Attending Knowledge Facts with BERT-like Models in Question-Answering : Disappointing Results and Some Explanations

Guillaume Le Berre and Philippe Langlais

Université de Montréal, Montréal QC, Canada
guillaume.le.berre@umontreal.ca
felipe@iro.umontreal.ca

Abstract. Since the first appearance of BERT, pretrained BERT inspired models (XLNet, Roberta, ...) have delivered state-of-the-art results in a large number of Natural Language Processing tasks. This includes question-answering where previous models performed relatively poorly particularly on datasets with a limited amount of data. In this paper we perform experiments with BERT on two such datasets that are OpenBookQA and ARC. Our aim is to understand why, in our experiments, using BERT sentence representations inside an attention mechanism on a set of facts tends to give poor results. We demonstrate that in some cases, the sentence representations proposed by BERT are limited in terms of semantic and that BERT often answers the questions in a meaningless way.

1 Introduction

Question answering has long been a core task in Natural Language Processing (NLP). Due to the booming of deep learning, there has been recently a resurgence of work on question-answering, leading to multiplications of benchmarks. Despite the amazing progress made by deep learning methods, current models fail to achieve human performance on a lot of these benchmarks. Similarly to other NLP tasks, question answering features a wide range of sub-tasks. Some extractive question answering datasets provide an open question on a short text and require the models to select a chunk in the text (often corresponding to an entity) that answers the question. SQuAD 1 and 2 [1] [2] are two popular exemples of this type of benchmark. On the other hand, datasets like CoQA [3] rather ask models to generate an answer that is typically not a span of the input text or like in RACE [4] provide multiple answer choices from which to choose while still using a provided text as reference. In our work, we will focus on two datasets, OpenBookQA [5] and ARC [6], representative of another type of question answering task. OpenBookQA and ARC feature questions adjoined with 4 possible answers (similar to RACE), but unlike the datasets presented above do not provide a reference text with each question. Instead, both datasets are adjoined with a set of common sense sentences supposedly containing all the

knowledge needed to answer the questions but not directly linked to any question in particular. The competing models thus can retrieve information from this set of sentences in order to answer the proposed question. However, most state of the art models rather choose to ignore this additional information and instead rely on learned world knowledge. We believe that learning to use this knowledge can lead to improved performances and higher generalization capability.

Recently, [7] introduced BERT, a pretrained deep learning model which showed huge improvements in a large number of NLP tasks including question answering. Models inspired from BERT are currently widely used on a lot of question answering datasets and often hold the first places in the leaderboards. Notably, on SQuAD, most recent models even beat the human level performance.

These pretrained models can take advantage of a massive quantity of unlabelled data and are thus particularly useful for tasks that require common sense knowledge since they already embed some semantic knowledge from the pretraining. Furthermore, pretraining allows for shorter training time on specific data and is particularly advantageous for relatively small datasets like ARC and OpenBookQA that each only gathers a few thousands of questions. Nearly all current state-of-the-art models on ARC and OpenBookQA are pretrained models, but they still do not reach human level performances.

In this work, we report the results of multiple experiments we conducted with BERT-like models on OpenBookQA and ARC. More precisely, our objective is to evaluate the possibility of using external common sense knowledge to enhance the current models. Although OpenBookQA and ARC do not provide a reference text for each question, both datasets are adjoined with a list of common sense items written in natural language, that is, short sentences such as "A bee is a pollinating animal". To the best of our knowledge, among all the models proposed to address this tasks, most of them only use the common sense knowledge acquired during training (including pretraining) and only a few models really used this dataset of common knowledge. A model able to use this extra data may allow at test time to add sentences into the common sense database and thus adapt to some extent to new domains without retraining. In addition, using this common sense database is a first step toward building a model able to reason using the short sentences (facts) in the database and combining them to assess the rightness of an answer.

2 Related Work

In this section, we first sketch how BERT works. Then we introduce other models and techniques we implemented in this work.

