CSE 576: Project 17
Instructions Paradigm : An Alternative to Crowd sourcing

Pavan Kumar Raja
MS in Computer Science, School of Computing and Augmented Intelligence, ASU.
praja3@asu.edu

ABSTRACT

An experiment on the capability of state-of-the-art NLP models to generate a version of CoLA benchmark datasets using instruction prompts.

The premise of this project is to use the Instruction paradigm on the SOTA models like GPT-3 and T0_pp to generate datasets corresponding to i. The Quora Question Pairs (QQP), ii. MultiNLI, and iii. The Corpus of Linguistic Acceptability (CoLA). I evaluated these against the original dataset and reported the results.

1 INTRODUCTION

The prevailing way of creating a dataset is using the method of crowdsourcing which is expensive (ex: DROP dataset was created using an expense of 60k USD [1]) and poses the risk of containing artifacts. The advent of the SOTA language models like GPT-3 and T0_pp which supports sentence generation via prompting has given us an interesting hypothesis - "Can the instruction paradigm help in replacing humans with machines for dataset creation?", and in this project we are trying to verify this hypothesis. Answering this hypothesis helps us in creating an NLP pipeline where i. Bias by human intuition is eliminated. ii. Labor costs are reduced. iii. Dataset cleaning, aggregation, and maintenance can be built into the pipeline.

Contributions. Although this project was taken up as a team, each team member had different datasets to work on (In my case it was QQP, MultiNLI, and CoLA). I'll only be talking about these datasets, and all the contributions for these were entirely my own.

2 METHODOLOGY (DATASET CREATION)

I used discrete prompt mining and prompt scoring methods where I took references from the original dataset to design a prompt and paraphrased it until the results were satisfying. I have further experimented with prompt augmenting using few-shot learning. I have experimented on GPT3_davinci-instruct-beta-v3 for few-shot prompt learning and T0_pp for zero-shot learning and resorted to T0_pp because it had similar/better results for tasks at hand and was open source.

2.1 The Quora Question Pairs (QQP)

The QQP dataset has three fields question1, question2, is_duplicate - Indicates whether these two questions mean the same [2]. I had to look at the original dataset for prompt engineering.

Code bases:
i. Project github repository link.
ii. Personal github link (backup)

One of the major findings is that a positive case is not always a pair of paraphrased questions. Positive pairs can also be relatively related questions. Example: *[334577, My questions are fine. Why does Quora keep saying they need improvement? It's frustrating., All my questions on Quora need improvement. What is the best way to ask a question on Quora?]* [2]

Points on experiments and dataset creation:

(1) I experimented on the GPT-3 davinci-instruct-beta-v3 model with few-shot prompting and T0_pp for zero-shot prompting. GPT-3 tends to follow the example template very closely and gives too many duplicates so I went with T0_pp for better diversity. Prompt for T0_pp was trained in steps. First I generated a random question, and then I paraphrased/ generated a related question from the first.
   - GPT-3 ***Prompt:*** *"Generate a question pair. Example: How do you remove spray paint from a mirror?".* ***Generated texts:*** *What is the best way to remove paint from glass? How do you remove spray paint from a mirror? What is the best way to remove paint from glass?*
   - T0_pp ***Prompt 1:*** *"Generate a random long question."* ***Generated text:*** *What is the name of the fictional country in the Harry Potter stories?* ***Prompt 2:*** *Generate a question on: q1.* ***Generated text:*** *Where is Harry Potter located?*

   We can see that T0_pp not just paraphrases the Q1, but generates a related question with the same answer. So I used T0_pp.
(2) For the Negative case I just generated 2 random questions.
(3) I ended up with 13K Question Question pairs. Further to pre-process these questions I filtered the dataset to remove duplicate questions, sentences that didn't end in '?', and sentences that got trimmed because of API character limitation. In the end, I got 7.5K rows of data.

2.2 MultiNLI.

The MNLI is a dataset with (sentence1-parse_tree, sentence2-parse_tree, sentence1, sentence2, label - with value entails, neutral, contradiction indicating relation between sentence1, sentence2) [5], we are interested in only 2 of the sentences and label. Example: [***Sentence1:*** *There are obstacles to innovation that the participants recognized.* ***Sentence2:*** *However, the participants recognized that one of the big obstacles for innovation in.* ***label:*** *entailment*]

MultiNLI mainly consists of hypotheses and premises belonging to genres like - Conversational, Book inference, Image caption, etc, and these were considered during prompt mining, and the resultant dataset has the best possible distribution of these genres.

