司法
導盲犬

專題報告

指導老師:蔡銘峰
成員:陳肇廷、梁栯睿、蘇胤翔、王冠智

Table of contents

01

Abstract

A Judicial dog is all you need

02

Architecture

We use RAG, Fine-tune for the cores of our project

03

Method of Developing

We set experiments and weave the components

04

Demo

Try out what we got

01

Abstract

A Judicial dog is all you need

Purpose statement

Assistance in legal advisory services

- Provide comprehensive legal advices based on the latest legal database
- Provide scenarios to illustrate their applications

Instant right relief.

- Offer an instant and convenience way to exercise your rights
- Provide advice toward law related issues

02

Architecture

The LLM pipeline

The brief architecture

01

RAG

Connect to the world

02

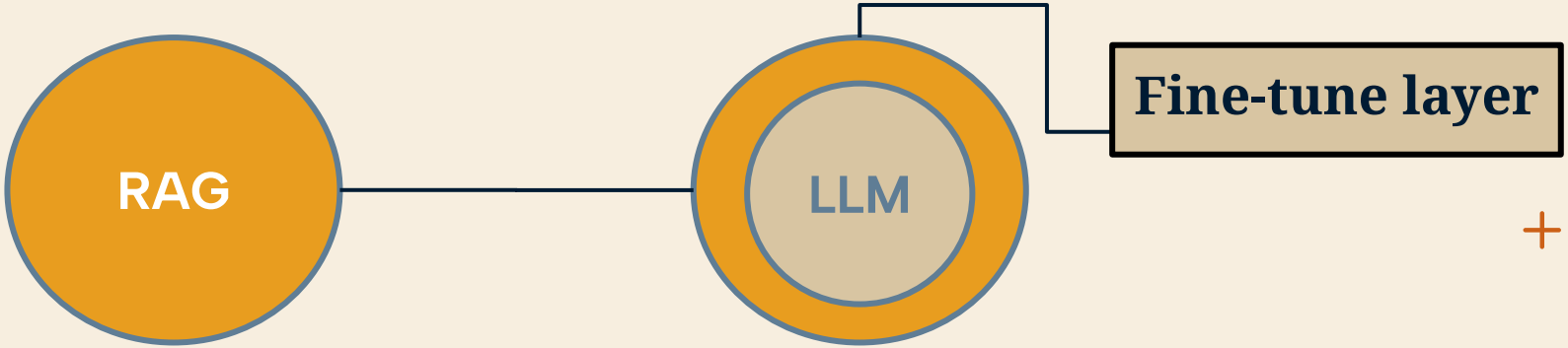
Fine-tune

Specialize the LLM

03

LLM

LLama-3, BERT, GPT



RAG (Retrieveral-Augmented Generation)

Connect and Retrieve

- Divide documents in to chunks
- Vectorize them and store into the **vector database**
- Find top-k relevant chunks and transfer to LLM to get response

1

Law

2024 Revised Edition of Traffic Regulations

2

Judgement

All Traffic Judgements from **2020 to 2024**

3

Terminology

Add Local Terminologies, like 諭知、智識程度...

RAG

Fine tune

A process of creating another layer using the original model by making additional settings on it to enhance the ability of specific tasks

- Modify the parameters
- Train the fine fune model
- Combine the downstreamed model and original model

LLama-3

We modify the parameters of the original LLM and use LLama-factory

BERT

- Token classification model
- Sequence classification model
- Question answering model

03

Method of Developing

Developments

Modern Tools

- For the latest data, we use python and the official judicial api to crawl the judgements
- For the storage, We use Redis as our vector database to store embeddings from source data
- For RAG, we use Llama-index
- And we use FastAPI to build the backend application to use the system easily

Experiments

- We use different LLMs and downstreamed models to see their quality of responses
- We test different retrieval algorithms to retrieve the match contexts accurately
- We test embedding models by seeing cosine similarity between sentences in similar meanings

04

Demo