DEIM Forum 2021 H14-1

Leveraging word representation for text simplification evaluation

Tianchi ZUO† and Naoki YOSHINAGA††

† The University of Tokyo, 7–3–1, Hongo, Bunkyo-ku, Tokyo 113–8656, Japan
†† Institue of Industrial Science, The University of Tokyo, 4–6–1 Komaba, Meguro-ku, Tokyo 153–8505, Japan
E-mail: †, ††{ztc, ynaga}@tkl.iis.u-tokyo.ac.jp

Abstract Although automatic evaluation metrics enable reproducible evaluation, existing metrics for text simplification correlate poorly with human judgements due to inconsideration of semantic information. This paper proposes methods of leveraging word representations that are meant for capturing both semantics and simplicity. Concretely, we utilize RoBERTa's context-aware word representations to compute semantic similarity between the input and the system output in addition to similarity between the reference and the system output. For evaluating the simplicity, we perform fine-tuning to the RoBERTa by multi-task learning with a self-supervised Simplified-sentence Identification task. We evaluate our methods by seeing the correlation between our evaluation scores and existing human ratings on Turkcorpus and ASSET dataset.

Key words text simplification, evaluation metric

1 Introduction

It is a burden to read sentences that include complex grammatical structures and unfamiliar vocabularies. Text simplification is a text generation task that aims to output simpler and more understandable text [2]. Text simplification has been widely used in education and medical fields, such as simplifying current affairs news for non-native readers [8] and for those who suffer from cognitive diseases such as dyslexia [10] and aphasia [3]. Also, in the field of natural language processing, text simplification can reduce the complexity of text in solving other NLP tasks [4, 7].

The evaluation method of text simplification has always been a concern since the text simplification is quite different from other text generation tasks. One of the major differences is that the simplified sentence should preserve the core meaning of the original sentence, and the comparison between them is thereby necessary rather than just comparing a system output and the reference outputs as other text generation tasks. Besides, how to measure the simplicity of long text is also a challenge [12, 14].

Besides automatic metrics for other text generation tasks, e.g., BLEU [9], new methods such as SARI [14] have been developed for evaluation of text simplification. However, Alva et al. [1] conduct experiments and show a weak correlation between human judgement and existing metrics. Very recently, different from the aforementioned metrics which only take advantage of surface-form information of sentences, metrics based on word representations such as BERTScore [15] have been proven to be effective on text generation tasks like machine translation. However, to our best knowledge, there are no embedding-based metrics for the evaluation of text simplification.

In this paper, we proposed leveraging word representation for the evaluation of text simplification by adapting BERTScore to the text simplification task by adding three modifications. First, we employ simplicity-aware word weighting to BERTScore to make the metric be aware of simplicity. Second, we proposed to incorporate the original sentence (input) into consideration for leveraging the information in input. Finally, we tried to optimize embeddings obtained from the pre-trained model fine-tuned by the simplified-sentence identification task.

We evaluate our methods by computing the correlation between published human ratings [1] and scores obtained by our methods. The experimental results show that the proposed method outperforms existing methods on human ratings. We also do ablation tests to see the performance of our methods on two criteria: meaning preservation and simplicity.

2 Related work

In this section, we introduce existing work on the evaluation of text simplification.

2.1 Human judgement

Human judgement is the most ideal method for the evaluation of text simplification [14]. Generally, people judge a simplified sentence based on the three aspects: meaning preservation (or adequacy), grammaticality (or fluency), and simplicity. Humans will score the simplified sentences subjectively using the Likert scale or continuous score. After that, data processing is performed on the collected scores, such as averaging the score of the three aspects, to get the final evaluation of the sentence. It is worth mentioning that human evaluation usually scores three aspects separately, so the final form is a tuple containing three scores. The demerit of human judgement is that it is not reproducible for comparing and tuning models, and a lot of human labor, time, and cost is required.

2.2 Automatic evaluation

To solve the issues in human judgement, recent years have witnessed some automatic evaluation metrics for the evaluation of text simplification. Automatic metrics rate the simplified sentences by comparing system outputs with the input and the reference outputs. Existing methods tend to give sentences a single score rather than scoring each aspect separately. Next, we introduce two commonly used evaluation metrics for the evaluation of text simplification.

