Improving Unsupervised Sentence Simplification Using Fine-Tuned Masked Language Models

Anonymous ACL submission

Abstract

Word suggestion in unsupervised sentence simplification is mostly done without considering the context of the input sentence. Fortunately, masked language modeling is a well-established task for predicting the most suitable candidate for a masked token using the surrounding context words. In this paper, we propose a technique that merges pre-trained BERT models with a successful edit-based unsupervised sentence simplification model to bring context-awareness into the simple word suggestion functionality. Next, we show that only by fine-tuning the BERT model on enough simplistic sentences, simplification results can be improved and even outperform some of the competing supervised methods. Finally, we introduce a framework that involves filtering an arbitrary amount of unlabeled in-domain texts for tuning the model. By removing useless training samples, this preprocessing step speeds up the fine-tuning process where labeled data, as simple and complex, are scarce.

1 Introduction

Sentence simplification (SS) is a natural language processing task in which a complex sentence is rewritten, using various edit operations including deletion, lexical substitution, splitting, and reordering, to be easier to be read and understood while preserving its original meaning as much as possible. It is helpful for improving reading comprehension for a broad range of users, e.g. people with linguistic disabilities (Canning et al., 2000; Carroll et al., 1999), non-native speakers (Paetzold and Specia, 2016), and the functionally illiterates (De Belder and Moens, 2010). It can also play a preprocessing role to boost the performance of some language processing models in tasks such as parsing (Chandrasekar et al., 1996) and summarization (Silveira and Branco, 2012).

Initially, SS was considered as a monolingual machine translation task where an input sentence is assumed to belong to a complex version of a certain language and a sequence-to-sequence model translates it into the simpler version of the same language. Recent advancement in unsupervised SS models (Martin et al., 2021; Zhao et al., 2020a) has surprisingly shown that this approach can be as effective as, and even in some cases better than, the ones on the supervised side.

In this paper, we focus on one of the recent successful and controllable edit-based SS methods known as Edit-Unsup-TS (Kumar et al., 2020a). This method iteratively generates multiple simplified candidates by performing word and phrase-level edits on a given complex sentence and picks the best-scored candidate based on a novel scoring function involving fluency, simplicity, and meaning preservation. We modify some of its components including the lexical substitution (LS) suggestion and the scoring function elements to achieve better simplifications. Specifically, we made use of BERT (Devlin et al., 2018) which is a deep transformer-based encoder optimized by two training objectives: masked language modeling (MLM) and next sentence prediction (NSP). MLM is a fill-in-the-blank task where a language model uses surrounding words of a missing word to predict the most suitable candidate.

In order to simplify a complex word within a given sentence, Edit-Unsup-TS suggests alternative words by retrieving synonyms from objectively constructed dictionaries or word embeddings. This means that the candidates are limited to equivalents of the original word and are suggested regardless of their context. For instance, suppose the word *perched* in the input sentence "The cat perched on the mat.". The top three candidates suggested by the classic method are *rested*, *sat*, and *landed*. On the other hand, the BERT model considers surrounding words to suggest *sat*, *laid*, and *was* as alternative words. This form of word suggestion is closer to how humans simplify sentences since it considers context and other possibilities besides synonyms.

To obtain more relevant suggestions, we focus on adapting the BERT language model to the SS task. The idea of fine-tuning a language model on simple data for a better understanding of simplicity has been discussed in previous research (Qiang et al., 2020) but never practiced to the best of our knowledge. We proceed by focusing on two main questions:

- Does the simplicity of fine-tuning data cause improving simplification results?
- How can we boost performance if labeled data, as simple/complex, is scarce in the target language?

Our analysis lead to a novel sentence selection framework that extracts the most beneficial data samples from a set of regular in-domain training sentences. This method requires a few labeled sentences in order to train a classifier that understands simplicity and is capable of separating simple sentences from complex ones in an arbitrary amount of fine-tuning data.

2 Related Work

2.1 Text Simplification

Edit-based simplification techniques are relatively new. For unsupervised SS, Narayan and Gardent (2015) built a pipeline-based framework including separate operations such as deletion, splitting, and lexical simplification which can only be executed in a fixed order. Surya et al. (2019) utilized style-transfer techniques to perform content reduction and lexical simplification. Kumar et al. (2020a) modeled text generation as an iterative search algorithm and designed search objectives specifically for sentences simplification. In this paper, we take advantage of this model's controllability and add a fine-tuned BERT MLM to its classically designed lexical simplification part.

