An Empirical Investigation of Commonsense Self-Supervision with Knowledge Graphs

Anonymous ACL submission

Abstract

Large knowledge graphs have been shown to benefit zero-shot evaluation of downstream tasks, through continual pre-training of language models. Yet, little is known about how to optimally learn from this knowledge, and what is the impact of the resulting models on different task partitions. This paper studies the effect of model architectures, loss functions, and knowledge subsets on the generalization of zero-shot models across task partitions. Our experiments show that data size, model size, model architecture, and loss function all play an important role in the accuracy and generalizability of the models. Most of the improvement occurs on questions with short answers and dissimilar answer candidates, which corresponds to the characteristics of the data used for pre-training. These findings inform future work that uses self-supervision with large knowledge graphs in order to create generalizable commonsense reasoning agents.

1 Introduction

Common sense is the common human knowledge about the world and the methods for making inferences from this knowledge (Davis, 2014): commonsense knowledge includes the basic facts about events (including actions) and their effects, facts about knowledge and how it is obtained, facts about beliefs and desires, as well as the basic facts about material objects and their properties (McCarthy, 1989). AI agents with common sense are expected to possess a wide range of everyday knowledge about naive physics, folk psychology, and causality. Rich commonsense knowledge can be found in public knowledge graphs (KGs), like ConceptNet (Speer et al., 2017), ATOMIC (Sap et al., 2019a), and Visual Genome (Krishna et al., 2017).

State-of-the-art commonsense reasoning systems are largely fueled by language models (LMs), as LMs are able to adapt to benchmarks effectively, insofar as training data is available (Ma et al., 2019). Recognizing that the assumption of always having benchmark-specific training data is unrealistic for open-domain reasoning, recent work has increasingly focused on zero- and few-shot tasks and reasoning models. Common methods for zero-shot reasoning rely on careful pre-training of LMs with external resources: commonsense KGs (Banerjee and Baral, 2020; Ma et al., 2021a), elicitation of pre-existing knowledge in the LM (Shwartz et al., 2020; Paranjape et al., 2021), or instruction-prompted training with a diverse set of tasks (Sanh et al., 2021). While pre-training with commonsense knowledge has been shown to improve model performance (Mitra et al., 2019; Ma et al., 2021a), prior work has not investigated how different architectural and data decisions affect model accuracy and generalization across tasks.

This paper studies the *effect of model architectures, loss functions, and knowledge subsets on the accuracy and generalization of language models, across commonsense tasks*. We measure *generalization* as the average model performance on a set of out-of-domain multiple-choice question answering benchmarks. We consider two LM architectures and two representative loss functions. We study the interplay of the model size with the pre-training knowledge size, and note that the optimal knowledge size is highly dependent on the model size, architecture, and loss function. Larger LMs and loss functions that score the answer candidates jointly tend to generalize better to out-of-domain datasets. Further analysis shows that vanilla LMs perform better on questions which are longer and have very similar answers, while pre-training with knowledge is able to close the gap for questions whose answer candidates are very different.

2 Problem Setup

Task formulation. Following Ma et al. (2021a), we formalize *generalizable commonsense reasoning* as the task of performing question answering

(QA) across out-of-domain commonsense tasks. We use the recently-introduced CommonSense Knowledge Graph (CSKG) (Ilievski et al., 2021b) to sample thousands of commonsense statements, and transform them into multiple-choice questions. Each question corresponds to a particular knowledge dimension (Ilievski et al., 2021a). We define *domain* as the dimensions of common sense necessary for solving a particular set of tasks.[1] Given a natural language question Q, and n possible answers $\{A_1, ..., A_n\}$, the LM will be asked to select the most probable single answer A during training. Once the LM pre-training is done, the updated LM is applied across QA tasks in a zero-shot manner.