From the Original training examples, I could infer that i. Premise can be short, ii. The hypothesis can be paraphrased. iii. Further looking at the dataset, domain-specific dataset premises were belonging to normal article premises and conversational premises.

With the above findings I experimented with prompt scoring:

(1) I again found that T0_pp was better than GPT-3 in generating the premise and hypothesis. GPT-3 tends to follow the examples closely, even with changes in model prediction temperature.
(2) The prompts to T0_pp was zero-shot and had a combination of generating a long/short/conversational premise, generating a random text from the premise. Example:
Premise: *She asks if the man in the bar was talking to someone in a bar.*
Hypothesis generated: *Very funny people who work at a bar.*
Label: *neutral*
Generating a hypothesis from the premise. Example:
Premise: *A cyborg has been created by melding together the DNA of five different cyborgs. It has no memories and no awareness of its surroundings. Its creator believes that it is the perfect soldier because it can be trained to do anything. However, the cyborg has a stutter and does not understand what he is saying. The cyborg kills several of the human soldiers and uses the experience to improve his fighting skills. It is now up to his creator to decide if the cyborg should stay in the military or be sent back to his home world.*
Hypothesis generated: *A cyborg has been created by melding together the DNA of five different cyborgs.*
Label: *entailment*
and contradicting statement of a hypothesis from the premise (It was not possible to generate contradicting hypotheses directly).
(3) I generated around 30k samples and ended up with 21k samples after filtering out duplicates and short premises/hypotheses.

Key findings. Generally, the T0_pp model failed to generate a premise when the hypothesis was presented/ended with a question and it rather tried to answer the question. **Premise:** *Why is the premise of the argument that women have a harder time in gaining employment true?* **Hypothesis generated:** *Women have a harder time in gaining employment than men.* **Label:** *entailment* A few data rows also reveal that if the premise is short, the hypothesis is just an addition or paraphrase of a few words. **Premise:** *A thigh high bar is my.* **Hypothesis generated** *A thigh high bar is a bar in my.*

These examples represent some drawbacks of this paradigm, first examples show that the model is not only trained in sentence inference and sentence generation but also Question answering tasks. the second example shows that the hypothesis generation task requires enough information from the prompt itself. These examples show that we would get a better result if we choose the state-of-the-art language models appropriately trained on related task as per our requirements.

I have left these examples in the dataset as it shows the drawback of the instruction paradigm.

2.3 The Corpus of Linguistic Acceptability (CoLA)

CoLA includes a dataset where each row has a sentence and 2 annotations indicating whether these sentences are acceptable grammatical sentences belonging to the English model or contain mistakes. The annotations are in front of the experts and the original author of the sentence/article [4].

This dataset was by far the hardest to prompt engineers, and I failed to come up with a mechanism to generate this kind of data from prompting state-of-the-art language models like GPT-3 and T0_pp.

(1) The negative cases in the dataset where there are grammatical errors are more semantical grammar-related albeit they were few syntactic examples. This is the major hurdle in prompt engineering, I couldn't identify/ aggregate all the semantic error types to try a few shot/one-shot learning.
(2) I tried one shot and a few shots prompting GPT3 'davinci-instruct-beta-v3' and zero-shot learning with T0_pp. The experiment included, i. Trying to generate an ungrammatical sentence with examples (GPT3) and without examples (T0_pp), ii. Given a prompt to generate grammatically correct sentences and then induce grammatical mistakes with and without examples (multi-steps). In all the permutations of the prompts and models, I couldn't get conclusive evidence that a prompt could generate ungrammatical sentences.
(3) Experiment examples of GPT-3 with Few-shot learning -
Prompt: *"Generate an ungrammatical sentence. Example:Bill reading Shakespeare and Maureen singing Schubert satisfy me. Example:A pastor was executed, notwithstanding on many applications in favor of him. Example:What exploded when and I warned you it would?"*
Generated text: *Example:The one that is related to the sentiment of pity is pleasing to me. Example:A picture of the boy striking a viper, it was exhibited at the Royal Academy. Example:I have a feeling that I am an item of a series."*
One-shot learning –
Prompt: *Generate an ungrammatical sentence. Example:George is having lived in Toledo for thirty years.*
Generated text: *When this sentence is fed into the parser, it will generate a syntactic parse tree as a sentence structure. In the example of George is living in Toledo for thirty years, the parser generates a sentence structure as: S (S (NP (D George)) (VP (V is) (NP (NP..]*
(4) To emphasize how semantically skewed the dataset was, I create the script to introduce 15 grammatical errors (including noun replacement, pronoun mismatch, tense change, adjective change, random capitalization, wrong punctuation, etc.) randomly to create a dataset of 31k samples. I created a pipeline for this task based on the bert_base_uncased encoder followed by a fully connected MLP trained on the same model first on the original dataset and second on the manually generated dataset and used the original test set, I got 78.97% and 39.94%.

