VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

VUONG THI HAI YEN

MODELING AND LEARNING TEXTUAL AND STRUCTURAL RELATIONS FOR DEEP LEGAL INFORMATION RETRIEVAL

DOCTOR OF PHILOSOPHY IN INFORMATION TECHNOLOGY DISSERTATION

Hanoi, 2024

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

VUONG THI HAI YEN

MODELING AND LEARNING TEXTUAL AND STRUCTURAL RELATIONS FOR DEEP LEGAL INFORMATION RETRIEVAL

Major: Information Systems

Code: 9480104.01

DOCTOR OF PHILOSOPHY IN INFORMATION TECHNOLOGY DISSERTATION

SUPERVISORS:
1. Asocc. Prof. Phan Xuan Hieu
2. Prof. Nguyen Le Minh

Hanoi, 2024

Abstract

With the recent advances in digitalization and digital transformation, legal professionals can now easily access a huge volume of online legal materials. This is extremely important because judges and lawyers frequently need to find relevant legal information when they are working on a new legal case, performing legal research, case analysis, court preparation, giving legal advice to a client, developing a defense strategy, or making decision on a current case. However, the larger a legal database is, the more difficult for them to find relevant materials manually. In addition, legal documents like statutory law, case law or contract are normally lengthy and complex, consisting of multiple parts, chapters, sections, articles, and so on. Therefore, building an intelligent and automated legal information retrieval (IR) system is significant to improve and accelerate their legal process and workflow. Generally, this thesis aims to propose different legal IR methods and solutions based on an in-depth understanding of the nature and characteristics of legal data as well as the complexity of legal IR problems.

Accordingly, two major issues we need to consider carefully in this study are legal materials and legal IR problems. Legal materials are diverse, consisting of many different types of documents like constitution, statutory law, regulation, decision, case law, court document, contract, legal notice, patent, trademark, and so on. Among them, we focus on two main types of legal texts – statutory law and case law – because working on all types of legal materials is too broad and goes beyond the scope of the thesis. Regarding legal IR problems, this study focuses on three major IR tasks: (i) case law retrieval; (ii) statutory – case law retrieval; and (iii) IR–based legal question answering. The first task locates and returns case law documents from a case law database that relate and entail the decision of an input legal case. The second task retrieves statutory laws from a statutory law database that are relevant to a query case. And the third task seeks and returns statutory law articles that are likely to contain answers to a given legal question.

The three legal IR problems stated above are much more challenging than traditional IR for general-domain texts. The concept of relevancy in these tasks is no longer about keyword or topic matching. The similarity between legal texts requires the understanding of legal arguments and logical reasoning that are far beyond the lexical or topical comparison. In addition, while working with legal data, we realized that legal language is rigorous and complicated. Legal documents are normally lengthy and heavily rely on domain-specific terminologies, jargons, and linguistic nuances. Furthermore, there is a complex graphical structure hidden in any legal dataset that results from frequent mentions, citations, references within and between legal materials. Also, the style and content of legal documents highly depend on the domain and the legal system of each country. And one more important issue is that annotated data is limited because labeling for legal data requires a lot of human effort and domain expertise. All of these reasons are both the challenges as well as the motivations behind our study.

The main objective of this thesis is to enhance the performance and accuracy of the three legal IR problems by making the most of textual and structural relations in the legal data. First, we propose a supporting model that encodes both the lexical and legal relations at different levels of granularity to deal with the case law retrieval problem. In addition, we introduce a method to automatically create a large weak-labeling dataset to overcome the limitation of labeled data. Second, a heterogeneous legal knowledge graph was defined and constructed to leverage the statutory–case relationships in the statutory – case law retrieval. Third, the thesis presents a novel approach that builds an article reference network to uncover both local and long-range dependencies between legal articles to enhance the performance of the IR–based legal question answering. Moreover, throughout the thesis, we propose appropriate deep learning architectures to encode the textual and structural characteristics of legal data and combine them with powerful pre-trained language models to enhance the overall performance of the three IR problems. Besides the technical contributions, the literature review, the analysis, and discussions throughout this thesis would provide a deeper and clearer understanding of the nature and the limitations in legal NLP in general and in legal IR in particular. It would also be a potential reference for future studies in the field, particularly for low-resource language like Vietnamese.

Keywords: statutory law, case law, legal case, deep legal information retrieval, legal question answering, case law retrieval, statutory – case law retrieval, IR–based legal question answering, legal case entailment, supporting model, weakly labeled data, relevancy, textual relation, structural relation, legal knowledge graph, article reference network, pre-trained language model.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my thesis advisors Asocc. Prof. Phan Xuan Hieu and Prof. Nguyen Le Minh, for the continuous support of my Ph.D. study and related researches, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisors, I would like to acknowledge honorary Assoc. Prof. Ha Quang Thuy and Dr. Nguyen Ha Thanh, for their insightful comments and encouragement. Without their precious support, it would not be possible to conduct this research.

I also own special thanks to all members of the Data Science and Knowledge Technology Laboratory, The Department of Information Systems (VNU University of Engineering and Technology) and Nguyen's Laboratory (School of Information Science, Japan Advanced Institute of Science and Technology), who have been a source of friendships as well as good advice and collaboration.

Finally, with all my love, I would like to thank my family for all their love and encouragement.

Thank you!

Declaration

I hereby declare that this Doctoral Dissertation was carried out by me for the degree of Doctor of Philosophy under the guidance and supervision of my supervisors.

This dissertation is my own work and includes nothing, which is the outcome of work done in collaboration except as specified in the text.

It is not substantially the same as any I have submitted for a degree, diploma or other qualification at any other university; and no part has already been, or is currently being submitted for any degree, diploma or other qualification.

Hanoi, May 2024
Author

Vuong Thi Hai Yen

Table of Contents

LIST OF ABBREVIATIONS

Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
ALQAC	Automated Legal Question Answering Competition
BERT	Bidirectional Encoder Representations from Transformers
biLSTM	Bidirectional Long Short-term Memory
BM25	BM25 Ranking Algorithm
CCC	Case-Court-Case
CDC	Case-Domain-Case
CNN	Convolutional Neural Network
COLIEE	The Competition on Legal Information Extraction/Entailment
DNN	Deep Neural Network
DSSM	Deep Structured Semantic Model
FN	False Negative
FP	False Positive
GCNs	Graph Convolutional Networks
GloVe	Global Vectors for Word Representation
GRNs	Graph Nerual Networks
IR	Information Retrieval

KB	Knowledge-base
KG	Knowledge Graph
L2R	Learning to Rank
LLMs	Large Transformer-based Language Models
LSTM	Long Short-term Memory
MLP	Multilayer Perceptron
NeuIR	Neural Information Retrieval
NLP	Natural Language Processing
P	Precision
PROLEG	PROlog-based LEGal reasoning support system
QA	Question Answering
R	Recall
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
TF-IDF	Term Frequency – Inverse Document Frequency
TN	True Negative
TP	True Positive
VSM	Vector-Space Model

List of Legal Terminologies

Terminology	**Meaning** (in Vietnamese)
article	điều luật
case law	án lệ (luật dựa trên các lập luận, tiền lệ, phán quyết của các vụ án trước đó; bổ sung cho statutory law)
code	bộ luật (của luật thành văn – statutory law, statute law)
constitution	hiến pháp
counsel	luật sư; cố vấn pháp lý
courtroom proceedings	thủ tục xét xử
defense strategy	chiến lược bào chữa
dispute	tranh chấp; tranh luận
judge	thẩm phán; phán xét
judgement	phán xét
judicial decision	quyết định của toà án; quyết định tư pháp; bản án
jurisdiction	thẩm quyền tài phán; quyền hạn xét xử
jury	bồi thẩm đoàn
lawmaker	nhà lập pháp; người làm luật
lawsuit	vụ kiện
lawyer	luật sư
legal argument	tranh luận pháp lý; lập luận pháp lý
legal case	vụ án; vụ kiện
legal filings	hồ sơ pháp lý
legal proceedings	thủ tục tố tụng pháp lý
legislation	lập pháp; quá trình xây dựng và ban hành luật
litigation	kiện tụng
rulings	phán quyết
statutory law (statute law)	luật thành văn (được ban hành chính thức bởi chính quyền dưới dạng văn bản)
trial	phiên toà; phiên xử; việc xét xử; sự xử án

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of Research Context and Challenges

Artificial intelligence (AI) has made significant advancements across various fields including the legal industry. AI technologies are reshaping legal practices, eliciting substantial transformations in areas such as legal research, document analysis, contract review, and courtroom proceedings, etc. Thanks to its ability to process vast amounts of data quickly and efficiently, AI is transforming the way legal professionals work and practice as well as enhancing their experience and productivity.

Initially, AI methods in the legal field were primarily focused on knowledge representation and the development of rule-based systems. Between the 1970s and 1990s, pioneering projects in the intersection of AI and law were directed towards the formalization of legal arguments into formats that computers could process and computationally modeling. The aim was to create computer systems capable of understanding and manipulating legal concepts and reasoning. During this period, there was a growing interest in exploring the intersection of AI and law, leading to the establishment of the International Association for Artificial Intelligence and Law (IAAIL[1]), the first International Conference on Artificial Intelligence and Law (ICAIL[2]) in 1987, and the Artificial Intelligence and Law Journal (AILJ) in 1992 [2]. This journal has since become prominent for publishing advancements in the application of AI techniques to the legal domain. Since the year 2000s, there has been a significant transformation in the utilization of AI within the legal domain. Approaches based on machine learning have ascended to

[1] IAAIL: http://www.iaail.org

[2] ICAIL history: http://www.iaail.org/?q=page/past-icails

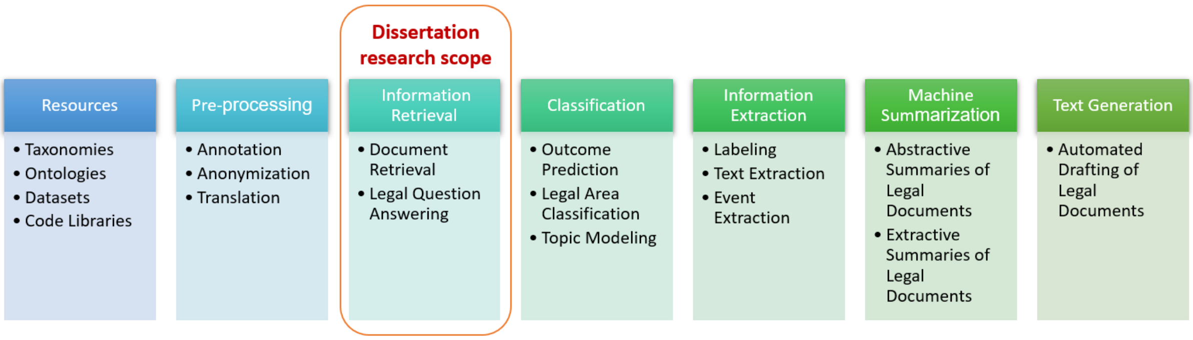


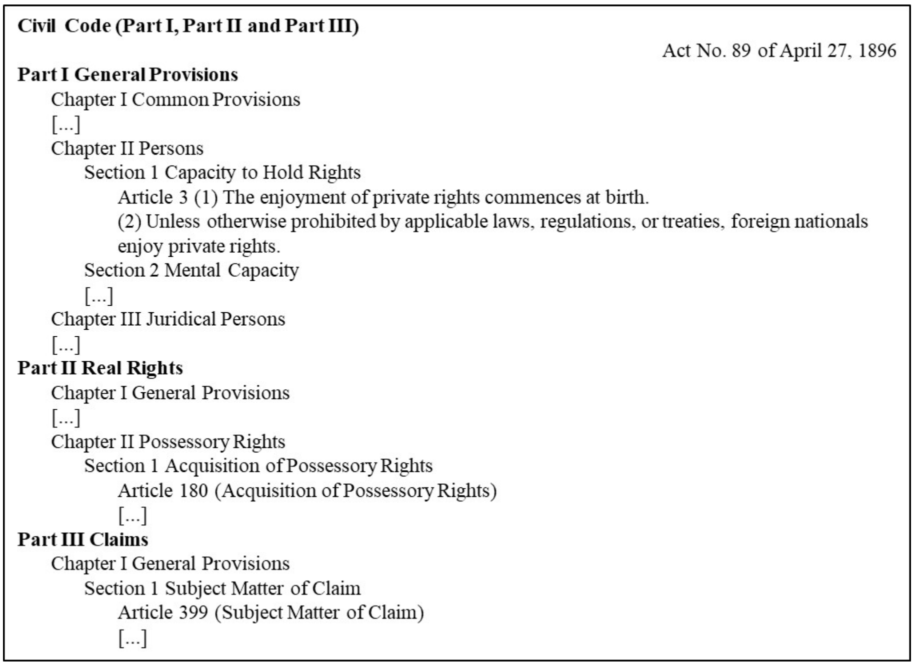
Figure 1.1: Categories and tasks in legal natural language processing

prominence within the sphere of AI and law, indicative of the technological progress in AI. In the recent years, a multitude of researches has published, utilizing (deep) machine learning and natural language processing (NLP) to augment both the performance and efficacy of the legal system through diverse methodologies.

There are a number of ways to classify legal NLP problems and applications from a lawyer's perspective. However, in this study, legal NLP tasks are categorized from a technical standpoint. Katz et al. divided legal NLP problems into seven primary categories including resources, pre-processing, information retrieval, classification, information extraction, summarization, and text generation as shown in Figure 1.1 [60].

Among these problem categories, **information retrieval** (IR) plays a critical role in legal NLP [57, 63] because reading, scanning, and locating desired information from lengthy legal documents are even difficult for a trained lawyer. In a large collection of legal materials, manual search for similar cases or supporting statutory articles for a given legal circumstance is extremely hard and time-consuming. That is why automated IR in this area is really meaningful. Technically, legal IR encompasses problems such as legal document retrieval and question answering. Document retrieval aims to locate relevant legal documents or even articles or paragraphs based on user queries or specific criteria, facilitating efficient access to a vast amount of useful legal information available. Legal question answering (QA), based on both IR and NLP techniques, involves developing systems that can analyze, comprehend, and respond to legal queries or questions by extracting relevant information from legal texts. This enables legal professionals and practitioners to obtain relevant, concise, and accurate answers to their legal inquiries.

It can be clearly seen that IR has a wide range of important applications and use cases in the legal field, serving as a foundational aspect of effective legal practice. This process not only supports to legal procedures but also accesses and ensures legal com-



Civil Code (Part I, Part II and Part III)

Act No. 89 of April 27, 1896

Part I General Provisions

Chapter I Common Provisions
[...]
Chapter II Persons
Section 1 Capacity to Hold Rights
Article 3 (1) The enjoyment of private rights commences at birth.
(2) Unless otherwise prohibited by applicable laws, regulations, or treaties, foreign nationals enjoy private rights.
Section 2 Mental Capacity
[...]
Chapter III Juridical Persons
[...]

Part II Real Rights

Chapter I General Provisions
[...]
Chapter II Possessory Rights
Section 1 Acquisition of Possessory Rights
Article 180 (Acquisition of Possessory Rights)
[...]

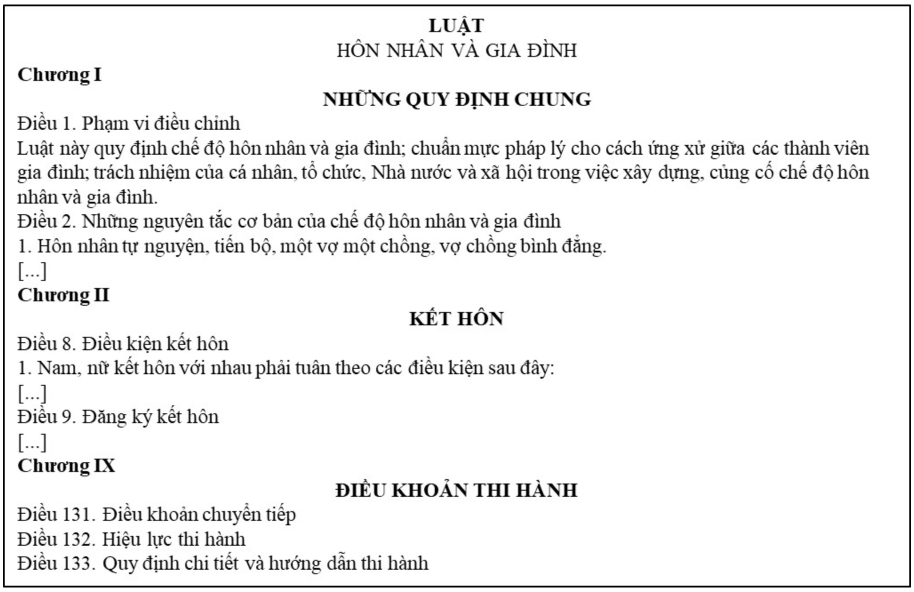
Part III Claims

Chapter I General Provisions
Section 1 Subject Matter of Claim
Article 399 (Subject Matter of Claim)
[...]

Figure 1.2: A sample of Japanese statutory (civil) law

pliance. Legal documents, encompassing statutes, case law, and legal filings, serve as the foundation upon which legal professionals build their arguments, strategies, and decisions in their current case. The ability to access and analyze these documents allows lawers, judges, and legal scholars to understand laws, precedents, and legal standards, ensuring that their actions and decisions are well-informed. Timely retrieval of relevant legal contents is also crucial for meeting court deadlines in the legal process. Moreover, legal QA systems not only saves time and resources for lawyers, judges, and legal experts, but also ensures that decisions and legal advice are based on comprehensive and accurate information. These systems further enhance people's access to justice a clearer understanding of the law and their rights.

In spite of various useful applications of legal IR, it is impossible to process and analyze legal texts effectively without a clear understanding of the nature and characteristics of legal materials. First, the legal documents are diverse, consisting of many different types like consitutions, statutory laws, decrees, decisions, regulations, case laws, contracts, memoranda, patents, and so on. Basically, the legal field is normally divided into two sectors: public and private law. Public law is about government and society that includes consitutional law, administrative law, and criminal law. Private law concerns disputes or litigations between two or more legal parties (both individuals and organizations) related to property, contracts, or commercial issues. Another way to classify legal documents is to put them into three major categories: legislation, court documents, and



LUẬT
HÔN NHÂN VÀ GIA ĐÌNH

Chương I

NHỮNG QUY ĐỊNH CHUNG

Điều 1. Phạm vi điều chỉnh
Luật này quy định chế độ hôn nhân và gia đình; chuẩn mực pháp lý cho cách ứng xử giữa các thành viên gia đình; trách nhiệm của cá nhân, tổ chức, Nhà nước và xã hội trong việc xây dựng, củng cố chế độ hôn nhân và gia đình.
Điều 2. Những nguyên tắc cơ bản của chế độ hôn nhân và gia đình
1. Hôn nhân tự nguyện, tiến bộ, một vợ một chồng, vợ chồng bình đẳng.
[...]

Chương II

KẾT HÔN

Điều 8. Điều kiện kết hôn
1. Nam, nữ kết hôn với nhau phải tuân theo các điều kiện sau đây:
[...]
Điều 9. Đăng ký kết hôn
[...]

Chương IX

ĐIỀU KHOẢN THI HÀNH

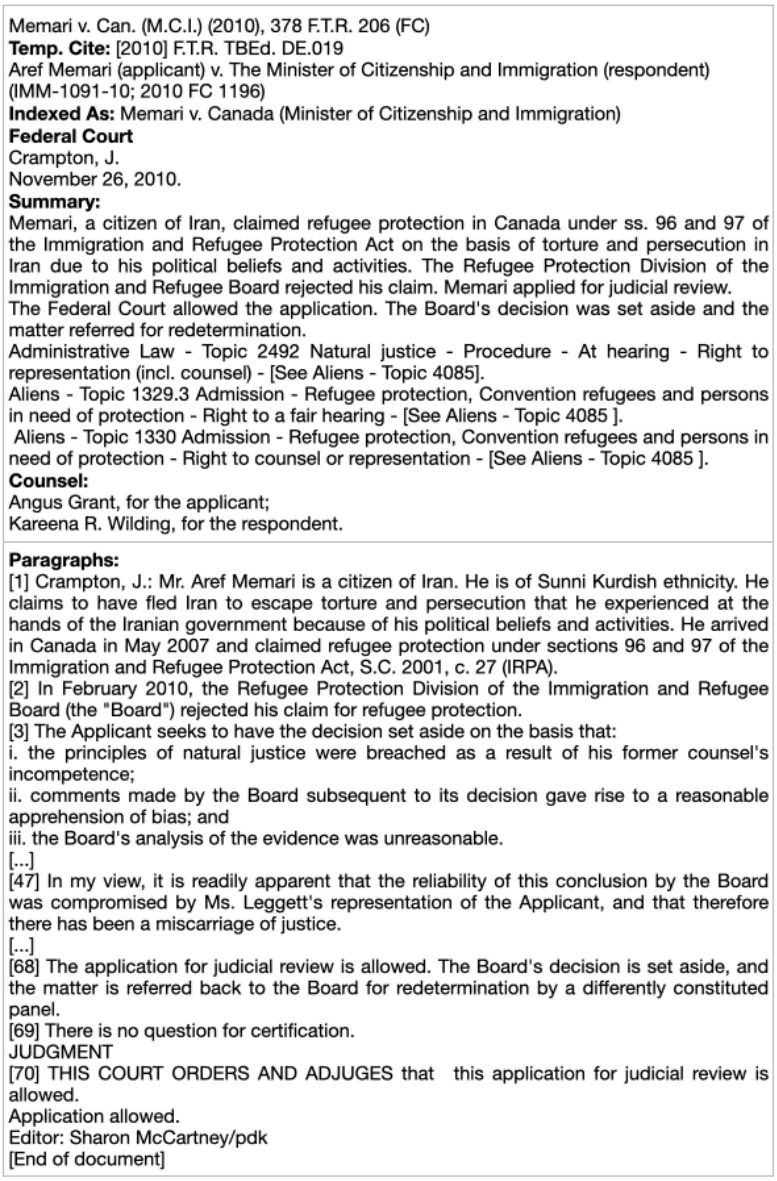
Điều 131. Điều khoản chuyển tiếp
Điều 132. Hiệu lực thi hành
Điều 133. Quy định chi tiết và hướng dẫn thi hành

Figure 1.3: A sample of Vietanmese statutory (marriage and family) law

legal agreements [24]. Legislative documents are prepared by lawmakers and issued by a legislature like parliament or congress depending on the political system of each country. Statutory laws, the most well-known legislative documents, are the foundation for litigation, trial, or any legal activities in different areas of civil and criminal laws. Figure 1.2 and Figure 1.3 show two examples of Japanese and Vietnamese statutory laws (i.e., legislation), respectively. Court documents are any written materials related to legal cases and proceedings. Case law is a popular type of court documents that describes the details of a legal case including the meta-data (e.g., court name, summary, counsel, etc.) as well as the content, the court judgment, and the court decision. Figure 1.4 shows a case law example from the Federal Court of Canada case law database. The content of a case law normally consists of multiple paragraphs. Finally, legal agreements are documents involving contracts between two or more parties about any business issues.

In order to foster and stimulate futher researches and applications in this field, several workshops and conferences on legal NLP and IR are organized annually or biannually like NLLP[3], JURISIN[4], and ICAIL[5]. Various datasets have also been published for model training and evaluation. Chinese AI and Law (CAIL) 2018 [155], dataset containing more than 2.6 million criminal cases from the Supreme People's Court of China, was published for judgment prediction. Duan et al. introduced the Chinese judicial reading comprehension (CJRC) dataset comprising approximately 10,000 documents and

[3] The Natural Legal Language Processing Workshop: https://nllpw.org/workshop
[4] The International Workshop on Juris-Informatics
[5] The International Conference on Artificial Intelligence and Law: https://dl.acm.org/conference/icail



Memari v. Can. (M.C.I.) (2010), 378 F.T.R. 206 (FC)
Temp. Cite: [2010] F.T.R. TBEd. DE.019
Aref Memari (applicant) v. The Minister of Citizenship and Immigration (respondent)
(IMM-1091-10; 2010 FC 1196)
Indexed As: Memari v. Canada (Minister of Citizenship and Immigration)
Federal Court
Crampton, J.
November 26, 2010.
Summary:
Memari, a citizen of Iran, claimed refugee protection in Canada under ss. 96 and 97 of the Immigration and Refugee Protection Act on the basis of torture and persecution in Iran due to his political beliefs and activities. The Refugee Protection Division of the Immigration and Refugee Board rejected his claim. Memari applied for judicial review.
The Federal Court allowed the application. The Board's decision was set aside and the matter referred for redetermination.
Administrative Law - Topic 2492 Natural justice - Procedure - At hearing - Right to representation (incl. counsel) - [See Aliens - Topic 4085].
Aliens - Topic 1329.3 Admission - Refugee protection, Convention refugees and persons in need of protection - Right to a fair hearing - [See Aliens - Topic 4085].
Aliens - Topic 1330 Admission - Refugee protection, Convention refugees and persons in need of protection - Right to counsel or representation - [See Aliens - Topic 4085].
Counsel:
Angus Grant, for the applicant;
Kareena R. Wilding, for the respondent.

Paragraphs:
[1] Crampton, J.: Mr. Aref Memari is a citizen of Iran. He is of Sunni Kurdish ethnicity. He claims to have fled Iran to escape torture and persecution that he experienced at the hands of the Iranian government because of his political beliefs and activities. He arrived in Canada in May 2007 and claimed refugee protection under sections 96 and 97 of the Immigration and Refugee Protection Act, S.C. 2001, c. 27 (IRPA).
[2] In February 2010, the Refugee Protection Division of the Immigration and Refugee Board (the "Board") rejected his claim for refugee protection.
[3] The Applicant seeks to have the decision set aside on the basis that:
i. the principles of natural justice were breached as a result of his former counsel's incompetence;
ii. comments made by the Board subsequent to its decision gave rise to a reasonable apprehension of bias; and
iii. the Board's analysis of the evidence was unreasonable.
[...]
[47] In my view, it is readily apparent that the reliability of this conclusion by the Board was compromised by Ms. Leggett's representation of the Applicant, and that therefore there has been a miscarriage of justice.
[...]
[68] The application for judicial review is allowed. The Board's decision is set aside, and the matter is referred back to the Board for redetermination by a differently constituted panel.
[69] There is no question for certification.
JUDGMENT
[70] THIS COURT ORDERS AND ADJUGES that this application for judicial review is allowed.
Application allowed.
Editor: Sharon McCartney/pdk
[End of document]

Figure 1.4: A case law sample from The Federal Court of Canada case law database

close to 50,000 questions [41]. LMTC [26] is large-scale multi-label legal text classification that contains 57,000 legislative documents from EUR-LEX and annotated with approximately 4,300 EUROVOC labels. JEC-QA is an extensive question answering dataset from the National Judicial Examination of China [166]. Liu introduced a dataset for named entity recognition in German federal court decisions including about 67,000 sentences and over 2 million tokens with 54,000 entities belonging to 19 semantic categories [78]. Additionally, competitions and shared tasks are also organized every year to challenge research teams working on legal NLP and IR all over the world. COLIEE, the competition on legal information extraction and entailment, is an annual event designed to foster advancements in the field by challenging participants to develop systems capable of retrieving, extracting relevant legal information and determining legal entailment from vast corpora of legal texts [58, 111, 112]. They release a database of predominantly Federal Court of Canada case laws, provided by Compass Law. Especially, an automated legal question answering competition (ALQAC) provided a legal QA dataset based on the statutory laws of Vietnam [143].

Being aware of the potential of AI applications in the legal field, the legal NLP research community has been growing significantly in the last few years. Legal NLP has also become the topic of various doctoral theses. Nguyen improved the attention mechanism in deep neural networks to deal with different problems in legal text processing [92]. Liga proposed the use of hybrid AI models to extract patterns and rules from argumentative and legal texts [77]. Chalkidis proposed the use of deep neural network to discover useful information from legal texts [24]. Horton introduced a conceptual framework for the law and technology knowledge domain [50]. And, Navas Loro addressed important issues in legal NLP, that are the identification and representation of temporal expressions and events in legal documents [90].

However, although there are more resources (e.g., data, computing power) and the emergence of advanced processing and learning methods in recent years, legal NLP in general and legal IR problems in particular are still very challenging. While working with legal documents, we have observed and realized that there are a number of reasons making legal IR problems really hard. They can be the complexity of the legal language, the limitation of labeled data, the complex references, ciations among legal entities, etc. These reasons are both the major challenges for our study as well as the motivations behind our proposed approach and methods. Let us go into more detail and discuss them from the viewpoint of legal IR research.

The first reason is that **the legal IR problems themselves are difficult to solve**. They require deeper processing and analysis than IR for general-domain texts because legal language is much more logical and rigorous. IR and QA for legal information, therefore, need to be capable of legal reasoning and inference. With legal IR and QA, the traditional concept of relevancy is not enough and the retrieval results are beyond the merely lexical or topical relevance. To see how challenging they are, let us consider the shared tasks of the Competition on Legal Information Extraction/Entailment (COLIEE) organized annually in recent years by AMII[6], University of Alberta, NII[7], vLex Canada[8], and other partners. The first task of COLIEE is “The Casw Law Retrieval”

[6] Alberta Machine Intelligence Institute: https://www.amii.ca

[7] National Institute of Informatics, Japan: https://www.nii.ac.jp/en

[8] vLex Canada: https://ca.vlex.com

that involves reading a new case law Q, and retrieving supporting cases $\{S_1, S_2, \ldots, S_n\}$ for the decision of Q from the entire case law corpus. This *supporting* relation is non-trivial because legal experts examine and decide that supporting cases are only “noticed cases”, i.e., only those cases that are considered to be useful for the potential decision of the input case. This is totally different from the traditional IR. The concept of *relevancy* in legal IR should be defined based on the legal relation that goes beyond the lexical or topical relevance [126, 147]. This is a real challenge because normal text matching would not work. The legal relations can be supporting or against, usually containing logical argument and reasoning that need a deeper and more complicated formulation of similarity and relation between legal text segments.

The second task of COLIEE is “The Case Law Entailment” that involves the identification of a paragraph from a relevant case law that entails the decision of a new case. In other words, given a decision Q of a new case law and a relevant (i.e., supporting) case R, a specific paragraph of R that entails the decision Q needs to be identified. The COLIEE organizers confirmed that the answer paragraph cannot be identified merely by normal IR techniques because while many paragraphs in R can be relevant to decision Q, only a small fraction of them have the real legal *entailment* relation with the decision Q [111, 112]. In order to better understand the concept of legal entailment in the COLIEE competition, we need to connect to the study of textual entailment problem in NLP. Textual entailment [35, 36], also known as Natural Language Inference (NLI), is basically about understanding how one piece of text relates to another. Specifically, it focuses on whether the meaning of one text segment (called *the hypothesis*) can be inferred from the meaning of another (called *the text*). The textual entailment relation asks if, based on the meaning of the text, it is reasonable to believe or conclude the hypothesis is true. Textual entailment is an important part of NLP because it helps computers understand the semantic and logical relationships between different pieces of text. This is useful for many tasks, such as IR, QA, and machine translation. Similarly, COLIEE also has other challenging tasks related to IR and QA for statutory laws that require a profound understanding and a deep analysis of legal texts to accomplish. As introduced later in this thesis, our research problems about legal IR are highly relevant to and as complicated as the shared tasks of the COLIEE competition. That is why we consider and call them as **deep legal information retrieval** to emphasize the depth and the complexity of the legal IR problems we have to deal with in this dissertation.

The second reason making legal IR problems hard to handle is **the intricate nature of legal texts**. Clearly, the legal language has it own unique characteristics and complexity. The intricate nature of the legal system goes beyond being purely scientific [59, 122]. First, legal documents are normally written in legal language that is highly formal and relies heavily on domain-specific terminologies, jargons, and linguistic nuances that may not be present in everyday language. Some popular words are used with special meanings in the legal domain (e.g., 'action' refers to a lawsuit, 'sentence' means punishment, etc.). There are many terms borrowed from French and Latin [24]. This poses a challenge for both non-expert users and general NLP models to understand and interpret legal texts accurately. Legal texts may also exhibit ambiguity and vagueness, as they are subject to different interpretations and legal precedents. This adds an additional layer of complexity in accurately processing and analyzing legal language. Additionally, legal texts are diverse and cover various areas of law, including statutes, court cases, regulations, contracts, patents, memoranda, and other related materials. Moreover, legal documents can be complex, often containing multiple sections, clauses, references, and cross-references, making it difficult to normalize, parse, segment, and extract relevant information as well as establish connections between different parts of the texts. Also, legal documents are normally lengthy, consisting of several paragraphs and containing around 3000 words on average. Sentences in legal texts are also much longer, can be up to 60, 70 words per sentence comparing to 15-20 in common English texts [24]. This requires advanced machine learning techniques that are capable of representing longer contexts and capturing long-range dependencies in legal documents.

The third reason is **the complex connections among legal entities**. While a legal database can be measured in volume (i.e., the number of materials) and the language complexity, the hidden structure of the data really matters. Legal data is not flat because the documents are connected to each other in different ways [22, 32, 137]. Laws cite or mention other laws, articles consists of references to other articles. There are a lot of intra- and inter-document (and inter-article, inter-paragraph, etc.) citations. Furthermore, legal materials normally involve different types of entities like courts, cases, laws, and domains. Therefore, we can consider a legal database in two different views: a collection of texts and a heterogeneous graph of legal entities. That is, in addition to the legal text contents, the graphical characteristics of a legal data need to be examined and leveraged in order to boost the performance of the legal IR and QA problems.

Another important reason is **the limitation of annotated data**. Creating datasets in the legal domain faces several challenges. First, legal texts are often subject to copyright restrictions and privacy concerns, which restrict the availability and sharing of annotated or labeled legal datasets. Moreover, legal texts are diverse and cover various areas of law, including contracts, court cases, statutes, and regulations. Collecting a representative and comprehensive dataset that encompasses the breadth of legal topics and jurisdictions is a laborious and time-consuming task. Additionally, legal texts require expert knowledge for accurate annotation and labeling, as legal concepts and nuances may be challenging for non-experts to interpret correctly. Annotating legal datasets requires legal expertise, which may limit the availability of annotators and increase the cost and time required for dataset creation [28, 79, 117]. Furthermore, the dynamic nature of the legal field necessitates regular updates and maintenance of legal datasets to reflect changes in laws and legal precedents. Overall, these limitations make the creation of high-quality and diverse legal datasets a challenging endeavor, requiring collaboration between legal professionals, NLP experts, and access to relevant legal resources.

The last reason is **the locality of legal languages and systems**. One of the major issues in legal data is the diversity of languages. Although legal datasets exist in diverse languages, the linguistic properties vary significantly. It is challenging to transfer knowledge from one language to another. In this regard, the English language has a distinct advantage given its historical prominence as the de facto lingua franca in both computing and international business law. As a result, English predominates in the distribution of languages in public papers [28, 29, 112], with other major world languages such as Chinese [41, 155], German [71], and French [15] being the closest competitors. Moreover, legal systems and regulations vary across jurisdictions, which necessitates the development of country-specific and domain-specific models and resources to account for these differences. Building IR and QA systems for different languages, therefore, needs different solutions that should take the locality (i.e., the language, the legal system, and the legal domain) into account.

As stated earlier, although these reasons are the challenges for legal IR research, they are also the basis and the indication for us to propose new ideas and solutions to the legal IR problems later in this thesis.

1.2 Scope of Research

Before presenting the motivations and the objectives of the thesis, we need to clarify the scope of our research in terms of both legal text data and legal IR problems.

1.2.1 The Legal Data of Interest

As mentioned earlier, legal materials are diverse. There are various types of legal documents including consitutions, statutory laws, decrees, decisions, regulations, case laws, contracts, memoranda, patents, and so on. Depending on the legal sub-domains, they can be much different in length, structure, terminology usage, style of writing, and content. Dealing with all types of legal documents is complicated and goes beyond the scope of this study. In this dissertation, we aim to work with two main types of legal documents: **statutory law** and **case law**.

Statutory law (a.k.a. statute law or statute for short) refers to laws that are enacted by a legislative body, such as a parliament, congress, or state legislature depending on the political system of each country. These laws are normally created through the legislative process, which involves proposing, debating, amending, and ultimately passing bills that become statutes. Statutory laws can cover a wide range of subjects and areas of law, including administrative law, regulatory law, civil law, and criminal law. They are often codified into written codes or statutes, which are organized by subject matter and serve as the primary source of law in many legal systems. Figure 1.2 and Figure 1.3 show a sample of Japanese statutory (civil) law and a sample of Vietnamese (marriage and family) law, respectively. The structure of a statutory law may include parts, chapters, sections, articles, etc. depending on the legal system.

