FINDING A JACK-OF-ALL-TRADES: AN EXAMINATION OF SEMI-SUPERVISED LEARNING IN READING COMPREHENSION

Rudolf Kadlec*, **Ondrej Bajgar***, **Peter Hrincar & Jan Kleindienst**
IBM Watson
V Parku 4, 140 00 Prague, Czech Republic
{rudolf_kadlec,obajgar,phrincar,jankle}@cz.ibm.com

ABSTRACT

Deep learning has proven useful on many NLP tasks including reading comprehension. However, it requires large amounts of training data which are not available in some domains of application. Hence we examine the possibility of using data-rich domains to pre-train models and then apply them in domains where training data are harder to get. Specifically, we train a neural-network-based model on two context-question-answer datasets, the BookTest and CNN/Daily Mail, and we monitor transfer to subsets of bAbI, a set of artificial tasks designed to test specific reasoning abilities, and of SQuAD, a question-answering dataset which is much closer to real-world applications. Our experiments show very limited transfer if the model is not shown any training examples from the target domain however the results are encouraging if the model is shown at least a few target-domain examples. Furthermore we show that the effect of pre-training is not limited to word embeddings.

1 INTRODUCTION

Machine intelligence has had some notable successes, however often in narrow domains which are sometimes of little practical use to humans – for instance games like chess (Campbell et al., 2002) or Go (Silver et al., 2016). If we aimed to build a general AI that would be able to efficiently assist humans in a wide range of settings, we would want it to have a much larger set of skills – among them would be an ability to understand human language, to perform common-sense reasoning and to be able to generalize its abilities to new situations like humans do.

If we want to achieve this goal through Machine Learning, we need data to learn from. A lot of data if the task at hand is complex – which is the case for many useful tasks. One way to achieve wide applicability would be to provide training data for each specific task we would like the machine to perform. However it is unrealistic to obtain a sufficient amount of training data for some domains – it may for instance require expensive human annotation or all domains of application may be difficult to predict in advance – while the amount of training data in other domains is practically unlimited, (e.g. in language modelling or Cloze-style question answering).

The way to bridge this gap – and to achieve the aforementioned adaptability – is *transfer learning* (Pan & Yang, 2010) and closely related *semi-supervised learning* (Zhu & Goldberg, 2009) which allow the system to acquire a set of skills on domains where data are abundant and then use these skills to succeed on previously unseen domains. Despite how important generalization is for general AI, a lot of research keeps focusing on solving narrow tasks.

In this paper we would like to examine transfer of learnt skills and knowledge within the domain of text comprehension, a field that has lately attracted a lot of attention within the NLP community (Hermann et al., 2015; Hill et al., 2015; Kobayashi et al., 2016; Kadlec et al., 2016b; Chen et al., 2016; Sordoni et al., 2016; Dhingra et al., 2016; Trischler et al., 2016; Weissenborn, 2016; Cui et al., 2016b;a;

[*] These authors contributed equally to this work.

Li et al., 2016; Shen et al., 2016). Specifically, we would like to address the following research questions:

1. Whether we could train models on natural-language tasks where data are abundant and transfer the learnt skills to tasks where in-domain training data may be difficult to obtain. We will first look into what reasoning abilities a model learns from two large-scale reading-comprehension datasets using artificial tasks, and then check whether it can transfer its skills to real world tasks. Spoiler: both these transfers are very poor if we allow no training at all on the target task.
2. Whether pre-training on large-scale datasets does help if we allow the model to train on a small sample of examples from the target tasks. Here the results are much more positive.
3. Finally we examine whether the benefits of pre-training are concentrated in any particular part of the model - namely the word-embedding part or the context encoder (the reasoning part). It turns out that pre-training is useful for both components.

Although our results do not improve current state of the art in any of the studied tasks, they show a clear positive effect of large-dataset pre-training on the performance of our baseline machine-learning model. Previous studies of transfer learning and semi-supervised learning in NLP focused on text classification (Dai & Le, 2015; Mou et al., 2016) and various parsing tasks (Collobert et al., 2011; Hashimoto et al., 2016). To our knowledge this work is the first study of transfer learning in reading comprehension, and we hope it will stimulate further work in this important area.

We will first briefly introduce the datasets we will be using on the pre-training and target sides, then our baseline model and afterwards in turn describe the method and results of each of the three experiments.

2 Datasets

2.1 Pre-training datasets

We have mentioned that for the model pre-training we would want to use a task where training data are abundant. An example of such task is context-dependent cloze-style-question answering since the training data for this task can be generated automatically from a suitable corpus. We will use two such pre-training datasets in our experiments: the BookTest (Bajgar et al., 2016) and the CNN/Daily Mail (CNN/DM) news dataset (Hermann et al., 2015).

The task associated with both datasets is to answer a cloze-style question (i.e. fill in a blank in a sentence) the answer to which needs to be inferred from a context document provided with the question.

2.1.1 BookTest

In the BookTest dataset, the context document is formed from 20 consecutive sentences from a book. The question is then formed by omitting a common noun or a named entity from the subsequent 21st sentence. Among datasets of this kind, the BookTest is among the largest with more than 14 million training examples coming from 3555 copyright-free books avalable thanks to Project Gutenberg.

2.1.2 CNN/Daily Mail

In the CNN/DM dataset the context document is formed from a news article while the cloze-style question is formed by removing a named entity from one of the short summary sentences which often appear at the top of the article.

To stop the model from using world knowledge from outside the context article (and hence truly test the comprehension of the article), all named entities were replaced by anonymous tags, which are further shuffled for each example. This may make the comprehension more difficult; however, since the answer is always one of the anonymized entities, it also reduces the number of possible answers making guessing easier.

2.2 Target datasets

2.2.1 bAbI

The first target dataset are the bAbI tasks (Weston et al., 2016) – a set of artificial tasks each of which is designed to test a specific kind of reasoning. This toy dataset will allow us to observe what particular skills the model may be learning from each of the three training datasets.

For our experiments we will be using an architecture designed to select one word from the context document as the answer. Hence we have selected Tasks 1,2,3,4,5,11,12,13,14 and 16 which fulfill this requirement and added task 15 which required a slight modification. Furthermore because both pre-training datasets are cloze-style we converted also the bAbI task questions into cloze style (e.g. ”Where is John?” to ”John is in the XXXXX.”).

For the models pre-trained on CNN/DM we also anonymized the tasks in a way similar to the pre-training dataset - i.e. we replaced all names of characters and also all words that can appear as answers for the given task by anonymous tags in the style of CNN/DM. This gives even models that have not seen any training examples from the target domain a chance to answer the questions.

Full details about these alterations can be found in Appendix A.

2.2.2 SQuAD

Secondly, we will look on transfer to the SQuAD dataset (Rajpurkar et al., 2016); here the associated task may be already useful in the real world. Although cloze-style questions have the huge advantage in the possibility of being automatically generated from a suitable corpus – the path taken by CNN/DM and the BookTest – in practice humans would use a proper question, not its cloze-style substitute. This brings us to the need of transfer from the data-rich cloze-style training to the domain of proper questions where data are much scarcer due to the necessary human annotation.

