Document Recommendation on HKLII
Final Report

Ma Chun Lai (Jonathan)

University Number: 3035605603
Supervisor: Prof. Benjamin Kao

April 18, 2022
Abstract

Legal professionals around the world often reference and use online document libraries such as WorldLII to conduct legal research. In particular, under the legal system in Hong Kong, relevant judicial precedents are taken into consideration in deciding the outcome of the case, highlighting the importance of retrieving historical decisions during legal research. This project studies, implements and evaluates various algorithms and techniques to introduce semantic querying and document summarization to the Hong Kong Legal Information Institute (HKLII), with an aim to facilitating the searching and reading of documents on the system. With results conducted in previous researches, this project focused on using traditional document retrieval methods, topic modelling and transformers techniques to perform retrieval and summarization of legal documents. By introducing both features on HKLII, it is hoped that the system can better understand the intention of users’ queries and retrieve more relevant documents, while also reducing the time needed to recognize key information of the judgement, allowing users to more efficiently identify the relevance of the judgement to their needs. The system is hoped to be eventually rolled into production on the HKLII website, benefiting users of the website and optimizing the performance of the queries. The prototype of the two above-mentioned features were implemented and integrated to the modern HKLII interface, and preliminary evaluations and results are elaborated in the latter sections of the report.
Acknowledgements

I would like to express my gratitude to Professor Benjamin Kao for supervising this project and advising me by providing his knowledge in this area. I also wish to thank Mr Kevin Wu, a PhD student working under Professor Kao, for his preliminary work on applying topic modelling to document recommendation on HKLII, and his assistance in implementation of the prototype on dedicated server. The assistance from Gary Leung, Stephanie Wong, Tze Ching Yu and the HKLII development team in creating a clone of the existing HKLII database and providing the new interface of HKLII is key to the creation of the prototype; their efforts are greatly appreciated. The work is also not possible without the assistance of the legal research team at the Faculty of Law in the University of Hong Kong led by Dr TM Cheung in helping us label the legal documents, which are essential to the topic model created in this project.

I would also like to credit Michael Lam and Ryan Ho, two students who contributed to the current evaluation of the implementations of semantic search and summarization.

Finally, I appreciate the work from Mr Liuchen Yu and Mr Huijie Pan, who work independently but are also contributing to HKLII in their Final Year Projects. I am also grateful for the help from Ms Mable Choi in guiding me to complete the project reports.
## Contents

Abstract i
Acknowledgements ii
List of Figures v
List of Tables v
Abbreviations and Acronyms vi

1 Introduction 1
   1.1 Current status of HKLII ..................................... 1
   1.2 Semantic Search .............................................. 2
   1.3 Document Summarization ................................. 3
      1.3.1 Query-driven Summarization ............................... 3
      1.3.2 Aspect-driven Summarization ............................... 4
   1.4 Objective ..................................................... 5
   1.5 Related work ................................................. 6
   1.6 Outline of the report ...................................... 7

2 Methodology and Algorithms 8
   2.1 Data Collection and Development of Prototype ............... 8
   2.2 Algorithms used in the Project ............................ 8
      2.2.1 Data and Query pre-processing: Named Entity Recognition and Regular
            Expressions ............................................. 8
      2.2.2 Sentence Transformers ...................................... 10
      2.2.3 PageRank ................................................ 11
      2.2.4 BM25 .................................................. 11
      2.2.5 Topic Modelling ........................................ 11
      2.2.6 Alternative algorithms ................................... 13
   2.3 Efficiency of the Models ................................... 14
   2.4 Evaluation Metrics ......................................... 14
   2.5 Summary .................................................... 15

3 Project Deliverables 16
   3.1 Implemented Features ...................................... 16
      3.1.1 Extractive Aspect-Driven Summarization using Topic Modelling .... 16
      3.1.2 Extractive Query-driven summarization using sBERT ............ 19
      3.1.3 Semantic search using an aggregation of different models ....... 19
      3.1.4 Incomplete algorithms .................................... 20
   3.2 Design and Deployment of the prototype ..................... 21
   3.3 Demonstration of the Prototype ............................. 23
      3.3.1 Conducting document search ................................ 23
      3.3.2 Sending a query to highlight sections of a judgment ........... 24
   3.4 Evaluation of the results ................................... 25
List of Figures

1.1 A comparison between the interface in HKLII and Google .................. 3
1.2 An example of question answering (with query “injuries in right hand”) .... 5
1.3 An example of an aspect-driven summarization ............................... 5
1.4 The interface design of the new version of HKLII ............................. 7
2.1 Use of Sentence BERT in Semantic Search ................................. 10
2.2 Examples of word clusters in topic modelling ................................. 12
2.3 A comparison between keyword search and semantic search ................ 12
2.4 Work flow of semantic search in action .................................... 13
3.1 Work flow of query-driven summarization with sBERT ...................... 19
3.2 Work flow of semantic search .............................................. 20
3.3 Homepage of the interface of the prototype .................................. 21
3.4 Infrastructure for the prototype .............................................. 22
3.5 Document Search box of the User Interface .................................. 23
3.6 Interface of the Search Results .............................................. 23
3.7 Interface of a judgment page ................................................. 24
3.8 Interface of the query box in a judgment page .............................. 24
3.9 An example of a highlighted sentence after querying ....................... 24
3.10 Example of a missed sentence from the summarizer ........................ 27

List of Tables

1.1 Statistics of use cases in HKLII ................................................. 1
1.2 Distribution of different types of queries in HKLII ........................... 1
3.1 Examples of Topic Modelling Results ........................................ 18
3.2 nDCG and Precision of the top 10 (or fewer if returned fewer than 10) results of the 7 tested queries ........................................ 25
3.3 Precision of sentences ......................................................... 26
## Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AustLII</strong></td>
<td>The Australasian Legal Information Institute</td>
</tr>
<tr>
<td><strong>BERT</strong></td>
<td>Bidirectional Encoder Representations from Transformers</td>
</tr>
<tr>
<td><strong>BM25</strong></td>
<td>Okapi Best Matching 25</td>
</tr>
<tr>
<td><strong>CNN</strong></td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td><strong>ELMo</strong></td>
<td>Embeddings from Language Model</td>
</tr>
<tr>
<td><strong>FAISS</strong></td>
<td>Facebook AI Similarity Search</td>
</tr>
<tr>
<td><strong>HKLII</strong></td>
<td>The Hong Kong Legal Information Institute</td>
</tr>
<tr>
<td><strong>HKLII-v2</strong></td>
<td>The Second Version of The Hong Kong Legal Information Institute</td>
</tr>
<tr>
<td><strong>LDA</strong></td>
<td>Latent Dirichlet Allocation</td>
</tr>
<tr>
<td><strong>LLDA</strong></td>
<td>Labelled Latent Dirichlet Allocation</td>
</tr>
<tr>
<td><strong>LSA</strong></td>
<td>Latent Semantic Analysis</td>
</tr>
<tr>
<td><strong>LSTM</strong></td>
<td>Long Short Term Memory networks</td>
</tr>
<tr>
<td><strong>NER</strong></td>
<td>Named Entity Recognition</td>
</tr>
<tr>
<td><strong>NLP</strong></td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td><strong>pLSA</strong></td>
<td>Probabilistic Latent Semantic Analysis</td>
</tr>
<tr>
<td><strong>POC</strong></td>
<td>Proof of Concept</td>
</tr>
<tr>
<td><strong>sBERT</strong></td>
<td>Sentence Embeddings of Bidirectional Encoder Representations from Transformers</td>
</tr>
<tr>
<td><strong>TF-IDF</strong></td>
<td>Term Frequency–Inverse Document Frequency</td>
</tr>
<tr>
<td><strong>WorldLII</strong></td>
<td>The World Legal Information Institute</td>
</tr>
</tbody>
</table>
1 Introduction

In common law, there is a great emphasis on resolving disputes with reference to previous court decisions. As a region that practices common law, Hong Kong has a legal structure that makes court decisions legally binding on the lower courts [1][2]. Thus, researching historically relevant judgements plays an important role in the legal community, and the Hong Kong Legal Information Institute (HKLII) serves as a source of research materials for legal professionals.