Case law refers to the body of judicial decisions and interpretations of law established through the court system. When legal disputes are brought before courts, judges make rulings based on existing laws, legal precedents, and interpretations of statutes. These rulings become part of the body of case law, which serves as a guide for future cases with similar legal issues. Case law plays a crucial role in the legal system because it helps interpret statutes, fills in gaps where legislation may be unclear, and provides consistency and predictability in legal outcomes. It is commonly cited and relied upon by lawyers, judges, and legal scholars to support legal arguments and decisions. In common law systems, such as those in the United States and the United Kingdom, case law is a significant source of law alongside statutes and regulations. Judges in these systems are bound to follow precedent set by higher courts within the same jurisdiction, creating a hierarchical structure of legal authority. Figure 1.4 shows a case law sample from The Federal Court of Canada case law database. It contains the meta information (e.g., citation, indexing terms (indexed as), court name, time, summary, etc.) and the body with multiple paragraphs describing different aspects of the case law.

Additionally, it is worth to emphasize that **legal case** and **case law** are closely related to each other but they are two different concepts. A legal case refers to a dispute or controversy brought before a court of law for resolution. It involves parties presenting arguments and evidence to support their positions, and a judge or jury rendering a decision based on the applicable law and facts presented. A legal case can encompass various types of disputes, including civil cases and criminal cases. A case law, on the other hand, refers to the body of law created by judicial decisions in an individual case. When judges issue rulings or opinions in legal cases, they often provide interpretations of statutes, regulations, and legal principles that become precedent for future cases. Case law forms an essential part of the legal system, as it helps guide future decisions, establish legal principles, and interpret the law.

Regarding the experimental data, we used the COLIEE 2019 and 2021 datasets [111, 112] that include subsets for Task 1 (The Legal Case Retrieval) and Task 2 (The Legal Case Entailment) for the Case Law Retrieval problem in Chapter 3. In Chapter 4, we built a legal knowledge graph for Vietnamese from 9,000 legal cases, 255 statutory law documents, 693 courts, and 185 legal domains. Chapter 5 used the Task 3 (The Statute Law Retrieval) datasets of the COLIEE 2021 and 2022 [112, 113] for experiments. In Chapter 5, we also used two Vietnamese datasets including the ALQAC (Automated Legal Question Answering Competition) 2021 [143] and a QA dataset for Vietnamese civil law with 5,000 labeled data samples [62].

1.2.2 The Deep Legal Information Retrieval Problems

IR and QA for legal texts are any tasks related to retrieving information relevant to an input query or finding a correct answer to an input question. There are also various types of legal documents. Therefore, we can have different ways to define IR and QA tasks. However, as mentioned in the previous subsection, in the scope of this study, we only work with two main types of legal materials: statutory law and case law. Accordingly, we will limit three primary IR and QA problems for these two types of legal documents in this thesis. Those problems are:

(i) Case law retrieval;

(ii) Statutory – case law retrieval;

(iii) IR–based legal question answering.

All of our proposed ideas and methods as well as our technical contributions in this thesis are around these three IR and QA problems. We will address here these problems in more detail for a precise understanding of what they are since we will encounter them frequently throughout this thesis.

1.2.2.1 Case Law Retrieval

In legal practice, there are many different scenarios when a judge or a lawyer need to find a similar case law. Typically, it occurs during the process of legal research, court preparation, or when making a decision on a current case. Also, judges often look for similar cases to establish legal precedent. If a case involves similar facts or legal issues to those in previous cases, the judge may examine those precedents to guide their decision-making. Another scenario is that judges may search for cases that have addressed similar issues or questions of law when they need to interpret statutes or legal principles. Analyzing how those cases were decided can help the judge interpret the law correctly. Additionally, judges may use analogical reasoning to apply legal principles from existing case law to new situations. They search for cases with analogous circumstances to draw parallels and make informed decisions. Regarding court proceedings, lawyers may cite similar case law to support their arguments. Judges may then review those cases to evaluate the strength of the legal arguments presented. That is why locating and retrieving case laws that are really relevant to a current circumstance from a large legal database are critical.

With the recent advances in digitalization and digital transformation, judges and lawyers can now easily access a huge volume of online legal materials. However, the larger number of legal documents is, the more difficult to find most relevant case laws. Therefore, developing an automated case law retrieval system will significantly accelerate and improve the performance of the judge's and lawyer's workflow. The case law retrieval problem was defined to meet this need. This problem consisits of two phases (or two sub-tasks) that are the Task 1 (The Legal Case Retrieval) and Task 2 (The Legal Case Entailment) of the COLIEE competition [58, 111, 112], respectively. Figure 1.5 shows the two phases and the logical flow of the case law retrieval problem.

The legal case retrieval phase: Let $\mathbf{C}$ be the space of all possible legal cases and case laws and let $C \subset \mathbf{C}$ be a corpus of case laws (i.e., a case law database). Given an input query case $c_q \in \mathbf{C}$. The query c_q is normally a new legal case that a judge or a lawyer is currently working on. The aim of this phase is to locate and retrieve a set of all

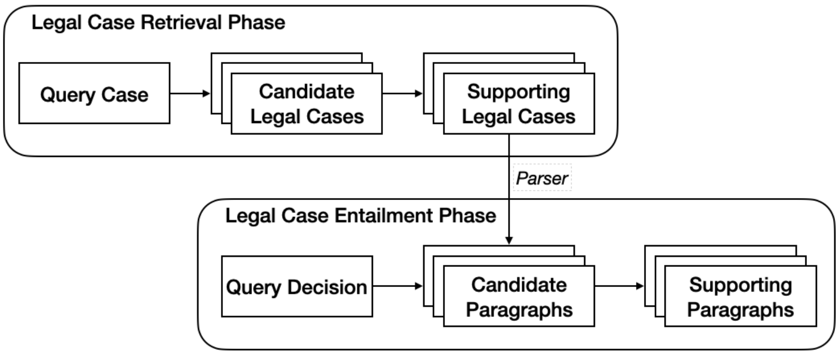


Figure 1.5: The logical flow of the case law retrieval problem

relevant case laws $C^r = \{c_1^r, c_2^r, \ldots, c_k^r\} \subset C$ that support the decision of c_q. In the legal domain, these supporting cases $c_1^r, c_2^r, \ldots, c_k^r$ are also called "noticed cases". Technically, this case retrieving phase can be expressed as the following mapping:

$$f_{case\_retrieval}(c_q, C) \rightarrow C^r \tag{1.1}$$

The legal case entailment phase: Given a triplet including the input query case c_q, a decision d_q of the query case c_q, and the list of all supporting cases C^r returned from the previous phase. Let P^r be the set of all text paragraphs being segmented from a given supporting case $c^r \in C^r$. The aim of this phase is to identify a set of supporting paragraphs $P^e = \{p_1^e, p_2^e, \ldots, p_l^e\} \subset P^r$ that entail the decision d_q of the query case c_q. Technically, this case retrieving phase can be expressed as the following mapping:

$$f_{case\_entailment}(c_q, d_q, P^r) \rightarrow P^e \tag{1.2}$$

As explained earlier in Section 1.1, the entailment relation between two legal text paragraphs is similar to the concept of textual entailment in natural language understanding and inference [35, 36]. That is the relationship between two text segments where one (called *the hypothesis*) can be inferred or implied by the other (called *the text* or *premise*). In other words, if the text is true, then the hypothesis is likely to be true as well. Both the supporting in the first phase and the entailment in the second phase are complicated relations that are based on legal and logical reasoning. They are much deeper and go beyond the normal concept of relevancy in traditional IR that is merely based on the lexical and topical proximity. That is why we call and consider these tasks as **deep legal information retrieval** problems. These problems will be described and discussed in more detail and the solution will be proposed in Chapter 3 of the thesis.

1.2.2.2 Statutory – Case Law Retrieval

Statutory law, also referred to as statutes or codes, forms the foundation of the legal system in many countries. These are the written laws passed by legislative bodies that govern a particular jurisdiction. Thus, judges and lawers need to find relevant statutory laws whenever they are working on a legal case (i.e., the current case); and they need to perform this search at various stages of the legal process. For example, when building or preparing a case, a lawyer representing a client needs to identify specific laws that apply to the situtation and how they might be used to argue their case. A comprehensive understanding of the relevant statutes helps them build their arguments, prepare witness examinations, anticipate potential legal issues, and develop their overall legal and trial strategy. Another need is legal analysis. Throughout the course of a legal proceeding, judges may need to interpret and apply statutory laws to resolve disputes between parties. Therefore, judges must carefully analyze the language of the statutes and apply legal principles to the specific facts of the case. And when making a ruling, a judge presiding over a case needs to determine which statutes are relevant to the facts presented and how they should be interpreted in reaching a decision. There are still many other situtations judges and lawyers need to do this search when handling appeals, giving legal advice, and so on. That is why locating and retrieving statutory laws pertinent to a particular legal inquiry is extremely important. Technically, this problem is addressed as follows.

Let S be a statutory law corpus (i.e., a database of statutory laws). Given an input query case c_q (normally, c_q is the new legal case that judges and lawyers are currently working on), the aim of this problem is to locate and retrieve all statutory laws $S^r = \{s^r_1, s^r_2, \ldots, s^r_k\}$ from the corpus S that are most relevant to the query case c_q. This can be expressed as the following mapping:

$$f_{law\_retrieval}(c_q, S) \rightarrow S^r \tag{1.3}$$

This problem will be described and discussed in more detail and the solution will be proposed in Chapter 4 of the thesis.

1.2.2.3 IR–based Legal Question Answering

There are many times that judges and lawyers need to ask a question related to any legal issues and expect to obtain specific legal articles that are most relevant and likely to answer their question. This is crucial for legal research, court preparation, legal reasoning, legislative drafting, litigation, and legal scholarship [4, 72, 98]. For example, when preparing for a case, lawyers and judges often conduct research to find relevant statutes and regulations. Sometimes, they need to examine specific articles within those statutes to understand how they apply to the case at hand. In several cases where the interpretation of a specific statute is in question, lawyers and judges may need to delve into the articles within that statute to analyze its language, context, and legislative intent. Also, judges may need to examine how previous cases have interpreted specific articles within laws to determine the applicability of those interpretations to the current case. Another usecase is that lawyers may need to reference specific articles within laws when drafting legal documents such as contracts, wills, or business agreements to ensure compliance with relevant regulations. Finding relevant laws at the article level, therefore, is frequently performed in legal practice. However, reading and locating specific answers from a lengthy statutory law document is laborious and time-consuming. That is why an automated QA solution that can retrieve and extract most relevant articles from lengthy statutory law documents is critical to the modern legal process. We call this problem "IR–based legal question answering". The problem is formally stated as follows.

Let A be a corpus (i.e., a database) of statutory law articles. Given a question q about any legal issues that can be covered by the corpus A, the goal of this problem is to find the most relevant statutory articles $A^r = \{a^r_1, a^r_2, \dots, a^r_k\}$ from the corpus A that are most likely to contain the answers to the input question q. This can be expressed as the following mapping:

$$f_{statute\_retrieval}(q, A) \rightarrow A^r \tag{1.4}$$

This problem will be described and discussed in more detail and the solution will be presented in Chapter 5 of the thesis.

1.3 Motivations and Objectives

1.3.1 Research Motivations

As mentioned and discussed earlier in Section 1.1, there are several reasons why legal IR problems are still challenging. They can be the complexity of the problems, the intricate nature of the legal texts, the complex graphical connections among legal entities, the limtation of labeled data, and the locality of legal languages and legal systems. Generally, all of these reasons are partly the motivation behind our study. However, in this section, we will take a closer look at the nature of the legal IR problems, the characteristics and the representation of legal data, as well as the previous studies in order to identify what we should focus on to enhance the performance of the legal IR problems.

First of all, in the legal IR, the concept of relevancy between legal texts are not just about keywords or topics. A legal document mentions a specific term of a query does not necessarily mean that document is relevant to the query. Legal IR requires more than just lexical or topical matching. Legal IR systems must consider the context of legal documents including statutory language, judical opinions, legal precedents, etc. and understand the nuances of legal arguments and reasoning in order to accurately retrieve relevant information. However, legal documents are normally lengthy (around 3,000 words on average) and have complex hierarchical structures (e.g., parts, chapters, sections, articles, etc.). A single statutory law or case law document with multiple text paragraphs can mention various concepts or entities and describe different situations in different contexts. As a result, modeling the whole document, even with advanced embedding techniques, is still a coarse-grained representation. Unfortunately, when doing legal research or preparing a new case, judges and lawyers often need to retrieve and locate concise and specific information, normally at the paragraph or article level. Actually, two of the legal IR problems (Case Law Retrieval and IR–based Legal Question Answering) stated earlier in Section 1.2.2 require to return results at paragraph (entailment) and article level. This is natural because the supporting and entailment relations are defined between text paragraphs. However, most related studies about the case law retrieval and entailment tasks based on deep neural networks only modeled and learnt at the level of case law document [110, 132, 145]. Shao et al. considered the paragraph-level interactions but still integrated them into their model as a whole document [132]. This is important because the way we represent and learn legal textual relations significantly influences the model architecture, the retrieval performance, as well as the conciseness of the results.

The second important issue is the limitation of labeled data in this domain. As discussed earlier in Section 1.1, data selection and annotation for legal documents requires much legal expertise and thus this is an expensive and time-consuming activity [28, 79, 117]. In order to label the relevance, supporting, entailment relations between query legal cases and case law or statutory law documents, legal experts need to read every single piece of texts and consider whether there are relations with others or not.

That is why most of the labeled legal corpora are small or medium. However, most of the recent advanced learning models are based on deep neural networks that require to learn from large labeled data to perform well. Unfortunately, no previous studies proposed a solution to overcome the short of annotated data for the legal IR problems.

Like scientific literature, legal materials frequently refer to each other. Laws cite or mention other laws, articles include references to other articles. Therefore, legal documents are connected in different ways with a lot of intra- and inter-document relations [22, 32, 137]. Furthermore, legal materials also involve different concepts or entities like courts, cases, legal parties, laws, judges, lawyers, domains. These characteristics form a complicated graph or network structure in any legal database. Obviously, modern legal IR methods should to make the most of this structural information to enrich the IR models and enhance the IR performance. To this end, some previous work attempted to build knowledge graph for case law retrieval [39, 137]. However, there were no studies trying to build a heterogeneous knowledge graph with different types of entities to improve the performance of the statutory – case law retrieval problem. For the IR–based legal question answering, most of the previous works relied on the lexical and semantic models [96, 158, 160]. No one utilized the network of references between legal articles to improve the answering accuracy. From our observation, the structural legal relations (i.e., mentions, citations, references, etc.) are rich features that are significant for connecting legal elements and therefore help improving the retrieval and answering accuracy.

Finally, representing and learning textual and structural relations in legal data are important issues. We need to integrate these rich features into a unified model to solve legal IR problems. For each problem, we need to introduce an appropriate model architecture that can encode textual and structural characteristics of legal data. To this end, pre-trained language models such as Multilingual-BERT[9] and mono-T5[10] should be utilized to embed and make the most of structural relations mentioned above.

1.3.2 Research Questions and Objectives

Based on the research challenges and motivations, as well as what have been done in the previous studies and what remain unsolved, this thesis addresses the following research questions:

[9] Multilingual-BERT: https://huggingface.co/bert-base-multilingual-cased

[10] mono-T5: https://huggingface.co/castorini/monot5-base-msmarco

- **Q1**: How will complex and lengthy legal documents be processed and represented? How to formulate and learn the textual legal relations and similarity between legal texts at different levels of granularity (case, paragraph, decision . . .) to enhance the relevancy and accuracy for the IR and QA problems?

  This question will be clarified in Chapter 3. We represent support relations at the case-case and paragragh-paragraph levels to solve the case law retrieval problem.

- **Q2**: How to overcome the limitation of annotated legal data in the legal IR and QA problems? How can we have more labeled data in this domain to improve the retrieval performance?

  This question will also be discussed and clarified in Chapter 3 and Chapter 5 where we propose a data agumentation method for creating weakly labeled data based on the assumption of supporting relations in legal texts.

- **Q3**: How can we represent and learn the structural rations that are the graphical connections among legal texts (e.g., local and long-range references) and the links among legal entities (e.g., courts, cases, laws, domains) to help enhance the performance of the IR and QA problems?

  This question is addressed throughout the dissertation. The question is clarified in Chapter 4 where we introduce a heterogeneous knowledge graph for statutory – case law retrieval and in Chapter 5 where we propose an article reference network for IR–based legal question answering.

- **Q4**: How to integrate and leverage the legal textual and structural characteristics with powerful deep learning models (including pre-trained language models) to improve the performance of the IR and QA problems?

  To address this question, we propose to integrate legal textual and structural information with pre-trained language models to tackle the IR and QA problems in Chapters 3, 4, and 5.

The overall goal of this thesis is to enhance the performance and efficiency of the legal IR and QA problems in different ways. Technically, we have three concrete objectives as folows:

- **O1**: Proposing new approaches and models to enhance the effectiveness of the legal IR and QA problems.

- **O2**: Leveraging and making the most of the nature and characteristics of legal data (i.e., both the textual and structural legal relations) to boost the performance of the three legal IR problems stated in Section 1.2.2: case law retrieval, statutory – case law retrieval, and IR–based legal question answering.

- **O3**: Proposing suitable methods to combine and integrate the textual and structural features of legal data with powerful deep learning models (including pre-trained language models) to further improve the efficiency of the legal IR and QA tasks.

1.4 Research Methodology

To address the research objectives and questions, this dissertation obeys the following research methods:

- Quantitative research methods involve the statistical analysis of legal text data, testing hypotheses addressed within the dissertation.

- Qualitative research methods focus on understanding the content and context of legal documents, identifying the strengths and weaknesses of current research approaches to propose new solutions and models to deal with the legal document retrieval and question answering problems.

- Experimental research methods were intensively used to conduct experiments in order to confirm the hypotheses and validate the accuracy and effectiveness of the proposed models.

Each method plays a critical role in problem understanding, designing and developing models for the legal document retrieval and question answering tasks. This dissertation integrates these research methods to leverage the strengths of each method and ensure a comprehensive, appropriate, and reliable evaluation of both the technical and practical aspects of developing NLP methods in the legal field.

1.5 Contributions

This dissertation offers significant contributions in different aspects: the learning representation of legal features, the data agumentation, the definition and creation of the legal knowledge graph, the uncovering and usage of graphical relationships, and the graph-inspired deep learning model integration. First, the dissertation focuses on exploring and representing the legal relations between texts at different levels of granularity to deal with lengthy documents as well as make the most of both lexical and complex logical relations into a so-called *supporting model* to solve the case law retrieval task. Second, we propose a weak-labeling strategy to overcome the short of annotated data and improve the retrieval efficiency. Third, we define and create *a heterogeneous knowledge graph* of different types of legal entities to boost the performance of the statutory – case law retrieval problem. We also define and build *a reference network* that captures and uitilizes the graphical connections or relationships among legal texts to enhance the performance of the question answering task. Moreover, throughout this dissertation, we propose deep model architectures to smoothly integrate both legal textual and structural characteristics of the legal data to improve the performance of the IR and QA models. The model architectures introduced in this Experimental research methods design and conduct experiments to validate the accuracy and effectiveness of the proposed models in the dissertation demonstrate better performance compared to the current benchmarks, with some achieving unparalleled results on established data collections. Performance enhancement demonstrated through the thorough experiments, analysis, evaluation elucidate the effectiveness of the proposed approaches and methods. Finally, the analysis and discussions throughout this work would help provide a deeper understanding of legal texts and processing problems, present the advancements and remaining limitations of legal NLP in general and legal IR and QA in particular; and would also suggest the future legal IR and QA research directions, especially for low-resource languages like Vietnamese.

All in all, the disseration makes three major contributions:

- We study the supporting relation in the legal texts, and propose an approach called *supporting model* that can deal with both the retrieval and the entailment phases in the case law retrieval task in Chapter 3. The underlying idea is the case-case, the paragraph-paragraph as well as the decision-paragraph supporting relations to enhance the relevancy for legal text retrieval. Additionally, based on the supporting relation, we also propose a method to automatically create a large weak-labeling dataset to overcome the short of annotated data.

  This contribution was published in the **Artificial Intelligence and Law** journal (SCIE, ISI Q1 journal) 2022 [VTHY1]. It was also applied to build a multi-task and ensemble approaches in legal information processing in the **Review of Socionetwork Strategies** journal (ESCI, WoS journal) 2024 [VTHY2].

- We propose and construct *a heterogeneous knowledge graph* encompassing different types of legal entities (case law, courts, statutory laws, and legal domains) to improve legal information organization and retrieval in the statutory – case law retrieval task in Chapter 4.

  This contribution was published in the **15th International Conference on Knowledge and Systems Engineering** (**KSE**) 2023 (indexed by Scopus) [VTHY3].

- We study the citation, reference relationships between the legal articles and propose *a reference network* approach to enhance the performance of the legal document question answering task in Chapter 5. Embedding and encoding the local references and the global (long-range) dependencies among legal articles into deep pre-trained language models make the final QA model more robust and accurate. Also, by uncovering hidden connections between laws, our method can assist in the identification of inconsistencies and gaps in the legal system, ultimately improving its effectiveness and reliability.

  This contribution was published in the **JSAI-isAI 2022**. **Lecture Notes in Computer Science**, **Springer** [VTHY4]. It was also applied to build a solution in the AQLAC competition 2022-2023 and publiced paper in **KSE-2022**, **KSE-2023** conferences (indexed by Scopus) [VTHY5, VTHY6].

This PhD dissertation contributes to both the scientific and practical areas. The dissertation presents a comprehensive overview of legal NLP for legal document IR and QA. It also provides insights into the characteristics of legal documents and the relationships among them. Additionally, the methods of representation, architectural designs of models, and the procedural steps for training and evaluating these models are elaborately described within this dissertation.

1.6 Dissertation Structure

The organization of the dissertation is depicted in Figure 1.6, encompassing five chapters, and a conclusion section. Publications related to the dissertation are described in their respective chapters:

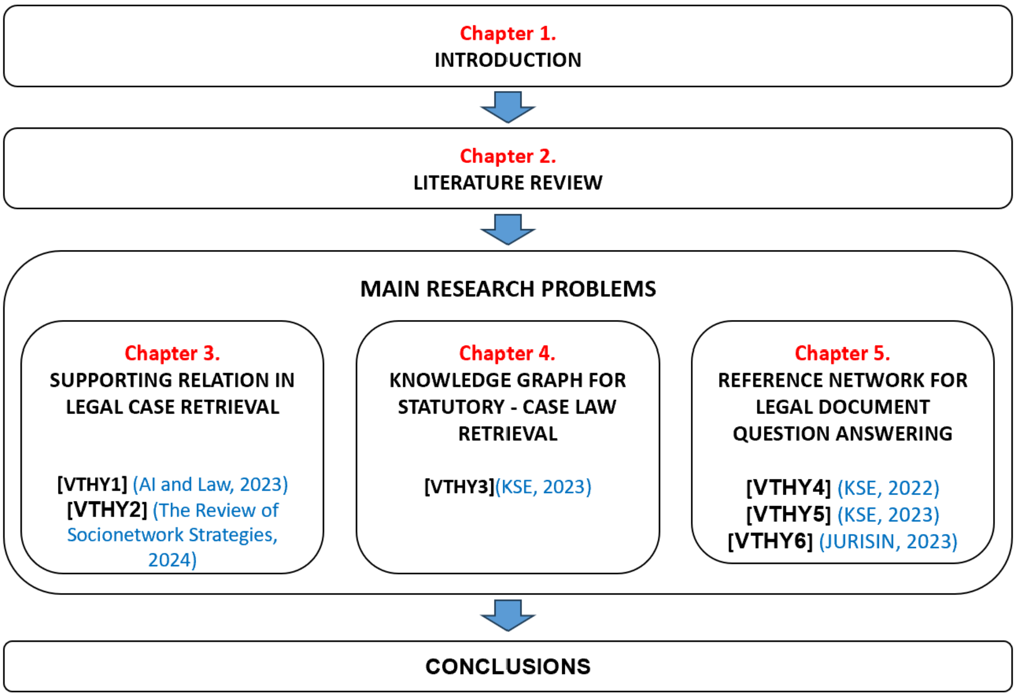

Chapter 1.
INTRODUCTION
Chapter 2.
LITERATURE REVIEW
MAIN RESEARCH PROBLEMS
Chapter 3.
SUPPORTING RELATION IN LEGAL CASE RETRIEVAL
[VTHY1] (AI and Law, 2023)
[VTHY2] (The Review of Socionetwork Strategies, 2024)
Chapter 4.
KNOWLEDGE GRAPH FOR STATUTORY - CASE LAW RETRIEVAL
[VTHY3](KSE, 2023)
Chapter 5.
REFERENCE NETWORK FOR LEGAL DOCUMENT QUESTION ANSWERING
[VTHY4] (KSE, 2022)
[VTHY5] (KSE, 2023)
[VTHY6] (JURISIN, 2023)
CONCLUSIONS

Figure 1.6: The dissertation outline

Chapter 1: INTRODUCTION gives an overview of key concepts in legal NLP, IR and QA problems that are relevant throughout this research. The primary content of this chapter is on the introduction, motivations, challenges, and the problem statement.

Chapter 2: LITERATURE REVIEW OF PROBLEMS AND METHODS reviews existing studies related to this disseration topic and setting the background for the research questions.

Chapter 3: SUPPORTING RELATION MODEL FOR CASE LAW RETRIEVAL presents our approach in the case law retrieval problem, the task of locating truly relevant case laws given an input query case. This chapter presents the difficulties and challenges of legal natural language, the complex concepts and structures of case law documents, as well as the limitation of labeled data. Then, this chapter presents our main approach called supporting model to deal with these challenges.

Chapter 4: KNOWLEDGE GRAPH FOR STATUTORY – CASE LAW RETRIEVAL develops a novel approach to define and construct a heterogeneous knowledge graph encompassing case laws and relevant legislative documents to improve legal information organization and retrieval. Our method involves data collection, entity extraction, and graph construction using NLP techniques. The constructed heterogeneous graph connects courts, cases, domains, and laws, significantly enriching information provided by retrieval systems. The proposed approach demonstrates potential in case analysis, legal recommendations, and decision support, providing valuable insights and resources for the legal domain.

Chapter 5: ARTICLE REFERENCE NETWORK FOR IR–BASED LEGAL QUESTION ANSWERING presents a novel approach to statutory – case law retrieval that utilizes a reference network to uncover connections between laws. By presenting laws as a network of references, our method allows users to quickly identify relevant laws and navigate the intricate web of legal documents. We evaluate the performance of our approach using a large corpus of statute laws and show that it outperforms existing retrieval methods. The proposed approach can contribute to the development of AI-assisted legal research tools, making it easier for legal practitioners to find relevant laws and precedents. Additionally, this chapter also presents models of legal QA on several Vietnamese datasets.

Finally, Chapter Conclusions summarizes the important points in the dissertation, the main contributions as well as limitations of the dissertation. It also points out the future research problems in the legal document IR and QA topic.

Chapter 2

Literature Review of Problems and Methods

In this chapter, we will first give an overview of legal NLP research in Section 2.1. Then, we will describe and discuss previous studies that are the basis or closely related to three legal IR problems: case law retrieval, statutory – case law retrieval, and IR–based legal question answering in Sections 2.2, 2.3, and 2.4, respectively. Next, Section 2.5 presents several techniques for representing and encoding legal textual data as well as several methods to represent graphical structures like TextRank, knowledge graph, etc. Finally, Section 2.6 reviews both traditional models and recent deep learning based methods for IR problems.

2.1 Legal Natural Language Processing

Although legal NLP is a sub-direction, it includes almost all important tasks of NLP. There are various ways to classify legal NLP problems and applications from a layer's perspective. However, in this thesis, legal NLP tasks are categorized from a technical point of view. According to Katz et al., legal NLP problems are divided into seven primary groups [60] as shown in Figure 1.1: resources, pre-processing, information retrieval, classification, information extraction, summarization, and text generation.

Resources [43] include taxonomies, ontologies, and datasets, which provide valuable support for various legal applications. Taxonomies and ontologies organize legal concepts and relationships in a structured manner, enabling better categorization and understanding of legal knowledge. They facilitate effective information retrieval, summarization, question answering, legal reasoning, and the development of intelligent legal systems. Datasets specific to the legal domain are essential for training and evaluating AI models in tasks such as legal text classification, named entity recognition, or legal question answering. These datasets provide annotated examples that assist in learning patterns and relationships within legal texts.

Pre-processing [33] involves tasks like tokenization, segmentation, annotation, anonymization, and translation to prepare the text for further analysis. Annotation labels specific information in the text, anonymization protects sensitive data, and translation facilitates cross-lingual analysis. Pre-processing ensures that legal texts are structured, standardized, and ready for further effective analysis.

Information retrieval (IR) [57, 63] is a crucial task in legal text processing. It encompasses tasks such as legal document retrieval and question answering (QA). Document retrieval aims to retrieve relevant legal documents based on user queries or specific criteria, facilitating efficient access to the vast amount of legal information available. Legal QA, based on both IR and NLP techniques, involves developing systems that can comprehend and respond to legal queries or questions by extracting relevant information from relevant legal texts. This enables legal professionals and individuals to obtain relevant and accurate answers to their legal inquiries.

Classification involves various subtasks such as outcome prediction and legal area classification. Outcome prediction [16, 47] aims to forecast the potential outcome of case laws based on textual evidence and historical data. This predictive capability assists legal professionals in assessing the likelihood of success or failure in a given case. Legal area classification [102, 139] involves categorizing legal documents into specific domains or areas of law, enabling efficient organization and retrieval of relevant information. Topic modeling [75], on the other hand, focuses on extracting latent themes or topics from legal texts, providing insights into the main issues and concepts discussed within the documents.

Information extraction encompasses tasks such as labeling, text extraction, and event extraction, which aim to automatically extract structured information from unstructured legal texts. Labeling involves identifying and classifying specific elements in the text, such as named entities (e.g., names, organizations, locations), legal concepts, or relationships between entities. Text extraction [40] focuses on extracting relevant textual content, such as clauses, provisions, or citations, from legal documents, enabling the retrieval of specific information for analysis or summarization. Event extraction [43, 134], on the other hand, involves identifying and extracting events or actions described in legal texts, such as court decisions or contract clauses.

Summarization [13, 55, 127] of legal documents aims to condense large legal texts into concise summaries. Abstractive summarization generates dynamic summaries, while extractive summarization selects salient sentences from the original texts.

Text generation is specifically the automated drafting of legal documents. It leverages NLP techniques to generate legal texts, such as contracts, agreements, or legal opinions, automatically. Automated drafting aims to streamline the process of creating legal documents by providing template-based or customized text generation capabilities. By utilizing machine learning algorithms and language models, NLP systems can generate accurate and contextually appropriate legal language, saving time and effort for legal professionals. Additionally, these systems can assist in ensuring consistency, compliance, and accuracy in legal document creation.

Automated processing of legal texts is a well-established research domain. The methods applied to various challenges within the automatic legal text processing have developed with advancements in computing power, and scientific and technological underpinnings. Legal NLP is a complex field, individual studies often address only a fraction of the existing issues. Nonetheless, these contributions serve as critical in supporting the progression of automated legal document processing.

Zhong et al. [167] provides a comprehensive overview of approaches, methods, and applications within the legal AI. It categorizes the previous studies of automated legal processing into two distinct groups: symbol–based methods and embedding–based methods. The symbol–based methods leverage knowledge bases to construct a system, whereas the embedding–based methods rely on patterns discerned by the model from data to inform decision-making processes. The legal AI applications capture a representative sample but do not encompass the entire spectrum of practical applications.

The early works in this field are rule-based systems [18, 119, 140, 146]. These systems are expert or lexical matching systems, facilitating the search and retrieval of information within the legal domain. Moreover, they are capable of conducting logical inferences within the legal framework, provided that the data is adequately described and represented in a manner comprehensible to computers. Nonetheless, a notable drawback of these systems is their reliance on human-crafted rules. Simple rule sets render these systems inflexible, while the development of more complex rule sets demands significant human input and effort.

The endeavor of teaching machines to comprehend legal language has proven to be a challenging task. However, significant progress has been made in the past decade regarding the quality and performance of language models. The advancements in neural network research have reshaped the broader field of natural language processing during this period [69, 123]. While early NLP studies focused on word embeddings [87, 107], the latest generation of language models is built on the transformer architecture [148]. This architecture enables intelligent manipulation of the attention mechanism, facilitating more efficient parallelization during training tasks. Despite some criticisms, successive iterations of increasingly large transformer-based language models (LLMs) have yielded truly remarkable outcomes [38, 131, 162].

Pre-trained language models have demonstrated significant advancements in legal tasks, although there are cases where their full potential has not been realized. Combining pre-trained language models with domain-specific knowledge can create models with higher performance, leveraging the strengths of pre-trained language models to rapidly generate highly accurate legal domain models for various real-world applications [14, 27, 165]. In other words, general NLP models cannot overcome the performance of domain-specific models trained specifically for the legal field with same size. Additionally, the development of large-scale computing systems has led to the emergence of large language models. These models may outperform low-scale domain models but require substantial resources and costs.

With the rapid growth of the digital data, the adoption of machine learning methods [7, 109], particularly deep learning [25, 61, 145], is increasingly prevalent in the field of natural language processing and, more specifically, in automated legal processing. Along with this research trend, numerous datasets have been published:

- CAIL2018 [155] marking the inception of the large-scale Chinese legal dataset for judgment prediction tasks. Comprising over 2.6 million criminal cases released by the Supreme People's Court of China. It features comprehensive and detailed annotations of judgment outcomes, including applicable law articles, charges, and prison terms.

- Duan et al. introduce the Chinese judicial reading comprehension (CJRC) dataset, which comprises approximately 10,000 documents and close to 50,000 questions, each paired with its answer [41].

- LMTC [26] is legal large-scale multi-label text classification. They unveiled a novel dataset containing 57,000 legislative documents from EUR-LEX. These documents are annotated with approximately 4,300 EUROVOC labels, making the dataset apt for LMTC as well as few-shot and zero-shot learning applications.
- JEC-QA represents an extensive question answering dataset within the legal field, compiled from the National Judicial Examination of China [166].
- Liu introduced a dataset created for named entity recognition within German federal court decisions, encompassing approximately 67,000 sentences and over 2 million tokens. This data collection includes 54,000 entities that have been manually annotated and classified into 19 specific semantic categories [78].

Additionally, competitions are organized to consolidate and identify the best solutions for specific tasks in legal NLP. The competition on legal information extraction/entailment (COLIEE) [58, 111, 112] is an annual event designed to foster advancements in the field of legal informatics by challenging participants to develop systems capable of extracting relevant legal information and determining legal entailment from vast corpora of legal texts. Especially, automated legal question answering competition (ALQAC) [143] provided the legal question answering dataset, which is a manually annotated collection based on the statute laws in the Vietnamese language.

While legal AI and NLP present numerous benefits to the legal profession, it also raises important ethical and legal considerations. Questions surrounding data privacy, bias in algorithms, and the potential impact on employment within the legal industry need to be carefully addressed and regulated. In conclusion, AI and NLP are transforming the practice of law, empowering legal professionals with powerful tools for research, document analysis, and decision-making. While there are challenges to be addressed, the integration of AI in the legal field has the potential to improve access to justice, enhance efficiency, and ultimately shape the future of legal practice.

2.2 Case Law Retrieval

Information retrieval (IR) has a long history of research and development. Traditional IR models use lexical methods, which are based on the term matching between the query and documents. BM25 [120] is a special weighting and normalization of TF-IDF model that ranks documents based on the query terms appearing in each document.

Vector space model (VSM) [23] represents queries and documents as vectors of weights which are computed on the term frequency in the documents. The relevance between a query and a document is usually calculated by the cosine similarity between the query vector and the document vector. Gerard Salton et al. [125] represent the documents and the query as term vectors where each term is assigned a weight value between 0 and 1. The higher weight is, the more frequency of the term is in the document. The cosine vector similarity function has been used to calculate the relevance between the query vector and the document vector. The language model also applies in IR [136, 164], documents ranking base on the probability of generating a query from the language model of each document. Berger et al. proposed a translation method to estimate the probability that the document is corresponding to the query, they assume that the query is generated from the document [10]. Machine learning is applied to solve the IR problem called by learning to rank (L2R) method. Liu et al. [78] proposed L2R approaches based on their manual feature engineering. Gradient descent is usually used for training L2R models such as pairwise or listwise loss function in RankNet [20], LambdaRank [21] and LambdaMART [154].

In ad-hoc retrieval, one of the earliest neural network methods, semantic hashing proposed by Salakhutdinov and Hinton [124], is a document encoder-based approach. Semantic hashing presents documents and queries by condensed binary vectors, retrieval documents have the same hash vectors matching with query vectors.

