EFFICACY OF MACHINE-GENERATED ANNOTATIONS

Samaksh Gulati, Anshit Verma, Manoj Parmar, Palash Chaudhary *
{sgulati, averma373, mparmar, palash.choudhary}@gatech.edu

ABSTRACT

Large "instruction-tuned" language models (i.e., finetuned to respond to instructions) have demonstrated a remarkable ability to generalize zero-shot to new tasks. Nevertheless, they depend heavily on human-written instruction data that is often limited in quantity, diversity, and creativity, therefore hindering the generality of the tuned model. We conducted a quantitative study to figure out the efficacy of machine generated annotations, where we compare the results of a fine-tuned BERT model with human v/s machine-generated annotations. Applying our methods to the vanilla GPT-3 model, we saw that machine-generated annotations were 78.54% correct and the fine-tuned model achieved a 96.01% model performance compared to the performance with human labelled annotations. This result shows that machine-generated annotations are an resource and cost effective way to fine-tune down-stream models.

1 INTRODUCTION

The recent NLP literature has witnessed a tremendous amount of activity in building models that can follow natural language instructions Mishra et al. (2022); Sanh et al. (2022). These have shown that fine-tuning language models on a collection of data sets described via instructions—substantially improves zero-shot performance on unseen tasks.Wei et al. (2021). These developments are powered by two key components: large pre-trained language models (LM) and human-written instruction data (e.g., PROMPTSOURCE (Bach et al. (2022)) and SUPERNATURALINSTRUCTIONS (Wang et al. (2022), SUPERNI for short)). However, collecting such instruction data is costly and often suffers limited diversity given that most human generations tend to be popular NLP tasks, falling short of covering a true variety of tasks and different ways to describe them. Continuing to improve the quality and coverage of instruction-tuned models necessitates the development of alternative approaches for supervising the instruction tuning process.

The objective of the paper is to distil Instruction Fine-Tuning Data from Closed-Source Models to reduce resource overhead and determine how good these models are, in generating annotations for our chosen task. We aim to tackle the task of categorizing textual data into predefined categories or labels. To do this, we come up with three different methodologies. First, we use GPT-3 with zero-shot training. Secondly, we Fine-tune BERT Devlin et al. (2019) with human-annotated data. Finally, we fine-tune the BERT model Devlin et al. (2019) with GPT annotated data to general categories. In this paper we compared the performance of a fine-tuned BERT model with human and machine generated instructions on these three methods. We tried the following experiments throughout our project:

- We generated synthetic labels for two different tasks - Classification and Question and Answering. The two task helped us evaluate the correctness of labels generated and performance of fine-tune model holistically as there cover the complexities and semantic logic of English language.
- We also implemented 3 different methods to generate labels as we wanted to compare the quality of generated labels on varying involvement of human input.

*Code at - https://github.com/samaksh97/Deep-Learning-Project-

We hope that this project can serve as a reference for quantization of the how good the performance of a fine-tuned model is on machine-generated annotations. We believe that we would also be able to contribute to the cost dynamics management which is a major part of fine-tuning large language models.

1.1 DATASETS

1.1.1 CONFERENCE TITLE CLASSIFICATION DATASET

The study presents a dataset comprising 2,507 research paper titles manually classified into 5 categories, representing different conferences. With an average title length of 9 words, the dataset exhibits a class imbalance, with the majority class comprising 34.5% of records. We explore the application of stratified sampling to address the class imbalance and enhance the performance of a classification model.

	Title	Conference
0	Innovation in Database Management: Computer Sc...	VLDB
1	High performance prime field multiplication fo...	ISCAS
2	enchanted scissors: a scissor interface for su...	SIGGRAPH
3	Detection of channel degradation attack by Int...	INFOCOM
4	Pinning a Complex Network through the Betweenn...	ISCAS

Figure 1: Dataset Description.

