JBNU at SemEval-2020 Task 4: BERT and UniLM for Commonsense Validation and Explanation

Jong-Hyeon Lee and **Seung-Hoon Na**
Computer Science and Engineering, Jeonbuk National University, South Korea
{gus8423,nash}@jbnu.ac.kr

Abstract

This paper presents our contributions to the SemEval-2020 Task 4 Commonsense Validation and Explanation (ComVE) and includes the experimental results of the two Subtasks B and C of the SemEval-2020 Task 4. Our systems rely on pre-trained language models, i.e., BERT (including its variants) and UniLM, and rank 10th and 7th among 27 and 17 systems on Subtasks B and C, respectively. We analyze the commonsense ability of the existing pretrained language models by testing them on the SemEval-2020 Task 4 ComVE dataset, specifically for Subtasks B and C, the explanation subtasks with multi-choice and sentence generation, respectively.

1 Introduction

SemEval-2020 Task 4 aims to evaluate whether a system identifies and rationalizes a given natural language statement to be comprehensible under commonsense knowledge (Wang et al., 2020). Starting from the pilot study (Wang et al., 2019), SemEval-2020 Task 4 consists of three subtasks: 1) Subtask A: differentiating statements that make sense from those that do not, 2) Subtask B and C: selecting a reason or explaining why the statement does not make sense. This paper presents an overview of our systems that were examined for Subtasks B and C, as well as the final results.

Recently, inspired from the success of ELMo (Peters et al., 2018), GPT (Radford et al., 2018), and BERT (Devlin et al., 2019) which demonstrated considerable improvements on various language understanding tasks, including the GLUE benchmark (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016), significant studies have been conducted on pretrained language models, including RoBERTa and XLNet, which efficiently reduce the training and inference costs (Liu et al., 2019; Lan et al., 2019), Transformer-XL, which extends to large contexts (Dai et al., 2019), XLNet (Yang et al., 2019), MASS and UniLM, which support pretrained language models for generation tasks (Song et al., 2019; Dong et al., 2019), DistillBERT for lightweight pretrained models (Sanh et al., 2019), analyzing BERT's syntactic knowledge (Hewitt and Manning, 2019), knowledge-enhanced language models (Peters et al., 2019), cross-lingual pretrained language models (Conneau and Lample, 2019), few-shot learning using language models (Radford et al., 2019; Brown et al., 2020), scaling up pretrained language models (Raffel et al., 2019) and the retrieval-based language model using dense retrieval on the external corpus (Guu et al., 2020). A critical review of the pretrained language models that include classical neural language models (Bengio et al., 2003) has been presented in (Young et al., 2018).

Given the era of pretrained language models, the main design goal of our system on ComVE is to explore the effect of using pretrained language models on Subtasks B and C, both for the understanding and generation tasks. In the top-level design, BERT is selected for Subtask B, and UniLM, a generalized pretrained language model that supports both understanding and generation capabilities, is employed for Subtask C. Our system architectures for Subtasks B and C are summarized as follows.

1. **BERT+FNN** and **BERT+BiLSTM** (**Subtask B**): For Subtask B, we present two types of models, *BERT+FNN* and *BERT+BiLSTM* based on *BERT* (Devlin et al., 2019). Given an input statement,

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 527–534
Barcelona, Spain (Online), December 12, 2020.

we first apply BERT (Devlin et al., 2019) or its variants to the [SEP]-based concatenation of the statement and i-th optional statement. Given k optional statements, *BERT+FNN* transforms the k BERT-encoded representations obtained over all optional statements using a feedforward neural network (FNN) on the last embeddings of [CLS] tokens, to compute the k scores. The probabilities of k optional statements being the reason are computed based on the softmax function. A simple extension is made on *BERT+BiLSTM* to further transform the BERT-encoded representations using bidirectional *long shot-term memory* (LSTM) before applying FNN on the [CLS] embedding[1]. To compare the performances of the BERT variants on Subtask B, we replace the BERT module in *BERT+FNN* and *BERT+BiLSTM* with *RoBERTa* (Liu et al., 2019) or *ALBERT* (Lan et al., 2019). In this paper, all models induced from BERT, RoBERTa, and ALBERT, are referred to as *BERT-style model*s.

