SACH / सच

SOURCING ACCURATE CITATIONS without HALLUCINATIONS

Automating legal research efficienct and justice

By - Apoorva Gupta

Abstract

The path to justice is often time-consuming and a major problem statement for legal systems worldwide. This project is a demo to automate some parts like legal reading and case understanding to get required citations for creating case briefings and argumentation. This will also reduce time consumption by leveraging the power of Retrieval-Augmented Generation over the responses by ChatGPT and other LLMs

Introduction

Legal procedures around the globe take years to solve cases. It not only involves multiple rounds of hearings, but bulk of paperwork, scrutiny, and readings from past cases. There are more than 80,000 pending cases alone in the Supreme Court of India which with the present strike rate will take many years to resolve with no new cases registered which is impossible. This indicates that the workload on the legal system will keep on increasing leading to more time consumption in the eventual delivery of justice and as it is said justice delayed is justice denied it will lead to wider dissatisfaction among common people.

Basis my conversation with a lawyer practicing in the Supreme of India, I gained the insight that particular case-related research work takes ~30-40% of the total case time for the lawyers, which mandates extensions in dates for cases to be heard. If we reduce this time consumption the eventual impact can have compounding effects.

Following up on why existing LLMs are not a good fit for problem-solving, it was brought up that LLMs generalize the case study and they provide very standard outputs that lack context and judicial interpretation. Moreover, hallucinations just digress from the entire case under study and move toward the mundane assertions.

In this work, I propose a demo Retrieval-Augmented Generation(RAG) intervention to reduce time consumption in case-specific research. This RAG can act as a filter over the responses by ChatGPT and other LLMs and pick only the original citations, with higher semantic matching without hallucinations[4].

This work is different in two ways:

1. Most of the work done in the field of legal LLMs is based on developing legal reasoning in models alongside attempting to make them hallucinate less, with this model we try to remove this hallucination issue by a simple supervised learning method.
2. This work is not generating anything, the given inputs will be embedded in the trained embedding space to decipher the relationship between the given input case-citation pairing. A smaller used case for someone working on a case will then also be the basis for the historical judgments is the citation mentioned for a case right or wrong.

