Matching pre-training and Fine-tuning Methods for Knowledge Retrieval from pretrained Language Models

Ahmad Pouramini [*1] **Hesham Faili** [*1]

Abstract

In this paper we study different methods for pre-training and fine-tuning a transformer-based language model for generating commonsense knowledge or the KB completion task in few-shot settings. The model can be trained in unsupervised and supervised methods with different pre-training objectives. We investigate the effect of each type of these training objectives on the performance of the model in knowledge generation and retrieval. We analyze the results from both plausibility and variety and novelty aspects. The results show that mixing both objectives in pre-training and fine-tuning stages can provide more novel and accurate results in few shot settings. These considerations can be taken into account for selecting and fine-tuning a model for a specific task.

1. Introduction

Recently pre-trained language models (PLMs) have been used as an alternative for knowledge bases. Petroni et al. tried to retrieve factual and commonsense knowledge directly from these models by converting a query into cloze-style prompts (Petroni et al., 2020). They found that this approach is more suitable for retrieving one-to-one factual knowledge. When prompted to complete commonsense declarative relationships, PLMs exhibit limited ability to map their language modeling abilities to this task (Petroni et al., 2020; Da et al., 2021b).

Other works attempted to fine-tune a PLM on a commonsense knowledge graph tuples for KB completion and generation of commonsense tuples (Bosselut et al., 2020; Hwang et al., 2020; Da et al., 2021a). The tuples are typically in the form of {head, relation, tail}. The model must learn the commonsense relationships and generalize them to situations it has not seen during fine-tuning and provide a plausible tail.

In these works, it is assumed that commonsense knowledge is implicitly encoded in the pretrained model, and fine-tuning serves to learn an interface to the encoded knowledge (Hwang et al., 2020). With this assumption, a commonsense knowledge model can be trained effectively in a few-shot setting to hypothesize commonsense knowledge (Da et al., 2021a).

One of the methods that can accelerate learning commonsense knowledge is the use of natural language prompts to elicit knowledge from PLMs (Feldman et al., 2020; Da et al., 2021a). Prompts are mainly effective in a few-shot setting where there is little signal to learn a relation embedding from scratch.

The related works have usually employed encoder-decoder models such as T5 (Raffel et al., 2020) to generate the tail of a triplet by feeding the model with the head and the relation. The relation is either a special token appended to the head as a prefix, or is represented using natural language prompts.

However the current works are often fine-tune the model in a supervised fashion where a mapping must be made between the input and output. One drawback of this method is that the model may overfit to the training data and its generalization power decreases. Also in few-shot settings, its accuracy may decrease due to the lack of sufficient training examples. Therefore, a trade-off must be made between the diversity and novelty of the results that comes from the pre-training stage and their accuracy and generalization that are gained during fine-tuning.

We assumed that a fine-tuning method that is more similar to the unsupervised method in which the model was pretrained on a large amount of text data can provide better results in some cases and allows for a better distillation of the knowledge that is encoded in the language model.

In this work, we investigate the performance of a commonsense knowledge model under different pre-training and fine-tuning settings for few-shot learning. We assume that commonsense knowledge can be retrieved from pre-trained

[*] Equal contribution [1] Department of Computer Engineeing, University of Tehran, Tehran, Iran. Correspondence to: Ahmad Pouramini <ahmad.pouramini@ut.ac.ir>.

ICML workshop on Knowledge Retrieval and Language Models, Baltimore, Maryland, USA, 2022.

Table 1. Examples of inputs and targets for two unsupervised objectives in the T5 model (Raffel et al., 2020)

Objective	Input	Target
prefix language modeling	PersonX goes to a library, because he wants	to borrow a book
I.i.d noise replace spans	personX teaches X in schools. He is seen as Y	X history Y knowledgeable

Table 2. different versions of T5 models based on the pre-training objective

Model	Pre-training Objective	Data Sets
t5-v1	I.i.d noise replace spans	C4 corpus
t5-lm	noise replace spans + Language modeling	C4 + 100K steps on the LM objective
t5-base	noise replace spans + supervised text-to-text	C4 + WikiDPR — various supervised tasks

language models because it must generate a plausible output in diverse natural language phrases. However, our results can be generalized to other types of knowledge such as factual knowledge.