1.1 Current status of HKLII

HKLII is a document database that gives the general public free access to primary legal materials. As of April 17, 2022, the website has been accessed over 14.8 million times and received approximately 4,000 visits daily since 11 November, 2011. Its deployment and infrastructure, including the search engine and database structure, is provided by the Australasian Legal Information Institute (AustLII), the Australasian counterpart of the worldwide Legal Information Institute systems [11]. Currently, HKLII supports the searching of legal judgements for retrieval, but is limited to simple lexical searches. The tables below show the current state and usages of HKLII, based on data collected from December 2017 to July 2019 [12].

Table 1.1: Statistics of use cases in HKLII

<table>
<thead>
<tr>
<th>Use case</th>
<th>Explanation of the use case</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browsing</td>
<td>Accessing the table of contents and reading ordinances and judgements</td>
<td>25.9%</td>
</tr>
<tr>
<td>Querying</td>
<td>Use the existing search bar and advanced search function in HKLII to look for documents</td>
<td>74.1%</td>
</tr>
</tbody>
</table>

Table 1.2: Distribution of different types of queries in HKLII

<table>
<thead>
<tr>
<th>Query type</th>
<th>Distribution</th>
<th>Average Session Lengths</th>
<th>Percentage of Long Sessions (&gt;10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity Search</td>
<td>26.6%</td>
<td>3.25</td>
<td>6.42%</td>
</tr>
<tr>
<td>Case Search</td>
<td>24.8%</td>
<td>2.18</td>
<td>2.35%</td>
</tr>
<tr>
<td>Legislation Search</td>
<td>6.1%</td>
<td>3.94</td>
<td>9.27%</td>
</tr>
<tr>
<td>Concept Search</td>
<td>26.4%</td>
<td>5.00</td>
<td>14.16%</td>
</tr>
<tr>
<td>Others</td>
<td>16.0%</td>
<td>4.08</td>
<td>9.75%</td>
</tr>
</tbody>
</table>

As seen in table 1.1, visit logs of HKLII have revealed that about three-quarters of visitors to the website made queries using the search bar or advanced search functions. Table 1.2 specifies the distribution between different categories of queries in HKLII. Session lengths mentioned in the third and fourth column refer to the number of actions, including additional queries and document clicks, after the user has typed the initial query; as a result, low session lengths represent users arriving at the desired document in fewer actions. Among the queries, case
search, entity search and concept search each account for about a quarter of the queries. In particular, case search and entity search refer to queries related to a particular historical case and person or organization respectively. In this scenario, it is not difficult to find the judgments users want in a keyword-based search due to the uniqueness of terms such as names and case identifiers in these queries. This is reflected in low average session lengths and percentages of long sessions for these two categories in table 1.2. On the other hand, concept searches, such as political affiliation or just and equitable winding up, concern the judgments relevant or exemplary of legal concepts on a higher level, which are more difficult to be captured accurately by a non-expressive lexical search. The difficulty in retrieving relevant judgments on such query is also observed in table 1.2 - based on the statistics of activity logs of the website, concept searches have the highest session lengths and the highest percentage of long sessions before users arrive at a document that matches their requirements. Apart from the poor ranking in keyword search results, which makes it more likely to require multiple searches before retrieving the desired article, lengthy search times are also an issue for users in HKLII, and such problem can be attributed to users requiring more time to identify if the judgement suits their needs, unlike case searches or entity searches where users can easily tell the difference between judgments.

The low recall rate with keyword searches also presents a big challenge for legal researchers in searching for all the judgments that suit their needs. This can be attributed to the fact that legal judgments and documents are often prone to specific jargons and polysemes, thereby decreasing both the precision and recall of a legal search engine that merely uses lexical searches and boolean searches like HKLII. Introducing semantic search can improve the ranking of the judgments, while document summarization will shorten the time needed to understand the article and determine the relevance to users’ needs, overall improving the user experience in searching legal documents.

1.2 Semantic Search

When querying an abstract legal concept, the user might not be able to provide the exact wording or technical terms of the concept, resulting in the keyword search engine missing out on the judgments that the user might want. For example, wounded hand only returns 2 results, while hand injury and injured hand returns 66 and 39 results respectively. It is noteworthy that some results in injured hand are not present in hand injury queries, and vice versa, so potentially relevant results are not brought up to the user if the user does either of the searches, often resulting in low recall when performing retrieval tasks. Such situations can be best avoided with the aid of semantic searches. There are many algorithms that can achieve different levels of semantic search, and given previous research in topic modelling on HKLII [12], this project started with topic modelling as the baseline to identify clusters of keywords that have strong associations with the query and compare such clusters against that of the documents. It is hoped that an algorithm that applies topic modelling would be able to associate phrases such as finger injury by its synonyms or more specific terms of fingers, such as hand, thumb and joint through the generation of conceptual topics (in this case, a topic that describes words related to a finger injury). By using topic modelling, a mapping between judgments and certain topics is constructed, and when the query arises, the most relevant topics to the query will be identified and matched against the topics in the judgments.
to rank the most relevant judgements to the user. In theory, the more coverage a topic has in a judgement, the more relevant the judgement should be in the context of the topic. On the other hand, modern NLP techniques such as ELMo and BERT embeddings can also determine similarities between a query and a judgment through similarities of embeddings, achieving semantic understanding between a query and a judgment.

1.3 Document Summarization

1.3.1 Query-driven Summarization

In the current HKLII interface, users will only be able to see the title of the judgements before clicking on the hyperlink text to inspect the document in detail. Such a design is not very telling because users are unable to see why the search engine included this result. In contrast, popular search engines like Google often highlight words in the description section of each query result, revealing some detail of the search result and helping the user determine the usefulness of the document before viewing it in full.

As seen in figure 1.1, the user interface on the right (Google) highlights the keywords that justify the reason behind the query result (in the above illustration, “HKLII” is the query, hence “HKLII” is shown bold in the description of the result). Although the final prototype of the project did not show highlights in the search results page, this project contains a similar feature through highlighting sentences and sections of the judgment that are the most relevant to a query after the user has clicked into the judgment, justifying why the judgments have been returned and saving users from having to read through the entire case, only to realize that the case does not suit their intents. This feature is called search results highlighting in this paper. From a technical perspective, this is considered an extractive summarization of documents, where sentences or paragraphs of a judgement are selected based on their importance and relevance.

Once the user has found the case useful, they would often want to retrieve sections of the judgment that are the most relevant to their intentions to facilitate their legal research. While different researchers may end up in the same document, the information that interests them may vary a lot. For instance, a user may be interested in the verdict of a judgment, while another user may be interested in the background of the plaintiff. In view of this, a question-answering tool is implemented in the judgment, aiming to help the user extract the sentences that are the most relevant to a query that is specific to the judgment. As such, search results highlighting can be viewed as a subset of this feature, as the feature above simply sets the question-answering query to be the query that the user used to perform searching.
In earlier reports, the feature search results highlighting is named as “query-driven summarization”, as this feature produces summarization based on queries from document retrieval, but in this report, query-driven summarization also includes the extraction of judgment-specific question answering, as it is also a type of summarization driven by queries. The following subsection explores summarization that do not require queries from the user.