Evaluation. We evaluate on five benchmarks for multiple-choice commonsense question answering. Two datasets have been known to have domain overlap with existing KGs (Mitra et al., 2019; Ma et al., 2021a): 1) *CommonsenseQA (CSQA)* (Talmor et al., 2019), which evaluates a broad range of common sense aspects, has been devised based on knowledge in ConceptNet; 2) *SocialIQA (SIQA)* (Sap et al., 2019b), which requires reasoning about social interactions, has been created based on the ATOMIC KG (Sap et al., 2019a). We refer to CSQA and SIQA as **in-domain (ID) datasets**. We also evaluate on three **out-of-domain (OOD) datasets:** *1. Abductive NLI (aNLI)* (Bhagavatula et al., 2019), a natural language inference task, where, given the beginning and the ending of a story, the task is to choose the more plausible hypothesis out of two options; *2. PhysicalIQA (PIQA)* (Bisk et al., 2020), which tests physical reasoning; and *3. WinoGrande (WG)* (Sakaguchi et al., 2019), an anaphora resolution task. We measure LM's *accuracy* on a benchmark as the ratio between the correctly-answered questions and the total number of questions in a benchmark.

3 Method

Language Models. We adopt two widely-used pre-trained models: RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2019). RoBERTa is an encoder-only masked language model (MLM), whereas T5 is an encoder-decoder model which converts tasks into text-to-text format. We use RoBERTa's large model, which has 355M parameters. We experiment with three T5 models of different sizes: small (60M parameters), large (740M), and 3b (2.85B).

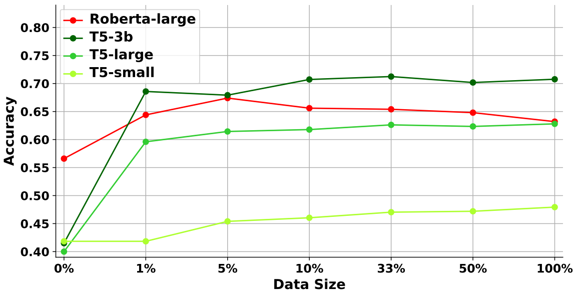

Figure 1: Evaluation results of 4 models with different data sizes. Each point represents the average performance of a model over the five datasets. We show results for RoBERta with J loss, and T5 with I loss.

Loss functions. Following Ma et al. (2021a), for RoBERTa the input sequences are concatenations of the question and each of its answer candidates. We mask one non-stop token in the sequence at a time, and compute the masked token's loss. We then take the averaged loss for the sequence and train the model with margin loss:

$$L = \frac{1}{m} \sum_{\substack{i=1 \\ i \neq y}}^{m} max(0, \eta - S_y + S_i)$$

where S_y and S_i are the negative averaged loss for correct answer and distractor respectively. During inference, we take the candidate with highest score S as the answer.

For T5, we add a task-specific prefix, "reasoning:", to the input sequence following how Raffel et al. (2019) adapt it to downstream task. The model is pre-trained to predict either true or false, for each candidate, separately.[2] During inference, we concatenate the benchmark question with one candidate answer at a time, and we compute $d = p(true) - p(false)$ for that candidate based on our model. The candidate with the highest difference d is chosen as the model answer.

Notably, the loss for RoBERTa is computed over all candidates jointly (`J`), whereas the loss for T5 is for each candidate independently (`I`). To make the two models more comparable, we also: 1) pre-train RoBERTa models with `I` loss, by appending a true-false label to the input sequence and computing the loss only for this masked label; and 2) pre-train T5 with a joint function by computing the difference

[1] See (Ilievski et al., 2021a; Ma et al., 2019) for more details about the relation between dimensions and tasks.

[2] We also tried to concatenate the question with all answer candidates, and teach the model to predict the position or make a copy of the right candidate, following (Khashabi et al., 2020). These loss strategies performed consistently worse, and we leave them out of the paper.

Table 1: Evaluation results on 5 benchmarks of 4 models with their optimal data size. Best results are in bold.