The SQuAD dataset is a great target dataset to use for this. As opposed to the bAbI tasks, the goal of this dataset is actually a problem whose solving would be useful to humans - answering natural questions based on an natural language encyclopedic knowledge base.

For our experiments we selected only a subset of the SQuAD training and development examples where the answer is only a single word, since this is an inherent assumption of our machine learning model. This way we extracted 28,346 training examples out of the original 100,000 examples and 3,233 development examples out of 10,570.

3 Machine Learning Model: AS Reader

We perform our experiments using the Attention Sum Reader (AS Reader) (Kadlec et al., 2016b) model. The AS Reader is simple to implement while it achieves strong performance on several text comprehension tasks (Kadlec et al., 2016b; Bajgar et al., 2016; Chu et al., 2016). Since the AS Reader is a building block of many recent text-comprehension models (Trischler et al., 2016; Sordoni et al., 2016; Dhingra et al., 2016; Cui et al., 2016b;a; Shen et al., 2016; Munkhdalai & Yu, 2016) it is a good representative of current research in this field.

A high level structure of the AS Reader is shown in Figure 1. The words from the document and the question are first converted into vector embeddings using a look-up matrix. The document is then read by a bidirectional Gated Recurrent Unit (GRU) network (Cho et al., 2014). A concatenation of the hidden states of the forward and backward GRUs at each word is then used as a *contextual embedding* of this word, intuitively representing the context in which the word is appearing. We can also understand it as representing the set of questions to which this word may be an answer.

Similarly the question is read by a bidirectional GRU but in this case only the final hidden states are concatenated to form the *question embedding*.

The attention over each word in the context is then calculated as the dot product of its contextual embedding with the question embedding. This attention is then normalized by the softmax function and summed across all occurrences of each answer candidate. The candidate with most accumulated attention is selected as the final answer.

For a more detailed description of the model including equations check Kadlec et al. (2016b).

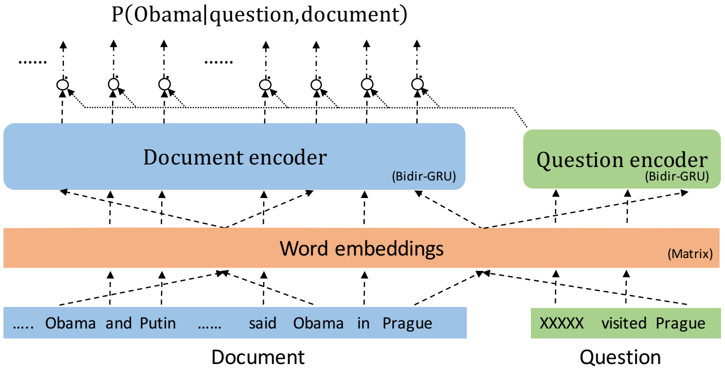


Figure 1: Structure of the AS Reader model.

4 Experiments: Transfer Learning in Text Comprehension

Now let us turn in more detail to the three kinds of experiments that we performed.

4.1 Pre-trained without target adjustment

In the first experiment we tested how a model trained on one of the large-scale pre-training datasets performs on the bAbI tasks without any opportunity to train on bAbI. Since the BookTest and CNN/DM tasks involve only cloze-style questions, we can't expect a model trained on them to answer natural ?-style questions. Hence we did not study the transfer to SQuAD in this case, only the transfer to the (cloze-converted) bAbI tasks.

4.1.1 Method

First we tested how the AS Reader architecture (Kadlec et al., 2016b) can handle the tasks if trained directly on the bAbI training data for each task. Then we tested the degree of transfer from the BookTest and CNN/DM data to the 11 selected bAbI tasks.

In the first part of the experiment we trained a separate instance of the AS Reader on the 10,000-example version of the bAbI training data for each of the 11 tasks (for more details see Appendix B.1). On 8 of them the architecture was able to learn the task with accuracy at least 95% [1] (results for each task can be found in Table 4 in Appendix C). Hence if given appropriate training the AS Reader is capable of the reasoning needed to solve most of the selected bAbI tasks. Now when we know that the AS Reader is powerful enough to learn the target tasks we can turn to transfer from the two large-scale datasets.

The main part of this first experiment was then straightforward: we pre-trained multiple models on the BookTest and CNN/DM datasets and then simply evaluated them on the test datasets of the 11 selected bAbI tasks.

4.1.2 Results

Table 1 summarizes the results of this experiment. Both the models trained on the BookTest and those trained on the CNN/DM dataset perform quite poorly on bAbI and achieve much lower accuracy than

[1] It should be noted that there are several machine learning models that perform better than the AS Reader in the 10k weakly supervised setting, e.g. (Sukhbaatar et al., 2015; Xiong et al., 2016; Graves et al., 2016), however they often need significant fine-tuning. On the other hand we trained plain AS Reader model without any modifications. Hyperparameter and feature fine-tuning could probably further increase its performance on individual tasks however it goes directly against the idea of generality that is at the heart of this work. For comparison with state of the art we include results of DMN+ (Xiong et al., 2016) in Table 1 which had the best average performance over the original 20 tasks.

Table 1: The mean performance across 11 bAbI tasks. The first two columns show a random baseline[2] and a baseline that selects the most frequent word from the context which also appears as an answer in the training data for the task. The following three columns show performance of the AS Reader trained on different datasets, the last column shows the results of DMN+ (Xiong et al., 2016), the state-of-the-art-model on the bAbI 10k dataset. For more detailed results listing per task accuracies see Appendix C.

Model	Rnd.	Most freq. cand.	AS Reader			DMN+
Train dataset	not trained	bAbI 10k	BookTest 14M	CNN/DM 1.2M	bAbI 10k	bAbI 10k
bAbI mean (11 tasks)	6.1	29.9	34.8	38.1	92.7	95.7

the models trained directly on each individual bAbI task. However there is some transfer between the tasks since the AS Reader trained on either the BookTest or CNN/DM outperforms a random baseline[2] and even an improved baseline which selects the most frequent word from the context that also appears as an answer in the training data for this task.

The results also show that the models trained on CNN/DM perform somewhat better on most tasks than the BookTest models. This may be due to the fact that bAbI tasks generally require the model to summarize information from the context document, which is also what the CNN/DM dataset is testing. On the other hand, the BookTest requires prediction of a possible continuation of a story, where the required kind of reasoning is much less clear but certainly different from pure summarization. Another explanation for better performance of CNN/DM models might be that they solve slightly simpler task since the candidate answers were already pre-selected in the entity anonymization step.

Readers interested in how the training-dataset size affects this kind of transfer can check (Kadlec et al., 2016a) where we show that the target-task performance is a bit better if we use the large BookTest as opposed to its smaller subset, the Children's Book Test (CBT) (Hill et al., 2015).

Conclusions from this experiment are that the skills learned from two large-scale datasets generalize surprisingly poorly to even simple toy tasks. This may make us ask whether most teams' focus on solving narrow tasks is truly beneficial if the skills learnt on these tasks are hard to apply elsewhere. However it also brings us to our next experiment, where we try to provide some help to the struggling pre-trained models.