The deep structured semantic model (DSSM) [52] is specially designed for short text mapping. DSSM includes two parts of a neural network to represent the query and document title, which are a fully-connected layer for concatenating representation vectors and a distance function in the last layer. To improve query and document representation, more complex architectures are proposed such as convolutional neural network (CNN) [46, 51, 135]. In 2016, Palangi et al. proposed a model using recurrent neural networks (RNN) with long short-term memory (LSTM) unit to embed each word's information in a sentence into a semantic vector [105]. The model can improve the semantic representation of the sentence because of its ability to capture long-term memory. Tai et al. proposed two variant tree neural networks: the n-ary tree-LSTM and the child-sum tree-LSTM, where each LSTM unit takes in the information from multiple child units [141]. These variants address the problem of preserve sequence information over long periods. They also overcome the limitation of the LSTM networks in that they only allow for strictly linear chains. Well-known models include interaction-based networks based on part-level matching is explored for both short text matching [106, 150] and long text matching [89]. Pang et al. modeled text matching as image recognition and use a convolutional neural network to capture significant matching patterns from phrases and sentences with layer-by-layer composition [106]. Guo et al. construct a neural ranking IR survey and introduced a unified formulation over neural ranking models for information retrieval [48]. While, Marchesin et al. analyze improvements of two progressive neural information retrieval (NeuIR) models (neural vector space model and the deep relevance matching model) [83].

Pretrained language model has a great contribution to NLP problems, especially BERT models won in 11 NLP tasks [38]. BERT model is also applied in ad-hoc retrieval [37, 159]. It is worth investigating how to utilize the pre-trained language models, especially BERT to model the relationship between long text in case law retrieval task.

Legal information retrieval is an important problem in both retrieval and legal communication. A number of benchmark datasets was published such as Legal TREC [103], AILA [12], COLIEE, etc.

Retrieval of relevant materials is an important research topic in the legal field. Case law retrieval and entailment from prior cases are challenging because of legal knowledge and characteristics of legal materials. Bench-Capon et al. shown various statistical methods, learning methods, logical analysis, and expert knowledge in this task [9]. Zeng et al. build a domain knowledge for the retrieval system, which breaks down legal issues and categorizes them [163]. Saravanan et al. use a keyword-based query to solve this problem [127]. To improve the weaknesses of the keyword method, they also use a synonymy dictionary and domain ontological framework. Mandal et al. analyze structural information of case law documents. Query-document similarities are assessed based on lexical features (TF-IDF, topic modeling) and word/document vector representations [82].

To minimize dependence on expert and domain knowledge, neural networks have been applied to case law retrieval and entailment problems [81, 110, 132, 133, 145]. In the case law retrieval phase, Tran et al. combine document encoding by summarizing and lexical matching via phrase scoring framework. While Shao et al. proposed BERT-PLI, which utilizes BERT to capture the semantic relationships. BERT-PLI considers the relevance relationship between two case laws by a neural network based on aggregation paragraph-level interactions [132]. In the case law entailment phase, Rabelo et al. apply a transformer-based technique to tackle identifying entailment relationships between a decision and candidate entailing paragraph [110].

All previous studies only focus on the support relationship in the case unit in the case law retrieval phase. We propose a supporting case concept based on our supportive component extraction method. In other words, there exist cases in which some paragraphs support only some paragraphs in query case. The relation between supporting paragraphs and a given decision in the case law entailment phase is similar to the relationship between paragraphs in supportive cases and a query case in the retrieval phase. Therefore, we want to build only one model, which could capture the supportive relationships in both of phases, instead of two independent models as in previous researches. We also take advantage of the BERT model to build the supporting model in case law tasks. Our model is trained from our weak-labeling supporting dataset (without labeled original dataset) to reduce the effort and cost of data construction.

2.3 Statutory – Case Law Retrieval

In the early stages of natural language processing research in the legal domain, foundational efforts were dedicated to rule-based systems [119, 140]. These systems, encompassing expert systems or lexical matching approaches, provided valuable advancements by facilitating information retrieval and comprehension within the legal context.

In recent times, machine learning methods [9, 82], particularly deep learning [96, 145], have also been applied to address challenges in automated legal text processing.

The establishment of knowledge graphs has gained substantial attention as an effective approach to represent and organize legal information:

Filtz explore the challenge of data representation and retrieval, as illustrated in the the legal context. An approach for representing legal information, including legal norms and court decisions from Austria, is proposed. This approach demonstrates how such data can be leveraged to construct a legal knowledge graph. This graph holds potential for diverse applications, benefiting lawyers, attorneys, citizens, or journalists [45].

Tang et al. introduces SALKG, a semantic annotation system designed to construct a high-quality legal knowledge graph through a semi-automatic approach [142].

Sovrano et al. constructed an integrated knowledge graph based on combining open Knowledge extraction and natural language processing techniques, along with key ontology design patterns specific to the legal domain [137]. These patterns include event, time, role, agent, right, obligations, and jurisdiction. A question answering model has been developed from the legal knowledge graph to facilitate information retrieval and respond to these queries efficiently.

Dhani et al. used a legal knowledge graph built from court cases, judgments, laws, and other legal documents, which could facilitate applications such as question answering, document similarity, and search capabilities. In this demonstration, they detail their approach to predicting similar nodes within a case graph, which is derived from the overarching legal knowledge graph. [39].

Li et al. introduce an application in the legal domain known as legal provision prediction, designed to forecast the relevant legal provisions for specific cases. This task is conceptualized as a complex knowledge graph completion challenge, necessitating both comprehension of text and reasoning within a graph structure. [76].

Contextual search is implemented to address the limitations of keyword-based searches [84]. Nonetheless, lengthy queries pose a challenge in information retrieval. Both representation learning methods and matching learning methods have limitations in handling lengthy documents. Moreover, during the composition of documents within the court environment, legal judgments may contain typographical errors, punctuation inaccuracies, or may suffer from conversion issues when transitioning from PDF files. Additionally, the style of these case laws is contingent upon the court clerk responsible for recording the proceedings.

In recent years, deep learning methods such as convolutional neural networks (CNNs) [46, 51], recurrent neural networks (RNNs) [105], language models [8, 38], and large language models [100, 144] have demonstrated promising results in NLP in general, and specifically in legal NLP. In broader domains, labeled data is often abundant, whereas in the narrower legal domain, labeled datasets are relatively scarce. One of the reasons is legal texts are often subject to copyright restrictions and privacy concerns, which restrict the availability and sharing of annotated or labeled legal datasets. Additionally, legal texts require expert knowledge for accurate annotation and labeling, as legal concepts and nuances may be challenging for non-experts to interpret correctly. Annotating legal datasets requires legal expertise, which may limit the availability of annotators and increase the cost and time required for dataset creation. There are some public legal datasets in English [111], Chinese [53], German [70], or Japanese [111]. However, using them can be challenging due to the variations in legal systems across different countries. In the case of Vietnam, the process of digitizing the legal system is underway, and creating a standardized dataset remains a significant challenge. Furthermore, generating a training dataset for supervised learning tasks could be time-consuming and costly.

2.4 IR–based Legal Question Answering

In an increasingly complex and highly specialized world, legal professionals are required to navigate vast arrays of statutory laws that are constantly evolving and increasing in volume. The task of identifying relevant laws from a large corpus is not only laborious but also crucial for legal reasoning, legislative drafting, litigation, and legal scholarship [4, 72, 98]. The shift towards digitized legal documents has driven the demand for efficient and effective law retrieval systems that can aid legal professionals in this endeavor.

Traditional legal research methods, predominantly reliant on manual search or simple keyword-based searches, are often insufficient to cope with the intricate nature of legal texts. The structure of legal documents is characterized by a network of references, where laws cite other laws, creating a web of interdependent statutes. Understanding and navigating these interdependencies is essential for comprehensive legal analysis. However, due to the sheer volume and complex language of legal texts, manually tracking these references can be a daunting and error-prone task.

The advent of information retrieval (IR) technology, coupled with recent advancements in natural language processing (NLP) and machine learning, has spurred the development of IR systems that can process large volumes of texts to find relevant information. However, the specific challenges posed by legal texts, such as domain-specific language, the necessity for high precision, and the importance of context and inter-document references, require tailored and more sophisticated solutions.

Legal Question Answering based Document Retrieval

Before neural networks became widely used, classical NLP approaches were employed to solve information retrieval tasks [31, 80, 125]. These methods primarily relied on various lexical matching techniques. The authors suggested both logical and statistical models to determine the similarity between queries and candidates. These methods had advantages like quick computation and versatility, but they mainly depended on text morphology to make decisions. Since morphological similarity does not guarantee semantic similarity, it is challenging to ensure high accuracy in semantic similarity using these approaches. Thus, their performance is limited when document-query pairs contain non-overlapping text but semantic relevance or overlapping text but no semantic relation.

Legal language can be translated into logical language [68, 97]. A well-known system utilizing logical models for legal retrieval and reasoning in statute law is PROLEG (PROlog-based LEGal reasoning support system) [129], which is empowered by the Japanese presupposed ultimate fact theory [54]. PROLEG is based on the burden of proof concept, where automatic rule calling is used to reason a query. However, this system requires queries and legal documents to be in logical form, making it unsuitable for lay users. To address the semantic morphology difference and the burden of logical representation, several neural approaches for information retrieval in both general and legal domains have been proposed [99, 105, 130]. Most of these systems use classical neural network architectures like CNN or LSTM.

For legal text, Sugathadasa et al. [138] and Tran et al. [145] suggest using neural networks, achieving remarkable results. The authors examine legal document structures and propose novel representation methods based on their characteristics. Their experimental results show that these proposals effectively work in the legal domain. Kien et al. introduce a neural network architecture combining CNN and attention mechanisms, achieving state-of-the-art results on the Vietnamese legal question-answering dataset with a lightweight design [62]. These works also reveal that combining semantic vectors and lexical features can enhance the systems' overall performance.

Early pre-trained models include pre-trained word embeddings (Word2Vec [87], GloVe [107], or FastText [88]), which can easily find semantic relationships between words. In the legal domain, Law2Vec [25] authors introduce a word embedding variant trained on legal corpora, demonstrating its effectiveness. Recently, pretrained transformer-based models [148] have achieved state-of-the-art results in many benchmarks, both in general [17, 38, 73, 114, 115, 118] and legal domains [93, 95, 158, 160]. Pretrained approaches are especially useful when training data is limited in quantity.

Nguyen et al. propose the use of attentive neural network-based text representation for statute law document retrieval in their paper "Attentive deep neural networks for legal document etrieval" [96]. They develop a general approach using deep neural networks with attention mechanisms and introduce two hierarchical architectures with sparse attention, named Attentive CNN and Paraformer, to represent long sentences and articles. The methods are evaluated on datasets of varying sizes and characteristics in English, Japanese, and Vietnamese. The experimental results demonstrate that (i) attentive neural methods significantly outperform non-neural methods in retrieval performance across datasets and languages, (ii) pretrained transformer-based models provide better accuracy on small datasets but with high computational complexity, while lighter-weight Attentive CNN achieves better accuracy on large datasets, and (iii) the proposed Paraformer surpasses state-of-the-art methods on the COLIEE dataset, achieving the highest recall and F2 scores in the top-N retrieval task.

LegalGNN, proposed by Yang et al., is a legal information enhanced graph neural network designed for recommendation in the legal scenario (Legal-Rec) [157]. Legal-Rec is a specialized recommendation task aiming to provide potentially helpful legal documents for users, with three main differences from traditional recommendations: (1) the importance of both structural connections and textual content of legal information, requiring effective feature fusion; (2) users' preference for the newest case laws, leading to a severe new-item problem; and (3) the need to accurately model user interests, as most Legal-Rec users are domain-related experts with more stable information needs. LegalGNN addresses these challenges by designing a unified legal content and structure representation model, incorporating user queries into a heterogeneous legal information network (HLIN), and applying a graph neural network with relational attention mechanisms for Legal-Rec. Experiments on a real-world legal dataset show that LegalGNN significantly outperforms several state-of-the-art methods, making it the first graph neural model for legal recommendation.

In the recent years, Large Language Models (LLMs) such as GPT-3 [17] and GPT-4 [1] known for their superior text comprehension capabilities, and had applications in re-ranking for IR-based QA tasks. LLMs could enhance the quality and relevance of retrieved results by re-ranking them based on their deep contextual understanding. Beside their significant strengths and the improvements they bring to IR-based QA systems, the deployment of these models requires substantial computational resources, which can be a limiting factor for their widespread adoption.

Node Classification in Graph-Based Systems

Node classification is a critical task in graph-based systems, which involves predicting labels for unlabeled nodes based on observed labels in a network. This task is fundamental in applications pertaining to social network analysis, bioinformatics, and other domains where data can be naturally represented as graphs [11].

Several techniques have emerged to address the node classification challenge. Early methods were focused on using graph information as features and applying traditional classifiers iteratively. However, these techniques, while effective in incorporating local network structure, often overlook global graph properties. To capture the comprehensive network structure, label propagation methods using random walks have been proposed. These methods harness the connectivity patterns of graphs to propagate labels and infer node classes [11].

Deep learning approaches have recently been adapted for graph data, resulting in the development of Graph Convolutional Networks (GCNs). GCNs have shown promising results in node classification tasks by inherently capturing the graph's topology within the learning process. Nevertheless, deploying deep GCNs is often hampered by over-fitting and over-smoothing issues, which can deteriorate the quality of node embeddings. The DropEdge technique addresses these challenges by randomly removing edges during training, thus acting as a form of data augmentation and helping mitigate the negative effects of over-smoothing [121].

Research in this area is continually evolving, and a survey by Xiao et al.. reviews the use of graph neural networks (GNNs) in the context of node classification, dividing state-of-the-art methods into categories based on the underlying mechanisms, such as convolutional, attentional, and autoencoder [156]. Each mechanism brings a unique perspective on how best to encode graph structure and node information into the learning process.

Another innovative approach to node classification involves incorporating deep learning methods that can directly work with the graph structure. One such method involves using deep stacked sparse autoencoders alongside a softmax classification layer within a singular framework to learn node representations and perform node classification concurrently [74]. This end-to-end model learns embeddings that encapsulate both the structural and semantic patterns within the graph.

With the rise of semi-supervised techniques, researchers have explored data augmentation strategies for graph data to improve model generalization. NodeAug introduces a "parallel universe" augmentation scheme that prevents interference between nodes during augmentation, providing a new way to boost GCN performance. Additionally, it proposes subgraph mini-batch training, which is more efficient than training on the entire graph, further enhancing the scalability of the method [151].

Node classification in graph-based systems has witnessed significant advancements in recent years, moving from traditional classifiers to sophisticated deep-learning models that effectively leverage the graph structure. The ongoing development of new architectures and training strategies continues to shape the future of this field.

Building upon the innovations in graph construction and the incorporation of graph information into retrieval models from the above-mentioned studies, our work presents a unique reference network tailored specifically for the legal domain. We capture the intricate connections among legal statutes and enrich their representations by considering both the content and the context of the references. Our model, informed by graph neural networks and node classification techniques, seamlessly incorporates the information from the reference network into the retrieval process, making it more context-aware and relevant. This novel approach has the potential to significantly outperform existing retrieval systems in the legal domain, enhancing the efficiency and accuracy of legal research.

2.5 Representation of Legal Data

Legal text data is known for its complexity, often using long sentences, domain-specific clauses, concepts, and terminologies, and even including Latin phrases. In addition, there is an intricate web of connections within and between legal documents. Consequently, representing legal data requires both text-oriented and structure-oriented methods in order to capture both their textual and structural characteristics and relations. This section reviews a number of representation methods ranging from dense (embedded) vectors to graphs of texts or legal entities.

2.5.1 Textual Representation of Legal Data

Word embedding is a method in NLP where words (and sentences, paragraphs, or even documents) are encoded and transformed into a dense and low-dimensional space, ensuring that words with similar meanings are closed to each other in this space. Essentially, it transforms words into continuous vectors, where each dimension of the vector space represents a distinct feature or facet of the word. Word embedding learns this representation by analyzing a large amount of text data and refining the vectors according to the contextual occurrences of words. Word embedding has been utilized in various NLP tasks, such as text classification, sentiment analysis, machine translation, and information retrieval. They provide an effective way to capture the semantic and relationship between words, which can improve the accuracy of these applications. Common word embedding models are Word2Vec, GloVe, and fastText.

GloVe [107], short for Global Vectors for Word Representation, is a groundbreaking word embedding technique designed to capture semantic relationship between words. Developed by researchers from Stanford University, GloVe offers an advanced approach to word embedding that combines global statistics from the entire corpus with local context windows observed in individual text samples. Unlike traditional methods such as Word2Vec, which focus solely on local context, GloVe leverages co-occurrence statistics derived from the entire corpus to construct a global word-word co-occurrence matrix. Through a process of factorization, GloVe optimally generates word vectors that encapsulate both local and global semantic contexts. This approach enables GloVe embeddings to effectively capture nuances in word meanings and relationships, making them useful for a wide array of NLP tasks such as sentiment analysis, machine translation, and document clustering. GloVe embedding has gained significant attention and adoption within the NLP community due to its ability to produce high-quality word representations that facilitate and enhance various NLP applications.

fastText [56], an NLP toolkit introduced by Facebook Research, was built upon the Word2Vec model with additional improvements. Unlike Word2Vec, which treats each word in isolation, fastText breaks words into smaller constituents termed character n-grams. This approach enables fastText to comprehend the semantics of rare or unseen words more effectively. Furthermore, fastText integrates a hierarchical softmax function, enhancing both training velocity and precision. Consequently, it allows to efficiently handle large vocabularies, making it particularly useful for processing languages with rich morphology such as Finnish, Turkish, and Russian. fastText also supports Vietnamese. This allows Vietnamese NLP researchers and developers to construct more accurate and efficient NLP frameworks.

Contextual embedding involves representing words or phrases while considering their surrounding contexts. Unlike conventional word embedding models, which treat words as static vectors, contextual embeddings capture the contextual information of words by examining the words or phrases preceding and succeeding them in the texts. This approach enables a more nuanced and adaptable representation of words, capable of reflecting their evolving meanings depending on the context of their usage.

Contextual embeddings are typically built using deep learning architectures like recurrent neural networks (RNNs) or transformer-based models, which require training on large text datasets. These models are good at encoding contextual nuances by analyzing neighboring words or phrases, thus producing word embeddings suitable to their specific context. Contextual embeddings are useful for various NLP tasks, including sentiment analysis, named entity recognition, machine translation, and text classification, where word meanings within their contexts are crucial for accurate and meaningful analysis. Well-known examples of contextual embedding models include BERT (Bidirectional Encoder Representations from Transformers) [38], GPT (Generative Pre-trained Transformer) [114], and ELMO (Embeddings from Language Models) [108].

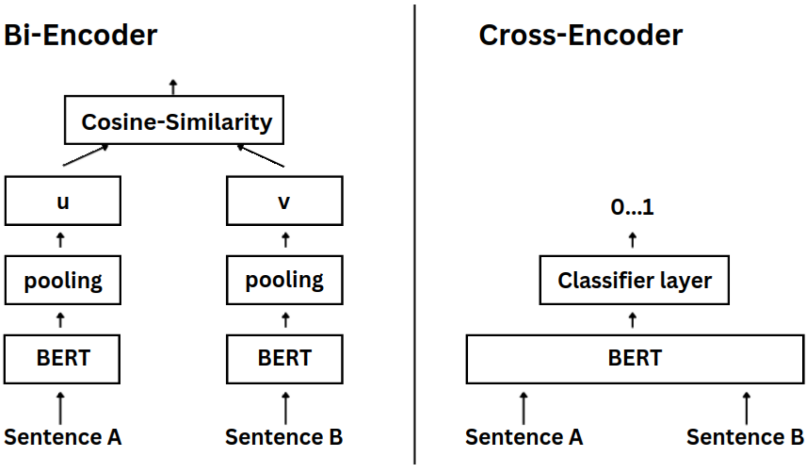


Figure 2.1: The difference between Bi-Encoder and Cross-Encoder

Bi-Encoder and Cross-Encoder architectures are two distinct frameworks employed in NLP to produce sentence embeddings. These embeddings are compact vector representations of sentences designed to encapsulate both their meaning and context. A Bi-Encoder architecture includes two distinct encoders, one for the input sentence and another for the candidate sentence, to generate embeddings. Subsequently, the similarity between these embeddings is computed using a similarity metric like dot product or cosine similarity to determine the proximity between the input and candidate sentences. Bi-Encoders have many applications such as retrieval and similarity ranking. Conversely, a Cross-Encoder architecture generates a unified embedding for both the input and candidate sentences utilizing a single encoder. This joint embedding then undergoes classification to identify whether the texts are semantically equivalent. Cross-Encoders are typically used in tasks such as text classification, natural language inference, and question answering. Each architecture, Bi-Encoder or Cross-Encoder, presents its advantages and drawbacks depending on specific tasks and datasets. While Bi-Encoders are faster and more memory-efficient, they may have lower performance in tasks requiring intricate inference. Conversely, Cross-Encoders offer greater capability and flexibility but demand more computational resources and may be susceptible to overfitting.

2.5.2 Structural Representation of Legal Data

TextRank [85] is an unsupervised graph-based ranking algorithm for text processing based on global information from the entire graph. The fundamental concept behind TextRank is based on the idea of voting. In this approach, the importance of a node is determined by both the number of votes it receives and the importance of the voters in the graph. Therefore, the score of a node is calculated by considering the number of connections it has with other nodes and the scores of those connected nodes. Mathematically, the TextRank algorithm is presented by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. In addition, the weighted graph is used to capture relations between nodes better. Edges in the graph are assigned weights, representing similarity scores between the nodes. The TextRank algorithm has usefl applications in automated summarization, specifically in keyword and sentence extraction tasks. For keyword extraction, a graph is constructed where nodes represent lexical units (such as words or phrases), and edges indicate the co-occurrence relations between connected nodes. The process's output is a list of terms or phrases for the given input. Sentence extraction is similar to keyword extraction, which aims to identify representative sentences from a text document. However, the co-occurrence relation in keyword extraction cannot be applied in this task, since sentences are much more complex than lexical units. Therefore, the relation between two sentences is defined by a similarity score. Traditionally, similarity scores in TextRank are determined using lexical representations such as TF-IDF or the number of overlapped tokens between two sequences. However, with the advent of semantic learning, these lexical representations are being replaced by contextual embeddings generated by Transformer-based models. Contextual embeddings capture the semantic meaning and context of words and phrases, enabling more accurate and contextually aware similarity in the TextRank algorithm. Because of its simplicity and good performance, TextRank plays a key role in summarization tasks as an initial step to identify candidates from large amounts of texts.

Knowledge graph, in the legal domain, serves as a complex framework for organizing, analyzing, and accessing vast repositories of legal information [45, 137]. It systematically captures the intricate relationships between legal entities, concepts, and principles, with nodes representing various legal entities such as laws, regulations, cases, and courts, and edges denoting the connections between them. To this end, a knowledge graph facilitates nuanced understanding and contextual exploration of legal information, enabling legal professionals and researchers to navigate in complex legal graphs, helping uncover hidden connections, and extract meaningful insights from legal data. With the increasing volume and complexity of legal information generated every day, knowledge graphs would be a scalable and adaptable solution for organizing and harnessing legal information effectively.

Heterogeneous graph: In legal knowledge representation, leveraging heterogeneous graphs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ offers a dynamic approach for presenting complex legal information. A heterogeneous graph helps integrateing diverse legal entities, such as laws, regulations, cases, courts, domains, and many other legal concepts. By utilizing heterogeneous graphs, which accommodate various types of nodes $\mathcal{V}$ and edges $\mathcal{E}$, the presentation of legal knowledge becomes more versatile and comprehensive. This approach enables legal professionals, researchers, and practitioners to explore intricate legal networks efficiently and achieve structural insights that cannot be done through traditional representations.

2.6 Information Retrieval Models

Legal IR models incorporate both conventional and deep learning approaches. Section reviews some of the most popular traditional IR methods and recent advanced deep learning-based IR models.

2.6.1 Traditional Information Retrieval Models

Term Frequency Matching is a commonly used method IR and text mining to assess the relevance of documents by considering the frequency of a search term in them. This concept forms the basis for popular text representation and ranking algorithms like TF-IDF and BM25, which are now integral parts of contemporary IR and search systems.

TF-IDF stands for term frequency – inverse document frequency. It is a text representation method that is commonly used in IR, text mining, and NLP to determine the importance of a term in a document in the context of an entire corpus. The meaning behind TF-IDF is that a term is important if it appears frequently in a single document while appears only in a small fraction of documents in the corpus. TF-IDF is calculated by multiplying the term frequency (TF) and the inverse document frequency (IDF) of a term. The term frequency is the number of times a term appears in a document, while the inverse document frequency measures how often a term appears across a corpus of documents. TF-IDF is now a an important text representation and indexing option in any IR and search engines. The formula for calculating TF-IDF score for term t of document $\mathbf{d}$ in a corpus $\mathbf{D}$ is:

$$\text{TF}(t, \mathbf{d}) = \frac{f(t, \mathbf{d})}{|\mathbf{d}|} \tag{2.1}$$

$$\text{IDF}(t, \mathbf{D}) = log\frac{|\mathbf{D}|}{n(t)} \tag{2.2}$$

$$\text{TF-IDF}(t, \mathbf{d}, \mathbf{D}) = \text{TF}(t, \mathbf{d}) \cdot \text{IDF}(t, \mathbf{D}) = \frac{f(t, \mathbf{d})}{|\mathbf{d}|} \cdot log\frac{|\mathbf{D}|}{n(t)} \tag{2.3}$$

in which:

- $f(t, \mathbf{d})$ is the frequency of term t in document $\mathbf{d}$;
- $|\mathbf{d}|$ is the total number of terms in document $\mathbf{d}$;
- $|\mathbf{D}|$ is the total number of documents in the corpus $\mathbf{D}$;
- $n(t)$ is the number of documents in $\mathbf{D}$ containing term t

Let $\mathbf{q}$ be a query consisting of k unique terms $\{t_1, t_2, \ldots, t_k\}$, the TF-IDF score for query $\mathbf{q}$ with a document $\mathbf{d}$ of a text database $\mathbf{D}$ is:

$$\text{TF-IDF}(\mathbf{q}, \mathbf{d}, \mathbf{D}) = \sum_{i=1}^{k} \text{TF-IDF}(t_i, \mathbf{d}, \mathbf{D}) \tag{2.4}$$

TF-IDF can be used for quanifying the importance of terms in each documents of a data collection. This can be used for classification, clustering, or IR applications. In retrieval systems, documents with higher TF-IDF scores are likely to be more relevant to the query.

BM25 [120] was introduced in 1994 by Stephen Robertson and Karen Spärck Jones. BM25 emerged as an improved document ranking algorithm in information retrieval systems. BM25 has gained widespread adoption, being a cornerstone ranking method in IR and search engines like Elasticsearch and Solr.

Given a question $\mathbf{q}$, containing n unique tokens $\{t_1, t_2, \ldots, t_n\}$, the BM25 score of a document $\mathbf{d}$ in document collection $\mathbf{D}$ is:

$$\text{BM25}(\mathbf{q}, \mathbf{d}, \mathbf{D}) = \sum_{i=1}^{n} \text{IDF}(t_i, \mathbf{D}) \cdot \frac{f(t_i, \mathbf{d}) \cdot (k_1 + 1)}{f(t_i, \mathbf{d}) + k_1 \cdot (1 - b + b \cdot \frac{|\mathbf{d}|}{avgdl})} \tag{2.5}$$

in which:

- $f(t_i, \mathbf{d})$ is the frequency of term t_i in document $\mathbf{d}$;
- $|\mathbf{d}|$ is the total number of terms in document $\mathbf{d}$;
- $avgdl$ is the average document length in the document collection $\mathbf{D}$;
- k_1 is a saturation curve parameter of term frequency;
- b is the importance of document length;
- IDF$(t_i, \mathbf{D})$ is the inverse document frequency of term t_i, and estimated as the following equation: $\text{IDF}(t_i, \mathbf{D}) = \ln\left(1 + \frac{N - n(t_i) + 0.5)}{n(t_i) + 0.5}\right)$. N is the total number of documents in $\mathbf{D}$, and $n(t_i)$ is the number of documents in $\mathbf{D}$ containing t_i.

In summary, while both algorithms rely on term frequency matching, BM25 incorporates additional factors such as document length and term specificity that will be significant for ranking documents in comparison to the pure TF-IDF.

2.6.2 Deep Learning–based Retrieval Models

BERT, Bidirectional Encoder Representations from Transformers, is a revolutionary language model introduced by Google in 2018. It marks a significant advancement in natural language understanding, particularly in tasks like text classification, named entity recognition, and question answering. Unlike previous models, BERT utilizes a Transformer architecture, enabling it to capture bidirectional contexts from a vast amount of text data. This bidirectional understanding allows BERT to grasp the meaning of words within their full context, leading to remarkable improvements in various NLP tasks. BERT notably attained an F_1 score of 93.2% on SQuAD 1.1, surpassing the previous best result of 91.6% and even outperforming human-level performance at 91.2%. Following the release of the research paper, both the source code and the pre-trained model were publicly shared, enabling the development of NLP and machine learning models utilizing BERT as a foundational component. BERT's pre-trained models are derived from extensive datasets, including BookCorpus with 800 million words and English Wikipedia with 2.5 billion words.

The idea behind BERT originates from the fact that although current word embedding models are trained on large-scale datasets with various neural architectures, they still fall short in encoding contextual information from limited supervised data. BERT was designed to train language text representation vectors through both left and right contexts. BERT applies a fine-tuning method that requires minimal specific architecture for each task, aiming to reduce the reliance on prior human knowledge and focus on extracting knowledge from data. BERT pre-trained models come with two types of parameter-based settings: BERT-base (12 Transformer layers, 12 attention heads, 110 million parameters) and BERT-large (24 Transformer layers, 16 attention heads, 340 million parameters).

T5 model: NLP involves a wide range of applications, from text classification, sentence similarity, question answering, machine translation to text summarization. Different models, learning objectives, and training strategies have been studied to address these tasks. However, an innovative research by Google presented the T5 model, a unified text-to-text architecture to treat the mentioned tasks as a sequence-to-sequence problem. In other words, with an input text and an appropriate prompt, the model will generate some target text as shown in Table 2.1. T5 is the first to address transfer learning across various tasks and domains. The architecture of this model follows the original Transformer design, consisting of encoder-decoder blocks. T5 applies the masked language model learning objective inspired by BERT. During training, 15% of the tokens in the input sequence are randomly masked. Researchers from Google developed their dataset "Colossal Clean Crawled Corpus" (C4) to pre-train T5 models. This dataset contains 750GB of cleaned English unlabeled text gathered from the Web. The authors released models with different sizes: small (60M), base (220M), large (770M), 3B, and 11B. Among 24 NLP tasks considered in [116], the T5 (11B) model achieved state-of-the-art results in 18 tasks, except the machine translation tasks.

The workflow to address document retrieval is estimating the probability of a document belongs to a relevant class, and then ranking candidate documents by their estimates in descending order. Typically, encoder-only based architectures (BERT and variants) are utilized to compute the relevance score of a (query, document) pair. However, the authors in [101] presented a novel method of applying sequence-to-sequence models in the document ranking task. Specifically, the researchers formulate the original

Table 2.1: Text processing tasks by T5 model.

Input	Output
translate English to German: That is good	Das ist gut.
cola sentence: The course is jumping well	not acceptable.
stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field.	3.8
Summarize: state authorities dispatched emergency crews Tuesday to survey the damage after an onslaught of severe weather in Mississippi...	six people hospitalized after a storm in attala county.

problem into a relevance prediction task. The input follows a specific template: "*query* $[\mathrm{q}]$, *document* $[\mathrm{d}]$, *relevant:* " where $[\mathrm{q}], [\mathrm{d}]$ are query and document text respectively. The seq2seq models are trained to generate true/false tokens as output to indicate the relevance relation of the query-document pair. Applying in document retrieval task, each pair of query and document is fed independently into the sequence-to-sequence model. The model then computes the probability $P(relevant = 1|\mathbf{q}, \mathbf{d})$ as the relevance score in this manner.

In [101], the authors conducted experiments using T5-family models (base, large, 3B). The models are trained on a passage ranking dataset called MS MACRO. Compared to baseline and encoder-only models, the proposed method achieved superior results, especially in low-data and zero-shot transfer setting. The authors claimed that seq2seq models can exploit external knowledge that BERT does not have.

2.6.3 Sumary

Chapter 2 provides an overview of legal natural language processing and legal information retrieval. We delve into three specific legal information retrieval problems: case law retrieval, statutory–case law retrieval, and IR-based legal question answering, related work and methodologies relevant to sub-problems. Subsequently, we explore various techniques for representing and encoding legal textual data, including methods for representing graphical structures. In subsequent chapters, we will decrised on the specific sub-problems, exploring their challenges and presenting advanced methodologies to address these challenges.

Chapter 3

Supporting Relation Model for Case Law Retrieval

Case law retrieval is the task of locating truly relevant case laws given an input query case. Unlike information retrieval for general texts, this task includes two phases (*case law retrieval* and *case law entailment*) and is much harder due to a number of reasons. First, both the query and candidate cases are long documents that consist of several paragraphs. This makes it difficult to model them with representation learning that usually has restriction on input length. Second, the concept of *relevancy* in this domain is defined based on the legal relation that goes beyond the lexical or topical relevance. This is a real challenge because normal text matching will not work. Third, building a large and accurate case law dataset requires a lot of effort and expertise. This is obviously an obstacle to creating enough data for training deep retrieval models. In this chapter, we propose a novel approach called *supporting model* that can deal with both phases. The underlying idea is the case-case supporting relation as well as the paragraph-paragraph and the decision-paragraph matching strategy. In addition, we propose a method to automatically create a large weak-labeling dataset to overcome the lack of data. The experiments showed that our solution has achieved the state-of-the-art results for both case retrieval and case entailment phases.

This work was published in the **Artificial Intelligence and Law** journal (SCIE, ISI Q1 journal) 2022 [VTHY1]. It was also applied to build a multi-task and ensemble approaches in legal information processing in the **Review of Socionetwork Strategies** journal (ESCI, WoS journal) 2024 [VTHY2].

3.1 Case Law Supporting Relation

In case law, the relevance of citing paragraphs from supporting cases lies in its capacity to fortify legal arguments and interpretations. By referencing supporting cases, legal practitioners demonstrate how established legal principles and reasoning can be applied to the current case, even in different factual contexts. This practice not only adds persuasive authority to the argument but also showcases the consistency and coherence of legal doctrine over time.

