Report Template for CSE 447/517 Winter 2024 Final Project

Ray Song
447
syx1995@uw.edu

Hao Xu
447
xuhao510@uw.edu

Hongjian Yu
447
hjyu@uw.edu

Project Information:

Fill out this part for the midway report.

Including one of the following three tables depending your project type.

Default 517 Project: If you choose to replicate and extend an existing research paper, please specify the information of this paper in the following table.

Project Type	Default 517 Project
Original Paper Title	VERA: A General-Purpose Plausibility Estimation Model for Commonsense Statements
Venue	EMNLP
Year	2023
URL	https://arxiv.org/abs/2305.03695

You don't need to fill out this part for the midway report, but you should fill this out for your final report.

Specify the individual contributions.

- **Author One:** A sentence or two describing the contribution of author one.
- **Author Two:** A sentence or two describing the contribution of author two.
- **Author Three:** A sentence or two describing the contribution of author three.

1 Introduction (mandatory)

We replicate the paper proposing VERA, a general-purpose commonsense statement verification model by Liu et al.. This model is able to measure the plausibility of a statement based on commonsense knowledge.

Large Language Models (LLMs) appear and grow rapidly in recent years, with remarkable capabilities in various fields such as text generation, conversation, and classification. However, LLMs still sometimes make mistakes on even simple commonsense knowledge. This model compensates the lack metric in measuring the correctness of commonsense statements.

The experiment in this report is divided into two parts. First, we reproduce the experiment described in VERA paper, aiming for results to be close to ones reported in original paper. Then, we build on the source code to tackle potential limitation proposed by the paper, *[PLACEHOLDER: new experiment]*.

VERA is built on T5 (Raffel et al., 2020), and fine-tuned on commonsense statements sourced from knowledge bases (KB) and question answering (QA) datasets. The model is trained using two-stage training, which helps mitigate the negative effects of quality and diversity of large dataset from various sources. Then the model is evaluated on three tasks: (1) repurposing for verification, (2) filtering LM-generated commonsense knowledge, and (3) detecting commonsense errors in GPT-generated sentences.

[Note for midway report: up to midway report deadline, we focus on first part of the experiment (i.e reproduce VERA using provided source code). Expect extensions to be finished by final report deadline. After the milestone, we plan to enhance VERA for Long-form Statements by developing an advanced module for VERA to effectively process and evaluate long-form statements.]

2 Background (optional)

You're encouraged to describe the background of your project here. For instance, you may discuss related work or provide background knowledge about the problem that you study (especially if it involves a specific domain, e.g., chess).

3 Method (mandatory)

Describe your current method and how you're planning to improve on it in your final report. Discuss any advantages of your method (e.g., requiring fewer resources), and provide intuition about why your method makes sense for the problem you are trying to solve. Aim for your explanation to be understandable to any other student in the class.

3.1 Data Collation

Training data of VERA have been collated from knowledge bases (KBs) and question answering (QA) datasets. Boolean QA statements are converted to a predicate and its correctness label. Multiple-choice QAs are converted to sets of right and wrong statements depending on the correct choices.

Statements originating from the same problem count as a statement group. In a statement group, at most one statement is correct. When training, statements from one group must be batched together.

In order to prevent the model from overfitting non-generalizable patterns in QA and KB data, multiple-choice QA and KB entries are augmented for additional false statements. Responses to QAs are supplemented by a small language model from its outputs with least generation probability; Each KB entries are paired with three copies of itself with the subject replaced from random selections within the dataset.

3.2 Model Architecture

VERA is the finetuned result of a pretrained transformer-based language model. A forward pass of VERA encompasses the following processes:

- The underlying model takes the input x to generate an encoded representation $\mathbf{h} = f_{LM}(x)$ at the position of EOS token (i.e. the last hidden state of the left-to-right sequence).
- A linear classification layer projects the encoded representation $\mathbf{h}$ to a scalar logit $z = f_{linear}(h)$.
- The logit z is converted to a sigmoid score $s = \sigma(z)$.

3.3 Training Objectives

During training, the model aims to minimize a linear combination of three losses, summarized by the equation $\mathcal{L} = \alpha\mathcal{L}_{bin} + \beta\mathcal{L}_{mc} + \gamma\mathcal{L}_{ctr}$:

- For the commonsense verification task, $\mathcal{L}_{bin}$ is the binary classification loss to evaluate prediction s, given the corresponding label y. For each input x, the equation reads:
$$\mathcal{L}_{bin} = -y \log s - (1-y) \log (1-s)$$
- For the multiple-choice find-the-correct-statement questions, $\mathcal{L}_{mc}$ measures the multi-class classification loss for statement groups, which is calculated by the negative log of the softmax probability for the correct statement's logit z_*:
$$\mathcal{L}_{mc} = -\log \frac{\exp z_*}{\sum_c \exp z_c}$$
- To improve generalization and robustness of the model, a third metric, supervised contrastive loss $\mathcal{L}_{ctr}$, is introduced to ensure consistency across predictions for correct and incorrect statements. $\mathcal{L}_{ctr}$ is small when the distance between contrastive predictions is large. For an anchor statement x_i, its contrastive loss reads:
$$\mathcal{L}_{ctr} = -\log \frac{\Sigma_{j \in \mathcal{P}(i)} \exp \left[\cos \left(\mathbf{h}(x_i), \mathbf{h}(x_j)\right)/\tau\right]}{\Sigma_{j \in \mathcal{P}(i) \cup \mathcal{N}(i)} \exp \left[\cos \left(\mathbf{h}(x_i), \mathbf{h}(x_j)\right)/\tau\right]}$$
where τ is the temperature hyperparameter, $\cos(\cdot, \cdot)$ is the cosine similarity function, and $\mathcal{P}(i)$ and $\mathcal{N}(i)$ are the index sets of positive and negative examples derived from x_i respectively.