Model	Loss	Data Size	OOD			ID		Avg(OOD)	Avg(ID)	Avg
			aNLI	WG	PIQA	SIQA	CSQA			
Roberta-large	(Ma et al., 2021a)		70.5	60.9	72.4	63.2	67.4	67.9	65.3	66.8
	I	5%	68.1	60.1	67.7	60.8	62.1	65.3	61.5	63.8
	J	5%	72.0	60.2	72.5	**65.4**	66.9	68.2	66.2	67.4
T5-small	I	33%	51.4	51.3	56.3	41.9	34.3	53.0	38.1	47.0
	J	33%	50.6	52.2	56.0	42.5	36.9	52.9	39.7	47.6
T5-large	I	33%	64.6	58.4	70.2	57.2	62.7	64.4	60.0	62.6
	J	33%	65.5	59.0	70.6	57.2	62.9	65.0	60.0	63.0
T5-3b	I	33%	75.1	70.2	76.6	63.9	**70.4**	74.0	67.2	71.2
	J	33%	**76.6**	**71.0**	**76.7**	65.3	69.9	**74.7**	**67.6**	**71.9**

between the true and false labels, for each candidate, followed by the same margin function used for RoBERTa's J loss function. More details about the model training can be found in appendix A.

Knowledge sampling. We use the subset of CSKG which combines ATOMIC, ConceptNet, WordNet (Miller, 1995), Wikidata (Vrandečić and Krötzsch, 2014), and Visual Genome. Unlike Ma et al. (2021a), who use 14 semantic relations, we use the entire set of relations in CSKG, and randomly sample subsets of 1, 5, 10, 33, 50, and 100%. We also explored sampling strategies based on training indicators and knowledge dimensions, but these consistently performed worse than random sampling (see appendix B for more information).

4 Results

How much data is needed to pre-train the models? Figure 1 shows the average accuracy for the four models (Roberta-Large, T5-small, T5-large, and T5-3b) when trained with different data sizes. We observe that models have different optima in terms of the data size that they are pre-trained with. RoBERTa-Large performs best with only 5% of the artificial data, reaching an average score of 67.4% across the five datasets. Meanwhile, the best T5 model, T5-3b peaked with 33% of the data, which shows that it benefits from more data for pretraining. Both T5-large and T5-small achieve higher averaged accuracy with increased data size, however the gains plateaus at about 33% of the data. Thus, we use 5% of data for RoBERTa and 33% for T5 in our later experiments. We also provide the learning curves for the four models in appendix C.

Which model generalizes best overall? The accuracies of the four models with I and J losses are shown in table 1. Overall, T5-3b with joint loss achieves the best performance. Its average accuracy outperforms the best RoBERTa model by about 4 points on average, as well as the previous top scoring model (Ma et al., 2021a) by 5 points, setting a new SotA zero-shot accuracy. The other T5 models perform worse than RoBERTa: even though T5-large has 2x more parameters than RoBERTa, its performance is 5% lower on average.

What causes the difference in model performance? The models differ in three aspects: model size, architecture, and loss function. The obtained results for T5 reveal a clear positive impact of the model size, as T5-3b > T5-large > T5-small. Yet, the superiority of RoBERTa-large over T5-large reveals that the model architecture and loss function also play an important role. The choice of the loss function has much higher impact for RoBERTa, yielding 4 points difference between the J and the I loss. For the T5 models, the impact of the loss function is minimal. This could be because RoBERTa with J loss setup has masked token prediction for multiple tokens in the sequence, which may increase its prediction power.

How do models perform on out-of-domain benchmarks? The results show that the average improvement of T5-3b is mostly due to its improved performance on out-of-domain benchmarks. T5-3b's improvement over RoBERTa is on average 6.5% on the OOD benchmarks, but only 1.4% on the ID benchmarks. This generalization ability of T5-3b can largely be attributed to the larger capacity of the T5-3b model, which allows it to represent additional knowledge and associations between terms.