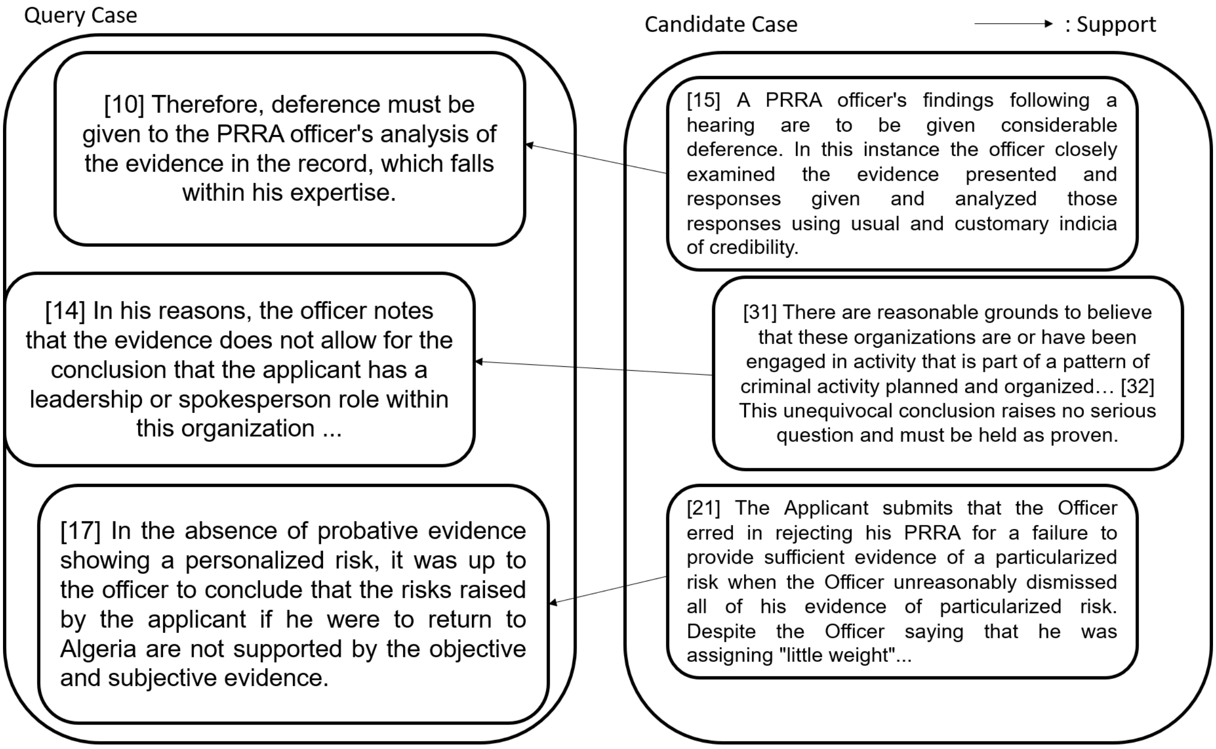
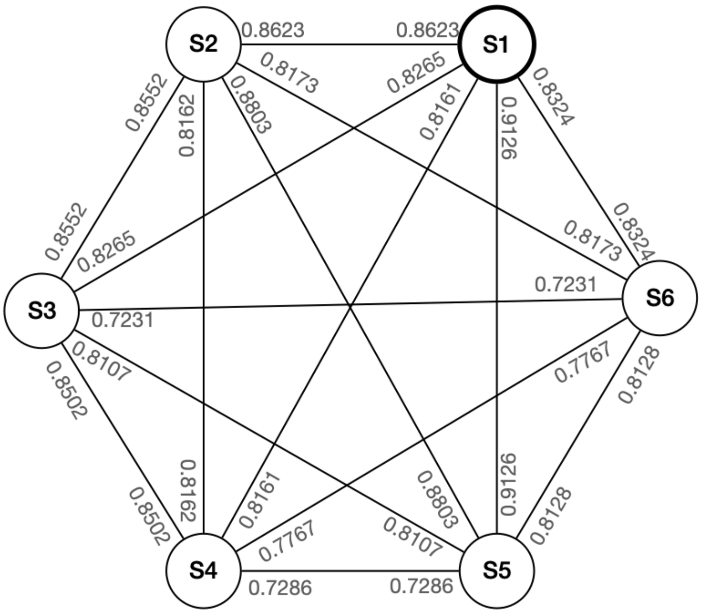

Query Case
Candidate Case
: Support
[10] Therefore, deference must be given to the PRRA officer's analysis of the evidence in the record, which falls within his expertise.
[14] In his reasons, the officer notes that the evidence does not allow for the conclusion that the applicant has a leadership or spokesperson role within this organization ...
[17] In the absence of probative evidence showing a personalized risk, it was up to the officer to conclude that the risks raised by the applicant if he were to return to Algeria are not supported by the objective and subjective evidence.
[15] A PRRA officer's findings following a hearing are to be given considerable deference. In this instance the officer closely examined the evidence presented and responses given and analyzed those responses using usual and customary indicia of credibility.
[31] There are reasonable grounds to believe that these organizations are or have been engaged in activity that is part of a pattern of criminal activity planned and organized… [32] This unequivocal conclusion raises no serious question and must be held as proven.
[21] The Applicant submits that the Officer erred in rejecting his PRRA for a failure to provide sufficient evidence of a particularized risk when the Officer unreasonably dismissed all of his evidence of particularized risk. Despite the Officer saying that he was assigning "little weight"...

Figure 3.1: Example of supporting component extraction between a query case and a candidate case

The case-case supporting relation does not only involve similar situations. The supporting cases can be mentioned and cited to support the query case law. According to our observation, a case law s is a noticed case of a query case qc, which does not mean that all parts of s support qc. In other words, if there are only some paragraphs in s that support some paragraph in qc, we can conclude that s support qc. Therefore, we introduct a supporting case concept for the case law supporting based on the supportive component. The long-text case law is splited into paragraph-like components and the supporting relation on the component level instead of focusing on the support relationship in the case law unit like in the previous studies [110, 132, 145]. Figure 3.1 illustrates an example of our supportive component. This is the part of supporting relation between a query case and a candidate case.

Similarly, case law paragraphs are typically structured in an argumentative form, presenting legal arguments with clarity, precision, and logical coherence. Each paragraph focuses on a specific legal issue or specific legal point; utilizing logic, evidence, and specific citations to elucidate the issue or viewpoint presented. There is thematic unity within each legal paragraph, ensuring that the narrative is presented clearly. Figure 3.2 shows an example of supporting relation among sentences in the case law paragraph.



Case IMM-2683-96, paragraph 4:
S1: The applicant alleges that the Board used standard form "boiler-plate" reasons and therefore denied the applicant a fair hearing.
S2: Key passages in the Board's reasons are identical or virtually identical to two other decisions, Jafari v. Minister of Citizenship & Immigration , Board
S3: In all three cases involving Iranian refugee claimants, Mr. Jack Davis was the presiding Board member.
S4: Jafari was heard three days after the case at bar.
S5: The respondent in turn argues that the Board clearly made an independent decision.
S6: The identical passages are nothing more than digests of the law on such legal questions as credibility and the documentary evidence

Figure 3.2: An example of supporting relation among sentences in the case law paragraph, each sentence in the paragraph is represented as a vertice, edges are semantic similarity between the sentences. S1 is topic sentence in this example.

3.2 Supporting Relation in Case Law Retrieval

With the recent advances in digitalization and digital transformation, lawyers can now easily access a huge volume of online legal materials. However, the larger number of legal documents is, the more difficult to find most relevant case laws that assist the lawyer's court preparation. Thus, developing an automated law retrieval system is significant to accelerate the lawyer's workflow.

Legal information extraction and entailment (COLIEE) is an annual competition for researchers to tackle the problems of information retrieval, extraction, and reasoning in the legal domain. One of the main challenges in the competition is the case law task. The data for this task is based on The Federal Court of Canada case law provided by vLex Canada[1].

A case law is typically a collection of previous legal conclusions written by courts. A lawyer can find relevant case laws and use appropriate conclusions to support the decision in the current case. Figure 3.4 illustrates an example of case law with its complex structure. *Temp. Cite*: including the applicant, the respondent and case code, which are the case law identification. *Indexed As*: is the name representing the case law, used to index into the database and refer to in other cases; *Indexed As* is usually a combination of the applicant and the respondent. *Federal court*: a place where trials and case laws happen. *Summary*: the summary of the case and referring to authoritative documents and sources such as court decisions, treaties, regulations, and government documents. *Counsel*: Information of the lawyer in charge. *Paragraphs*: the detailed description of the court. Typically, paragraphs include an introduction, background, issues, or facts, the decision of the director, analysis, and conclusion. The case law can vary in structure, the components may not be the same in all cases, which requires significant effort in processing. It is even difficult for trained lawyers to read, scan and find truly relevant case laws from a large legal database. Case law retrieval is, therefore, a complicated task that have a number of challenges as follows:

Challenge 1: Both the query and supporting cases are extremely long texts which contain around 3000 words on average.

The long query is a challenge in the retrieval task. Both representation learning and matching learning methods have limitations in processing lengthy documents. It is challenging to learn representation for long text in a limited vector space. Constructing and aggregating long documents in matching learning is also a difficult problem.

Challenge 2: The definition of relevance in the legal domain is quite different from the general definition of topical relevance [147]. Saracevic [126] proposed a definition of "relevance" as "pertaining to the matter at hand," or, more extended: "As a cognitive notion relevance involves an interactive, dynamic establishment of a relation by inference, with intentions toward a context."

In the legal scenario, relevant cases are those that can support the decision of a new case, which usually have similar situations and appropriate regulations. It is crucial

[1] https://ca.vlex.com/

to identify the supportive relationship between case laws. This relationship is far beyond the topical and lexical relevancy. Matching between the query case and candidate cases, between the query decision and supporting cases becomes much more difficult in comparison with general text retrieval.

Challenge 3: Creating a large and accurate dataset for the case law task requires much effort and expert knowledge in the legal domain. Tables 3.1 and 3.2 illustrate the small number of samples in both legal retrieval and entailment tasks. The lack of labeled data is an obstacle to training and evaluation of large deep neural models.

In this study, we propose a deep learning approach with a supporting model for case law retrieval called SM-BERT-CR to deal with the above challenges. We propose a supporting case concept for the case law retrieval phase based on our supportive component in the case law supporting relation (**Challenges 1 and 2**). The relation between supporting paragraphs and a given decision in the case law entailment phase is similar to the relation between paragraphs in supporting cases and a query case in the retrieval phase.

Denoting a support relation as $support(a, b)$ (a supports b), the case law retrieval and entailment tasks are formalized as follows:

Case law retrieval phase :Let C be the space of all possible legal cases and case laws and let $C \subset$ **C** be a corpus of case laws (i.e., a case law database). Given an input query case $c_q \in$ **C**. The query c_q is normally a new legal case that a judge or a lawyer is currently working on, the task is to extract a set of supporting cases $C^r = \{c_i^r \mid c_i^r \in C \wedge support(c_i^r, c_q)\}$. We assume that a candidate case C_i^r supports the query case c_q if and only if there are one or more paragraphs in s which support a decision in c_q:

$$support(c_i^r, c_q) \iff \exists p_j \in c_i^r \wedge \exists p_k \in c_q : support(p_j, p_k)$$

Case law entailment phase: Given a triplet including the input query case c_q, a decision d_q of the query case c_q, and the list of all supporting cases C^r returned from the previous phase. Let P^r be the set of all text paragraphs being segmented from a given supporting case $c^r \in C^r$, the task is to identify a set of entailing paragraphs:

$$P^e = \{p_i^e \mid p_i^e \in P^r \wedge support(p_i^e, d_q)\}.$$

The previous works usually tackle finding the supportive relationship between query-case/decision and candidate-case/candidate-paragraph indirectly through similarity measures. Unlike previous studies, we build a supporting model to predict the supportive relationship directly (**Challenge 2**). Inspired by the success of the pre-trained language model BERT [38] on a wide variety of natural language processing tasks, we adapt the BERT model to build our supporting model in case law tasks.

Besides the supporting model, we also exploit multiple similarity measurements such as lexical similarity (keyword matching) [82] and semantic similarity (context matching). Although lexical similarity and semantic similarity are quite different from each other, they can be combined and complementary. The lexical similarity can be obtained by matching word by word with some alteration such as stemming, stopword removal, lemmatization, etc. A higher score in lexical similarity can show high matching between two documents, but with low lexical similarity, it does not mean that these documents do not have any relation. Thus, we combine the supporting model with the lexical model in our case law retrieval system.

To tackle the challenge of lacking labeled data, we use some heuristics to automatically construct the training dataset about the supporting relationship in case law called a weak-labeled supporting dataset (**Challenge 3**). This dataset is constructed based on our supporting relation in the case law paragraph that a paragraph contains a decision sentence and the remaining sentences support this decision sentence. Moreover, we assume that the decision sentence is the topic sentence in the candidate paragraph. To identify the decision sentence in the candidate paragraph, we apply the TextRank algorithm [85] - a graph-based ranking model for automatic sentence extraction. The introduction of this dataset can reduce the dependency of neural models on labeled data.

3.3 General Architecture

A small amount of training data brings obstacles to the training process of deep neural models. While creating an accurate large dataset for supporting relations can be challenging. To taclke this challenge, we design some heuristics to automatically extract supporting text-pairs from the training dataset in the COLIEE 2020 case law retrieval task and construct a "weak-labeling" dataset.

Our proposed supporting model is illustrated in Figure 3.3. We train the supporting model on the task of supporting text-pair recognition in case law. The goal of the task is to determine whether a text supports a decision. The pipeline consists of the following components:

- **Retrieval dataset**: collection of candidate cases in the task 1 COLIEE training

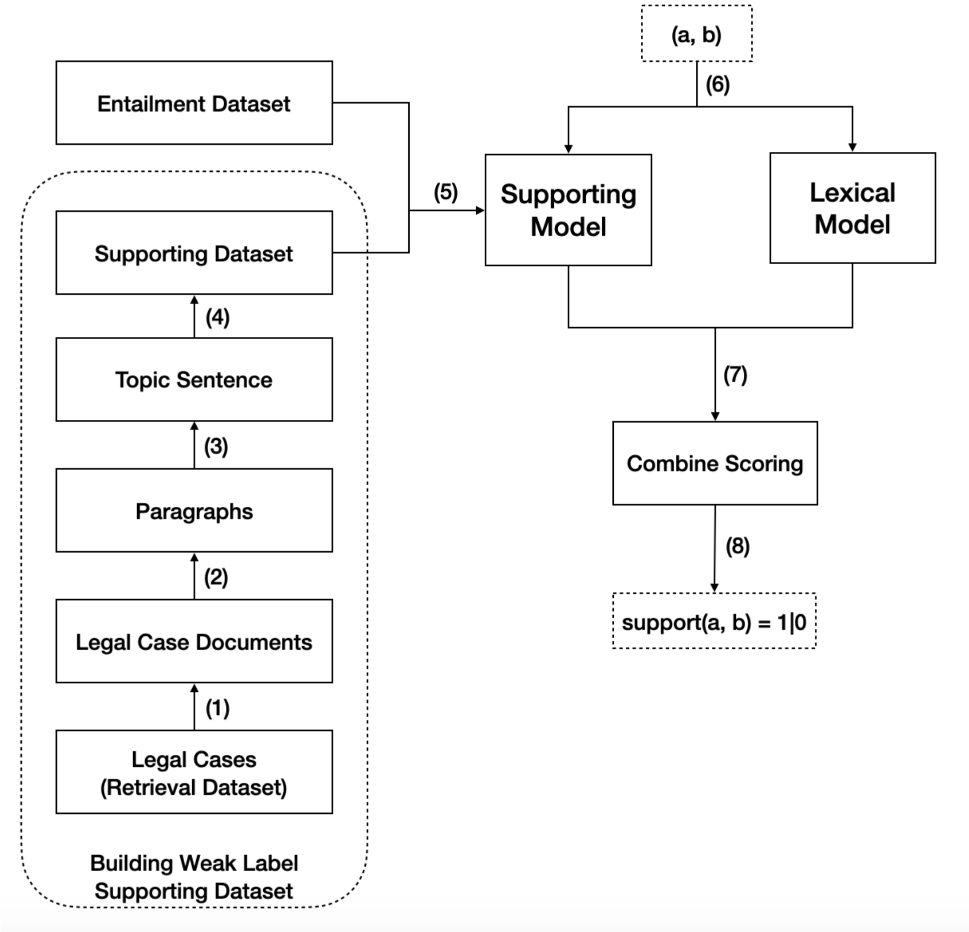


Figure 3.3: Scoring method pipeline in supporting text-pair recognition task

2019 dataset.

- **Case law documents**: built by cleaning and removing duplicates case laws from retrieval dataset.
- **Paragraphs**: a cleaned list of paragraphs, which are segmented from the body of case law documents.
- **Topic sentence**: the TextRank algorithm - a graph-based ranking algorithm is applied to identify the decision sentence in the paragraph.
- **Weak label supporting dataset**: constructed based on text-pairs consisting of a decision sentence and the remaining sentences in the paragraph.
- **Entailment dataset**: training data consists of tuples of a decision query, a paragraph number of the noticed case, and the gold label.
- **Supporting model**: trained from the weak label supporting dataset and the entailment dataset based on the BERT classification model with pre-trained parameters.

- **Lexical model**: the BM25 similarity score is used in this component, which could capture lexical similarity and semantic similarity between a text-pair.
- **Combine scoring**: the final score of a text-pair is weighted combination of scores from the supporting model and the lexical model.

3.4 Datasets

In this section, we introduce the datasets utilized in our experiment. In addition to analyzing the original data, we also describe the process of constructing the weakly labeled dataset.

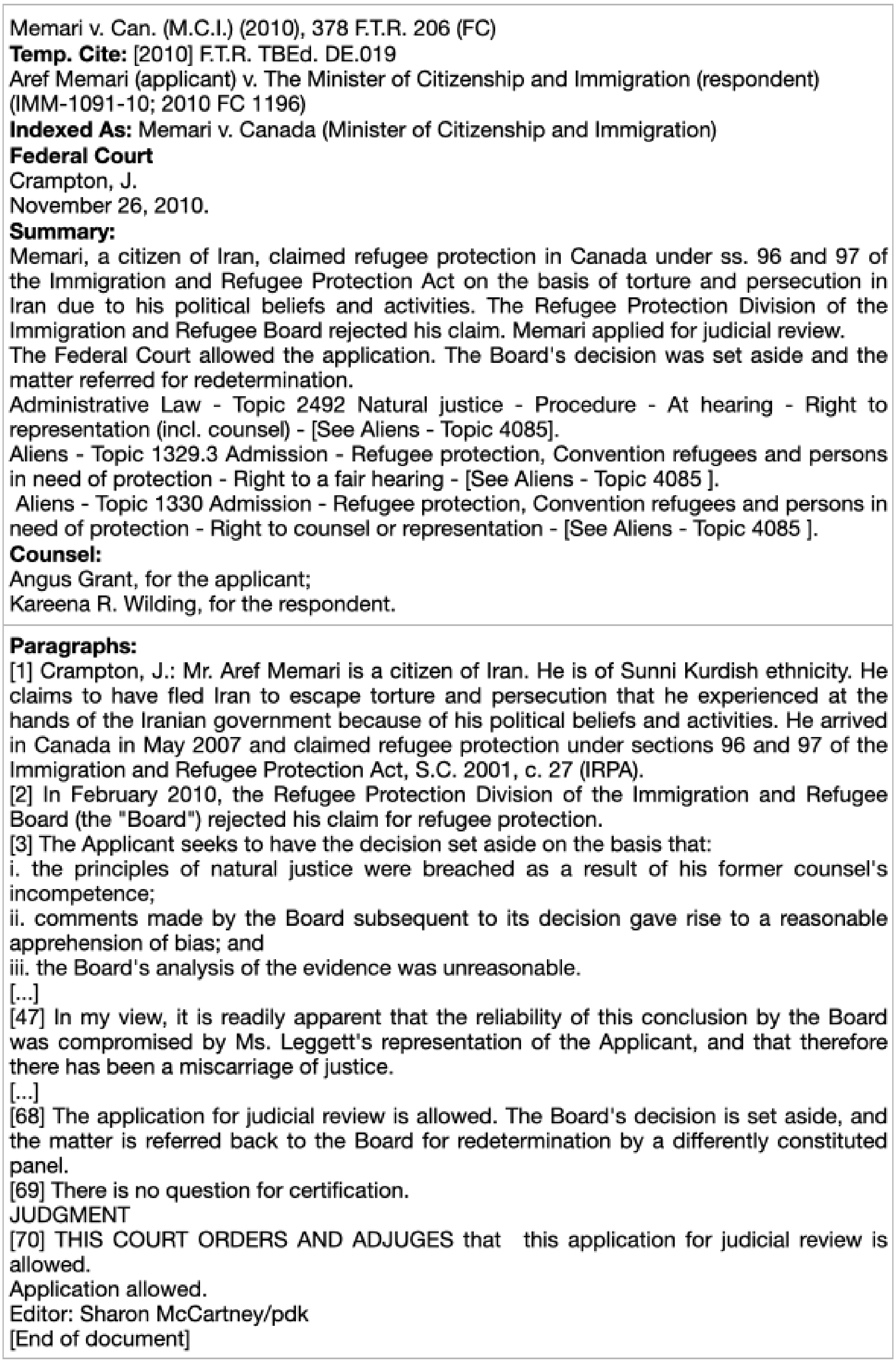
3.4.1 The Case Law Task in COLIEE Dataset

The competition on legal information extraction and entailment (COLIEE) was held to tackle the challenges of case law retrieval and case law entailment. Figure 3.4 illustrates an example of case law.

Data in these two tasks are sampled from Federal Court of Canada case laws. Table 3.1 and Table 3.2 give a statistical summary of the dataset. In the case law retrieval task, the training 2019, the testing 2019, and the testing 2020 set consist of 285, 61, and 130 query cases, respectively. Each query in training data has 200 candidate cases. Each query has 5.21 notice cases on average. In the case law entailment task, the training 2019, the testing 2019, and the testing 2020 set include 181, 44, and 100 case queries respectively, along with the corresponding decision query extracted from the query case. In this training data, each case has 32.12 candidate paragraphs on average for recognizing entailment relation, and on an average of 1.12 candidate paragraphs have an entailment relation with a case query.

3.4.2 Weak-labeling Supporting Dataset

The weak-labeled supporting dataset is constructed based on our assumption that a candidate paragraph contains a decision sentence and the remaining sentences support this decision sentence. Moreover, we assume that the decision sentence is the topic sentence in the candidate paragraph.



Memari v. Can. (M.C.I.) (2010), 378 F.T.R. 206 (FC)
Temp. Cite: [2010] F.T.R. TBEd. DE.019
Aref Memari (applicant) v. The Minister of Citizenship and Immigration (respondent)
(IMM-1091-10; 2010 FC 1196)
Indexed As: Memari v. Canada (Minister of Citizenship and Immigration)
Federal Court
Crampton, J.
November 26, 2010.
Summary:

Memari, a citizen of Iran, claimed refugee protection in Canada under ss. 96 and 97 of the Immigration and Refugee Protection Act on the basis of torture and persecution in Iran due to his political beliefs and activities. The Refugee Protection Division of the Immigration and Refugee Board rejected his claim. Memari applied for judicial review.

The Federal Court allowed the application. The Board's decision was set aside and the matter referred for redetermination.

Administrative Law - Topic 2492 Natural justice - Procedure - At hearing - Right to representation (incl. counsel) - [See Aliens - Topic 4085].

Aliens - Topic 1329.3 Admission - Refugee protection, Convention refugees and persons in need of protection - Right to a fair hearing - [See Aliens - Topic 4085].

Aliens - Topic 1330 Admission - Refugee protection, Convention refugees and persons in need of protection - Right to counsel or representation - [See Aliens - Topic 4085].

Counsel:
Angus Grant, for the applicant;
Kareena R. Wilding, for the respondent.

Paragraphs:

[1] Crampton, J.: Mr. Aref Memari is a citizen of Iran. He is of Sunni Kurdish ethnicity. He claims to have fled Iran to escape torture and persecution that he experienced at the hands of the Iranian government because of his political beliefs and activities. He arrived in Canada in May 2007 and claimed refugee protection under sections 96 and 97 of the Immigration and Refugee Protection Act, S.C. 2001, c. 27 (IRPA).

[2] In February 2010, the Refugee Protection Division of the Immigration and Refugee Board (the "Board") rejected his claim for refugee protection.

[3] The Applicant seeks to have the decision set aside on the basis that:

i. the principles of natural justice were breached as a result of his former counsel's incompetence;

ii. comments made by the Board subsequent to its decision gave rise to a reasonable apprehension of bias; and

iii. the Board's analysis of the evidence was unreasonable.

[...]

[47] In my view, it is readily apparent that the reliability of this conclusion by the Board was compromised by Ms. Leggett's representation of the Applicant, and that therefore there has been a miscarriage of justice.

[...]

[68] The application for judicial review is allowed. The Board's decision is set aside, and the matter is referred back to the Board for redetermination by a differently constituted panel.

[69] There is no question for certification.

JUDGMENT

[70] THIS COURT ORDERS AND ADJUGES that this application for judicial review is allowed.

Application allowed.
Editor: Sharon McCartney/pdk
[End of document]

Figure 3.4: A sample of a case law from The Federal Court of Canada case law database

Table 3.1: Case law retrieval task data analysis

	Train set 2019	Test set 2019	Test set 2020
#Words/Doc	2462	2443	3232
#Paragraphs/Doc	23	23	28
#Maximum Words	10827	10827	127263
#Maximum Paragraphs	119	119	1139
#Samples	285	61	130
#Candidate cases	57000	12200	26000
#Average notice cases/Case query	5.21	5.41	4.89

Table 3.2: Case law entailment task data analysis

	Train set 2019	Test set 2019	Test set 2020
#Samples	181	44	100
#Candidate paragraphs/Decision query	32.12	32.91	36.72
#Entailed paragraphs/Decision query	1.12	1.02	1.25

To identify the decision sentence, we apply the TextRank algorithm [85] - a graph-based ranking algorithm for automatic sentence extraction. For each candidate paragraph, we build a graph to represent the text, where the graph vertices are representative of the units to be ranked. For the task of topic sentence extraction, each sentence is represented as a vertice in the graph. Let $G = (V, E)$ is an undirected graph with a set of vertices V and a set of edges E, where E is a subset of $V \times V$.

We establish a connection between two sentences if these sentences are semantically connected. For example, the topic sentence states the main idea of the paragraph, and the remaining sentences in the paragraph give specific details related to the topic sentence. Therefore a link can be drawn between any two such sentences that share common content. To estimate the semantic similarity between two sentences, we use cosine similarity as follows:

$$Sim_{ij} = \frac{s_i^T \cdot s_j}{||s_i||||s_j||} \tag{3.1}$$

where s_i and s_j represent the vector representations of two sentences s_i and s_j.

We need to build a sentence embedding method that can successfully model the semantic similarity between two sentences. For this reason, we employed smooth inverse frequency [3] to embed each sentence in the text. This method does much better than sentence embeddings generated from simply computing the arithmetic mean for all word vectors constituting a sentence. This method is derived from an assumption that a sentence has been generated by a random walk of a discourse vector on a latent word embedding space, and by including smoothing terms for frequency words. The sentence embedding is calculated as follows:

$$v_s = \frac{1}{|s|} \cdot \sum_{w \in s} \frac{a}{a + p(w)} \cdot v_w \tag{3.2}$$

where each word $w \in s$ is represented as GloVe 300-dimension word vectors v_w. The weight of a word embedding w is $a/(a+p(w))$, where a is a parameter that is typically set to 0.001, and $p(w)$ is the estimated frequency of the word in the corpus. The weighting scheme emphasizes low probability words that likely carry more semantic content and de-emphasizes commonly used words with high probability. One of the problems of just adding up word vectors to generate a sentence embedding is that the resultant vector has huge components in semantically meaningless directions (Sanjeev Arora and Yingyu Liang and Tengyu Ma, 2017 [3]).

To diminish the influence of semantically meaningless directions common to the whole corpus, the smooth inverse frequency method removes the projections of the average vectors on their first singular vector. In other words, all sentence vectors s in the set of sentences S are concatenated into a matrix M from which the first singular vector u is removed from each weighted average as $v_s = v_s - uu^T v_s$. Removing them yields better quality sentence vectors that even do better than sentence embeddings generated by sequence models like RNN or LSTMs on sentence similarity tasks where word order does not matter.

After representing the paragraph as a weighted graph, each vertex's score is set to an initial value of 1. The model computes the score of vertices by using the graph-ranking algorithm from TextRank. Given a vertex v_i, let $Adj(v_i)$ be the set of vertices connected to v_i. The vertex score in the graph-based ranking algorithm is calculated as follows:

$$Score(v_i) = \frac{1-d}{\mid V \mid} + d * \sum_{v_j \in Adj(v_i)} \frac{EdgeWeight(v_i, v_j)}{\sum_{v_k \in Adj(v_j)} EdgeWeight(v_j, v_k)} \cdot Score(v_j) \tag{3.3}$$

where $\mid V \mid$ is the total number of vertices in a given document, and d is a damping factor that is usually set to 0.85. The score of vertices is set by specific initialized values, and the computation iterates until convergence below a given threshold is achieved.

After running the ranking algorithm on the graph, the top-ranked vertex from the graph is selected as a topic sentence for the input text. We assume that the decision sentence is supported by the other sentences. The negative samples are selected randomly from the scopus. The ratio between negative and positive samples is 2:1. The weak label supporting dataset is built from 9608 Canada case law and include 293532 supporting text pairs.

3.5 Case Law Retrieval with Supporting Model

3.5.1 Supporting Model

In recent years, pre-train language models, which were utilized by large unlabeled data sets, illustrate usefulness in natural language processing tasks. Especially, BERT is trained on masked language model and next sentence prediction improved common language representation with a huge corpus. Additionally, BERT [38] architecture contains 12 transformers block, which could capture rich contextual information. Therefore, we apply BERT to build a supporting model.

We use BERT base pre-train model and tune all parameters of it on the weak-labeling supporting dataset in section 3.4.2. The input is a pair of decisions and support sentences. The input is presented in a sequence includes two segments, the first token is a special token “[CLS]” and another special token “[SEP]” which separates two segments. We take the final hidden state **h** of the first token as the presentation of the decision and supporting sentence pair. A single fully-connected layer is added on the top of BERT as a binary classification:

$$P(y|\mathbf{h}) = \mathit{sigmoid}(W\mathbf{h}) \tag{3.4}$$

where W is the trainable parameters. The loss function is a cross-entropy loss:

$$\mathit{Loss} = -(y\ log(p) + (1-y)\ log(1-p)) \tag{3.5}$$

In the training phase, we use Adam [66] to optimize all parameters of the model with the learning rate is $2e^{-5}$.

3.5.2 Combination of Supporting Model and Lexical Model

Besides supporting the model, we also exploit multiple similarity measurements such as lexical similarity (keyword matching) and semantic similarity (context matching). Although lexical similarity and semantic similarity are quite different from each other, they can be combined and complementary. We use BM25 [120] to build the lexical model.

BM25 is a bag-of-words retrieval function that ranks a document based on the query terms appearing in each document to estimates the relevance of the document to a given search query [120].

Given a query Q, containing keywords q_1,..., q_n, the BM25 score of a document D is:

$$\mathit{score}(D, Q) = \sum_{i=1}^{n} \mathit{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)} \tag{3.6}$$

where $f(q_i, D)$ is q_i's term frequency in the document D, $|D|$ is the length of the document D in terms, and $avgdl$ is the average document length in the text corpus from which documents are drawn. k_1 is a parameter that controls term frequency saturation curve. Parameter b controls the importance of document length. $IDF(q_i)$ is the inverse document frequency weight of the query term q_i. It is calculated as:

$$\mathit{IDF}(q_i) = \ln\left(1 + \frac{N - n(q_i + 0.5)}{n(q_i) + 0.5}\right) \tag{3.7}$$

where N is the total number of documents in the corpus, and $n(q_i)$ is the number of documents containing q_i.

Given a query q and list of document $D = \{d_1, d_2, ..., d_n\}$, the final score is combination of supporting score and lexical score, it was calculated as follow:

$$\mathit{score}(q, d_i) = \alpha \times \mathit{Norm}(\mathit{score}_{supporting}(q, d_i)) + (1 - \alpha) \times \mathit{Norm}(\mathit{score}_{lexical}(q, d_i)) \tag{3.8}$$

where α is hyperparameter selected during the experiment, and $\mathit{Norm}()$ is Min-Max normalization with the general fomula:

$$Norm(score(q, d_i)) = (score(q, d_i) - min\_score)/(max\_score - min\_score) \tag{3.9}$$

in which, $min\_score = \min_j(score(q, d_j))$ and $max\_score = \max_j(score(q, d_j))$

3.5.3 Case Law Retrieval With Scoring Method

In previous work, case law retrieval and case law entailment models are built independently and separately. In this work, we try to construct a general model for the supportive relationship in both phases.

Figure 3.1 illustrates an example of our supportive component between a query case and a candidate case. According to our definition of support relation at the Introduction section: a support relation as $support(a, b)$ (a supports b), case law retrieval and entailment phase tasks are formalized in section 3.2

We apply a combined score to identify the supporting relation in both phases of case law retrieval and entailment system. In case law retrieval phase, the scoring method is used to extract supporting components between the query case and candidate cases. While the scoring method is applied directly for the decision and each query case paragraph in case law entailment phase.

3.6 Experiments and Results

In our experiment, we evaluate our system on the datasets provided by Competition on Legal Information Extraction/Entailment (COLIEE) in 2020. The data for this task is based on the Federal Court of Canada case law[2].

3.6.1 Evaluation Metric

We adopt precision, recall, and F-measure (F_1) for task 1 and task 2 in COLIEE [111]. Precision means how many retrieved cases (paragraphs) for all queries are

[2] https://ca.vlex.com/

correct and recall means how many target cases (paragraphs) are retrieved. F-measure is computed as:

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.10}$$

3.6.2 The Case Law Retrieval Results

In the case law retrieval phase, we use the data in the COLIEE 2020 task 1, which includes the 2019 train set, 2019 test set, and 2020 test set, respectively. Each example consists of 1 query case and 200 candidate cases.

The case law documents present in raw text, the metadata information is provided as in Section 3.2, the content of the case present as the list of paragraphs. The metadata has a high rate of missing, so we build our model based on the textual content from the list paragraphs. Some cases were written both in English and French. We only use the English version as inputs in our experiments.

Deep learning models are expensive time and resource consumption. Therefore, we use lexical scoring to filter top n cases from the given set of candidate cases together with combined scores. To ensure time performance, we report the results with $n = 25$ and $n = 30$ in our experiment. On average, to retrieve the list of supporting case law of the given case, it takes from 15 minutes to 1 hour.

Subsequently, we separate the whole query cases and candidate cases into each set of paragraphs. We calculate the combined score between the query paragraph and the candidate paragraphs. Where the query paragraph is each one in a given query case, and the candidate paragraph is each one in the corresponding candidate cases. The supporting relation is established when the score is greater than a given threshold. Typically, a candidate paragraph supports a query paragraph. So, by our experiment, we set the threshold of 0.97.

Table 3.3: The results on COLIEE 2019 task 1 train set

Method	**Precision**	**Recall**	**F1**
Lexical model (BM25)	0.4065	0.5189	0.4559
Supporting model (task 2 data, $n = 30$)	0.4818	0.3381	0.3973
Supporting model (task 2 data, $n = 25$)	0.4731	0.3646	0.4118
Supporting model(task 2 + weak labeled data) ($n = 30$)	0.5221	**0.6568**	0.5817
Supporting model(task 2 + weak labeled data) ($n = 25$)	0.5464	0.6464	**0.5922**
Combination model ($n = 30$)	0.6104	0.5411	0.5737
Combination model ($n = 25$)	**0.6248**	0.5382	0.5783

Table 3.4: The results on COLIEE 2019 task 1 test set

Method	**Precision**	**Recall**	**F1**
Lexical model (BM25)	0.4181	0.6030	0.4938
Supporting model (task 2 data, $n = 30$)	0.4394	0.3680	0.4006
Supporting model (task 2 data, $n = 25$)	0.4485	0.3371	0.3849
Supporting model(task 2 + weak labeled data, $n = 30$)	0.5519	**0.6606**	0.6014
Supporting model(task 2 + weak labeled data, $n = 25$)	0.5710	0.6455	**0.6060**
Combination model($n = 30$)	0.5910	0.6000	0.5955
Combination model ($n = 25$)	**0.6125**	0.5939	0.6031

Table 3.5: The result comparision with task 1 COLIEE leaderboard 2019

Team/Method	**Precision**	**Recall**	**F1**
IITP	0.6260	0.3850	0.4770
HUKB	**0.7020**	0.4000	0.5100
ILPS	0.6810	0.4333	0.5296
JNLP	0.6000	0.5545	0.5764
BERT-PLI [132]	0.6026	0.5697	0.5857
Supporting model(task 2 + weak labeled data) ($n = 25$)	0.5710	**0.6455**	**0.6060**
Combination model ($n = 25$)	0.6125	0.5939	0.6031

Although our supporting model does not use the "noticed" case-case relation provided in the training data for COLIEE task 1, the results in Tables 3.3, 3.4, 3.5, 3.6 and 3.7 show that the supporting model and the combination model achieve good performance. Tables 3.3, 3.4 and 3.6 show the performance on the train 2019, test 2019, and test 2020 set for task 1. The supporting model and the combination model achieve approximately similar performance in the F1 score. The two methods significantly outperform the baseline methods (the lexical model) by a large margin. The system based on our supporting case law definition can extract the components of the candidate case support to which part of the query case correctly. The approach is suitable for an arbitrary length of raw case law documents. The supporting model has a better semantic understanding than the traditional method like bag-of-words IR models. The supporting model achieves F1 higher by 10-12% than the lexical model. Therefore, the combination model set the α of 0.85, which means the support score has more influence than the lexical score.

Experiments in Table 3.3, 3.4 and 3.10 also show that the supporting model built from weak label and task 2 dataset gives much higher results than the supporting model built only on task 2 dataset. One of the reasons is that the task 2 dataset is too small, which is difficult to build a comprehensive support relationship model.

