Is Plausibility All You Need?

Modeling Semantic Plausibility and Beyond

Chih-Yi Lin, Quy Nguyen & Wen Wen

Overview

1. Machine Learning Approaches
   - Random Forest
   - Decision Tree
2. BERT-based models
   - RoBERTa Fine-tuning vs. Prompt-learning
3. Generative approach with LLMs
   - Fine-tuning Llama 2 with QLoRa
4. Model Comparison
5. Conclusion

Machine Learning Approaches

Methods

- Random Forest + Sentence embedding (+ Hyper parameters tuning)
- Decision Tree + Bag of words (+ Hyper parameters tuning)

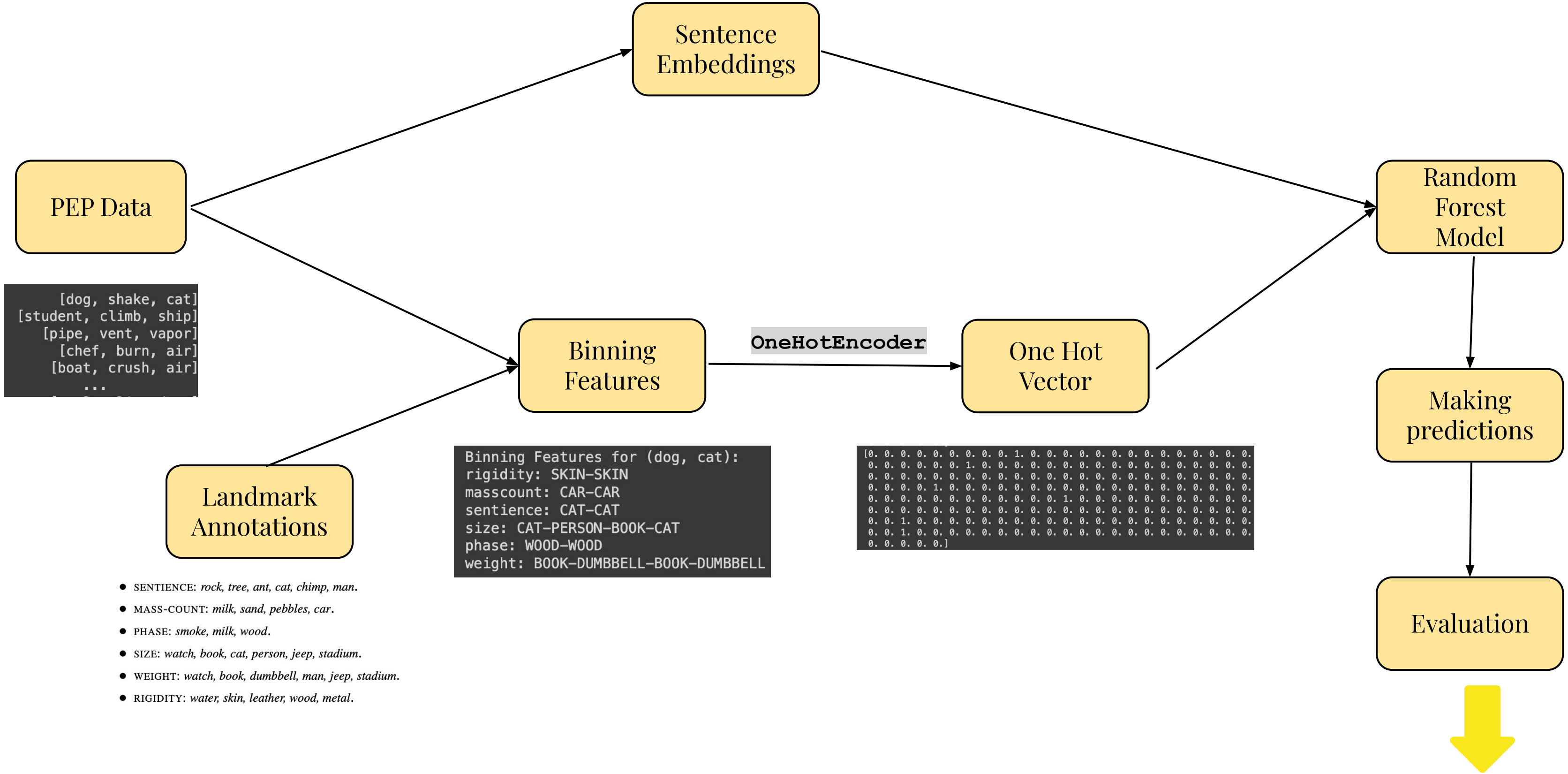
PAP

PEP
+
Landmark
Annotations

ADEPT

[modifier, noun]