What is the relation between the gain in generalization and the properties of the task? To better understand the gains from pretraining, we breakdown the task performance by different properties. We select PIQA for this analysis as its answers are diverse in many aspects. Specifically, we measure the accuracy of the models in each data quartile based on answer similarity (Jaccard similarity measure between the answer candidates' tokens), answer length, and vocabulary overlap (w.r.t pre-training

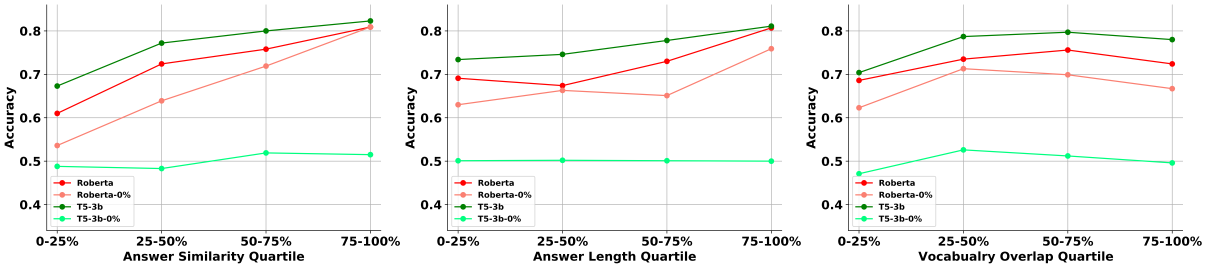

Figure 2: Accuracy of the best performing RoBERTa-large and T5-3b models in relation to the answer similarity, answer length, and vocabulary overlap between the data used for pretraining and testing.

data) in figure 2. We use RoBERTa's tokenizer.[3] We see that both models perform better on questions with similar answers. Interestingly, vanilla RoBERTa already achieves high performance on this set, and pre-training only improves the performance on the questions with rather different answers. Given that the data used for pre-training is designed to only include questions with non-overlapping answers, this finding is intuitive and explains where the improvement of performance with pre-training comes in (Ma et al., 2021a). T5-3b's accuracy gain over RoBERTa-large also owes to this data subset, which again shows that larger models have more capacity to learn from the pre-training knowledge. In addition, both models perform best on questions with longer answers, while T5-3b is advantageous for short answers. Notably, the data used for pre-training mostly consists of short answers, showing again that the performance gain of T5-3b owes to its capacity to extend original knowledge during the additional pre-training. While we expect that vocabulary overlap is proportional to accuracy, figure 2 (right) does not show a clear correlation between overlap and accuracy. Further analysis is reported in appendix D.

5 Related Work

Zero-shot Commonsense Reasoning methods often elicit knowledge from pre-trained LMs, by using self-talk clarification prompts (Shwartz et al., 2020) or asking LMs to generate contrastive explanations (Paranjape et al., 2021). Models can be taught to answer questions by adapting an external dataset (Abdou et al., 2020). To use KGs for zero-shot pretraining and evaluation, Banerjee and Baral (2020) pre-train a LM to perform knowledge completion, Bosselut et al. (2020) enhance the question based on knowledge completion models, whereas Ma et al. (2021a) generate synthetic QA pairs from a consolidated KG to pre-train LMs. Our work complements that by (Ma et al., 2021a), because we investigate the impact of model architecture, size, and training setup on different task partitions.

Model Generalization and Data Selection. Sen and Saffari (2020) analyzed LM's ability to generalize across 5 different QA datasets. Ma et al. (2021b) showed that models can have drastically different performances by fine-tuning on different subset of the data. Swayamdipta et al. (2020) proposed to select training instances based on models' confidence and variability, and they show that training on less-confident examples is more beneficial for generalization. While prior work analyses model robustness by sub-sampling instances from the task's training set, we investigate the impact of data sizes, model architectures, and loss functions when models are pre-trained on large KGs. Our work complements that of Ilievski et al. (2021a), which splits synthetic data from KGs into 12 commonsense dimensions, revealing that some kinds of knowledge are much more useful for pre-training compared to others, without measuring the impact of pre-training decisions.

6 Conclusions

This paper studied the impact of strategies for self-supervision of LMs over KGs, differing in terms of their model architecture, loss function, data size, and model size. We noted that optimal strategies depend on all these factors: larger models generally perform better, and the optimal training loss differs per model. Most of the improvement of the largest generative model comes from questions with short answers and dissimilar answer candidates, which is expected, given that the pre-training data has these properties. These findings inform future work that uses self-supervision with large KGs to create generalizable commonsense reasoning agents.

[3] We observed similar results with NLTK's tokenizer.