Popular lexical simplification (LS) approaches are rule-based that usually retrieve word synonyms from WordNet (Miller, 1995) for a complex word, and select the simplest possible candidate (Carroll et al., 1998; De Belder et al., 2010). However, rule-based systems do not take a complex word's context into consideration and need a lot of human involvement. In order to avoid the requirement of lexical resources, LS systems based on word embeddings were proposed (Glavaš and Štajner, 2015). They extract the top closest word vectors based on cosine similarity to the initial complex word. Qiang et al. (2020) presented a BERT-based approach only in the context of lexical simplification and did not tackle the fine-tuning aspect.

We apply a similar approach to the sentence simplification problem focusing on fine-tuning the contextual word suggestion model based on a proposed data selection heuristic.

2.2 Data Selection

Selection and augmentation of data for fine-tuning a transformer model has been explored in natural language processing research (Moore and Lewis, 2010; Ruder and Plank, 2017; Kumar et al., 2020b; Rashid and Amirkhani, 2021). The motivation behind this task is that all data points from a source domain are not equally useful for fine-tuning a model and irrelevant samples can add noise and cause overfitting. Dai et al. (2019) focused on identifying the most suitable corpus to pre-train a language model for the task of named entity recognition.

Khandelwal et al. (2019) introduced kNN-LM that allows easy domain adaptation of pre-trained language models by only adding a datastore per domain. Yilmaz et al. (2019) found that fine-tuning BERT on a number of out-of-domain datasets can be beneficial to the ad hoc document retrieval task. Nogueira et al. (2020) confirmed this finding and further improved the zero-shot fine-tuning effectiveness. Ma et al. (2019) presented a novel two-step domain adaptation framework based on curriculum learning and domain-discriminative data selection. Our study is related to classifying each sentence from a collection of in-domain textual data into one of two simple or complex categories and utilizing the simple sentences to fine-tune BERT and improve simplification results.

3 Proposed Method

We first modify Edit-Unsup-TS (Kumar et al., 2020a) by applying the context-awareness of BERT as well as representing the candidate sentences using SentenceBERT (Reimers and Gurevych, 2019) to be used in the scoring function. Then, we present a framework for fine-tuning the BERT model by selecting the appropriate instances from an arbitrary amount of fine-tuning data.

3.1 Modified Edit-Unsup-TS

In order to create simplified candidate sentences from a given complex sentence, Edit-Unsup-TS uses four main edit operations, namely removal (RM), extraction (EX), lexical substitution (LS), and reordering (RO).

The LS operation, which we will modify, follows a rule-based approach. For each phrase, it identifies the most complex word according to the inverse document frequency (IDF) score and generates all possible substitutes using the following two-step strategy:

1. Obtaining the union of WordNet synonyms and the most similar words retrieved from Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) embeddings.

2. Filtering out candidate words that do not meet some predefined semantic and grammatical conditions, such as having the same part-of-speech and dependency tree tags as the complex word.

Besides being expensive to produce, the set of synonym words retrieved from linguistic resources like WordNet does not consider the context. In contrast, an MLM treats the whole sentence as input and is likely to give more appropriate suggestions. Also, the suggestions are grammatically correct and do not require any manual filtering. We follow the approach proposed by Qiang et al. (2020) which masks the current complex word within the sentence and joins the result to the original sentence by a [SEP] token. This helps output words to be more relevant to the original word. The BERT suggestions are used for generating candidate sentences if they are more frequent than the original complex word, calculated based on their log-based IDF values.

After generating all candidate sentences, they are evaluated by a product-of-experts scoring (Hinton, 2002). One of the elements used in this scoring is the cosine similarity between the embedding vectors of the generated candidate sentence and the original sentence calculated based on the weighted average of individual word embeddings. If the resulting similarity is less than a certain threshold, the final score for that candidate will be set to 0 and it is practically ignored. We replace this average embedding method with SentenceBERT, a modification of the pre-trained network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity.

