OCHADAI-KYOTO at SemEval-2021 Task 1: Enhancing Model Generalization and Robustness for Lexical Complexity Prediction

Yuki Taya[1], **Lis Kanashiro Pereira**[1], **Fei Cheng**[2], **and Ichiro Kobayashi**[1]

[1]Ochanomizu University, Japan
[2]Kyoto University, Japan
g1620525@is.ocha.ac.jp, kanashiro.pereira@ocha.ac.jp,
feicheng@i.kyoto-u.ac.jp, koba@is.ocha.ac.jp

Abstract

We propose an ensemble model for predicting the lexical complexity of words and multiword expressions (MWEs). The model receives as input a sentence with a target word or MWE and outputs its complexity score. Given that a key challenge with this task is the limited size of annotated data, our model relies on pretrained contextual representations from different state-of-the-art transformer-based language models (i.e., BERT and RoBERTa), and on a variety of training methods for further enhancing model generalization and robustness: multi-step fine-tuning and multi-task learning, and adversarial training. Additionally, we propose to enrich contextual representations by adding hand-crafted features during training. Our model achieved competitive results and ranked among the top-10 systems in both subtasks.

1 Introduction

Predicting the difficulty of a word in a given context is useful in many natural language processing (NLP) applications such as lexical simplification. Previous efforts (Paetzold and Specia, 2016; Yimam et al., 2018; Zampieri et al., 2017) have focused on framing this as a binary classification task, which might not be ideal, since a word close to the decision boundary is assumed to be just as complex as one further away (Shardlow et al., 2020). To alleviate this issue, SemEval-2021 Task 1 (Shardlow et al., 2021a) formulates this task as a regression task, where a model should predict the complexity value of words (Subtask 1) and MWEs (Subtask 2) in context.

This paper describes the system developed by the Ochadai-Kyoto team for SemEval-2021 Task 1. Given that a key challenge in this task is the limited size of annotated data, we follow best practices from recent work on enhancing model generalization and robustness, and propose a model ensemble that leverages pretrained representations (i.e. BERT and RoBERTa), multi-step fine-tuning, multi-task learning and adversarial training. Additionally, we propose to enrich contextual representations by incorporating hand-crafted features during training. Our model ranked 7th out of 54 participating teams on Subtask 1, and 8th out of 37 teams on Subtask 2, obtaining Pearson correlation scores of 0.7772 and 0.8438, respectively.

Task	Domain	Train	Trial	Test
Subtask 1 (single-word)	Europarl	2512	143	345
	Biomed	2576	135	289
	Bible	2574	143	283
	All	7662	421	917
Subtask 2 (MWE)	Europarl	498	37	65
	Biomed	514	33	53
	Bible	505	29	66
	All	1517	99	184

Table 1: Summary of the Complex dataset.

2 Task Description

SemEval-2021 Task 1 provides participants with an augmented version of the CompLex dataset (Shardlow et al., 2020), a multi-domain English dataset with sentences containing words and MWEs annotated on a continuum scale of complexity, in the range of [0,1]. Easier words and MWEs are assigned lower complexity scores, while the more challenging ones are assigned higher scores. This corpus contains a balanced number of sentences from three different domains: Bible (Christodouloupoulos and Steedman, 2015), Europarl (Koehn, 2005) and Biomedical (Bada et al., 2012). The task is to predict the complexity value of single words (Subtask 1) and MWEs (Subtask 2) in context. The statistics of the corpus are presented in Table 1. Our team participated in both subtasks, and the next section outlines the overview of our model.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 17–23
Bangkok, Thailand (online), August 5–6, 2021.

3 System Overview

We focus on exploring different training techniques using BERT and RoBERTa, given their superior performance on a wide range of NLP tasks. Each text encoder and training method used in our model are detailed below.

3.1 Text Encoders

BERT (Devlin et al., 2019): We use the $\text{BERT}_{\text{BASE}}$ model released by the authors. It consists of 12 transformer layers, 12 self-attention heads per layer, and a hidden size of 768.