Key findings. I was not able to generate a dataset that had similar distribution as the CoLA dataset. Manually created datasets with

syntactical errors did not belong to the actual distribution from the evaluation report.

3 EVALUATION

The main idea for evaluating these datasets is to find if they both belong to the same distribution syntactically and semantically. We do this by creating 2 fined tuning models based on all-MiniLM-L12-v2 with the same hyperparameters, the first one trained on our prompt generated dataset and the second one trained on the original dataset. We evaluate both models on the evaluation dataset from the original dataset. Comparing evaluation matrices of the models gives us information on how close the generated dataset is to the original dataset.

3.1 The Quora Question Pairs (QQP)

I used fine-tuning on pre-trained model **all-MiniLM-L12-v2** [3], I experimented on binary cross-entropy loss and (OnlineContrastiveLoss + MultipleNegativesRankingLoss) latter resulted in better results on both original and generated datasets.

Hyper parameters: lr: 2e-05, weight_decay: float = 0.01, batch_size: 64, epochs: 30

Figure 1 shows us the evaluation result of the models on evaluation data taken from the original QQP dataset.

Dataset	Cossim acc	Cossim f1	Cossim recall	Cossim precision
Original	0.82	0.77	0.85	0.7
Prompted	0.77	0.73	0.85	0.63

Figure 1: Evaluation of QQP Generated vs Original

Key findings. From the results, we can confirm that the dataset created from the zero-shot prompting on T0_pp had similar distribution as the actual database and with further fine-tuning or training we can further improve the quality of the dataset.

3.2 MultiNLI.

I used fine-tuning on pre-trained model **all-MiniLM-L12-v2** [3], I experimented with softmax loss function and label accuracy evaluator which resulted in a very low evaluation metric on original evaluation dataset. So instead I used multiple negative rankings and augmented the training dataset with [premise, hypothesis] and [hypothesis, premise pairs]. This model was evaluated using a scaled STS dataset (0-1 scores) to evaluate the scoring of texts entailment in the model.

Hyper parameters: lr: 2e-05, weight_decay: float = 0.01, batch_size: 64, epochs: 30.

Figure 1 shows us the evaluation result of the models on evaluation data taken from original MultiNLI dataset.

Dataset	Cosine Pearson	Cosine Spearman
Original	.84	.839
Prompted	.83	.82

Figure 2: Evaluation of MultiNLI Generated vs Original

Key findings. Pearson and Spearman correlation values confirm that the model has trained well to score the hypothesis & premise pairs on a similarity scale. We can see that these 2 datasets belong to the same distribution, it can be attributed to consideration of genres used MultiNLI during prompt engineering.

4 CONCLUSION

In this project, I have experimented with using an instruction-based paradigm to generate datasets belonging to QQP, MultiNLI, CoLA. I have found that the prompt paradigm works well when the data belongs to the general language model and ground truth can be extracted/labeled during dataset generation (SST, QQP, MNLI, etc,.)

I have used variants of GPT-3 and T0_pp models for these experiments. GPT3 works well to extract information from generated data with the help of few-shot learning, whereas T0_pp is good at zero-shot prompts to generate data belonging to a specific domain and manipulate the existing data(QQP) and thus, I preferred T0_pp for this particular task. Furthermore, I experimented with the pipeline to evaluate the prompt-based datasets generated for QQP, MNLI, and manual datasets for CoLA. Although QQP and MNLI showed promising results upon closer look at the actual dataset we can say they had their drawbacks (Ex: I had to filter out almost 30% of the data from QQP, as the paradigm failed to generate related questions instead it answered them, which could be due to T0_pp being trained on QA dataset and mistaking the prompt/task at hand.) I also noticed that this Instructional prompting approach does not work very well with datasets like CoLA as writing prompts for generating semantically wrong text are harder and often the result is left to interpretation, whereas it worked very well for QQP (where I could filter dataset easily).

To answer the hypothesis "Can the instruction paradigm help in replacing humans with machines for dataset creation?". For simple tasks indeed it can, but for complicated tasks, human inference of the language outweighs the LM's ability to generate coherent sentences.

REFERENCES

[1] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. (2019). arXiv:cs.CL/1903.00161

[2] Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First Quora Dataset Release: Question Pairs. https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs. (2017).

[3] Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. (2020). arXiv:cs.CL/2004.09813

[4] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. (2019). arXiv:cs.CL/1805.12471

[5] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101