Report

1. Introduction

In recent years, the focus of Large Language Model (LLM) research has been whether these models can truly understand language as humans do [0, 1]. Semantic plausibility is one such aspect. To check if LLMs can distinguish whether a statement is plausible or not given a context, the task of plausibility will be mapped to predicting hallucinations. For that, I will be using the dataset SHROOM [2] to check whether a model can accurately predict if a statement given a context is hallucinated or not.

It is common knowledge that training on more data, given that the training data is of high-quality, leads to better model performance. When only training set is small, it usually is best practice to collect more training data before doing anything else. In this work, we will test if synthesizing data to get more training data is helpful for boosting the performance on hallucination detection.

2. Goal Definition

The goal of this work is to determine if using synthesized data can enhance the accuracy of predicting whether a model's output is hallucinated.

3. Data

The model-agnostic dataset contains LLM generations given a context across three tasks. The dataset consists of a train, dev and test split. Additionally, 5 human annotations for each sample in the validation and test set are available. The annotators were asked to classify an LLM generation binarily whether it is hallucinated or not.

The train split contains 30k samples, which are unlabeled. The dev split contains only 499 samples and the test split 1500. The dataset is composed of three tasks, namely definition modeling, machine translation, and paraphrase generation.

The following image displays the count of these three tasks in each split:

```
Count of each task in train set: {'PG_count': 10000, 'DM_count': 10000, 'MT_count': 10000}
Count of each task in valid set: {'DM_count': 187, 'MT_count': 187, 'PG_count': 125}
Count of each task in test set: {'PG_count': 375, 'DM_count': 563, 'MT_count': 562}
```

Each sample has multiple columns and the columns that I will consider in this work are task (name of the task), src (model context for generation), tgt (gold / reference generation), hyp (actual model generation) and p(Hallucination) (the probability of the sample being hallucinated calculated with the 5 human annotations).

4. Model

The model used is "OpenAssistant/reward-model-deberta-v3-large-v2" available on the HuggingFace Hub [3]. The rationale is that the model was trained to serve as a reward model for Reinforcement Learning from Human Feedback. That is, it was trained on human preference data such that it yields a high score for human preferred generations and a low score for bad generations. As no hallucination is naturally preferred by humans, this model should be a good baseline for my work.

To fit the model into memory, we used peft LoRA with the following parameters: LoraConfig(task_type=TaskType.SEQ_CLS, r=128, lora_alpha=128, use_rslora=*True*, target_modules="all-linear"). After training the model with peft, I merged the peft adapter with the model.

5. Experimental setup

5.1. Training a Classifier for Synthesizing Data

The labeled dev set will be used to train the classifier. It will be split up by 80% for training and 20% for evaluation. I made sure that the splits have a balanced amount of different difficulty levels (defined in Section 5.1.1.). To leverage the more fine-grained hallucination probability instead of the binary label hallucination, no hallucination, I used the Mean-Squared Error (MSE) as a training objective, where the hallucination probabilities serve as the labels. To use the classifier for inference, I cutoff is set, e.g. at 0.5, to classify samples above the threshold as hallucinated and samples below as not hallucinated.

5.1.1. Curriculum Learning [4]

In curriculum learning, models are first trained on the easy samples and then the harder ones. Intuitively, for shroom hard samples would be the ones were human annotators had a high disagreement, i.e., p(Hallucination) values of 0.4 and 0.6, and easy samples would be where human annotators had a high agreement, i.e., p(Hallucination) values of 0.0 and 1.0. To test whether curriculum learning is useful, we will check if training via curriculum learning is significantly better than training conventionally, i.e., all data samples mixed irrespective of their hallucination probabilities. For testing significance, we will compare with 5 different seeds and using the paired t-test.

5.1.2. Discarding 'difficult' train samples?

Another intuition is that samples with high human disagreement contain a lot of noise. Such samples could hinder the training progress. To test whether this hypothesis is true, I will first train solely on each difficulty level and secondly also check for significance by training the classifier with all data but wih and without the hard samples on 5 different seeds and using the paired t-test.