RoBERTa (Liu et al., 2019b): We use both the $\text{RoBERTa}_{\text{BASE}}$ and $\text{RoBERTa}_{\text{LARGE}}$ models released by the authors. Similar to BERT, $\text{RoBERTa}_{\text{BASE}}$ consists of 12 transformer layers, 12 self-attention heads per layer, and a hidden size of 768. $\text{RoBERTa}_{\text{LARGE}}$ consists of 24 transformer layers, 16 self-attention heads per layer, and a hidden size of 1024.

3.2 Training Procedures

Standard fine-tuning: This is the standard fine-tuning procedure where we fine-tune BERT and RoBERTa on each subtask-specific data.

Feature-enriched fine-tuning (FEAT): During training, we enrich BERT and RoBERTa representations with word frequency information of the target word or MWE. We compute the log frequency values using the Wiki40B corpus (Guo et al., 2020). For MWEs, we compute the log of the average of the frequency of each component word. After applying the min-max normalization to this feature, we concatenate it to the CLS token vector obtained from the last layer of BERT and RoBERTa.

Multi-step fine-tuning (MSFT): Multi-step fine-tuning works by performing a second stage of pre-training with data-rich related supervised tasks. It has been shown to improve model robustness and performance, especially for data-constrained scenarios (Phang et al., 2018; Camburu et al., 2019). Due to the limited size of the data provided for Subtask 2, we first fine-tune BERT and RoBERTa on the Subtask 1 dataset. This model's parameters are further refined by fine-tuning on the Subtask 2 dataset.

Multi-task learning (MTL): Multi-task learning is an effective training paradigm to promote model generalization ability and performance (Caruana, 1997; Liu et al., 2015, 2019a; Ruder, 2017; Collobert et al., 2011). It works by leveraging data from many (related) tasks. In our experiments, we use the MT-DNN framework (Liu et al., 2019a, 2020b), which incorporates BERT and RoBERTa as the shared text encoding layers (shared across all tasks), while the top layers are task-specific. We used the pre-trained BERT and RoBERTa models to initialize its shared layers and refined them via MTL on both subtasks (i.e. Subtask 1 and Subtask 2).

Adversarial training (ADV): Adversarial training has proven effective in improving model generalization and robustness in computer vision (Madry et al., 2017; Goodfellow et al., 2014) and more recently in NLP (Zhu et al., 2019; Jiang et al., 2019; Cheng et al., 2019; Liu et al., 2020a; Pereira et al., 2020). It works by augmenting the input with a small perturbation that maximizes the adversarial loss:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\max_{\delta} l(f(x+\delta;\theta),y)] \quad (1)$$

where the inner maximization can be solved by projected gradient descent (Madry et al., 2017). Recently, adversarial training has been successfully applied to NLP as well (Zhu et al., 2019; Jiang et al., 2019; Pereira et al., 2020). In our experiments, we use SMART (Jiang et al., 2019), which instead regularizes the standard training objective using *virtual adversarial training* (Miyato et al., 2018):

$$\begin{aligned}\min_{\theta} \mathbb{E}_{(x,y)\sim D}[l(f(x;\theta),y)+ \\ \alpha \max_{\delta} l(f(x+\delta;\theta),f(x;\theta))]\end{aligned} \quad (2)$$

Effectively, the adversarial term encourages smoothness in the input neighborhood, and α is a hyperparameter that controls the trade-off between standard errors and adversarial errors.

3.3 Ensemble Model

Ensemble of deep learning models has proven effective in improving test accuracy (Allen-Zhu and Li, 2020). We built different ensemble models by taking an unweighted average of the outputs of a few independently trained models. Each single model was trained on standard fine-tuning, multi-step fine-tuning, multi-task learning, or adversarial training, using different text encoders (i.e. BERT or RoBERTa).

4 Experiments

4.1 Implementation Details

Our model implementation is based on the MT-DNN framework (Liu et al., 2019a, 2020b). We

use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) as the text encoders. We used ADAM (Kingma and Ba, 2015) as our optimizer with a learning rate in the range $\in \{8 \times 10^{-6}, 9 \times 10^{-6}, 1 \times 10^{-5}\}$ and a batch size $\in \{8, 16, 32\}$. The maximum number of epochs was set to 10. A linear learning rate decay schedule with warm-up over 0.1 was used, unless stated otherwise. To avoid gradient exploding, we clipped the gradient norm within 1. All the texts were tokenized using wordpieces and were chopped to spans no longer than 512 tokens. During adversarial training, we follow (Jiang et al., 2019) and set the perturbation size to 1×10^{-5}, the step size to 1×10^{-3}, and to 1×10^{-5} the variance for initializing the perturbation. The number of projected gradient steps and the α parameter (Equation 2) were both set to 1.