2.3 BLEU

BLEU [9] is an exact matching metric designed for machine translation at first and later was used in other text generation tasks like data-to-text generation and dialog generation. In text simplification, it is used for evaluating the meaning preservation and grammaticality of system outputs based on partial matching between output and multiple references. Although it is not designed for text simplification, existing studies show a high correlation between BLEU's score and meaning preservation and grammaticality. However, in terms of simplicity, BLEU is not appropriate from either lexical or structural aspects.

2.4 SARI

While the automatic metrics are easy to reproduce and fast to obtain, the correlation between human judgement is far from high, especially for simplicity. SARI is an evaluation metric dedicated to text simplification proposed by Xu in 2016 [14]. The computation is based on the number of successful editing operations among input, system output, and multiple reference outputs. SARI defines three editing operations. The first operation is ADD. Generally, SARI will count all the n-grams that appear in both output and reference but not in input. These n-grams are seen to be good simplification added by the simplification system. The second operation is KEEP. For this operation, SARI will count all the n-grams that appear in the input, output, and references. These n-grams are the necessary part of the original sentence, and they should remain after simplification. The third operation is DEL. For this operation, SARI will count all the n-grams that appear in input but neither in output nor in references. These n-grams may be an unnecessary part of the original sentence or parts need to be simplified. Xu et al. conduct experiments on TurkCorpus and showed that SARI achieved a correlation of about 0.34 in Simplicity, which is better than other metrics. However, Alva conduct [1] experiments in ASSET which show that SARI falls behind BLEU in all aspects, which shows that SARI is not suitable for the dataset with multiple simplification operations.

2.5 Embedding-based evaluation metrics

In recent years, there have been some evaluation metrics using contextual word embeddings, such as RUBER [13] in Dialog generation task, and BERTScore [15] in machine translation. However, text simplification is different from others. Firstly, while evaluating a simplified sentence, only comparing output and multiple references are not enough because whether the output can express the same meaning as the input (maybe omit the least important information) or not is very important. So the input sentence must be exploited. Also, the most different part between text simplification and other text generation tasks is the criterion of simplicity. Existing studies show that simple and surface clues have little correlation with the simplicity of sentences. Thus, how to measure simplicity is the key to the evaluation metric of text simplification.

2.6 BERTScore

BERTScore [15] is the recent embedding-based evaluation metrics for text generation tasks. BERTScore computes the sum of pairwise cos-similarities of token's contextual embeddings to yield the similarity of the system and reference outputs.

Formally, given a system output $\boldsymbol{x} = w_1^k$ (k is the number of tokens in $\boldsymbol{x}$) and a reference output $\hat{\boldsymbol{x}}$, firstly we obtain the tokenized embedding sequence of each sentence, such as $\langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \rangle$ and $\langle \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{k'} \rangle$. Then for each token in one sentence, we compute the cosine similarities of the tokens' embedding with all the tokens' embeddings from another sentence. And then we take the max value of all the similarities so that every token is matched to the most similar token in another sentence. We compute precision by matching a token in $\boldsymbol{x}$ and tokens in $\hat{\boldsymbol{x}}$, and compute recall by matching a token in $\hat{\boldsymbol{x}}$ and tokens in $\boldsymbol{x}$ and the F_1 score is computed by combining the precision and recall, as follows:

$$P_{\text{BERT}} = \frac{1}{|\hat{\boldsymbol{x}}|} \sum_{\hat{\boldsymbol{x}}_j \in \hat{\boldsymbol{x}}} \max_{\boldsymbol{x}_i \in \boldsymbol{x}} \boldsymbol{x}_i^\top \hat{\boldsymbol{x}}_j \tag{1}$$

$$R_{\text{BERT}} = \frac{1}{|\boldsymbol{x}|} \sum_{\boldsymbol{x}_i \in \boldsymbol{x}} \max_{\hat{\boldsymbol{x}}_j \in \hat{\boldsymbol{x}}} \boldsymbol{x}_i^\top \hat{\boldsymbol{x}}_j \tag{2}$$

$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \tag{3}$$

BERTScore is reported to be effective in text generation tasks such as machine translation and image captioning. However, it is not directly applicable to text simplification.

3 Methodology

In this section, we introduce our methods of leveraging word representation for the evaluation of text simplification. Concretely, we add three modifications to BERTScore in order to make the metric to be simplicity-aware.