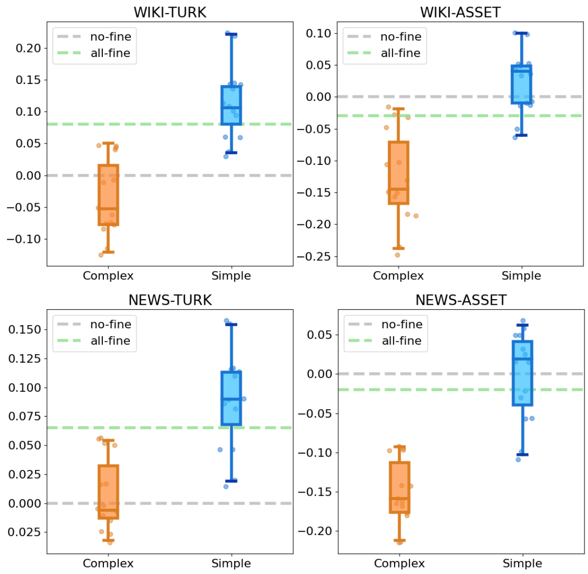

Figure 1: SARI gain of fine-tuning BERT MLM on randomly picked batches of complex and simple sentences from Wikilarge (top) and Newsela (bottom) training sets. Simplifications are performed on TurkCorpus (left) and ASSET (right) validation sets. The results of fine-tuning on all available training data are labeled as *all-fine*. Higher is better.

3.2 Fine-tuning Framework

Fine-tuning is a method for fitting pre-trained models to a target domain. Here, our target domain is a simpler version of the English language. Our assumption is that the BERT MLM will learn to prioritize simpler terms in its suggestions if it is fine-tuned on a considerable number of simplistic sentences. We test this assumption by randomly picking multiple batches of simple and complex sentences from labeled simplification corpora and observing their fine-tuning effects on Edit-Unsup-TS performance. Results shown in Figure 1 show that, in general, fine-tuning on simple sentences will enhance simplification quality while complex sentences could even have negative impacts (details of this experiment are presented in §4.2).

Unfortunately, this is only possible if a large number of labeled sentences are available, where the simple sentences are already separated from the complex ones. To address this issue, we propose a framework that requires a few labeled sentences in order to learn to distinguish simple sentences from complex ones. The learned model is then ex-

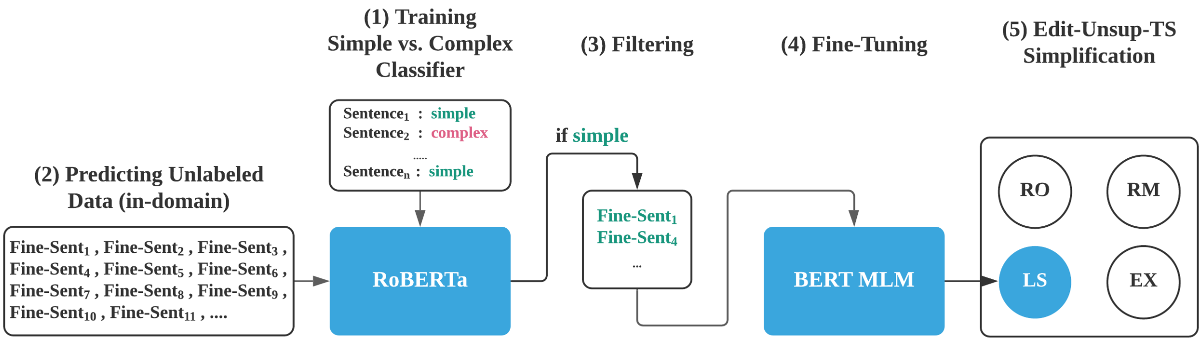

Figure 2: The architecture of the proposed sentence selection framework. (1) A RoBERTa classifier is trained on a small simple/complex labeled corpus. (2,3) A large number of unlabeled texts are filtered using the trained model. (4) The detected simple sentences are handed to the BERT MLM fine-tuning process. (5) The final BERT model is used for sentence simplification.

ploited to extract simple sentences from unlabeled in-domain texts, which are easy to gather. Figure 2 shows an overall view of the proposed framework.