2. **UniLM** (**Subtask C**): For Subtask C, we need to use generation capable pretrained language models beyond BERT, because it is difficult to apply BERT to natural language generation (NLG) tasks, due to its bidirectionality nature (Wang and Cho, 2019). Hence, we employ *UniLM* (Dong et al., 2019) which was recently found to be successful in NLG. UniLM employs three language model (LM) tasks for pretraining, consisting of the unidirectional LM toward pretrained language models for NLG tasks (Peters et al., 2018), bidirectional LM (Devlin et al., 2019) and sequence-to-sequence prediction LM tasks, thereby enabling to fine-tune it on natural language understanding (NLU) and NLG tasks. We examine the effect of UniLM on commonsense reasoning for Subtask C.

The remainder of this paper is organized as follows: Section 2 briefly summarizes the data description for the SemEval-2020 ComVE task. Section 3 presents the details of our system architecture, Section 4 provides the preliminary and official experimental results, while our concluding remarks and a description of the future work are presented in Section 5.

2 Data Description for SemEval-2020 ComVE

Each instance in the dataset for explanation subtasks of ComVE is composed of seven sentences $< s, o_1, o_2, o_3, r_1, r_2, r_3 >$. Statement s, which does not make sense, is given as an input sentence, commonly to Subtasks B and C. For Subtask B, o_1, o_2 and o_3 are the three *optional* sentences to explain why the statement does not make sense; the only a single sentence o_i is marked as a correct (positive) reason and the others as negative ones. For Subtask C, r_1, r_2, and r_3, are the additional sentences for *referential reasons* and, are used for training and evaluation. During the preliminary experiment for Subtask B, we found that some of the topics included only two optional sentences[2]. Thus, we excluded those topics from the dataset. In addition, we attached a punctuation mark to the sentences that did not originally end with any punctuation marks.

3 System Description

This section presents detailed descriptions of our system and the methods that used for Subtasks B and C at SemEval-2020 ComVE.

3.1 Model for Subtask-B

Let us recall the definition of Subtask B: given statement s, the system is required to select the correct reason from three optional reasons by examining why statement s does not make sense.

To address Subtask B, we propose two types of BERT-style models, *BERT+FNN* and *BERT+BiLSTM*, whose model architectures are depicted in Figures 1 and 2, respectively, where inputs for BERT are provided at the bottom and the output from BERT is presented at the top right.

[1]Here, unlike the standard BiLSTM, which concatenates the forward and backward representations, we used the forward and backward LSTM separately and perform the mean pooling on the resulting hidden represention.

[2]The topics are 5618th, 7238th, 9941th in the training set and 1998th in the trial set.

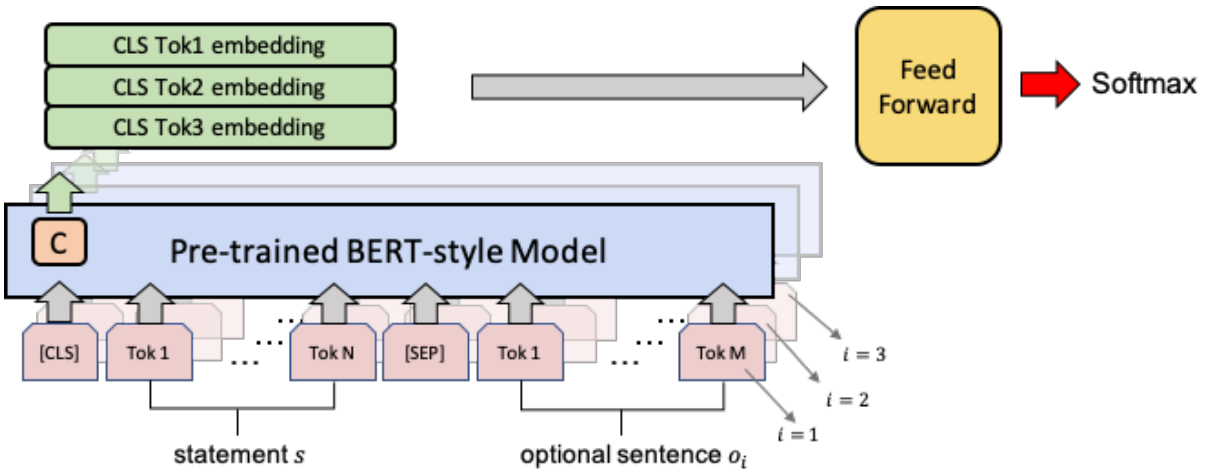

Figure 1: Architecture of BERT+FNN. BERT+FNN only uses [CLS] token embedding to obtain learned representation of sentences from pre-trained BERT-style models.