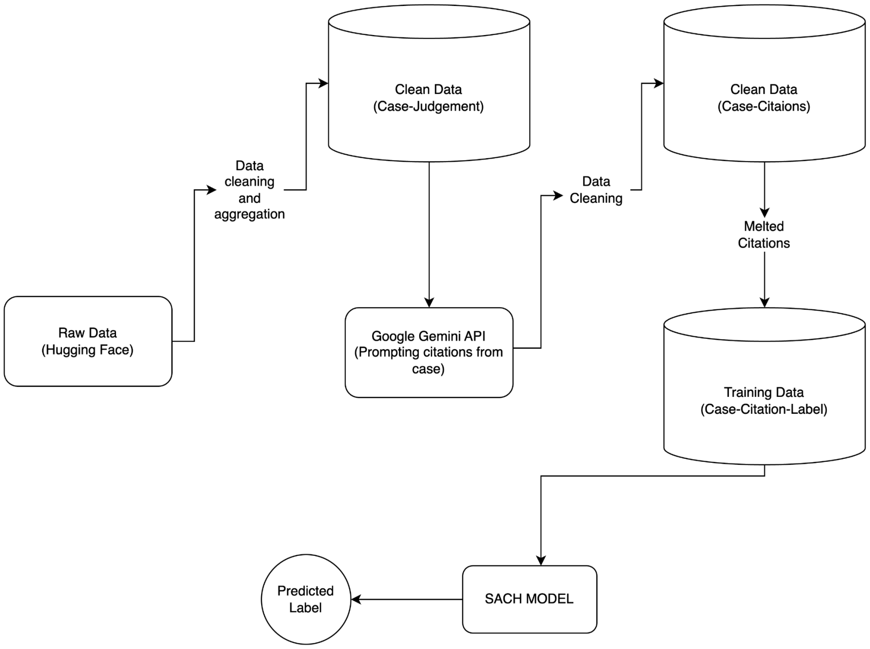
Related works

1. The way to improve the more user-centric output for legal interventions is the most important aspect as the law is not one size fits all, it works with context and similar laws applied to similar situations might lead to different outcomes based on interpretation and context. It is the most challenging aspect of the modern-day judiciary. To overcome this Joel et al[1] devised instruction tuning. It enhances language models for direct user interaction, but most legal tasks are still beyond the reach of open LLMs due to a lack of large-scale instruction datasets. To resolve this, they developed LawInstruct, a comprehensive legal instruction dataset with 12M examples spanning 17 jurisdictions and 24 languages
2. Another issue related to legal research is the violation of law in the existing text data. This is subjected to interpretation again as legal frameworks often overlap due to the myriad of lawbooks and rules with multiple views from multiple angles. What interpretation fits best in the context therefore becomes imperative. Dor Bernsohn et al [2] discuss this in detail.

Datasets

1. S1 - https://huggingface.co/datasets/joelniklaus/legal_case_document_summarization
2. S2 - https://huggingface.co/datasets/Yashaswat/Indian-Legal-Text-ABS?row=1
3. S3 - https://huggingface.co/datasets/Sahil2507/Indian_Legal_Dataset/viewer
4. Final Training Data - Link
5. Test Predictions output (BERT Model) - Link

Methodology

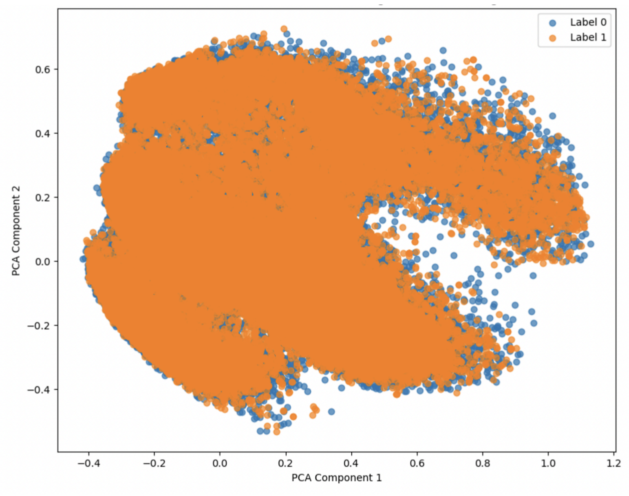


1. Data from 3 Hugging face sources was taken cleaned and filtered for our used case
    a. S1 - columns case and judgment were taken