Table 3.6: The results on COLIEE 2020 task 1 test set

Method	**Precision**	**Recall**	**F1**
Lexical model (BM25)	0.4071	0.7579	0.5297
Supporting model (task 2 data, $n = 25$)	0.6855	0.4575	0.5488
Supporting model (task 2 data, $n = 30$)	0.6981	0.4387	0.5388
Supporting model(task 2 + weak labeled data) ($n = 30$)	0.5146	**0.8050**	0.6278
Supporting model(task 2 + weak labeled data) ($n = 25$)	0.5323	0.7893	0.6358
Combination model ($n = 30$)	0.5423	0.7563	0.6316
Combination model ($n = 25$)	**0.6395**	0.6667	**0.6528**

Table 3.7: The result comparision with task 2 COLIEE leaderboard 2020

Team/Method	**Precision**	**Recall**	**F1**
cyber	-	-	**0.6774**
TLIR	-	-	0.6682
UB	-	-	0.5866
iiest	-	-	0.5288
TR	-	-	0.3800
DACCO	-	-	0.2077
UB	-	-	0.0592
taxi	-	-	0.0457
Supporting model ($n = 25$) (JNLP)	0.5323	0.7893	0.6358
Combination model ($n = 25$)	0.6395	0.6667	0.6528

In previous works, the supporting relation is identified through the similarity method, while our system captures this one directly. The supporting model achieves F1 of 0.5922 on the COLIEE 2019 train set and F1 of 0.6060 on the COLIEE 2019 test set, respectively. On the COLIEE 2020 test set, this model achieves F1 of 0.6358. Our model is trained based on the supporting relation, while the gold labels are “notice” relation, which supporting cases are chosen by lawyers. In Table 3.1, the average number of “noticed” cases in the COLIEE 2019 test set (4.89) is lower than both in the COLIEE 2019 train set (5.21) and the COLIEE 2019 test set (5.41). It leads to a higher recall in the COLIEE 2020 test set (0.8050). The precision score of the supporting model is quite stable in all three sets.

The combination of the supporting model and the lexical model does not improve much performance in F1. Mainly, the improvement shows more balance in recall and precision. So the combination model gets the best performance in precision score on all three sets. The lexical model provides useful information for identifying relevant cases.

Table 3.5 and 3.7 show the best results of the top teams in the COLIEE competition leaderboards in 2019 and 2020. Although other teams use “noticed” case-case relation provided by the training data, our models (not use “noticed” case-case data) still get good performance in both of the competitions in 2019 and 2020 (In 2020 competition, we join as JNLP team).

As shown in Table 3.9, a lot of supportive cases are about medicine even though they are not “noticed” cases, which leads to redundant prediction results and a large gap between recall and precision. The gold noticed case is 87, while the predicted cases are 118, 76, and 87. On the other side, Table 3.8 shows a sample of a large number of “noticed” cases. Our model predicts 14 candidate cases as supportive cases, while the gold label is 13 ones. The number of correct predictions is 10/14. The model usually works well in a large number of ‘noticed’ case samples.

The goal of our system is to provide maximum support to lawyers, which could retrieve as many supporting cases as possible. Therefore, we appreciate it when the system has a high recall.

Table 3.8: The output of sample 522 in COLIEE task 1 2020

Paragraphs of Query Case 522
[...][2] The Applicant is a Sri Lankan woman who came to Canada on May 12, 2010, seeking refugee protection and claiming to be a human rights activist who was targeted because she obtained information that could embarrass the government in Sri Lanka [...] [...][3] for permanent residence from within Canada on H&C grounds. Both of these applications were refused, so the Applicant again sought relief[...] [...][14] The Applicant submits that no deference is owed to the Officer on questions of procedural fairness. For the other issues raised by the application, the Applicant acknowledges that the standard of review is one of reasonableness , but emphasizes that the Court must assess the Officer’s reasons on their own merits and not substitute better reasoning to justify the outcome[...] [...][15] The Applicant argues that H&C applicants are owed more than a minimal level of procedural fairness, and it includes a right to an interview when credibility is at stake[...] [...][25] Furthermore, the Respondent submits that the Applicant bore the burden to supply evidence and was given 30 days[...] [...][31] The general rule in this regard is that the evidentiary record for purposes of a judicial review application is restricted to that which was before the decision-maker[...]

<table>
<tr><td colspan="2">[...][34] As both parties acknowledge, the Supreme Court has said that the standard of review for procedural issues is nominally correctness. Reviewing courts are responsible for determining whether the process was fair, although relief may be withheld if any error[...]
[...][35]As for the other issues, the standard of review is reasonableness. Accordingly, this Court should not intervene so long as “the reasons allow the reviewing court to understand why the tribunal made its decision and permit it to determine whether the conclusion is within the range of acceptable outcomes”[...]
[...][40] With respect to the Applicant’s request for an interview, the Supreme Court has said that “an oral hearing is not a general requirement for H&C decisions”. However, that is not invariably the case, and an interview should be held if credibility is a determinative issue[...]
[...][45] The Officer simply observed that, although not conclusive, an applicant’s past personal experiences were relevant to establishing a link to the country conditions. The Officer rejected some of the accounts in the Applicant’s statutory declaration [...]. A review of the Applicant’s statutory declaration shows that finding is justifiable, and the Court cannot disturb it without re-weighing the evidence[...]</td></tr>
<tr><th>Predicted Case</th><th>Noticed</th></tr>
<tr><td>Candidate Case 72: [...][1] This is an application of the Immigration and Refugee Protection for judicial review of two decisions of a Senior Immigration Officer, both dated November 30, 2012, which refused the Applicant’s Pre-Removal Risk Assessment application and her application for permanent residence from within Canada on humanitarian and compassionate grounds [...]</td><td>Yes</td></tr>
<tr><td>Candidate Case 1: [...][12]The grounds upon which the application was based were stated as being the Applicant’s establishment in Canada and the risk the Applicant would face if returned to Sudan. The Officer stipulated that the Applicant bore the onus of demonstrating that the hardship of having to obtain a permanent residence visa from outside Canada would be unusual and undeserved or disproportionate[...]</td><td>Yes</td></tr>
</table>

Candidate Case 23: [...][42] The officer had a duty to assess this evidence and determine if it supports a finding of an unusual and undeserved or disproportionate hardship, but instead the officer deliberately ignored the evidence of this potential hardship entirely. That was due to an incorrect interpretation of subsection 25(1.3) and I cannot determine from the reasons whether the result would have been the same had that error not been made.	Yes
Candidate Case 176: [...][54] The situation in the case at bar is strikingly similar to the case before Justice Mactavish in Adu . The only significant difference is the Officer's reliance on Uddin to the effect that the H&C process is not designed to eliminate hardship, but rather is designed to provide relief from unusual, undeserved or disproportionate hardship.[...]	Yes
Candidate Case 148: [...][47]The Applicants submit that the Officer identifies numerous positive factors about the Applicants' establishment in Canada but then concludes, without reasons, that the hardship they would experience if they return to Lebanon would not constitute unusual, undeserved or disproportionate hardship were they to return to Lebanon to apply for permanent residence.	Yes
Candidate Case 189: [...] [2] The Applicant, after June 29, 2010, made an application for permanent residence in Canada on humanitarian and compassionate grounds. [...]. In a decision dated January 26, 2012, that application was denied. This is a judicial review of that decision [...]	Yes
Candidate Case 19: [...][27] In addition to the breach of procedural fairness the applicants have raised a number of grounds to attack the impugned decision. The broad issue is whether the officer's decision, considered as a whole, can sustain a somewhat probing examination by the Court.	Yes
Candidate Case 182: [...][13] An officer is not obliged to disclose, prior to making a decision, all the information consulted where the information consists of commonly consulted public information as opposed to novel and significant information which may affect the disposition of the matter [...]	Yes

Candidate Case 101: [...] [34] The applicant had failed to comment on additional evidence when given the opportunity to do so. The documents were only provided as part of the certified tribunal record in response to this application for judicial review. The applicant has them now and can make informed submissions to the next immigration officer who will consider the matter [...]	Yes
Candidate Case 42: [...][5] The parties agree that the decision is reviewable on a standard of reasonableness. The role of the court on review of a decision on a reasonableness standard is to determine of whether "the decision falls within a range of possible, acceptable outcomes which are defensible in respect of the facts and the law"[...]	No
Candidate Case 20: [...][7] I agree with the Applicant that the RPD conflated its credibility findings about the Applicant with a no credible basis finding. The RPD failed to properly consider whether there was any credible evidence, including the testimony of the other witnesses, to support the Applicant's claim [...]	No
Candidate Case 129: [...][41] I agree with the respondent that in reviewing and comparing the updated documents, the applicant did not provide convincing evidence that there were novel and significant changes in the general country conditions[...]	Yes
Candidate Case 7: [...][34] Again, I find nothing argumentative about this evidence, as it is purely factual. However, the Applicants' submissions, that this evidence is irrelevant and was not before the PMRA when it made its decision, require more detailed consideration[...]	No
Candidate Case 3: [...] [5] The PRRA officer found at page 3 of his decision that the new evidence that had been presented by the applicant did not establish a personalized risk to the applicant[...]	No

Table 3.9: The output of sample 521 in COLIEE task 1 2020

Query case paragraphs	Support paragraphs
[10] Therefore, deference must be given to the PRRA officer's analysis of the evidence in the record, which falls within his expertise.	Candidate case 118: [15] A PRRA officer's findings following a hearing are to be given considerable deference. In this instance, the officer closely examined the evidence presented and responses given and analyzed those responses using usual and customary indicia of credibility.
[14] In his reasons, the officer notes that the evidence does not allow for the conclusion that the applicant has a leadership or spokesperson role within this organization. Although the name of the applicant is identified in one of the newspaper articles, the officer notes that there is no mention to suggest that he has acted as representative of the CASS.	Candidate case 76: [31] There are reasonable grounds to believe that these organizations are or have been engaged in activity that is part of a pattern of criminal activity planned and organized by a number of persons acting in concert in furtherance of the commission of an offense punishable under an Act of Parliament by way of indictment.[32] This unequivocal conclusion raises no serious question and must be held as proven. The applicant is a member of these organizations
[17] In the absence of probative evidence showing a personalized risk, it was up to the officer to conclude that the risks raised by the applicant if he were to return to Algeria are not supported by the objective and subjective evidence.	Candidate case 87: [21] The Applicant submits that the Officer erred in rejecting his PRRA for a failure to provide sufficient evidence of a particularized risk when the Officer unreasonably dismissed all of his evidence of particularized risk. Despite the Officer saying that he was assigning ”little weight” to the evidence, it is clear that he actually assigned no weight to the evidence. Had this evidence not been discounted, the Applicant could have established the links between the various events.

3.6.3 The Casw Law Entailment Results

The corresponding noticed cases in the previous phase are separated into a set of paragraphs. Given a decision, we calculate the combined score between the given decision and each of the paragraphs. Similar to the previous phase, the threshold of score is 0.97 and $\alpha = 0.85$.

In this phase, we conduct two experiments. For the first one, we use directly the already trained supporting model in task 1 together with BM25 scoring for finding the support paragraphs for the given entailed decision. For the second one, we enhance the system in the first experiment by tuning the supporting model using the training data for task 2 COLIEE.

Table 3.10: The results on COLIEE 2019 task 2 train set

Method	Precision	Recall	F1
Lexical model (BM25)	0.6257	0.5792	0.6015
Supporting model(weak labeled data)	0.5926	0.6337	0.6124
Combination model (weak labeled data)	**0.6753**	**0.6485**	**0.6616**

Table 3.11: The results on COLIEE 2019 task 2 test set

Method	Precision	Recall	F1
Lexical model (BM25)	0.5682	0.5556	0.5618
Supporting model (task 2 data)	0.6667	0.6667	0.6667
Combination model (task 2 data)	0.7111	0.6889	0.6882
Supporting model (weak labeled data)	0.6818	0.6667	0.6742
Combination model (weak labeled data)	0.6818	0.6667	0.6742
Supporting model (task 2 + weak labeled data)	0.7045	0.6889	0.6966
Combination model (task 2 + weak labeled data)	**0.7174**	**0.7333**	**0.7253**

Table 3.12: The result comparision with COLIEE task 2 leaderboard 2019

Team/Method	Precision	Recall	F1
UA	0.6538	0.7556	0.7010
IITP	0.7045	0.6889	0.6966
TRCase	0.6818	0.6667	0.6742
JNLP	0.5909	0.5778	0.5843
TTCL	0.4000	0.8000	0.5333
ielab	0.4545	0.4444	0.4494
UBLTM	0.1273	0.6222	0.2113
Combination model (weak labeled data)	0.6818	0.6667	0.6742
Combination model (task 2 + weak labeled data)	**0.7174**	**0.7333**	**0.7253**

Table 3.13: The results on task 2 2020 test set

Method	**Precision**	**Recall**	**F1**
Lexical model (BM25)	0.528	0.6346	0.5764
Supporting model (task 2 data)	0.5600	0.5691	0.5645
Combination model (task 2 data)	0.5440	0.6126	0.5763
Supporting model (weak labeled data)	0.5368	0.5840	0.5594
Combination model (weak labeled data)	0.6727	0.5920	0.6298
Supporting model (task 2 + weak labeled data)	0.6014	**0.6640**	0.6312
Combination model (task 2 + weak labeled data)	**0.7358**	0.6240	**0.6753**

Table 3.14: The result comparision with COLIEE task 2 leaderboard 2020

Team/Method	**Precision**	**Recall**	**F1**
taxi	-	-	0.6180
TLIR	-	-	0.6154
cyber	-	-	0.5897
iiest	-	-	0.5867
UA	-	-	0.5425
TR	-	-	0.4107
DACCO	-	-	0.0622
Combination model (weak labeled data)	0.6727	0.5920	0.6298
Combination model (task 2 + weak labeled data)	**0.7358**	**0.6240**	**0.6753**

In this phase, Tables 3.10, 3.11, 3.13, 3.12 and 3.14 show that the combination model is stable working and outperformance than the lexical model. Despite of not using the training data for task 2 COLIEE, the combination model still achieves good performance in competition 2020, our combination model gets over than other teams with F1 score of 0.6298 presented in Table 3.7. The results in Table 3.11 and 3.13 also show that the F1 scores of the supporting model built from the weak label dataset and the supporting model built from the task 2 training dataset are approximately equal. This shows the amazing efficiency of the weak label supporting dataset, the supporting model built without labeled task 2 dataset work well in the entailment task. Different from the previous phase, the combination model gets better than the supporting model in almost all experiments. It means that the decision and the candidate paragraphs have more lexical similarities. So the combination score has a positive effect on improving performance. In case law entailment, precision and recall get more balanced than the previous phase, because of the stability of the average number of "noticed" paragraphs: 1.12 on the 2019 train set, 1.02 on the 2019 test set, and 1.25 on the 2020 test set (shown in Table 3.2).

Moreover, fine-tuning the supporting model on the weak label dataset and the train set for task 2 with the gold label further significantly improves the performance. The combination model plus turning supporting model achieves the best performance in both the 2019 and 2020 test sets. Especially, in the 2020 COLIEE competition, our combination model by passes the other team's ones by a large margin (about 6%). In previous work, they usually combined learning model and heuristic features to solve the case law entailment task. In our approach, the supporting relation is learned automatically and directly. So the learning model is stable working and gets the best result on both of the COLIEE competition leaderboards in 2019 and 2020 (Table 3.5 and 3.14).

Table 3.15: The output of sample 414 in COLIEE task 2 2020 (P: Prediction, G: Gold label)

<table>
<tr><td colspan="3"><b>Decision:</b> Where a grievance procedure, as prescribed in a statute, constitutes an adequate alternate remedy, it ought to be completely followed before turning to courts</td></tr>
<tr><td>Rank</td><td>Candidate Paragraphs</td><td>P/G</td></tr>
<tr><td>1</td><td>[29] Put another way, the redress authority must suspend until such time as the court of law has decided the issue brought before it by the complainant. Paragraph 16 cannot be taken to mean that a member can, at his discretion, abandon the grievance process provided under art. [...]</td><td>1/1</td></tr>
<tr><td>2</td><td>[23] Mr. Justice Stone then examined whether the remedy afforded to the member, pursuant to the redress of grievance procedure set out at art. [...]. Although he recognized that the time required to pursue the matter by way of judicial review would probably be less than that required to pursue the matter through the grievance process, [...]</td><td>0/1</td></tr>
<tr><td>3</td><td>[30] Like Mr. Justice Stone in Anderson , there is no necessity to consider the first issue. However, I will do so in the event that I am wrong regarding the existence of the alternate remedy.</td><td>0/0</td></tr>
<tr><td>4</td><td>[8] A second issue was raised by the respondent. The respondent submits that I should dismiss the applicant's application for judicial review "on the ground that the applicant has an adequate alternate remedy, namely redress of grievance to the Chief of Defence Staff and ultimately to the Minister of National Defence". [...]</td><td>0/0</td></tr>
</table>

5	[28] The applicant submits that it does not appear that art. 19.26(16) was brought to the attention of the Court of Appeal in Anderson . Counsel argues that it is clear from art. 19.26(16) that the was not designed to prevent a member of the Armed Forces from seeking a remedy before the courts. [...]	0/0
...	...	...
39	[31] The first issue is whether Lieutenant General Fischer was wrong in construing I am in agreement with his interpretation.	0/0

Table 3.15 presents a sample in case law entailment task. The model predicts only one candidate paragraph supporting the given decision because the average number of "noticed" paragraphs in the 2019 train data and the 2019 test data is 1.12 and 1.02. It also has a negative impact on recall when models often predict fewer supporting paragraphs than the gold label.

3.7 Summary

In this chapter, we have proposed a novel approach, called *supporting model*, to the retrieval and entailment problems for case law data. Through data analysis and observation, we found the critical features of the supporting relationship in the paragraphs of the cases, thereby building a weak-label dataset that can represent such features. This dataset our approach apart from the previous work. Making use of the strength of the pre-trained models, we can directly formulate the support relationship in case law. Besides, in our system, we proposed to use multiple measures in combination to evaluate the supporting relationship. With the proposed method, our system has achieved significant results in the COLIEE 2020 competition and created a bold gap in comparison with the results of other teams in the entailment task.

Chapter 4

Knowledge Graph for Statutory – Case Law Retrieval

In this chapter, e develop a novel approach to a knowledge graph encompassing case law documents and relevant legislation to improve legal information organization and retrieval. Our method involves data collection, entity extraction, and graph construction. The constructed heterogeneous graph connects courts, cases, domains, and laws, significantly enriching information provided by retrieval systems. Our approach demonstrates potential in case analysis, legal recommendations, and decision support, providing valuable insights and resources for the legal domain.

This work was published in the **15th International Conference on Knowledge and Systems Engineering (KSE) 2023** (indexed by Scopus) [VTHY3].

4.1 Legal Knowledge Graph

A legal knowledge graph [42, 104, 128] represents structured legal information in a graph format, capturing relationships between legal entities such as statutes, regulations, cases, and concepts. This graph-based representation enables a more comprehensive understanding of legal domains by organizing and connecting disparate legal data points. By modeling legal knowledge as interconnected nodes and edges, legal knowledge graphs facilitate various tasks, including legal research, information retrieval, and decision support systems.

In case law and statute law presentation, knowledge graphs provide structured frameworks for organizing and analyzing legal information derived from cases and statutes, respectively. Utilizing graph theory, these knowledge graphs represent legal entities, such as cases, statutes, regulations, legal domains, and relationships, as nodes and edges.

The method of constructing a knowledge graph serves as a suitable tool for identifying and representing the relationships between case laws and relevant laws [22, 44]. Knowledge graphs can effectively depict vast amounts of knowledge with semantic meaning, facilitating easy access and structured querying. These knowledge graphs are designed in a user-friendly manner, catering to non-expert users such as lawyers, judges, scholars, etc., enabling them to easily utilize and explore the information. Moreover, knowledge graphs can be applied to enhance various downstream tasks in the legal domain such as information retrieval [32], question-answering [34, 137], classification [5], and more.

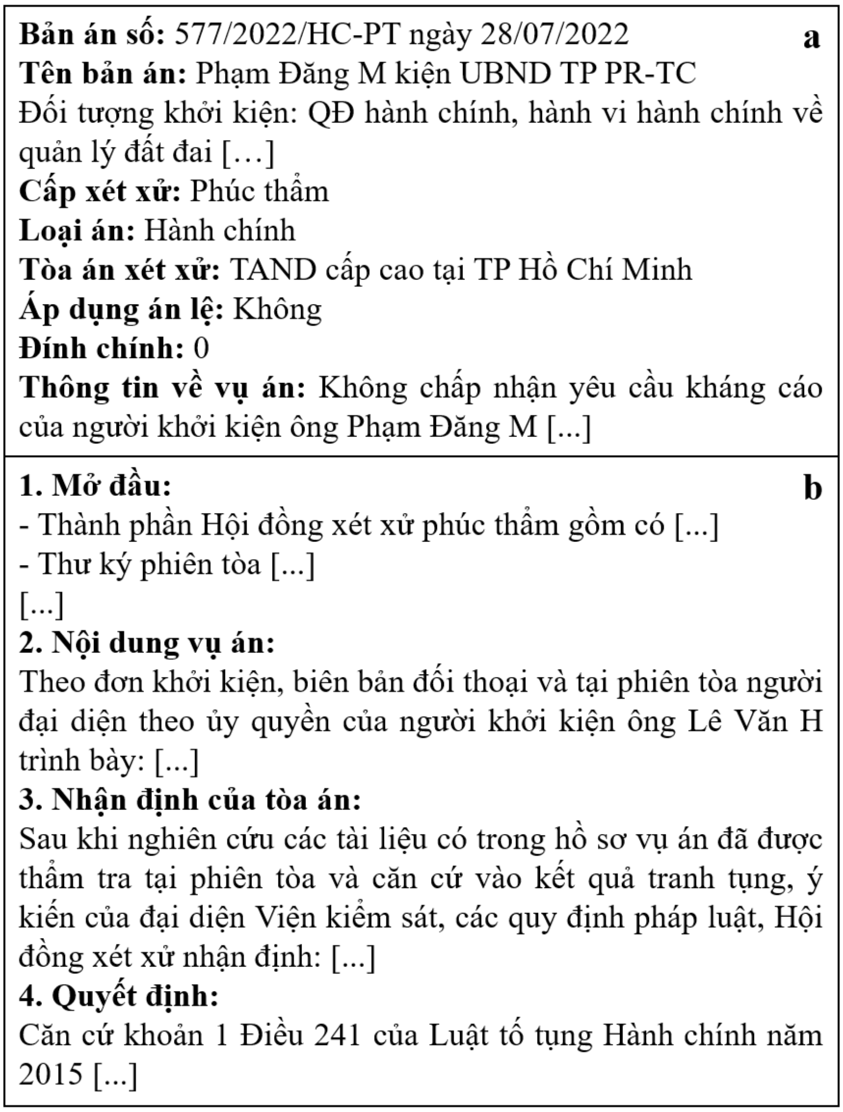
4.2 Vietnamese Legal Case Knowledge Graph Definition

A case law archived on the website of the Vietnam Supreme People's Court consists of two parts: metadata and the case document. Figure 4.1 illustrates the structure and content of a case law. The metadata contains basic information about the case, including the case number, case name, type of case, etc. The body of the case law document comprises four sections: the Introduction, the Content of the case, the Court's Judgment, and the Court's Decision. The description of each part is shown in Table 4.1.

Table 4.1: The description of a law document.

Part	**Description**
Introduction	details of case, court, defendant, plaintiff, related parties (e.g full name, date of birth, address of parties)
Content of the case	opinions of case, court, defendant, plaintiff, related parties
Court's judgment	Opinions, analysis of the court
Court's decision	Decisions of the court based on above parts

The aim of this study is to represent relationships within legal cases. Therefore, we construct a knowledge graph comprising legal actors such as: statute laws, case laws,



Bản án số: 577/2022/HC-PT ngày 28/07/2022 **a**
Tên bản án: Phạm Đăng M kiện UBND TP PR-TC
Đối tượng khởi kiện: QĐ hành chính, hành vi hành chính về quản lý đất đai […]
Cấp xét xử: Phúc thẩm
Loại án: Hành chính
Tòa án xét xử: TAND cấp cao tại TP Hồ Chí Minh
Áp dụng án lệ: Không
Đính chính: 0
Thông tin về vụ án: Không chấp nhận yêu cầu kháng cáo của người khởi kiện ông Phạm Đăng M [...]

1. Mở đầu: **b**
- Thành phần Hội đồng xét xử phúc thẩm gồm có [...]
- Thư ký phiên tòa [...]
[...]
2. Nội dung vụ án:
Theo đơn khởi kiện, biên bản đối thoại và tại phiên tòa người đại diện theo ủy quyền của người khởi kiện ông Lê Văn H trình bày: [...]
3. Nhận định của tòa án:
Sau khi nghiên cứu các tài liệu có trong hồ sơ vụ án đã được thẩm tra tại phiên tòa và căn cứ vào kết quả tranh tụng, ý kiến của đại diện Viện kiểm sát, các quy định pháp luật, Hội đồng xét xử nhận định: [...]
4. Quyết định:
Căn cứ khoản 1 Điều 241 của Luật tố tụng Hành chính năm 2015 [...]

Figure 4.1: The structure of a case law (a: meta-data, b: case content)

courts, legal domains, as well as their interrelations. This approach facilitates a comprehensive understanding of the legal landscape by capturing the intricate connections between legal entities and concepts. Moreover, it enables the exploration of legal precedent, legislative frameworks, and jurisprudential trends within specific legal domains. By modeling legal knowledge as interconnected nodes and edges, this knowledge graph serves as a valuable resource for legal research, information retrieval, and decision support systems.

We construct the Vietnamese legal case knowledge graph based on a heterogeneous graph, which is can have nodes and edges of different types. A heterogeneous graph $G = (V, E)$ contains an entity set V and a relation set E with an entity type mapping function $f : V \rightarrow A$ and a relation type mapping function $g : E \rightarrow R$. A and R denote the sets of entity types and relations types, where $|A| + |R| > 2$. Particularly, we define

4 types of entity based on the characteristic of the Vietnamese case law, including:

- Case node, which embeds information about each judgment/trial that is currently in effect.
- Domain node, which embeds information about crimes, types of disputes and decisions.
- Court node embeds information about every court's name and level in the juridical system.
- Law node contains the name of specific law/code of law

There are a total of 3 types of relations between entities, including:

- Decide relation between courts and cases, indicating the relationship of a particular court hearing the trial.
- Belong-to relation between cases and domains, indicating the relationship of a particular domain and subdomain under which the case falls.
- Based-on relation between cases and laws, indicating the relationship of a particular judgment or decision that has referenced a set of laws/codes of law to support its verdict.

In a heterogeneous graph, two entities can be connected via different paths. Formally, these path are called meta-paths. A meta-path P is defined in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_k} A_{k+1}$, which presents a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_k$ between A_1 and A_{k+1}, where $\circ$ denotes the composition operator on relations. Two case laws can be connected via different meta-paths, e.g. Case-Court-Case (CCC) or Case-Domain-Case (CDC). Different meta-paths describe semantic relationships in different views. For instance, the CCC path means these cases were judged by the same court, while the CDC path denotes that they belong to the same domain.

4.3 Knowledge Graph Construction

In this section, we describe our approach to constructing a knowledge graph for Vietnamese case laws. Each of the following subsections presents the steps of our approach in detail. Figure 4.2 visualizes a portion of the graph with a case, court, domain, and law nodes.

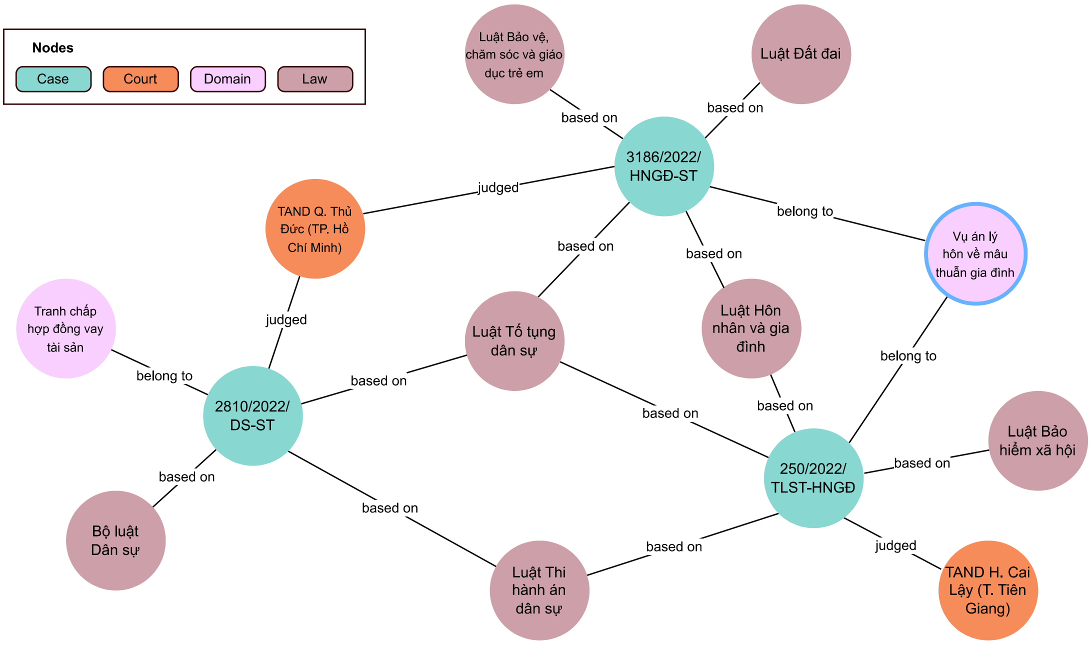


Figure 4.2: The knowledge graph visualization

4.3.1 Data Crawler

The database contains 9,578 court cases published by the Supreme People's Court of Vietnam and 225 Vietnamese laws/codes in plain text. The names of the 225 Vietnamese laws/codes were crawled from the website[1]. The court cases were gathered using a Python-based engine from the website[2], which archives many legal documents from various courts and tribunal repositories. The database includes case laws from many domains, such as criminal law, civil law, and marriage and family law. Each crawled case is divided into metadata and case document parts.

4.3.2 Information Extraction

Information extraction is performed to extract information on entities and relations from a case law document. All the entities are extracted to form useful information about

[1] https://thuvienphapluat.vn
[2] https://congbobanan.toaan.gov.vn

the context of the case. Table 4.2 presents the list of entities in the case law document.

Table 4.2: The entity types and their atributes for data annotation

Entity	**Attributes**	**Description**
Case	case_id	id of the case
	case_number	number of the case (e.g 577/2022/HC-PT)
	document_type	type of the document (Verdict or Decision)
	case_level	level of the court (Trial, Appellate, and Cassation/Reopening)
	case_content	basic information of the case
	case_text	full content of the case
	date	documented and relevant dates
	court_id	id of the court
	domain_id	id of the case's domain
Domain	domain_id	id of the domain
	domain_name	type of the case (e.g Criminal, Civil, etc)
	subdomain	crimes, legal relations in the domain
Court	court_id	id of the court
	court_name	name of the court (e.g Hanoi Supreme People's Court)
	court_level	level of the court (e.g Provincial People's Court)
Law	law_id	id of the law
	law_name	name of the law (e.g Criminal Code, Civil Code)

Entities such as court, domain, and case are extracted from both meta-data and case document parts using regular expressions. However, the meta-data path lacks information on the laws/codes involved in the case. To retrieve these laws/codes, we first extracted sentences containing information of laws/codes using regular expression. These sentences may contain noise and redundant information. To address this, we proceed to match these sentences with our database of 225 Vietnam laws/codes. An illustration of the Law entity extraction step is presented in Table 4.3.

Table 4.3: Law entity extraction in a case law document.

Extracted sentence	Laws/Codes corpus
Điều 19 của Luật Hôn nhân và Gia đình *(Article 19 of the Marriage and Family Law)*	Luật Thi hành án dân sự sửa đổi 2014 *(Amended Civil Judgment Enforcement Law 2014)*
điều 81, 82 và 83 của Luật Hôn nhân và Gia đình *(Articles 81, 82, and 83 of the Marriage and Family Law)*	Luật thi hành án dân sự 2008 (Civil Judgment Enforcement Law 2008)
khoản 1 Điều 51, các điều 56, 81, 82 và 83 của Luật Hôn nhân và Gia đình *(Clause 1 of Article 51, Articles 56, 81, 82, and 83 of the Marriage and Family Law)*	Luật tổ chức Tòa án nhân dân 2014 *(People's Court Organization Law 2014)*
điều 28, 35, 39, 147, 227, 228 và 273 của Bộ luật Tố tụng dân sự *(Articles 28, 35, 39, 147, 227, 228, and 273 of the Civil Procedure Code)*	Luật thi hành án hình sự 2010 *(Criminal Judgment Enforcement Law 2010)*
điều 6, 7, 7a và 9 của Luật Thi hành án dân sự *(Articles 6, 7, 7a, and 9 of the Civil Judgment Enforcement Law)*	Bộ luật Tố tụng dân sự 2004 *(Civil Procedure Code 2004)*
Điều 30 của Luật Thi hành án dân sự *(Article 30 of the Civil Judgment Enforcement Law)*	Luật Hôn nhân và gia đình 2014 *(Marriage and Family Law 2014)*

4.3.3 Knowledge Graph Deployment

Table 4.4 shows the statistics of the legal knowledge graph. It has a total of 10,181 nodes, of which 9078 are case nodes. The total number of edges is 54,110 edges, 35,954 of that are between case nodes and law nodes, while both the relations of (case, court) and (case, domain) pairs have 9,078 edges. This is due to the one-to-one link among case nodes, domain nodes and court nodes. The density D of the graph is 0.001, and the ratio R of the number of edges per node is 5.314. These scores are calculated by the Formula 4.1 and Formula 4.2, respectively.

$$D = \frac{|E|}{|V| \times (|V| - 1)} \tag{4.1}$$

$$R = \frac{|E|}{|V|} \tag{4.2}$$

where $|E|$ is the number of edge and $|V|$ is the number of vertex in the graph.

One interesting insight from the graph is its connectivity. Although the graph has 60 connected components, only one of them is significant; all others have only one node. A connected component containing only one node usually indicates content errors or failures in extracting relationships. The significant component comprises 10,122 nodes, with 9,078 nodes representing cases and 176 nodes representing laws. Further analysis reveals that there are approximately 4 'based-on' relations per case node, meaning that for each case, 4 laws or codes of law are referenced. This connected component indicates a strong relation among all case nodes via different meta-paths.

Table 4.4: The statistics of the knowledge graph

Property	**Quantity**
Case node	9,078
Court node	693
Domain node	185
Law node	225
Total	10,181
Case-law edge	35,954
Case-domain edge	9,078
Judgement-court edge	9,078
Total	54,110
Connected components	60

4.4 Statutory – Case Law Retrieval Model

Along with the development of technology, the volume of digital documents has significantly increased, especially in the legal field. This advancement has made it easier to search for and access legal information more efficiently. Legal documents are often lengthy, structured, and presented in a specific writing style. Effectively harnessing this data largely depends on how it is organized and standardized. In the legal domain, particularly in case law documents, one can find information about the cases, court decisions, and laws related to those cases. Although the information is available, retrieving legal information can be complex, especially when dealing with specific case law or investigating a particular case law as a legal expert. The desired information may need to be searched for from various sources and approached in different ways.

We use BM25 to retrieve most relevant laws in the database. BM25 [120] is a powerful lexical engine used for ranking a collection of documents based on the frequency of query terms in each document. Given a input query case c_q, containing tokens $\{t_1, t_2, \ldots, t_n\}$, the BM25 [120] score of a statutory law s in corpus S is computed as follow:

$$score(c_q, s) = \sum_{i=1}^{n} IDF(t_i) \times \frac{f(t_i, s) \cdot (k_1 + 1)}{f(t_i, s) + k_1 \cdot (1 - b + b \cdot \frac{|s|}{avgdl})}$$

where $f(t_i, s)$ is term frequency of t_i in the article s, $|s|$ is the length of the statutory law s in terms, and $avgdl$ is the average article length in the database, $k1$ and b are free parameters.

There are two approaches to aggregating relevant law sets of different cases: union and intersection. Given two law sets A and B, the union set an intersection set of A and B are defined as follow:

$$A \cup B = \{x : x \in A \text{ or } x \in B\}$$

$$A \cap B = \{x : x \in A \text{ and } x \in B\}$$

if $A \cap B = \emptyset$ then the system would return an empty list.

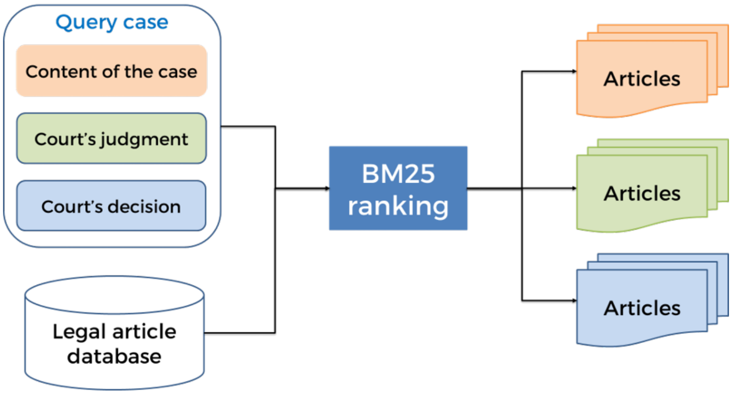


Figure 4.3: The case-law matching method

We implement four methods based on the BM25 search engine for the task of determining relevant laws as follows:

1. The first method is called *case-law matching* as shown in Figure 4.3, which means each part of a case (Content of the case, Court's judgment, Court's decision) is directly fed into the BM25 engine to retrieve top-k relevant law.