PEP+Landmark Annotations



Experimental Results

Average (

Table 1: The 1st Time

	PAP		PEP		PEP+ Landmark		ADEPT	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
RF	0.722	0.555	0.629	0.629	0.746	0.746	0.694	0.614
RF+ Tuning	0.713	0.510	0.557	0.557	0.759	0.759	0.706	0.595
DT	0.708	0.497	0.577	0.578	0.779	0.778	0.703	0.547
DT+ Tuning	0.708	0.497	0.681	0.492	0.769	0.769	0.700	0.582

Table 2: The 2nd Time

	PAP		PEP		PEP+ Landmark		ADEPT	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
RF	0.685	0.514	0.573	0.573	0.746	0.746	0.694	0.601
RF+ Tuning	0.727	0.529	0.593	0.593	0.759	0.759	0.702	0.596
DT	0.708	0.497	0.577	0.578	0.779	0.778	0.703	0.547
DT+ Tuning	0.713	0.500	0.681	0.492	0.762	0.762	0.700	0.582

Table 3: The 3rd Time

	PAP		PEP		PEP+ Landmark		ADEPT	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
RF	0.699	0.580	0.603	0.644	0.746	0.746	0.690	0.622
RF+ Tuning	0.699	0.634	0.619	0.672	0.759	0.759	0.710	0.577
DT	0.708	0.497	0.577	0.578	0.765	0.765	0.703	0.547
DT+ Tuning	0.704	0.494	0.676	0.488	0.772	0.772	0.700	0.582

) =

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

Experimental Results

Average score

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

- **PEP: Greatly improved** after combining landmark annotation
- **Generally, slightly improved** after tuning hyper parameters
  - **But**, there is a case that after tuning, the score actually **dropped...**
- **High Acc score** but **low Auc score.**

Result analysis

- But, there is a case that after tuning, the score actually **dropped...**

PEP Performance

	PEP(1)		PEP(2)		PEP(3)		PEP(Average)	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
RF	$\boxed{0.629}$	0.629	$\boxed{0.573}$	0.573	$\boxed{0.603}$	0.644	0.602	0.615
RF+ Tuning	$\boxed{0.557}$	0.557	$\boxed{0.593}$	0.593	$\boxed{0.619}$	0.672	0.590	0.607
DT	0.577	0.578	0.577	0.578	0.577	0.578	0.577	0.578
DT+ Tuning	0.681	0.492	0.681	0.492	0.676	0.488	0.679	0.491

A substantial decrease outweighed two minor upticks.

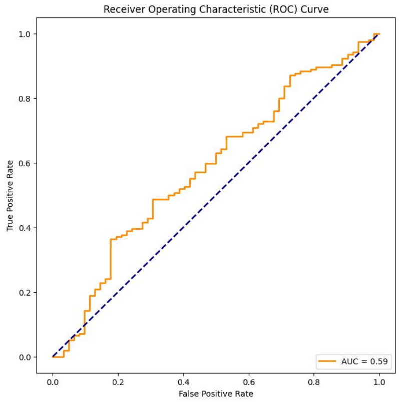
➔ Increase the number of runs and calculate the average.

➔ Exclude the highest and lowest values, then calculate the average.