4.2 Pre-trained with target adjustment

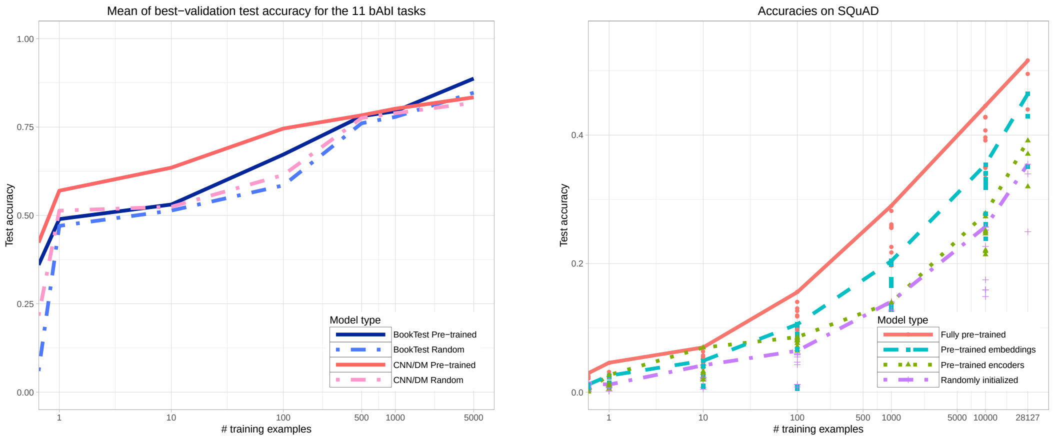
After showing that the skills learnt from the BookTest and CNN/DM datasets are by themselves insufficient for solving the toy tasks, the next natural question is whether they are useful if helped by training on a small sample of examples from the target task. We call this additional phase of training *target adjustment*. For this experiment we again use the bAbI tasks, however we also test transfer to a subset of the SQuAD dataset, which is much closer to real-world natural-language question answering.

The results presented in this and the following section are based on training 3701 model instances.

4.2.1 Method

Common to bAbI and SQuAD datasets. In this experiment we started with a pre-trained model which we used in the previous experiment. However, after it finished training on one of the large pre-training datasets, we allowed it to train on a subset of training examples from the target dataset. We tried subsets of various sizes ranging from a single example to thousands. We tried training four different pre-trained models and also, for comparison, four randomly-initialized models with the same hyperparameters (see Appendix B.2 for details). The experiment with each task-model couple was run on 4 different data samples of each size which were randomly drawn from the training dataset

[2] The random baseline selects randomly uniformly between all unique words contained in the context document.



(a)

(b)

Figure 2: Sub-figure (a) shows the average across the 11 bAbI tasks of the best-validation model's test accuracy. (b) shows the test accuracy on SQuAD of each model we trained (the points) and the lines join the accuracies of the best-validation models for each training size.

of the task to account for variations between these random samples – which may be substantial given the small sample size.[3]

bAbI. For each of these models we observed the test accuracy at the best-validation epoch and compared this number between the randomly initialized and pre-trained models. Validation was done using 100 examples which were set aside from the task's original 10k training data.[4] We perform the experiment with models pre-trained on the BookTest and also on CNN/DM.

SQuAD subset. In the SQuAD experiment, we trained the model on a subset of the original training dataset where answers were only single words and its sub-subsets. We report the best-validation accuracy on a development set filtered in the same way. This experiment was performed only with the models pre-trained on BookTest.

4.2.2 RESULTS

The results of these experiments are summarized in Figures 2 and 3.

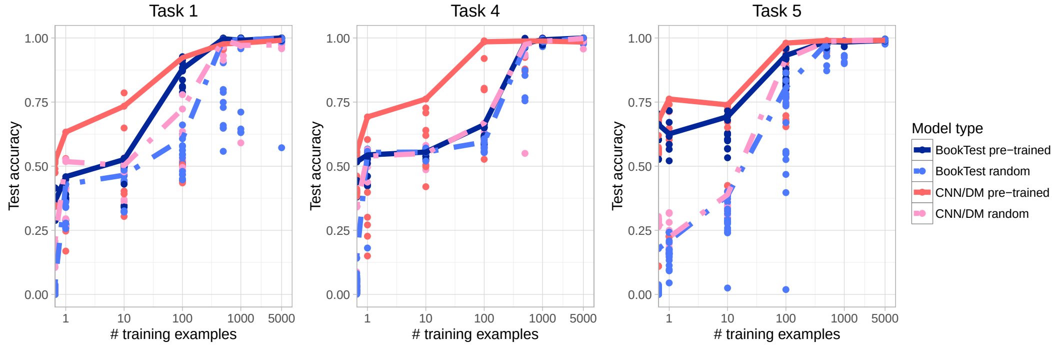


Figure 3: Example of 3 bAbI tasks where pre-training seems to help. Note that the task may be easier for the CNN/DM models due to answer anonymization which restricts the choice of possible answers.

[3] We are planning to release the split training datasets soon.

[4] The other models trained on the full 10k dataset usually use 1000 validation examples (Sukhbaatar et al., 2015; Xiong et al., 2016), however we wanted to focus on low data regime thus we used 10 times less examples.

bAbI. Sub-figure 2a shows mean test accuracy of the models that achieved the best validation result for each single task. The results for both BookTest and CNN/DM experiments confirm positive effect of pre-training compared to randomly initialized baseline. Figure 3 shows performance on selected bAbI tasks where pre-training has clearly positive effect, such plot for each of the target tasks is provided in Appendix C.2 (Figure 4).

Note that the CNN/DM models cannot be directly compared to BookTest results due to entity anonymization that seems to simplify the task when the model is trained on smaller datasets.

Since our evaluation methodology with different training set sizes is novel, we can compare our result only to MemN2N (Sukhbaatar et al., 2015) trained on a 1k dataset. MemN2N is the only weakly supervised model that reports accuracy when trained on less than 10k examples. MemN2N achieves average accuracy 93.2%[5] on the eleven selected tasks. This is substantially better than both our random baseline (78.0%) and the BookTest-pre-trained model (79.5%), however our model is not tuned in any way towards this particular task. One important conceptual difference is that the AS Reader processes the whole context as one sequence of words, whereas MemN2N receives the context split into single sentences, which simplifies the task for the network.

SQuAD subset. The results of SQuAD experiment also confirm positive effect of pre-training, see Sub-figure 2b, for now compare just lines showing performance of the fully pre-trained model and the randomly initialized model – the meaning of the remaining two lines shall become clear in the next section.

More detailed statistics about the results of this experiment can be found in Appendix D.

We should note that performance of our model is not competitive with the state of the art models on this dataset. For instance the DCR model (Yu et al., 2016) trained on our SQuAD subset achieves validation accuracy 74.9% in this task which is better than our randomly initialized (35.4%) and pre-trained (51.6%) models[6]. However, the DCR model is designed specifically for the SQuAD task, for instance it utilizes features that are not used by our model.

4.3 Partially pre-trained model

Since our previous experiment confirmed positive effect of pre-training if followed by target-domain adjustment, we wondered which part of the model contains the knowledge transferable to new domains. To examine this we performed the following experiment.

4.3.1 Method

Our machine learning model, the AS Reader, consists of two main parts: the word-embedding look-up and the bidirectional GRUs used to encode the document and question (see Figure 1). Therefore a natural question was what the contribution of each of these parts is.