5.1.3. Flipping hallucination probabilities for training

The selected (reward) model outputs high scores for good generations. However, we further train the model on the hallucination probabilities and therefore, a higher score would indicate a higher chance of hallucination, which corresponds to bad generations. To not 'confuse' the model, I flipped the hallucination probabilities such that a score of 1 would indicate the probability of no hallucination and vice versa. This ensures that the further training is aligned with how the model was trained.

5.2. Synthesizing the unlabeled training data with the classifier

The trained classifier was used to synthesize labels for the unlabeled training data. Note that the model does not output a binary label but a score due to the use of MSE as the training objective. To leverage this score, I sorted the training data by the scores and filtered the training data such that only the top and bottom n samples are considered, and the rest being discarded. Intuitively, the top n samples correspond to classifying non-hallucinated samples for which the model is most confident in, and the bottom n samples correspond to classifying hallucinated samples for which the model is most confident in. I selected n=5000 which leads to 10000 training samples in total.

5.3. Training a model on synthesized data

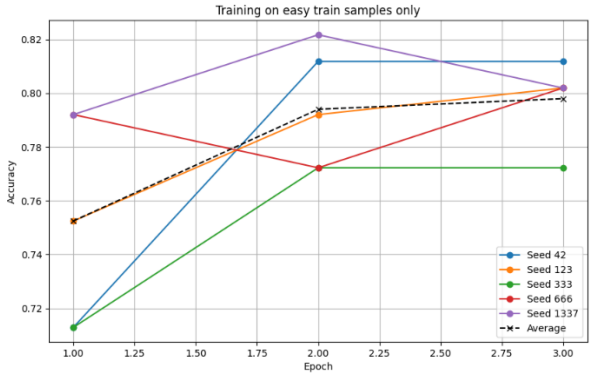
The same baseline reward model will be used to train on the MSE objective on the 10000 synthesized data samples. As mentioned earlier, the model outputs scores not necessarily between 0 and 1. Hence, the best cutoff point might not be 0.5 for hallucination or no hallucination. To get a good 'read' on the best cutoff point, we will be using the original dev set to calculate the accuracy on different cutoff points and select the one with the highest value.

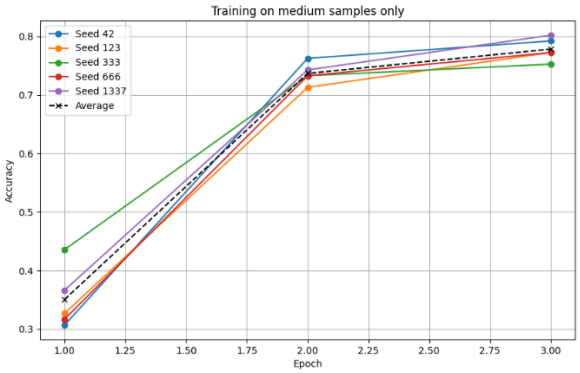
5.4. Visualization on Test Set

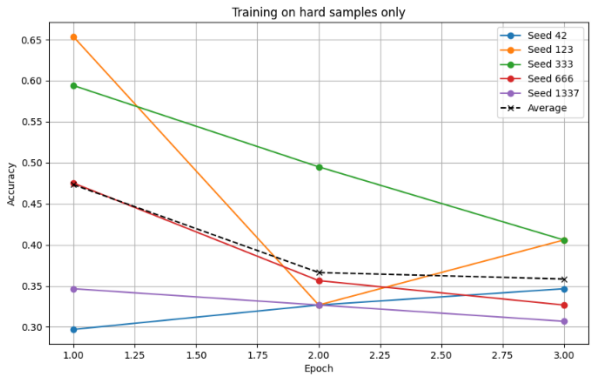
Finally, I will visualize the model's performance on the test set by creating a histogram that displays the distribution of scores, along with the corresponding labels for each sample that produced those scores.