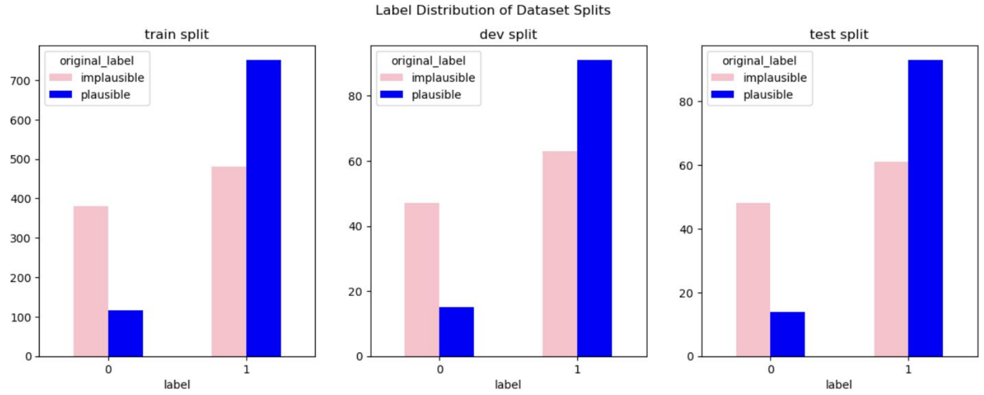
Thanks to @Quy Nguyen for the enlightening PAP data analysis illustration

Result analysis

- **PAP: High Acc score** but **low Auc score.**



ROC curve of PAP data in RF model.
Accuracy: 0.713
AUC: 0.587



An imbalanced dataset appears to be more plausible.

The model might be biased towards predicting the majority class, leading to a high accuracy.

➔ Adjust class weights during model training to give more importance to the minority class.

RoBERTa Fine-Tuning vs. Prompt-Learning: ADEPT

Fine-Tuning vs Prompt-Learning

- **Fine-Tuning**: adapt a pre-trained language model (PLM) on a specific task/dataset
- **Prompt-Learning**: provides PLM with additional context (e.g., instructions, examples) to guide its responses. Objectives:
  - **Reduce required data**: Enable models to adapt on another task with only a few examples (**few-shot learning**)
  - **More Parameter-efficient training**: only train the prompt parameters and keep the PLM frozen
- **RoBERTa-Base**: 125M parameters

Elements in Prompt-Learning (OpenPrompt API)

- **Template**: convert an input text into the instruction
  - **Manual**: `Compared with the statement {"placeholder":"text_a"}, does {"placeholder":"text_b"} become more plausible or less plausible? {"mask"}.`
  - **Soft**: `{"placeholder":"text_a"} {"soft"} {"soft"} {"soft"} {"placeholder":"text_b"} {"soft"} {"soft"} {"soft"} {"soft"} {"soft"} {"soft"} {"mask"}.`
    - `{"soft"}`: trainable
- **Verbalizer**: maps the original class labels to the words that we consider are valid predictions
  - **Manual**: `['impossible'] -> ['impossible', 'no', 'incorrect']`
  - **Soft**

Experimental Design of Prompt-Learning (ADEPT)

1. Zero-shot Inference: Only tune the prompt parameters (soft tokens and verbalizer) while keep the PLM frozen

- Preliminary result of four settings:

manual template + manual verbalizer
manual template + soft verbalizer
soft template + manual verbalizer
soft **template** + soft **verbalizer** 👑

Trainable soft template and soft verbalizer are more **efficient** than manually defined ones

2. Few-shot Prompt Learning (10 epochs): tunes the prompt parameters and the PLM with 16 samples for each class

3. Full-data Prompt Learning (3 epochs)

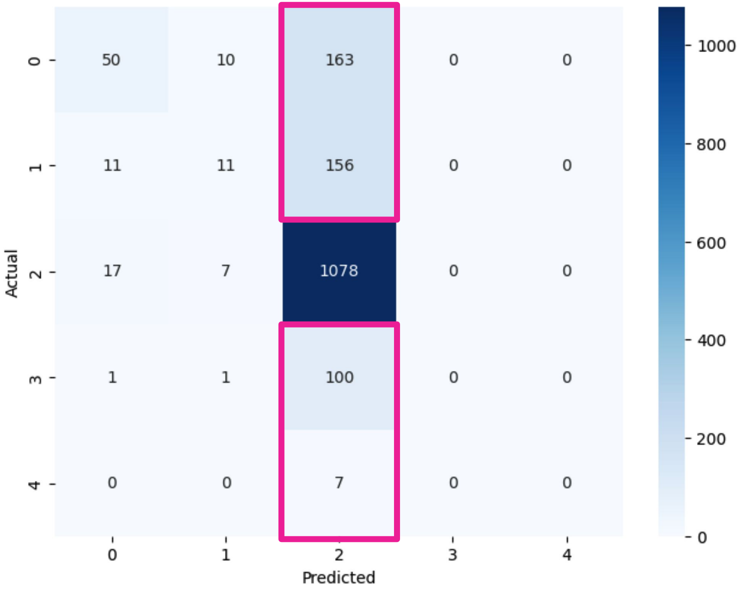
Experimental Results: Prompt-Learning vs. Fine-Tuning (ADEPT, 5 labels)