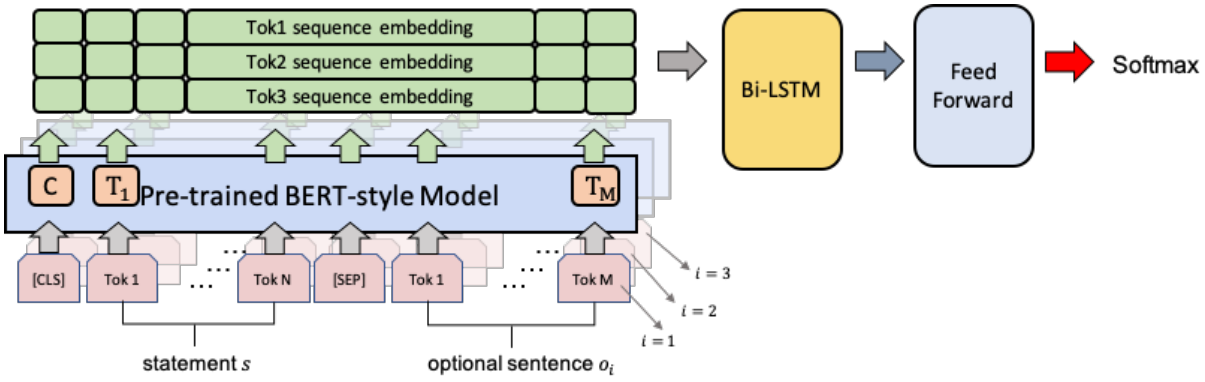

Figure 2: Architecture of BERT+BiLSTM. BERT+BiLSTM uses all token embeddings obtained from pre-trained BERT-style models.

Formally, given an optional sentence o_i, let $\mathbf{W}_i = [\mathbf{w}_{i,0}, \mathbf{w}_{i,1}, \mathbf{w}_{i,2}, \cdots, \mathbf{w}_{i,n}]$ be the sequence of output embeddings of the BERT-style model, where $\mathbf{w}_{i,0}$ denotes the [CLS] token embedding for the i-th sentence, and $\mathbf{w}_{i,t} \in \mathbb{R}^n$ denotes each output embedding[3].

In BERT+FNN, we use only [CLS] tokens embedding, which is the first token embedding acquired from the BERT-style model for classification. The [CLS] token embedding is usually used to represent the whole meaning a given sentence. BERT+FNN is composed of a FNN and the softmax function to select a correct sentence that makes sense.

$$P(correct|i) = \texttt{softmax}_{1 \leq i \leq 3}(\texttt{FeedForward}(\mathbf{w}_{i,0})) \tag{1}$$

which indicates the probability that o_i is the correct reason for why s does not make sense.

In BERT+BiLSTM, we use all token embeddings acquired from the BERT-style models. To obtain sentence representations, we employ a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) architecture. Using the LSTM, given the i-th optional sentence, the hidden state at time t, denoted by $\mathbf{h}_{i,t} \in \mathbb{R}^m$, is computed via

$$\mathbf{h}_{i,t+1}, \mathbf{c}_{i,t+1} = \texttt{LSTM}(\mathbf{w}_{i,t}, \mathbf{h}_{i,t}, \mathbf{c}_{i,t}) \tag{2}$$

where $\mathbf{c}_{i,t}$ denotes the cell state of the LSTM.

[3]For notational simplicity, we often drop the i dependency on the output sequence.

input i	[SOS] s [EOS] o_i [EOS]

Table 1: Input segment representation for BERT on Subtask B, where s is a given statement, $o_i \in \{o_1, o_2, o_3\}$ is an optional sentence, and [SOS] and [EOS] are the special tokens for indicating the starting and ending of a sentence, respectively.

input i	[SOS] "s" does not make sense. [EOS] Because o_i [EOS]

Table 2: Input segment representation using extra words for BERT on Subtask B, where s is a given statement, $o_i \in \{o_1, o_2, o_3\}$ is an optional sentence.