BERT [7] is a deep language model pretrained on the BooksCorpus [8] and English Wikipedia. The model itself is composed of a multi-layer transformer [9]. Once pretrained, it provides a context dependant embedding of all the words in a sentence and thus can provide a sentence embedding as well by either taking the first token's embedding ('CLS' special symbol) or the mean of the words embeddings. BERT is pretrained on two unsupervised tasks: "Masked LM" and

"Next Sentence Prediction". In Mask LM, the model is fed with a sentence where some words are randomly masked and has to predict the missing words. While in Next Sentence Prediction, the model is fed with two sentences and has to determine if the last is the actual sentence following the first. Two versions of BERT have been released: the "base" version with 12 layers and representation vectors of dimension 768 and the "large" version with 24 layers and vectors of dimension 1024.

Following the release of the initial paper, multiple adaptations of BERT have been proposed (XLNet [10], Roberta [11]), each one improving the model or the pretraining procedure. Sentence-BERT (SBERT) [12] is one of these models that aim to increase the semantic meaningfulness of the sentence representation provided by BERT. The authors propose to add to the standard BERT pretraining an additional pretraining step on the SNLI dataset [13]. SNLI is a dataset containing sentence pairs labelled as entailment, contradiction or neutral. They fine-tune BERT using a Siamese neural architecture so that two sentences marked as entailment have a representation close to one another (cosine distance) while two sentences marked as contradiction have representations that are far apart. Their claim is that the resulting sentence representations are more semantically relevant than the representations obtain by a vanilla BERT model.

In this work, we also experiment with MAC Cells [14]. This architecture was first introduced on the CLEVR dataset [15] (question-answering using an image) in order to improve the reasoning capability of attention-based neural networks. This model is build as a recurrent network maintaining two state vectors (memory and control). Control vector is used to determined which reasoning action must be performed at each step while memory vector is a representation of all the information the model has obtained. The MAC cell is composed of 3 modules (see figure 1): at each step the "control" module updates the control vector using the previous control vector. A "read" module then uses this new control vector and the previous memory vector to perform an attention on a database (attention over the image in the case of CLEVR) and thus creates a proposed new memory vector. The final new memory is a linear combination of the previous and proposed memory decided by the "write" module as a function of the control state. MAC Cells obtained good results on CLEVR and showed they were capable of basic reasoning.

3 Experimental Protocol

For this study, we use 2 datasets: OpenBookQA [5] and ARC [6]. Both are multiple-choice question datasets with 4 choices for each question. The questions are about a broad variety of subjects related to everyday life logic or general knowledge.

The OpenBookQA dataset is composed of 3 parts: train, validation and test with respectively 4958, 500 and 500 questions in each. It is adjoined with two small datasets of common sense sentences. They are similar in their content but one of them is composed of all the sentences provided to the annotators as

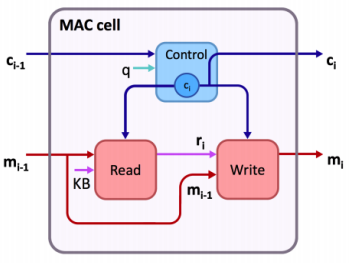

Fig. 1: Inner working of a MAC cell. The cell is a recurrent unit maintaining 2 state vectors c_i and m_i. c_i is the control state and determine which action is to be done at a given time step and m_i is the memory state that stores the information from previous steps. Image from [14]

an inspiration during the creation of the dataset (each question is linked to 1 sentence but each sentence may have been used for multiple questions) and thus these sentences most of the time contain the necessary information to answer the questions. The correspondence between the questions and these sentences is known. The second one is composed of the same kind of sentences but these ones are not directly related to any question in particular. The first and second datasets are composed of 1327 and 5168 sentences respectively. See figure 2 for examples of a question and fact sentences.