6. Results & Analysis

6.1. Training on each difficulty level separately

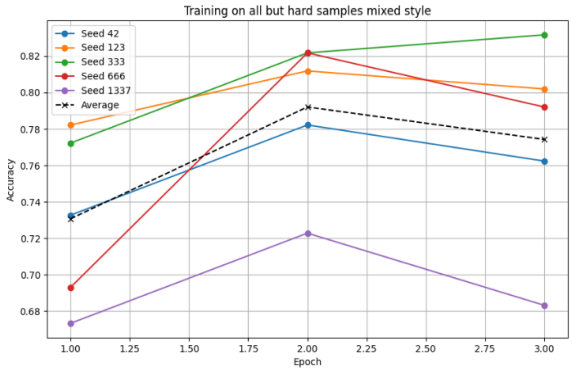


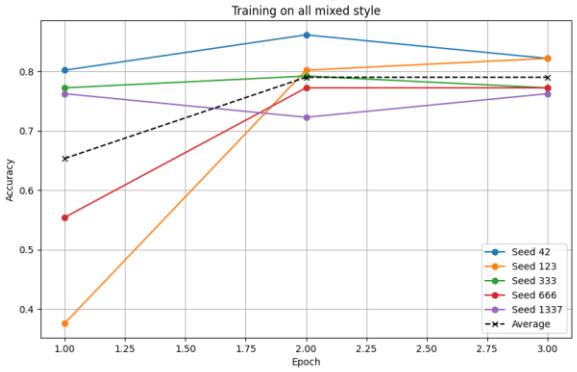




From the above plots, easy and medium samples can be regarded as high-quality samples as the accuracy on both almost reach 80% on average. In contrary, training only on the hard samples plummets the accuracy after each epoch for every training seed indicating samples of low quality. However, at this point, no decision can be made about the usefulness of hard samples in training as training consists of all samples.

6.2. All Training Data vs All but Hard

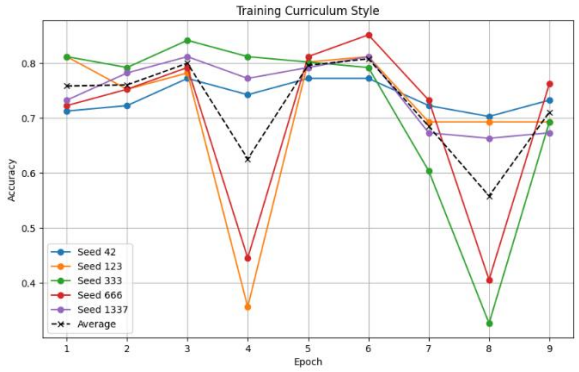


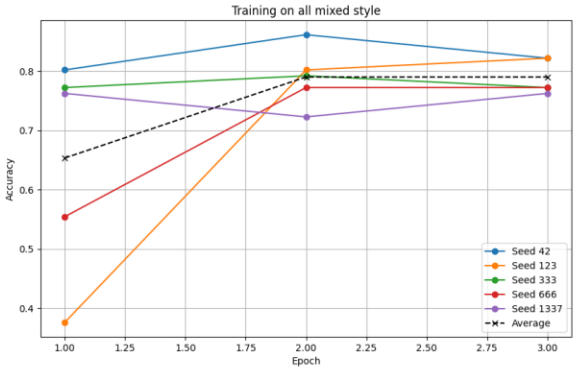



```
Comparing mixed data without hard samples and mixed data with all samples
T-statistic: -0.6246950475544235
P-value: 0.5660371777000412
```

The T-statistics value are slightly negative which indicates that on average not using the hard samples for training slightly improves the accuracy. However, with a p-value of 0.566 this result is not statistically. An interesting observation is that solely training on hard samples hurts the performance a lot (see Section 6.1.) whereas training on all samples including the hard ones does not. So in conclusion, even though the result was not statistically significant, I still left out the hard samples out of the training due to the performance on training only on hard samples.