1.3.2 Aspect-driven Summarization

While judgments can vary a lot depending on the nature and charge of the plaintiff, certain categories of judgments still attain attributes in common. This is where aspect-driven summarization comes into play, as it aims to highlight information that are generally perceived as important for judgments based on its category (currently for personal injury and drug trafficking). Information that were generally perceived as important, such as the amount of drugs and the background of the plaintiff for drug trafficking, or background and treatment for personal injury cases, were identified and compiled as a list of attributes. The two categories were chosen due to the prevalence of such judgments in HKLII, the presence of a common structure between these judgments, and most importantly, the availability of the manually highlighted summaries in such topics. From a legal perspective, the sentencing in drug trafficking cases and the quantification of compensatory damage in personal injury cases also particularly depend on past precedents, making these two categories of cases more frequently studied. When this feature is rolled to production, it is aimed that when the user is browsing a personal injury or drug trafficking judgment, they can select the attributes that they would like to see from the list of attributes and be provided a summary of such attributes, in an objective that crucial particulars of the case can be understood in a short period of time. Applying topic models to achieving summarization of such judgments were shown to have promising results in earlier research results [12]; however, the two models constructed for personal injury and drug trafficking cases were domain-specific and non-extensible to other types of judgments or even queries that were not available in the list of attributes. The availability of results in previous research [12] and the lack of extensibility for such models has therefore prompted the creation of a general-purpose question-answering tool, mentioned in the previous paragraph, that can highlight sentences relevant to the judgment regardless of the query and the category of the judgment. As such, the implementation of the general-purpose question-answering tool has been prioritized, and while the design for aspect-driven summarization using transformers is also discussed in this report, it was only partly implemented and not integrated in the prototype.

Overall, summarization tasks of legal judgments in HKLII can be divided into two categories - the first type requires input of a query, be it a query used for document retrieval or question answering, from the user, and the model is required to retrieve sentences or sections relevant to the query. This is the type of summary used in search query highlighting and question answering of a judgment, and is denoted as query-driven summarization. The second type of query is concerned with retrieving information that is commonly identified as important in legal research given a category of judgment, and does not require any query input from the user. It is called aspect-driven summarization in this paper. While aspect-driven summarization can also be simply converted to a query to be fed into the question answering model, their designs were different, as discussed in the next chapter. Query-driven summarization is also
not intended to replace aspect-driven summarization, as aspect-driven summarization serves as a way for readers, especially users without a legal background, to understand the common important aspects of a judgment. In other words, it is a way to guide the user to understand the most important parts of a judgment when they don’t know what they would like to query, making the process of legal research more accessible to people without a legal background.

Figures 1.2 and 1.3 illustrate an example of query-driven summarization and aspect-driven summarization of a personal injury case. As seen in the interface, factors such as backgrounds of the case, compensation, loss of future earnings are perceived as important information in personal injury cases, and the user can toggle between different options to view sentences relevant to the attribute. The figures are extracted from preliminary research work done [12].

Figure 1.2: An example of question answering (with query “injuries in right hand”)

Figure 1.3: An example of an aspect-driven summarization

1.4 Objective

This project aims to improve the overall user experience of the HKLII website, in particular during the phase of document searching, by improving the ranking system and understanding
queries at a semantic level, as well as providing summaries of the document, both from a query perspective and a category perspective. The project aims to perform the tasks by adopting topic modelling, a technique that was investigated and implemented in past related research [12], as well as embeddings from transformers, to evaluate and compare the performance of different algorithms. Finally, a proof-of-concept (POC) prototype has been implemented to demonstrate the functionalities on the user interface for viewers to better understand the improvements made in this project compared to the current HKLII. It is hoped that this functionality will eventually be integrated and deployed to HKLII-v2, which includes a new user interface and up-to-date infrastructure, and is being developed now by a team of developers.

1.5 Related work

This project is related to recent work done in applying topic modelling to semantic queries and document summarization to HKLII, which has shown promising results [12]. Along with related work on legal text processing, it is observed that Latent Dirichlet Allocation (LDA), labelled LDA (LLDA) and hierarchical LDA are amongst the topic modelling techniques adopted for use [6].

Apart from topic modelling, many papers have also suggested traditional and novel methods and algorithms in achieving legal information retrieval and summarization. For information retrieval, Zamani et al. suggest having a ranking system that takes into consideration the field that the satisfying phrases appear in a search result using algorithms similar to BM25F [5]. Besides, several Siamese networks were raised as methods to rank documents, the most notable of which, from Huang et al., suggested the use of Deep Structured Semantic Models (DSSM), which consists of two deep models for the query and the document with fully-connected layers and train the model on clickthrough data [9].

Since its introduction in the late 2010s, supervised models in the NLP area, including ELMo and BERT, has been widely adopted in retrieval systems in non-legal organizations and industries, as they were shown to have a better semantic understanding on the queries. LinkedIn has developed an open-source framework called DeText, which aims to handle all sorts of NLP tasks, ranging from Information Retrieval to Query Autocompletion and Document Classification. The network structure of DeText consists of both traditional approaches such as TF-IDF and modern techniques in embeddings generated from transformers, with the final ranking being determined by an aggregated score that propagates through a multilayer perceptron (MLP) trained using online metrics collected from actual users of LinkedIn [14]. Westermann et al. suggested annotations of legal documents using sentence transformers and sBERT. Along with topic modelling and traditional methods, a hybrid of these methods have been implemented in the prototype of this project [4].

Two other final year undergraduates of the year 2021-22, Yuchen Liu and Huijie Pan, are also working on adding features to HKLII by adopting natural language processing (NLP) tools to improve the user experience of browsing documents. Their work will include introducing taggings to the documents, which includes information such as key legal concepts and unique words in the document, as well as a document recommendation system that recommends highly related cases to the reader, based on their activity on the website. Figure 1.4 illustrates
a designed interface by the development team, which not only improves the user experience and upgrades the interface to match modern web design principles, but also includes all the proposed features for all final year students working on the project: document summarization (on the left as the synopsis), tags of legal concepts (upper right) and related documents (lower right).

1.6 Outline of the report

The report is structured in four chapters. The first chapter (this chapter) describes the background of the project, the state of the current HKLII website, as well as preliminary and related work and this project’s contribution to the new HKLII.

The second chapter elaborates and explains the methodology and algorithms used to achieve semantic search and document summarization in detail, with a focus on using topic modelling and sBERT. It also goes over some other algorithms that are implemented in the project, as well as the processing of the available data and the performance evaluation benchmark for different models.

Chapter three mentions the research done in the project and the details of the implementation of the prototype for HKLII, as well as a demonstration of it. It is followed by an evaluation of the results from the model. A reflection of the initial project objectives is also formulated, followed by a discussion of challenges faced in the project.

The final chapter concludes the report by mentioning future work that can be done in this project and summing up the contribution of this project.
2 Methodology and Algorithms

This chapter highlights the details of the algorithms in achieving the above-mentioned features in HKLII. The methodology of achieving semantic searching and summarization of legal documents using topic modelling and sBERT will be discussed, with a brief inclusion of other algorithms used.

2.1 Data Collection and Development of Prototype

Access logs, annotated judgments for personal injury and drug trafficking cases, and legal documents of the current HKLII website are readily available online in a server dedicated for research in HKLII. This makes data collection in this project straightforward. By investigating the logs, one can also identify the problems in HKLII and develop additional features in the prototype to benchmark the performance of new algorithms and best avoid the issues from happening again.

In an attempt to prove the integrability of the project to HKLII-v2, during the development of the prototype, an integration is made between the NLP implementation of semantic search and summarization, a proxy server to the current server for HKLII, and the new frontend service that will make calls to both the proxy and the prototype server. Due to the abundance of community support and packages, Python is the programming language used in the implementation of the semantic search and summarization model, as well as the prototype server that responds to HTTP requests for these two services, which is enabled by FastAPI, a server-side framework for Python that is highly efficient and performs better than conventional frameworks such as Flask. The details of the implementation of the prototype, including the packages used in the implementation of the Python server, will be discussed in the next chapter.

2.2 Algorithms used in the Project

This subsection goes through all the algorithms and models that were experimented or implemented throughout the project.