1.1.2 SQUAD DATASET

The Stanford Question Answering Dataset (SQuAD) Rajpurkar et al. (2016) stands out with its extensive set of 107,785 question-answer pairs sourced from 536 Wikipedia articles in SQuAD 1.1. Within these chosen articles, they extracted 23,215 individual paragraphs, ensuring the exclusion of overly small paragraphs. The dataset was then partitioned by articles, allocating 80% for training, 10% for development, and 10% for testing, maintaining a balanced and representative distribution across sets.

NO.	Question	Document	Answer
1	Which NFL team represented the AFC at Super Bowl 50?	Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion **Denver Broncos** defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title...	Denver Broncos
2	Who was in charge of the papal army in the War of Barbastro?	The legendary religious zeal of the Normans was exercised in religious wars long before the First Crusade carved out a Norman principality in Antioch. They were major foreign participants in the Reconquista in Iberia. In 1018, Roger de Tosny traveled to the Iberian Peninsula to carve	William c Montreui

Figure 2: SQuAD description

1.2 Fine Tuning Language Models

Fine-tuning large language models is a process where a pre-trained model, initially trained on a vast and diverse dataset, is further trained (or "fine-tuned") on a smaller, more specific dataset. This process adapts the model to perform better on tasks related to the characteristics of the smaller dataset. The pre-trained model is then trained further on a more specialized dataset. This dataset is usually much smaller than the original training set and focused on a specific domain or task. During fine-tuning, the model's parameters are adjusted to better align with the specifics of the new data. This includes learning task-specific features and nuances. By fine-tuning, the model often achieves higher accuracy and better performance on tasks closely related to the fine-tuning dataset.

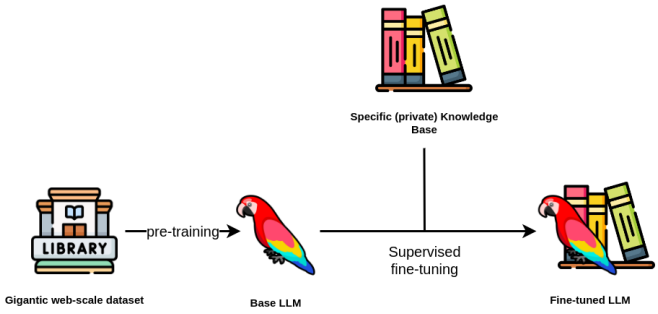

Figure 3: Fine-Tuning LLM

While there are strengths, fine-tuning also has limitations - there's a risk that the model may become too specialized to the fine-tuning dataset and lose its ability to generalize.
In summary, fine-tuning large language models is a critical step in tailoring these models for specific tasks or domains, enhancing their performance while mitigating the need for extensive resources required for training large models from scratch.

2 Related Works

Fine-tuning language models require collecting a small but specific dataset, in order to adapt and better align the model to a given task. In past works, this data has been manually collected by humans which result in an overhead resource cost in terms of humans and money. Like Wei et al. (2021), which transformed existing datasets from the research community into an instructional format and aggregated 62 text datasets that are publicly available on Tensorflow Datasets, including both language understanding and language generation tasks, into a single mixture as created an instruction instruction tuning dataset with many tasks from scratch would be resource-intensive. Other limitations which come from previous methods Wang et al. (2023), which aims to enhance the instruction-following capabilities of pre-trained language models (LMs) using self-generated instructions, is that only 58% of the outputs were found to be correct and acceptable responses to the instructions, indicating a need for further refinement. Even state-of-the-art models like Open AI's InstructGPTOuyang et al. (2022), was fine-tune on data which was more aligned to human values. They hired a team of 40 contractors to label the data and collected a dataset of human-written demonstrations of the desired output behavior and some labeler-written prompts. The limitations as not limited to data collection, ensuring the quality and relevance of the data for the specific task at hand is also crucial. Poor

quality data, including inaccuracies, inconsistencies, or irrelevant information, can lead to suboptimal model performance.