2. Experimental Setup

2.1. Model

In the related works several models such as GPT2-LX as an autoregressive model, and BART and T5 as encoder-decoder models were employed. Among them, the BART and T5 models provided better results. The GPT model tended to copy the input or generate unnecessary full sentences particularly in few-shot settings (Da et al., 2021b;a) . This issue was also observed in our experiments. Therefore, we decided to focus on the T5 model, which provided better results in few-shot settings and uses both encoder and decoder parts of the transformer model (Vaswani et al., 2017).

The T5 model is an encoder-decoder model pre-trained in unsupervised and supervised methods with different pre-training objectives. In the following, we review its pre-training objectives.

- **Supervised Training** In this setup, the input sequence and output sequence are a standard sequence-to-sequence input-output mapping. A prefix representing the task instruction can be appended to each input example.

- **Unsupervised Training** The T5 model was pre-trained on a huge collection of unlabeled texts in an unsupervised method. Various techniques can be used to format a sentence as input and output of the model. In Table 1 two objectives are shown. In a basic ”prefix language modeling” a span of text is randomly split into prefix and target portions, one to use as inputs to the encoder and the other to use as a target sequence.

  In a denoising objective, spans of the input sequence are randomly sampled and masked by so-called sentinel tokens (a.k.a unique mask tokens). The target then corresponds to all of the dropped-out spans of tokens, delimited by the same sentinel tokens used in the input sequence plus a final sentinel token to mark the end of the target sequence. The model is trained to reconstruct the original sequence by predicting the dropped-out spans of tokens.

There are multiple versions of the T5 model published by its creators (Raffel et al., 2020). Table 2 shows the models that we used in our experiments. They mainly differ in the pre-training method and some details in architecture. The T5-v1 model was only pre-trained on C4 dataset [1] with an unsupervised denoising objective. The T5-lm model is based on T5-v1 but additionally was trained on 100k steps in the unsupervised LM objective. This adaptation improves the ability of the model to be used for prompt tuning. The T5-base was pretrained on a mixture of supervised and unsupervised tasks in multitasking fashion. It uses the denoising objective (replace spans) for unsupervised training. The t5-v1 and t5-lm models use GEGLU (Shazeer, 2020) activation in the feedforward hidden layer rather than RELU.

2.2. Data and Tasks

One of the main commonsense databases used in the related works is the ATOMIC knowledge base (Sap et al., 2019). It is an atlas of everyday commonsense reasoning, organized through 877k textual descriptions of inferential knowledge. The tuples are in the form of {`head h, relation r, tail t`} triplets. The head is the description of a situation involving social agents and the tails are social commonsense relating to them along 9 dimensions, such as the causes and the effects of the event, and the intentions, the character or possible reactions of the participants.

Each of these relations can be viewed as a task. In

[1] `http://www.tensorflow.org/datasets/catalog/c4`

Table 3. different Methods to format the input and output of a T5 model

Name	Format
map	**Input**: $\langle xIntent \rangle$ PersonX goes to a library **Target**: to borrow a book
natural	**Input**: personX goes to a library because he wants **Target**: to borrow a book
natural_x	**Input**: personX goes to a library because he wants **X** **Target**: **X** to borrow a book

our experiments, we report the results for the relation **xIntent** which shows the intention of the agent of doing something. The tuples can be organized in the form of typed if-then relations with variables. For example, "if PersonX goes to a library, he wants to borrow a book" shows the intention relation and for his or her character the tuple is "if PersonX goes to a library, he is seen as intelligent" where the heads and tails are underlined. Each head can have multiple tails for each relation. The tails are plausible or possible consequences of the head event.