The ARC dataset is split into 2 parts. The "Easy" part corresponds to questions well answered by classical machine learning models while more complicated questions belong to the "Challenge" part. The train sets for "Easy" and "Challenge" contain 2252 and 1120 questions respectively. Similarly to the fact dataset in OpenBookQA, we have access in ARC to a 1.4GB dataset of common sense sentences data-mined from the web but no information about which ones can help a given question.

The metric used for evaluation is accuracy defined as the percentage of questions correctly answered. Current state-of-the-art models on OpenBookQA reach 78% accuracy using Roberta [11] and an additional pretraining on RACE dataset [4] while BERT Large with no additionnal pretraining achieves 60% accuracy. On ARC challenge, the best model scores 68% accuracy. The scores of BERT Base and Large on ARC challenge are around 36% and 40% respectively.

4 Models

In this section we present different model architectures we implemented. In all of these models, we use BERT or SBERT as a sentence embedding technique. To obtain the embedding of a sentence we use a mean pooling over the word representations provided by (S)BERT. We observed no significant differences in performance between the mean pooling and other methods of pooling (e.g. take the start token representation vector as sentence embedding) as long as

OpenBookQA	
Question	Stars are ... - A: warm lights that float B: made out of nitrate **C: great balls of gas burning billions of miles away** D: lights in the sky
Related fact	A star is made of gases
Other interesting facts	▷The Earth rotating on its axis causes the sun to appear to move across the sky at night ▷The Earth rotating on its axis causes stars to appear to move across the sky at night ▷The north star does not move in the sky in the Northern ▷Hemisphere each night ▷Burning wood is used to produce heat
ARC	
Question	Which is a nonrenewable resource? - **A: oil** B: trees C: solar energy D: food crops

Fig. 2: An example of 2 questions in ARC and OpenBookQA with the fact given to the annotator and 4 additional facts automatically selected by word co-occurrence with the question and all possible choices.

the weights of BERT are fine-tuned on the end task. Although we only refer to BERT in the following model description, BERT and SBERT embeddings are commutable, and we tested both.

4.1 Model A

This model is the vanilla question-answering setting for BERT used by the large majority of the proposed models on ARC and OpenBookQA. The answer choices are concatenated to the question thus obtaining 4 "question + choice" sequences. From there, we use BERT to get a sentence embedding vector and we send this embedding vector through a 2 layer perceptron with a ReLU activation function in between to obtain a single scalar score for each sequence. The scores corresponding to the 4 choices are then gathered and a softmax is applied on them. During training, we use a cross-entropy loss and at test time, the choice with the highest score is selected as the predicted answer. See figure 3A.

4.2 Model B

In addition to the "question + choice" sequence embedding we provide the model with an additional sentence also embedded with (S)BERT. Note that this model and the following ones are trained on OpenBookQA only since they require the

link between the common sense database and questions. We use the common knowledge sentence associated to the question in OpenBookQA. This has been provided to annotators as an inspiration to write the question and thus, this sentence is supposed to give enough information to the model to answer the question. The idea behind this model is to evaluate the semantic quality of the sentence embedding. Since the additional sentence is supposed to have a meaning closer to the answer than the other choices, their embedding should also be closer. The new sentence embedding is concatenated to the "question + choice" embedding and fed to the final perceptron. See figure 3B.

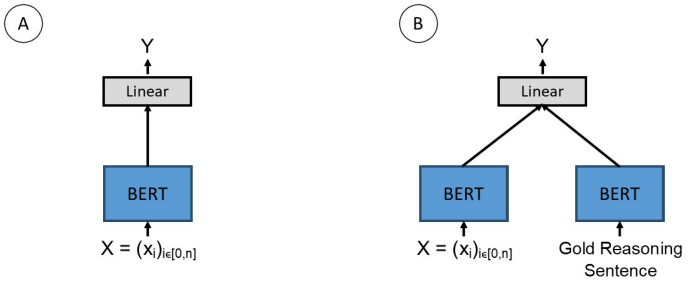

Fig. 3: Description of model A (left) and B (right). X represents the input sequence consisting of the question plus one answer choice. Y represents the output scalar score for this particular answer choice. The scores for all answer choices are then gathered and passed through a softmax function (omitted here).