To test this we created two models out of each pre-trained model used in the previous experiment. The first model variant uses the pre-trained word embeddings from the original model while the GRU encoders are randomly initialized. We say that this model has *pre-trained embeddings*. The second model variant uses the opposite setting where the word embeddings are randomly initialized while the encoders are taken form a pre-trained model. We call this *pre-trained encoders*.

bAbI. For this experiment we selected only a subset of tasks with training set of 100 examples where there was significant difference in accuracy between randomly-initialized and pre-trained models. For evaluation we use the same methodology as in the previous experiment, that is, we report accuracy of the best-validation model averaged over 4 training splits.

SQuAD subset. We evaluated both model variants on all training sets from the previous SQuAD experiment using the same methodology.

[5] MemN2N trained on each single task with PE LS RN features, see (Sukhbaatar et al., 2015) for details.

[6] We would like to thank Yu et al. (2016) for training their system on our dataset.

Table 2: The effect of pre-training different components of the model for selected tasks. The first row shows performance (average test accuracy across all trained model instances in each category) of a randomly initialized baseline model. The following three rows show increase in accuracy (measured in percent absolute) when the model is initialized with weights pre-trained on the BookTest. The last line shows results for models initialized with Google News word2vec word embeddings (Mikolov et al., 2013).

Task	bAbI task (100 ex.)				SQuAD
Model variant	1.	5.	11.	14.	(28k ex.)
Random init	53%	66%	71%	33%	31%
Δ Pre-trained encoders	+6	+25	+4	+2	+4
Δ Pre-trained embeddings	+17	+6	+8	+8	+10
Δ Pre-trained full	+34	+22	+14	+13	+17
Δ Pre-trained word2vec	-2	+5	+1	-1	+5

4.3.2 Results

bAbI. Table 2 shows improvement of pre-trained models over a randomly initialized baseline. In most cases (all except Task 5) the fully pre-trained model achieved the best accuracy.

SQuAD subset. The accuracies of the four model variants are plotted in Figure 2b together with results of the previous SQuAD experiment. The graph shows that both pre-trained embeddings and pre-trained encoders alone improve performance over the randomly initialized baseline, however the fully pre-trained model is always the best.

The overall result of this experiment is that both pre-training of the word embeddings and pre-training of the encoder parameters are important since the fully pre-trained model outperforms both partially pre-trained variants.

5 Conclusion

Our experiments show that transfer from two large cloze-style question-answering datasets to our two target tasks is suprisingly poor, if the models aren't provided with any examples from the target domain. However we show that models that pre-trained models perform significantly better than a randomly initialized model if they are shown at least a few training examples from the target domain. The usefulness of pre-trained word embeddings is well known in the NLP community however we show that the power of our pre-trained model does not lie just in the embeddings. This suggests that once the text-comprehension community agrees on sufficiently versatile model, much larger parts of the model could start being reused than just the word-embeddings.

The generalization of skills from a training domain to new tasks is an important ingredient of any system we would want to call intelligent. This work is an early step to explore this direction.

References

Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. Embracing data abundance: BookTest Dataset for Reading Comprehension. *arXiv preprint arXiv:1610.00956*, 2016.

Murray Campbell, A Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1): 57–83, 2002.

Danqi Chen, Jason Bolton, and Christopher D. Manning. A Thorough Examination of the CNN / Daily Mail Reading Comprehension Task. In *Association for Computational Linguistics (ACL)*, 2016.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for

Statistical Machine Translation. *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. URL http://arxiv.org/abs/1406.1078v3.

Zewei Chu, Hai Wang, Kevin Gimpel, and David Mcallester. Broad Context Language Modeling as Reading Comprehension. 2016.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch. *Journal ofMachine Learning Research 12*, 12:2461–2505, 2011.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-Attention Neural Networks for Reading Comprehension. 2016a. URL http://arxiv.org/abs/1607.04423.

Yiming Cui, Ting Liu, Zhipeng Chen, Shijin Wang, and Guoping Hu. Consensus Attention-based Neural Networks for Chinese Reading Comprehension. 2016b.

Andrew M. Dai and Quoc V. Le. Semi-supervised Sequence Learning. *NIPS*, 2015. ISSN 10495258. URL http://arxiv.org/abs/1511.01432.

Bhuwan Dhingra, Hanxiao Liu, William W. Cohen, and Ruslan Salakhutdinov. Gated-Attention Readers for Text Comprehension. 2016. URL http://arxiv.org/abs/1606.01549.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid Computing Using a Neural Network with Dynamic External Memory. *Nature*, 2016. ISSN 0028-0836. doi: 10.1038/nature20101.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A JOINT MANY-TASK MODEL: GROWING A NEURAL NETWORK FOR MULTIPLE NLP TASKS. *submitted to ICLR 2017*, 2016.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1684–1692, 2015.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. From Particular to General : A Preliminary Case Study of Transfer Learning in Reading Comprehension. *MAIN Workshop at NIPS*, 2016a.

Rudolf Kadlec, Martin Schmid, Ondej Bajgar, and Jan Kleindienst. Neural Text Understanding with Attention Sum Reader. *Proceedings of ACL*, 2016b.

Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, and Kentaro Inui. Dynamic Entity Representation with Max-pooling Improves Machine Reading. *Proceedings of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies (NAACL-HLT)*, 2016.

Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu. Dataset and Neural Recurrent Sequence Labeling Model for Open-Domain Factoid Question Answering. 2016. URL https://arxiv.org/abs/1607.06275.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 2013. ISSN 15324435. doi: 10.1162/153244303322533223. URL http://arxiv.org/pdf/1301.3781v3.pdf.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. How Transferable are Neural Networks in NLP Applications? *EMNLP*, 2016.

Tsendsuren Munkhdalai and Hong Yu. Reasoning with Memory Augmented Neural Networks for Language Comprehension. 2016. URL https://arxiv.org/abs/1610.06454v1.

Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, oct 2010. ISSN 1041-4347. doi: 10.1109/TKDE.2009.191. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5288526.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. (ii), 2016. URL http://arxiv.org/abs/1606.05250.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. ReasoNet: Learning to Stop Reading in Machine Comprehension. 2016. URL http://arxiv.org/abs/1609.05284.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. ISSN 0028-0836. doi: 10.1038/nature16961. URL http://dx.doi.org/10.1038/nature16961.

Alessandro Sordoni, Phillip Bachman, and Yoshua Bengio. Iterative Alternating Neural Attention for Machine Reading. 2016.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-To-End Memory Networks. pp. 1–11, 2015. URL http://arxiv.org/abs/1503.08895.

Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. Natural Language Comprehension with the EpiReader. 2016. URL http://arxiv.org/abs/1606.02270.

Dirk Weissenborn. Separating Answers from Queries for Neural Reading Comprehension. 2016. URL http://arxiv.org/abs/1607.03316.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merri, Armand Joulin, and Tomas Mikolov. Towards AI-complete Question Answering: A Set of Prerequisite Toy Tasks. 2016. URL https://arxiv.org/abs/1502.05698.

Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic Memory Networks for Visual and Textual Question Answering. *ICML*, 2016. URL http://arxiv.org/abs/1603.01417.

Yang Yu, Wei Zhang, Kazi Hasan, Mo Yu, Bing Xiang, and Bowen Zhou. End-to-End Reading Comprehension with Dynamic Answer Chunk Ranking. (1), 2016. URL http://arxiv.org/abs/1610.09996.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

A CLOZE STYLE BABI DATASET

Since our AS Reader architecture is designed to select a single word from the context document as an answer (the task of CBT and BookTest), we selected 10 bAbI tasks that fulfill this requirement out of the original 20. These tasks are: *1. single supporting fact, 2. two supporting facts, 3. three supporting facts, 4. two argument relations, 5. three argument relations, 11. basic coreference, 12. conjunction, 13. compound coreference, 14. time reasoning* and *16. basic induction*.

Task 15 needed a slight modification to satisfy this requirement: we converted the answers into plural (e.g. ”Q: What is Gertrude afraid of? A: wolf.” was converted into ”A: wolves” which also seems to be the more natural way to formulate the answer to such a question.).

Also since CBT and BookTest train the model for Cloze-style question answering, we modify the original bAbI dataset by reformulating the questions into Cloze-style. For example we translate a question ”Where is John ?” to ”John is in the XXXXX .”

For the models pre-trained on CNN/DM we also replace two kinds of words by anonymized tags (e.g. "@entity56") in a style similar to the pre-training dataset. Specifically we replace two (largely overlapping) categories of words:

1. Proper names of story characters (e.g. John, Sandra)
2. Any word that can appear as an answer for the particular task (e.g. kitchen, garden if the task is asking about locations).

B Method details

B.1 Direct training on bAbI – method

Here we give a more detailed description of the method we used to arrive to our results. We highlight only facts particular to this experiment. A more detailed general description of training the AS Reader is given in (Kadlec et al., 2016b).

The results given for AS Reader trained on bAbI are each for a single model with 64 hidden units in each direction of the GRU context encoder and embedding dimension 32 trained on the 10k training data provided with that particular task.

The results for AS Reader trained on the BookTest and the CNN/DM are for a greedy ensemble consisting of 4 models whose predictions were simply averaged. The models and ensemble were all validated on the validation set corresponding to the training dataset. The performance on the bAbI tasks oscillated notably during training however the ensemble averaging does somewhat mitigate this to get more representative numbers.

B.2 Hyperparameters for the target-adjustment experiments

Table 3 lists hyperparameters of the pre-trained AS Reader instances used in our experiments with target adjustment.

Table 3: Hyperparameters for both the randomly initialized and the pre-trained models.

Dataset	Hid. Units	Emb.	L. rate	Dropout
BookTest	768	256	0.0001	0
BookTest	384	384	0.0005	0.2
BookTest	384	384	0.0005	0.4
BookTest	512	384	0.0001	0
CNN/DM	128	128	0.001	0
CNN/DM	256	128	0.001	0
CNN/DM	384	128	0.001	0
CNN/DM	384	384	0.001	0

C Detailed results

C.1 Experiments without target adjustment

Table 4 shows detailed results for the experiments on models which were just pre-trained on one of the pre-training datasets without any target-adjustment. It also shows several baselines and results of a state-of-the-art model.

C.2 Target-adjustment experiments

C.2.1 Results for all bAbI tasks

Figure 4 shows the test accuracies of all models that we trained in the target-adjustment experiments as well as lines joining the accuracies of the best-validation models.

Table 4: Performance of the AS Reader when trained on the bAbI 10k, BookTest and CNN/DM datasets and then evaluated on bAbI test data. The Dynamic Memory Network (DMN+) is the state-of-the-art model in a weakly supervised setting on the bAbI 10k dataset. Its results are taken from (Xiong et al., 2016). MemN2N (Sukhbaatar et al., 2015) is the state-of-the-art model on the 1k training dataset; for completeness we also include its results with the 10k training.

	Model:	Random	Rnd cand.	MemN2N (single) (PE LS RN)	MemN2N (single) (PE LS LW RN)	DMN+ (single)	ASReader		
	Train dataset Test dataset	not trained	bAbI 10k	bAbI 1k	bAbI 10k	bAbI 10k	bAbI 10k	BookTest 14M	DM+CNN 1.2M
1	Single supporting fact	7.80	31.20	100.00	100.00	100.00	100.00	37.30	51.50
2	Two supporting facts	4.40	26.96	91.70	99.70	99.70	91.90	25.80	28.90
3	Three supporting facts	3.40	19.14	59.70	97.90	98.90	86.00	22.20	27.40
4	Two-argument relations	10.50	33.58	97.20	100.00	100.00	100.00	50.30	54.90
5	Three-argument relations	4.40	21.42	86.90	99.20	99.50	99.80	67.60	68.10
11	Basic coreference	6.20	30.42	99.10	99.90	100.00	100.00	33.00	20.80
12	Conjunction	6.70	27.25	99.80	100.00	100.00	100.00	30.40	37.70
13	Compound coreference	5.60	27.73	99.60	100.00	100.00	100.00	33.80	14.00
14	Time reasoning	5.00	27.82	98.30	99.90	99.80	95.00	27.60	50.50
15	Basic deduction	5.20	37.20	100.00	100.00	100.00	96.70	39.90	17.60
16	Basic induction	7.50	45.65	98.70	48.20	54.70	50.30	15.10	48.00
	bAbI mean (11 tasks)	6.06	29.85	93.73	94.98	95.69	92.70	34.82	38.13