3 Data Collection

We first started our data collection with widely available datasets - Conference title and SQuAD dataset. For our study we are limiting the size of dataset to 2500 entries for each dataset. This number translates to 2500 question for SQuAD dataset. Second, we generated synthetic data using GPT-3 Brown et al. (2020), where we provided instructions to the agent depending on the task.

```
messages=[
        {"role": "system", "content": '''You are an assistant which can search for answers within a reading passage (context) for a given question.
           The answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.
           Following are a few examples of answers. Generate the entity only and not the complete sentence in the answer for the given questions
         and context and return answers in a numbered list.
           Question: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?
           Answer: Saint Bernadette Soubirous
           Question: What is in front of the Notre Dame Main Building?
           Answer: a copper statue of Christ
           Question: The Basilica of the Sacred heart at Notre Dame is beside to which structure?
           Answer: the Main Building
           '''},
        {"role": "user", "content": f"Context: {context}\n{ques_str}"}
]
```

(a) SQuAD Dataset

```
messages = messages=[
            {"role": "system", "content": ''' You are an assistant which can classify each paper
            in the list based on its title to one of the five following conferences:
               'VLDB', 'ISCAS', 'SIGGRAPH', 'INFOCOM', 'WWW' and return answers as a numbered list.'''}]
```

(b) Conference Title Dataset

Figure 4: Instructions for synthetic data generation

Separate data was generated for each methodology, with and without few shots. Fig. 4(a) shows instructions for SQuAD data generation with few shots, while Fig. 4(b) shows instructions for Conference Title data generation without few shots. For the classification task we also gave options to promote GPT-3 to give answers within those answers so as to decrease chances of hallucination.

4 Methodology/Experimental Setup

4.1 Classification via GPT-3.5

In this method, we will generate labels using GPT-3.5 with zero shot training and generate synthetic labels for the dataset. The idea here is to evaluate the performance of GPT-3 on creating synthetic labels based on its pre-training corpus and only instruction tuning and no fine-tuning or even few-shots prompting. We will evaluate the efficacy of generated labels by calculating classification accuracy against the actual labels in the dataset.

4.2 Fine-tuning BERT with human annotated data

In this method, we are fine-tuning pre-trained BERT model with human annotated labeled data. The idea here is to evaluate the performance of a fined-tuned model on actual labeled data and see the upliftment in accuracy for validation data. We will evaluate the performace of the fine-tuned model by calculation

Figure 5: Classification using zero shot GPT annotated data without fine-tuning.

classification accuracy on validation data against the actual labels in the dataset. Fig. 6 shows the process we followed for this methodology.

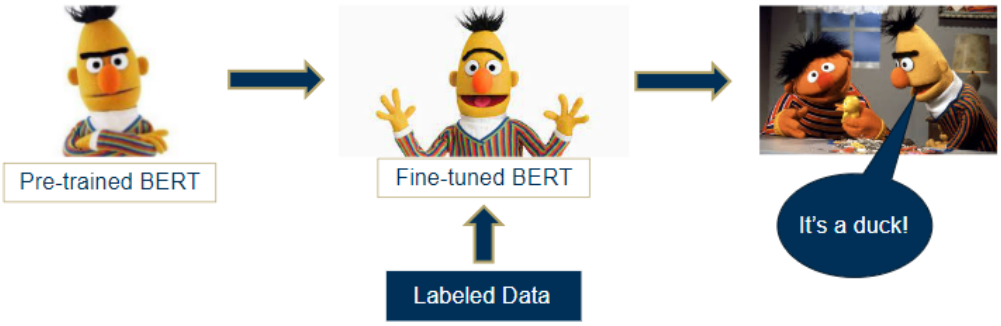

Figure 6: Fine-tuning BERT with human annotated data.

4.3 Fine-tuning BERT with GPT 3.5 annotated data

This approach combines elements from both methodologies mentioned earlier. Initially, synthetic labels generated through Methodology 1 with instruction tuning are utilized. Subsequently, a pre-trained BERT model is fine-tuned using these synthetic labels or data annotated by GPT. The goal is to assess the performance of a fine-tuned BERT model on GPT-annotated data, comparing its efficacy against actual labels. This proves crucial in scenarios where human-annotated labels are not feasible, and machine annotations via GPT are leveraged to create synthetic labels for fine-tuning. The approach aims to evaluate the model's adaptability to GPT-generated annotations in the absence of human-labeled data. Fig. 7 shows the process we followed for this methodology.

