IIE-NLP-NUT at SemEval-2020 Task 4: Guiding PLM with Prompt Template Reconstruction Strategy for ComVE

Luxi Xing, Yuqiang Xie, Yue Hu*, Wei Peng
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{xingluxi,xieyuqiang,huyue,pengwei}@iie.ac.cn

Abstract

This paper introduces our systems for the first two subtasks of SemEval Task4: Commonsense Validation and Explanation. To clarify the intention for judgment and inject contrastive information for selection, we propose the input reconstruction strategy with prompt templates. Specifically, we formalize the subtasks into the multiple-choice question answering format and construct the input with the prompt templates, then, the final prediction of question answering is considered as the result of subtasks. Experimental results show that our approaches achieve significant performance compared with the baseline systems. Our approaches secure the third rank on both official test sets of the first two subtasks with an accuracy of 96.4 and an accuracy of 94.3 respectively.

1 Introduction

Natural Language Understanding (NLU) requires the systems can not only figure out the semantic of the text but also comprehend text with the constraint of commonsense knowledge about the world. The ability to identify the natural language statement against common sense and produce the explanation of its fault is the foundation towards realizing natural language understanding (Wang et al., 2019c). The SemEval-2020 Task 4 provides a well-formed evaluation mission that aims to evaluate the capacity of the system on commonsense validation and explanation (Wang et al., 2020).

The Commonsense Validation and Explanation (ComVE) task is divided into three subtasks including validation, explanation selection, and explanation generation. We mainly focus on and participate in the first two subtasks. The goal of the first validation subtask (subtaskA) is to inspect the ability of a system about distinguishing natural language statements that are against commonsense (i.e. false statements for short). The goal of the second explanation selection subtask (subtaskB) is to test if the system can correctly understand the reason for making against common sense. In subtaskA, the challenge with distinguishing the false statements lies in that this kind of statement usually conforms to the linguistic structure in syntactic level but its meaning does not fit the general commonsense in semantic level. In subtaskB, the difficulty of selecting an appropriate explanation for false statements is that, albeit the candidate explanations are relevant to the content of the false statements, they may not contain the main reason to account for the false statements and will distract the system.

To address the above challenges, we first formalize both subtasks as a type of multiple-choice Question Answering (QA) task. Recently, the the large Pre-trained Language Models (PLMs), such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), demonstrate its excellent ability in various natural language understanding tasks (Levesque et al., 2012; Zellers et al., 2018; Wang et al., 2019b; Wang et al., 2019a; Zellers et al., 2019). Moreover, according to recent research (Trinh and Le, 2019; Davison et al., 2019), they reveal that PLMs have already learned certain commonsense knowledge through pre-training with large scale corpus. Hence, we not only resort to the PLMs as the contextual encoder to generate the representation of sentences but also consider the PLMs as knowledge storage which can implicitly provide commonsense knowledge during question answering.

*Corresponding author.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 346–353
Barcelona, Spain (Online), December 12, 2020.

Aiming at coping with the challenges mentioned previously, we devise the approaches for solving the two subtasks in the multiple-choice QA fashion with the following two guiding intention: (a) How to arouse and utilize the implicit knowledge of PLMs to commonsense validation and explanation; (b) How will the extension of the context help the system to select the correct explanation for the false statement. For the first point, we explore a prompt template-based approach to reconstruct the input to the encoder. In subtaskA, we devise a prompt question and transfer this subtask into the multiple-choice QA style format where the statements are taken as candidate answers. In subtaskB, we reformat the false statement with a prompt template into a question to be answered with candidate explanation. The prompt is designed to activate the commonsense knowledge inner the pre-trained models, and it can be treated as a query to retrieve commonsense knowledge inside the PLMs. In addition, the prompt templates enrich the input of the PLMs to explicitly express the intention of the subtasks. For the second point, we propose to extend the prompt question in subtaskB with more context which can supply informative tips to locate the evidence of causing against common sense, i.e. the contrastive information between correct statements and false statements.