References

Mostafa Abdou, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott, and Anders Søgaard. 2020. The sensitivity of language models and humans to Winograd schema perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7590–7604, Online. Association for Computational Linguistics.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. *ArXiv*, abs/2005.00316.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 7432–7439.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2020. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering.

Ernest Davis. 2014. *Representations of commonsense knowledge*. Morgan Kaufmann.

Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L McGuinness, and Pedro Szekely. 2021a. Dimensions of commonsense knowledge. *arXiv preprint arXiv:2101.04640*.

Filip Ilievski, Pedro Szekely, and Bin Zhang. 2021b. Cskg: The commonsense knowledge graph. In *Extended Semantic Web Conference (ESWC)*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 22–32.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021a. Knowledge-driven Data Construction for Zero-shot Evaluation in Commonsense Question Answering. In *35th AAAI Conference on Artificial Intelligence*.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Satoru Ozaki, Eric Nyberg, and Alessandro Oltramari. 2021b. Exploring strategies for generalizable commonsense reasoning with pre-trained models. *EMNLP 2021*.

John McCarthy. 1989. Artificial intelligence, logic and formalizing common sense. In *Philosophical logic and artificial intelligence*, pages 161–190. Springer.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *arXiv preprint arXiv:1909.08855*.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Prompting contrastive explanations for commonsense reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4179–4192, Online. Association for Computational Linguistics.

Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. *arXiv preprint arXiv:2001.10528*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *Proc. of AAAI*, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQa: Commonsense reasoning about social interactions. In *Proc. of EMNLP-IJCNLP*, pages 4463–4473.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online. Association for Computational Linguistics.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. *ArXiv*, abs/2004.05483.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proc. of AAAI*, AAAI'17, page 4444–4451.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proc. of NAACL*, pages 4149–4158.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

A Training details

Among all the training sets, we are using learning rate of $1e^{-5}$, batch size of 32, training epochs of 5, adam-epsilon of $1e^{-6}$, $\beta 1 = 0.9, \beta 2 = 0.98$, warm-up proportion of 0.05.

For T5 training, we add the prefix "reasoning:" in front of every concatenation of question and answer, then ask the model to predict "1" for true, and "2" for false.

Regarding libraries, we used python 3.7.10, pytorch 1.9.0 and transformers 4.11.3.

For CPUs, we used Intel(R) Xeon(R) Gold 5217 CPU @ 3.00GHz (32 CPUs, 8 cores per sockets, 263GB ram).

For GPUs, we used Nvidia Quadro RTX 8000, and Nvidia Geforce 2080Ti.

B Sampling strategies

We experimented with the following selection strategies:

1. *random* - draw X% of the data points by chance, without replacement;

2. *high/low vanilla-conf* - select the X% of the points for which the model has the highest/lowest confidence before pretraining;

3. *high/low confidence* - select the X% of the points for which the model has the highest/lowest mean confidence for the true label across the pretraining epochs (Swayamdipta et al., 2020);

4. *high/low variability* - select the X% of the points for which the model has the highest/lowest standard variation for the true label across the pretraining epochs (Swayamdipta et al., 2020);

5. *high/low margin* - select the X% with the highest/lowest mean relative confidence of the correct answer to the incorrect ones (Pleiss et al., 2020);

6. *dimension-based* Select a random X% of the training points that belong to a knowledge dimension. We consider the 12 dimensions defined in (Ilievski et al., 2021a): Lexical, Similarity, Distinctness, Taxonomic, Part-whole, Spatial, Creation, Utility, Desire, Quality, Temporal, Relational-other.

7. *vocab-novelty* We draw the X% of the questions with the highest/lowest average vocabulary novelty of its tokens;

The results for RoBERTa (table 2) and T5-large (table 3) show that all of the sampling strategies perform worse than random sampling, which is consistent with the initial finding in (Ma et al., 2021a). The performance of sampling with the dimensions *quality* and *temporal* comes close to random sampling. Among the training indicators, the best sampling strategy for RoBERTa is using examples with low confidence or margin, while for T5-large, it is best to rely on examples with low vanilla confidence.