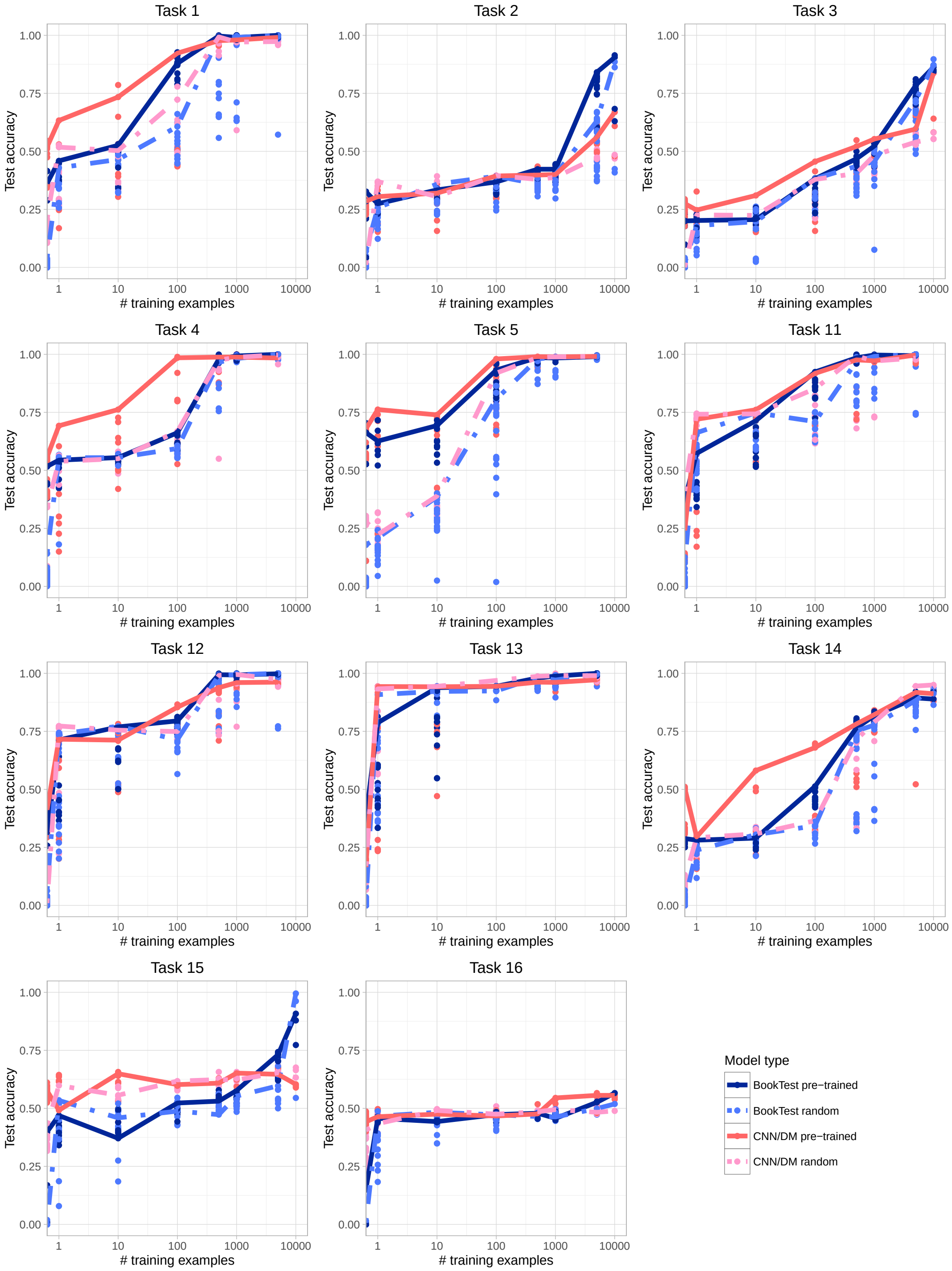


Figure 4: The test accuracies of all models that we trained in the target-adjustment experiments. The line joins the test accuracies of the best-validation models of each model type.

C.2.2 Average over all models trained on bAbI tasks

Figure 5 plots mean accuracy of all models trained in our experiments. This suggests that pre-training helped all models, not only the top performing ones selected by validation as already shown in Figure 2a.

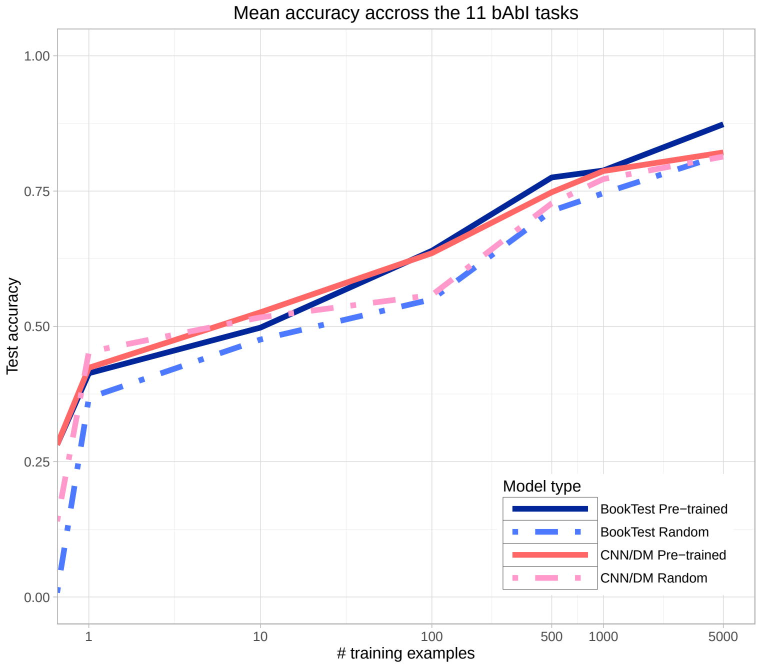


Figure 5: The average of the mean test accuracies across the 11 bAbI tasks. For the average of the best validation results see Figure 2a.

D Means, standard deviations and p-values by experiment

Table 5 shows the mean accuracy across all models trained for each combination of task, pre-training dataset and target-adjustment dataset size. Table 6 shows the corresponding standard deviations.

Table 7 then shows the p-value that whether the expected accuracy of pre-trained models is greater than the expected accuracy of randomly initialized models. This shows that the pre-trained models are statistically significantly better for all target-adjustment set sizes on the SQuAD dataset. On bAbI the BookTest pre-trained models perform convincingly better especially for target-adjustment dataset sizes 100, 500 and 1000, with Task 16 being the main exception to this because the AS Reader struggles to learn it in any setting. For the CNN+DM pre-training the results are not conclusive.