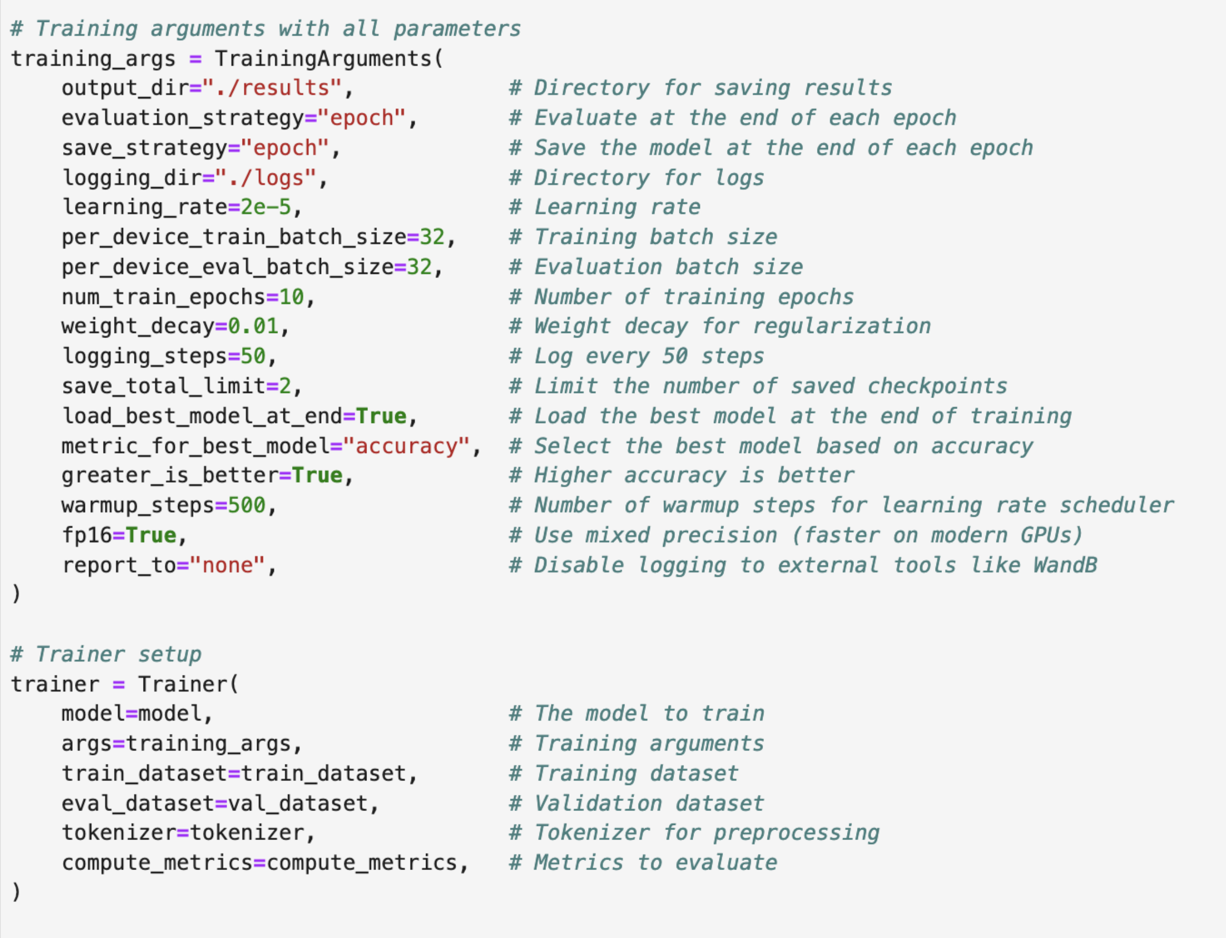
   b. S2 - filtered only for cases from India and then columns case and judgement were taken
   c. S3 - columns prompt, text, instruction, and input were deleted and cases and judgment were taken.
2. After filtering and aggregating all these data points in a single dataset, they were combined in a single column called case.
3. The resultant file is then chunked into 29190 to prevent the max_token overflow on Gemini API while collecting citations.
4. Each batch had 5 chunks for which citations had to be prompted using the prompt and the chunked citation

‘‘*You are given a prompt that takes a chunk from the case judgment given by a judicial court in India. Your*
the task is to read through the entire prompt and give the citations from that judgment those citations can be
1. In case you are putting a section/article like article 15 also put the source like section 15 of the constitution of India.
2. references to past cases
3. do not repeat any citations, give me a unique Python list only
You need to put all these in a Python list and give an output that will look like this [Section 120 B, Indian Penal Code]
if it is not like this just return 0 do not throw any error. Here article 10 is given as an example do not put it in all of your responses’’

5. Due to the limited free prompts allowed on Gemini API, it took me 5 days to collect all the data. I used two variants of the Gemini Model to get the citations.
   a. Gemini Pro
   b. Gemini - Flash 1.5
6. Obtained data had a case and a list of citations about it, so each of the 13,648 cases had a list of citations corresponding to it.
7. A Python script ran on this dataset converted it into case-citation data with each case having one citation and labeling to it. Labelling was
   a. 0 - for a citation not present in the case
   b. 1 - for a citation present in the case
8. The shape of this data is - (112996, 3). Since negative labels were underrepresented in this data, I oversampled them by picking out citations that were not present in the list of a particular case and negatively labeling them randomly to get the augmented data with shape - (199457, 3) and Positive Samples = 99748, Negative Samples = 99709 with unique citations = 65,269. The obtained data can be seen in the embedding space below for the given case citation pair. These embeddings are almost overlapping with each other.