	Accuracy	AUC	Training Time
Zero-Shot Prompt Inference	0.1203		
Few-Shot Prompt-Learning	0.5676 (+0.4473)	0.6910	6 mins for 10 epochs
Full-Data Prompt-Learning	0.7066 (+0.139)	0.7059 (+0.0149)	36 mins for 3 epochs
Full-Data Fine-Tuning	***0.7295 (+0.0229)**	**0.7243 (+0.0184)**	36 mins for 3 epochs

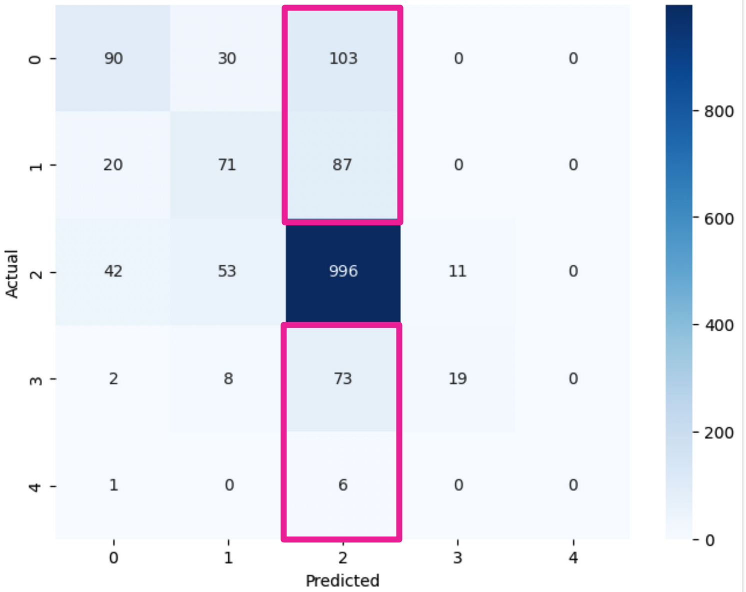
- **Prompt-Learning: 👍 Full-data > 🤨 Few-shot > 😱 Zero-shot**
  - **Few-shot**: drastic improvement in accuracy compare to zero-shot, much faster to train than full-data ⤜ demonstrates the potential of data-efficient prompt-learning
  - **Full-data**: best performance, but longer to train
- **Fine-Tuning > Prompt-Learning**
  - *Fine-Tuning result outperforms the original paper (0.708)