To exploit the contextual information in a bidirectional manner, we process the input embeddings using a bidirectional LSTM, which reads an input in both forward and backward orders. We then perform the mean pooling instead of the concatenation: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, thereby yielding the mean vector of both representations to combine them. In particular, we compute the hidden state at time t, $\mathbf{h}_{i,t} \in \mathbb{R}^m$, for an input embedding of length T, using the following:

$$\mathbf{h}_{i,t+1}^{forward} = \texttt{LSTM}(\mathbf{w}_{i,t}, \mathbf{h}_{i,t}^{forward}, \mathbf{c}_{i,t}^{forward}) \tag{3}$$

$$\mathbf{h}_{i,t+1}^{backward} = \texttt{LSTM}(\mathbf{w}_{T-t}, \mathbf{h}_{i,t}^{backward}, \mathbf{c}_{i,t}^{backward}) \tag{4}$$

$$\mathbf{h}_{i,T} = \texttt{mean}(\mathbf{h}_{i,T}^{backward}, \mathbf{h}_{i,T}^{forward}) \tag{5}$$

The rest of the architecture is the same as that of BERT+FNN, where the feedforward neural network is applied to $\mathbf{h}_{i,T}$ instead of the [CLS] embedding.

3.1.1 Input representation for BERT encoder

The remaining part includes determining the input that should be used for the BERT encoder. In particular, let a test (or training) instance for Subtask B be an entry consisting of 4 sentences $< s, o_1, o_2, o_3 >$. To pass an input to *BERT+FNN* and *BERT+BiLSTM*, we pack a given sentence s and an optional sentence o_i using sentence starting and ending tokens, [SOS] and [EOS], where the two sentences are packed as "[SOS] S1 [EOS] S2 [EOS]," wherein S1 and S2 denote the first and second sequences, respectively. Table 1 presents how an optional sentence is packed as an input with an original sentence s.

To make the input more natural, we optionally further apply a simple preprocessing on the dataset by supplementing *extra words*, such as "because" between s and o_i, as illustrated in Table 2.

3.1.2 Extension using BERT variants

Under the architectures of Figures 1 and 2, we compare three pretrained language models, i.e., BERT, RoBERTa and ALBERT, to examine the effectiveness of the BERT-style models on the commonsense reasoning ability in the setting of Subtask B.

3.2 Model for Subtask C

It should be noted that Subtask C is a type of NLG task, where the goal is to generate reasons for determining why input sentence s does not make sense. To address Subtask C, we use *UniLM*, as mentioned in the introduction, inspired by the recent works wherein UniLM demonstrates promising performance on NLG tasks, such as abstractive summarization, question generation, and generative question answering. While BERT is used mainly for NLU tasks, UniLM provides various types of language models based on its encoders and decoders, which enable it to be fine-tuned for both NLU and NLG tasks, including our addressed Subtask C.

To train UniLM for Subtask C, let a training (or test) instance be an entry consisting of four sentences $< s, r_1, r_2, r_3 >$. The input sentence s is encoded by UniLM and the generation model based on UniLM is fine-tuned for the loss function of Subtask C, such that it generates a correct reason for why the input sentence s does not make sense. For Subtask C, we select only one reason, r_1, from the three possible references.

4 Experimental Results

We use the official release dataset of SemEval Task 4 for the experiments. The dataset is split into train/trial/dev/test sets, and, we use the dev (development) set to obtain the model with the best performance.

4.1 Model training

In our submitted model for Subtask B, the hidden dimension of LSTM was 768. We used the Adam optimizer for BiLSTM and FNN with a learning rate of 1e-3. The number of layers for the LSTM and dropout were 1 and 0.3, respectively. For finetuning BERT, we used the Adam optimizer with a learning rate of 5e-6.

For Subtask C, the submitted model was trained with Adam using $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for optimization. The learning rate and dropout rate were 3e-5 and 0.1, respectively.

4.2 Results for Subtask B

For Subtask B, we first evaluate BERT+FNN across several BERT-style models, i.e., BERT, RoBERTa, and ALBERT. Table 3 presents the results of BERT+FNN on the trial and dev datasets released from the SemEval-2020 organizers. Note that all models for Subtask B use the exactly same evaluation pipeline, which makes them directly comparable. As evident in Table 3, the RoBERTa-large-based model performs significantly well on this task, outperforming other models on the dev set. Our results partly suggest that some of the commonsense knowledge is entailed from the BERT-style models.

