Exploring the Effect of Semantic Similarity on Model Generalization

Hong Ju Jeon [1] Dustin Zubke [1]

[1]Computer Science, Stanford

Abstract

Research in instruction fine tuning (IFT) has shown significant potential to improve generalized model performance. To further the understanding of IFT, our work studies the relationship between semantic similarity of tasks used in finetuning and model performance. We first develop an approach to calculating a measure of semantic similarity between tasks and then utilize the Sup-NatInst dataset[2] to construct training datasets by varying the amount of similarity. We find that performance improves for both 1) increasing the number of tasks in training and also 2) decreasing the semantic similarity of the training tasks. We use our findings to train T-Diff-Instruct-300, which shows a ROUGE-L score of 26.6, which is within 11.3% of our best model while using 2.5x less training data. We hope that this work can assist in building language models for low-resource tasks or languages by allowing researchers to identify potential tasks that will more effectively boost model generalization.

Introduction

Background: Research in instruction fine tuning (IFT) has shown significant potential to improve generalized model performance on unseen tasks (i.e. zero-shot) by describing NLP tasks using natural language[1, 3]. However, having a large amount of data is often necessary to perform well at a task, but not all tasks or languages have abundant data.

Our work: We study the relationship between **semantic similarity of tasks in a group and the zero-shot performance** of models finetuned on various task groupings based on similarity. We show how this concept of similarity can be used to improve model generalization capabilities in low-resource environments.

Dataset

The Sup-NatInst dataset is made up of 76 task categories, 1616 tasks, and 5 million task instances. Each task instance is an example of an "instruction" to be used in IFT, which can be thought of as text descriptions written in natural language that you would give to a human to solve the corresponding task. Figure1 shows a task example.

Task Type	Cause Effect Classification
Task ID	task828_copa_cause_effect_classification
Definition	In this task your given two statements. You must judge whether the second sentence is the cause or effect of the first one. Label the instances as "cause" or "effect" based on your judgment. The sentences are separated by a newline character.
Positive Example	**Input**: The women met for coffee. They wanted to catch up with each other. **Output**: cause **Explanation**: The women met for coffee because they wanted to catch up with each other.
Negative Example	**Input**: My body cast a shadow over the grass. The sun was rising. **Output**: effect **Explanation**: The rising of the sun isn't an effect of casting a shadow over the grass.
Instance	**Input**: The woman tolerated her friend's difficult behavior. The woman knew her friend was going through a hard time. **Valid Output**: ["cause"]

Figure 1. An example of one of the tasks, which includes a task type, definition, positive and negative examples, and instance.

The task instance corresponds to a single training example, and the distribution of task instances across tasks and categories in the dataset is not uniform (Figure 2).

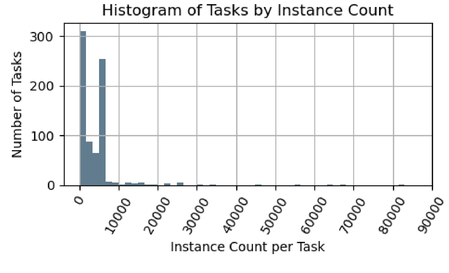

Figure 2. A histogram showing the number of tasks with a given instances-per-task count.

Approach

1. We develop an approach to **calculating a measure of the semantic similarity between tasks and categories** in the Sup-NatInst dataset[2],
2. Create a few training datasets that contain task in groups of **varying levels of semantic similarity**, finetune a T5 model on each training set, and evaluate the models on the Sup-NatInst test set to measure the effects of semantic similarity on unseen tasks using the ROUGE-L score.
3. We take the best performing model that is trained on a subset of the training data and compare against the model trained on the full dataset.

Experiments

Data: We set up our experiments to vary the amount of semantic similarity among tasks in the training dataset. We take the following steps:

1. **Balance the dataset** by filtering tasks with less than a threshold number of instances and only use a fixed number of instances per task in the experiment.
2. **Create embeddings** for each task from the definition, category, and positive & negative examples.
3. **Calculate the cosine similarity** between each pair of embedding creating a complete graph with similarity edge weights.
4. **Construct the training dataset** by selecting the tasks that are either the most semantically similar or different.