Training. This part of the proposed procedure is essentially training a standard binary text classifier. The idea behind this step comes from the classic definition of sentence simplification. It was treated as a monolingual machine translation task, with *original* and *simplified* as *source* and *target* languages, respectively (Alva-Manchego et al., 2020b). Since the main principle of language detection is to recognize common words and expressions of the target language, we can implement a model capable of distinguishing the simple and complex versions of a certain language. We achieved this by adding a classifier layer to the RoBERTa pre-trained model (Liu et al., 2019). This model has shown substantially improved performance in text classification compared to the base BERT model by training for longer with bigger batches and more data. The labeled dataset required for this step is relatively small and offers good generalization.

Selecting. After preparing the classifier, it should be able to recognize the patterns of simplicity in a given sentence and label it as either *simple* or *complex*. This enables us to input any amount of in-domain text gathered from the internet and extract its simple sentences for fine-tuning. If the assumptions and implementation are correct, these sentences should be more beneficial than the unfiltered data. This is investigated in §4

Fine-tuning. An out-of-the-box transformer model like BERT typically treats domain-specific words in the target corpus as rare tokens, which can negatively affect the resulting performance. By fine-tuning the language model on in-domain data, we can boost the performance in downstream tasks. This aligns with our method of selecting simple sentences based on vocabulary and dialect. During the training process, simplistic tokens will be randomly replaced by a [MASK] placeholder more frequently than usual. Predicting these words would encourage the model to prioritize simpler vocabulary at its suggestion ranking, which will affect the generation of simplified candidates.

4 Experiments

4.1 Metrics and Datasets

We use the EASSE framework[1] (Alva-Manchego et al., 2019) to analyze the quality of our results. Evaluation metrics are described below.

SARI. Introduced in (Xu et al., 2016), it measures simplicity changes based on the words added, deleted, and kept by the system and computes the average F1 score for these operations. This is currently the primary measure for evaluating simplification models.

FKGL: A linear weighted formula that relies on the average sentence lengths and the number of syllables per word. It measures the ease of reading a text (Kincaid et al., 1975).

Table 1 shows the statistics of the datasets used for training and evaluation of our method. In the following, we present more details about these datasets.

WikiLarge: This is the largest Wikipedia complex-

[1] Easier Automatic Sentence Simplification Evaluation - available at https://github.com/feralvam/easse

Dataset	Type	Original	Refs.
WikiLarge	Train	296,402	1
Newsela	Train	28,557	4
TurkCorpus	Validation	2000	8
	Test	359	8
ASSET	Validation	2000	10
	Test	359	10

Table 1: Simplification corpora that are used in the experiments. *Original* refers to the number of complex (source) sentences, and *Refs.* indicates the number of simplified versions provided for each source sentence.

Dataset	Class	Prec.	Recall	F1
WikiLarge	Complex	0.72	0.68	0.70
	Simple	0.69	0.73	0.71
Newsela	Complex	0.88	0.78	0.83
	Simple	0.79	0.89	0.84

Table 2: Evaluation of the simple vs complex classifier trained on WikiLarge and Newsela.

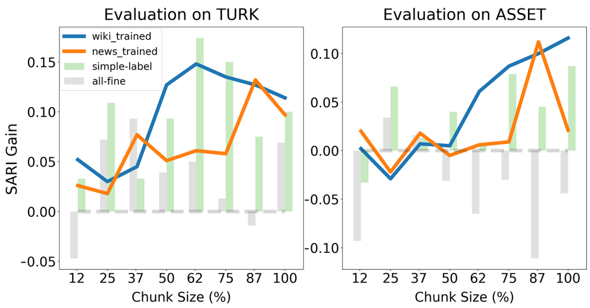

Figure 3: SARI gain of fine-tuning BERT MLM on selectively picked simple sentences from chunks of the fine-tuning data. *Simple-label* and *all-fine* refer to fine-tuning on human-annotated simple sentences and the entire chunk, respectively. Simplifications are performed on TurkCorpus (left) and ASSET (right) validation sets.

to-simple parallel corpus compiled by (Zhang and Lapata, 2017).It has a massive number of samples and fulfills our need for simple and complex labels. Additionally, since it is a parallel dataset, every original (complex) sentence is mapped to its simplified version. This feature is not necessary for classifier training in our fine-tuning framework since we only focus on finding patterns of simplicity.