9. This data is now fed to the transformer model with the following parameters in two ways
    a. With Legal Bert Embeddings
    b. With Bert Emdeddings



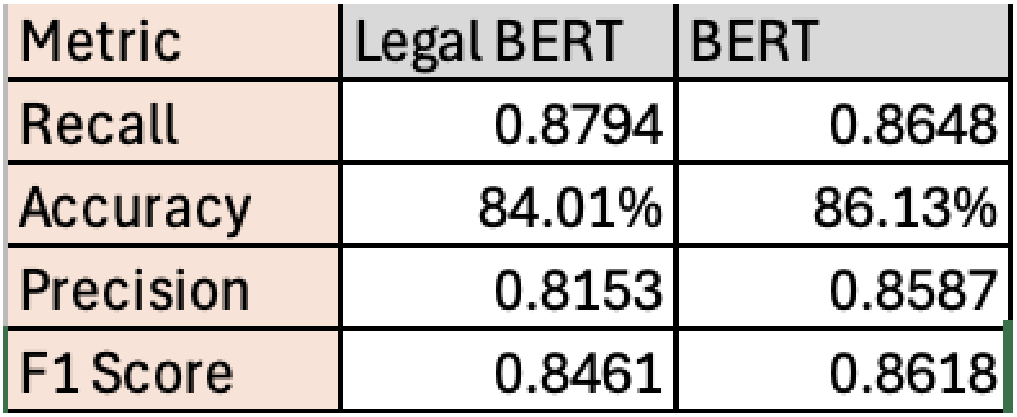
```
# Training arguments with all parameters
training_args = TrainingArguments(
    output_dir="./results",            # Directory for saving results
    evaluation_strategy="epoch",       # Evaluate at the end of each epoch
    save_strategy="epoch",             # Save the model at the end of each epoch
    logging_dir="./logs",              # Directory for logs
    learning_rate=2e-5,                # Learning rate
    per_device_train_batch_size=32,    # Training batch size
    per_device_eval_batch_size=32,     # Evaluation batch size
    num_train_epochs=10,               # Number of training epochs
    weight_decay=0.01,                 # Weight decay for regularization
    logging_steps=50,                  # Log every 50 steps
    save_total_limit=2,                # Limit the number of saved checkpoints
    load_best_model_at_end=True,       # Load the best model at the end of training
    metric_for_best_model="accuracy",  # Select the best model based on accuracy
    greater_is_better=True,            # Higher accuracy is better
    warmup_steps=500,                  # Number of warmup steps for learning rate scheduler
    fp16=True,                         # Use mixed precision (faster on modern GPUs)
    report_to="none",                  # Disable logging to external tools like WandB
)

# Trainer setup
trainer = Trainer(
    model=model,                       # The model to train
    args=training_args,                # Training arguments
    train_dataset=train_dataset,       # Training dataset
    eval_dataset=val_dataset,          # Validation dataset
    tokenizer=tokenizer,               # Tokenizer for preprocessing
    compute_metrics=compute_metrics,   # Metrics to evaluate
)
```