Experiments:

- **TDiff-Instruct** - T5 finetuned on **most semantically different** tasks.
- **TSim-Instruct** - T5 finetuned on **most semantically similar** tasks.
- **Random** - Randomly selected set of tasks with uniform probability.
- TSim-Category-Instruct - T5 finetuned on **most semantically similar** categories.
- TDiff-Category-Instruct - T5 finetuned on **most semantically different** categories.

Baselines:

- **T5** - 60M-parameter T5 with no finetuning.
- **Full dataset** - 60M-parameter T5 trained on all 767 task types.
- Tk-Instruct - 60M-parameter T5 trained on all tasks presented in Wang et. al., 2022[2].

Results

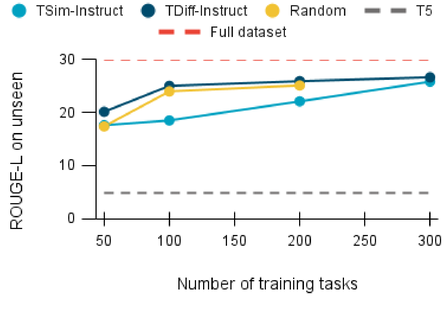

Figure 3. The model's ROUGE-L performance to unseen tasks as a function of the number of task types used in training. We observe that 1) increasing the number of training tasks and 2) increasing the task diversity generally leads to better performance for every number of task types that we varied.

Analysis

Performance increases with more training tasks. Figure 3 shows that performance for every model increases with more training tasks, which is consistent with the results of Wang et al. 2022[2].

Performance increases with lower semantic similarity. Comparing the performance of the TDiff-Instruct, Random, and TSim-Instruct models in Figure 3 shows that the models with lower semantic similarity scores in Figure 4 have higher performance. The "Category groupings" in Table 1 also show this result of lower similarities in the training set improving the ROUGE-L score. However, the performance gains were more limited in the categories experiments due to some challenges in grouping the categories.

More training tasks has a stronger impact on performance than lower semantic similarity. Since the TDiff-Instruct models on larger datasets have better performance but higher semantic similarity, the positive effect of increasing the number of task instances is overwhelming the negative effects of increasing the semantic similarity.

TDiff-Instruct-300 shows a competitive ROUGE-L score within 11.3% of our best model (26.6 vs 30.0) while using $\sim 2.5x$ less training data, suggesting room for further research on the efficient use of data during instruction finetuning.

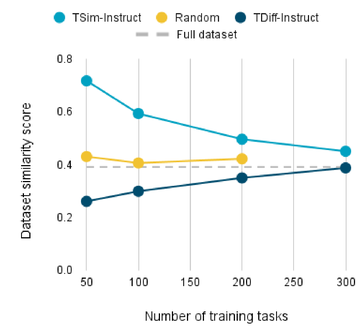

Figure 4. The cosine similarity of the dataset grouping is plotted as a function of the number of tasks. As the datasets get larger, their semantic similarity trends toward the average semantic of the full training dataset.

Models	ROUGE-L score	Number of tasks seen
Baselines		
T5-small	4.9	0
Random (ours)	25.1	200
Full dataset (ours)	**30.0**	757
Tk-Instruct [2]	40.1*	757
Category grouping		
TSim-Category-Instruct	21.0	364
TDiff-Category-Instruct	**22.0**	233
Task grouping		
TSim-Instruct-200	22.1	200
TSim-Instruct-300	25.8	300
TDiff-Instruct-200	25.9	200
TDiff-Instruct-300	**26.6**	300

Table 1. Overall performance of models by finetuning on different groupings of tasks. We show that T-Diff-Instruct outperformed Random, and all T-Sim-Instruct variants, even while using an equal or smaller number of training tasks. *We note that results for Tk-instruct are not directly comparable to our experimental setup.

Conclusions

- Decreasing the semantic similarity of tasks is an alternative to increasing the number of task types to increase performance.
- As the number of tasks increases, model performance will improve. However, the rate of improvement is based on the semantic similarity of additional data.
- The impact of semantic similarity holds for not only task types but also categories, though more research is needed to effectively assess category semantics.
- We hope our results could provide some guidance on how researchers could use semantic similarity to efficiently collect future datasets to address low-resource tasks and languages.

References

[1] Long Ouyang, Jeff Wu, Xu Jiang, and Diogo Almeida et al.
Training language models to follow instructions with human feedback, 2022.

[2] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, and et al.
Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022.

[3] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le.
Finetuned language models are zero-shot learners, 2021.