Newsela: Introduced by Xu et al. (2015), this corpus includes thousands of news articles professionally leveled to different reading complexities.[2] The original article is leveled as zero, and the simplified versions take levels 1 to 4 (the highest being the simplest). These simplifications were produced manually by professional editors, considering children of different grade levels as the target audience.

TurkCorpus: This is a multi-reference dataset for the evaluation of sentence simplification in English (Xu et al., 2016). The dataset consists of sentences from the Parallel Wikipedia Simplification corpus. Each sentence is associated with 8 crowd-sourced simplifications that focus on only lexical paraphrasing, meaning there is no deletion or sentence splitting.

ASSET: Conducted by Alva-Manchego et al. (2020a), this dataset uses the same sentences from TurkCorpus, while each sentence is associated with 10 human-written simplifications. However, the simplifications in ASSET encompass a variety of rewriting transformations.

4.2 Fine-tuning on Random Samples

This experiment was introduced in §3.2. Here, we discuss it in more detail. To see the effect of fine-tuning on simple and complex sentences, we randomly pick 20,000 sentences from each class of our training datasets, independently. Next, we fine-tune BERT on each batch and pass it to Edit-Unsup-TS to simplify both of the evaluation sets. We repeat this process 15 times for a more reliable judgment. Sentences are allowed to be shared to avoid overfitting to a certain configuration. Figure 1 shows the results. It is clear that, on average, fine-tuning with simpler data is more beneficial than fine-tuning with complex ones.

4.3 Training Simple vs Complex Classifier

The Huggingface library (Wolf et al., 2019) is used for fine-tuning a RoBERTa-based classifier to distinguish simple sentences from complex ones. To train the classifier, we selected two different simplification datasets. WikiLarge contains 296,402 original sentences and provides one simplified reference per each. However, the Newsela corpus offers four references that incrementally simplify the previous version. To address this issue, we assumed the original sentence (V0) and the first modification (V1) to be complex and the last two versions (V3 and V4) to be simple.

After these changes, we shuffled both datasets and grabbed small but equal subsets since our goal

[2]This dataset is not publicly available and can be requested from `https://newsela.com/data/`.

	TurkCorpus		ASSET	
	SARI ↑	FKGL ↓	SARI ↑	FKGL ↓
Complex	26.29	10.01	20.73	10.01
Supervised Models				
Hybrid (Narayan and Gardent, 2014)	31.50	**5.17**	34.65	**5.17**
NTS-SARI (Nisioi et al., 2017)	36.10	8.18	34.02	8.18
Dress-LS (Zhang and Lapata, 2017)	36.97	7.66	36.59	7.66
EditNTS (Dong et al., 2019)	37.65	8.37	34.94	8.37
PBMT-R (Wubben et al., 2012)	38.04	8.84	34.63	8.84
DMASS-DCSS (Zhao et al., 2018)	39.92	7.70	38.67	7.70
ACCESS (Martin et al., 2020a)	**41.38**	7.29	40.12	7.29
MUSS (Martin et al., 2020b)	40.85	8.79	**42.65**	8.23
Unsupervised Models				
UNMT (Surya et al., 2019)	34.83	8.97	32.78	8.97
UNTS (Surya et al., 2019)	36.29	7.60	35.19	7.60
BTRLTS (Zhao et al., 2020b)	33.09	8.39	33.95	7.59
Edit-Unsup-TS (Kumar et al., 2020a)	37.27	7.33	36.67	7.33
Edit-Unsup-TS + BERT	37.95	6.51	38.87	6.51
Edit-Unsup-TS + FT-BERT (Labels)	**38.09**	6.44	38.93	6.44
Edit-Unsup-TS + FT-BERT (Selections, Wikilarge-trained)	37.97	**6.39**	**38.94**	**6.39**
Edit-Unsup-TS + FT-BERT (Selections, Newsela-trained)	38.00	6.40	38.93	6.40

Table 3: Results on the TurkCorpus and ASSET test sets. All reported variants of Edit-Unsup-TS were set to perform all operations (RM+EX+LS+RO). FT-BERT (Labels) uses an MLM fine-tuned on human-annotated simple data while FT-BERT (Selections) is based on sentences detected by the simple vs complex classifier. ↑ means higher is better and ↓ means lower is better. All results are calculated based on the EASSE framework resource files.

is to train the classifier on a small number of labeled data. In both cases, the train split contained 9000 instances from each class with 1000 in the validation set and 1000 in the test set.