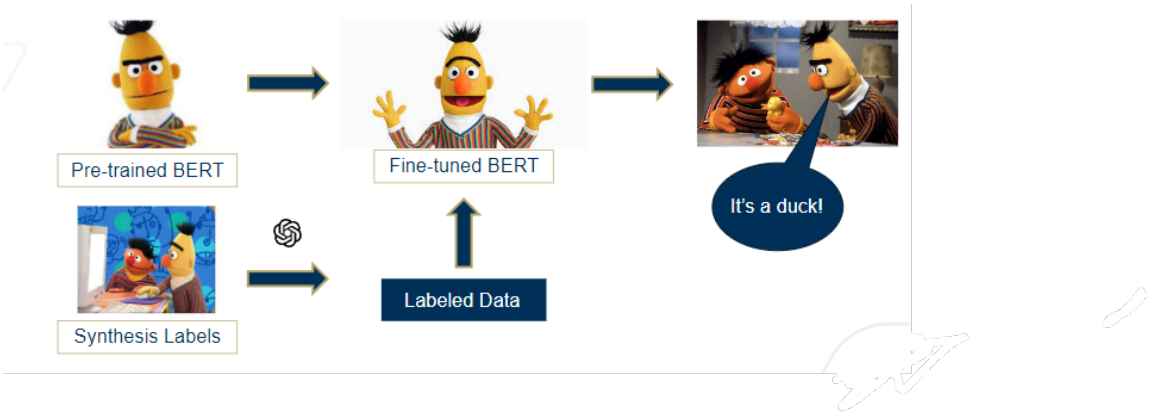

Figure 7: Fine-tuning BERT with GPT annotated data.

5 Results

We will compare the results for the above three methodologies on two datasets - Conference Title Data and SQuAD dataset. Intuitively, Methodology 2 should have the best results since the fine-tuning is done on actual labels followed by Methodology 3 in which we are fine-tuning on machine generated labels using GPT. Methodology 1 will have the worst results since we are generating labels using GPT without fine-tuning and zero shot prompting. We are comparing the results on these tasks in the results table 2

The below table 1 represents the parameters used while generating synthetic labels and fine-tuning the BERT model.

Type	**Parameters**	**Value**
API Call	GPT Model Name	gpt-3.5-turbo
Fine Tuning	Model Name	bert-base-uncased
	Optimizer	AdamW
	Learning Rate	1e-5
	Epsilon	1e-8
	Epochs	5

Table 1: Fine Tuning Parameters

5.1 Conference Title Data

From the table 2, we can see that Methodology 1 has a weighted average f1-score of 78.25%. Methodology 2 has the highest f1-score of 82.8% which is what we expected intuitively. Methodology 3 has an f1-score if 79.5% which is a slight upliftment from the Methodolgy 1. Fine-tuning on synthetic labels (Methodology 3) vs fine-tuning on actual labels (Methodology 2) only have a 3.3% absolute difference.

The above table 3 compares the results of human annotated data vs GPT annotated data on a relative scale. We can see that the cost for creating synthetic labels is negligible ,i.e. 0.12% of the cost of generating human annotated labels via MTurk or other surveys. However, the model performance is not depreciating a lot. It is 96.01% of what we would expect with a human annotated label.