Table 2: Evaluation results on 5 benchmarks of **RoBERTa-Large** with different sampling strategies. All samples have equivalent sizes, corresponding to 5% of the training data. The best result per column is marked in bold.

Strategy		OOD			ID		Avg(OOD)	Avg(ID)	Avg
		aNLI	WG	PIQA	SIQA	CSQA			
Random	5%	72.0	60.2	**72.5**	**65.4**	**66.9**	68.2	**66.2**	**67.4**
Dimension	temporal	**72.7**	61.1	72.1	62.3	65.8	**68.6**	64.1	66.8
	desire	70.2	59.5	72.4	60.9	64.3	67.4	62.6	65.5
	taxonomic	67.0	58.0	69.2	0.51	59.0	64.7	55.0	60.8
	quality	71.3	**61.8**	72.0	58.5	64.6	68.4	61.6	65.6
	rel-other	65.3	55.5	69.7	51.5	58.1	63.5	54.8	60.0
Vanilla-conf	high	63.3	59.1	67.6	49.4	47.2	63.3	48.3	57.3
	low	57.9	51.9	55.6	33.1	21.7	55.1	27.4	44.0
Conf	high	66.2	58.9	70.3	59.4	62.2	65.1	60.8	63.4
	low	71.4	59.2	72.1	62.6	65.7	67.6	64.2	66.2
Varibility	high	67.4	56.8	65.5	48.2	44.0	63.2	46.1	56.4
	low	65.4	56.0	68.6	54.4	61.0	63.3	57.7	61.1
Margin	high	67.1	58.2	70.7	60.1	62.3	65.3	61.2	63.7
	low	72.3	60.5	71.2	62.7	65.0	68.0	63.9	66.3

Table 3: Evaluation results on 5 benchmarks of **T5-large** with different sampling strategies. All samples have equivalent sizes, corresponding to 5% of the training data. The best result per column is marked in bold.

Strategy		OOD			ID		Avg(OOD)	Avg(ID)	Avg
		aNLI	WG	PIQA	SIQA	CSQA			
Random	5%	64.0	**57.5**	**69.7**	54.3	**61.7**	**63.7**	**58.0**	**61.4**
Dimension	temporal	**65.1**	56.5	68.9	54.1	59.9	63.5	57.0	60.9
	desire	64.4	55.9	69.3	56.4	57.4	63.2	56.9	60.7
	taxonomic	60.8	54.6	67.9	54.1	52.3	61.1	53.2	57.9
	quality	65.0	56.0	69.1	**56.1**	56.0	63.4	56.1	60.4
	rel-other	54.6	51.3	60.1	44.0	39.5	55.3	41.8	49.9
Vanilla-conf	high	63.7	56.0	68.0	55.0	55.8	62.6	55.4	59.7
	low	64.8	55.5	69.4	53.4	59.0	63.2	56.2	60.4
Conf	high	62.7	56.2	67.8	53.1	55.4	62.2	54.3	59.0
	low	45.8	48.1	39.8	24.4	09.7	44.6	17.1	33.6
Varibility	high	57.3	51.6	58.8	39.7	36.9	55.9	38.3	48.9
	low	63.5	55.2	68.7	53.4	54.7	62.5	54.1	59.1
Margin	high	62.9	55.5	68.8	54.4	55.5	62.4	55.0	59.4
	low	47.8	51.5	42.5	26.2	12.5	47.3	19.4	36.1

C Training curves

The curve in figure 3 shows that for the RoBERTa, the initial loss is small, probably because we are using the same training loss as it was pre-trained, which can also explain why the vanilla RoBERTa model performs well. For T5, the initial loss is about 30, which is left out of the figure in order to preserve an informative range of $[0-2]$. This high initial loss is expected, given that we are using a novel prefix, for which T5 has not been pre-trained. The difference in the training curves is also due to the different training loss in Roberta and T5 (margin loss and Cross Entropy loss).