3.1 Simplicity-aware weighting

In order to make BERTScore [15] be aware of simplicity, we propose to inject token-based simplicity knowledge into this metric. Inspired by existing studies on lexicon simplification, we employ simplicity-aware weighting to BERTScore based on the assumption that if the tokens in sentences are simpler, the whole sentence is also simpler to understand. The weight of a token is computed by counting the token frequency of the token in Simple-Wikipedia since the text in Simple-Wikipedia is manually simplified from Wikipedia for easier understanding, tokens with high frequency are seen to be more understandable and simple.

Formally, given a system output $\boldsymbol{x} = w_1^k$ (k is the number of

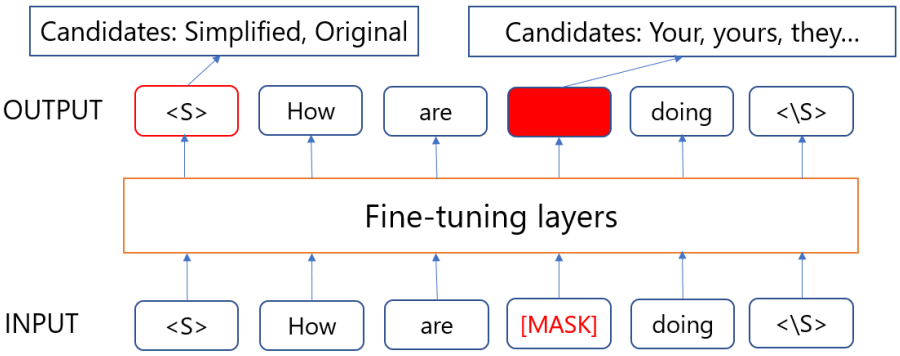


Figure 1: Basic training procedure of proposed multi-task fine-tuning.

tokens in $\boldsymbol{x}$) and a reference output $\hat{\boldsymbol{x}}$, firstly we obtain the tokenized embedding sequence of each sentence as $\langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k'} \rangle$ and $\langle \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_k \rangle$. Then for each token in candidate and reference, we compute the token frequency as

$$\mathrm{tf}(w) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[w \in x^{(i)}\right] \tag{4}$$

while M means all the sentence in simple wikipedia and $\mathbb{I}$ is an indicator function. Similarly to the original BERTScore, we compute the precision and recall to get an F_1 score. Here is an example of a recall score.

$$R_{\mathrm{BERT}} = \frac{\sum_{i}^{k} \mathrm{tf}\,(w_i) \max_{\hat{\boldsymbol{x}}_j \in \hat{\boldsymbol{x}}} \boldsymbol{x}_i^{\top} \hat{\boldsymbol{x}}_j}{\sum_{i}^{k} \mathrm{tf}\,(w_i)} \tag{5}$$

3.2 Considering input

One of the important differences between text simplification and other text generation tasks like machine translation is that the original sentence (input) is another reference of system output besides references made by humans, and incorporating input into consideration is desirable. Existing metrics like SARI have taken advantage of it. In this section, we introduce our modification of incorporating input into BERTScore by calculating how close output is relative to references than the input. For this score, when the output is equal to one of the references, the score will be 1, which means that it is a good simplification. When the output is closer to reference than input, the score will be closer to 0 (bigger than 0). Considering another case when output is equal to the input, in other words, the model does nothing to the input sentence, the score will be 0. when input is closer than output, the score will also be closer to 0. The denominator is a normalization term that takes into account that the degree of simplification differs depending on the input.

$$RelBertScore = 1 - \frac{1 - BertScore(\text{output}, \text{reference})}{1 - BertScore(\text{input}, \text{reference})} \tag{6}$$

3.3 Multi-task fine-tuning

BERTScore is based on contextual word embeddings extracted from pre-trained models like BERT and RoBERTa. These pre-trained models are trained by the masked language modeling task, and the contextual word embeddings do not contain any simplicity-related information of lexicon and sentence. Recent studies show that we can inject more domain-oriented or task-oriented knowledge by fine-tuning pre-trained contextual word embedding model [5].