This paper describes approaches for subtaskA and subtaskB developed by the Natural Language Processing group of Institute of Information Engineering of the Chinese Academy of Sciences. Our contributions are summarized as the followings: (a) We employ the prompt template-based approaches on two subtasks to reconstruct original statements into the prompt to bring out the potential of the PLMs' commonsense knowledge; (b) We also explore the scoring-based approach to achieve the Validation subtask; (c) Experiments demonstrate that the proposed strategies achieve significant improvement compared with the PLMs baseline and we obtain the third-place in subtaskA and subtaskB on the final official evaluation.

In the following, we describe the approaches used for the two subtasks in Section 2.1 and Section 2.2 respectively. In Section 3, we elaborate our settings of experiments and report the performance on the public development set and final hidden test set. In Section 4, we analyze our approaches with cases.

2 Approaches

Before diving into the detail, we first present the description of symbols and the multi-choice based model which we use in both subtasks.

Formally, suppose there are five key elements in the two subtasks, i.e. $\{S^1, S^2, O^1, O^2, O^3\}$. We suppose the S^* denotes the ture and false statements, the O^* denotes the candidate explanation for subtaskB. S^1 and S^2 are the inputs to the validation subtask while the false statement and O^* are the inputs to the explanation subtask. And $y^A \in \{1, 2\}$ and $y^B \in \{1, 2, 3\}$ denote the labels of these two tasks respectively.

A multiple-choice based QA model $\mathcal{M}$ consists of a PLM encoder and a task-specific classification layer which includes a feed-forward neural network $f(\cdot)$ and a softmax operation. For each pair of question-answer, the calculation of $\mathcal{M}$ is as follow:

$$score_i = \frac{exp(f(C^i))}{\sum_{i'} exp(f(C^{i'}))}, C^i = \text{PLM}(inp) \tag{1}$$

where the inp is the input constructed according to the instruction of PLMs, and the C^* is the final hidden state of the first token (`[CLS]`). For more details, we refer to the original work of PLMs (Devlin et al., 2019; Liu et al., 2019). The candidate answer which owns a higher $score$ will be identified as the final prediction. The model $\mathcal{M}$ is trained end-to-end with the cross-entropy objective function.

2.1 Approach for Sense-Making Statement Validation

In the validation subtask, the system is required to select the statement which is against commonsense. We adjust this subtask into the multiple-choice style QA problem as $\hat{y}^A = argmax_{i \in \{1,2\}} P(S^i|Q^A)$, where Q^A is the additional prompt question, two statements are the candidate answers and y^A stands for the index of the commonsensible statement. We employ the RoBERTa-based multiple-choice model as our model $\mathcal{M}^{\mathcal{A}}$ to solve this subtask.

Intuitively, the function of the prompt question is two folds: (a) acting as the role of a potential question to be answered with the making-sense statement; (b) acting as the role of a query to retrieve commonsense knowledge inner PLMs. Hence, we directly construct a heuristic prompt question Q^A as: *If the following statement is in common sense?*. We suppose that this prompt question could contain the intention behind the validation task from the perspective of semantic.

With the proposed prompt question, the input, inp in Equation 1, to $\mathcal{M}^A$ is the concatenation of question and statement in the following format: `[CLS]` Q^A `[SEP]` S^i `[SEP]`. Then, we take the final representation of `[CLS]`, which represents the global semantic of question-answer pair, as the input to the task-specific classification layer. The statement which owns a higher score will be identified as the commonsensible statement. Furthermore, for limiting the length of the input and improving the computational efficiency, we propose another way to combine the prompt question Q^A and the statement S^i. When constructing input, we consider the phrase *the following statement* as the placeholder and replace it with S^i. Thus, the inp to $\mathcal{M}^A$ is as: `[CLS]` *If " S^i " is in common sense?* `[SEP]`. We will compare the performance of two ways in the Section 3.