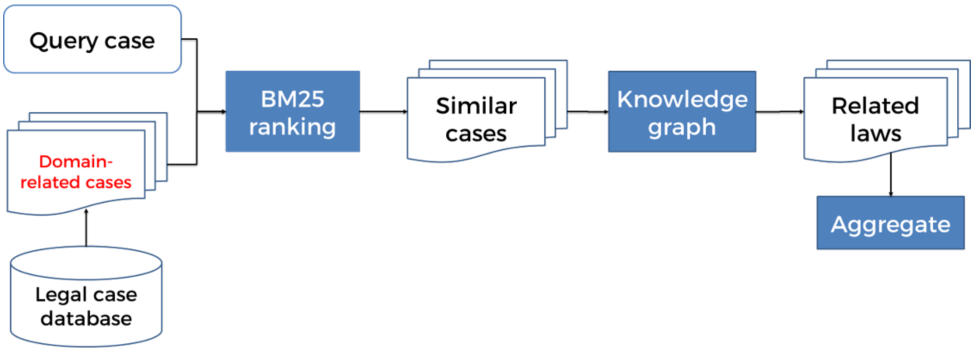


Figure 4.4: The Domain case-case matching and KG method

2. The second method is *improved case-law matching*. We combine the results from the first method (run 1, 2, and 3 in Table 4.5) using union and intersection aggregate functions.

3. The third method is *case-case matching and KG*. First, we employ the BM25 lexical matching model to select the top-k cases relevant to the query case. Subsequently, we utilize the knowledge graph to identify statute law (vertices) linked to these $top - k$ relevant cases (vertices), assuming that these statute law will be relevant to the input query case. With $k > 1$, the relevant statute laws are aggregated using union or intersection operations to produce the final relevant statute laws for the query case.

4. The last method is *Domain case-case matching and KG* as shown in Figure 4.4. Instead of querying similar cases in the whole dataset as in method 3, search space is reduced via the meta-path Case-Domain-Case in the knowledge graph. This means that cases from different legal domains will be excluded from the case-case matching search space. Subsequently, we proceed to perform similar tasks as in method 3.

4.5 Experiments and Results

In this work, a baseline model based on the BM25 engine is applied to evaluate the KG application in the law retrieval task. Particularly, we performed 11 runs of 4 methods on the test set of 500 case laws. The heterogeneous graph does not contain these test cases. Table 4.5 presents the details of experimental results.

The runs of methods 1 and 2 limited results. One of the reasons is the length of

Table 4.5: The results of the relevant law retrieval

#	Description	F1	Recall	Precision
Method 1 - Case-law matching				
1	Content of the case	0.061	0.037	0.18
2	Court's judgment	0.154	0.093	0.093
3	Court's decision	0.231	0.139	**0.676**
Method 2 - Improved case-law matching				
4	Mix 3 queries (Union)	0.288	0.347	0.245
5	Mix 3 queries (Intersection)	0.029	0.015	0.321
Method 3 - Case-case matching and KG				
6	Top-1 similar case	0.449	0.429	0.472
7	Top-2 similar cases (Union)	0.471	0.554	0.409
8	Top-2 similar cases (Intersection)	0.386	0.281	0.616
Method 4 - Domain case-case matching and KG				
9	Top-1 similar case	0.47	0.442	0.503
10	Top-2 similar cases (Union)	**0.503**	**0.583**	0.441
11	Top-2 similar cases (Intersection)	0.411	0.303	0.642

both the query case and the law, and furthermore, the vocabulary correlation between them is not substantial. For the third run, the content of the court's decision part is input into the BM25 engine to retrieve relevant laws/codes. This part refers to a lot of laws/codes information to support decisions. As a result, the third run achieves the highest precision, with a score of 0.676.

Run 4-5 combines results from 3 runs using Method 1, either by union or interaction. For example, if the matching results of 'Content of the case,' 'court's judgment,' and 'court's decision' are $[id_1, id_{10}]$, $[id_{10}]$, $[id_{10}, id_{25}]$ respectively, the union result is $[id_1, id_{10}, id_{25}]$, while the intersection result is $[id_1]$. This significantly improves the recall metric of Run 4 but restricts precision. Conversely, the recall metric of Run 5 is severely limited.

The methods using knowledge graphs achieve outstanding results. After matching case-case and utilizing KG, selecting the top-1 result yields a recall measure of 0.429 and precision of 0.472 (run 6). Opting for the union of top-2 results maximizes recall, resulting in an approximately 2$\%$ increase in F1 score compared to using only the top-1 result. Conversely, the intersection of top-2 results reduces the recall due to returning fewer relevant results, thereby decreasing f1-score.

In the run 10, similar cases are extracted via the meta-path Case-Domain-Case in the knowledge graph, which achieves the highest F1 score of 0.503 and a Recall of

0.583. Compared to methods 1 and 2, the utilization of legal knowledge graphs results in a remarkable increase of 21% in the F1 score. Subsequently, the combination of domain-specific information from the legal knowledge graph also contributes to reducing the search space and increasing accuracy, thereby improving the model's performance.

Furthermore, experiments show that there is a trade-off between F1, Recall, and Precision scores when using Union and Intersection aggregation approaches. Run 10 returns related laws that support at least one similar case, resulting in a higher recall. Meanwhile, Run 11 only retains laws that support all the similar cases, leading to higher precision.

Table 4.6: Some output examples of Run 10 and Run 11

Run 10	**Run 11**
Luật Hôn nhân và gia đình 2014 (Marriage and Family Law 2014)	**Luật Hôn nhân và gia đình 2014 (Marriage and Family Law 2014)**
Bộ luật tố tụng dân sự 2015 (Civil Procedure Code 2015)	Luật thi hành án dân sự 2008 (Civil Judgment Enforcement Law 2008)
Luật phí và lệ phí 2015 (Fees and Charges Law 2015)	Bộ luật tố tụng hình sự 2015 (Criminal Procedure Code 2015)
Luật Thi hành án dân sự sửa đổi 2014 (Amended Civil Judgment Enforcement Law 2014)	
Luật Bảo hiểm xã hội 2014 (Law on Social Insurance 2014)	
Luật thi hành án dân sự 2008 (Civil Judgment Enforcement Law 2008)	
Bộ luật tố tụng hình sự 2015 (Criminal Procedure Code 2015)	

For error analysis, Table 4.6 presents output examples from Run 10 and Run 11 (matching using KG). Run 10 returns 7 laws/codes, in which the first 2 laws/codes are correct. Although the result contains all related laws in the given case law, its precision score is low due to an excessive number of retrieved cases. Compared to Run 10, Run 11 only returns 3 laws/codes, in which one of them is correct. Therefore, this run achieves higher precision and lower recall scores.

4.6 Summary

This study presents a unique approach to create a heterogeneous knowledge graph for legal documents and relevant laws, helping to improve the organization and representation of legal data. Our technique encompasses the data acquisition, information extraction, and knowledge graph construction. Regarding the information extraction, we have successfully identified entities and connections within the unstructured legal texts to populate a diverse graph. Beside helping to enhance the performance of the statutory – case law retrieval task, this method also facilitates a wide range of other applications in the legal field, including case analysis, legal guidance, and decision-making support. The baseline model, using unsupervised learning techniques and the knowledge graph, showed promising outcomes in recognizing pertinent laws for a specific case law. The future research can concentrate on refining information extraction, incorporating advanced graph-based learning approaches, and broadening the knowledge graph's range for enhanced performance and wider utility.

Chapter 5

Article Reference Network for IR-based Legal Question Answering

The increasing complexity of statute law has led to a growing demand for efficient and effective retrieval methods. This chapter presents a novel approach to statute law retrieval that utilizes reference networks to uncover connections between laws. By presenting laws as a network of references, our method allows users to quickly identify relevant laws and navigate the intricate web of legal documents. The key point is that the reference network can encode both internal and external legal relations, helping to integrate both the local relevancy and the long-range dependencies into the final retrieval model. We evaluate the performance of our approach using a large corpus of statute law documents and demonstrate that it outperforms existing retrieval methods. Our approach can contribute to the development of AI-assisted legal research tools, making it easier for legal practitioners to find relevant laws and precedents. Furthermore, by uncovering hidden connections between laws, our method can assist in identifying inconsistencies and gaps in the legal system, ultimately improving its effectiveness and reliability.

Additionally, this chapter synthesizes models that represent relationships within the legal domain to address the problem of Vietnamese legal document question-answering.

This work in Section 5.6 was published in the **JSAI-isAI 2022. Lecture Notes in Computer Science. Springer** [VTHY4]. It was also applied to build solution of my team in AQLAC competition 2022-2023 and publiced paper in **KSE 2022, KSE 2023** conferences (indexed by Scopus) [VTHY5,VTHY6].

5.1 The Article Reference Relation Network

Numerous countries and regions, including France, the United Kingdom, Japan, and Vietnam, adhere to the statute law system. In this system, written laws are promulgated, and issued by governments or legislatures. The statute law provides clear, precise, and authoritative rules for governing a specific subject or matter. Each law prescribes regulations and rules for a different subject, such as civil code, criminal code, marriage, and family law. Legal documents are characterized by substantial length and a stringent organizational structure, typically partitioned into various hierarchical levels such as parts, chapters, sections, articles, and clauses with the article level being the predominant and widely employed tier.

Internal reference: Within the context of legal documents, successive articles in a chapter frequently exhibit a proximate relationship in terms of content or through direct references using co-referential terms such as “it”, “the preceding articles”, “then”, and analogous linguistic constructs as shown in the Table 5.1, we named it as *internal reference*. Articles in Table 5.1 are located in Chapter VII: Prescription, Part I: General Provisions in the Japanese Civil Code. Article 162 establishes the principle of acquisitive prescription for ownership of property, depending on certain circumstances. Meanwhile, Article 163 broadens the notion of acquisitive prescription, encompassing property rights beyond mere ownership, similar to Article 162. Articles 164 and 165 consider the cases of discontinuation of possession to support preceding articles.

External reference: Additionally, it is common for articles to make references to antecedent articles within the same or even different legal documents, as delineated in Table 5.2, we named it as *external reference*. The articles in Table 5.2 are located in different Sections of Chapter II in Part III Claims of the Japanese Civil Code. Particularly, Article 551 establishes a Donor’s Obligation to Deliver regarding gift transactions. However, the provisions of Article 551 are applicable, with necessary modifications, to a loan for consumption without a special agreement as prescribed in clause (1) of Article 590, and loans for use as shown in Article 596.

In practice, legal documents encompass a substantial volume of reference rela-

Table 5.1: An example of internal reference in the Japanese Civil Code

第百六十二条 Article 162	（所有権の取得時効） 二十年間、所有の意思をもって、平穏に、かつ、公然と他人の物を占有した者は、その所有権を取得する。 ２十年間、所有の意思をもって、平穏に、かつ、公然と他人の物を占有した者は、その占有の開始の時に、善意であり、かつ、過失がなかったときは、その所有権を取得する。 (Acquisitive Prescription of Ownership) (1) A person that possesses the property of another for 20 years peacefully and openly with the intention to own it acquires ownership thereof. (2) A person that possesses the property of another for 10 years peacefully and openly with an intention to own it acquires ownership thereof if the person was acting in good faith and was not negligent at the time when the possession started.
第百六十三条 Article 163	（所有権以外の財産権の取得時効） 所有権以外の財産権を、自己のためにする意思をもって、平穏に、かつ、公然と行使する者は、前条の区別に従い二十年又は十年を経過した後、その権利を取得する。 (Acquisitive Prescription of Property Rights Other Than Ownership) A person that exercises a property right other than ownership peacefully and openly with the intention to do so on the person's own behalf acquires that right after the passage of 20 years or 10 years, according to the distinction provided for in the **preceding Article**.
第百六十四条 Article 164	（占有の中止等による取得時効の中断） 第百六十二条の規定による時効は、占有者が任意にその占有を中止し、又は他人によってその占有を奪われたときは、中断する。 (Renewal of Acquisitive Prescription Due to Discontinuation of Possession) The prescription under the provisions of Article 162 is renewed if the possessor discontinues the possession voluntarily or is deprived of that possession by another person.
第百六十五条 Article 165	前条の規定は、第百六十三条の場合について準用する。 The provisions of the **preceding Article** apply mutatis mutandis to the case under Article 163.

Table 5.2: An example of external reference in the Japanese Civil Code

第五百五十一条 Article 551	（贈与者の引渡義務等） 贈与者は、贈与の目的である物又は権利を、贈与の目的として特定した時の状態で引き渡し、又は移転することを約したものと推定する。 ２負担付贈与については、贈与者は、その負担の限度において、売主と同じく担保の責任を負う。 (Donor's Obligation to Deliver) (1) The donor is presumed to have promised to deliver or transfer the thing or right that is the subject matter of the gift, while maintaining its condition as of the time when it is specified as the subject matter of the gift. (2) With respect to gifts with burden, the donor provides the same warranty as that of a seller, to the extent of that burden.
第五百九十条 Article 590	（貸主の引渡義務等） 第五百五十一条の規定は、前条第一項の特約のない消費貸借について準用する。 ２前条第一項の特約の有無にかかわらず、貸主から引き渡された物が種類又は品質に関して契約の内容に適合しないものであるときは、借主は、その物の価額を返還することができる。 (Lender's Obligation to Deliver) (1) The provisions of **Article 551** apply mutatis mutandis to a loan for consumption without a special agreement referred to in paragraph (1) of the preceding Article. (2) Irrespective of whether there is any special agreement referred to in paragraph (1) of the preceding Article, if the thing delivered from the lender does not conform to the terms of the contract with respect to the kind or quality, the borrower may return the value of the delivered thing.
第五百九十六条 Article 596	（貸主の引渡義務等） 第五百五十一条の規定は、使用貸借について準用する。 (Lender's Obligation to Deliver) The provisions of **Article 551** apply mutatis mutandis to loans for use.

tions, and disregarding these relations results in a significant loss of information. In this study, we propose the construction of a knowledge graph that could capture reference relations within legal documents. The legal reference relation graph was constructed based on a heterogeneous graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as shown in Figure 5.1. The nodes in the graph are legal articles $\mathcal{V} = \{a_1, a_2, ..., a_N\}$. There are a total of two types of relations of entities, including the internal reference edges from a_i are $\mathcal{E}_{a_i}^{IN} = \left\{ in_{a_{i-1}}^{a_i} | a_i, a_{i-1} \in \mathcal{V} : InSameChapter(a_i, a_{i-1}) = 1 \right\}$ and the external reference edges from a_i are $\mathcal{E}_{a_i}^{EX} = \{ex_1^{a_i}, ex_2^{a_i}, \ldots, ex_{n_i}^{a_i}\}$.

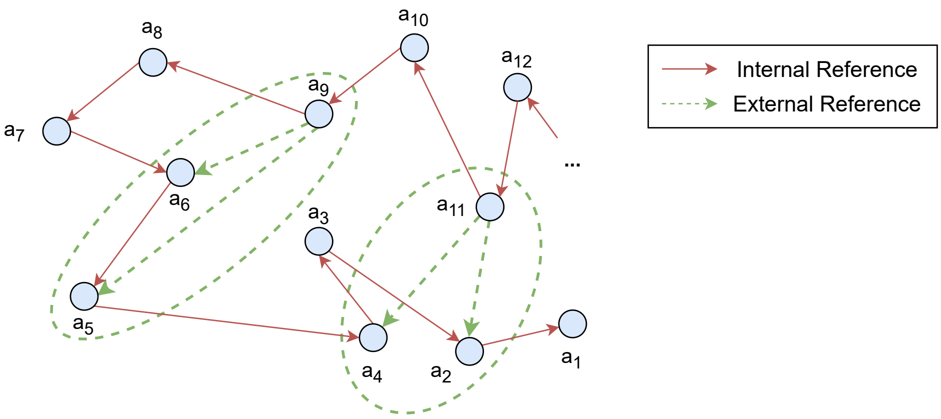


Figure 5.1: Illustration of reference relations between articles

Table 5.3: The statistics of references in the corpus

Attributes	**Values**
The number of chapters	10
The number of articles	768
The number of articles that have external reference	135
The total number of external reference	229
The maximum external references of an article	6
The average number of external references	0.2982

First, we segment the statute law into hierarchical levels of chapters and articles. Subsequently, a sliding window is employed to determine the quantity of the internal references. Finally, we utilize regular expressions to extract external references. In this study, we construct the legal reference relation graph based on the Japanese civil code, and a breakdown of the graph components is enumerated in Table 5.3. There are 10 chapters with 768 articles in the Japanese Civil Code. In the corpus, 135 articles refer directly to others, resulting in a total of 229 external references. On average, each article has approximately 0.2982 external references, which indicates a strong relationship between articles. Notably, the highest number of external references found in an article is 6. Up to 20% of the legal articles exhibit external reference relations, indicating the prevalence of this relational aspect within the dataset. This successful exploitation of these reference relations enhances features and representational capabilities of model.

5.2 Reference Network for IR-based Legal Question Answering

5.2.1 Introduction

In this chapter, we introduce a novel law retrieval approach that utilizes the concept of reference networks to enhance the retrieval process. Our method capitalizes on the observation that legal statutes are not isolated entities; rather, they function within a network of references, with laws often citing other laws. By treating laws as nodes within a reference network, we can explore the direct and indirect connections between statutes, thereby enabling more effective identification of relevant laws.

We propose an architecture that incorporates information from cited laws to enrich the representation of a given law, thus capturing both the content and the context of the references. This approach represents a significant departure from traditional document retrieval techniques that typically rely on content similarity alone. By considering the reference network, our system is better equipped to understand the legal context and relevance of documents, enabling it to yield more accurate retrieval results.

Recent studies related to law retrieval have employed various neural network techniques, such as CNNs, LSTMs, attention mechanisms, and graph neural networks, to achieve remarkable results in the legal domain [62, 96, 138, 145, 157]. These works examine legal document structures and propose novel representation methods based on their characteristics. Some of these studies also show the benefits of combining semantic vectors and lexical features to enhance the overall performance. However, these approaches mainly focus on content-based similarity and may not fully capture the complex web of references within legal documents. In contrast, our approach aims to tackle this issue by harnessing the power of reference networks, especially making the most of both the internal (i.e., local) relevancy and the external (i.e., long-range) dependencies to enhance the final retrieval model.

Through the comprehensive evaluations of a large corpus of statute law documents, we demonstrate that our method outperforms existing retrieval methods in terms of relevance and efficiency. Additionally, we discuss the potential contributions of our model to the development of AI-assisted legal research tools, which can streamline the legal discovery process.

We delineate the methodology employed for leveraging correlations among legal articles to construct a data representation aimed at enhancing the outcomes of the legal retrieval task. The comprehensive structure of the model is depicted in Figure 5.2. First, we present a comprehensive overview of the legal article retrieval problem. Following this, we introduce various symbols, knowledge graph structures, and the methodology involved in their construction process. Ultimately, we elucidate the architecture and training process of a model that integrates a graph representation of legal relations with pre-trained language models.

Legal article retrieval is one of the most traditional and common in the field of legal text processing. Let A be a corpus (i.e., a database) of statutory law articles. Given a question q about any legal issues that can be covered by the corpus A, the system aims to retrieve a subset $A^r \subset A$ that every article $a_i^r \in A^r$ semantically related or support to a given query q (legal question or statement). The problem can be described as follows:

$$\mathit{Relevance}(q, a_i^r) = \begin{cases} 1 & \text{if } a_i^r \text{ is semantically related to } q \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

$$A^r = \{a_i^r \in A : \mathit{Relevance}(q, a_i^r) = 1\} \tag{5.2}$$

5.2.2 Reference Network Model

To assess the relevance between a question and a legal provision in addition to the content of the current provision, we also examine its references. The two types of reference relations mentioned in section 5.1 are extremely crucial and indispensable when analyzing the semantics of any legal article. Legal articles cited under either of these reference relations contribute significantly and constitute a comprehensive meaning for the legal article under consideration. To leverage these two reference relations, this study proposes an architecture based on a knowledge graph of reference relations and pre-trained language models. The ultimate goal is to integrate the semantics of the legal article under consideration with the legal articles cited within that article. The model comprises three primary steps: representing the knowledge graph and text data layers, propagation layers, and prediction layer.

Figure 5.2 illustrates an overview of our proposed architecture. The input of the entire structure comprises the query q, the legal article under consideration a_i, a set of internal reference articles, and a set of external reference articles. In which, the set of articles belonging to the internal reference relation of the main article is determined based on a sliding window with M preceding articles $\mathcal{D}_i^{IN} = \{a_{i-m_i}, \ldots, a_{i-1} | m_i < M\}$. For example, in Figure 5.1, if we set the window size to 4, but “Article 2” is positioned as the second article within the chapter, its internal references are limited to only one (which is smaller than the window size), “Article 1”. The set of legal articles belonging to the external reference relation of main article $\mathcal{D}_i^{EX} = \{a_j^i, \ldots, a_{n_i}^i\}$, in Figure 5.1, “Article 11” will have two external references, which are “Article 2” and “Article 4”.

Presentation layers: In addition to assessing the semantic correlation between the input query and the current legal article, we also examine the correlation between the query and referenced legal articles. Pretrained language models, with their advantages in text semantic representation, are employed to encode the input data [38, 101]. The representation phase can be depicted using Equation 5.3. r_{a_i} represents the semantics for the pair (q, a_i); R_{in} comprises vectors representing the semantics for the pair (q, a_{in}) where $d_{IN} \in D_i^{IN}$; R_{ex} consists of vectors representing the semantics for the pair (q, a_{ex}) where $a_{ex} \in D_i^{EX}$.

$$\begin{aligned} R_{in} &= [\mathit{encode}(q \oplus a_{i-m}), \ldots, \mathit{encode}(q \oplus a_{i-1})] \\ r_{a_i} &= \mathit{encode}(q \oplus a_i) \\ R_{ex} &= [\mathit{encode}(q \oplus a_j), \ldots, \mathit{encode}(q \oplus a_l)] \end{aligned} \tag{5.3}$$

Where $r_{a_i} \in \mathcal{R}^d$, $R_{in} \in \mathcal{R}^{m \times d}$, $R_{ex} \in \mathcal{R}^{n \times d}$. The d represents the dimensionality of the encoding representation, the m denotes the size of the sliding window, determines the internal references, and the n denotes the number of external references, $\oplus$ is concatenation function.

Embedding propagation layers: After the semantic encoding process in the internal and external reference relation, we obtain a sequence of encoded vectors representing the pairs between the question and the legal article. Multi LSTM layers [49] are utilized for modeling contextual information within these sequences of vectors. The concatenation

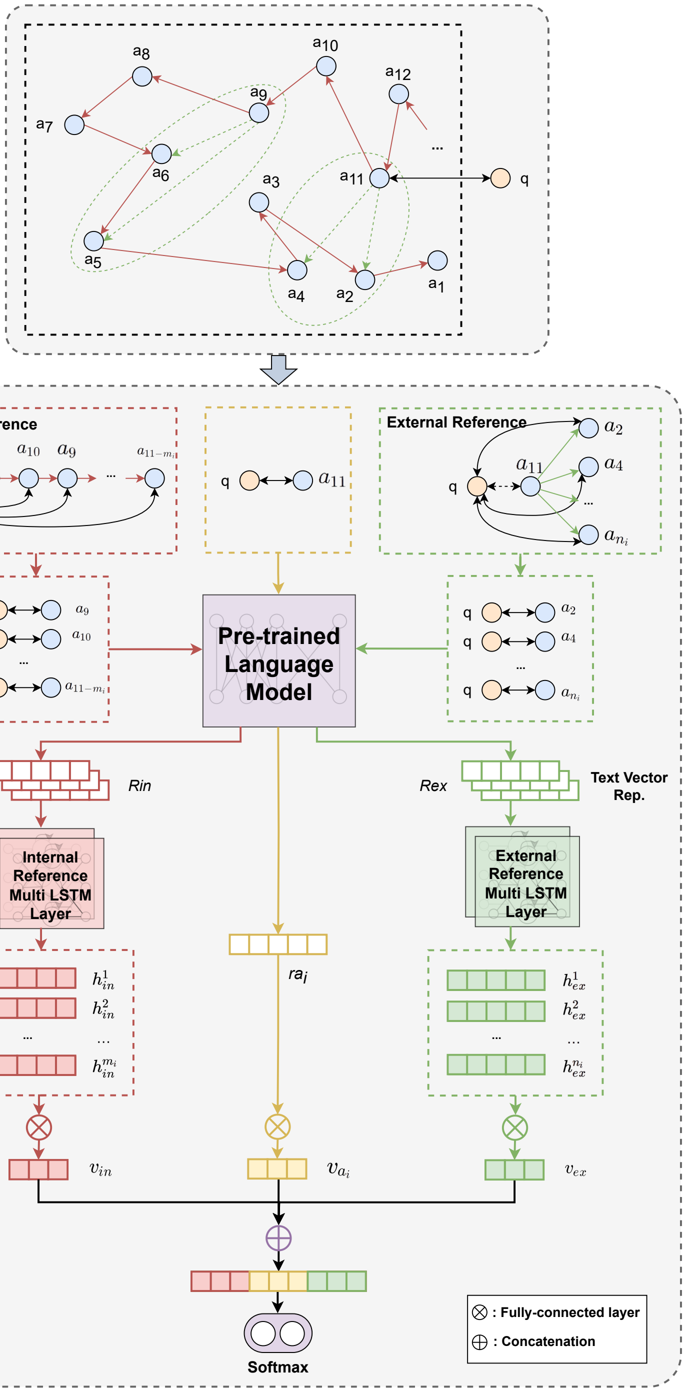


Figure 5.2: The reference network architecture to identify the relation of the query q and article a_i

of forward and backward LSTM represents the contextual embedding vector in state t, as expressed in Equation 5.4.

$$\begin{aligned} h^t_{in} &= [\overleftarrow{h^t_{in}}, \overrightarrow{h^t_{in}}] \\ h^t_{ex} &= [\overleftarrow{h^t_{ex}}, \overrightarrow{h^t_{ex}}] \end{aligned} \tag{5.4}$$

Where $\overleftarrow{h^t} = \overleftarrow{LSTM}(r_t, \overleftarrow{h^{t-1}})$, $\overrightarrow{h^t} = \overrightarrow{LSTM}(r_t, \overrightarrow{h^{t-1}})$, and $h_t \in \mathcal{R}^d$. The last state of the internal and external multi-LSTM layers will be utilized as representative vectors for the internal and external references.

To enhance representational capacity, vectors of internal reference, external references, and current articles are passed through feed-forward networks. Consequently, this process of feature projection can be articulated as follows:

$$\begin{aligned} v_{a_i} &= r_{a_i} * W_{a_i} \\ v_{in} &= h^m_{in} * W_{in} \\ v_{ex} &= h^n_{ex} * W_{ex} \end{aligned} \tag{5.5}$$

Where $v_{a_i}, v_{in}, v_{ex} \in \mathcal{R}^d$ is the final representation vector of the current article, internal references and external references with d is the hidden dimension. W_{a_i}, W_{a_i}, W_{ex} are the weight matrices.

Prediction layer: finally, the final embedding vector of the current article is the concatenation of all those output vectors. The synthesized semantic vectors passed through a final feed-forward network layer to calculate the correlation score for the root pair $(q - a_i)$. This phase is formulated by the equation 5.6.

$$\mathit{score}(q, a_i) = \mathit{softmax}((v_{a_i} \oplus v_{in} \oplus v_{ex}) * W_s) \tag{5.6}$$

Where $\mathit{score}(q, a_i)$ is a vector containing the probabilities of two labels: relevant and irrelevant. W_s are the weight matric. $\oplus$ is concatenation function.

Model learning: cross-entropy loss is utilized for this legal articles retrieval task. Due to the data imbalance between relevant and irrelevant labels, weights for each label are added to the loss function. The loss function of the query q and legal article a_i is presented by Equation 5.7.

$$\mathcal{L}_{(q,a_i)} = -\sum_{c=0}^{1} w_c y_{i,c} \log(\mathit{score}(q, a_i)_c) \tag{5.7}$$

where $\mathit{score}(q, a_i) \in \mathcal{R}^2$ is the vector containing the probabilities of each label for the root pair $(q - a_i)$, w_c is the weight of label c, and $y_i \in \mathcal{R}^2$ is the one-hot vector of the label. In the training phase, we use SGD optimizer [86] for tuning all parameters with a learning rate of $1e^{-4}$.

5.2.3 Trail-threshold Ranking

The weighted ensemble can be effectively employed to aggregate the outputs' scores of different retrieval models to enhance the overall performance. Each model would be assigned a specific weight that reflects its importance. The grid search method is applied to the validation set to find the optimized weight set for the ensemble equation as in Equation 5.8.

$$\mathit{relevance\_score} = \sum_{i=1}^{2} w_i * \mathit{score}_i \tag{5.8}$$

$$s.t : w_i \in [0, 1], \sum_{i=1}^{2} w_i = 1$$

where w_i represents the weight of model i, and s_i represents the relevance score computed by the model i. Before aggregating scores from lexical and *reference network* models, we applied min-max normalization to these scores to preserve the relative order while reducing the variance.

The trail-threshold ranking strategy determines retrieved articles in the reference phase. Candidate articles whose relevance score is greater than a threshold T are selected. The threshold T value is tuned on the development set.

5.2.4 Experiments and Results

Evaluation Metric

For this task, evaluation measures are precision, recall and F2-measure. All the metrics are macro-average (evaluation measure is calculated for each query and their average is used as the final evaluation measure) instead of micro-average (evaluation measure is calculated using results of all queries). The definition of these measures is as follows:

$$\text{Precision} = \text{average of}\frac{\text{\# correctly retrieved articles for each query}}{\text{\# retrieved articles for each query}} \tag{5.9}$$

$$\text{Recall} = \text{average of}\frac{\text{\# correctly retrieved articles for each query}}{\text{\# relevant articles for each query}} \tag{5.10}$$

$$\text{F-measure} = \text{average of}\frac{5 \times \text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}} \tag{5.11}$$

Unlike the traditional F1 score, the F2 score places greater emphasis on recall. In retrieval tasks, returning as many relevant items as possible is crucial, even at the expense of precision.

Datasets

To conduct experiments, we employed datasets from COLIEE 2021 and 2022, which is an annual workshop in the field of legal text processing. The Competition on Legal Information Extraction and Entailment (COLIEE) aims to develop a worldwide research community and establish state-of-the-art methods for information retrieval and entailment using legal texts. COLIEE has gained great interest from both researchers and legal experts from more than 25 different countries.

For the Statute Law Retrieval task, legal questions related to Japanese civil law are selected from Japanese Bar exams. Every year, the competition has the same legal article corpus, which contains 768 articles with official English translations. Most articles consist of approximately 50 to 200 tokens, as shown in Figure 5.3. Table 5.4 presents an overview of the statistics of the legal corpus. The length of legal articles can vary significantly, ranging from a minimum of 5.0 tokens to a maximum of 867.0 tokens for English and from 10.0 tokens to 866.0 tokens for Japanese. Therefore, the great number of articles with varying lengths may indeed pose challenges during the retrieval process.

Table 5.4: The statistics of article length in COLIEE statute law retrieval task

	Min	**Max**	**Mean**	**25%**	**50%**	**75%**
Japanese	10.0	886.0	109.6	57.0	87.0	133.0
English	5.0	867.0	100.2	47.0	78.0	124.0

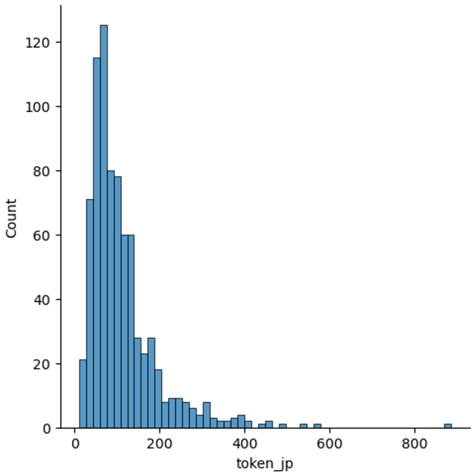


Figure 5.3: The length distribution of the legal articles in the corpus

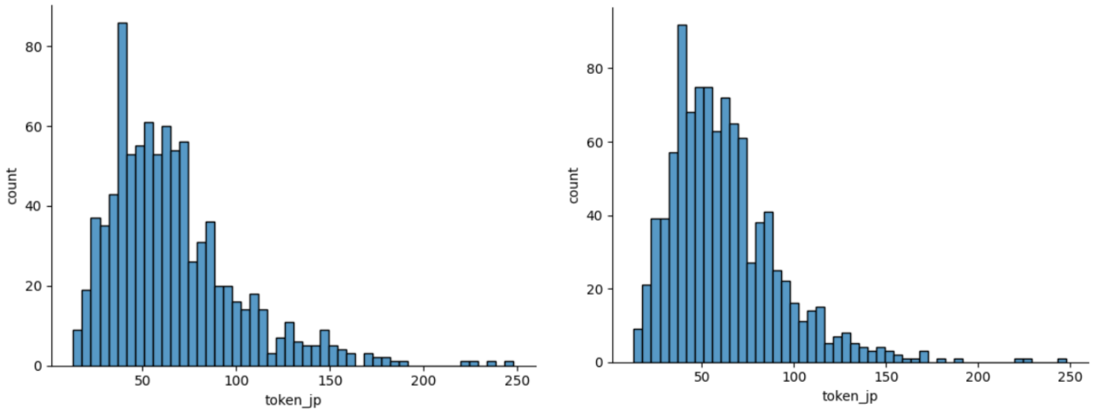
In COLIEE 2021, there are 806 questions in the training set, while the testing set contains 81 questions. In the subsequent year, COLIEE 2022, the training set was formed by combining the training and test sets from the previous year. A new testing set contains 109 legal questions selected from the Japanese Bar exams. These data sets are relatively small and limited for the deep learning approach, especially language models. With limited data, models may not comprehend the legal language and logical reasoning. This can pose challenges and cause low performance during the model learning process.

When analyzing the impact of external reference relations, our focus is on the quantity of samples where relevant articles have external references. Additionally, there is a slight difference in the number of external references for each dataset. As shown in Table 5.5, there are approximately 11-19% of total queries in datasets whose relevant articles have external references. This number shows the potential of our proposed methods in enhancing the features and representational capabilities of the model.

Further analysis on Table 5.5 reveals some challenges of the legal article retrieval task. Firstly, the number of relevant articles per query can vary significantly from 1 to 6 articles, which may pose difficulties in determining the number of retrieved articles during the inference phase. There is also an imbalance phenomenon in datasets where the number of relevant articles is small compared to the total number of articles in the corpus. Legal queries have a large margin distribution, ranging from 13 to 248 tokens in the Japanese version. Figure 5.4 presents length frequency histograms of legal queries in the datasets. We can observe that most queries have approximately 45 to 100 tokens in length.

Table 5.5: The statistics of references in the COLIEE statute law retrieval task

Attribute	**2021**		**2022**	
	Train	**Test**	**Train**	**Test**
Number of queries	806	81	887	109
Number of queries with external reference	162	15	177	12
Maximum number of relevant articles	6	4	6	5
Average number of relevant articles	1.29	1.25	1.27	1.20
Minimum length of a query	13	16	13	22
Maximum length of a query	248	163	248	139
Average length of a query	65.26	71.18	62.14	62.80



(a) COLIEE 2021 dataset.

(b) COLIEE 2022 dataset.

Figure 5.4: The query length distribution of the datasets

Experimental Setup

The experimental procedures were executed utilizing a GPU 2080 Ti. In consideration of the imperative to engage with datasets encompassing diverse languages, the pre-trained Multilingual-BERT[1] was enlisted for the subject models' parameters initialization. Furthermore, given the retrieval nature of the task at hand, an additional evaluation involved the application of a re-ranker model mono-T5[2] based on the T5 architecture and fine-tuned on the MS–MARCO dataset [6].

In the training process, a query and an article are paired and passed through the re-ranking model. The re-ranking model then learns to discern the semantic correlation between the content of the query and the article. To strike a balance between training

[1] https://huggingface.co/bert-base-multilingual-cased

[2] https://huggingface.co/castorini/monot5-base-msmarco

time and model capability, the top-k articles most relevant to the query, evaluated using the BM25 model, are selected for pairing with the given query. Due to the relatively conditional accuracy of the BM25 model, it is plausible that the top–k articles retrieved by this model may insufficiently contain truly relevant articles stated on the gold label. Consequently, to ensure the adequacy of positive labels in the training set, any missing relevant articles are supplemented. To select the most suitable top-k for the dataset, recall scores have been computed for the BM25 model on the training dataset as shown in Table 5.6.