We follow (Devlin et al., 2019), and set the first token as the [CLS] token when encoding the input. For Subtask 1, we separate the input sentence and the target token with the special token [SEP]. e.g. [CLS] This was the *length* of Sarah's life [SEP] *length* [SEP]. For Subtask 2, such encoding led to lower performance of our system. Therefore, we consider only the target MWE when encoding the input, e.g. [CLS] *financial world* [SEP].

For each subtask, we used the trial dataset released by organizers as development set (see Table 1). We select the best epoch and the best hyper-parameters using performance (measured in terms of Pearson correlation score) on this development set. We also experimented on saving the best epoch and best hyper-parameters for each domain (Bible, Biomedical and Europarl).

4.2 Main Results

Submitted systems were evaluated on five metrics: Pearson correlation (R), Spearman correlation (Rho), Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2). The systems were ranked from highest Pearson correlation score to lowest. We built several models that use different text encoders and different training methods, as described in Section 3. See Table 2 for the results. First, we observe that ensembling different single models yield better performance on both tasks. Furthermore, models that use feature-enriched representations, multi-task learning, multi-step fine-tuning and adversarial training surpass models that use the standard fine-tuning approach. We detail next the results for each subtask.

For Subtask 1, the single models that used RoBERTa, adversarial training, multi-task learning and feature-enriched representations performed best on the development set. Moreover, saving the best epoch and hyper-parameters for each domain performed better than saving the best epoch and hyper-parameters without domain distinction. Among the single models, the model that performed best on the development set was the model that uses $\text{RoBERTa}_{\text{LARGE}}$ and adversarial training ($\text{RoBERTa}_{\text{LARGE}}(\text{ADV})_{\text{domain}}$ model, with a Pearson score of 0.8441). The second best single model was the model that uses $\text{RoBERTa}_{\text{BASE}}$ and feature-enriched contextual representations ($\text{RoBERTa}_{\text{BASE}}(\text{FEAT})_{\text{domain}}$ model, with a Pearson score of 0.8391). The third best single model was the model that uses $\text{RoBERTa}_{\text{LARGE}}$ and multi-task learning ($\text{RoBERTa}_{\text{LARGE}}(\text{MTL})_{\text{domain}}$ model, with a Pearson score of 0.8371). Thus, we ensemble these three single models in different ways when making our submissions. The ensemble model that performed best on the test set (Ensemble $2_{\text{single\_word}}$) was the model that combined feature-enriched contextual representations ($\text{RoBERTa}_{\text{BASE}}(\text{FEAT})_{\text{domain}}$), adversarial training ($\text{RoBERTa}_{\text{LARGE}}(\text{ADV})_{\text{domain}}$), and multi-task learning ($\text{RoBERTa}_{\text{LARGE}}(\text{MTL})_{\text{domain}}$). This ensemble model obtained development and test set Pearson scores of 0.8570 and 0.7772, respectively.

For Subtask 2, the single models that use $\text{BERT}_{\text{BASE}}$ outperformed models that use RoBERTa, on the development set. Moreover, we noted that using the Subtask 1 dataset as auxiliary dataset by performing multi-step fine-tuning and multi-task learning greatly help to improve the performance. For instance, the $\text{BERT}_{\text{BASE}}(\text{MSFT})$ outperformed the $\text{BERT}_{\text{BASE}}$ model by 0.0405 Pearson correlation points (0.7965 vs 0.8370). The ensemble model that performed best on the test set (Ensemble 1_{MWE}) was the model that combined multi-step fine-tuning and multi-task learning using BERT, i.e. $\text{BERT}_{\text{BASE}}$ (MSFT) and $\text{BERT}_{\text{BASE}}(\text{MTL})$ models, respectively, and multi-task learning using RoBERTa ($\text{RoBERTa}_{\text{LARGE}}(\text{MTL})$ model). This ensemble model obtained development and test set Pearson scores of 0.8461 and 0.8438, respectively. Different from Subtask 1, we observe that saving the best epoch and hyper-parameters for each domain on the development set performed worse than saving the best epoch and hyper-parameters without do-