2.2 Approach for Explanation Selection

In the explanation selection subtask, the system needs to select the most reasonable explanation from three candidates to account for the false statement. The false statement is represented symbolically by S^f, one comes from $\{S^1, S^2\}$. This subtask is formalized as the multiple-choice style in nature. However, we argue that the false statement is only a standard natural language sentence in surface and direct concatenation of the false statement and each candidate explanation will distract the model. Specifically, it is hard to avoid that model focuses more on the similarity between statement and explanation instead of the causal relationship. In which case, the model is unaware of the first sentence is a question and the situation it is dealing with is scoring for a possible candidate explanation to this question.

Based on the above consideration, we propose to reconstruct the false statement with a prompt template in order to make the model perceive the false statement as a question to be answered. Here, we design the prompt template to reformat a false statement as a question: S^f *is against common sense because* __ . The underline will be replaced by the candidate explanation to construct a complete question-answer pair. In this subtask, we also employ the RoBERTa-based multiple-choice model as our system $\mathcal{M}^B$. The formal input to $\mathcal{M}^B$ is in the following format: `[CLS]` S^f *is against common sense because* O^j `[SEP]`, where $j \in \{1, 2, 3\}$. Finally, we take the explanation which scores highest among the three template-based inputs as the selection result.

However, it is inadequate for selecting the most reasonable explanation merely with information of the false statement. It restricts and distracts the model's capability to discover the causal relationship between the false statement and candidate explanation. Based on the observation of data, we find that the true statement usually shares the same topic with the false one and the content of the true statement is in common sense. Consequently, we can resort to the true statement, denoted by S^t, to supply the contrastive information. We assume the true statement acting as the role of context in the multiple-choice QA framework. With additional context, we construct a merged input with the prompt template as the following: `[CLS]` *If* S^t *is in common sense.* `[SEP]` S^f *is against common sense because* O^j `[SEP]`, which will be the input to $\mathcal{M}^B$.

3 Experiments and Results

3.1 Experimental Setup

In subtaskA, the training/trial/development/test set contains $10,000/2,020/997/1,000$ pairs of statements. And the subtaskB shares the same size of the datasets with subtaskA where each example includes one false statement and three candidate explanations. Our system is implemented with PyTorch and we use the PyTorch version of the pre-trained language models*. We employ RoBERTa (Liu et al., 2019) large model as our PLM encoder in Equation 1. The Adam optimizer (Kingma and Ba, 2015) is used to fine-tune the model. We introduce the detailed setup about the best model on the development dataset. For subtaskA,

*https://github.com/huggingface/transformers (version 2.2.1)

A	inp
Orig.	`[CLS]` S^i `[SEP]`
P1	`[CLS]` *If the following statement is in common sense?* `[SEP]` S^i `[SEP]`
P2	`[CLS]` *If* S^i *is in common sense?* `[SEP]`
B	inp
Orig.	`[CLS]` S^f `[SEP]` O^j `[SEP]`
P	`[CLS]` S^f *is against common sense because* O^j `[SEP]`
P+C	`[CLS]` *If* S^t *is in common sense.* `[SEP]` S^f *is against common sense because* O^j `[SEP]`

Table 1: The input format for Validation subtask (A) and Explanation Selection subtask (B).

Model	Trial	Dev	Test
Baseline			
RoBERTa$_{\text{Large}}$	95.8	94.6	93.2
RoBERTa$_{\text{Large}}$+MNLI	95.9	94.4	93.4
RoBERTa$_{\text{OMCS}}$	97.1	96.2	95.6
Ours			
RoBERTa$_{\text{Large}}$+P1	96.1	95.5	95.8
RoBERTa$_{\text{Large}}$+P2	96.8	95.5	95.7
RoBERTa$_{\text{OMCS}}$+P1	97.3	**96.7**	**96.4**
RoBERTa$_{\text{OMCS}}$+P2	97.3	**96.9**	**96.4**

Table 2: Results (Accuracy) on Validation (A).