10. The results obtained are in the next section

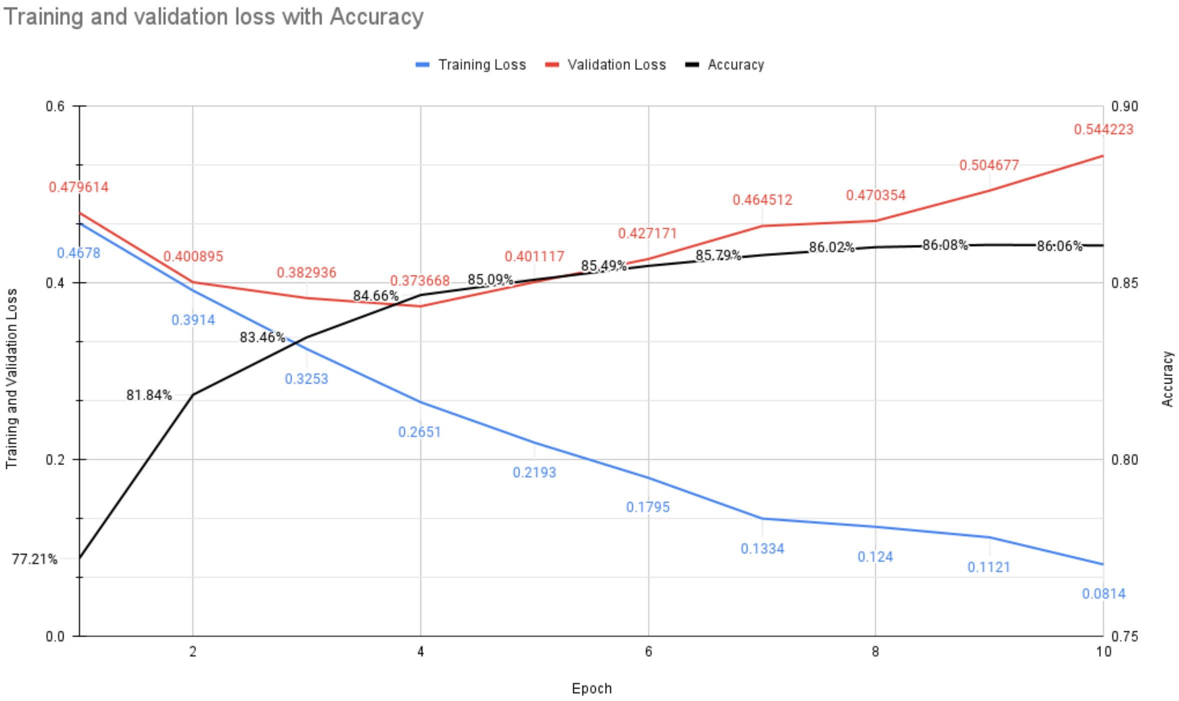
Evaluation

1. Basis the results shown in the table below are for the metrics on testing data. I used BERT results as they gave more balanced results



Metric	Legal BERT	BERT
Recall	0.8794	0.8648
Accuracy	84.01%	86.13%
Precision	0.8153	0.8587
F1 Score	0.8461	0.8618

2. As the BERT performed the best, here is a snapshot of its training process. The black line of training accuracy steadily increased till epoch 4 and then started consolidating around 86%. Maximum accuracy is obtained at the 9th epoch, and from there, the gap between training loss and validation loss also widened rapidly (red and blue lines).



Conclusion

1. With 86% accuracy, this model is performing decently well to give the right direction. I obtained this accuracy after doing hyperparameter tuning, however data with better quality can improve the chances of accuracy.
2. This work also enables us to understand that supervised learning models for specific used cases can enhance the usability of LLM applications in different domains like the legal industry.
3. This model further has the following applications
    a. Delivering accurate citations for the case research to a lawyer or a legal practitioner.
    b. As a RAG over responses by state-of-the-art LLMs to prevent hallucinated results.
4. With following benefits
    a. Time of judiciary will be saved with less time required in research
    b. Brief accuracy and generalizability will improve, leading to more overarching judgements

References

1. Niklaus, Joel, et al. "FLawN-T5: An Empirical Examination of Effective Instruction-Tuning Data Mixtures for Legal Reasoning." arXiv preprint arXiv:2404.02127 (2024).
2. Bernsohn, D., “LegalLens: Leveraging LLMs for Legal Violation Identification in Unstructured Text”, Art. no. arXiv:2402.04335, 2024. doi:10.48550/arXiv.2402.04335.
3. Guha, Neel, et al. "Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models." Advances in Neural Information Processing Systems 36 (2024).
4. Fan, Wenqi, et al. "A survey on rag meeting llms: Towards retrieval-augmented large language models." Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024.

Video Link -
https://www.canva.com/design/DAGYNjuO8nA/yx3yKaLDD0xvUnmIIHBbqg/view?utm_content=DAGYNjuO8nA&utm_campaign=designshare&utm_medium=link&utm_source=recording_view

GitHub Link -
https://github.com/mr-sesquipedalian/project-sach