In order to achieve the goal, we proposed multi-task [11] fine-tuning to make existing pre-trained models' embeddings be aware of simplicity. Generally, there are the main task and an auxiliary task. The main task is the masked language model task, which is the same task when training a contextual pre-trained model like BERT and RoBERTa. We retain this task for training to prevent catastrophic forgetting [6], which will lead to a large-scale missing of semantic information. Besides, we design a simple but useful task called Simplified-sentence Identification, whose goal is to classify if a given sentence is an original sentence or a simplified sentence. The pre-trained model is expected to learn multiple kinds of simplicity knowledge other than just lexicon knowledge by learning the differences between a large amount of original-simplified sentence pairs. For the labeling issue, we can label the sentence automatically by seeing the source of the sentence, if the sentence is from Simple-Wikipedia, we give the label 1, else if the sentence is from Wikipedia, we give the label 0.

Figure 1 shows the training procedure of proposed multi-task fine-tuning. The total loss is the sum of all the prediction losses with the weights of each task. In this study, we set the weight alpha to 0.5 to keep both semantic and simplicity.

4 Experiments

In this section, we measure the quality of proposed metrics by evaluating the correlations between our metrics and human ratings and compare the performance with existing metrics. We choose Pearson as the correlation metric. Next, we introduce the preprocessing pro of human ratings and settings for multi-task fine-tuning.

4.1 Reference datasets

We use two different datasets, Turkcorpus and ASSET for providing different references for original sentences. Both these datasets are using the same original sentences, and for Turkporpus there are 8 references for each original sentence by conducting lexicon

Dataset	Turkcorpus	ASSET
Input	Wikipedia	Wikipedia
Reference	Crowd-sourcing	Crowd-sourcing
Operations	Lexicon	Multiple
#Annotators	8	42
#data	359	359
#References per sent.	8	10

Table 1: A comparison of two dataset for text simplification: Turkcorpus, ASSET.

simplification, while for the latest dataset ASSET, there are 10 references for each original sentence by conducting multiple types of simplification. Table 1 shows the details of two dataset.

4.2 Preprocessing of human ratings

The human ratings are collected from human annotators by Alva et al [1] in their recent work on text simplification dataset. There are 100 original sentences randomly chosen from the test dataset of TurkCorpus [14]. Simplified sentences are produced by existing text simplification models. For each simplified sentence, 15 annotators give their scores of a continuous scale (0-100) for meaning preservation, grammaticality, and simplicity. To get the final score, we normalized the scores of each annotator by their individual mean and standard variation and then take the mean value of all normalized scores.

4.3 Multi-task fine-tuning settings

4.3.1 Dataset

We use WikiLarge [16] as our training data for multi-task fine-tuning. This dataset contains 296,402 original-simplified sentence pairs for training, 2,000 sentence pairs for validation, and 359 sentence pairs for testing. All of the original sentences are extracted from Wikipedia, and simplified sentences in the training dataset are extracted from simple Wikipedia.

4.3.2 Settings

For the auxiliary simplified sentence classification task, we labeled all the original sentences from Wikilarge as 1, and all the simplified sentences from simple Wikipedia as 0, which end up with 592,802 labeled sentences for training. As the same procedure, we got 4,000 labeled sentences for validation.

We use the large version of RoBERTa as a basic contextual word embedding model. Since there are 24 encoder layers, we choose the 17th layer as the embedding layer following previous studies. We fine-tuning this model for 10 epochs, with 32 as batch size and 1e-05 as the learning rate. We choose the checkpoint which gets the best accuracy of the auxiliary task for later evaluation.

4.4 Results

In this section, we compare the performance of proposed metrics and existing metrics. Since there are precision, recall, and F_1 score for metrics based on BERTScore, we only show the metric which shows the highest correlation with human ratings. The correlation results on Turkcorpus and ASSET are showing on Table 2 and Table 3 respectively.

Metric	Meaning	Grammar	Simplicity
BLEU	0.46	0.24	0.17
SARI	0.23	0.15	0.22
BERTScore	**0.60**	0.38	0.30
Proposed	0.55	**0.53**	**0.42**

Table 2: Experimental results on Turkcorpus.

Metric	Meaning	Grammar	Simplicity
BLEU	0.59	0.42	0.36
SARI	0.12	0.11	0.25
BERTScore	**0.62**	0.44	0.35
Proposed	0.59	**0.53**	**0.44**

Table 3: Experimental results on ASSET.

