BLOG ›

Introducing FLAN: More generalizable Language Models with Instruction Fine-Tuning

WEDNESDAY, OCTOBER 06, 2021

Posted by Maarten Bosma, Research Engineer and Jason Wei, AI Resident, Google Research

For a machine learning model to generate meaningful text, it must have a large amount of knowledge about the world as well as the ability to abstract. While language models that are trained to do this are increasingly able to automatically acquire this knowledge as they scale, how to best unlock this knowledge and apply it to specific real-world tasks is not clear.

One well-established technique for doing this is called fine-tuning, which is training a pretrained model such as BERT and T5 on a labeled dataset to adapt it to a downstream task. However, fine-tuning requires a large number of training examples, along with stored model weights for each downstream task, which is not always practical, particularly for large models.

In "Fine-tuned Language Models Are Zero-Shot Learners", we explore a simple technique called instruction fine-tuning, or *instruction tuning* for short. This involves fine-tuning a model not to solve a specific task, but to make it more amenable to solving NLP tasks in general. We use instruction tuning to train a model, which we call Fine-tuned LAnguage Net (FLAN). Because the instruction tuning phase of FLAN only takes a small number of updates compared to the large amount of computation involved in pre-training the model, it's the metaphorical dessert to the main course of pretraining. This enables FLAN to perform various unseen tasks.

An illustration of how FLAN works: The model is fine-tuned on disparate sets of instructions and generalizes to unseen instructions. As more types of tasks are added to the fine-tuning data model performance improves.

Background

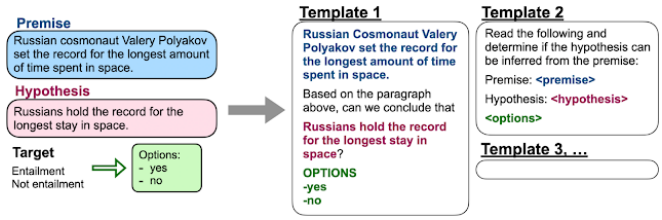
One recent popular technique for using language models to solve tasks is called zero-shot or few-shot prompting. This technique formulates a task based on text that a language model might have seen during training, where then the language model generates the answer by completing the text. For instance, to classify the sentiment of a movie review, a language model might be given the sentence, "*The movie review 'best RomCom since Pretty Woman' is _*" and be asked to complete the sentence with either the word "*positive*" or "*negative*".

Although this technique demonstrates good performance for some tasks, it requires careful prompt engineering to design tasks to look like data that the model has seen during training — an approach that performs well on some but not all tasks and also can be an unintuitive way for practitioners to interact with the model. For example, the creators of GPT-3 (one of the largest language models in use today) found that such prompting techniques did not result in good performance on natural language inference (NLI) tasks

Instruction Tuning

FLAN instead fine-tunes the model on a large set of varied instructions that use a simple and intuitive description of the task, such as "Classify this movie review as positive or negative," or "Translate this sentence to Danish."

Creating a dataset of instructions from scratch to fine-tune the model would take a considerable amount of resources. Therefore, we instead make use of templates to transform existing datasets into an instructional format.



Example templates for a natural language inference dataset.

We show that by training a model on these instructions it not only becomes good at solving the kinds of instructions it has seen during training but becomes good at following instructions in general.

Search blog by keyword or author

Archive +

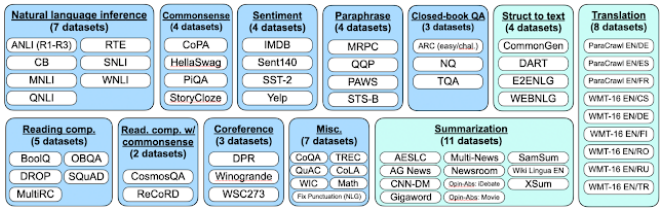
Labels +

Evaluating the Model

To compare FLAN against other techniques in a meaningful way, we used established benchmark datasets to compare the performance of our model with existing models. Also, we evaluated how FLAN *performs* without having seen any examples from that dataset during *training*.

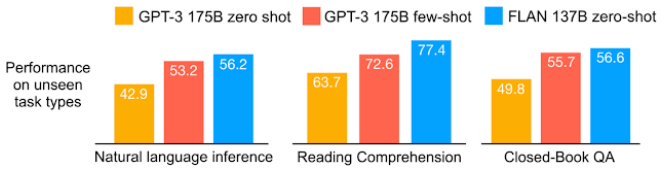
However, if we trained on datasets that were too similar to an evaluation dataset, that might still skew the performance results. For example, training on one question-answering dataset might help the model do better on another question-answering dataset. Because of this, we group all datasets into clusters by type of task and hold out not just the training data for the dataset, but the *entire task cluster* to which the dataset belongs.

We grouped our datasets into the clusters below.



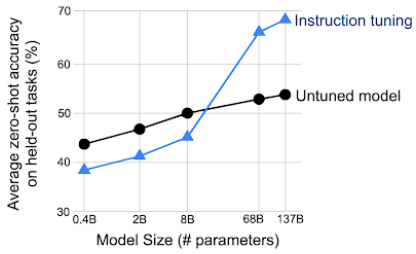
Results

We evaluated FLAN on 25 tasks and found that it improves over zero-shot prompting on all but four of them. We found that our results are better than zero-shot GPT-3 on 20 of 25 tasks, and better than even few-shot GPT-3 on some tasks.



*For various models, we show the average accuracy over all datasets in a task cluster. Natural language inference datasets: **ANLI R1–R3**, **CB**, and **RTE**. Reading comprehension datasets: **BoolQ**, **MultiRC**, **OpenbookQA**. Closed-book QA datasets: **ARC**, **NQ**, **TriviaQA**.*

We also find that model scale is very important for the ability of the model to benefit from instruction tuning. At smaller scales, the FLAN technique actually degrades performance, and only at larger scales does the model become able to generalize from instructions in the training data to unseen tasks. This might be because models that are too small do not have enough parameters to perform a large number of tasks.



Instruction tuning only improves performance on unseen tasks for models of certain size.

Conclusion

The FLAN model is not the first to train on a set of instructions, but to our knowledge we are the first to apply this technique at scale and show that it can improve the generalization ability of the model. We hope that the method we presented will help inspire more research into models that can perform unseen tasks and learn from very little data.

We also released the code to perform the transformations so that other researchers can reproduce our results and build on them.

Acknowledgements

We thank our collaborators Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le at Google Research.