		Correctly Predicted		
Category	**Total Count**	**Methodology 1**	**Methodology 2**	**Methodology 3**
VLDB	63	52	43	50
ISCAS	130	106	123	109
SIGGRAPH	48	44	39	41
INFOCOM	77	57	59	59
WWW	56	33	44	37
F1-score	**-**	**78.25%**	**82.8%**	**79.5%**

Table 2: Conference Title Data Results

Metric Comparison	**Human**	**GPT**
Label Accuracy	100%	78.54%
Model Performance	100%	96.01%
Cost	100%	0.12%

Table 3: Comparison of Metrics

5.2 SQUAD DATA

From the table 4, we can see that Methodology 1 achieves an accuracy of 39.54%. Methodology 2, i.e. finetuning BERT with Squad data sample, yields the best results with an accuracy of 48.99%. Methodology 3 yields the least favourable result with an accuracy of 33.62%. This is mainly because evaluating subjective question answers on the basis of accuracy might not yield the best results.

Dataset	**Methodology 1**	**Methodology 2**	**Methodology 3**
Squad data sample	39.54%	48.99%	33.62%

Table 4: Squad Dataset Results

On evaluating the GPT-generated training data labels, we observed that a lot of incorrect answers were because of one additional word or a different format for the output. Thus, we evaluated the quality of synthetic data using the BLEU score. Using the BLEU score distribution for training sample, as shown in 8, we obtained an accuracy of 70.80% by defining a threshold of BLEU score = 0.5.

For methodology 2, with Squad data sample of 2500 data points, we fine-tuned the BERT model with human-annotated labels. We trained the models for 4 epochs and achieved a validation accuracy of 48.99%, as shown in 5.

For methodology 3, with Squad data sample of 2366 data points, we fine-tuned the BERT model with GPT-annotated labels. We trained the models for 4 epochs and achieved a validation accuracy of 33.62%, as shown in 5.

6 CONCLUSION

During our experiments, we observed that GPT-annotated fine-tuning produces equivalent results to human annotated fine-tuning in the multi-class classification task(79.5% for GPT-annotated as compared to 82.8% for human annotated), whereas the performance deteriorates drastically in the question answering

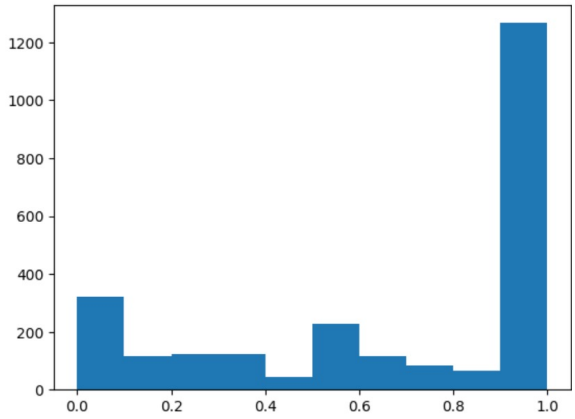

Figure 8: Squad training data BLEU score distribution

Epochs	Train Accuracy	Validation Accuracy
1	20.99%	40.52%
2	60.60%	47.68%
3	75.73%	48.61%
4	83.55%	48.99%

Table 5: BERT Finetuning using Squad Data (human annotated labels)

Epochs	Train Accuracy	Validation Accuracy
1	22.75%	25.57%
2	51.83%	35.18%
3	68.21%	38.52%
4	79.22%	33.62%

Table 6: BERT Finetuning using Squad Data (GPT-annotated labels)

task (33.6% accuracy for GPT-annotated as compared to 48.99% for human-annotated). We believe that the difference in performance is because of the difference in the objective tasks. The question-answering task is complex as compared to the classification task and requires a deeper understanding of the training data during fine-tuning. Thus, it is important that training data during fine-tuning is as accurate as possible. Further, the overall performance of the question-answering task is below par with the performance of the classification task. This might be because of limited data and a smaller number of epochs in training.

Future work can include experimenting with larger data, training for higher number of epochs, experimenting with prompts for generating more accurate synthetic response, and identifying better evaluation metric for subjective question-answering.

REFERENCES

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli,

Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-Jian Jiang, and Alexander M. Rush. Promptsource: An integrated development environment and repository for natural language prompts, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions, 2022.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

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, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization, 2022.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

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. *arXiv preprint arXiv:2109.01652*, 2021.