Evaluation results of these classifiers are reported in Table 2.

4.4 Fine-tuning on Selected Samples

The fine-tuning data needs to be a set of unlabeled in-domain sentences.We used 80,000 randomly selected sentences from WikiLarge without their labels as our fine-tuning data. Each simple vs complex classifier is independently asked to filter this data based on their understanding of sentence simplicity. We then proceed to fine-tune the BERT MLM using their selections. To investigate the effect of fine-tuning data size, this process is performed for eight different sizes of the original fine-tuning data with an interval of 12.5% (10,000 samples). Therefore, the chunk ratios are {0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0} of the original data size. The result are shown in Figure 3.

It is clear that fine-tuning on selected samples is almost always more effective than fine-tuning on all available data for each chunk size. However, one limitation to this method is when the fine-tuning data does not contain enough simple sentences. This leads to the risk of having a worse performance based on our selections rather than the entire data.

4.5 Comparative Results

Finally, we compare the best results of our proposed method with different supervised and unsupervised SS models as shown in Table 3. The first row (Complex) is an evaluation of the source sentences with no simplifications performed.

For unsupervised methods, we compare our results with BTRLTS (Zhao et al., 2020b), UNMT (Surya et al., 2019), UNTS (Surya et al., 2019), and of course, Edit-Unsup-TS (Kumar et al., 2020a). As supervised methods, we considered NTS-SARI (Nisioi et al., 2017), Dress-LS, (Zhang and Lapata, 2017), EditNTS (Dong et al., 2019), PBMT-R (Wubben et al., 2012), DMASS-DCSS (Zhao et al., 2018), and the state-of-the-art models, ACCESS (Martin et al., 2020a) and MUSS (Martin et al., 2020b). Besides the improvements, results show that our approach is on par with most of the supervised methods and even outperforms a few, compared to the original Edit-Unsup-TS.

Our results show that by fine-tuning BERT on sentences labeled as *simple* in the dataset, we can boost the simplification performance of Edit-Unsup-TS. In case of unavailable labeled data, our selections from unlabeled data are almost as effective.

5 Conclusion

We proposed a context-aware word suggestion method for an edit-based sentence simplification technique by adapting the idea of mask language modeling instead of the classic synonym-based approach. Additionally, our experiments showed that fine-tuning the BERT model on simplistic data can positively affect simplification performance. Therefore, we presented a framework to extract simple sentences from unlabeled data by training a RoBERTa classifier on a small number of simple and complex samples. The proposed method is helpful in preprocessing steps, namely filtering out highly complex texts and exploiting useful samples.

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020a. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *arXiv preprint arXiv:2005.00481*.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

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

Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley. 2000. Cohesive generation of syntactically simplified newspaper text. In *International Workshop on Text, Speech and Dialogue*, pages 145–150. Springer.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Citeseer.

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

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

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2019. Using similarity measures to select pretraining data for ner. *arXiv preprint arXiv:1904.00585*.

Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. Lexical simplification. In *Proceedings of ITEC2010: 1st international conference on interdisciplinary research on technology, education and communication*.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Prroceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM; New York.

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*.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval

Technical Training Command Millington TN Research Branch.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020a. Iterative edit-based unsupervised sentence simplification. *arXiv preprint arXiv:2006.09639*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020b. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with bert-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020a. Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020b. Multilingual unsupervised sentence simplification. *CoRR*, abs/2005.00352.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2021. Multilingual unsupervised sentence simplification. *arXiv preprint arXiv:2005.00352*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Robert C Moore and Will Lewis. 2010. Intelligent selection of language model training data.

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

Shashi Narayan and Claire Gardent. 2015. Unsupervised sentence simplification using deep semantics. *arXiv preprint arXiv:1507.08452*.

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

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

Gustavo Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8649–8656.

Mohammad Amin Rashid and Hossein Amirkhani. 2021. The effect of using masked language models in random textual data augmentation. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–5. IEEE.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. *arXiv preprint arXiv:1707.05246*.

Sara Botelho Silveira and António Branco. 2012. Enhancing multi-document summaries with sentence simplification. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, page 1. Citeseer.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

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

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

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3490–3496.

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

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

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020a. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9668–9675.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020b. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9668–9675.