Task	Pretrain. set	Model	Target-adjustment set size								
			0	1	10	100	500	1000	5000	10000	28174
SQuAD	BookTest	pre-trained	0.025	0.027	0.049	0.122	NA	0.245	NA	0.388	0.484
SQuAD	BookTest	rand. init.	0.004	0.006	0.018	0.042	NA	0.107	NA	0.214	0.315
Task 1	BookTest	pre-trained	0.356	0.383	0.459	0.870	0.992	0.995	0.999	NA	NA
Task 1	BookTest	rand. init.	0.010	0.327	0.431	0.529	0.888	0.916	0.976	NA	NA
Task 1	CNN+DM	pre-trained	0.295	0.385	0.519	0.689	0.969	0.985	0.990	NA	NA
Task 1	CNN+DM	rand. init.	0.100	0.354	0.450	0.582	0.954	0.941	0.977	NA	NA
Task 2	BookTest	pre-trained	0.206	0.295	0.318	0.339	0.398	0.410	0.755	0.783	NA
Task 2	BookTest	rand. init.	0.003	0.225	0.290	0.332	0.358	0.361	0.528	0.645	NA
Task 2	CNN+DM	pre-trained	0.177	0.265	0.288	0.359	0.410	0.398	0.539	0.586	NA
Task 2	CNN+DM	rand. init.	0.005	0.280	0.320	0.380	0.371	0.396	0.478	0.469	NA
Task 3	BookTest	pre-trained	0.159	0.192	0.227	0.314	0.440	0.508	0.759	0.857	NA
Task 3	BookTest	rand. init.	0.005	0.135	0.182	0.219	0.370	0.419	0.542	0.482	NA
Task 3	CNN+DM	pre-trained	0.164	0.213	0.222	0.303	0.450	0.489	0.585	0.687	NA
Task 3	CNN+DM	rand. init.	0.001	0.175	0.227	0.272	0.385	0.429	0.551	0.563	NA
Task 4	BookTest	pre-trained	0.452	0.490	0.545	0.631	0.986	0.989	1.000	NA	NA
Task 4	BookTest	rand. init.	0.032	0.532	0.556	0.582	0.846	0.982	0.993	NA	NA
Task 4	CNN+DM	pre-trained	0.323	0.413	0.596	0.766	0.946	0.986	0.992	NA	NA
Task 4	CNN+DM	rand. init.	0.234	0.536	0.554	0.593	0.926	0.990	0.986	NA	NA
Task 5	BookTest	pre-trained	0.601	0.604	0.632	0.877	0.983	0.982	0.991	NA	NA
Task 5	BookTest	rand. init.	0.013	0.162	0.295	0.635	0.964	0.973	0.989	NA	NA
Task 5	CNN+DM	pre-trained	0.448	0.492	0.581	0.842	0.969	0.984	0.989	NA	NA
Task 5	CNN+DM	rand. init.	0.185	0.252	0.350	0.844	0.982	0.984	0.988	NA	NA
Task 11	BookTest	pre-trained	0.334	0.415	0.620	0.847	0.986	0.988	0.998	NA	NA
Task 11	BookTest	rand. init.	0.008	0.540	0.692	0.711	0.922	0.951	0.974	NA	NA
Task 11	CNN+DM	pre-trained	0.119	0.492	0.671	0.762	0.820	0.972	0.977	NA	NA
Task 11	CNN+DM	rand. init.	0.207	0.679	0.737	0.734	0.853	0.934	0.980	NA	NA
Task 12	BookTest	pre-trained	0.307	0.429	0.694	0.786	0.988	0.991	0.999	NA	NA
Task 12	BookTest	rand. init.	0.006	0.499	0.705	0.721	0.917	0.966	0.962	NA	NA
Task 12	CNN+DM	pre-trained	0.236	0.518	0.650	0.779	0.866	0.968	0.970	NA	NA
Task 12	CNN+DM	rand. init.	0.009	0.661	0.765	0.735	0.855	0.921	0.965	NA	NA
Task 13	BookTest	pre-trained	0.330	0.505	0.793	0.944	0.959	0.976	0.998	NA	NA
Task 13	BookTest	rand. init.	0.004	0.617	0.920	0.937	0.950	0.966	0.992	NA	NA
Task 13	CNN+DM	pre-trained	0.114	0.612	0.830	0.942	0.949	0.946	0.975	NA	NA
Task 13	CNN+DM	rand. init.	0.094	0.828	0.941	0.944	0.951	0.961	0.971	NA	NA
Task 14	BookTest	pre-trained	0.270	0.266	0.273	0.465	0.775	0.807	0.896	0.912	NA
Task 14	BookTest	rand. init.	0.007	0.228	0.277	0.328	0.597	0.675	0.852	0.905	NA
Task 14	CNN+DM	pre-trained	0.280	0.314	0.351	0.458	0.677	0.790	0.840	0.904	NA
Task 14	CNN+DM	rand. init.	0.054	0.247	0.297	0.337	0.543	0.788	0.901	0.929	NA
Task 15	BookTest	pre-trained	0.085	0.417	0.436	0.491	0.544	0.546	0.689	0.853	NA
Task 15	BookTest	rand. init.	0.003	0.414	0.430	0.496	0.517	0.523	0.584	0.834	NA
Task 15	CNN+DM	pre-trained	0.563	0.604	0.591	0.608	0.611	0.635	0.644	0.597	NA
Task 15	CNN+DM	rand. init.	0.392	0.469	0.534	0.587	0.623	0.630	0.656	0.658	NA
Task 16	BookTest	pre-trained	0.036	0.456	0.451	0.465	0.469	0.474	0.528	0.566	NA
Task 16	BookTest	rand. init.	0.001	0.363	0.449	0.460	0.469	0.475	0.489	0.519	NA
Task 16	CNN+DM	pre-trained	0.444	0.467	0.468	0.474	0.480	0.505	0.519	0.547	NA
Task 16	CNN+DM	rand. init.	0.280	0.428	0.480	0.476	0.483	0.489	0.489	0.496	NA

Table 5: Mean test accuracy for each combination of task, model type and target-adjustment set size.

Task	Pretrain. set	Model	Target-adjustment set size								
			0	1	10	100	500	1000	5000	10000	28174
SQuAD	BookTest	pre-trained	0.025	0.027	0.049	0.122	NA	0.245	NA	0.388	0.484
SQuAD	BookTest	rand. init.	0.004	0.006	0.018	0.042	NA	0.107	NA	0.214	0.315
Taks 1	BookTest	pre-trained	0.356	0.383	0.459	0.870	0.992	0.995	0.999	NA	NA
Taks 1	BookTest	rand. init.	0.010	0.327	0.431	0.529	0.888	0.916	0.976	NA	NA
Taks 1	CNN+DM	pre-trained	0.295	0.385	0.519	0.689	0.969	0.985	0.990	NA	NA
Taks 1	CNN+DM	rand. init.	0.100	0.354	0.450	0.582	0.954	0.941	0.977	NA	NA
Taks 2	BookTest	pre-trained	0.206	0.295	0.318	0.339	0.398	0.410	0.755	0.783	NA
Taks 2	BookTest	rand. init.	0.003	0.225	0.290	0.332	0.358	0.361	0.528	0.645	NA
Taks 2	CNN+DM	pre-trained	0.177	0.265	0.288	0.359	0.410	0.398	0.539	0.586	NA
Taks 2	CNN+DM	rand. init.	0.005	0.280	0.320	0.380	0.371	0.396	0.478	0.469	NA
Taks 3	BookTest	pre-trained	0.159	0.192	0.227	0.314	0.440	0.508	0.759	0.857	NA
Taks 3	BookTest	rand. init.	0.005	0.135	0.182	0.219	0.370	0.419	0.542	0.482	NA
Taks 3	CNN+DM	pre-trained	0.164	0.213	0.222	0.303	0.450	0.489	0.585	0.687	NA
Taks 3	CNN+DM	rand. init.	0.001	0.175	0.227	0.272	0.385	0.429	0.551	0.563	NA
Taks 4	BookTest	pre-trained	0.452	0.490	0.545	0.631	0.986	0.989	1.000	NA	NA
Taks 4	BookTest	rand. init.	0.032	0.532	0.556	0.582	0.846	0.982	0.993	NA	NA
Taks 4	CNN+DM	pre-trained	0.323	0.413	0.596	0.766	0.946	0.986	0.992	NA	NA
Taks 4	CNN+DM	rand. init.	0.234	0.536	0.554	0.593	0.926	0.990	0.986	NA	NA
Taks 5	BookTest	pre-trained	0.601	0.604	0.632	0.877	0.983	0.982	0.991	NA	NA
Taks 5	BookTest	rand. init.	0.013	0.162	0.295	0.635	0.964	0.973	0.989	NA	NA
Taks 5	CNN+DM	pre-trained	0.448	0.492	0.581	0.842	0.969	0.984	0.989	NA	NA
Taks 5	CNN+DM	rand. init.	0.185	0.252	0.350	0.844	0.982	0.984	0.988	NA	NA
Taks 11	BookTest	pre-trained	0.334	0.415	0.620	0.847	0.986	0.988	0.998	NA	NA
Taks 11	BookTest	rand. init.	0.008	0.540	0.692	0.711	0.922	0.951	0.974	NA	NA
Taks 11	CNN+DM	pre-trained	0.119	0.492	0.671	0.762	0.820	0.972	0.977	NA	NA
Taks 11	CNN+DM	rand. init.	0.207	0.679	0.737	0.734	0.853	0.934	0.980	NA	NA
Taks 12	BookTest	pre-trained	0.307	0.429	0.694	0.786	0.988	0.991	0.999	NA	NA
Taks 12	BookTest	rand. init.	0.006	0.499	0.705	0.721	0.917	0.966	0.962	NA	NA
Taks 12	CNN+DM	pre-trained	0.236	0.518	0.650	0.779	0.866	0.968	0.970	NA	NA
Taks 12	CNN+DM	rand. init.	0.009	0.661	0.765	0.735	0.855	0.921	0.965	NA	NA
Taks 13	BookTest	pre-trained	0.330	0.505	0.793	0.944	0.959	0.976	0.998	NA	NA
Taks 13	BookTest	rand. init.	0.004	0.617	0.920	0.937	0.950	0.966	0.992	NA	NA
Taks 13	CNN+DM	pre-trained	0.114	0.612	0.830	0.942	0.949	0.946	0.975	NA	NA
Taks 13	CNN+DM	rand. init.	0.094	0.828	0.941	0.944	0.951	0.961	0.971	NA	NA
Taks 14	BookTest	pre-trained	0.270	0.266	0.273	0.465	0.775	0.807	0.896	0.912	NA
Taks 14	BookTest	rand. init.	0.007	0.228	0.277	0.328	0.597	0.675	0.852	0.905	NA
Taks 14	CNN+DM	pre-trained	0.280	0.314	0.351	0.458	0.677	0.790	0.840	0.904	NA
Taks 14	CNN+DM	rand. init.	0.054	0.247	0.297	0.337	0.543	0.788	0.901	0.929	NA
Taks 15	BookTest	pre-trained	0.085	0.417	0.436	0.491	0.544	0.546	0.689	0.853	NA
Taks 15	BookTest	rand. init.	0.003	0.414	0.430	0.496	0.517	0.523	0.584	0.834	NA
Taks 15	CNN+DM	pre-trained	0.563	0.604	0.591	0.608	0.611	0.635	0.644	0.597	NA
Taks 15	CNN+DM	rand. init.	0.392	0.469	0.534	0.587	0.623	0.630	0.656	0.658	NA
Taks 16	BookTest	pre-trained	0.036	0.456	0.451	0.465	0.469	0.474	0.528	0.566	NA
Taks 16	BookTest	rand. init.	0.001	0.363	0.449	0.460	0.469	0.475	0.489	0.519	NA
Taks 16	CNN+DM	pre-trained	0.444	0.467	0.468	0.474	0.480	0.505	0.519	0.547	NA
Taks 16	CNN+DM	rand. init.	0.280	0.428	0.480	0.476	0.483	0.489	0.489	0.496	NA