Error Analysis

Prompt-Learning



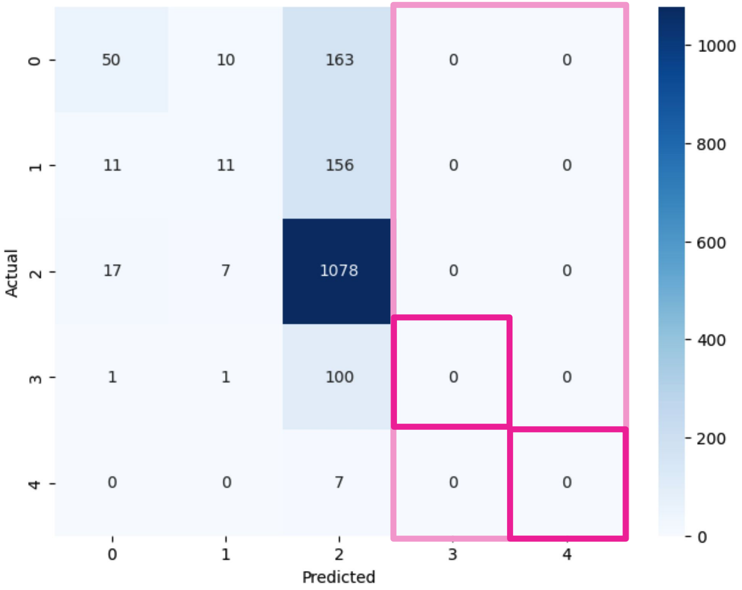
Fine-Tuning



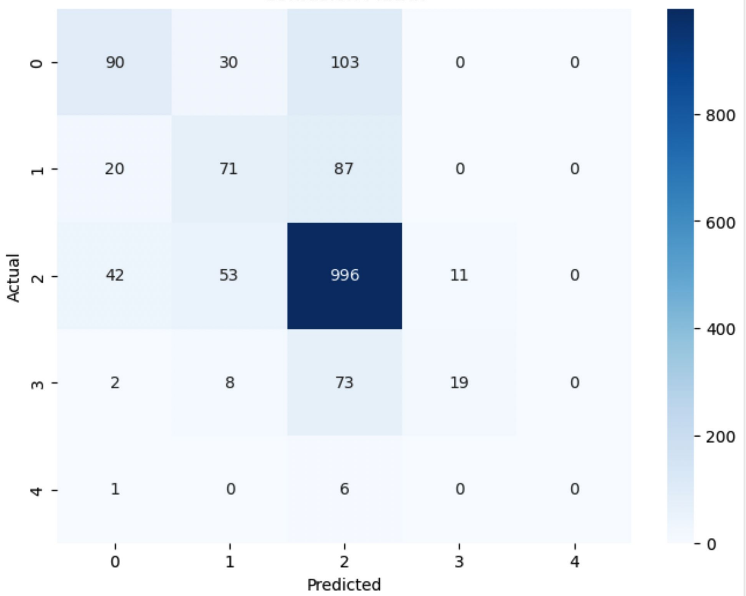
- Most common error in both models: misclassify other examples as **equally likely** (class 2)
  ↣ **align with label distribution** (60%+ of examples belong to class 2 in the dataset)