2.2.1 Data and Query pre-processing: Named Entity Recognition and Regular Expressions

As mentioned in Chapter 1, search queries for document retrieval for HKLII can roughly be classified into five categories. This project concerns entity search, case search and concept search, which are the three most common types of search as per the data collected in 2019 and presented in Table 1.2. Entity search concern the searching of a person or organization, such as the name of the plaintiff, the judge, or the representations of the judgment. Case search refers to the search of a judgment based on its citation number, such as “[2016] HKCA 94” or “[2016] 4 HKC 204”. Concept search are searches that relate to a legal concept, such as an ordinance. A search can, of course, also be a hybrid of the above, such as “[2016] HKCA 94 Chan Tai Man”, which contains both an entity and the name of the case. Semantic search algorithms and sentence embeddings, especially models that were not fine-tuned with legal documents in HKLII, generally do not work well with named entities, cases identifiers and names of legislations and ordinances, since the words are very specific, and do not affect the semantic meaning.
in the sentence. On the other hand, entity searches are likely looking for phrases with an exact match, thus, performing keyword searches is likely a better option. Conversely, it is aimed that semantic search can play a role in assisting concept searches, which may not appear in the exact same form as that in a relevant judgment. The variety of different types of searches in HKLII calls for the need to pre-process queries prior to the retrieval process, by determining algorithms that suit the need of the search the best. In particular, regular expressions and named entity recognition are used to chunk sections that uses different algorithms for retrieval.

Case search with regular expressions. Search results in the HKLII search engine are returned in the unit of judgments, but legal documents also exist in the form of cases and reports. Both reports and cases have a many-to-many relationship with judgments, meaning that a report/case can contain multiple judgments, and a judgment can also be contained in multiple reports/cases. Since there is a universal notation for identifiers on reports, cases and judgments, regular expressions were used to parse if the query contains a notation in the form of either a report ID, case ID or judgment ID. Judgment IDs are in the form “[<YEAR>] <COURT> <JUDGMENT ID>”, while case IDs are in the form “<CASE TYPE><CASE ID>/<YEAR>”, and Report IDs are in the form of “[<YEAR>] <COURT NAME> <JUDGMENT ID>”. All the patterns above can be identified and parsed by regular expressions. Once a pattern that matches any of the above ones, the model will search for exact matches of such judgments using the indexes created in the database.

Named Entity Recognition for Entities. Unlike citations of judgments, named entities do not attain a specific structure, making the task of extracting entities (in this case, names and organizations) more challenging. In this project, named entity recognition (NER) is used to facilitate the recognition of such entities in a query. The named entity recognition model ner-english-ontonotes-large provided by Flair is a 18-class NER model that supports recognition for 18 classes, which includes organization (ORG) and person names (PERSON), with a F1 Score of 90.9 on the Ontonotes dataset. Such names are then extracted to perform fuzzy keyword searches, leaving the query without entities for semantic search.

Fuzzy Search on Keywords. Searches of names are often prone to spelling or translation errors. For instance, one may be trying to translate the Chinese name of a person, and mixed up Chen and Chan, or Chi, Tsz and Chee. Fuzzy search is introduced with typos in mind, in hopes of returning judgments even when the user types the wrong translation or has typos in his/her query. Currently, the implementation in the prototype supports fuzzy search uses a hybrid of similarity measures between strings, including weighted Levenshtein distance, ratio of partial matches between the strings, and similarity between the tokens in the string. The framework rapidfuzz, which is an improved version of the popular fuzzy search framework fuzzywuzzy, is used to efficiently compare between the named entities in the string and the keys of a mapping between names and judgments by using indexes in memory. While currently not implemented, with the current database of over 350 Chinese (Cantonese) word names translation, a mapping in the form of a graph between all words can be implemented to handle Chinese name words with similar pronunciations in Chinese (Cantonese).
2.2.2 Sentence Transformers

The introduction of ELMo and BERT in 2017 and 2018 provides a more contextual understanding of words in sentences, giving them an edge against word embedding models such as word2vec or GloVe. Sentence BERT (sBERT) is a Python framework, backed by the architecture of BERT, that generates multidimensional vectors from a sentence. These vectors can then be used to find the sentences, among a database of sentences, that are the most relevant to the query. Thus, it can be applied to both document retrieval and query-driven summarization.

The Facebook AI Similarity Search (FAISS) is a Python framework, with high customizability, that can quickly handle and retrieve the number of sentences that are spatially the closest in the \( n \)-dimensional space of the vectors. During semantic search, the top 1000 most relevant sentences are selected among the set of about 9 million sentences across 93,000 judgments (denoted later as a set of sentences \( S \)) by comparing to the query using cosine similarity. All the judgments that have at least one sentence included in \( S \) are considered relevant judgments.

The judgments are then ranked based on the sum of the similarity scores between query \( q \) and \( s \) for all \( s \in D \cap S \), where \( D \) is the judgment. The algorithm is designed to only consider the sum of sentences that are already in \( S \), because it is expected that a query inputted by the user will rarely be highly similar to every sentence in a judgment \( D \). For example, the query “freedom of expression” is likely to only semantically match sentences that mention such a concept in the judgment, even for judgments that are deemed relevant to this query by a human. Therefore, it is proposed that only the relevant sentences should be considered in computing the final score of a judgment. The figure below shows the flow of semantic search using Sentence Transformers.

![Figure 2.1: Use of Sentence BERT in Semantic Search](image)

The implementation of sBERT in question-answering query summarization is more straightforward. Depending on the length of a judgment \( D \), the summarizer selects about 5% of sentences, with a minimum of 5 and a maximum of 15, that are semantically the closest to the query typed by the user. Cosine similarity is being used in the implementation, though sBERT also supports similarity measures of Euclidean distances and dot products.
2.2.3 PageRank

Invented initially by L. Page [7], PageRank is an algorithm that measures the value of a webpage in the public internet, based on the number of hyperlinks that direct to the webpage. This is analogous to the case for legal judgments in HKLII, where a hyperlink reference can be thought of as a neutral citation or a reference of a judgment from another judgment. This algorithm is proposed due to the observation that while exceptions exist, legal researchers tend to look for influential judgments and cases that laid a foundation or a decision for judgments at a later stage. The citations database at HKLII includes judgment-to-judgment citations, but the majority of citations in HKLII were actually cited against a case or a report, which can include multiple cases. Should multiple judgments be included in a case or report, the judgments tend to be highly similar, if not consecutive judgments appearing in a case. As such, for simplicity, it is assumed that whenever judgment J cites report R or case C, it assumes that J cites judgement j for all j ∈ R (or C). The current implementation applies PageRank by default, but it can also set to be toggleable in case the feature is not desired for a researcher.

During the testing of queries, it is seen that potentially influential judgments that are more recent (for instance, within 5 years) have been pushed down the priority list. This can be understood as judgments that are more recent, while being potentially very influential, are simply too recent that there isn’t enough judgments citing it. As such, a multiplier factor is given to judgments that occur within the last 5 years (since 2017), in order to reduce the time factor. An alternative to such will be to compare the citation score divided by the age (number of days since the judgment date) of the judgment, which can be explored in future studies.

2.2.4 BM25

Okapi Best Matching 25 (BM25) is a probabilistic retrieval framework that uses a bag-of-words retrieval function and ranks a set of documents purely based on the query terms without taking into consideration the proximity of it. In mathematical terms, for every query word q₁, q₂, . . . , qₙ ∈ q, the BM25 score of a document score(D, Q) is defined as

$$
\sum_{i=1}^{n} \frac{\text{IDF}(q_i) \cdot f(q_i, D)(k_1 + 1)}{f(q_i, D) + k_1(1 - b + b \times \frac{|D|}{\text{avgdl}})}
$$

where \(\text{IDF}(q_i) = \ln \left( \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right)\).

The reasons that BM25 is introduced are twofold - on one hand, while NER is able to capture most of the named entities in a query, it may still miss out on some, and semantic search tends to perform poorly on entity searches. BM25 serves as a backup to NER so that names can still be captured to a certain extent. On another hand, the semantic search model, especially when it is not fine-tuned to legal judgments, may perform poorly when facing some very specific vocabularies and terms in law. BM25 comes in to help identify these phrases and extract such judgments.