Table 6: Standard deviation in accuracies for each combination of task, model type and target-adjustment set size.

Task	Pretraining	Target-adjustment set size								
		0	1	10	100	500	1000	5000	10000	28174
SQuAD	BookTest	1.01e-45	4.07e-05	7.40e-05	7.82e-08	NA	5.17e-08	NA	3.93e-08	8.52e-03
Task 1	BookTest	3.34e-83	1.81e-03	1.33e-01	2.35e-19	9.41e-04	1.67e-02	1.32e-01	NA	NA
Task 2	BookTest	1.24e-34	3.86e-07	7.29e-03	2.59e-01	1.39e-08	2.63e-06	7.54e-09	2.04e-01	NA
Task 3	BookTest	9.84e-55	1.27e-05	7.66e-03	1.48e-03	3.18e-04	2.18e-03	2.16e-04	1.03e-01	NA
Task 4	BookTest	7.25e-78	9.50e-01	9.71e-01	1.04e-05	6.38e-03	1.70e-02	1.81e-02	NA	NA
Task 5	BookTest	6.55e-115	9.88e-22	8.87e-19	5.25e-05	3.66e-03	8.61e-02	5.65e-03	NA	NA
Task 11	BookTest	6.78e-152	1.00e+00	9.94e-01	4.07e-09	2.50e-04	2.28e-02	6.37e-02	NA	NA
Task 12	BookTest	2.27e-90	9.10e-01	6.46e-01	1.89e-05	2.78e-04	1.43e-02	2.36e-02	NA	NA
Task 13	BookTest	5.30e-91	9.75e-01	9.99e-01	2.88e-02	2.74e-02	1.03e-01	7.06e-02	NA	NA
Task 14	BookTest	1.97e-200	1.01e-03	6.79e-01	2.22e-14	3.40e-05	2.93e-03	3.66e-06	3.97e-01	NA
Task 15	BookTest	3.64e-09	4.75e-01	4.12e-01	6.70e-01	1.68e-03	3.70e-03	1.03e-05	4.54e-01	NA
Task 16	BookTest	1.81e-05	8.28e-04	4.38e-01	2.72e-01	4.89e-01	5.71e-01	7.40e-03	NA	NA
Task 1	CNN+DM	9.43e-09	2.99e-01	1.11e-01	1.05e-01	9.54e-02	1.45e-01	3.97e-03	NA	NA
Task 2	CNN+DM	9.38e-17	6.93e-01	9.02e-01	9.15e-01	1.05e-03	4.20e-01	2.64e-03	8.49e-02	NA
Task 3	CNN+DM	2.42e-16	4.95e-02	6.30e-01	1.75e-01	2.13e-03	6.59e-04	4.68e-02	1.24e-01	NA
Task 4	CNN+DM	5.84e-03	9.70e-01	1.37e-01	4.83e-03	3.33e-01	8.84e-01	1.08e-01	NA	NA
Task 5	CNN+DM	1.17e-10	7.00e-03	7.93e-04	5.20e-01	9.70e-01	5.66e-01	1.83e-01	NA	NA
Task 11	CNN+DM	1.00e+00	9.84e-01	9.73e-01	2.58e-01	7.17e-01	1.45e-01	6.95e-01	NA	NA
Task 12	CNN+DM	1.93e-14	9.32e-01	9.92e-01	2.57e-02	4.06e-01	6.65e-02	2.09e-01	NA	NA
Task 13	CNN+DM	8.69e-02	9.61e-01	9.72e-01	9.89e-01	6.22e-01	9.44e-01	2.83e-01	NA	NA
Task 14	CNN+DM	2.17e-12	6.64e-02	1.11e-01	2.05e-02	3.66e-02	4.52e-01	9.10e-01	8.24e-01	NA
Task 15	CNN+DM	1.36e-52	5.30e-03	3.48e-02	7.21e-02	8.36e-01	3.09e-01	8.47e-01	9.84e-01	NA
Task 16	CNN+DM	6.39e-35	4.56e-02	9.66e-01	5.95e-01	7.19e-01	4.09e-02	2.51e-02	2.22e-03	NA

Table 7: One-sided p-value whether the mean accuracy of pre-trained models is greater than the accuracy of the randomly initialized ones for each combination of task pre-training dataset. p-values below 0.05 are marked in green.