Training Methods	Ensemble			R	Rho	MAE	MSE	R2
Subtask 1 (Single Word Lexical Complexity Prediction Task)								
$BERT_{BASE}^{dev}$				0.7794	0.7423	0.0664	0.0077	0.1898
$RoBERTa_{BASE}^{dev}$				0.8139	0.7498	0.0628	0.0064	0.4325
$RoBERTa_{BASE}(FEAT)^{dev}$	✓			0.8348	0.7579	0.0603	0.0058	0.6955
$RoBERTa_{BASE}(FEAT)^{dev}{}_{domain}$		✓	✓	0.8391	0.7640	0.0599	0.0057	0.6976
$RoBERTa_{LARGE}^{dev}$				0.8213	0.7629	0.0627	0.0062	0.5381
$RoBERTa_{LARGE}(FEAT)^{dev}{}_{domain}$				0.8218	0.7513	0.0634	0.0063	0.6025
$RoBERTa_{LARGE}(MTL)^{dev}{}_{domain}$		✓	✓	0.8371	0.7694	0.0609	0.0062	0.3640
$RoBERTa_{LARGE}(ADV)^{dev}$	✓			0.8328	0.7760	0.0603	0.0059	0.5509
$RoBERTa_{LARGE}(ADV)^{dev}{}_{domain}$		✓		**0.8441**	**0.7873**	**0.0572**	**0.0054**	**0.7123**
$Ensemble\ 1_{single\_word}^{dev}$	◯			0.8481	0.7825	0.0578	0.0053	0.7175
$Ensemble\ 2_{single\_word}^{dev}$		◯		**0.8570**	**0.7902**	**0.0553**	**0.0050**	**0.7335**
$Ensemble\ 3_{single\_word}^{dev}$			◯	0.8548	0.7816	0.0560	0.0051	0.7300
$Ensemble\ 1_{single\_word}^{test}$	◯			0.7590	0.7174	0.0640	0.0069	0.5719
$Ensemble\ 2_{single\_word}^{test}$		◯		**0.7772**	**0.7313**	**0.0617**	**0.0065**	**0.6015**
$Ensemble\ 3_{single\_word}^{test}$			◯	0.7761	0.7244	0.0622	0.0065	0.6003
Top Team Result (JUST BLUE)$_{single\_word}^{test}$*				**0.7886**	**0.7369**	**0.0609**	**0.0062**	**0.6172**
Subtask 2 (MWE Lexical Complexity Prediction Task)								
$BERT_{BASE}(full\ context)^{dev}$†				0.7903	0.7839	0.0770	0.0090	0.6240
$BERT_{BASE}^{dev}$				0.7965	0.7856	0.0761	0.0086	0.3552
$BERT_{BASE}(FEAT)^{dev}$				0.8166	0.8033	0.0730	0.0080	0.6610
$BERT_{BASE}(MSFT)^{dev}$	✓			0.8370	0.8361	**0.0661**	0.0071	0.5276
$BERT_{BASE}(MSFT)^{dev}{}_{domain}$		✓	✓	**0.8498**	**0.8492**	0.0669	0.0068	0.7099
$BERT_{BASE}(MTL)^{dev}$	✓			0.8176	0.8202	0.0725	0.0081	0.5086
$BERT_{BASE}(MTL)^{dev}{}_{domain}$		✓	✓	0.8442	0.8323	0.0667	**0.0067**	**0.7125**
$RoBERTa_{BASE}^{dev}$				0.7689	0.7659	0.0771	0.0098	0.3767
$RoBERTa_{LARGE}^{dev}$				0.8110	0.8181	0.0737	0.0082	0.4363
$RoBERTa_{LARGE}(MTL)^{dev}$	✓			0.8176	0.8202	0.0725	0.0081	0.5086
$RoBERTa_{LARGE}(MTL)^{dev}{}_{domain}$			✓	0.8341	0.8276	0.0675	0.0075	0.6790
$RoBERTa_{LARGE}(ADV)^{dev}$				0.8119	0.8019	0.0718	0.0080	0.4785
$RoBERTa_{LARGE}(ADV\&MSFT)^{dev}$				0.8247	0.8092	0.0685	0.0076	0.4748
$RoBERTa_{LARGE}(ADV\&MSFT)^{dev}{}_{domain}$		✓		0.8283	0.8176	0.0676	0.0074	0.6858
$Ensemble\ 1_{MWE}^{dev}$	◯			0.8461	0.8441	0.0672	0.0068	0.7080
$Ensemble\ 2_{MWE}^{dev}$		◯		0.8543	0.8444	0.0642	**0.0064**	**0.7270**
$Ensemble\ 3_{MWE}^{dev}$			◯	**0.8571**	**0.8509**	**0.0640**	**0.0064**	0.7267
$Ensemble\ 1_{MWE}^{test}$	◯			**0.8438**	**0.8285**	**0.0660**	**0.0070**	**0.7103**
$Ensemble\ 2_{MWE}^{test}$		◯		0.8376	0.8231	0.0682	0.0076	0.6840
$Ensemble\ 3_{MWE}^{test}$			◯	0.8312	0.8157	0.0708	0.0080	0.6686
Top Team Result (DeepBlueAI)$_{single\_word}^{test}$*				**0.8612**	**0.8526**	**0.0616**	**0.0063**	**0.7389**

