Is Plausibility All You Need?
Modeling Semantic Plausibility and Beyond

Chih-Yi Lin, Quy Nguyen, Wen Wen
{st180953, st181068, st186079}@stud.uni-stuttgart.de

Abstract

In this report, we explore various methods, including machine learning approaches, RoBERTa with prompting and fine-tuning approaches and a large language model Llama 2, to evaluate semantic plausibility across three distinct datasets: PEP, PAP, and ADEPT. We performed comprehensive evaluations and conducted in-depth analysis of results for each model, examining various facets to gain a thorough understanding of their performance. From the results, we saw that fine-tuning pre-trained language model, RoBERTa, performed the best in addressing semantic plausibility tasks. Overall, our discoveries highlight the importance of using advanced NLP methods to handle complex language tasks, such as semantic plausibility, effectively.

1 Introduction

The ability to distinguish between plausible and implausible events is crucial in natural language understanding tasks; however, it is often challenging for models to make accurate predictions. To address this challenge, researchers have developed datasets and models aimed at evaluating semantic plausibility.

In this report, we focus on evaluating the effectiveness of both machine learning paradigms and fine-tuning pre-trained language model (PLM) paradigms on three distinct datasets: PEP (Wang et al., 2018), PAP (Eichel and Schulte Im Walde, 2023) and ADEPT (Emami et al., 2021).

2 Goal Definition

Our study compares various approaches to gain a comprehensive understanding of the task, including:

1. Random forest and decision tree
2. RoBERTa fine-tuning and prompt learning
3. Llama fine-tuning

We aim to identify which approach yields the most effective results across the PAP, PEP, and ADEPT datasets, providing insights into the strengths and limitations of each approach in addressing the semantic plausibility assessment task. Details of the setup of each approach are described in Section 4. Experimental results and analysis are described in Section 5, and conclusions drawn in Section 6.

3 Description of Code Execution

Please refer to the ReadMe file and individual scripts in our shared **folder**[1] to execute our code.

4 Experimental Setup

4.1 Datasets

4.1.1 PEP

The PEP dataset employs a clear Subject-Verb-Object (SVO) structure, excluding modifiers. With 584 terms repeating significantly among the 9186 words, the focus lies on concrete words, addressing physical semantic plausibility. The dataset achieves perfect balance, presenting an equal distribution of plausible and implausible events. Moreover, the introduction of new terms in the test dataset is minimal (1.71%).

4.1.2 PAP

PAP is a collection of 1733 SVO triples annotated for semantic plausibility by human raters. The dataset contains 27 different combination of abstractness levels for the SVO constituents, based on an external source of concreteness ratings (Brysbaert et al., 2014). Our experiments use binary labels only. Importantly, the dataset is skewed towards *Plausible* class. There are more 500 triples that have not been attested from the corpus (i.e., originally *pseudo-implausbile*), but ended up being rated as *Plausible*.

[1] https://shorturl.at/cmQS9

4.1.3 ADEPT

Consisting of 16 thousand sentence pairs, where the second sentence differs from the first one only by adding an extra adjective before a noun. There are five labels denoting how plausibility changes comparing the modifying sentence with the original sentence, namely, *Impossible, Less likely, Equally likely, More likely, Necessarily true*. The dataset is biased towards *Equally likely*, with 67% of instances.

4.2 Models

4.2.1 Machine Learning Approaches

We employed two machine learning models: Random Forest with Sentence Embedding and Decision Tree with Bag of Words. Both models underwent hyperparameter tuning and were applied to three distinct datasets: PAP, PEP, and ADEPT. Notably, for the ADEPT dataset, we opted for the [modifier, noun] format due to stop words being excluded and structural similarity to the other two datasets.

In the Random Forest Model, we initiate data preprocessing by loading train, validation, and test sets. Text is tokenized for subsequent feature extraction. Utilizing Word2Vec models, we capture semantic relationships, producing sentence vectors that encapsulate overall meaning. These vectors serve as features for both regular training and hyperparameter tuning with GridSearchCV. For PEP data, landmark annotations are integrated into features using binning and OneHotEncoder. Features, along with sentence embeddings, are applied to the Random Forest Model for predictions and evaluations.

We also introduce a Decision Tree model using the Bag of Words approach, representing sentences as word bags. CountVectorizer captures word frequency for semantic plausibility prediction. Similar to Random Forest, the Decision Tree undergoes data preprocessing and training with and without hyperparameter tuning.

