

Navarna - Hindi LLM with in-built sentence retrieval task capability

Navarna 7B is an OpenHermes2.5 (Mistral v0.1 series) fine-tune. Built to be good in Hindi chat while adding sentence retrieval (RAG) tasks capability without losing much of its original performance.

Inspired by the e5 mistral LLM release from Microsoft that topped the MTEB board, we wanted to build something similar to that but for an Indic language.

@andersonbcdefg on twitter generated synthetic triplets for sentence retrieval tasks and @AashaySachdeva translated the triplets dataset to Hindi using BARD api.

Dataset links -

English: https://huggingface.co/datasets/andersonbcdefg/synthetic_tuples_gpt35_turbo

Hindi: <https://huggingface.co/datasets/aashay96/translated-dataset-synthetic-retrieval-tasks>

Microsoft did not release any official fine-tuning code for this so I found an alternative - <https://github.com/kamalkraj/e5-mistral-7b-instruct>, due to an issue with the lora adapters generated from this repo I had to abandon the idea of fine-tuning LLM as an embedding lora adapter. I didn't want to do it in any other way than lora because I wanted to experiment with the idea of a LLM used for answering the QA + context pair while the lora adapter was specialised for sentence retrieval tasks. With this idea, it was acceptable to me that we can justify having an LLM with 7B params fine-tuned for sentence retrieval.

Alternate strategy for fine-tuning LLMs for sentence retrieval

After observing the triplets of query, pos, neg in sentence retrieval dataset, i started wondering if it would be effective for me to try query as instruction, pos and chosen data and neg as rejected data pair and do a DPO over it.

In my past experiments with DPO, I observed how quickly it hacks the reward and with my multilingual dataset, it was happening even faster. I found out that working with SFT + DPO combo was better than purely trying to do a DPO.

I split the dataset into 90% and 10%, used the 90% dataset with just query and pos as instruction-response pair and then DPO over the 10% dataset. The results were quite good.

Dataset links -

SFT: https://huggingface.co/datasets/TokenBender/Hindi_SFT_sentence_retriever_set

DPO: https://huggingface.co/datasets/TokenBender/e5_FT_sentence_retrieval_task_Hindi_mini

As a result, I found that the final model could be used as a lora adapter purely with SFT or we

can stack the adapters. It allows us to not worry about relying on another LLM for question answering and instead uses the same base LLM for this task.

The notebooks for this task should be uploaded in the hugging face repo for the model itself.

Notebook link -

https://huggingface.co/TokenBender/navaran_hindi_dpo_merged/tree/main/notebooks

Selecting base model for fine-tuning

Several models were tried for this as base, the criteria was the model with seemingly best performance in the Hindi/Hinglish queries. Mistral base v0.1, v0.2 and Mistral FT optimized and OpenHermes 2.5 were tried.

Due to chatml compatibility and overall Hindi/Hinglish demonstration, I finalised OpenHermes as the choice of the base model.

Challenges

Many challenges exist, especially when it comes to the field of Indic/low-resource languages.

- Multi-turn coherence,
- Hallucinations,
- Instruction following

We will need to evaluate and improve each Indic LLM/low resource language model on these items irrespective of existing benchmarks and ensure improvements with each version.