4.3 Model C

Model B is not applicable to any real case scenario since it presupposes having access to the sentence that inspired the question. In this third architecture, we add an attention mechanism to B. Instead of a unique sentence, the model now has to select the right sentence from a set of 10 sentences extracted from the common knowledge dataset of OpenBookQA. We select the 9 sentences with the highest number of words in common with the question and add the "target" sentence if not already selected. See figure 4C.

4.4 Model D

Finally, we experimented with MAC Cells [14]. Usually, multiple facts are relevant for a given question. The reasoning capability of MAC Cells can thus be useful in order to assemble multiple pieces of information from different facts.

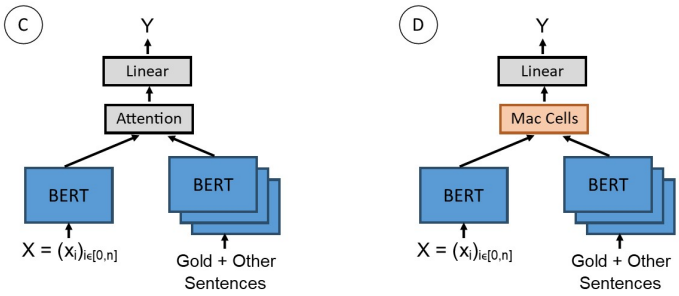

Fig. 4: Description of model C (left) and D (right). The only difference with model B is that instead of a single help sentence, the model has access to multiple sentences including the help and needs to select the right one with an attention (model C) or MAC cells (model D).

We replace the simple attention of model C by a MAC Cell in the hope, this would help the model to extract more information from multiple sentences. See figure 4D.

5 Implementation Details

We use the Hugging Face Pytorch implementation of BERT [16] and the implementation of SBERT provided with the original paper. We use the base version of both BERT and SBERT. This means that the word embeddings are of dimension 768 and we kept the same vector size along all the steps of the models. We implemented the rest of the code using Pytorch.

The training of all the models is made with a mini-batch of size 8. The shorter sentences in the mini-batch are padded using a special token and we cropped the sequences longer than 40 tokens by removing the first tokens thus ensuring that we keep the answer choice untouched (Table 3 provides some length statistics for OpenBookQA). We train the models for a maximum of 5 epochs after which the models always starts to overfit since the datasets are small and BERT has a huge capacity. To compensate for overfitting, BERT weights are frozen during the first 2 epochs to let a chance to the other parts of the network to converge. On models that require an attention mechanism, we added an additional cross-entropy loss directly on the attention distribution to help the models to quickly identify what is the sentence that gives the required information. A weighting coefficient is applied to this part of the loss. It starts to 1.0 and decreases to 0.2 after the first 2 epochs. All the hyper-parameters above are chosen to maximize accuracy on validation set.

When training on the ARC dataset, we use both "Easy" and "Challenge" training sets and report separated results for test and validation sets.

6 Experiments

In this section, we present the experiments we made with BERT on OpenBookQA and ARC. All the experiments shown below were made in an attempt to explain why in our preliminary experiments we failed to apply an attention mechanism (model C) over a knowledge base of sentences embedded with BERT.

First, in order to appreciate what parts of a question BERT is using while answering it, we decrease the number of tokens available to the model. To do this, we use the A setup described in section 4. We provide the results (in terms of accuracy) of a model that has access to the complete question and a model with access to only the last 4 tokens of the question. Eventually, we completely removed the question thus feeding only the answer choice to BERT. We of course expect the model to reach around 25% accuracy (OpenBookQA has 4 answer choices for each question) with this setup since without the question, there is supposedly no way to differentiate the right answer from the other choices. We verified that the dataset (train, validation and test) is balanced in the sense that all answers choices (A, B, C and D) appear approximately 25% each.