ATOMIC splits into training, development, and test subsets such that no head entities in one set appear in any other. Models trained on this knowledge base can generate a plausible sequence by receiving a new situation and a specific relation from a defined set of 9 relations.

2.3. Training

To train the model, the head and the relation of each example serve as the inputs to the model and the tail is used as the target. The model is trained to minimize the negative log-likelihood of the tokens of the tail entity for each tuple. We use the AdaFactor optimizer with a constant learning rate of 0.001, a mini-batch size of 4, and train the model for 3 epochs. In few-shot settings, we set the number of examples per relation n=50 and n=100 examples per relation.

2.4. Evaluation

The generated tails for a given head and relation can be evaluated in terms of plausibility as well as variety and novelty with both automatic metrics such as ROUGE (Lin & Och, 2004) and BERT score (Yuan et al., 2021) and human verification. For the human evaluation, the workers were shown the complete tuple and asked whether it is valid or not. The evaluation was performed on 100 random generated samples. The automatic evaluations were also performed on 300 unique heads with multiple targets consisting of 1000 examples.

2.5. Fine-tuning methods

Table 3 shows different methods to format the input and the output of the model. They are based on the formats that were used as the supervised or unsupervised objective in the pre-training stage:

- **map:** This method is the same used in (Bosselut et al., 2020) in which the model is fine-tuned in a supervised fashion and a unique token is appended to the tokens of head entity for each relation. This token maps to a unique learnable embedding for that relation.
- **natural:** This method is similar to the above, but the input tuples are formatted into natural language prompts to represent the relations. We used the templates introduced in (Da et al., 2021b) for each relation.
- **natural_x:** This method is again similar to the **natural** methed in that the relations are mapped to natural language prompts. However, according to the T5 unsupervised training format, we use a unique mask token **X** as a placeholder for the tail. The input and the missing tail when replaced form a complete natural language sentence. It is similar to the unsupervised objective used in the pre-training stage of the T5 model.

2.6. Discussion

We evaluated the commonsense learning capabilities of different models. Table 2.5 shows the results for different methods described in the previous section using n = 50 and n = 100 examples per relation **xIntent**.

Findings Using $n = 50$ examples per relation, the **natural** method provides better results for all models compared to the **map** method which doesn't use natural prompts. This finding is in agreement with the related works (Da et al., 2021a) and shows that using natural language prompts for relation is effective in a few-shot setting.

However, the **natural_x** method produces better results compared to both the **natural** and **map** methods in the few-shot settings for t5-base. It shows that making prompts close to the denoising objective can benefit the models that were trained using this objective. This advantage can be even seen for the t5-v1 model which was only pre-trained using the denoising objective. However, the quality of the outputs are not as good as the other models but their diversity

Table 4. Results of different models using different fine-tuning methods for 50 and 100 samples

n = 50

method	model	**ROUGE Score**	**BertScore**	**Human**	**Unique Tails**
map	T5-v1	0.10	0.31	0.10	291
	T5-lm	0.43	0.54	0.62	160
	T5-base	0.39	0.50	0.58	198
natural	T5-v1	0.14	0.32	0.05	300
	T5-lm	0.45	0.54	0.64	155
	T5-base	0.41	0.52	0.67	182
natural_x	T5-v1	0.42	0.51	0.41	190
	T5-lm	0.15	0.37	0.11	193
	T5-base	**0.46**	**0.56**	**0.72**	125

n=100

method	model	**ROUGE Score**	**BertScore**	**Human**	**Unique Tails**
map	T5-v1	0.10	0.30	0.01	292
	T5-lm	0.46	0.54	065	112
	T5-base	0.43	0.52	058	145
natural	T5-v1	0.12	0.37	0.02	300
	T5-lm	0.47	0.54	0.68	108
	T5-base	0.42	0.52	0.62	118
natural_x	T5-v1	0.46	0.51	0.55	97
	T5-lm	0.30	0.43	0.41	254
	T5-base	0.46	0.54	0.68	93

(number of unique predicted tails) is high relative to its accuracy. This model also has very low performance with the **natural** method because it wasn't trained on the LM objective or supervised data like the other models. This shows that formatting the task into a format that was used in the pre-training stage can help to distill knowledge from the model in few-shot settings. To augment data for fine-tuning a model in this setup, specific words and longer phrases suitable for a task can be omitted from the sentences of a corpus. We leave this to future work.