Model	Trial	Dev	Test
Baseline			
RoBERTa$_{\text{Large}}$	96.4	93.1	92.4
RoBERTa$_{\text{Large}}$+MNLI	96.2	92.6	92.0
RoBERTa$_{\text{OMCS}}$	96.4	92.0	91.9
Ours			
RoBERTa$_{\text{Large}}$+P	96.3	93.8	92.9
RoBERTa$_{\text{OMCS}}$+P	96.5	93.9	93.1
RoBERTa$_{\text{Large}}$+P+C	96.5	**94.8**	**93.8**
RoBERTa$_{\text{OMCS}}$+P+C	96.5	**94.5**	**94.3**

Table 3: Results (Accuracy) on Explanation (B).

we set the batch size to 24, initial learning rate to 1.5e$-$5 and the max length of input to 50. And the training of subtaskA is about 5 epochs. For subtaskB, we set the batch size to 36, initial learning rate to 1e$-$5 and the max length of input to 50 for only introducing prompt template and 86 for introducing additional context. And we train our model for 8 epochs.

For injecting more commonsense knowledge into the PLM, we introduce an intermediate pre-training based on the original PLM. Specifically, we conduct a second pre-training on the original RoBERTa model with the textual corpus from Open Mind Common Sense (Singh et al., 2002) through the Masked Language Modeling (MLM) task (Devlin et al., 2019). We use RoBERTa$_{\text{OMCS}}$ to stand for the intermediate pre-trained RoBERTa.

3.2 Evaluation Results

The validation subtask and explanation selection subtask use accuracy as the metric. For the purpose of clear comparison, we summary the reconstructed input format based on the prompt template into the Table 1. We select three PLM-based multiple-choice models with the original input format, which is shown as the rows start with "Orig." in Table 1, as the comparison baseline methods. In particular, the RoBERTa$_{\text{Large}}$+MNLI (Li et al., 2019) is also an intermediate pre-trained model that conducts second training with a supervised task, MNLI (Williams et al., 2018), and then is used to fine-tune on the target task. The baseline models are fine-tuned on the target dataset of subtasks with original input format.

On the subtaskA, i.e. the Statement Validation subtask, the evaluation results are illustrated in Table 2. Comparing with the original RoBERTa large model, the RoBERTa$_{\text{OMCS}}$, equipping with the additional commonsense textual corpus, obtains an improvement of 1.6 over RoBERTa$_{\text{Large}}$ on the development dataset, which provides evidence that the additional textual corpus facilitates the PLM with commonsense knowledge to a certain degree. Comparing with the baselines, the models with reconstructed input based on prompt template obatin strong improvement over baselines on both the development set and test set. We observe that two types of prompt template, denoted by P1 and P2, show up almost the same performance

on test set based on RoBERTa$_{\text{OMCS}}$. We suppose the reason for that two prompt templates cause the same effect is the same semantic of the intention behind the different forms of the surface. And both of prompt templates offer the same hints about the task to the PLM. In the final official evaluation, we submit the prediction from RoBERTa$_{\text{OMCS}}$+P2 to the leaderboard as our final result.

On the subtaskB, the Explanation Selection subtask, the experiment results are shown in Table 3. In the group of baseline model, the RoBERTa$_{\text{OMCS}}$ surprisingly gets a lower score compared with RoBERTa$_{\text{Large}}$ on the development set. When reconstructing the original input with prompt template, it brings $0.8 \sim 1.9$ gain over baseline models, i.e. RoBERTa$_{\text{Large}}$ and RoBERTa$_{\text{OMCS}}$, on development set. The reversal of RoBERTa$_{\text{OMCS}}$'s performance on both the development and test set proves that the prompt template achieves the role of activating the potential knowledge inner PLM. Moreover, with the additional information from true statement, the models with +P+C further get improved on the development set. Though the performance of RoBERTa$_{\text{OMCS}}$ exhibits no advanced over RoBERTa$_{\text{Large}}$ on the development set, its performance shows $0.2 \sim 0.5$ gains over RoBERTa$_{\text{Large}}$ on test set. In the final official evaluation of Explanation Selection subtask, we commit the result of RoBERTa$_{\text{OMCS}}$+P+C to the leaderboard as our final result.