4.2.2 BERT-Based Models

We choose RoBERTa as it performed the best in the literature. Our aim is to compare the performance of fine-tuning and prompt-based learning on RoBERTa.

For prompt-based learning, we utilized the OpenPrompt API[2], for which a prompt template and a verbalizer (mapping from labels to target words) must be defined. Preliminary experiments compared various combinations of manually defined or soft templates and verbalizers. The results showed that the soft template along with the soft verbalizer performs the best, and therefore, we used this setting for the experiments. The soft template is defined as:

```
"placeholder":"text_a"
"soft" "soft" "soft"
"placeholder":"text_b"
"soft" "soft" "soft"
"soft" "soft" "soft"
"mask".
```

where the soft tokens are trainable tokens, and the mask token represents the model's prediction. We aim to compare the model performance among three prompt-based methods. The first one is zero-shot inference, which only tunes the prompt parameters (soft tokens and soft verbalizer) while keeping the PLM frozen. The second one is few-shot prompt learning (10 epochs), which tunes the prompt parameters and the PLM with 16 samples for each class. The third one is using the full data with prompt learning (3 epochs).

4.2.3 Generative Approach with LLMs

We utilize Large Language Models (LLMs) for a generative approach to model semantic plausibility. Specifically, we use Llama2-7B-Chat model, which was optimized for dialogues between human and a virtual assistant. The key to improve the performance of Llama-Chat is fine-tuning it on a high quality conversation dataset. Due to limitation of computing resources, we use a parameter-efficient fine-tuning technique, namely QLoRA (Dettmers et al., 2023). This technique involves quantizing the model's parameters to reduce memory usage and computational overhead while also introducing trainable Low-Rank Adapter layers.

Experiment 1: Fine-tuning using PAP

To utilize Llama-Chat for text classification task, we first have to transform each instance into the conversational format similar to training data of the LLM. We first verbalize the PAP dataset example (i.e., from numerical labels {0,1} to textual labels {*Plausible*, *Implausible*}), which are used to train the model to generate appropriate continuation for a given preceding text.

Experiment 2: Fine-tuning using PAP-explainer

[2] https://github.com/thunlp/OpenPrompt

In this experiment, the LLM is fine-tuned using the PAP-Explainer dataset. The PAP-Explainer dataset includes additional explanations or justifications for the plausible or implausible labels in the PAP dataset. The fine-tuning process involves incorporating the explanations from the PAP-Explainer dataset into the training procedure, enabling the model to generate more informative and coherent responses or explanations for given prompts.

To this end, we augmented the PAP dataset with justifications for each event-label pair to obtain a new conversational dataset, named *PAP-Explainer*.

- **Step 1: Select Seeding Examples** Choose one typical example from each abstractness combination present in the PAP dataset. For example, *c-c-c* represents a triple with three concrete words. For each combination, we select 1 plausible and 1 implausible event to serve as seeding examples.

- **Step 2: Generate Explanation for seeding examples** For each seeding example chosen in the previous step, we utilize Llama 2 to generate justification for each ground truth label. For example, *cat eats strawberries* and *grape drinks church* are two seeding examples for the *c-c-c* abstractness combination.

- **Step 3: Few-Shot Prompting with Seeding Examples** For each training instance, we provide the LLM with 2 seeding examples (1 positive and 1 negative) of the same abstractness combination to guide its responses.

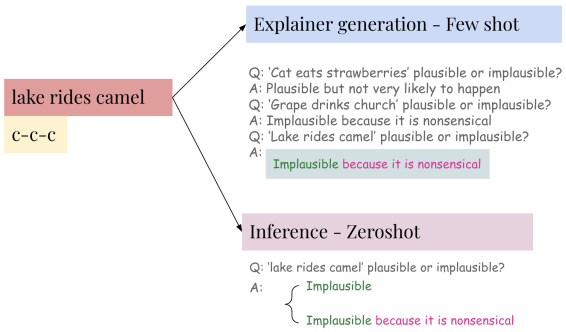

Figure 1: Workflow for using Llama-Chat to generate prediction and explanation

5 Results and Analysis

5.1 Machine Learning Approaches

Due to the stochastic elements in the models, such as random tree selection or random root choice, we adopted a strategy of averaging results over all runs for robust conclusions, here we have runned for three times, seen as in Table 1.

