Foundations of Natural Language Processing

Peking University, 2025

Assignment 4 Project 2:
Multi-Defendant Legal Judgment Prediction

1. Directions

Please first read the **general instructions** of Assignment 4.

If you choose this project,

- Please submit your homework as a zip file through **Course**, which should include one report in PDF and your source code in Python.
- Please include the score you achieved on the leaderboards of two sub-tasks in the report.
- The code should be paired with a README file describing dependencies, code structures, etc.

We will not simply grade your homework based on the model performance, but consider **the models you use**, the **novelty** of your method, the **workload**, and the **analysis** in your report.

If you graduate this summer, given that you have less time to complete the project, we will apply a more relaxed grading scale.

2. Task Description

In this project, you are going to build a Legal Judgment Prediction system for the multi-defendant scenario. Legal Judgment Prediction (LJP) aims to predict judgment outcomes (e.g., law articles, charges) given the fact description of a case.

Sub-Task 1: Charge Prediction

Given the fact description of a case, the aim is to predict charges for each defendant. For each case, predicting the charges against each defendant relies on one or more articles of criminal law. We additionally provide a collection of criminal law articles for reference. Note that a defendant may face multiple charges, you should additionally predict the length of fixed-term imprisonment for each defendant in Sub-Task 2. We have provided a collection of all charges on Kaggle.

Example:

Input:

fact：某县人民检察院指控，2017年6月1日，为控制和操纵全县民间唢呐演出市场，被告人王某A组织部分某城唢呐艺人在某城西门街"晁家馆"酒店开会，成立"某县唢呐学会"非法组织。任命会长、副会长、秘书长等，划分管理，将唢呐艺人吸收为会员，建立唢呐微信群。只要唢呐演出收费低于其规定标准，王某A等人就纠集在一起，多次开车到唢呐艺人演出现场、回家路上辱骂、恐吓、拦截、追逐等方式寻衅滋事，多次在唢呐微信群内有针对性的辱骂、恐吓。形成以王某A为主，刘某A、焦某某、耿某A为成员，韩某某、孙某A和王某9、刘某1（均另案处理）等积极参加的恶势力犯罪集团……

defendants： ["王某A", "刘某A", "韩某某", "孙某A", "焦某某", "耿某A"]

Output:

"寻衅滋事罪;寻衅滋事罪;寻衅滋事罪;寻衅滋事罪;寻衅滋事罪;寻衅滋事罪,伪造公司、企业、事业单位、人民团体印章罪"

In the output, use a semicolon (;) to separate each defendant, and use a comma (,) to separate all charges for a specific defendant. The order of predicted charges should be consistent with the order of defendants in the 'defendants' list.

Sub-Task 2: Penalty Prediction

Given the fact description of a case, on the basis of predicting the charges, you should additionally predict the length of fixed-term imprisonment for each defendant. The unit of imprisonment prediction is months.

The output is a two-dimensional list. The 0th dimension represents each defendant, and the 1st dimension represents the imprisonment lengths for all charges of a particular defendant.

Input:

The same with Subtask 1

Output:

"[[18], [21], [36], [15], [12], [60, 0]]"

Method

There is **no constraint** on the method you use. You can implement your own model from scratch, finetune on (large) language models, or use LLM APIs. If you use APIs, please use the qwen-max API as mentioned in the general instructions. Please clearly describe the method you use in the report.

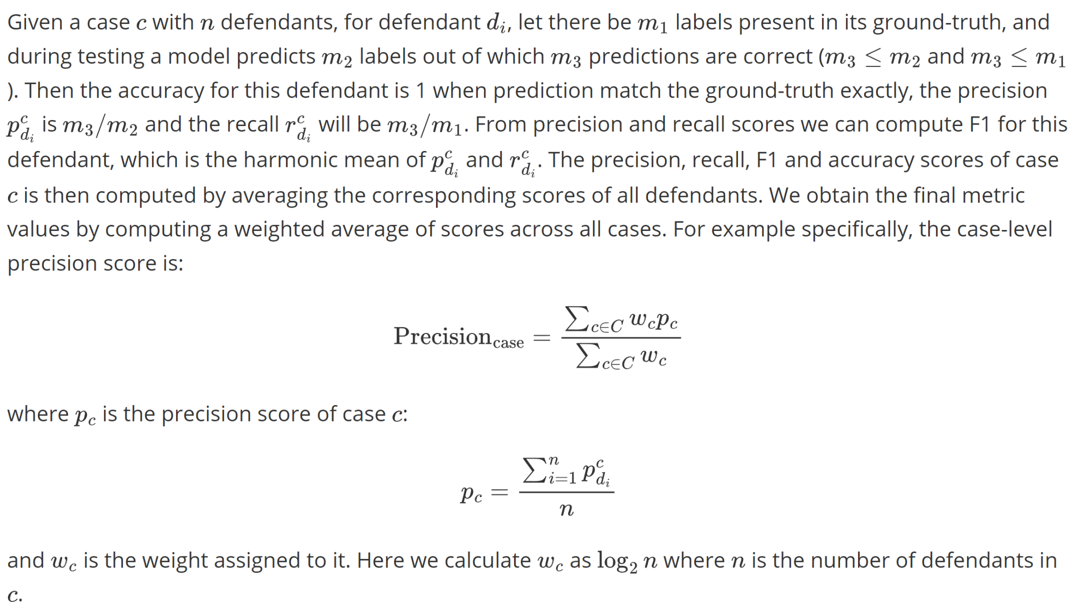
The data for two sub-tasks are posted on two separate Kaggle competitions:

Sub-task 1: https://www.kaggle.com/t/853b5d7dc4eb490e834745d900df9e3d

Sub-Task 2: https://www.kaggle.com/t/92afa88d912a478e8cb70223a4427c2a

Evaluation

We use the case-level f1 score to evaluate the prediction quality. This figure shows the metric for subtask 1, and the metric for subtask 2 is similar to it



Given a case c with n defendants, for defendant d_i, let there be m_1 labels present in its ground-truth, and during testing a model predicts m_2 labels out of which m_3 predictions are correct ($m_3 \leq m_2$ and $m_3 \leq m_1$). Then the accuracy for this defendant is 1 when prediction match the ground-truth exactly, the precision $p^c_{d_i}$ is m_3/m_2 and the recall $r^c_{d_i}$ will be m_3/m_1. From precision and recall scores we can compute F1 for this defendant, which is the harmonic mean of $p^c_{d_i}$ and $r^c_{d_i}$. The precision, recall, F1 and accuracy scores of case c is then computed by averaging the corresponding scores of all defendants. We obtain the final metric values by computing a weighted average of scores across all cases. For example specifically, the case-level precision score is:

$$\text{Precision}_{\text{case}} = \frac{\sum_{c \in C} w_c p_c}{\sum_{c \in C} w_c}$$

where p_c is the precision score of case c:

$$p_c = \frac{\sum_{i=1}^{n} p^c_{d_i}}{n}$$

and w_c is the weight assigned to it. Here we calculate w_c as $\log_2 n$ where n is the number of defendants in c.

We provide the code to calculate metrics for each subtask on Kaggle.

3. Resources

1. Here are some tutorials on prompting LLMs:

   https://www.promptingguide.ai/zh/introduction/basics
   https://learnprompting.org/zh-Hans/docs/basics/intro