4 Discussion

4.1 Probing Commonsense Knowledge within PLM

As mentioned previously, the PLMs have learned commonsense knowledge through pre-training. We further explore how well the commonsense knowledge inside the PLMs can benefit for the commonsense validation task and we also investigate the performance of the PLM with an intermediate pre-training on OMCS corpus. Inspired by LM scoring-based methods in previous work (Trinh and Le, 2019; Wang et al., 2019c), we calculate the score for each statement following the instruction of Wang et al. (2019c) in a zero-shot fashion. The statement which gets a higher score will be regarded as against commonsense. In the development set, the RoBERTa$_{\text{Large}}$ gets an accuracy of 79.5 while the RoBERTa$_{\text{OMCS}}$ obtains an accuracy of 86.3.

The examples of the LM scoring output are illustrated in Figure 1. The higher score of a token represents the token is not common under the current context. It is clear that the PLMs could capture the keywords among the sentence which cause the statement is uncommon. However, the PLM after intermediate pre-training tends to focus on the beginning of the sentence, as shown in the second example in Figure 1. Moreover, there lacks a normalization to compare the scores between different statements which leads LM scoring is not stable. Towards probing existence of commonsense knowledge, the improvement of RoBERTa$_{\text{OMCS}}$ indeed demonstrates that the intermediate pre-training with the specific corpus injects more commonsense knowledge into the PLMs.

Example	L+Orig.	O+Orig.	L+P1	O+P1	L+P2	O+P2
False: A tuna is a mammal True: A dolphin is a mammal	✗	✗	✓	✓	✓	✓
False: the bleach cleaned the house True: he cleans up with bleach	✗	✓	✗	✗	✓	✓
False: TV's are found in the ocean. True: Tim bought a new TV yesterday.	✗	✗	✗	✗	✗	✗

Table 4: Prompt templates effect on subtaskA. (L: RoBERTa$_{\text{LARGE}}$, O: RoBERTa$_{\text{OMCS}}$.)

4.2 The Effect of Prompt Templates on subtask A

We perform case study on the effect of prompt templates (P1 and P2) on subtask A. Orig. represents the original input format for RoBERTa. As shown in Table 4, we sample three standard examples from Dev set of subtask A. From the first example, pre-training on OMCS can not work but each prompt template

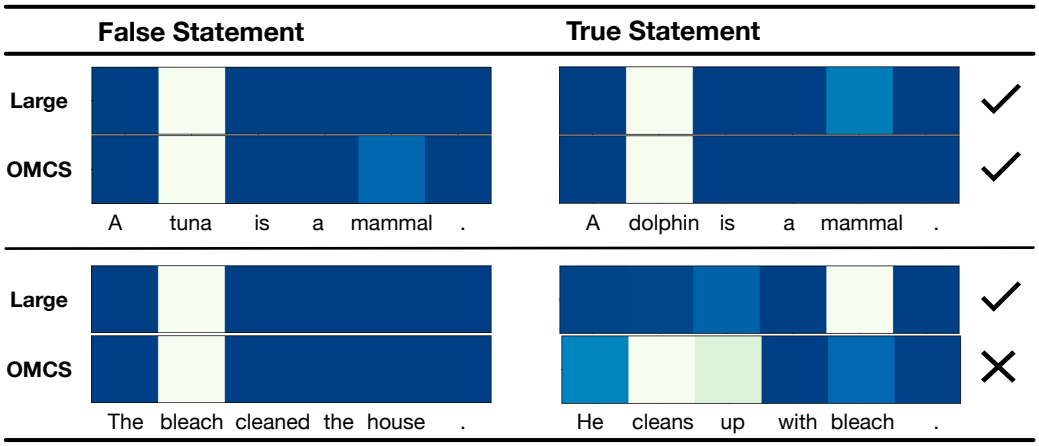

Figure 1: The visualization of token-level scores of the LM (RoBERTa) scoring results. The check mark stands for the right judgment while the cross mark stands for the wrong judgment. The lighter color represents a higher proportion of the whole score of the current sentence.