2.2.5 Topic Modelling

Topic modelling is a statistical framework that extracts abstract topics from a collection of documents. It builds upon an assumption that words that appear closer in the documents are likely to attain similar meanings. In topic modelling, a topic is made up of words and a document is made up of a mixture of topics. Formally, given a context of documents (also known as corpus), performing topic modelling by inputting the desired number of topics \(n\) returns a
set of topics \( \{T_1, \ldots, T_n\} \), with each topic \( T \) containing word and coefficient pairs that indicate how close a word represents the topic. Hence, a topic can be considered as a cluster of words. Figure 2.2 illustrates examples of good outputs of word clusters, from an intuitive perspective, while Figure 2.3 demonstrates the difference between a keyword search and semantic search.

There are many topic modelling techniques that can potentially be successful to HKLII, including LDA, LLDA, hierarchical LDA, etc. In previous research work, LDA has been selected to perform topic modelling, which generates the set of topics based on the word distribution of each topic, and the topic distribution of the document (the probability of each topic occurring in the document) [12]. Three approaches of pre-processing for LDA had been discussed and put forward with varying degrees of labelling to extract the topics of a judgement. The method with the lowest intervention feeds in all the text, with only removal of numbers and stop words, to the model, while the second approach extracts manually annotated words as labelled text for LDA to remove the unimportant details of a topic. The final approach with most human effort further categorizes the important information into aspects (such as plaintiff’s background, injury, treatment and loss for personal injury cases) and information in each aspect are fed into LDA separately. It has been shown that the labelling with categorized aspects has the best performance.

In previous research, semantic search is performed using the topics generated from topic modelling and compares them against the judgements in terms of the topics. Figure 2.4 explains the intuitive flow that a query has under a semantic search with the generated topics. An example of a quantitative way to measure the relevance between a given query and a judgement is to construct probability vectors \( p \in \mathbb{R}^n \) for the query and each judgement in the database, with each entry \( p_i \) denoting the likelihood that the query/judgement is relevant to a topic \( T_i \in \{T_1, \ldots, T_n\} \). Once all the probability vectors are computed (only the probability vector of the query should be computed in query time, probability vectors of the judgements can be pre-computed), a function \( f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \) that can measure the distance between the
two vectors, such as the dot product, can be used to determine and rank the relevance of each judgement against the query (shown as “compared and ranked” in the figure). This is a proven way to convert topic modelling results to ranking documents in a query, and will serve as a baseline model to be compared against the new design discussed above.

An alternative to topic modelling with LDA is Latent Semantic Analysis (LSA) or Probabilistic Latent Semantic Analysis (pLSA), which uses Term Frequency-Inverse Document Frequency (TF-IDF) and probabilistic models respectively to capture hidden concepts and topics.

In terms of document summarization with topic modelling, the methodology for query-driven summarization and aspect-driven summarization are different. In the current sentences, query-driven summarization uses the topic model generated from all sentences to compute the similarity between a topic and a paragraph/sentence, and selects the sentences that are most relevant to the topics of the user’s queries.

Aspect-driven summarization, on the other hand, first identifies the important attributes of a judgement, before comparing the paragraphs in the judgement and the topics using similar functions and algorithms. The project will build upon the current work by both trying to alter the topic modelling functions to generate different topics and the mechanism for the processing of the topics.

2.2.6 Alternative algorithms

The algorithms above were adopted and implemented in this project, but there are also other algorithms that may potentially help the summarization and retrieval of judgments, among which include the use of Convolutional Neural Networks (CNN) to predict the likelihood that a sentence should be included in a summarization. Traditional sentence ranking algorithms such as TextRank may also be used to determine the importance of a sentence. In turn, this can be taken into consideration in the ranking of judgments in semantic search, by considering not only the relevance of a sentence and the query, but also the importance of the sentence in the judgment.
2.3 Efficiency of the Models

Both semantic search and summarization are designed with efficiency in mind, as the algorithms are practical only if the user can receive the results within seconds, ideally within 3 seconds. In the current implementation of the prototype, a search query takes roughly 3-5 seconds to complete, while a question-answering summarization of a judgment is done within 2 seconds.

The models, especially semantic search, can be further improved in efficiency with the use of parallel computing such as multithreading. In the pre-processing stage, named entity recognition is done with an external model, which is loaded into memory when the server starts. This makes named entity recognition for a query very fast. The process of using rapidfuzz is also optimized, as named entity recognition on all judgments were run in advance and saved in a JSON file, before being loaded into memory when the server starts. The rapidfuzz algorithm is also very efficient, returning keys of names that are highly similar to that in the query within 100 milliseconds.

Similarly, FAISS makes use of pre-computed sentence embeddings of every sentence in every judgment and spatial indexes to retrieve sentences that are semantically the most similar very efficiently. Even without indexes and segments on different clusters of sentences, FAISS is currently able to retrieve the 1000 most relevant sentences in about 1 second.

The BM25 algorithm, which is handcrafted without any packages in the implementation of the prototype, proved to be the bottleneck of the search algorithm, taking about 2 to 3 seconds to complete. It is expected that with the help of external packages, which run C++ code under the hood, the code will be able to execute more efficiently.

The aggregation and sorting of scores is done in linear and $O(n \log n)$ time, as it is simply a key-value pair between the judgment ID and its score.

2.4 Evaluation Metrics

The evaluation of the performance of algorithms on document summarization and semantic search involves both quantitative and qualitative measures. While legal experts are available for consultation, the project has run out of time in the evaluation phase and insufficient results were retrieved. It is planned and hoped that they will be invited to evaluate the summaries, even after the project, to grade the significance of each paragraph in the summary and increase the quantitative results available for the paper.

For semantic search, normalized discounted cumulative gain (nDCG), which is defined to be a score between 0 and 1 that represents the relevance score of the judgment to the query, and precision are the two metrics chosen for quantitative evaluation, with the baseline being the current HKLIΠ system. The evaluation is then partnered with cases that are relevant but not returned in HKLIΠ, as HKLIΠ is likely having a high precision but low recall due to the nature of keyword searches.
As for query-driven summarization, metrics against the optimal summary, which is a summary conducted manually, will be compared and measured quantitatively as a benchmark. In particular, the precision of the question answering sBERT model is compared against the baseline topic modelling model. While initially intended to be a useful metric, recall is being omitted in the current evaluation, due to the difference in the size of selection between manual summaries and computer-generated summaries. Currently, instead of setting a threshold on the similarity score between a sentence and a query, computer-generated summaries are constructed by simply selecting and highlighting the top \( n \) sentences \( (5 \leq n \leq 15) \), but manual summaries have selected all sentences that are relevant to the query. Depending on the nature of the query, it is possible that the number of sentences in the judgment that are highly relevant are way more or fewer than \( n \). This means that even if the summarizer ranked the sentences perfectly, the recall rate could still be less than 100\% simply due to the design of the feature, which makes the metric less telling on the actual performance of the model. Hence, only precision is being used in summarization.

While not being used in the current evaluation, evaluation tools such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE), as suggested in other research papers [3], are also popular metrics in document summarization and allow effective comparison between human-written summaries and computer-generated ones. ROUGE could be used in both query-driven and aspect-driven summaries in HKLII to measure how well the machine has captured the key information, words and legal terms in the summaries generated.

### 2.5 Summary

This chapter has introduced and elaborated the algorithms behind semantic searches and document summarization. The flow of the project and designs of implementations were also laid down, followed by a discussion on the methodology of evaluating the performance of algorithms. The next chapter will go over the results of the implementation.
3 Project Deliverables

This chapter first outlines completed work of the project, before elaborating on the details and features of the implemented prototype. It is followed by an evaluation of the results of the models, before a reflection of the project objectives and schedule. The chapter ends with a discussion of the challenges faced in the project.