	full	4 tokens	none
BERT (model A)	**55.8**	52.0	51.2
SBERT (model A)	53.2	**53.0**	**53.6**
BERT (model A + help)	**64.5**	64.9	-
SBERT (model A + help)	63.6	**65.2**	-
BERT (model B)	53.0	53.4	-
SBERT (model B)	**55.0**	**61.3**	-
SBERT (model C)	54.8	56.8	-
SBERT (model D)	**56.8**	**56.8**	-

Table 1: Accuracy on OpenBookQA when the model has access to the full question (full) or only the 4 last tokens of the question (4 tokens) or no question at all (none). model A + help refers to model A in which the input is the concatenation of the help sentence, question and answer choice instead of just question + answer choice.

	full	4 tokens	none
BERT (Easy)	51.7	46.9	36.5
BERT (Challenge)	36.5	36.2	34.2

Table 2: Accuracy on ARC (Easy and Challenge) when model A has access to the full question (full), only the 4 last tokens of the question (4 tokens) or no question at all (none).

Table 1 shows the resulting accuracies on the validation and test sets on OpenBookQA (first 2 lines) for both BERT and SBERT (for comparison purposes since we use SBERT in later experiments). What comes out of this first experiment is that, surprisingly, giving only the last 4 tokens of the question to BERT only slightly decreases the accuracy. Furthermore, if we completely remove the question, BERT is still managing to get an accuracy greater than 50%, not so far from the original accuracy and even, in the case of SBERT, a better accuracy.

We ran the same experiment on both parts (Easy and Challenge) of ARC. The results are reported in table 2. The challenge set results are quite similar to what we observed for OpenBookQA: the accuracy for 4 tokens and no token are close to the accuracy obtained by the vanilla BERT model. On the Easy set however, we see a clear degradation of the accuracy when removing tokens but we still obtain at worse 36% accuracy which is significantly better than the expected 25%. This shows that in any case, a significant part of the questions can be answered without even looking at the question itself.

Length in words (train set/test set)	mean	min	max
Questions	10.7/10.3	1/1	68/61
Wrong answers	2.7/3.0	1/1	20/16
Correct answers	3.0/3.3	1/1	21/15

Table 3: Mean/min/max length of the questions and answers in OpenBookQA. Correct answers tend to be slightly longer than wrong ones.

Word frequency (train set/test set)	mean	min	max
Wrong answers	0.0054/0.0056	0.00029/0.00033	0.016/0.017
Correct answers	0.0050/0.0058	0.00014/0.00019	0.016/0.019

Table 4: Mean/min/max frequencies of tokens in an answer choice averaged over the dataset in OpenBookQA. Correct answers tend to contain at least a token that is more specific (less frequent) than the other answers.

Our models are thus mainly learning what the characteristics of a good answer are instead of learning the logical link between a question and the corresponding answer. We found some pieces of explanation for this. First, we compute statistics about the length of the questions and answers. These statistics are reported in table 3 and show that the right answers are on average longer than their counterparts. In practice, a dummy model that selects the longest answers among the 4 proposed achieves a 33% accuracy. However, this does not explain the entirety of the 51.2% of BERT on OpenBookQA.

Question	What impacts an objects ability to reflect light?
Answer choices	**A: color pallete** B: weights C: height D: smell
4 tokens	...ability to reflect light?

Fig. 5: An example of question in OpenBookQA (the spelling error is from the dataset) for which it is relatively easy to answer using only the last 4 tokens. Note that the correct answer is more complex than the other three and so it might be possible to guess the answer without reading the question.

In table 4, we further provide statistics about the relative frequencies of the tokens in correct vs. incorrect answers. What is interesting here is that the least frequent token in the correct answers is on average less frequent than the least frequent token in incorrect answers. This means that the correct answers tends to include more specific words than incorrect ones. This could be a bias linked to the annotation method in which annotators are required to invent incorrect answers. It is possible that one tends to be more generic when trying to write a wrong answer without inspiration. A dummy model that selects the answer with the least frequent token achieves 36.8% accuracy on OpenBookQA. Finally, our experiments shows that 54.6% of the correct answers are either the longest sequences or the one with the least frequent token in it. These two experiments although they do not explain the entirety of the phenomenon, show that the models (BERT and SBERT) can be heavily biased by factors unrelated to the question-answering logic.