In the case of t5-lm, the performance surprisingly drops after using the **natural_x** method. This can be due to the additional 100K pre-training steps on the LM objective which can cause the model to forget its pre-training capability on the denoising objective. The model may need to see more samples to recover its capability. When the number of samples increases to 100, the performance also increases, however, its quality based on human judgment is still not very good. This model has good performance particularly with the **natural** method and this method is again close to the LM objective.

The other major point is the performance of t5-base that increases with both natural and **natural_x** methods where this increase in the case of **natural_x** method is even higher. The t5-base model was actually pre-trained on a mixture of supervised and unsupervised objectives. This enables the model to have good performance in both formats without forgetting either one. The denoising objective is suitable for completing a sentence with natural compliments seen during the pre-training stage and the LM or supervised objectives are suitable to learn from examples and generalize them to unseen data. This consideration can be useful in pre-training or fine-tuning a model for distilling novel and accurate knowledge from pre-training models.

3. Conclusion

In this paper we compared different methods for pre-training and fine-tuning a T5 model to generate commonsense knowledge. We proposed that converting the training examples to a format similar to the unsupervised objective of the model can provide better results. The model provides better results when it is pre-trained on a mixture of denoising and LM objectives in a multi-tasking setup. The resulting model when fine-tuned with a similar method can produce more diverse and novel knowledge by relying more on the stored knowledge in the pre-training language model.

References

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., and Choi, Y. CoMET: Commonsense transformers for automatic knowledge graph construction. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 4762–4779, Florence, Italy, 2020. Association for Computational Linguistics. ISBN 9781950737482. doi: 10.18653/v1/p19-1470. URL https://www.aclweb.org/anthology/P19-1470.

Da, J., Bras, R. L., Lu, X., Choi, Y., and Bosselut, A. Understanding few-shot commonsense knowledge models. *arXiv preprint arXiv:2101.00297*, 2021a.

Da, J., Le Bras, R., Lu, X., Choi, Y., and Bosselut, A. Analyzing commonsense emergence in few-shot knowledge models. In *3rd Conference on Automated Knowledge Base Construction*, 2021b.

Feldman, J., Davison, J., and Rush, A. M. Commonsense knowledge mining from pretrained models. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 1173–1178, Hong Kong, China, 2020. Association for Computational Linguistics. ISBN 9781950737901. doi: 10.18653/v1/d19-1109. URL https://www.aclweb.org/anthology/D19-1109.

Hwang, J. D., Bhagavatula, C., Bras, R. L., Da, J., Sakaguchi, K., Bosselut, A., and Choi, Y. COMET-ATOMIC 2020: On Symbolic and Neural Commonsense Knowledge Graphs. 2020. URL http://arxiv.org/abs/2010.05953.

Lin, C.-Y. and Och, F. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir Workshop*, 2004.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. Language models as knowledge bases? In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 2463–2473, Hong Kong, China, 2020. Association for Computational Linguistics. ISBN 9781950737901. doi: 10.18653/v1/d19-1250. URL https://www.aclweb.org/anthology/D19-1250.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21: 1–67, 2020. ISSN 15337928.

Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A., and Choi, Y. ATOMIC: An atlas of machine commonsense for if-then reasoning. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 3027–3035, 2019. ISBN 9781577358091. doi: 10.1609/aaai.v33i01.33013027.

Shazeer, N. GLU variants improve transformer. *CoRR*, abs/2002.05202, 2020. URL https://arxiv.org/abs/2002.05202.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Yuan, W., Neubig, G., and Liu, P. BARTScore: Evaluating Generated Text as Text Generation. *preprint arxive 2106.11520*, 2021.