6.3. Curriculum Learning vs Mixed Style (Conventional)






```
Comparing Curriculum Style Training with all mixed data training
T-statistic: -4.103913408340618
P-value: 0.014805588185026699
```

Please note that while I always trained for 3 epochs, the curriculum style implementation results in 9 epochs being shown, with 3 epochs dedicated to each of the 3 difficulty levels.

The T-statistics are negative indicating that that on average the curriculum style learning has lower accuracy. With a p-value of 0.0148 and a significance level of 0.05, this result is also statistically significant. Thus, I trained the classifier conventionally on all the samples shuffled.

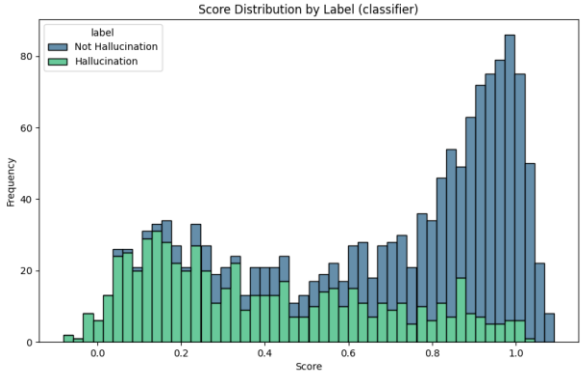
An interesting observation is that huge drop-off in performance for some seeds at epoch 4 and 8 corresponding to after starting training on the medium samples for 1 epoch and after training on the hard samples for 2 epochs respectively. I am not exactly sure what caused this pattern, but I hypothesize that this could be due to the low sample size and the corresponding high variation which can occur during training.

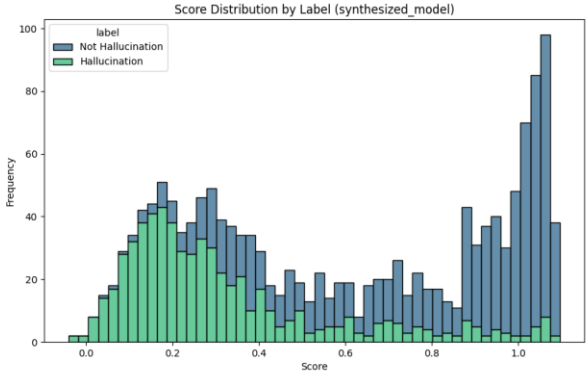
6.4. Accuracy on Synthesized Data 10k Samples vs Real Data 499 Samples

	Synthesized Model	Classifier
Cutoff Point	**0.35**	**0.4**
Accuracy	**0.803333**	**0.787333**
ROC AUC	**0.786997**	**0.754818**

The synthesized model shows a slightly higher accuracy and ROC AUC score than the classifier model. However, the result might not be statistically significant as no different seeds were tested for training due to time constraints.

6.5. Score Distribution by Label in Histogram on Test Set





The histogram for the classifier model seems to perform better on classifying hallucinations as only few were wrongly labeled around the score range of 0.0 to 0.4. On the contrary, the synthesized model does classify a lot of samples wrongly as hallucination between the score range of 0.2 and 0.4, but for classifying not hallucination it does seem more accurate than the classifier model as almost no samples were wrongly classified as hallucination.

7. Conclusion

Overall, training the model with synthesized data shows interesting results compared to training the model conventionally. The classifier model seem to classify hallucinations better whereas the synthesized model the not hallucinations better. This finding suggests a potential strategy: using both models to classify a sample and only accepting the classification if both models agree. This approach would result in highly confident classifications.

Also, the results indicate that synthesizing data to get more training data does increase the accuracy of classifying hallucinations compared to training on little but real-world data. This concludes that synthesizing data could be a valid strategy to look in when trying to get more training data.

References

[0] https://arxiv.org/pdf/2402.00858

[1] https://arxiv.org/pdf/2402.04614

[2] https://arxiv.org/pdf/2403.07726

[3] https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2

[4] https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/15972/Bengio%2C%202009%20Curriculum%20Learning.pdf