As explained in section 3, OpenBookQA provides along each question a common knowledge short sentence that was provided to the annotator as an inspiration for the question. Although using this sentence directly causes the results to be incomparable with results showed above and state-of-the-art, we can use it to evaluate how much accuracy could be gained if we could make a decent selection of related knowledge in a database. We refer at this particular sentence as the help sentence in this section.

In the following, we still use BERT/SBERT in the A setting on OpenBookQA but instead of the simple question+answer sequence, we concatenate at the beginning of the sequence the common sense sentence related to the question. The idea for now on is to see if the representations from BERT/SBERT are usable in a configuration where the solution is directly given to the model. Table 1 shows that as expected, we obtain a significant increase in performance with both BERT and SBERT. Now, what happens if the new sentence is not given prior to BERT embedding but rather posterior to it? To test this, we use the B setting. In this configuration, the model now has to rely on a good semantically significant sentence embedding to answer since it has to evaluate the similarity between the help sentence and each of the answer choices. The accuracy is again described in table 1 (lines 5 & 6). We observe that BERT performs relatively

poorly at this task and shows no significant change of its accuracy between this configuration and no help sentence at all. SBERT however obtains a gain similarly to what happens when concatenating the help sentence to the question. Thus, the representations of SBERT seem to be more adapted to find semantic similarity links between sentences.

To confirm this observation we ran yet another dummy model. This model compute the similarity (cosine distance) between the BERT/SBERT representation of the help sentence and the representations of the question+answers. The process is done without any training using only the pretrained BERT and SBERT. Using this configuration, SBERT achieves 57.6% accuracy on OpenBookQA test set while BERT only reaches 37.4% once again demonstrating that SBERT is more adapted to the task.

Finally, we tested more realistic settings. Instead of giving the help sentence directly, we "hide" it among other similar sentences thus simulating a scenario in which we use a selector able to accurately select a pool of fact sentences useful for answering the question. As explained in section 4, we select 9 sentences from the fact dataset of OpenBookQA according to the maximum word overlap with the question and we add the help sentence to ensure a good selection. We report the results of SBERT for configuration C and D in table 1. BERT performs poorly for both configurations during our experiments (likely due to the reasons previously exposed) and so only the accuracy of SBERT are shown here. Overall, the models C and D are weaker than the configuration B with direct help but still perform better than model A where no help is given. This tends to show that there is value to be earned by adding additional common sense information to the inputs of the model and in the future, we intend to continue working on new ways to achieve this objective. We would like for example to try to re-balance correct and incorrect answers by picking correct answers from another question as new incorrect answers. Like this, we could potentially improve the training of BERT by removing the local minimum created by the length and frequencies bias.

7 Conclusion

In this work, we compared a number of models based on BERT-like models for question-answering. We report disappointing but informative results. Our experiments show that in the case of OpenBookQA more than 50% of the questions can be answered without even looking at them, which represents a big bias that has to be taken into account when considering state-of-the-art results. This observation also transposes to some extent to other question-answering benchmarks such as the ARC dataset. In addition we present a comparison of BERT and SBERT representations in term of semantic usefulness and show that SBERT is more apt to combine sentence representations together in order to answer the question.

References

1. Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018.
2. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.
3. Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042, 2018.
4. Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683, 2017.
5. Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. *CoRR*, abs/1809.02789, 2018.
6. Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
8. Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv e-prints*, page arXiv:1506.06724, Jun 2015.
9. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, Jun 2017.
10. Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.
11. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
12. Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv e-prints*, page arXiv:1908.10084, Aug 2019.
13. Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
14. Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. *CoRR*, abs/1803.03067, 2018.
15. Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016.
16. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing, 2019.