Model	Name	Accuracy(%)	
		Tiral-set	Dev-set
BERT	BERT-base-uncased	90.15	85.76
	BERT-large-uncased	92.97	85.46
RoBERTa	RoBERTa-base	92.08	87.56
	RoBERTa-large	96.34	**92.58**
ALBERT	ALBERT-base-v1	87.28	81.85
	ALBERT-base-v2	91.83	86.16
	ALBERT-large-v1	90.54	84.85
	ALBERT-large-v2	93.61	88.87
	ALBERT-xlarge-v1	88.07	85.26
	ALBERT-xlarge-v2	93.37	89.37
	ALBERT-xxlarge-v1	96.44	92.48
	ALBERT-xxlarge-v2	**96.63**	92.28

Table 3: Comparison results across BERT-style models under BERT+FNN of Figure 1 for Subtask B. The default input representation of Table 1 is applied.

Given the effectiveness of the RoBERTa-large model, we use RoBERTa-large for evaluating BERT+BiLSTM with two input variants:

1. **BERT+BiLSTM**: The default input representation of Table 1 is applied.
2. **BERT+BiLSTM + extra words**: The extended input representation by adding extra words in Table 2 is applied.

Table 4 presents the results. Interestingly, the extended input representation of using Table 2 makes some improvement over that of using the default one. This result enables us to explore the issue of determining which natural expression is effective for the BERT input in the ComVE task. The run "BERT+BiLSTM + extra words" is our final submission for Subtask B. Table 4 summarizes the performances of the top three results in the leaderboard for reference. Finally, our system ranked 10th out of the 27 valid submissions for Subtask B.

Team Name	Accuaracy(%)		
	trial	dev	test
ECNU-ICA (Top 1)	-	-	**95.0**
hit-itnlp (Top 2)	-	-	94.8
NUT (Top 3)	-	-	94.3
BERT+BiLSTM (ours)	96.56	92.98	90.9
BERT+BiLSTM + extra words (ours)	97.01	**94.08**	91.4

Table 4: Comparison results using RoBERTa-large models under BERT+LSTM of Figure 2 for Subtask B.

4.3 Results on Subtask C

Table 5 presents the results of our model using UniLM for Subtask C, comparing the performances of the top three results in the leaderboard.

In Table 5, the model achieves a BLEU score of 70.20% on the trial set. From the further analysis, we found that the trial set contained a large number of instances in the training set. Given this redundancy, in the trial set where the input sentence is likely included in the training set, the model tends to accurately generate correct sentences that explain why the input sentence does not make sense. For new test sentences, it is observed that the model generates relatively low-quality sentences, compared to the results from the trail set.

Finally, our submission ranks 7th out of the 17 valid submissions on Subtask C.

Team Name	Bleu(%)		
	trial	dev	test
BUT-FIT (Top 1)	-	-	**22.4**
Solomon (Top 2)‘	-	-	19.3
LuoJunNB (Top 3)	-	-	18.5
UniLM	70.20	14.97	15.90

Table 5: Results of UniLM for Subtask C, comparing the top three submissions in the leaderboard.

5 Conclusion

We participated in Subtasks B and C at the SemEval-2020 ComVE. Our system was based on BERT and UniLM for Subtasks B and C, respectively.

For Subtask B, we explored the effects of the three variants of BERT-style models to study their commonsense reasoning ability. Finally, our submitted run for Subtask B ranked 10th out of the 27 submissions. Our results showed that the model capacity of BERT is highly related to the task accuracy, suggesting that BERT encodes the commonsense knowledge, but more in larger models. This implies that we should scale up the current models by exploring significantly larger models such as T5 (Raffel et al., 2019; Roberts et al., 2020), GPT-3 (Brown et al., 2020), or retrieval-based commonsense reasoning motivated by (Guu et al., 2020)

For Subtask C, the UniLM-based model ranked 7th out of the 17 submissions and 6th on the human score. We found that there still exists a large divergence between our results and human-level commonsense reasoning. Most of the sentences generated from the model were considerably different from the answers on the development dataset. Although we did not compare the other models on Subtask C, we expect that model capacity would be important here.

In future work, we plan to explore large pretrained models to enrich the commonsense knowledge in neural models under closed book (Roberts et al., 2020) or open book (Guu et al., 2020) settings. Further, we would like to incorporate pretrained language models with external knowledge, such as ConceptNet (Speer et al., 2017).

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 7059–7069.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19.

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*, NAACL '19.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32*, pages 13063–13075.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations.

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. *CoRR*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '18, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP '19, pages 43–54.

Alec Radford, Karthik Narasimhan, Tim Salimans, , and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model?

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. *CoRR*, abs/1905.02450.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a markov random field language model. *CoRR*, abs/1902.04094.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.