3.1 Implemented Features

3.1.1 Extractive Aspect-Driven Summarization using Topic Modelling

Preliminary results of research, implementation and evaluation of topic modelling algorithms in HKLII for extractive aspect-driven summarization of drug trafficking cases are available on Google Colaboratory (https://colab.research.google.com/drive/1n7GfXvA5M_BNzcXGRoG10iC2PVwVIeAj).

The design of the algorithm goes as follows: all annotated sentences are first separated as “sub-aspects”, which include things such as motive behind the offence, the criminal record of the defendant, the family of the defendant, etc. Topic models with 1 topic are then generated using LDA and TF-IDF based on these sub-aspects. At the same time, a word2vec model that consumes all words in the annotated text as the input is generated. The output of these sub-aspect models, which contain word-similarity pairs, were converted to a topic weighted vector. It is essentially the weighted average of the word2vec vectors of the top 20 words in the topic model, with the weight being the similarity coefficient of the topic model. Likewise, each sentence in the legal document is converted to a text weighted vector, which is the average of the word2vec vectors of all the words in the preprocessed sentence. The cosine similarity is used to determine the similarity (called topic-text similarity) of the topic weighted vector and the text weighted vector, so the lower the scalar value is, the more similar the text is compared to the topic. Hence, each sentence is compared against the topic weighted vector for each sub-aspect, and the sentence will be categorized as the sub-aspect whose topic-text similarity is the highest. When the user clicks the sub-aspect, sentences that have the highest topic-sentence similarity will be chosen and highlighted.

The above-mentioned design is arrived at after unsatisfactory results with techniques that were previously studied. Three types of pre-processing were done in extracting topic models. The first approach was to feed all of the labelled sentences into the model, with each labelled sentence serving as a “document”, thus generating multiple topics to be compared against the sentence. Results from this design were, however, not very appealing, with word combinations that one would expect to be in the same topic to appear in either multiple topics, or distinct but different topics. This will, in turn, generate a poor topic weighted vector. Even though separating annotated sentences into aspects and generating topic models on such level proved to be effective for personal injury cases in previous research, results were not encouraging for drug trafficking cases - topic models generated from an aspect also shows the same words appearing in different topics. This can be attributed to the observation that the variety in vocabulary for drug trafficking cases is much less than that of personal injury cases, which is in turn justified by the nature of the cases themselves, since personal injury judgements
involved cases with more diverse backgrounds and situations than drug trafficking judgements. Furthermore, it is noticed that words coming from sub-aspects in the same aspect should be considerably different from one another. For instance, international drug trafficking offences, persistent offences and offences while on bail are all aggravating factors to a drug trafficking case, but the vocabulary expected in these sub-aspects should be different from one another. This leads to the current solution of using labelling to its finest degree of separation and creating a topic model with only one topic for each sub-aspect. As reflected in the results, the *topic weighted vector* for such a model appears to be the most accurate, thus providing the most accurate summaries. Table 3.1 shows examples of sentences that were classified into the sub-aspect based on the judgement HKSAR v. KWOK KAM FUNG [2012] HKCFI 376; HCCC 464/2011.

The above examples and differences in implementation in topic models between *personal injury* and *drug trafficking* cases is a precise representation of where topic modelling falls short - the models were ad-hoc, lacked a generic structure and only worked well within a specific domain or category. The model also relied on over annotations of 1,000+ judgments in drug trafficking for training, which required extensive human efforts to produce. This makes the task of extending the model to different judgments and semantic searching of all judgments very difficult. On the other hand, using sBERT pre-trained with non-legal papers have shown to perform, on average, a better degree of understanding across different types of judgments in HKLII.
Table 3.1: Examples of Topic Modelling Results

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Sub-aspect</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offence</td>
<td>Quantity of Drugs</td>
<td>The Government Chemist certified that the eight bags of “Ice” were 11.14 grammes of a crystalline solid containing 10.90 grammes of methamphetamine hydrochloride, a salt of methamphetamine. The defendant has pleaded guilty to trafficking in a dangerous drug, namely 11.14 grammes of crystalline solid containing 10.90 grammes of methamphetamine hydrochloride. As a result of this subsequent amendment the first sentencing band relates to quantities up to 10 grammes and so in the present case, as the quantity in which this defendant trafficked was 10.90 grammes of “Ice”, this defendant is placed in the 7 to 10 years band for sentencing purposes.</td>
</tr>
<tr>
<td>Mitigating Factors</td>
<td>Assisting Authorities</td>
<td>At around 1905 hours, when the police started to conduct the house search, the defendant told a police officer that the dangerous drugs and packaging paraphernalia were kept inside a drawer of a white cabinet near the refrigerator.</td>
</tr>
<tr>
<td>Background</td>
<td>Education</td>
<td>The defendant is 40 years of age and has received education only up to secondary Form 1 level. She was unemployed at the time of her arrest. She is single, and lived alone. She claims to have been addicted to “Ice” for the past 10 years.</td>
</tr>
<tr>
<td>Penalty</td>
<td>Starting Point</td>
<td>The resulting sentence is 4 years and 8 months, and that is the sentence I impose. In the present case, the quantity of drugs is 10.90 grammes, just above the 10 grammes lower limit of the sentencing band. I have decided to extend to the defendant the last chance and the leniency that has been asked, and instead of increasing the starting point within this band, I will adopt the lower limit of the band as my starting point. That is, 7 years. I discount her sentence by one-third to allow for her plea of guilty.</td>
</tr>
</tbody>
</table>
3.1.2 Extractive Query-driven summarization using sBERT

The rationale behind applying sentence BERT to query-driven summarization was explained in Chapter 2. In summary, query-driven summarization highlights the sentences which appear to be the most similar to the query inserted by the user, by comparing the cosine similarity between the sentence embeddings of each sentence and the query. The flow of extracting sentences from sBERT embeddings of the query and sentences in a judgment are shown below.

Figure 3.1: Work flow of query-driven summarization with sBERT

3.1.3 Semantic search using an aggregation of different models

The implementation of semantic search in the prototype has used a hybrid of the algorithms in Chapter 2. When a query is received, named entity recognition and parsing of citations using regular expressions is done to identify potential entities and cases that the user inserted in the query. Besides, a simple parsing of double quotes is also done to see if the user wants to have an exact match for particular phrases. This is introduced in the engine because the engine is intended to be a superset of the current HKLII search engine, meaning that any queries in the present HKLII can also be expressed in the new engine. This process of constructing a semantic understanding of the query is denoted as the pre-processing phase of the search. In an event that there are patterns that resemble a case or named entity, the machine will prioritize judgments with these entities by assigning them a very high score.

After the pre-processing part of the query is complete, the whole query is fed to the BM25 model, while a query with named entities removed is fed to the Semantic Search model. Scores are obtained from both models and the result is aggregated to form a final score. Finally, the query results are shown to the user in descending order of scores.

The figure below explains the flow of semantic search in the current system.
3.1.4 Incomplete algorithms

Using Transformers for Abstractive Document Summarization. The lack of extensibility for summarization with topic modelling calls for an algorithm that do not require annotations in generating summaries that can still be of some use to the user. While being a supervised model in nature, PEGASUS, which stands for Pre-training with Extracted Gap-sentences for Abstractive Summarization, is a self-supervised model that generates abstractive summaries of documents [8]. During pre-training in PEGASUS, several sentences in the document were removed before being passed to the input and the transformer is tasked to recover them. Such design encourages the transformer to understand the language and the context of the corpus at a meticulous level, while minimizing human annotations needed in a conventional supervised learning algorithm.

