv preprint arXiv:2005.02324*.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013)*, pages 59–73.
- Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.
- Reno Kriz, Joao Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-weighted loss and diverse reranking for sentence simplification. *arXiv preprint arXiv:1904.02767*.
- Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2020. Controllable text simplification with explicit paraphrasing. *arXiv preprint arXiv:2010.11004*.
- Louis Martin, Benoît Sagot, Eric de la Clergerie, and Antoine Bordes. 2019. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.
- Makoto Miwa, Rune Saetre, Yusuke Miyao, and Jun’ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796.
- Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2017. A sentence simplification system for improving relation extraction. *arXiv preprint arXiv:1703.09013*.
- 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.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- David Pellow and Maxine Eskenazi. 2014. An open corpus of everyday documents for simplification tasks. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 84–93.
- Luz Rello, Ricardo Baeza-Yates, and Horacio Saggion. 2013. The impact of lexical simplification by verbal paraphrases for people with and without dyslexia. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 501–512. Springer.
- Jordan Schmidek and Denilson Barbosa. 2014. Improving open relation extraction via sentence restructuring. In *LREC*, pages 3720–3723.
- Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.
- Punardeep Sikka, Manmeet Singh, Allen Pink, and Vijay Mago. 2020. A survey on text simplification. *arXiv preprint arXiv:2008.08612*.
- Sanja Štajner and Maja Popović. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.
- Hoang Van, David Kauchak, and Gondy Leroy. 2020. Automets: The autocompleter for medical text simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1424–1434.
- Hoang Van, Vikas Yadav, and Mihai Surdeanu. 2021. Cheap and good? simple and effective data augmentation for low resource machine reading. *arXiv preprint arXiv:2106.04134*.
- David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352.
- Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

- 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*, 3:283–297.
- Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.
- Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.
- Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.
- Sanqiang Zhao, Rui Meng, Daqing He, Saptono Andi, and Parmanto Bambang. 2018. Integrating transformer and paraphrase rules for sentence simplification. *arXiv preprint arXiv:1810.11193*.