ICAI 2023
12th International Conference on Applied Informatics
Eger, Hungary, 2023

Improve Performance of Fine-tuning Language Models with Prompting

Zijian Győző Yang, Noémi Ligeti-Nagy

Hungarian Research Centre for Linguistics
{yang.zijian.gyozo,ligeti-nagy.noemi}@nytud.hu

Abstract

In the context of neural language models, prompts refer to the input text given to the model for generating a response. The prompt serves as a starting point for the model to generate a continuation of the text. The quality of the generated response heavily depends on the quality and specificity of the prompt. A well-crafted prompt can lead to more coherent and relevant responses, while a vague or ambiguous prompt can result in irrelevant or nonsensical outputs.

Prompting strategies in neural language models involve fine-tuning the model on specific tasks, genres or styles, or controlling the diversity of the generated outputs through various techniques like temperature tuning or top-k sampling. These strategies can be used to improve the model's performance on specific use cases and provide more tailored outputs to the user.

In prompt-based learning a language model is used to model the probability of the text directly, as opposed to traditional supervised learning, where a model is trained to take in an input x and predict an output y as $P(y|x)$. In prompt-based learning x is modified into a textual string prompt x' with some unfilled slots. The language model has to fill the unfilled information to form a string $\hat{}x$, from which the final output y can be derived.[2]

This framework is powerful and attractive for a number of reasons: it allows the language model to be pre-trained on massive amounts of raw text, and by defining a new prompting function the model is able to perform few-shot or even zero-shot learning, adapting to new scenarios with few or no labeled data.

In our experiment, we have fine-tuned a huBERT [3] model on various benchmark sets of HuLU [1]. Our research included experimentation on the HuCoPA, HuRTE and HuWNLI datasets. For fine-tuning, we used the same hyperparameter

setting in all cases. All models were fine-tuned in 10 epochs. The highest result scores were chosen in our comparison. We explored various versions of the separator prompt token or text. Below, you will see an example of different prompts:

- Original text from HuCoPA:
  - *premise*: *A sofőr felkapcsolta az autó fényszóróit.* 'The driver turned on the car's headlights.'
  - *choice1*: *Mennydörgést hallott.* 'He/she heard a thunderclap.'
  - *question*: cause
- Prompt using a separator token: *A sofőr felkapcsolta az autó fényszóróit. [SEP] Mennydörgést hallott.* – 'The driver turned on the car's headlights. [SEP] He/she heard a thunderclap.'
- Prompt using text as separator: *A sofőr felkapcsolta az autó fényszóróit.* ***Mert*** *mennydörgést hallott.* – 'The driver turned on the car's headlights. **Because** he/she heard a thunderclap.'

In the case of the HuCoPA dataset, we experimented with 27 different sets of prompts. In Figure 1, the performance of 8 selected prompt sets is depicted.

- [*empty*]: no separator token was used, the premise was concatenated.
- [CLS]: [CLS] separator token was used between premise and choice text.
- [SEP]: [SEP] separator token was used between premise and choice text.
- [SEP] – [CLS]: [SEP] separator token was used for the 'cause type' task and [CLS] separator token for the 'effect type' task. The inverse version was also tested.
- *cause - effect*: question type text was used as a separator token.
- *Ok? - Hatás? / Oka? - Hatása?* 'Cause? - Effect? / Its cause? - Its effect?': Hungarian short question sentence was used as a separator token text.
- *Mert - Ezért* 'Because - Because of this': Hungarian conjunction word was used as a separator token text. In this kind of experiment, to make the sentence grammatical, the 'choice sentence' had to be lowercased.

The best result we achieved was with the *Mert-Ezért* prompt set, where the subordinate conjunctive representing the inference (*cause* in the above example) is inserted (see example above). Typically, the [SEP] token version is used in the literature for fine-tuning on datasets similar to these. As our result suggest, the use of different prompt sets can result in a difference of nearly 8%.

In Figure 1, you can see the highest results with regard to epoch number (see in the text bubble). The use of the [*empty*] and the *conjunction* type prompts required the most epochs to achieve the maximum performance.

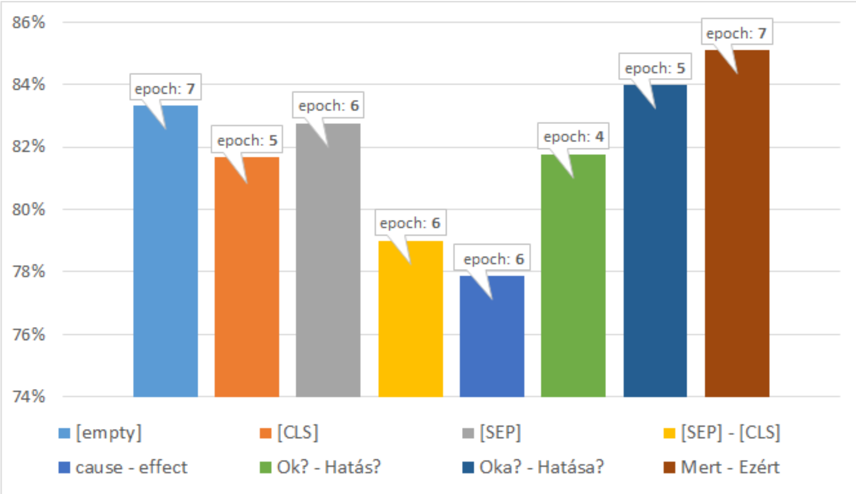

Figure 1. Performance of the model on the HuCoPA dataset with different prompts

References

[1] N. Ligeti-Nagy, G. Ferenczi, E. Héja, K. Jelencsik-Mátyus, L. J. Laki, N. Vadász, Z. G. Yang, T. Váradi: *HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából*, in: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged: Szegedi Tudományegyetem, Informatikai Intézet, 2022, pp. 431–446.

[2] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig: *Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*, ACM Comput. Surv. 55.9 (Jan. 2023), issn: 0360-0300, doi: 10.1145/3560815, url: https://doi.org/10.1145/3560815.

[3] D. M. Nemeskey: *Introducing huBERT*, in: XVII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Magyarország: Szegedi Tudományegyetem, Informatikai Intézet, 2021, pp. 3–14.