An important issue observed from PEGASUS is that when observing the test cases in applying PEGASUS to drug trafficking cases, it is able to extract some sentences that are important and mentioned in the attributes of drug trafficking, but it lacked a list of essential outcomes that one would expect in a judgment. The project has experimented generating summaries using a legal version of the PEGASUS model online on Hugging Face. The following example shows the summary generated from HKSAR v. KWOK KAM FUNG [2012] HKCFI 376; HCCC 464/2011, which is also the drug trafficking case used to demonstrate the topic model algorithm above.

“Police officers on duty in Mong Kok observed the defendant exiting from her flat in a furtive manner. When the police started to approach the defendant to make inquiry of her, she immediately tried to return to her flat. The total retail value of the drugs seized is estimated by a police drug expert to be $7,820. In a number of different cases the Hong Kong Court of Appeal has set out the principles and sentencing guidelines for judges to follow when sentencing a defendant convicted of a drug trafficking offence. The purpose of these sentencing bands is to lead to consistency in sentencing, and secondly, to deter those who might otherwise be tempted by
what they perceive as an easy and low-risk means to substantial profits. In the present case, the court heard that the defendant has a criminal record dating back to 1989, when she would have been only 18 years of age. She was then convicted and fined for possession of dangerous drugs. In 1994 she was convicted of robbery and sentenced to a total of 10 years’ imprisonment. After her release from prison she has continued to be involved with dangerous drugs, and in 2005 she was sentenced to 13 months’ imprisonment for the trafficking offence, which she pleaded guilty to. The court also heard that she has a long criminal history, including convictions for robbery, possession with intent to supply, and possession of a controlled substance. The judge agreed that she should be sentenced to 15 months imprisonment. The sentence will be determined by the court at a later date.”

The summary showed inaccuracies in the sentencing of the criminal, as the true imprisonment was 13 months instead of 15. This is reasonable because the model may have a poor sense of predicting the exact verdict of the cases when the relevant sections in the document were removed. However, essential details such as the value of the drugs and the details of the crime were correctly extracted. The lack of accuracy and customizability has made PEGASUS an unideal model for implementation of summarization in HKLII.

3.2 Design and Deployment of the prototype

The prototype is being developed on bigbaby, which is a server dedicated to research and development of HKLII. It has server-grade memory and GPU processing power for running Natural Language Processing, making it an ideal place to host and develop the prototype of this project. Currently, the website is being hosted on http://147.8.178.175:8242 and can be accessed within the University of Hong Kong network.

Figure 3.3: Homepage of the interface of the prototype

Figure 3.3 is a screenshot of the home page of the prototype. The User Interface is provided by the development team of HKLII-v2, and remained the same for most interfaces, except for its API calls to the backend for semantic search and summarization. A query box is also added
to the judgment page, so the user can enter a query and view the highlighted sentence after submitting the query.

As for the backend of the system, the interface is making API calls to two servers - a proxy server to HKLII and the NLP server that serves requests to queries on searching and summarization. The proxy server, in turn, forwards the requests to HKLII-v2 (available at v2.hklii.hk in the University of Hong Kong network). A proxy server is added in between the frontend and the backend of the server in order to go-around a Cross Origin Resource Sharing (CORS) error in HKLII-v2. Moving on to the NLP server, it is implemented with Python and FastAPI, a HTTP framework that supports server hosting in Python.

Figure 3.4: Infrastructure for the prototype

The diagram above illustrates the networking behind the prototype. A demonstration of using the features in HKLII can be seen in the following section.
3.3 Demonstration of the Prototype

As discussed above, semantic search and question-answering of query summarizations were implemented in the prototype. This subsection shows the screenshots of the prototype in achieving these two features.

3.3.1 Conducting document search

The homepage of HKLII contains a search bar where the user can type queries, shown in figure 3.5. After the search results were returned, search results were shown in the form of a list, as seen in figure 3.6.

Figure 3.5: Document Search box of the User Interface

![Figure 3.5: Document Search box of the User Interface](image)

Figure 3.6: Interface of the Search Results

![Figure 3.6: Interface of the Search Results](image)
3.3.2 Sending a query to highlight sections of a judgment

Figure 3.7 shows the interface of the page when viewing a judgment. If the user wants to type a query, it can insert a query using the query box on the top left panel of the judgment, in the “enter a query” query box. Figure 3.8 is an example of the query box after it is filled with a query “details of applicant” and submitted. A “reset” button shows up under the query, allowing the user the option to reset the highlights shown in the judgment.

Paragraphs and sentences that are deemed relevant to the model will be highlighted and shown in the form like Figure 3.9. The text will be highlighted in yellow, capturing the user’s attention.

Observations for the Applicant:

1. The Applicant is a 37-year-old national of Iraq who was deported from Singapore to Hong Kong on 5 May 2014, and when he was refused entry by the Immigration Department, he raised a non-refoulent claim on the basis that if he returned to Iraq he would be harmed or killed by his uncle who blamed him for the accidental death of his uncle's son and/or by a former member of the Islamic Army in Iraq (“IAI”) who had extorted money from his family and/or by the Iraqi Security Force (“ISF”) for having been involved with the US Army and IAI in the past. He was subsequently released on recognizance pending the determination of his claim.
3.4 Evaluation of the results

3.4.1 Semantic Search and Document Retrieval

Due to the tight schedule in conducting a thorough research for semantic search, only 5 query results were evaluated. This reflects a small-scale research, but queries have been designed to cover different aspects of judgments. Their nDCG scores and precision can be seen in the Table 3.3.

Table 3.2: nDCG and Precision of the top 10 (or fewer if returned fewer than 10) results of the 7 tested queries

<table>
<thead>
<tr>
<th>Query</th>
<th>nDCG (HKLII)</th>
<th>Precision (HKLII)</th>
<th>nDCG (new)</th>
<th>Precision (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>right to political participation</td>
<td>0.73</td>
<td>50%</td>
<td>1.0</td>
<td>100%</td>
</tr>
<tr>
<td>implied hearsay</td>
<td>0.80</td>
<td>20%</td>
<td>0.78</td>
<td>40%</td>
</tr>
<tr>
<td>land resumption</td>
<td>0.79</td>
<td>50%</td>
<td>1.0</td>
<td>100%</td>
</tr>
<tr>
<td>right to consultation</td>
<td>0.96</td>
<td>20%</td>
<td>0.90</td>
<td>60%</td>
</tr>
<tr>
<td>director fiduciary duty</td>
<td>1.0</td>
<td>100%</td>
<td>1.0</td>
<td>100%</td>
</tr>
<tr>
<td>freedom of expression</td>
<td>1.0</td>
<td>100%</td>
<td>1.0</td>
<td>100%</td>
</tr>
<tr>
<td>plane crash</td>
<td>0.50</td>
<td>33%</td>
<td>0.83</td>
<td>30%</td>
</tr>
</tbody>
</table>

Overall, it can be seen that the precision of the top 10 search results of the engine in this project is better than that of the current one in HKLII. During the analysis of these search results, some interesting observations can be seen from the queries above.

In the query “freedom of expression”, because of the large number of judgments that satisfy the query, both HKLII and the new engine are able to return relevant judgments in the top 10 results, but only 1 of which were common between HKLII and the new engine. In the query “right to political participation”, searching for “right to political participation” returned 0 results, and the diagram above considers results of “right” AND “to” AND “political” AND “participation”. The first few results of such a query in HKLII are highly irrelevant. If the query “right to” AND “political participation” is used instead, most of the query results were instead cases of refugees seeking for asylum in Hong Kong because of their political participation in their foreign countries. Similarly, one case for the query plane crash were about the effect that the 9/11 plane crash attack in 2001 had on the real estate market. This shows that the lack of understanding on a semantic level can also result in not only low recall, but also low precision.

It is very difficult to calculate the recall of a query in HKLII because it would take enormous time to determine the total number of relevant judgments among the 90,000+ judgments for just a query. In other words, the quantity of False Negatives cannot be determined. That said, one can still observe examples of higher recall from the new HKLII engine. From the query plane crash, the new engine is able to return two highly relevant judgments of cases with family members of a casualty in a plane crash suing the airline company for the accident. These cases
were not visible in HKLII.