Notably, we observed significant improvement after combining landmark annotations in the PEP dataset, along with overall enhancement after hyperparameter tuning. However, a specific instance in PEP showed a drop in scores post-tuning as the substantial decrease outweighed two minor upticks. To address variability, we suggest increasing the number of runs and calculating the average or excluding the highest and lowest values before calculating the average.

	PAP	PEP	PEP+BIN	ADEPT
	ACC	ACC	ACC	ACC
RF	0.702	0.602	0.746	0.693
RF+Tuning	0.713	0.590	0.759	0.706
DT	0.708	0.577	0.774	0.703
DT+Tuning	0.708	0.679	0.768	0.700
	AUC	AUC	AUC	AUC
RF	0.550	0.615	0.746	0.612
RF+Tuning	0.558	0.607	0.759	0.589
DT	0.497	0.578	0.774	0.547
DT+Tuning	0.497	0.491	0.768	0.582

Table 1: Machine Learning result in different datasets

Additionally, in the PAP dataset, a high accuracy score but low AUC score was noted, indicative of an imbalanced dataset potentially biased towards predicting the majority class. To deal with this, we can improve the model by adjusting class weights during model training to give more importance to the minority class in future model improvements.

5.2 RoBERTa Prompt Learning vs. Fine-Tuning

The experiment results primarily evaluate ADEPT. As shown in Table 2, few-shot prompt learning significantly improves accuracy compared to zero-shot, and it trains much faster than full data (6 minutes for 10 epochs vs. 36 minutes for 3 epochs) while achieving favorable AUC, demonstrating the potential of data-efficient prompt learning. Full-data prompt learning outperforms the few-shot setting; however, it is not superior to full-data fine-tuning.

	Acc.	AUC
Zero-Shot Prompt Inference	0.1203	–
Few-Shot Prompt Learning	0.5676	0.6910
Full-Data Prompt Learning	0.7066	0.7059
Full-Data Fine-Tuning	**0.7295**	**0.7243**

Table 2: Results of three prompt learning methods and fine-tuning.

Upon further examination of the confusion matrices (see Appendix A) for both full-data prompt learning and fine-tuning, we observed that the most common error for both approaches is the misclassification of examples with other labels as *equally likely* (class 2). This reflects the label distribution, with over 60% of examples belonging to this class in the dataset.

For the prompt learning model, it fails to predict any examples from the classes *more likely* (class 3) and *necessarily true* (class 4), which are the least represented classes in the dataset. Conversely, the fine-tuning model performs poorly on *necessarily true* but exhibits better performance on the classes *impossible* (0), *less likely* (1), and *more likely* (3).

Several optimization techniques for our prompt learning can be considered. First, providing example instances and answers in the prompt may clarify the task for the model, especially for PLMs like RoBERTa, which have a smaller size, with only 125M parameters. Second, techniques for searching optimal hyperparameters and prompt templates can be applied. One drawback of utilizing soft templates and soft verbalizers is their lack of interpretability.

5.3 Llama 2 Fine-Tuning

	Precision	Recall	Acc.	F1
PAP	0.674	0.250	0.379	0.365
PAP-explainer	**0.755**	**0.621**	**0.586**	**0.681**
PEP	0.530	0.516	0.531	0.523

Table 3: Evaluation of Llama-2-Chat on PAP/PEP datasets

Experiment 1 with PAP: The PAP dataset demonstrates high precision but low recall. This indicates that while the model correctly identifies many plausible examples, it misses a significant portion of them. The mapping function used for predictions relies solely on the first token of the response. That is, only when the first token of the response is *Plausible*, then we map the prediction to label **1**. Even when the model generate some arguably relevant emojis, the strict mapping function lead to a large amount of false negative prediction.

Experiment 2 with PAP-Explainer: The introduction of the PAP-explainer dataset shows a significant improvement in performance compared to the transformed PAP dataset. This improvement is evident in the F1 score[3], which increases by 0.316, indicating better balance between precision and recall.

Interestingly, the majority of problematic cases for the model fine-tuned with PAP-Explainer are **false negatives**. In numerous cases, our justifications were in alignment with the explanations provided by the model. For instance, in the case of *law needs certificate*: The model incorrectly labeled this as *implausible*, although the provided explanation offers acceptable reasoning for implausibility. Furthermore, there were 14 events that were initially labeled as *pseudo-implausible* but were later annotated as *plausbile* by annotators. Examples of these include *Gravestone manages butterfly* and *Motorway forbids distribution*. These instances highlight the model's ability to provide reasonable justifications for its predictions, despite occasional errors. This discrepancy also points to potential limitations in the dataset's annotations, where certain plausible scenarios might not have been adequately represented or understood.