Metric	Meaning	Grammar	Simplicity
proposed	0.59	**0.53**	0.44
- weighting (§ 3.1)	**0.73**	**0.53**	0.38
- input (§ 3.2)	0.71	**0.53**	**0.53**
- fine-tuning (§ 3.3)	0.55	0.50	0.39

Table 4: Ablation test.

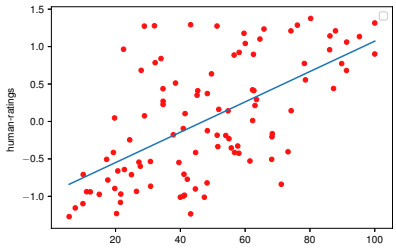
From Table 2 we can see that the proposed method with three modifications beat the original BERTScore in grammaticality and simplicity but failed in meaning preservation. Besides, both of them get a correlation of 0.53 on grammaticality, higher than BLEU and SARI. From the Table 3 we can find a similar trend as we have seen in Turkcorpus. By comparing the results on Turkcorpus and ASSET, we can find that while exact-matching metrics fluctuate on different references, embedding-based metrics are more stable.

We then perform ablation tests to show the effectiveness of our modifications on BERTScore. We only show experimental results on ASSET due to the limitation of space. Table 4 shows the ablation test of the proposed method. We can observe that fine-tuning with simplified sentence prediction helped us evaluate simplicity, whereas simplicity-aware weighting increased correlation for simplicity but decreased correlation for meaning preservation. The current method of considering input did not work effectively as we had expected.

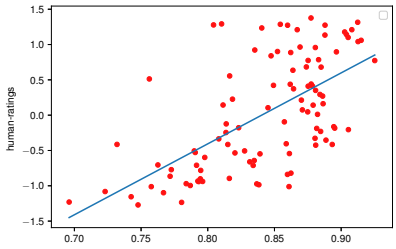
Figure 2 and Figure 3 show some scatter plots of experiment results on Turkcorpus and ASSET in the criterion of meaning preservation and simplicity.

4.5 Examples of evaluation

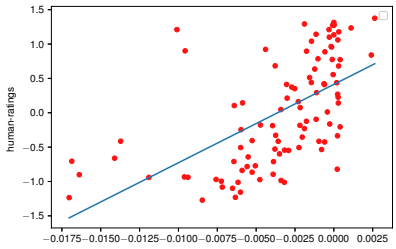
In this section, we give a case study on evaluation examples to show the quality of our studies. Since our methods pay attention to the meaning preservation and simplicity of the evaluation of text simplification, we give a successful and an unsuccessful example of our methods on these criteria by showing the scaled score of human



(a) BLEU (ρ = 0.59)

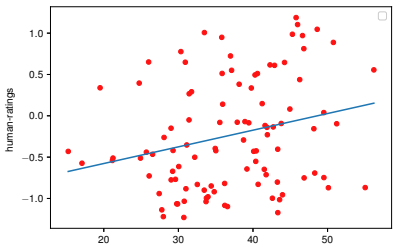


(b) BERTScore (ρ = 0.62)

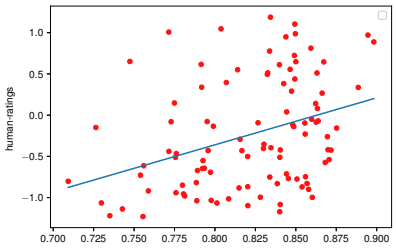


(c) Proposed without simplicity-aware weighting (ρ = 0.73)

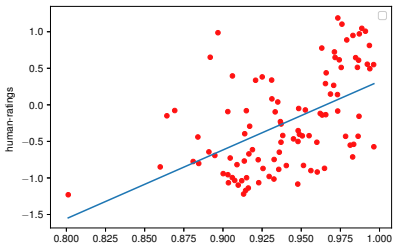
Figure 2: The scatter plots of metrics on meaning preservation.



(a) SARI (ρ = 0.25)



(b) BERTScore (ρ = 0.35)



(c) Proposed without considering input (ρ = 0.53)

Figure 3: The scatter plots of metrics on simplicity.

rating, existing metric, and our proposal. We compute the scaled score for each evaluation metric by scaling output scores of metrics into a number between 0 to 100.