Error Analysis

Prompt-Learning

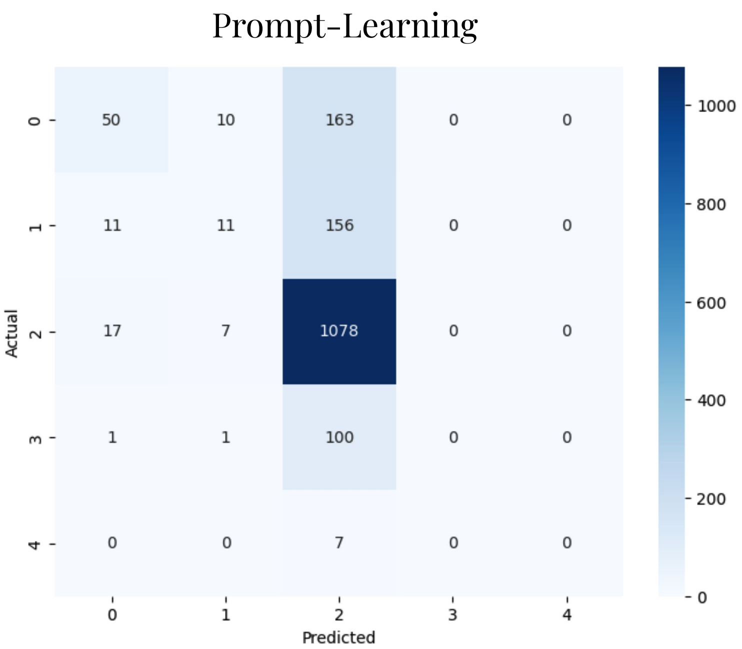


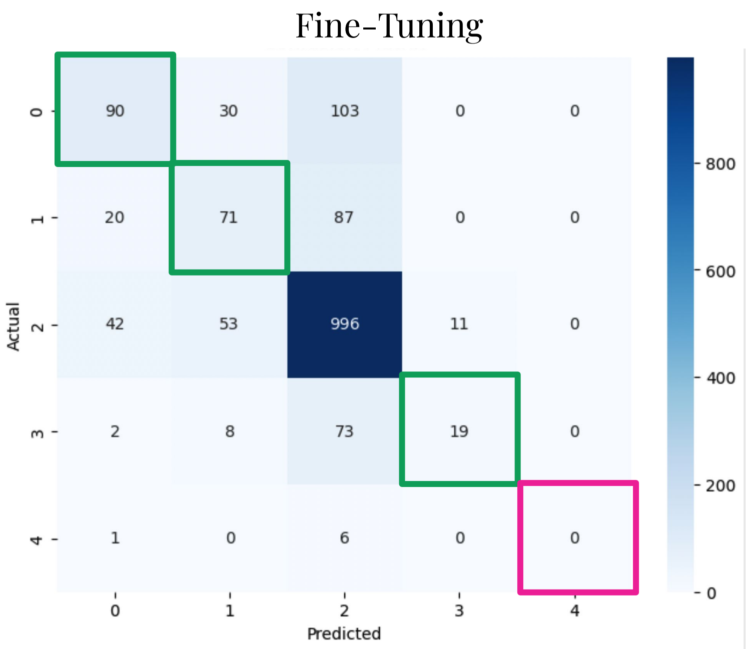
Fine-Tuning



- Prompt-learning model **fails** to predict any examples from the classes **more likely** (3) and **necessarily true** (4) ↣ least classes in the dataset (7%, 1% respectively)

Error Analysis





- Fine-Tuning model
  - performs **better** on the classes **impossible** (0), **less likely** (1) and **more likely** (3)
  - also performs badly on **necessarily true** (4)

Large Language Model:
PAP & PEP

Llama 2 TL;DR

- Open-source LLMs from Meta
- From 7 billion to 70 billion parameters
- **Llama-2-Chat**: optimized for **dialogue**
- Can improve the performance by fine-tuning it on a high-quality **conversational** dataset
- Google Colab Notebook
- 16 GB of VRAM: full fine-tuning is not possible, use parameter-efficient fine-tuning techniques **QLoRA**

Picture generated by DALL-E 3

Steps

1. Transform **PAP** and an augmented dataset called **PAP-explainer**
   a. Choose a typical example of each abstractness combination (e.g. 'c-c-c') as **seeding examples**
   b. Using Llama 2 to **generate explainer** for each combination
      - (1) cat eats strawberries: ***Plausible*** *because strawberries are not toxic to cats and are generally considered safe. However, they are not a necessary part of a cat's diet.*
      - (0) grape drinks church: ***Implausible*** *because it is a nonsensical event.*

   c. **Generate explanation** for each training triplets
      i. Use 2 seeding example's explainers as examples in **few-shot** prompting
      ii. Use Llama 2 to **generate explanation** for PEP-**train** » PAP-explainer training set

2. **Fine-tuning** using QLoRA
   a. Quantize Llama 2 to **4 bits** and **freezes** the parameters
   b. A small number of trainable Low-Rank Adapter layers. **Only adapter parameters are updated** during fine-tuning

Pictures generated by DALL-E 3

Experiment 1: Fine-tuning using PAP

Transform the dataset to fit the training format of the generative model, then **map** responses to labels

event	**label**	**prompt**/**response**	**predict**
group releases album	1	Human: Categorize the following events as plausible or implausible. You should only say either 'Plausible' or 'Implausible'. Event: group releases album Assistant: Plausible!!!	Plausible ⤐ 1
rich unfold interest	0	Human: Categorize the following events as plausible or implausible. You should only say either 'Plausible' or 'Implausible'. Event: rich unfold interest Assistant: Implausible ❌⛔	Implausible ⤐ 0

Experiment 2: Fine-tuning using PAP_Explainer

event	**label**	**prompt/response**	**predict**
group releases album	1	Human: Categorize the following events as plausible or implausible. Event: group releases album Assistant: Plausible because it is a common and typical event. Many groups and bands release albums all the time	Plausible » 1
rich unfold interest	0	Human: Categorize the following events as plausible or implausible. Event: rich unfold interest Assistant: Implausible because rich people may have a wide range of interests, but it is unlikely that their interests would "unfold" in the sense of becoming apparent or noticeable to others.	Implausible » 0