Table 5.6: The recall scores of the BM25 model corresponding to each top-k value on the training set of the COLIEE 2021 dataset

Top-k	**Recall Score**
30	0.7447
50	0.7801
100	0.8322
200	0.8765
500	0.9400

Data generation for the training phase necessitates minimizing data imbalance while still preserving a close approximation of the data distribution to ensure model performance. Hence, a top-k value of 30 was chosen to generate the training dataset, comprising query-article pairs. Conversely, during the testing and inference phases, retrieval results demand precision and comprehensiveness. Therefore, a top-k value of 500 was employed to maximize recall while leveraging the semantic correlation computing capabilities of the re-ranking model.

We employ the Adams optimizer [67] uniformly. As expounded in section 5.2.3, to balance precision and recall scores, during the validation and testing phases, the relevant scores of the trained model were subjected to a weighted ensemble with the relevant scores derived from the BM25 model. For the selection of final relevant articles, the ranking trail-threshold strategy is used.

Experimental Results

With the settings described in the preceding section, the proposed models were experimented on the COLIEE 2021 and 2022 datasets. The evaluation metrics include F2 score, recall, and precision. Recall score is calculated by the number of correctly retrieved articles divided by the number of relevant articles. The precision score is the

Table 5.7: The results of the reference network models on the COLIEE 2021 dataset

Model	**Japanese**			**English**		
	Precision	**Recall**	**F2**	**Precision**	**Recall**	**F2**
Single article						
mBERT-single	0.6276	0.7839	0.7123	0.6122	0.7839	0.7047
monoT5-single	-	-	-	0.6777	0.7808	0.7296
Reference network						
mBERT-ws2	0.6739	0.7654	0.7264	0.6926	0.7840	0.7444
mBERT-ws3	**0.7099**	0.7840	0.7509	0.6736	0.7716	0.7291
mBERT-ws4	0.7016	**0.8086**	**0.7648**	0.6757	0.7963	0.7424
monoT5-ws4	-	-	-	0.6755	0.7963	0.7371

division of the number of correctly retrieved articles to the total number of retrieved articles. Finally, the F2 score is computed by combining precision and recall scores, with a higher weight for recall.

In this work, we also experiment with the *single article* approach to provide a baseline for validation. This approach relies solely on the information contained within the query and the specific article being considered without incorporating additional information from reference relations. Furthermore, to analyze the impact of the internal reference relation on the retrieval results, different window sizes (W-Size column) are considered with values of 2, 3, and 4, as shown in Table 5.7 and Table 5.8.

In the *single-article* approach, two different pre-trained models are utilized: mBERT and monoT5-based. As presented in Table 5.7 and Table 5.8, the accuracy of the Japanese raw dataset achieves better results compared to the English dataset. Particularly, the F2 scores of the mBERT-single model improve by 1% in both datasets. This might be because the English dataset is a translated version of the original Japanese raw dataset, potentially leading to a loss of information during the translation process. For the dataset translated into English, it could be observed that the monoT5 model yields better results compared to the BERT model. Specifically, the F2 scores of monoT5-single on the English dataset in COLIEE 2021 and 2022 seem to have a rise of 1% to 2%, respectively. This strong performance of monoT5 may be because monoT5 is fine-tuned on the MS MARCO dataset [6] specifically for the reranking task.

For the *reference network* approach, models employ the same hyperparameters setting as in the *single-article* method. Notably, we observe significant improvements in F2 scores of reference network models in both COLIEE 2021 and 2022 datasets. Particularly, the F2 scores of mBERT models have improvements of 4% from 0.7047 to

Table 5.8: The results of the reference network models on the COLIEE 2022 dataset

Model	Japanese			English		
	Precision	Recall	F2	Precision	Recall	F2
Single article						
mBERT-single	0.6666	0.8974	0.7863	0.6584	0.8787	0.7757
monoT5-single	-	-	-	0.6616	0.8029	0.7348
Reference network						
mBERT-ws2	0.7704	0.8534	0.8101	0.7151	0.8206	0.7724
mBERT-ws3	0.7552	**0.8974**	**0.8266**	0.6973	0.8671	0.7878
mBERT-ws4	0.7615	0.8420	0.8016	**0.7699**	0.8237	0.7952
monoT5-ws4	-	-	-	0.6735	0.7962	0.7409

0.7444 in COLIEE 2021 English set and 5% from 0.7123 to 0.7648 in COLIEE 2021 Japanese set. For the COLIEE 2022 dataset, the Reference network method enhances the performance of models by 2% to 3%. One notable point is that the improvements of the Reference network method in the COLIEE 2021 set are better than those in the COLIEE 2022 set. A reasonable explanation is that the number of queries with external reference in the COLIEE 2021 test set is 19%, which is greater than 11% of the COLIEE 2022 test set. Indeed, our experiments show the contribution of reference relations to the features and representational capabilities of the model. These relations support the root article and provide a wider and more comprehensive legal context during the retrieving process. Furthermore, the results of the reference network models also exhibit higher F2 scores on the Japanese raw dataset compared to the dataset translated into English.

We construct experiments to validate the effects of increasing window size for reference network models on both datasets. The results of different window sizes are presented in Table 5.7 and Table 5.8. Particularly, the performances on the COLIEE 2021 Japanese set and COLIEE 2022 English set greatly improved when the window size was increased. Indeed, the F2 scores of mBERT rise by 5% and 3%, respectively. However, there are cases when increasing the window size does not yield noticeable changes in the F2 score. For example, in COLIEE 2021 English and COLIEE 2022 Japanese sets, the best scores are obtained with window sizes of 2 or 3. In these cases, the Reference network method achieves F2 scores around 0.74 and 0.82 for each dataset, respectively. This phenomenon may be because most relevant information often focuses on a range of 2 to 3 adjacent articles. Consequently, adding more internal reference relations may introduce more redundant information into the retrieving process. During experimentation with various window sizes, we observed that a window size of 2-4 achieves optimal performance. Increasing the window size further not only fails to improve the model accuracy but also demands additional computational resources.

To elucidate the individual impact of internal and external references, we conducted additional experiments for each component. The results in Table 5.9 illustrate that both internal and external reference information influence the model performance on the COLIEE 2021 Japanese set. Particularly, when considering only external reference relations, the F2 score shows a 1.5% improvement compared to mBERT-single model, increasing from 0.7123 to 0.7243. Similarly, internal reference relations also greatly contribute to the model's performance. In the case of only internal reference relations, there is an enhancement of approximately 6.2% in terms of F2 compared to mBERT-single model. Ultimately, incorporating both types of relations, internal and external, leads to the best F2 score with a score of 0.7648.

Table 5.9: The results of the different settings on COLIEE 2021 Japanese dataset

Model	**Precision**	**Recall**	**F2**
mBERT-single	0.6276	0.7839	0.7123
Only external reference			
mBERT-only-external	0.6142	0.7963	0.7243
Only internal reference			
mBERT-only-internal-ws2	0.6492	0.7963	0.7370
mBERT-only-internal-ws3	0.6901	0.7901	0.7478
mBERT-only-internal-ws4	**0.7370**	0.7778	0.7566
Internal + External reference			
mBERT-ws2	0.6739	0.7654	0.7264
mBERT-ws3	0.7099	0.7840	0.7509
mBERT-ws4	0.7016	**0.8086**	**0.7648**

Table 5.10 presents the best evaluation results of participants in COLIEE 2021. OvGU team [152] achieves the best F2 score among all runs with a F2 score of 0.73. They combined BERT contextual embedding with TF-IDF representations and data augmentation to achieve the best performance. JNLP team [94] is the runner-up with an F2 score of 0.723, but has the highest Recall score of 0.802. Their approach is based on techniques including text chunking using a sliding window, self-labeling to help mitigate noisy training samples, and model ensembling to enhance the performance of language models. The third-place team, UA [65], employed a probabilistic model, BM25, to finish with an F2 score of 0.709 in the competition. Compared to other teams in the COLIEE 2021, the proposed method improves the F2 score by 4.8% from 0.730 to 0.764. The result of monoT5-en-ws4 model is slightly better than the OvGU's with approximately

the same Precision score, but a little better performance on the Recall metric.

Table 5.10: The results of the participating teams in COLIEE 2021

	Precision	**Recall**	**F2-Score**
Other teams			
OvGU	0.675	0.778	0.730
JNLP	0.600	0.802	0.723
UA	**0.753**	0.704	0.709
LLNTU	0.666	0.617	0.705
TR	0.333	0.617	0.523
HUKB	0.290	0.698	0.522
Proposed method			
monoT5-en-ws4	0.675	0.796	0.737
mBERT-jp-ws4	0.701	**0.808**	**0.764**

Table 5.11: The results of the participating teams in COLIEE 2022

	Precision	**Recall**	**F2-Score**
Other teams			
HUKB	**0.8180**	0.8405	0.8204
OVGU	0.7781	0.8054	0.7790
JNLP	0.6865	0.8378	0.7699
UA	0.8073	0.7641	0.7638
LLNTU	0.6743	0.6391	0.6416
Proposed method			
monoT5-en-ws4	0.6735	0.7962	0.7409
mBERT-jp-ws3	0.7552	**0.8974**	**0.8266**

The results of participating teams and the proposed method in the COLIEE 2022 dataset are shown in Table 5.11. This year, most teams chose to ensemble scores from different approaches to enhance the overall performance [64]. The HUKB team [161] proposed new retrieval systems based on the similarity between questions and legal articles. They also utilized an ordinal BM25 system and contextual embedding models along with new proposed systems to finish at the first rank of the Legal Statute Retrieval task. The runner-up team, OvGU [153], extracted external knowledge from textbooks and incorporated it into the retrieval pipeline based on TF-IDF and sentence embedding. JNLP team [19] had a different approach when proposing a deep learning system with use-case identification. By categorizing the given legal query, they constructed specific retrieval models for tackling each type of query. With their proposed method, the JNLP team achieved the third rank in the competition with an F2 score of 0.7699. With two different pre-trained models, mBERT and monoT5, the *reference network* method slightly enhances the retrieval performance compared to other teams. However, the notable point is that mBERT-jp-ws3 achieved the highest Recall score of 0.8974 while maintaining a relatively good Precision result.

To provide more concise and clear contributions of reference articles, we provide Table 5.12, an example of a query supported by reference articles. The query has id R02-1-A and is extracted from the test set of COLIEE 2021. The query R02-1-A has two relevant articles: Article 398-11 and Article 376. We can observe that the provision of Article 398-11 does not match well with the content of the legal query in both lexical and semantic terms. Indeed, reference articles like Article 376 and 377 contain information support for the root Article 398-11. Ordinary retrieval pipelines based on probabilistic models or deep learning may not address this circumstance when these methods consider articles to be dependent units and have no relation to others. On the other hand, the proposed method can tackle this problem effectively with the contributions of the "Article Reference Relation Network", which considers legal articles inside a context of reference relations.

5.3 Vietnamese Legal Question Answering

5.3.1 General Architecture

Our proposed end-to-end article retrieval-based question-answering system architecture is demonstrated in Figure 5.5. The system comprises three primary phases: preprocessing, training, and inference phase, which work together to provide accurate and efficient responses to user queries.

The Preprocessing Phase

A database consisting of individual articles is generated by processing the original Vietnamese civil law documents. The resulting article-level database enables easy access and retrieval of specific information contained within the documents.

- ***Vietnamese Civil law*** is a corpus of Vietnamese legal documents.
- ***Parser*** segment legal documents into list of articles.
- ***Cleaning*** will filter out documents with metadata. Special symbol characters are also removed from the article. Numbers and vocabulary are retained and converted to lowercase.

Table 5.12: An example of a query and its supporting articles via reference links

Query: R02-1-A	債務者Ａが債権者Ｂのために自己の所有する不動産に根抵当権を設定した場合に関する次のアからオまでの各記述のうち，正しいものを組み合わせたものは，後記１から５までのうちどれか。Ｂは，元本の確定前は，Ａに対する他の債権者Ｃに対してその順位を譲渡することができる。 An obligee (B) may assign the order of priority of a mortgage to another obligee(C) of obligator (A) before the principal is crystallized.
第三百九十八条の十一 Article 398-11	（根抵当権の処分） 元本の確定前においては、根抵当権者は、第三百七十六条第一項の規定による根抵当権の処分をすることができない。ただし、その根抵当権を他の債権の担保とすることを妨げない。 ２ 第三百七十七条第二項の規定は、前項ただし書の場合において元本の確定前にした弁済については、適用しない (1) Before the principal is crystallized, a revolving mortgagee may not dispose of a revolving mortgage under the provisions of Article 376, paragraph (1); provided, however, that the revolving mortgagee is not precluded from applying that revolving mortgage to secure other claims. (2) The provisions of Article 377, paragraph (2) do not apply to payments made before the principal is crystallized in the cases provided for in the proviso to the preceding paragraph.
第三百七十六条 Article 376	（抵当権の処分） 抵当権者は、その抵当権を他の債権の担保とし、又は同一の債務者に対する他の債権者の利益のためにその抵当権若しくはその順位を譲渡し、若しくは放棄することができる。 ２ 前項の場合において、抵当権者が数人のためにその抵当権の処分をしたときは、その処分の利益を受ける者の権利の順位は、抵当権の登記にした付記の前後による。 (Disposition of Mortgages) (1) A mortgagee may apply the mortgage to secure other claims, or assign or waive that mortgage, or its order of priority, for the benefit of other obligees of the same obligor. (2) In the cases referred to in the preceding paragraph, if a mortgagee disposes of the mortgage for the benefit of two or more persons, the order of priority of the rights of persons who benefit from that disposition follows the chronological order of supplemental registration in the registration of the mortgage.
第三百七十七条 Article 377	（抵当権の処分の対抗要件） 前条の場合には、第四百六十七条の規定に従い、主たる債務者に抵当権の処分を通知し、又は主たる債務者がこれを承諾しなければ、これをもって主たる債務者、保証人、抵当権設定者及びこれらの者の承継人に対抗することができない。 ２ 主たる債務者が前項の規定により通知を受け、又は承諾をしたときは、抵当権の処分の利益を受ける者の承諾を得ないでした弁済は、その受益者に対抗することができない。 (Requirements for Perfection of Disposition of Mortgages) (1) In the cases in the preceding Article, the mortgagee may not duly assert the disposition of mortgages against principal obligors, guarantors, mortgagors or their respective successors unless the disposition is notified to the principal obligors or the principal obligors consent to that disposition in accordance with the provisions of Article 467. (2) If the principal obligors have received the notice or given the consent pursuant to the provisions of the preceding paragraph, payments made without the consent of the persons who benefit from the disposition of the mortgage may not be duly asserted against those beneficiaries.

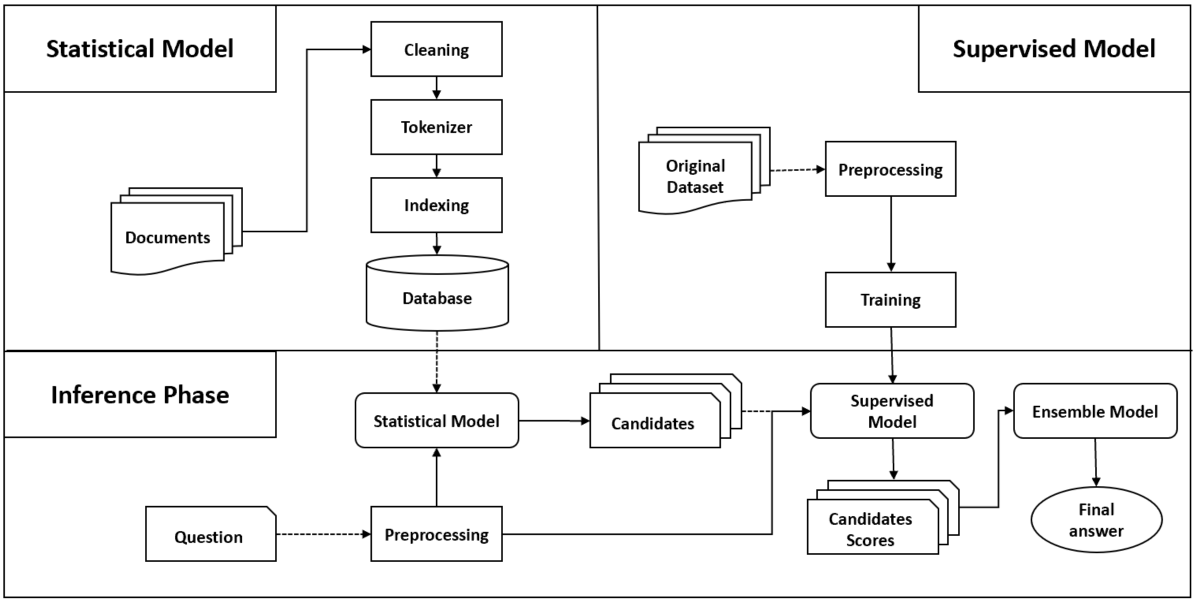


Figure 5.5: The pipeline of the end-to-end article retrieval-based QA system

- ***Tokenizer*** is crucial to the processing of Vietnamese natural language. Vietnamese word structure is quite complicated, a word might contain one or more tokens.
- ***Indexing*** is a task to represent and put articles into the database. Given a query, the search engine will return the response quickly and accurately.

The Training Phase

A supervised machine learning model is developed to rank the articles related to the input question. This model uses training data to learn patterns and relationships within the articles and applies this knowledge to provide accurate rankings of relevant articles.

- Labeled dataset is a legal question answering dataset.
- ***Preprocessing*** includes tasks similar to the preprocessing phase for question processing.
- ***Training***, we will construct a deep learning model to rank the texts related to the question.

Inference phase

Inference phase refers to the process of generating a response for a new input question. This phase typically involves applying a trained machine learning model to the input question and selecting the most appropriate response from a set of potential answers.

- ***Question*** is use's query in natural language.
- ***Preprocessing*** is same as previous phases to process input question.
- ***Quickview retrieval model*** matches questions and texts using unsupervised machine learning techniques . The processing speed of this model is typically fast.
- ***Candidates*** are a list of limited candidates returned from quickview retrieval model.
- ***Supervised model*** is result of the training phase. Its inputs are the question and the article candidates.
- ***Candicate scores*** are outputs of Supervised model.
- ***Ensemble model*** will combine the scores of the quickview retrieval model and the supervised model to make a final decision.

Indexing

During the indexing process, the words in the text will be analyzed, normalized, and assigned a corresponding index. When given a query, the system searches the index the most related. Word indexing helps to find and look up information in the text faster and more accurately.

The Quickview Retrieval Model

There are 117,575 legal articles in this corpus. This is a huge number, so in order to ensure the effectiveness of the QA system, we build a so-called Quickview Retrieval model using unsupervised machine learning techniques in order to rapidly return a limited candidate set.

Lexical matching to compare questions and articles in the word indexing database, we use the BM25 algorithm [120]. The bag-of-words retrieval function BM25 estimates the relevance of a document to a given search query by ranking documents according to the query terms that appear in each document.

Given a question Q, containing tokens $\{t_1, t_2, ..., t_n\}$, the BM25 score of a article A is:

$$BM25S(Q, A) = \sum_{i=1}^{n} IDF(t_i) \cdot \frac{f(t_i, A) \cdot (k_1 + 1)}{f(t_i, A) + k_1 \cdot (1 - b + b \cdot \frac{|A|}{avgdl})} \tag{5.12}$$

in which:

- $f(t_i, A)$: t_i's term frequency in the legal article A
- $|A|$: a number of word in in the legal article A in terms
- $avgdl$: the average article length in the legal corpus.
- k_1: a saturation curve parameter of term frequency.
- b: the importance of document length.
- $IDF(t_i)$ is the inverse document frequency weight of the given question t_i, follow as: $IDF(t_i) = \ln(1 + \frac{N - n(t_i + 0.5)}{n(t_i) + 0.5})$. N is amount of articles in the legal corpus, and $n(q_i)$ is amount of articles containing q_i.

The Supervised Model

Approach 1: We employ reference network as Section 5.1 and reference network models as Section 5.2.2 to construct a supervised learning model in this section.

Approach 2: Pre-trained language models have proven useful for natural language processing tasks. Particularly, BERT significantly enhanced common language representation [38]. We use the BERT pre-training model and adjust all its parameters to build the related classifier model. We use the first token's final hidden state h as the presentation for the question-article pair. The last layer is a single fully connected added on the top of BERT. The output of the model is a binary classification. Cross-entropy loss is applied to the loss function. Adam [66] is used to optimize all model parameters during the training phase with a learning rate of e^{-5}. The supervised score between the question and the legal article is the classification probability of label 1.

Lastly, we also use minmaxscaler to normalize scores and reranking a list of candidates. In this model, we proceed to build a related classification model based on two training datasets: the original dataset and a full dataset (original and weak label dataset, which contructed by method like Chapter 3). In the training process with the full dataset, we fit the model on weak label data first. Then use the best model to fine-tune with the original dataset.

The Ensemble Model

We utilize the quickview retrieval model to generate a list of the $top-k$ candidates. These candidates are then refined using a supervised ensemble model, which provides higher precision but is slower. The quickview model serves as a preliminary selection step due to its fast computation despite its lower precision.

We use a variety of measures of similarity, including lexical similarity (the quickview retrieval model) and semantic similarity (the supervised model). Despite the fact that lexical and semantic similarities are very different from one another, they can work in tandem and are complementary. The combined score of the question Q and the candidate article CA_i is calculated as follows:

$$CombineS(Q, CA_i) = \gamma * QS(Q, CA_i) + (1 - \gamma) * SS(Q, CA_i) \tag{5.13}$$

where $\gamma \in [0, 1]$.

The most relevant article $MRCA$ is returned by default, to determine a set of candidates to return, we would normalize the combined score and use the threshold parameter: a final returned articles set $FRA = \{CA_i | CombineS(Q, MRCA) - CombineS(Q, CA_i) < threshold\}$.

5.3.2 Reference Network Approach

To evaluate the effectiveness of the proposed model, we conducted experiments on the ALQAC 2021 dataset. The dataset that contains the reference relationships in Section 5.1 in the approach 1.

ALQAC 2021 Dataset Analysis

Automated Legal Question Answering Competition (ALQAC) is an annual contest in legal AI. In 2021, they introduced ALQAC dataset, which is a manually annotated dataset based on well-known statute laws in the Vietnamese Language, comprises 2,279 legal articles and 412 labeled questions.

Basic statistic such as the minimum, maximum, and average number of tokens in the training dataset are presented in Table 5.13, and a histogram depicting the distribution of token numbers in legal articles is shown in Figure 5.6. As indicated by Table 5.13, there is a considerable variance in the number of tokens across legal articles, ranging from a maximum of 1,606 tokens to a minimum of just 4 tokens. Such variability poses a substantial challenge for the model in consistently understanding the semantics of the documents.

	Legal Articles	**Question**
Max Tokens	1606	102
Min Tokens	4	7
Mean Tokens	197.4	30.2

Table 5.13: The statistics of the original dataset

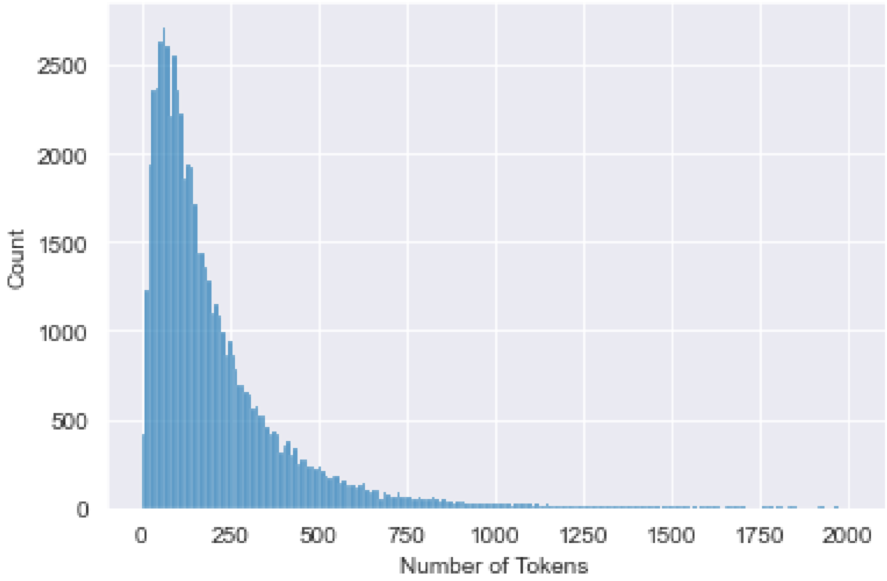


Figure 5.6: The distribution of the number of tokens in the legal articles

In this investigation, the foremost data preprocess procedure is word and sentence segmentation which is demonstrated to enhance the NLP model's efficiency in Vietnamese text. The segmentation based on RDRSegmenter [91] is executed using VnCoreNLP tools [149], which creates a segmented dataset.

The Quickview (BM25) Retrieval Results

As mentioned in section 5.3.2 that the number of legal articles is overlarge, the Okapi BM25 algorithm is exploited on the segmented dataset using rank-bm25 library [3] to rapidly generate top-k candidates for each query, which facilitates an efficient re-ranking process using more time-consuming and voluminous models. For finding the most appropriate k, a recall statistic in the training set with some predetermined k is executed, whose recall result is exhibited in the Table 5.14.

Top-k	**Recall Score**
100	1.0
50	0.9829
20	0.9773

Table 5.14: The recall score of BM25 on training set

For optimizing the computing resource and training time for the re-ranking phase, the top 50 BM25 candidates are chosen instead of the top 100 because it reduces the number of candidates by half (from 100 to 50 candidates) but only hurts the recall score by 0.0171 (from 1.0 down to 0.9829). After determining k, the top 50 most relevant legal articles with each query on the training set are retrieved by Okapi BM25 accompanied by all relevant ones. Finally, a new training set is created where each query has about 50 candidate articles.

The End-to-end Results

The results of the four above methods evaluated on ALQAC 2021's test dataset which include the F2 score on the test set. Table 5.15 showcases F2-scores from various teams or approaches in a ALQAC 2021 competition.

Team	**F2-score**
AnimeLaw	0.8061
Aleph	0.8807
Kodiac	0.7955
Single Approach	0.8355
Reference Approach	**0.8878**

Table 5.15: The result comparison with other teams

[3] https://pypi.org/project/rank-bm25/

Aleph team leads with the highest score of 0.8807 in this competition, who utilizes the pre-trained VnLawBERT [30] as a cross-encoder with the maximum sequence's length is 512 tokens. Meanwhile, pre-trained PhoBERT which is exploited in this investigation only has the maximum input length is 256, which prevents the model from entirely observing longer articles and so missing the correct candidates. Single Approach moderate success with scores of 0.8355 while the Reference Approach follows closely with 0.8645, suggesting its effectiveness possibly due to innovative methods or leveraging additional information.

The Error Analysis

To enhance the clarity and precision of the contributions from referenced articles, we present Table 5.16 as an illustration of a query underpinned by reference articles. This particular query, identified as "11/2017/QH14 article 8".

It is noted that the alignment of "11/2017/QH14 article 8". with the legal query's content is not strong, both lexically and semantically. "11/2017/QH14 article 6" offer supportive information for "11/2017/QH14 article 8" throught the internal reference. Traditional retrieval systems utilizing probabilistic models or deep learning techniques might not effectively navigate this scenario, as these approaches often treat articles as isolated units without considering inter-article relationships. Conversely, our proposed approach addresses this challenge through the "Article Reference Relation Network".

5.3.3 Supporting Relation for Automatic Data Enrichment Approach

In approach 2, we use the Vietnamese civil law question answering dataset to implement data augmentation methods based on supporting relationships as in chapter 3.

The Vietnamese civil law question answering dataset

Original dataset: the corpus is collected from Vietnamese civil law. The labelled dataset was introduced by Nguyen et al. [96]. Table 5.17 & 5.18 give a statistical summary of the corpus and dataset. There are 8,587 documents in the corpus. Vietnamese civil law documents have a long and intricate structure. The longest document contains up to 689 articles, and the average number of articles per document is also comparatively high at

Query: Khi thực hiện trợ giúp pháp lý cho các khách hàng thuộc đối tượng được hưởng trợ giúp pháp lý, luật sư được nhận tiền hoặc lợi ích khác nếu có thỏa thuận với khách hàng. *When providing legal assistance to clients eligible for legal aid, lawyers are entitled to receive money or other benefits if there is an agreement with the client.*
Gold: “law_id”: “11/2017/QH14”, “article_id”: 8
Single approach prediction: “law_id”: “11/2017/QH14”, “article_id”: 6 Trợ giúp pháp lý là việc cung cấp dịch vụ pháp lý miễn phí cho người được trợ giúp pháp lý trong vụ việc trợ giúp pháp lý theo quy định của Luật này, góp phần bảo đảm quyền con người, quyền công dân trong tiếp cận công lý và bình đẳng trước pháp luật. *Legal aid is the provision of free legal services to individuals eligible for legal aid in cases as regulated by this Law, contributing to the assurance of human rights and citizens’ rights in accessing justice and equality before the law.*
Reference network prediction: “law_id”: “11/2017/QH14”, “article_id”: 8 Quyền của người được trợ giúp pháp lý: 1. Được trợ giúp pháp lý mà không phải trả tiền, lợi ích vật chất hoặc lợi ích khác. 2. Tự mình hoặc thông qua người thân thích, cơ quan, người có thẩm quyền tiến hành tố tụng hoặc cơ quan, tổ chức, cá nhân khác yêu cầu trợ giúp pháp lý. 3. Được thông tin về quyền được trợ giúp pháp lý, trình tự, thủ tục trợ giúp pháp lý khi đến tổ chức thực hiện trợ giúp pháp lý và các cơ quan nhà nước có liên quan... *Rights of the legal aid recipient:* *1. To receive legal aid without the need to pay money, material benefits, or other benefits.* *2. To request legal aid personally or through relatives, legal proceedings authorities, or other agencies, organizations, or individuals.* *3. To be informed about the right to legal aid, and the procedures and processes for obtaining legal aid upon visiting legal aid organizations and relevant state agencies...*

Bảng 5.16: A sample of a query in Vietname question answering

13.69. The average title length in this dataset is 13.28 words, whereas the average content length is 281.83 words.

This is also worth noting because one of the challenges and restrictions is the presentation of long texts. On average, the questions are less than 40 words long. Because of the similarity in their distributions, it is expected that the model trained on the training set will yield good performance on the test set.

Weak labelled dataset: Chapter 3 have the assumption that the sentences in a legal article will support a topic sentence. On the basis of this supposition, the weak labelled dataset is created. There is also a similar relationship in this dataset. The title serves as a brief summary of the article, so the sentences in the article content support to title.

Bảng 5.17: The statistics of the Vietnamese legal document corpus

Attribute	**Value**
Number of legal documents	8,587
Number of legal articles	117,557
Number of articles missing title	1,895
The average number of articles per document	13.69
Maximum number of articles per document	689
The average length of article title	13.28
The average length of article content	281.83

Bảng 5.18: The statistics of the original dataset

	Train set	**Test set**
Number of samples	5329	593
Minimum length of question	4	5
Maximum length of question	45	43
Average length of question	17.33	17.10
Minimum number of articles per query	1	1
Maximum number of articles per query	11	9
Average number of articles per query	1.58	1.60

We apply this assumption to our method. By considering the title to be the same as the question, we will produce a dataset with weak labels. A title and content pair would be a positive example equivalent to a question and related articles pair. We randomly generated negative examples at a ratio of 1:4 to positive labels and obtained a weak label dataset consisting of 551,225 examples.

To ensure fairness in the training process and selection of hyperparameters, we divided the training dataset into training and validation with a ratio of 9:1.

The Experimental Setup

The processing phase and the quickview retrieval model are carried out on CPU Intel core i5 10500 and 32Gb ram. The supervised model is trained and inference on NVIDIA Tesla P100 GPU 15Gb. In the indexing step and the quickview retrieval model, we use Elasticsearch[4] with the configuration setting 8Gb heap size. Besides, during the experiment with some pre-trained BERT models, the BERT multilingual model produces the best results, so it is used to generate vector representation for the given question and the articles in the dense vector indexing and is used in a supervised model.

[4] https://www.elastic.co/

Top-k	**20**	**50**	**100**	**200**	**500**	**1000**
Time per Q (ms)	11.60	14.43	20,32	31.32	63.21	115,63
Training set						
article title only	0.4898	0.5674	0.6169	0.6644	0.7172	0.7536
article body only	0.4966	0.6169	0.6941	0.7586	0.8220	0.8659
article title and article body	0.5676	0.6739	0.7478	0.8060	0.8651	0.8998
boosting article title by 1.2	0.5930	0.6913	0.7598	0.8168	0.8719	0.9046
boosting article title by 1.5	**0.5979**	**0.6942**	**0.7612**	**0.8169**	**0.8740**	**0.9063**
boosting article title by 2.0	0.5492	0.6506	0.7193	0.7850	0.8557	0.8959
Testing set						
article title only	0.5079	0.5743	0.6282	0.6792	0.7259	0.7611
article body only	0.5171	0.6309	0.7103	0.7709	0.8368	0.8747
article title and article body	0.5802	0.6943	0.7728	0.8261	0.8798	0.9080
boosting article title by 1.2	0.6172	0.7208	0.7972	0.8442	0.8848	0.9124
boosting article title by 1.5	**0.6420**	**0.7214**	**0.7973**	**0.8453**	**0.8863**	**0.9128**
boosting article title by 2.0	0.5785	0.6830	0.7486	0.8106	0.8767	0.8979

Bảng 5.19: The top-k recall scores of the BM25 lexical matching method (i.e., Quickview) in different retrieval settings

The Quickview (BM25) Retrieval Results

In order to evaluate the impact of each component of the law article of the lexical matching method outcome, we carried out experiments using various combinations used of the article title, and article body. As shown in Table 5.19, several noteworthy aspects can be observed.

Initially, it can be observed that by using solely the article title, we still get a very considerate score compared to using only the article body, this reinforces our observation that the article title can serve as a brief summary of the article body and it alone can carry out most of the ideas presented in the article body. Secondly, using combinations of the law article components could boost the score significantly, to nearly 0.90 of Recall@1000. Additionally, it demonstrates that although the article title significantly contributed to the retrieval recall score, the document title was found to be relatively ineffective. Eventually, we decided to carry on my experiments with the combinations of article title and article body.

To determine an optimal boosting weight α for the article title, we performed experiments using various α values of range from 1.0 to 2.5 with a step of 0.1 in the ElasticSearch query. Some of the results obtained from the experiments are presented in Table 5.19, indicating that by boosting the article title weight by 1.5 achieve the

Bảng 5.20: The experimental results of the end-to-end QA system with top-k = 200

Model	R	P	F2
BM25 Model	0.4454	0.2399	0.3803
Supervised Model (original data)	0.6165	0.1461	0.3750
Supervised Model (full data)	0.6651	0.1998	0.4538
Ensemble Model (original data)	**0.6681**	0.4080	0.5925
Ensemble Model (full data)	0.6651	**0.4331**	**0.6007**

highest score of Recall@1000 on the train set of 0.9063 and on the test set of 0.9128. In addition, the test set achieved commendable scores of $Recall@k$ for the values of 100 and 200, which were 0.7973 and 0.8453, respectively. Overall, the word-matching technique has shown its undeniable strength in its simplicity while obtaining a high score and an efficient processing speed. As referring to table 5.19, this method can generate a list of 100 potential matches with an average processing time of 20.32 ms, while a list of 1000 candidates can be produced with an average processing time of 115.63 ms (with an approximate Recall score of 0.91).

The End-to-end Question Answering Results

Table 5.20 indicates the experimental results of the end-to-end question answering system result with a top 200 candidates from the quickview retrieval model. The word-matching model with BM25 and the supervised model built from the original data gives F2 score is about 0.38. The ensemble model outperforms the other models in F2 score with 0.6007, which is 22% higher than the single models. As was pointed out in the previous section, lexical and semantic similarity are highly dissimilar. But we believe they can cooperate and support one another. Results certainly support that. Table 5.20 also clearly illustrates the contribution of the weak label dataset. It improved the supervised machine learning model's F2 score by 8%. The weak label data continues to have an impact on the F2 score when the lexical and semantic matching models are combined. The ensemble model that used the weak label data had a 1% increase in F2 scores.

Additionally, there is a sizeable distinction between precision and recall. The recall is given more consideration because of its great impact on F2 score. We discovered that similarity in lexical and semantics has the same effect during the experimental and evaluation phases. Consequently, γ is set at 0.5. Infer time is also a remarkable point in the construction of the question-answering system, which shows the feasibility of the

Bảng 5.21: The results of the end-to-end QA system with ensemble model

Ensemble Model	**R**	**P**	**F2**	**Time(s)**
(full data, k=20)	0.5677	0.4034	0.5252	0.5
(full data, k=50)	0.5842	0.4428	0.5491	1
(full data, k=100)	0.6222	**0.4475**	0.5771	1.7
(full data, k=200)	0.6651	0.4331	**0.6007**	3.4
(full data, k=500)	**0.6793**	0.4015	0.5967	8.5
(full data, k=1000)	0.6583	0.4261	0.5936	17

Bảng 5.22: The result comparison with other research groups

Systems	**R**	**P**	**F2**
Attentive CNN [61]	0.4660	0.5919	0.4774
Paraformer [96]	0.4769	**0.5987**	0.4882
Our model (k=50)	0.5842	0.4428	0.5491
Our model (k=100)	0.6222	0.4475	0.5771
Our model (k=200)	**0.6651**	0.4331	**0.6007**

system when applied in practice.