Table 2: Comparison of different text encoders and different training methods on the single word lexical complexity prediction task (Subtask 1) and on the MWE lexical complexity prediction task (Subtask 2). Best results for single and ensemble models are highlighted in **bold**. † indicates that we consider the full context surrounding the MWE when encoding the input. In the other models for Subtask 2, we consider only the target MWE. * indicates results obtained from the Task's official leaderboard: (https://competitions.codalab.org/competitions/27420#results). ✓indicates each single model that was used in the ensemble, indicated in each column by ◯.

main distinction. We hypothesize that, due to the small size of the data provided for Subtask 2, saving the best epoch and hyper-parameters without domain distinction might avoid overfitting.

5 Analysis

We briefly analyse our best models' results on the test set for each subtask. Figure 1 (top) shows a comparison between our best ensemble model's predictions for Subtask 1 (Ensemble 2_{single_word}) and the gold answers. We observe that our model often fails to predict correctly in the range where samples have a complexity score below 0.2. We hypothesize this might be due to the skewed distribution of the golden complexity scores for each domain, as shown in Table 4. A possible solution might be to build domain-specific models.

Figure 1 (bottom) shows a comparison between the best ensemble model's predictions for Subtask

Domain	Sentence	Target	Prediction	Label
	Sub-task 1			
Europarl	The Swedish Presidency aims to maintain the debate on animal welfare and good animal *husbandry*.	*husbandry*	0.3270	0.53143
Biomed	We adopted the same strategy to investigate the relative contribution of the 129 *Chromosome* 1 segment and the Apcs gene to each disease trait.	*Chromosome*	0.4865	0.2237
Bible	God has gone up with a *shout*, Yahweh with the sound of a trumpet.	*shout*	**0.2032**	**0.2031**
	Sub-task 2			
Biomed	These studies strongly suggest that the hsp family of proteins has *other functions* in addition to protecting proteins and cells during stress.	*other functions*	0.2564	0.4167
Europarl	What plans does the Commission have to introduce *eco labelling* of 'sustainable' palm oils?	*eco labelling*	0.5277	0.3553
Bible	In the *dry season*, they vanish.	*dry season*	**0.2832**	**0.2857**

Table 3: Examples of successful and poor predictions on the test set by the best ensemble models submitted for each subtask (Ensemble $2_{\text{single\_word}}$ and Ensemble 1_{MWE} models). Successful predictions are highlighted in **bold**.