4.5.1 Meaning Preservation

We focus on a successful and an unsuccessful case on meaning preservation between BLEU and the proposed method considering input, which is showing in Table 5. In the first example, we can see that the proposed metric and human rating give similar scores while the BLEU score is much lower. The reason is that semantic-aware metric can easily know that "actually" and "in fact" is quite same while exact matching metric will treat those terms as totally different expression, end up with a lower score.

In the second case, we can find that the output seems to have no relationship to the input since the words are totally different, and human ratings and BLEU give a zero, while for the proposed metric, although it also gives a lower score, it is not enough for this case.

4.5.2 Simplicity

We take a look at a successful and a not successful example on simplicity between SARI and proposed method with simplicity-aware weighting and fine-tuning, which is showing in Table 6. In the first example, we can find that the simplification operations are mainly the deletion of unnecessary contents, and many attributes are deleted in the output sentence. Both human ratings and proposed metrics give a higher score since the core structure of the sentence became clear due to deletion and important words are not deleted. However, SARI gives a lower score just because in given references the content is more than the output, which leads to the penalty of deletion.

In the second example, we can find that the output is totally the same as the input, which reveals that the simplicity score should be lower. SARI gives a penalty to this pattern because of the lack of add score and deletion score, while the proposed metric does not take input into consideration. Since our other proposal that incorporating input into consideration performs lower than this metric, how to correctly consider input for evaluating simplicity is still a challenge.

Input	System outputs	Human	BLEU	Proposed
It is not actually a true louse.	It is not in fact a true louse.	96	39	95
Today NRC is organised as an independent, private foundation.	It is the largest country in the world.	0	0	36

Table 5: Case study on meaning preservation. The first one is a successful example of our proposal and the second one is a unsuccessful example.

Input	System outputs	Human	SARI	Proposed
After graduation he returned to Yerevan to teach at the local Conservatory and later he was appointed artistic director of the Armenian Philarmonic orchestra.	He returned to Yerevan to teach and he was appointed director.	82	47	82
Their eyes are quite small, and their visual Acuity is poor.	Their eyes are quite small, and their visual Acuity is poor.	33	16	90

Table 6: Case study on simplicity. The first one is a successful example of our proposal and the second one is a unsuccessful example.

5 Conclusion

In this study, we draw two issues in evaluating text simplification automatically. Firstly, exact-matching metrics with references are unstable. Besides, The embedding-base metric is not applicable to text simplification. To deal with it, our work adapts BERTScore by employing simplicity-aware word weighting, which shows better performance for measuring simplicity. Incorporating input into consideration, which shows better performance for measuring meaning preservation while failed to beat BERTScore using only output and reference on simplicity. And finally, Adapting word embedding by multi-task fine-tuning leads to better performance on simplicity. Future works mainly focus on how to utilize current proposals, such as find better ways to incorporate input into consideration and find a better auxiliary task for multi-task fine-tuning.

References

[1] Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, 2020.

[2] Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187, 2020.

[3] John A Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. Simplifying text for language-impaired readers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 269–270, 1999.

[4] Han-Bin Chen, Hen-Hsen Huang, Hsin-Hsi Chen, and Ching-Ting Tan. A simplification-translation-restoration framework for cross-domain smt applications. In *Proceedings of COLING 2012*, pages 545–560, 2012.

[5] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

[6] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[7] Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796, 2010.

[8] Gustavo Henrique Paetzold. *Lexical Simplification for Non-Native English Speakers*. PhD thesis, University of Sheffield, 2016.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[10] Luz Rello, Clara Bayarri, Azuki Górriz, Ricardo Baeza-Yates, Saurabh Gupta, Gaurang Kanvinde, Horacio Saggion, Stefan Bott, Roberto Carlini, and Vasile Topac. Dyswebxia 2.0! more accessible text for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pages 1–2, 2013.

[11] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[12] Elior Sulem, Omri Abend, and Ari Rappoport. Semantic structural evaluation for text simplification. 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 685–696, New Orleans, Louisiana, June 2018.

[13] Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

[15] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *Proceedings of the Eighth International Conference on Learning Representations*, 2020.

[16] Xingxing Zhang and Mirella Lapata. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark, September 2017.