help PLM reason out the false statement. It is possible that prompt templates offer beneficial hints about task A to the PLM. However, there are still some errors in our method. From the second example, template P1 misleads RoBERTa$_{\text{OMCS}}$ make the wrong decision, while template P2 supplies the PLM with a more suitable hint. As shown in the last example, all of the models make the wrong decision. It could be the PLMs' own problem: inner bias of pre-training data, word frequency of "TV", and so on. All in all, we conclude that prompt templates could help PLM understand the objective of the task. In addition, there are still unsolved problems to address.

Example	L+Orig.	O+Orig.	L+P	O+P	L+P+C	O+P+C
True: I eat all the cake. *False*: I eat all the supermarket. *A*: The supermarket is good to find food. *B*: Supermarket sells too much food. **C: In the supermarket, there is too much food to eat.**						
Prediction	✗ (B)	✗ (B)	✓	✓	✓	✓
True: He cooked the egg with a pan. *False*: He cooked a pan with the egg. *A*: Pans are usually red while the egg is yellow. *C*: An egg cannot cook a pan. **B: Pan is used to cook food like eggs.**						
Prediction	✗ (C)	✗ (C)	✗ (C)	✗ (C)	✓	✓
True: The largest animal on the land is the elephant. *False*: The largest animal is the elephant. *A*: Elephants need to live on the land. *C*: Elephants are larger than many animals. **B: Some animals living in the water is larger than the elephant.**						
Prediction	✗ (C)	✗ (C)	✗ (C)	✗ (C)	✓	✓

Table 5: Prompt templates effect on subtaskB. The correct explanation are in bold. (L: RoBERTa$_{\text{LARGE}}$, O: RoBERTa$_{\text{OMCS}}$.)

4.3 The Effect of Prompt Templates on subtask B

We sample a collection of examples to investigate the effect of the prompt templates on subtask B, and the detail of examples are shown in Table 5. We compare the difference between naive input (+Orig.), prompt template-based input (+P) and input with expanded context (+C). As illustrated in the first example, the systems, which only take original input, fail to make the correct prediction. As the input lacks the intention of the second sentence, the systems easily choose the wrong explanation that shares more text-based

information with the false statement. And with the template engaged in, the systems take the semantic of whole prompt template-based input into consideration and will select the option which can make the template-based input validity As seen in the last two examples, the systems, without prompt and additional context simultaneously, still tend to select the incorrect explanation which owns similar words with the false statements. It is obvious that the absence of the specific and contrastive contextual information about the false statements will weaken the selection ability of the systems. Based on the analysis of examples, we can conclude that the prompt template-based input is beneficial to the final selection and the additional context can also facilitate the ability of judgment of systems towards a specific question intention.

5 Conclusion

In this paper, we present our approaches for participating in the SemEval Task on Commonsense Validation and Explanation, which utilize the template to reconstruct the input with prompt information and inject additional context to provide contrastive information to improve the judgment and selection ability of systems. Experimental results manifest that both strategies benefit to the final performance. Moreover, the auxiliary probing experiments confirm that PLMs contain rich commonsense knowledge which can be mined to facilitate downstream tasks.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback. This work has been supported by the National Key Research and Development Program of China under Grant No.2016YFB0801003.

References

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178, Hong Kong, China, November. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*.

Zhongyang Li, Xiao Ding, and Ting Liu. 2019. Story ending prediction by transferable BERT. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1800–1806.

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*.

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

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002 Irvine, California, USA, October 30 - November 1, 2002, Proceedings*, pages 1223–1237.

Trieu H. Trinh and Quoc V. Le. 2019. Do language models have common sense?

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019c. 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, Florence, Italy, July. 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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, October-November. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July. Association for Computational Linguistics.