As for Case search, searching functionalities of judgments using case ID or report ID has given an edge to the new search engine. While one may search for judgments using its citation, such as “[2016] HKCA 20”, it is also common for users to search for judgments using the case ID and report ID. Regular expressions have presented a straightforward and understandable way to extract citations from a query, resulting in a quick search result.

The case for entity search also favours the new search engine. With the new search engine, fuzzy searches have allowed the search engine to return results of names that are not a 100% match to the query. For example, two of the tested queries, Fong Kwok San and Lai Chi Ying, are both able to return judgments related to Fong Kwok Shan and Lai Chee Ying (the correct spellings).

3.4.2 Query-driven summarization

The precision for performing query-driven summarization with sBERT were shown in the table below. The total sample size for the evaluation is 6. To test how well the algorithm performs across different categories of judgments, the judgments have come from different categories, ranging from criminal law and company law to constitutional law and tort law. The results for topic modelling are not available yet, and will be supplemented in the future, including during the final presentation.

<table>
<thead>
<tr>
<th>Judgment</th>
<th>Query</th>
<th>Precision (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002 HKCFI 156</td>
<td>name of the company</td>
<td>13%</td>
</tr>
<tr>
<td>2003 HKCFA 43</td>
<td>land use</td>
<td>71%</td>
</tr>
<tr>
<td>2016 6 HKC 58</td>
<td>decision being challenged</td>
<td>20%</td>
</tr>
<tr>
<td>2017 HKCFA 4</td>
<td>details of the expression</td>
<td>25%</td>
</tr>
<tr>
<td>2018 HKCA 282</td>
<td>price of goods</td>
<td>80%</td>
</tr>
<tr>
<td>2018 HKCA 602</td>
<td>amount of drugs</td>
<td>80%</td>
</tr>
</tbody>
</table>

In this test, some examples of searches above have pushed the functionality of the model to its limits - some of the queries, such as decision being challenged in an appeal case in 2016 6 HKC 58, as well as details of the expression in 2017 HKCFA 4, require a high level of understanding between sentences and paragraphs, as the latter is looking for the details of the expression made by the defendant in a freedom of expression case. This is exemplary of the weakness of sentence transformers in summarization, as there is a lack of cohesion between sentences in this approach. The system is also sometimes unable to identify and understand the full context of a sentence, resulting in a highlighting of adjacent sentences but not the most important sentence in a paragraph. For example, in the query “land use”, the system is able to highlight the sentence prior to “It was zoned for residential use”, but not this sentence presumably because this sentence does not mention the word “land” but instead used “It” to refer to “the land”.
in the previous sentence. This is shown in Figure 3.10.

Figure 3.10: Example of a missed sentence from the summarizer

The facts
8. The detailed facts can be found in the judgments of the Tribunal and the Court of Appeal, and there is no need to set them out again in full. In each case the land, which was resumed in 1999 for public housing, was unimproved agricultural land in the New Territories and was currently either vacant or used for open storage. It was zoned for residential use, close to an urban area with road frontage and suitable for residential development. In each case the land was held under a Crown (now a Government) lease, was

This is also a representation of the limits in sentence transformers. Nonetheless, the system tends to work well when the query is related to something factual, such as “details of XXX” or “amount of YYY”.

3.5 Challenges Faced

In a nutshell, the biggest challenges in this project have come from the lack of manual labelled data, lack of structured data in HKLII, and extensive labor needed for evaluation.

Lack of manual labelled data. Summarization tasks, in particular aspect-driven summarization, require extensive labelled data in different categories of judgments if a general-purpose summarization were to be achieved. As data is only available in drug trafficking cases and personal injury cases, it is very difficult to create accurate aspect-driven summaries for all judgments in HKLII. Availability of labelled data can also help semantic search in classifying the category of a judgment, but since such data is not available, only methods that do not require labels can be used.

Lack of structured data. Apart from the title and HTML body, there were no raw texts and sections of header, body and footer available on request. Manual parsing has to be done to segment the header and footer (with details such as the name of the judgment, names of plaintiff and defendant and representations) from the body before sentence embeddings were created, since performing sentence embeddings on such sections do not make sense. HTML parsing is also needed before the sentences can be passed into the model, as the HTML tags in the data do not represent anything regarding the meaning of a sentence. After the sentences were compared, the model will output sentences without HTML tags, but the highlighting of sentences, which is done through wrapping the sentence with a <div> tag with a specific class that highlights the words, requires a string matching that includes HTML tags between the sentences. For example, after the model determined that “Lorem ipsum dolor sit amet, consectetur adipiscing elit” is a relevant sentence, it needs to know that the highlighting should be wrapped between the substring “Lorem ipsum dolor <b> sit </b> amet, consectetur adipiscing elit”.

Hence, the searching task becomes very difficult as there is no information from the model on where the sentence offset is in the HTML of the judgment. Two workarounds are proposed, with the first being that regular expressions were used to find the text, by substituting every space character in the string obtained from the model to “,*”, but the execution has proved to be highly efficient due to lengthy regular expressions. The other workaround is to record
the offset of the sentence when the sentence is being processed to plain text, but this would require tremendous effort and is not being implemented in the prototype. As such, while most sentences can be highlighted, sentences which have HTML tags in between cannot be found and highlighted.

*Extensive labor needed for evaluation.* Evaluation for semantic search and document summarization, in particular semantic search, required special personnel to complete as the queries often contain legal terms that are well understood by people with a legal background. The process of conducting the experiments is also highly time-consuming, as the semantic search experiments require users to look at a maximum of 20 documents for every query, and manually labelling sentences relevant to a query in a document for document summarization. Combined with the tight schedule for the design and implementation of semantic search and summarization algorithms, the evaluation process is too time-consuming for this project to be completed on time. The lack of efficient evaluation results has also, in many cases, prevented a supervised learning approach where a ground truth is needed for parameters to improve in the network, but structures like Detext from LinkedIn have shown that a MLP is an effective way in combining scores from different algorithms before arriving at a final score for each judgment.
4 Conclusion

As the current HKLII system does not support searches beyond lexical keyword searches, users have found it challenging to find suitable judgements, especially with concept searches, on the current website. This project targets to improve the user experience in searching and reading documents by introducing semantic search and document summarization, which, on one hand, ranks judgements more sensibly in query results, and on the other, shortens the time needed to understand the contents of a document, and hence, the relevance of it against the user’s needs.

While the quantitative results were not large-scale and comprehensive, qualitative results have suggested that the current implementation of semantic search and document summarization has potential to provide an improvement in user experience for both HKLII and HKLII-v2. It is hoped that fine-tuning of the sentence transformers and models used in this project can continue to be conducted beyond this project, along with more comprehensive quantitative evaluations of the performance of the tasks. More features are also expected to be fed into both semantic search and document summarization, as discussed earlier. For semantic search, it is worthwhile to explore the possibility of classifying the category of the query alongside the use of sentence transformers, so as to narrow the category of judgments that the user expects.

For document summarization, alternative network structures for query-driven summarization can be explored, as the limitations of both topic modelling and sBERT were discussed in earlier chapters. Aspect-driven summarization can be done for more categories by feeding in more manual judgments with attributes and labels. Every judgment can be classified into a category by performing clustering or applying a neural network, determining the attributes expected in this judgment. The attributes can then highlight sentences that are the most similar to those highlighted manually in the data set. This is a method that can potentially be successful even with a very small highlighted dataset.

Taking into reference the structure of the neural network and the method of training from LinkedIn Detext, it is hoped that the models can be rolled into production once they are deemed mature enough, as online metrics such as click-through rate require less human labour dedicated to evaluation and tend to be more scalable. The feedback retrieved from the users can also, in turn, improve the neural network, prompting the network to return better results as it is being used more.
References


[7] L. Page, “The PageRank Citation Ranking: Bringing Order to the Web,”.