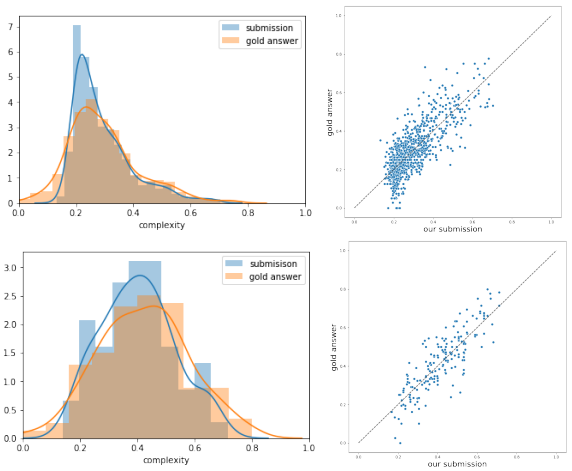


Figure 1: Comparison between the Ensemble $2_{\text{single\_word}}$ and Ensemble 1_{MWE} models' predictions submitted for Sub-task 1 (top) and Sub-task 2 (bottom), respectively, and the gold answers. On the left, we show the distribution of the correct complexity score and our submission. On the right, we show a scatter plot where the x-axis corresponds to our model's predictions and the y-axis corresponds to the gold answers.

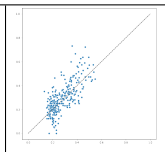
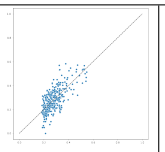
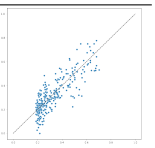
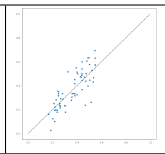
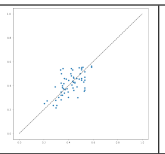
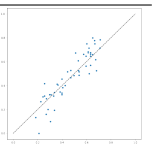
	Bible	Europarl	Biomed
	Sub-task 1		
			
MAE	0.0679	0.0549	0.0638
R	0.7329	0.7213	0.8358
	Sub-task 2		
			
MAE	0.0721	0.0592	0.0667
R	0.8114	0.6374	0.9104

Table 4: Performance of Ensemble $2_{\text{single\_word}}$ and Ensemble 1_{MWE} models on each domain and subtask.

2 (Ensemble 1_{MWE}), and the gold answers. Compared to Subtask 1, the data distribution of the development and test sets of Subtask 2 look more similar, hence a possible reason why the development and test set scores were closer than in Subtask 1 (the best ensemble models obtained development and test set scores of 0.8570 and 0.7772, respectively, in Subtask 1, and 0.8461 and 0.8438, respectively, in Subtask 2). Table 3 shows examples of successful and poor predictions made by Ensemble $2_{\text{single\_word}}$ and Ensemble 1_{MWE} models. Table 4 shows how the performance of these models varies across domains. The Biomedical domain obtained the highest Pearson correlation scores on both subtasks, which indicates that might be a sharper difference between simple and complex words in this corpus (Shardlow et al., 2021b).

6 Conclusion

In this paper, we have presented the implementation of the Ochadai-Kyoto system submitted to the SemEval-2021 Task 1. Our model ranked 7th out of 54 participating teams on Subtask 1, and 8th out of 37 teams on Subtask 2. We proposed an ensemble model that leverages pretrained representations, multi-step fine-tuning, multi-task learning and adversarial training. We also proposed to enrich contextual representations by incorporating hand-crafted features during training. In future efforts, we plan to further improve our model to better handle data-constraint and domain-shift scenarios.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. 2012. Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):1–20.

Oana-Maria Camburu, Vid Kocijan, Thomas Lukasiewicz, and Yordan Yordanov. 2019. A surprisingly robust trick for the winograd schema challenge.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40b: Multilingual language model dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2440–2452.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR (Poster) 2015*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020a. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, and Jianfeng Gao. 2020b. The microsoft toolkit of multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:2002.07972*.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Lis Pereira, Xiaodong Liu, Fei Cheng, Masayuki Asahara, and Ichiro Kobayashi. 2020. Adversarial training for commonsense inference. *arXiv preprint arXiv:2005.08156*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

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

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex: A new corpus for lexical complexity prediction from likert scale data.

Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021a. Semeval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2021)*.

Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2021b. Predicting lexical complexity in english texts. *arXiv preprint arXiv:2102.08773*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132*.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex word identification: Challenges in data annotation and system performance. *arXiv preprint arXiv:1710.04989*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for language understanding. *arXiv preprint arXiv:1909.11764*.