Labeling Legal Documents on HKLII

Subtask of Document Recommendation on HKLII

Project Plan

Liu Yuchen

Supervisor:
Prof. Benjamin Kao
1. Introduction

1.1 Background

HKLII, short for Hong Kong Legal Information Institute, serves as the main legal document repository for Hong Kong. More than a thousand legal practitioners use it to retrieve legal documents at work. The search function of the website is extremely useful when users want to find some prior cases as reference. However, since the current website was designed twenty years ago, the logic of the search engine is relatively naïve, which is mainly based on keyword matching. For example, by searching drug selling, the engine will not associate it with drug trafficking in the results. Our project aims to incorporate NLP (natural language processing) technologies in the search function to make the website more responsive than before. My work individually, is to label every court judgement by some tags. These tags are primarily predefined properties, such as the offence type, the severity of the crime, etc. In further phases, the tags will also be chosen from features extracted by topic modeling, which are more appropriate and flexible.

1.2 Problem Statement

Figure 1.1 shows the frontend design for the new HKLII website. On the left-hand column, there are five sections providing basic information and internal navigation function in the article. In the middle, is the main content of the document, it could be a court judgement, regulation, or an ordinance. The right column directs the user to external documents using recommendation algorithm, which will be our focus. This report aims to solve the technical issues in the "Relevant Legal Concepts" section on the right column, where users can choose a combination of tags to find other proper documents.

![Figure 1.1 Proposed Frontend interface of the new HKLII](image-url)
This problem could be subdivided to smaller issues. First, what should be the choice of the tags? Second, how to train the classifier (labeler)? More specifically, what should be the train and test set? Which machine learning method should be used? Finally, how to further optimize the algorithm learning from the user activity and feedback?

1.3 Objective
To tackle all the problems stated in section 1.2, the basic solution should contain a classifier model that can label a court judgement, a database that index all the processed document, and APIs for the backend that can easily retrieve all these information.

A comprehensive performance description for a good solution should be:
1) The tags cover all the essential aspects of a court judgement. The user can easily find a few tags corresponding to his needs.
2) The tags are mostly accurate.
3) For a new court judgement, it is fast to index it into the database.

Upon finished, this function can also benefit the search engine to restrict the result scope. Another function, finding similar cases, can possibly use these tags as features too. Most importantly, this work can provide more interpretability to the search result as a symbol-based approach, which is essential to the field of law [2].

2. Methodology
2.1 Tags Pool
Tags pool is the collection of all the possible tags to choose. It should also consist of multiple clusters. For example, one of the clusters may contain all the offence types, such as murder, burglary, traffic violations, blackmail, etc. Each cluster contains the tags that are mutually exclusive and complementary. Consultations with actual legal professionals and website users are needed in the formulation of this pool. In fact, the structure of this pool will significantly affect the model construction. Therefore, priority should be given to this task.

To describe the document in more details and flexibility, the final tagging may also incorporate the work from other team members working on summarization on topic modeling. In that case, part of the tags pool is fixed, while the remaining will be dynamically generated based on every document.

2.2 Classification Models
Below are just possible methods. In this project plan, the focus is on the empirical potential rather than the actual performance. Therefore, the final solution might not appear as any method in this section.

2.2.1 Multinomial Logistic Regression
This is a statistical model that can predict the possible outcomes of a categorically distributed dependent random variable. As the classical methods to all classification task, it might not be the most suitable methods for a task like this. Finding a significant feature for the regression in the plain text could be complicated unless you use the full text\textsuperscript{[3]}. Any word in the document could be crucial, so there is not a universal processing rule that can apply to all the documents. And due to its limited parameter number, the performance is not comparable to the state-to-art neural network \textsuperscript{[4]}. Therefore, this should be backup method, and will only be used as a baseline.

2.2.2 RNN

RNN has gain considerable application in speech recognition, digital signal processing, video processing and text data analysis \textsuperscript{[3]}. Certainly, it will be suitable for this task. Presumably, different variations of RNN models, such as simple RNN, bidirectional RNN and LSTM, will be experimented based on different context. For example, if we apply the model to the whole document, RNN will probably suffer from the vanishing gradient effect, which says during backpropagation, long-term dependencies suffer from insufficient weight change because of error information vanishing \textsuperscript{[5]}. LSTM could be used to overcome such problem \textsuperscript{[6]}. However, limited by the RNN length, some documents may not be able to fit in one RNN. A mechanism to continue with the last RNN, or to combine the results of consecutive RNNs, is needed in this case.

2.2.3 Transformer

If a method that can possibly solve more issues stated in section 1.2 than any other method exists, it must be the transformer model. The paper "Transformers: State-of-the-Art Natural Language Processing", called it the "model of choice for NLP problems" \textsuperscript{[7]}. By stacking multiple self-attention layers, the transformer can output the result taking every word in the document into consideration, theoretically speaking \textsuperscript{[8]}. The biggest difficulty with the transformer is related to training. There is few pretrained transformer for law documents. If we are training it ourselves, because of the large size of parameters, we will need a large training corpus, which has to be built from scratch.

2.3 Comparison Benchmark

Fairness of comparison among the methods is not about using the same setup and controlling other variables. Instead, we should fine-tune each model to achieve its best. Various popular measurements will be used to qualify the performance. According to this paper \textsuperscript{[9]}, Exact Matching Ratio, Labelling F-score, Retrieval F-score, Hamming Loss and the prediction time are indispensable measurements. We will analyze the performance of each model in terms of these indicator to draw the final conclusion.

3. Schedule and Deliverables

If the time field is empty, that row means the key emphasis in work during the period
starting from the above deadline and ending at the below time.

<table>
<thead>
<tr>
<th>Time</th>
<th>Deliverables or Key Emphasis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 3, 2021</td>
<td>• This project plan&lt;br&gt;• An initial page of the project website</td>
</tr>
<tr>
<td></td>
<td>• Constructing a tag pool&lt;br&gt;• Construct a training dataset&lt;br&gt;• Test one or two methods on the data</td>
</tr>
<tr>
<td>Dec 15, 2021</td>
<td>• First presentation&lt;br&gt;• Detailed intermediate report</td>
</tr>
<tr>
<td></td>
<td>• Test the remaining models&lt;br&gt;• Analyze the results</td>
</tr>
<tr>
<td>March, 2022</td>
<td>• Finish testing all the models&lt;br&gt;• Choose a model that can be used in production</td>
</tr>
<tr>
<td></td>
<td>• Create the new datatable containing the tags for all the documents&lt;br&gt;• Integrate the API with backend team&lt;br&gt;• Work on the requested deliverables</td>
</tr>
<tr>
<td>May, 2022</td>
<td>• Finish applying the model to the whole database&lt;br&gt;• Finish the APIs related to tags&lt;br&gt;• Final report&lt;br&gt;• Final presentation&lt;br&gt;• Project Exhibition and 3-min video</td>
</tr>
</tbody>
</table>

4. References