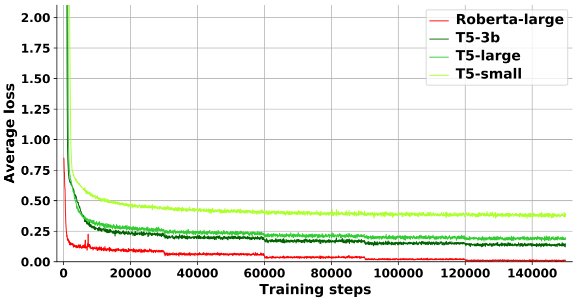

Figure 3: Training curves of the models: RoBERTa-large, T5-small, T5-large, and T5-3b. We use the RoBERTa-large model with J loss and 5% of the data. We show the T5 models with I loss and 33% of the data.

D Data size analysis

Table 4 shows the impact of different data sizes on the model performance on different quartiles of answer similarity, length, and vocabulary overlap. We see that both models perform better on the questions with dissimilar answers when they are trained with more data. At the same time, the models perform optimal on the questions with similar answers with less data. This confirms our explanation that the knowledge used for pre-training directs the models towards better performance on

Table 4: Evaluation results on the similarity, length, and vocabulary overlap quartiles of PIQA data for the models RoBERTa (with J loss) and T5-3b (with I loss) with different data sizes. Best results per model and similarity quartile are marked in bold.

Model	**Data Size**	**Similarity**				**Length**				**Vocabulary overlap**			
		25%	**50%**	**75%**	**100%**	**25%**	**50%**	**75%**	**100%**	**25%**	**50%**	**75%**	**100%**
Roberta	0%	53.6	63.9	71.9	**80.9**	63.0	66.3	65.1	75.9	62.3	71.3	69.9	66.7
	1%	56.6	**73.5**	75.6	78.7	**68.8**	**69.6**	68.0	78.0	**68.6**	71.3	74.3	70.2
	5%	**60.3**	72.4	**76.5**	80.4	66.7	68.3	**72.1**	**82.6**	**68.6**	**73.5**	**74.9**	**72.6**
	10%	58.2	71.1	73.2	79.8	67.1	65.9	70.2	79.1	68.0	72.4	70.6	71.3
	33%	58.8	72.0	74.7	78.0	68.0	68.7	70.6	76.3	66.9	71.7	72.8	72.2
	50%	57.1	70.4	73.9	80.2	66.4	66.1	71.2	77.8	65.8	70.0	**74.9**	70.9
	100%	55.6	68.3	69.3	74.8	62.7	65.2	66.9	73.0	63.4	65.9	71.9	66.7
T5-3b	0%	48.8	48.3	51.9	51.5	50.1	50.2	50.1	50.0	47.1	52.6	51.2	49.6
	1%	66.4	71.1	75.6	79.3	69.3	71.7	73.6	77.8	66.7	73.5	77.1	75.2
	5%	67.3	73.9	75.8	80.4	71.9	75.7	70.4	79.6	70.2	74.3	76.9	76.1
	10%	67.1	77.6	77.6	**83.3**	73.4	75.2	73.6	**83.3**	**73.2**	75.4	79.7	77.2
	33%	**69.5**	**78.7**	77.3	80.9	73.9	**75.9**	**76.0**	80.7	70.8	76.7	**80.0**	**78.9**
	50%	67.3	77.0	**77.8**	80.7	73.4	74.8	74.7	79.8	70.4	76.5	78.4	77.4
	100%	**69.5**	76.7	74.7	79.8	**74.5**	72.0	75.4	78.9	71.0	**77.2**	75.8	76.7

the questions with dissimilar answers.

In terms of answer length, we see that T5 is able to exploit maximum amount of data for short answers, which is expected, given that most of the pre-training questions are relatively short. When it comes to longer answers, T5 performs best with less data, which indicates that the pre-training data has limited utility for this set of questions. Curiously, this pattern is not observed for RoBERTa - RoBERTa is unable to leverage more than 1% of the data to improve its performance on the questions with short answers. We hypothesize that this is due to the limited model capacity of RoBERTa, causing limited ability to store additional knowledge about the data.

Again, in this table, we do not observe clear patterns between the model accuracy and vocabulary overlap across the different data sizes.