While data from different sources vary in size and quality. The authors of VERA made a choice to split training into two stages. The first stage involves training on large but noisier data; the second stage trains on better curated but smaller data. The result has empirically been proven to outperform models trained on mixed data.

3.4 Enhancement

In this subsection we will present our changes to the model after experimentation with our modified version.

We will try to achieve the following goals:

- Developing an advanced module for VERA to effectively process and evaluate long-form statements. We plan to build an inference-time module for vera to augment long training inputs. Input with length over a threshold will be fed into a summarization model, and the summarized text will be fed into VERA in parallel to the original input. The insert a summarization algorithm to condense the essence of extensive narratives, ensuring the core message is retained for accurate plausibility assessment. Alternatively, dissecting lengthy statements into coherent, smaller segments for individual analysis could be explored, aiming to maintain the contextual integrity of the original statement.

4 Experiments (mandatory)

Describe your current experiments and what future experiments you're planning to run. Be clear about your experimental set-up and metrics for measuring performance.

For the Default 517 project, be clear about which experiments are a reproduction from the original paper and which experiments are new. Please mention any design decisions you made because the information was missing from the original paper, or made differently due to different considerations.

4.1 Model (optional)

What models do you apply your method to?

We fine-tune VERA on top of T5 Raffel et al., 2020, a bidirectional encoder model. The original paper built VERA on top of T5-v1.1-XXL and also trained VERA on top of LLaMA Touvron et al., 2023 but found it perform worse than VERA-T5. Considering our computational resource limit and relative performance between different sizes of T5, we use `T5-small`, but expect our result to be close to paper's.

4.2 Datasets (optional)

What datasets do you evaluate on?

As described in original paper, we use two-stage training. In training stage A, we use about 6M statements from two commonsense knowledge bases. In training state B, we use about 400k statements from 19 commonsense question answering datasets. The source for dataset is provided in the appendix of original paper, and we thank the author for providing processed dataset for training.

We are trying to evaluate datasets for unseen data types 1 and 2, but we haven't got all the datasets for this phase. We are processing all raw datasets to UnifiedQA format and then use **question-converter-t5-3b** to convert all QAs to statements. With the current datasets we have, we have evaluated on the all seen datasets, here is the spreadsheet of all evaluation results.

4.3 Baselines (optional)

What baselines do you compare to? These may include methods from prior work as well as ablations of your method.

4.4 Code (mandatory)

Put a link to your code or the codebase that you use from others. If you used an existing codebase, please describe what kinds of revisions or additions you made (if any). You do not need to include this for the milestone report.

Github Repo

5 Results (mandatory)

Present and discuss your results. How does your method compare to baselines? Any surprising findings? For the Default 517 project, discuss whether your results match those from the original paper.

If it makes more sense, for instance if you have multiple experiments, you may also combine your Experiments and Results section, and interleave each experiment with their results.

Due to the computing power constraint, we can only use 1 Tesla P4 GPU, we cannot handle T5-xxl, so we trained based on T5-small(60M) instead.

Before midway report due, we finished two stages of training. We observe increase in accuracy and decrease in loss in training log. The log of training process can be found in link. We are doing evaluation and expect the evaluation results to be added in the following week.

After the evaluation of the all seen datasets, the highest accuracy is 52.08 on quartz, and the lowest is 23.80 on openbookqa, the average is 41.85. Compared to the baselines in the paper, ours(Vera-T5-small, 60M) is better than SKD Critic(355M), but worse than I2D2 Critic (355M).

6 Discussion (optional)

This is a flexible space for you to use. For instance, you might discuss implications of your results for the broader NLP community, hypotheses about why you see the results you do, or any insights, confusions, and new curiosities stemming from your project.

For the Default 517 project, you can also consider discussing any thoughts you have about the original paper after going through the process of reproducing it. If you were writing this paper, what would you have done similarly or differently? Do you believe its experimental setup and conclusions are sound?

7 Conclusion (mandatory)

In this section, you should briefly summarize your contributions, state the key takeaways, and potentially mention directions for future work. *[Placeholder: to be added in final report]*

Our initial naive thought is Vera does improve the model's capability on commonsense statements estimation, and increasing the base model size can also improve the performance.

References

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hanna Hajishirzi. Vera: A general-purpose plausibility estimation model for commonsense statements. *ArXiv*, abs/2305.03695, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

A Appendix

You may include other additional sections here if needed.

Acknowledgments

This template is modified from the COLM 2024 paper template. Instructions are written by Liwei Jiang, Alisa Liu, and Yegor Kuznetsov.