Experiment 2: Error Analysis

event	label (anno.)	original label (corp.)	explainer
gravestone manages butterfly	1	pseudo-implausible	**Implausible** because gravestones are for dead people and butterflies are living creatures. It is not possible for a gravestone to manage a butterfly. ✔
motorway forbids distribution	1	pseudo-implausible	**Implausible** because motorways are not capable of forbidding anything, and they do not have the authority to enforce such a rule. Additionally, it would not be feasible or practical to distribute anything on a motorway. ✔

✔Many *'Plausible'* triples are originally labelled as *'Implausible' (i.e. not attested in corpus)*.

Experiment 2: Error Analysis

event	**label (anno.)**	**original label (corp.)**	**explainer**
law needs certificate	1	plausible	**Implausible** because laws do not typically need certificates. Laws are rules and regulations that are enforced by government agencies and are not typically associated with certificates. It is unlikely that a law would need a certificate.⁉️

⁉️Wrong predictions, but reasonable reasoning.

Performance Analysis

	precision	recall	accuracy	auc
PAP	0.674	0.250	0.379	0.475
PAP-explainer	0.755	0.621	0.586	**0.560**
PEP (cross-domain test)	0.583	0.621	0.590	0.590

- High precision but low recall (most plausible triplets are predicted as implausible)
- Mapping function: **Only** when the **first** token of response is **“Plausible”** ↠ predict **Positive**
- Exact match: event if generate 🤔Plaus, 💯, 100% ↠ predict **Negative** class
- PAP-explainer shows **significant improvement** (+0.114 AUC compared PAP test)
- Cross-domain setting: Fine-tuning with PAP and **test on PEP** gives AUC of **0.590** 🥳

Model Comparison

Model Comparison

	PEP		PAP		ADEPT	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
RF+SE	0.746	0.746	0.702	0.550	0.693	0.612
RF+SE-t	0.759	0.759	0.713	**0.558**	0.706	0.589
DT+BOW	0.774	**0.774**	0.708	0.497	0.703	0.547
DT+BOW-t	0.768	0.768	0.708	0.497	0.700	0.582
RoBERTa-Ft	0.798	**0.865**	0.724	0.538/**0.560***	0.7295	**0.7243**
RoBERTa-Pt	–	–	–	–	0.7066	**0.7059**
Llama-Ft	0.590	0.590	0.557	**0.560**	–	–

**Note: distilled BERT gives significantly better AUC (0.560 vs 0.538) for PAP*

Model Comparison

- **PEP**
  - **RoBERTa fine-tuning** >> **ML** > **Llama cross domain**
    - 🚀Great improvement combining landmark features using ML approaches
    - Llama performs reasonably well in cross domain setting
- **ADEPT:**
  - **RoBERTa fine-tuning** > **RoBERTa prompt-tuning** > **ML**
    - **RoBERTa Prompt-Tuning Optimization**
      - 🤔RoBERTa-base (125M) may be too small for tasks demanding a nuanced comprehension of context
      - 🤔 Search for optimal hyperparameters and prompt templates
      - 😳Soft template and soft verbalizer are lack of interpretability
- **PAP**
  - **Llama PAP-explainer** ~ **ML** ~ **RoBERTa fine-tuning**
    - modeling plausibility with reasonable explanations🗣️

Conclusion

1. Models may be biased towards predicting the **majority class**.
2. In general, **fine-tuning** a PLM outperforms the ML approach.
3. However, a **Random Forest** with hyperparameter tuning still performs reasonably well.

Reference

- RandomForestClassifier
- GridSearchCV
- Classification metrics
- DecisionTreeClassifier
- RobertaForSequenceClassification
- AdamW
- OpenPrompt API
- Llama 2 is here - get it on Hugging Face

Contribution of each member

Model	**PEP**	**PAP**	**ADEPT**
Machine Learning	Wen	Quy, Wen	Wen
RoBERTa Fine-tuning	Wen, Chih-Yi	Quy	Chih-Yi
RoBERTa Prompting	-	-	Chih-Yi
DistilledBERT Fine-tuning	-	Quy	-
Llama Fine-tuning	Quy	Quy	-