Cross-Domain Performance on PEP: Fine-tuning the model with the PAP dataset and testing it on the PEP dataset, which represents a cross-domain setting, yields slightly better results compared to random guessing. This suggests that the model's performance generalizes reasonably well across different domains or datasets.

In summary, the performance analysis highlights the effectiveness of the PAP-explainer approach in enhancing the Llama 2 model's ability to identify and explain plausible scenarios, particularly when compared to the original PAP test. Moreover, the model shows promising results in a cross-domain setting, indicating its potential applicability across different datasets or domains.

5.4 Model Comparison

The models are compared as in Table 4. Not surprisingly, RoBERTa fine-tuning outperforms all other models, including machine learning approaches,

[3]Due to the nature of the generative approach, we use F1 to evaluate performance rather than AUC score.

	PEP	PAP	ADEPT
RF+SE	0.746	0.550	0.612
RF+SE-t	0.759	0.558	0.589
DT+BOW	0.774	0.497	0.547
DT+BOW-t	0.768	0.497	0.582
RoBERTa-Ft	**0.865**	**0.775**	**0.724**
RoBERTa-Pt	–	–	0.706
Llama-Ft-F1	0.523	**0.681**	–

Table 4: Model performance comparison. Except for Llama, which was evaluated with F1 score, all models are compared in AUC scores.

the prompt-learning approach and even LLMs.

We also observe a significant enhancement when incorporating landmark features through machine learning approaches. Random Forest and Decision Tree with hyper parameter tuning performs reasonably well on PEP and ADEPT, but not on PAP. It is evident that machine learning approaches require increased efforts when dealing with unbalanced datasets.

Prompt-learning demonstrates the potential of data-efficient learning, but requires optimal hyperparameters and prompt templates.

Llama fine-tuning shows significant improvement on PAP-Explainer with reasonable explanations, but performs poorly on PEP in a cross-domain setting.

6 Conclusion

In this project, we attempted to utilize various methodologies, including machine learning and BERT-based models, alongside large language models, to model semantic plausibility and beyond. Fine-tuning pre-trained language models emerged as a superior approach compared to traditional machine learning methods across datasets such as PEP, PAP, and ADEPT. We also demonstrated the potential of data-efficient prompt-learning, enabling adaptation of pre-trained language models with minimal examples and additional context. Finally, we leveraged Llama 2, a large language model from Meta, in generating reasonable explanations for plausibility judgments, further enhanced through fine-tuning with QLoRA. This project underscores the importance of leveraging advanced natural language processing techniques and innovative fine-tuning strategies to address complex linguistic tasks with precision and efficiency.

Contribution Details

- Code Implementation
  - Chih-Yi Lin: RoBERTa prompting and fine-tuning (ADEPT), RoBERTa fine-tuning (PEP)
  - Quy Nguyen: Random Forest (PAP), RoBERTa fine-tuning (PAP), Llama fine-tuning
  - Wen Wen: Random Forest (PEP, ADEPT), Decision tree (PEP, PAP, ADEPT), RoBERTa fine-tuning (PEP)
- Presentation
  - Chih-Yi Lin: RoBERTa Fine-Tuning vs. Prompt Learning: ADEPT
  - Quy Nguyen: Llama (PAP & PEP), Model Comparison
  - Wen Wen: Machine Learning Approaches
- Documentation
  - Chih-Yi Lin: ADEPT, RoBERTa fine-tuning and prompt learning, Introduction, Goal definition, Code execution
  - Quy Nguyen: PAP, Llama fine-tuning, Model comparison, Conclusion
  - Wen Wen: PEP, Machine Learning Approaches, Abstract

Use of AI Tools

ChatGPT is used for proofreading.

References

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46:904–911.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Annerose Eichel and Sabine Schulte Im Walde. 2023. A dataset for physical and abstract plausibility and sources of human disagreement. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 31–45, Toronto, Canada. Association for Computational Linguistics.

Ali Emami, Ian Porada, Alexandra Olteanu, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2021. ADEPT: An adjective-dependent plausibility task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7117–7128, Online. Association for Computational Linguistics.

Su Wang, Greg Durrett, and Katrin Erk. 2018. Modeling semantic plausibility by injecting world knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 303–308, New Orleans, Louisiana. Association for Computational Linguistics.

A Confusion Matrices of RoBERTa Prompt Learning and Fine-Tuning