Table 5.21 illustrate the results with the computational resources in the experimental environment, we can use the model with the top 50|100 candidates with an execution time of 1 second and 1.7 seconds per question. Their F2 scores are also only 2-5% lower than the best model.

Table 5.22 shows that our recall and F2 scores are incredibly high when compared to the Attentive CNN [61] and the Paraformer [96] models (0.6651 and 0.6007). Their models return small amounts of related articles, while our system is designed to return flexible amounts of articles with $threshold$. This explains why their precision is great, about 0.5987, whereas our precision is only 0.4331.

Table 5.23 describes an example of our legal question-answering system, compared with Paraformer [96]. A small number of related articles are frequently returned by Paraformer models. Our system is more flexible with 3 returned related articles. While the gold label number is 2. As an outcome, a paragraph model like Paraformer is produced that has great precision but low recall, whereas our method leans in the opposite direction. Since recall has a greater impact on F2 scores, our model has a significantly higher F2 score of 11%.

Our model predicts that “Article 466 from Doc 91/2015/QH13” is relevant to the given query but the gold label is 0. Considering this article, we believe the article is

Bảng 5.23: An output example of our system, compared with Paraformer.

Question: Vay tiền để kinh doanh nhưng không còn khả năng chi trả phải trả lãi suất thì như thế nào? *(In the case of insolvency, how does one address the issue of paying the interest on a business loan?)*	Ours	Para-former	Gold
Candidate 1: Id: Article 357 from Doc 91/2015/QH13 **Title:** Trách nhiệm do chậm thực hiện nghĩa vụ trả tiền *(Liability for late performance of the obligation to pay)* **Content:** 1. Trường hợp bên có nghĩa vụ chậm trả tiền thì bên đó phải trả lãi đối với số tiền chậm trả tương ứng với thời gian chậm trả. 2. Lãi suất phát sinh do chậm trả tiền được xác định theo thỏa thuận của các bên nhưng không được vượt quá mức lãi suất được quy định tại khoản 1 Điều 468; nếu không có thỏa thuận thì thực hiện theo quy định tại khoản 2 Điều 468. *(1. Where the obligor makes late payment, then it must pay interest on the unpaid amount corresponding to the late period.* *2. Interest arising from late payments shall be determined by agreement of the parties, but may not exceed the interest rate specified in paragraph 1 of Article 468 of this Code; if there no agreement mentioned above, the Clause 2 of Article 468 of this Code shall apply.)*	1	1	1
Candidate 2: Id: Article 466 from Doc 91/2015/QH13 **Title:** Nghĩa vụ trả nợ của bên vay *(Obligations of borrowers to repay loans)* **Content:** [...]5. Trường hợp vay có lãi mà khi đến hạn bên vay không trả hoặc trả không đầy đủ thì bên vay phải trả lãi như sau: a) Lãi trên nợ gốc theo lãi suất thỏa thuận trong hợp đồng tương ứng với thời hạn vay mà đến hạn chưa trả; trường hợp chậm trả thì còn phải trả lãi theo mức lãi suất quy định tại khoản 2 Điều 468 của Bộ luật này; b) Lãi trên nợ gốc quá hạn chưa trả bằng 150% lãi suất vay theo hợp đồng tương ứng với thời gian chậm trả, trừ trường hợp có thỏa thuận khác. *([...] 5. If a borrower fails to repay, in whole or in part, a loan with interest, the borrower must pay:* *a) Interest on the principal as agreed in proportion to the overdue loan term and interest at the rate prescribed in Clause 2 Article 468 in case of late payment;* *b) Overdue interest on the principal equals one hundred and fifty (150) per cent of the interest rate in proportion to the late payment period, unless otherwise agreed.)*	1	0	0
Candidate 3: Id: Article 468 from Doc 91/2015/QH13 **Title:** Lãi suất *(Interest rates)* **Content:** 1. Lãi suất vay do các bên thỏa thuận.[...] 2. Trường hợp các bên có thỏa thuận về việc trả lãi, nhưng không xác định rõ lãi suất và có tranh chấp về lãi suất thì lãi suất được xác định bằng 50% mức lãi suất giới hạn quy định tại khoản 1 Điều này tại thời điểm trả nợ. *(1. The rate of interest for a loan shall be as agreed by the parties.[...]* *2. Where parties agree that interest will be payable but fail to specify the interest rate, or where there is a dispute as to the interest rate, the interest rate for the duration of the loan shall equal 50% of the maximum interest prescribed in Clause 1 of this Article at the repayment time.)*	1	0	1

pertinent to the given question but it seems that the annotator's point of view is different. In addition, we discovered some similar cases in our error analysis.

Defining and agreeing on a measure of relevance is an important research question that needs the participation of the AI and Law community in its research. This not only benefits the development of automated methods but also makes legal judgments and decisions more reliable and accurate.

5.4 Summary

We have presented a general and hybrid reference network approach to the statutory case law retrieval task by making the most of both useful legal connections and the power of pre-trained language models. We again highlight our contributions and sum up important points that have been discussed throughout the chapter. First, we have proposed a novel approach to formulating and integrating useful structural information in legal texts into the final retrieval model. While the internal references help uncover the local and implicit relevance, the external references aim to capture explicit long-range legal dependencies. Our empirical study showed that the internal references are truly useful, and the retrieval performance tends to be even better when legal articles have more external references. Second, although the legal data are limited, both the legal texts and links have been embedded with powerful pre-trained language models, and therefore, the retrieval relevancy is boosted significantly. Third, the proposed method has been evaluated thoroughly on the COLIEE 2021 and 2022 datasets with various experimental settings. Our experimental results suggest that the embeddings of internal and external links in the legal reference network help enhancing the retrieval accuracy to a new level. This means that the internal and external references help add extra relevance that does not come from merely lexical or text-content embedding approaches. The results also show that our method performed better than all the competing methods for both the Japanese and English data collections. This demonstrates the efficiency, flexibility, and generality of our approach.

In this chapter, we also present a method to improve the performance for the task of legal question answering for Vietnamese using language models through weak labelling. By demonstrating the effectiveness of this method through experiments, we have verified the hypothesis that improving the quality and quantity of datasets is the right approach for this problem, especially in low-resource languages like Vietnamese. The results of our work can provide valuable insights and serve as a reference for future attempts to tackle similar challenges in low-resource legal question answering.

Conclusions

Summary of the Results and Contributions

The dissertation conducted a systematic and thorough study of the legal retrieval and question answering tasks, that are two of the most critical and challenging problems in legal NLP. According to the research challenges, motivations, and objectives addressed in Chapter 1, the disssertation have presented the problem statement, formulation, and proposed the use of various types of legal characteristics (i.e., features) as well as introduced several deep model architectures to integrate those features in order to enhance the performance of the three IR and QA tasks. All in all, the dissertation has the following important results and contributions:

- In order to leverage and make the most of the nature and characteristics of legal data to boost the performance of the three main IR and QA tasks addressed in this dissertation (i.e., the research objective - O2), we have introduced the supporting model (in Chapter 3) that helps to integrate the supporting relations at different levels of granularity (i.e., case-case, paragraph-paragraph, and decision-paragraph) for the case law retrieval problem. In addition to the legal textual features, structural or graph-based features are also really useful for the legal IR and QA tasks. We therefore defined and constructed a heterogeneous knowledge graph consisting of legal case documents and relevant legislative materials in order to improve the legal information organization and the statutory – case law retrieval task (in Chapter 4). The knowledge grahp links cases, courts, domains, and laws to enrich graph-based features and therefore help to improve the retrieving performance significantly. In Chapter 5, we proposed the use of a reference network to enhance the performance of the legal question answering problem. The reference network captures both the local citations and the long-range (global) dependencies among legal articles in order to uncover potential links that help to locate and retrieve truly relevant articles that cannot be found by traditional lexical matching, by using synonyms, or even by text embedding methods.

- In addition to the uncovering and utilizing legal characteristics, this dissertation attempted to introduce suitable deep learning based architectures for the IR and QA tasks (i.e., the research objectives O1 and O3). These model architectures help (i) leverage the existing resources, methods, and models (e.g., powerful pre-trained language models) and (ii) learn better representations and integration of legal textual and structural characteristics. These model improvements result in the further inhancement of the effificiency and performance of the tasks. Technicall, Chapter 3 proposed the SM-BERT-CR architecture, a supporting model for the case law retrieval task. In Chapter 4, the construction of the heterogeneous graph involves data collection, entity extraction, and graph construction using NLP techniques. Our approach demonstrates its potential in the statutory – case law retrieval task and other downstream tasks such as case analysis, legal recommendations, and decision support, providing valuable insights and resources for the legal domain. Chapter 5 proposed the reference network model to address the legal document question answering task. The local and global reference links in the network were embedded using powerful pre-trained models and then incorporated into the final question answering model to improve the efficiency.

- Moreover, the experimental results in this dissertation are competitive with the state-of-the-art results, in which some models perform better than the previous work. In Chapter 3, the supporting model, i.e., SM-BERT-CR, achieved F_1 scores of 0.6060 and 6528 for the case law retrieval phase on the COLIEE 2019 and 2020 datasets, respectively. These outcomes are only 2 percentage points less than the state-of-the-art results even we did not use training data for this phase. In the second phase (i.e., legal entailment), this model has achieved very high results with F_1 of 0.7253 and 0.6753 on the COLIEE 2019 and 2020 datasets, respectively. These results are significantly higher (around 6 percentage points) than the runner-up team. In the statutory – case law retrieval task (Chapter 4), our knowledge graph-based method attained an F_1 score of 0.503, much higher than the baseline ($F_1 = 0.288$) that did not utilize the knowledge graph. In Chapter 5, the reference network-based method gave significant results on both COLIEE 2019 and 2020 datasets with F_2 scores of 0.7648 and 0.8266. Additionally, we also built an end-to-end for the Vietnamese legal document QA task and achieved the highest results on several Vietnamese datasets.

- Besides the techincal contributions, the analysis and discussions throughout this dissertation would help provide a better understanding of legal texts and processing problems, present the advancements and remaining challenges of legal NLP in general and legal IR and QA in particular. This study would also suggest the future legal IR and

QA research directions, especially inspiring and stimulating further studies in legal NLP for low-resource languages like Vietnamese.

The Limitations of the Dissertation

Although the dissertationn has attempted to leverage various legal features and introduced different deep learning based architectures to enhance the efficiency and performance of the three IR and QA tasks, there are still several limitations and remaning issues that can be done better.

First, in the first phase of the legal case retrieval task, the SM-BERT-CR model has identified and retrieved supporting cases for a given query from the entire case law corpus based on both the textual proximity and legal relation. However, in legal domain, a real supporting case is called a "noticed case" which is assumed to be relevant to the query case by lawyers. This normally causes an inconsistency in the data. As result, the first phase of the task normally retrieves more relevant cases than needed. This is still an issue that can be improved more in further studies. Second, the information in the legal knowledge graph built in Chapter 4 has not been fully exploited. This is partly because the knowledge graph was defined and constructed to cover many other downstream tasks in legal NLP. Finally, the proposed models in this dissertation still require high power computing systems to train and inference due to both the complexity of the models as well as the use of pre-trained language models. This sill needs to be improved for practical applications.

The Future Direction

The future study will explore and improve the proposed method in a number of directions. First, continue to enhance methods for addressing problems related to the length and complexity of legal documents. Second, the efficiency of integrating the legal relations into the legal document IR and QA tasks suggests that we can extend our methods with larger and more sophisticated legal knowledge presentation, i.e., in terms of both scale and diversity. Expand research on logical representation in legal documents to improve accuracy for retrieval tasks in particular and legal NLP in general. Additionally, we can try larger pre-trained language models, especially models specialized for each particular language. Finally, developing solutions and models for legal IR and QA from various perspectives to serve various types of users including lawmakers, judges, plaintiffs, defendants, and non-expert users.

List of Publications

[VTHY1] **Yen Thi-Hai Vuong**, Quan Minh Bui, Ha-Thanh Nguyen, Thi-Thu-Trang Nguyen, Vu Tran, Xuan-Hieu Phan, Ken Satoh, and Le-Minh Nguyen. "SM-BERT-CR: a deep learning approach for case law retrieval with supporting model."*Artificial Intelligence and Law* 31, no. 3 (2023): 601-628. (SCIE, ISI Q1)

[VTHY2] **Thi-Hai-Yen Vuong**, Hai-Long Nguyen, Tan-Minh Nguyen, Hoang-Trung Nguyen, Thai-Binh Nguyen, and Ha-Thanh Nguyen. "NOWJ at COLIEE 2023: Multi-task and Ensemble Approaches in Legal Information Processing."*The Review of Socionetwork Strategies* (2024): 1-21. (ESCI, WoS)

[VTHY3] **Thi-Hai-Yen Vuong**, Hoang Minh-Quan, Tan-Minh Nguyen, Hoang-Trung Nguyen, and Ha-Thanh Nguyen. "Constructing a Knowledge Graph for Vietnamese Legal Cases with Heterogeneous Graphs."*In 2023 15th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1-6. IEEE, 2023. (Scopus)

[VTHY4] **Thi-Hai-Yen Vuong**, Ha-Thanh Nguyen, Quang-Huy Nguyen, Le-Minh Nguyen, Xuan-Hieu Phan. "Improving Vietnamese Legal Question-Answering System based on Automatic Data Enrichment". *In JSAI-isAI 2023. Lecture Notes in Computer Science*. Springer, Cham. (In press, Scopus)

[VTHY5] Hai-Long Nguyen, Thai-Binh Nguyen, Tan-Minh Nguyen, Ha-Thanh Nguyen, and **Hai-Yen Thi Vuong**. "Vlh team at alqac 2022: Retrieving legal document and extracting answer with bert-based model."*In 2022 14th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1-6. IEEE, 2022. (Scopus)

[VTHY6] Nguyen, Hai-Long, Dieu-Quynh Nguyen, Hoang-Trung Nguyen, Thu-Trang Pham, Huu-Dong Nguyen, Thach-Anh Nguyen, **Thi-Hai-Yen Vuong** and Ha-Thanh Nguyen. "NeCo@ ALQAC 2023: Legal Domain Knowledge Acquisition for Low-Resource Languages through Data Enrichment."*In 2023 15th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1-6. IEEE, 2023. (Scopus)

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.

[2] M. Araszkiewicz, T. Bench-Capon, E. Francesconi, M. Lauritsen, and A. Rotolo, “Thirty years of artificial intelligence and law: overviews,” *Artificial Intelligence and Law*, vol. 30, no. 4, pp. 593–610, 2022.

[3] S. Arora, Y. Liang, and T. Ma, “A simple but tough-to-beat baseline for sentence embeddings,” Jan. 2017, 5th International Conference on Learning Representations, ICLR 2017.

[4] K. Ashley, K. Branting, H. Margolis, and C. R. Sunstein, “Legal reasoning and artificial intelligence: How computers"think"like lawyers,” *University of Chicago Law School Roundtable*, vol. 8, no. 1, pp. 1–28, 2001.

[5] K. D. Ashley, *Artificial intelligence and legal analytics: new tools for law practice in the digital age*. Cambridge University Press, 2017.

[6] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen *et al.*, “Ms marco: A human generated machine reading comprehension dataset,” *arXiv preprint arXiv:1611.09268*, 2016.

[7] M. Basgalupp, R. Barros, A. de Carvalho, A. Freitas, and D. Ruiz, “Legal-tree: A lexicographic multi-objective genetic algorithm for decision tree induction,” 03 2009, pp. 1085–1090.

[8] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” *arXiv preprint arXiv:2004.05150*, 2020.

[9] T. Bench-Capon, M. Araszkiewicz, K. Ashley, K. Atkinson, F. Bex, F. Borges, D. Bourcier, P. Bourgine, J. G. Conrad, E. Francesconi *et al.*, “A history of ai and

law in 50 papers: 25 years of the international conference on ai and law," *Artificial Intelligence and Law*, vol. 20, no. 3, pp. 215–319, 2012.

[10] A. Berger and J. Lafferty, "Information retrieval as statistical translation," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 219–226.

[11] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," *Social network data analytics*, pp. 115–148, 2011.

[12] P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, P. Mehta, A. Bhattacharya, and P. Majumder, "Overview of the fire 2019 aila track: Artificial intelligence for legal assistance." in *FIRE (Working Notes)*, 2019, pp. 1–12.

[13] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh, "A comparative study of summarization algorithms applied to legal case judgments," in *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*. Springer, 2019, pp. 413–428.

[14] J. Bommarito, M. Bommarito, D. M. Katz, and J. Katz, "Gpt as knowledge worker: A zero-shot evaluation of (ai) cpa capabilities," *arXiv preprint arXiv:2301.04408*, 2023.

[15] P. Boniol, G. Panagopoulos, C. Xypolopoulos, R. E. Hamdani, D. R. Amariles, and M. Vazirgiannis, "Performance in the courtroom: Automated processing and visualization of appeal court decisions in france," *arXiv preprint arXiv:2006.06251*, 2020.

[16] L. K. Branting, C. Pfeifer, B. Brown, L. Ferro, J. Aberdeen, B. Weiss, M. Pfaff, and B. Liao, "Scalable and explainable legal prediction," *Artificial Intelligence and Law*, vol. 29, pp. 213–238, 2021.

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[18] B. G. Buchanan and T. E. Headrick, "Some speculation about artificial intelligence and legal reasoning," *Stan. L. Rev.*, vol. 23, p. 40, 1970.

[19] Q. M. Bui, C. Nguyen, D.-T. Do, N.-K. Le, D.-H. Nguyen, T.-T.-T. Nguyen, M.-P. Nguyen, and M. L. Nguyen, "Jnlp team: Deep learning approaches for tackling

long and ambiguous legal documents in coliee 2022," in *JSAI International Symposium on Artificial Intelligence*. Springer, 2022, pp. 68–83.

[20] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 89–96.

[21] C. J. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *Advances in neural information processing systems*, 2007, pp. 193–200.

[22] P. Casanovas, M. Palmirani, S. Peroni, T. Van Engers, and F. Vitali, "Semantic web for the legal domain: the next step," *Semantic web*, vol. 7, no. 3, pp. 213–227, 2016.

[23] P. Castells, M. Fernandez, and D. Vallet, "An adaptation of the vector-space model for ontology-based information retrieval," *IEEE transactions on knowledge and data engineering*, vol. 19, no. 2, pp. 261–272, 2006.

[24] I. Chalkidis, "Deep neural networks for information mining from legal texts," Ph.D. dissertation, Athens University Economics and Business, Greece, 2021.

[25] I. Chalkidis and D. Kampas, "Deep learning in law: early adaptation and legal word embeddings trained on large corpora," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 171–198, 2019.

[26] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on eu legislation," *arXiv preprint arXiv:1906.02192*, 2019.

[27] I. Chalkidis, M. Fergadiotis, and I. Androutsopoulos, "Multieurlex–a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer," *arXiv preprint arXiv:2109.00904*, 2021.

[28] I. Chalkidis, M. Fergadiotis, N. Manginas, E. Katakalou, and P. Malakasiotis, "Regulatory compliance through Doc2Doc information retrieval: A case study in EU/UK legislation where text similarity has limitations," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Online: Association for Computational Linguistics, Apr. 2021, pp. 3498–3511. [Online]. Available: https://aclanthology.org/2021.eacl-main.305

[29] I. Chalkidis, M. Fergadiotis, D. Tsarapatsanis, N. Aletras, I. Androutsopoulos, and P. Malakasiotis, "Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds. Online: Association for Computational Linguistics, Jun. 2021, pp. 226–241. [Online]. Available: https://aclanthology.org/2021.naacl-main.22

[30] C.-N. Chau, T.-S. Nguyen, and L.-M. Nguyen, "Vnlawbert: A vietnamese legal answer selection approach using bert language model," in *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE, 2020, pp. 298–301.

[31] W. S. Cooper, "A definition of relevance for information retrieval," *Information storage and retrieval*, vol. 7, no. 1, pp. 19–37, 1971.

[32] A. Crotti Junior, F. Orlandi, D. Graux, M. Hossari, D. O'Sullivan, C. Hartz, and C. Dirschl, "Knowledge graph-based legal search over german court cases," in *European Semantic Web Conference*. Springer, 2020, pp. 293–297.

[33] G. M. Csányi, D. Nagy, R. Vági, J. P. Vadász, and T. Orosz, "Challenges and open problems of legal document anonymization," *Symmetry*, vol. 13, no. 8, p. 1490, 2021.

[34] W. Cui, Y. Xiao, H. Wang, Y. Song, S.-w. Hwang, and W. Wang, "Kbqa: learning question answering over qa corpora and knowledge bases," *arXiv preprint arXiv:1903.02419*, 2019.

[35] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Proceedings of the First international conference on Machine Learning Challenges: evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, 2005, pp. 177–190.

[36] I. Dagan, B. Dolan, B. Magnini, and D. Roth, "Recognizing textual entailment: Rational, evaluation and approaches–erratum," *Natural Language Engineering*, vol. 16, no. 1, pp. 105–105, 2010.

[37] Z. Dai and J. Callan, “Deeper text understanding for ir with contextual neural language modeling,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 985–988.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, Jun. 2019, pp. 4171–4186.

[39] J. S. Dhani, R. Bhatt, B. Ganesan, P. Sirohi, and V. Bhatnagar, “Similar cases recommendation using legal knowledge graphs,” *arXiv preprint arXiv:2107.04771*, 2021.

[40] M. Dragoni, S. Villata, W. Rizzi, and G. Governatori, “Combining nlp approaches for rule extraction from legal documents,” in *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*, 2016.

[41] X. Duan, B. Wang, Z. Wang, W. Ma, Y. Cui, D. Wu, S. Wang, T. Liu, T. Huo, Z. Hu *et al.*, “Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension,” in *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*. Springer, 2019, pp. 439–451.

[42] D. Duarte, P. M. Lopes, and J. S. Sampaio, *Legal Interpretation and Scientific Knowledge*. Springer, 2019.

[43] Y. Feng, C. Li, and V. Ng, “Legal judgment prediction via event extraction with constraints,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 648–664.

[44] M. Fernández-Barrera and G. Sartor, *The legal theory perspective: doctrinal conceptual systems vs. computational ontologies*. Springer, 2011.

[45] E. Filtz, “Building and processing a knowledge-graph for legal data,” in *The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part II 14*. Springer, 2017, pp. 184–194.

[46] J. Gao, P. Pantel, M. Gamon, X. He, and L. Deng, “Modeling interestingness with deep neural networks,” in *EMNLP*, 2014.

[47] M. Grabmair, “Modeling purposive legal argumentation and case outcome prediction using argument schemes in the value judgment formalism,” Ph.D. dissertation, University of Pittsburgh, 2016.

[48] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng, “A deep look into neural ranking models for information retrieval,” *Information Processing and Management*, vol. 57, no. 6, p. 102067, 2020.

[49] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] F. G. Horton, “A conceptual framework for the law and technology knowledge domain,” Ph.D. dissertation, Southern Cross University, 2021.

[51] B. Hu, Z. Lu, H. Li, and Q. Chen, “Convolutional neural network architectures for matching natural language sentences,” in *Advances in neural information processing systems*, 2014, pp. 2042–2050.

[52] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 2333–2338.

[53] W. Huang, D. Hu, Z. Deng, and J. Nie, “Named entity recognition for chinese judgment documents based on bilstm and crf,” *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, pp. 1–14, 2020.

[54] S. Ito, “Lecture series on ultimate facts,” *Shojihomu (in Japanese)*, 2008.

[55] D. Jain, M. D. Borah, and A. Biswas, “Summarization of legal documents: Where are we now and the way forward,” *Computer Science Review*, vol. 40, p. 100388, 2021.

[56] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, “Fasttext.zip: Compressing text classification models,” *arXiv preprint arXiv:1612.03651*, 2016.

[57] A. Kanapala, S. Pal, S. Dara, and S. Jannu, “Applying an information retrieval approach to retrieve relevant articles in the legal domain,” *Annals of Data Science*, pp. 1–18, 2022.

[58] Y. Kano, M.-Y. Kim, M. Yoshioka, Y. Lu, J. Rabelo, N. Kiyota, R. Goebel, and K. Satoh, “Coliee-2018: Evaluation of the competition on legal information extraction and entailment,” in *New Frontiers in Artificial Intelligence: JSAI-isAI 2018 Workshops, JURISIN, AI-Biz, SKL, LENLS, IDAA, Yokohama, Japan, November 12–14, 2018, Revised Selected Papers*. Springer, 2019, pp. 177–192.

[59] D. M. Katz, C. Coupette, J. Beckedorf, and D. Hartung, “Complex societies and the growth of the law,” *Scientific Reports*, vol. 10, no. 1, p. 18737, 2020.

[60] D. M. Katz, D. Hartung, L. Gerlach, A. Jana, and M. J. Bommarito II, “Natural language processing in the legal domain,” *arXiv preprint arXiv:2302.12039*, 2023.

[61] P. M. Kien, H.-T. Nguyen, N. X. Bach, V. Tran, M. Le Nguyen, and T. M. Phuong, “Answering legal questions by learning neural attentive text representation,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 988–998.

[62] P. M. Kien, H.-T. Nguyen, N. X. Bach, V. Tran, M. L. Nguyen, and T. M. Phuong, “Answering legal questions by learning neural attentive text representation,” in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 988–998. [Online]. Available: https://aclanthology.org/2020.coling-main.86

[63] M.-Y. Kim, J. Rabelo, and R. Goebel, “Statute law information retrieval and entailment,” in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, 2019, pp. 283–289.

[64] M.-Y. Kim, J. Rabelo, R. Goebel, M. Yoshioka, Y. Kano, and K. Satoh, “Coliee 2022 summary: Methods for legal document retrieval and entailment,” in *Proceedings of the Sixteenth International Workshop on Juris-informatics (JURISIN 2022)*, 2022.

[65] M.-Y. Kim, J. Rabelo, K. Okeke, and R. Goebel, “Legal information retrieval and entailment based on bm25, transformer and semantic thesaurus methods,” *The Review of Socionetwork Strategies*, vol. 16, no. 1, pp. 157–174, 2022.

[66] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.

[67] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[68] R. Kowalski and A. Datoo, “Logical english meets legal english for swaps and derivatives,” *Artificial Intelligence and Law*, pp. 1–35, 2021.

[69] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[70] E. Leitner, G. Rehm, and J. Moreno-Schneider, “Fine-grained named entity recognition in legal documents,” in *International Conference on Semantic Systems*. Springer, 2019, pp. 272–287.

[71] E. Leitner, G. Rehm, and J. M. Schneider, “A dataset of german legal documents for named entity recognition,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4478–4485.

[72] E. H. Levi, “An introduction to legal reasoning,” 2013.

[73] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv preprint arXiv:1910.13461*, 2019.

[74] B. Li and D. Pi, “Learning deep neural networks for node classification,” *Expert Systems with Applications*, vol. 137, pp. 324–334, 2019.

[75] G.-K. J. Li, C. V. Trappey, A. J. Trappey, and A. A. Li, “Ontology-based knowledge representation and semantic topic modeling for intelligent trademark legal precedent research,” *World Patent Information*, vol. 68, p. 102098, 2022.

[76] L. Li, Z. Bi, H. Ye, S. Deng, H. Chen, and H. Tou, “Text-guided legal knowledge graph reasoning,” in *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction: 6th China Conference, CCKS 2021, Guangzhou, China, November 4-7, 2021, Proceedings 6*. Springer, 2021, pp. 27–39.

[77] D. Liga, “Hybrid artificial intelligence to extract patterns and rules from argumentative and legal texts,” 2022.

[78] T.-Y. Liu, “Learning to rank for information retrieval.” *Foundations and Trends in Information Retrieval*, vol. 3, pp. 225–331, 01 2009.

[79] A. Louis and G. Spanakis, “A statutory article retrieval dataset in French,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 6789–6803. [Online]. Available: https://aclanthology.org/2022.acl-long.468

[80] H. P. Luhn, “A statistical approach to mechanized encoding and searching of literary information,” *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[81] Y. Ma, Y. Wu, Q. Ai, Y. Liu, Y. Shao, M. Zhang, and S. Ma, “Incorporating structural information into legal case retrieval,” *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1–28, 2023.

[82] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, and S. Ghosh, “Measuring similarity among legal court case documents,” in *Proceedings of the 10th annual ACM India compute conference*, 2017, pp. 1–9.

[83] S. Marchesin, A. Purpura, and G. Silvello, “Focal elements of neural information retrieval models. an outlook through a reproducibility study,” *Information Processing and Management*, vol. 57, no. 6, p. 102109, 2020.

[84] K. T. Maxwell and B. Schafer, “Concept and context in legal information retrieval,” in *Legal Knowledge and Information Systems*. IOS Press, 2008, pp. 63–72.

[85] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[86] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, “Rnnlm-recurrent neural network language modeling toolkit,” in *Proc. of the 2011 ASRU Workshop*, 2011, pp. 196–201.

[87] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in neural information processing systems*, vol. 26, 2013.

[88] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, “Advances in pre-training distributed word representations,” in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[89] B. Mitra, F. Diaz, and N. Craswell, “Learning to match using local and distributed representations of text for web search,” in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1291–1299.

[90] M. Navas Loro, “Processing, identification and representation of temporal expressions and events in legal documents,” Ph.D. dissertation, ETSI_Informatica, 2021.

[91] D. Q. Nguyen, T. Vu, M. Dras, M. Johnson *et al.*, “A fast and accurate vietnamese word segmenter,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[92] H.-T. Nguyen, “Toward improving attentive neural networks in legal text processing,” *arXiv preprint arXiv:2203.08244*, 2022.

[93] H.-T. Nguyen, H.-Y. T. Vuong, P. M. Nguyen, B. T. Dang, Q. M. Bui, S. T. Vu, C. M. Nguyen, V. Tran, K. Satoh, and M. L. Nguyen, “Jnlp team: Deep learning for legal processing in coliee 2020,” *arXiv preprint arXiv:2011.08071*, 2020.

[94] H.-T. Nguyen, P. M. Nguyen, T.-H.-Y. Vuong, Q. M. Bui, C. M. Nguyen, B. T. Dang, V. Tran, M. L. Nguyen, and K. Satoh, “Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021,” *arXiv preprint arXiv:2106.13405*, 2021.

[95] H.-T. Nguyen, V. Tran, P. M. Nguyen, T.-H.-Y. Vuong, Q. M. Bui, C. M. Nguyen, B. T. Dang, M. L. Nguyen, and K. Satoh, “Paralaw nets–cross-lingual sentence-level pretraining for legal text processing,” *arXiv preprint arXiv:2106.13403*, 2021.

[96] H.-T. Nguyen, M.-K. Phi, X.-B. Ngo, V. Tran, L.-M. Nguyen, and M.-P. Tu, “Attentive deep neural networks for legal document retrieval,” *Artificial Intelligence and Law*, pp. 1–30, 2022.

[97] H.-T. Nguyen, F. Wachara, F. Nishino, and K. Satoh, “A multi-step approach in translating natural language into logical formula,” pp. 103–112, 2022.

[98] H. Nguyen, F. Toni, K. Stathis, and K. Satoh, “Beyond logic programming for legal reasoning,” vol. 3437, 2023. [Online]. Available: https://ceur-ws.org/Vol-3437/paper2LPLR.pdf

[99] T.-S. Nguyen, L.-M. Nguyen, S. Tojo, K. Satoh, and A. Shimazu, “Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts,” *Artificial Intelligence and Law*, vol. 26, pp. 169–199, 2018.

[100] J. Ni, G. H. Ábrego, N. Constant, J. Ma, K. B. Hall, D. Cer, and Y. Yang, “Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models,” *arXiv preprint arXiv:2108.08877*, 2021.

[101] R. Nogueira, Z. Jiang, and J. Lin, “Document ranking with a pretrained sequence-to-sequence model,” *arXiv preprint arXiv:2003.06713*, 2020.

[102] M. Y. Noguti, E. Vellasques, and L. S. Oliveira, “Legal document classification: An application to law area prediction of petitions to public prosecution service,” in *2020 International joint conference on neural networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[103] D. W. Oard and W. Webber, “Information retrieval for e-discovery,” *Information Retrieval*, vol. 7, no. 2-3, pp. 99–237, 2013.

[104] J. V. Orth, *The Tree of Legal Knowledge: Imagining Blackstone’s Commentaries*. Springer, 2023.

[105] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, “Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[106] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, “Text matching as image recognition,” ser. AAAI’16. AAAI Press, 2016.

[107] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[108] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[109] M. O. Pflueger, I. Franke, M. Graf, and H. Hachtel, “Predicting general criminal recidivism in mentally disordered offenders using a random forest approach,” *BMC psychiatry*, vol. 15, no. 1, pp. 1–10, 2015.

[110] J. Rabelo, M.-Y. Kim, and R. Goebel, “Combining similarity and transformer methods for case law entailment,” in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ser. ICAIL ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 290–296.

[111] J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, and K. Satoh, “A summary of the coliee 2019 competition,” in *New Frontiers in Artificial Intelligence: JSAI-isAI International Workshops, JURISIN, AI-Biz, LENLS, Kansei-AI, Yokohama, Japan, November 10–12, 2019, Revised Selected Papers 10*. Springer, 2020, pp. 34–49.

[112] ——, “Coliee 2020: methods for legal document retrieval and entailment,” in *New Frontiers in Artificial Intelligence: JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers 12*. Springer, 2021, pp. 196–210.

[113] J. Rabelo, R. Goebel, M.-Y. Kim, Y. Kano, M. Yoshioka, and K. Satoh, “Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021,” *The Review of Socionetwork Strategies*, vol. 16, no. 1, pp. 111–133, 2022.

[114] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” *The University of British Columbia Repository*, 2018.

[115] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[116] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[117] A. Ravichander, A. W. Black, S. Wilson, T. Norton, and N. Sadeh, “Question answering for privacy policies: Combining computational and legal perspectives,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan,

Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4947–4958. [Online]. Available: https://aclanthology.org/D19-1500

[118] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[119] E. Rissland, "Artificial intelligence and legal reasoning: A discussion of the field and gardner's book," *AI Magazine*, vol. 9, no. 3, pp. 45–45, 1988.

[120] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *SIGIR'94*. Springer, 1994, pp. 232–241.

[121] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," *arXiv preprint arXiv:1907.10903*, 2019.

[122] J. Ruhl, D. M. Katz, and M. J. Bommarito, "Harnessing legal complexity," *Science*, vol. 355, no. 6332, pp. 1377–1378, 2017.

[123] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[124] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[125] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[126] T. Saracevic, "Relevance reconsidered," in *Proceedings of the second conference on conceptions of library and information science (CoLIS 2)*. ACM New York, 1996, pp. 201–218.

[127] M. Saravanan, B. Ravindran, and S. Raman, "Improving legal information retrieval using an ontological framework," *Artificial Intelligence and Law*, vol. 17, no. 2, pp. 101–124, 2009.

[128] G. Sartor, P. Casanovas, M. Biasiotti, and M. Fernández-Barrera, "Approaches to legal ontologies: Theories, domains, methodologies. law," *Governance and Technology series. Springer*, 2011.

[129] K. Satoh, K. Asai, T. Kogawa, M. Kubota, M. Nakamura, Y. Nishigai, K. Shirakawa, and C. Takano, "Proleg: an implementation of the presupposed ultimate fact theory of japanese civil code by prolog technology," in *JSAI International Symposium on Artificial Intelligence*. Springer, 2010, pp. 153–164.

[130] J. Šavelka and K. D. Ashley, "Legal information retrieval for understanding statutory terms," *Artificial Intelligence and Law*, pp. 1–45, 2021.

[131] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[132] Y. Shao, J. Mao, Y. Liu, W. Ma, K. Satoh, M. Zhang, and S. Ma, "Bert-pli: Modeling paragraph-level interactions for legal case retrieval," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 3501–3507.

[133] Y. Shao, H. Li, Y. Wu, Y. Liu, Q. Ai, J. Mao, Y. Ma, and S. Ma, "An intent taxonomy of legal case retrieval," *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1–27, 2023.

[134] S. Shen, G. Qi, Z. Li, S. Bi, and L. Wang, "Hierarchical chinese legal event extraction via pedal attention mechanism," in *Proceedings of the 28th international conference on computational linguistics*, 2020, pp. 100–113.

[135] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, 2014, pp. 101–110.

[136] F. Song and W. B. Croft, "A general language model for information retrieval," in *Proceedings of the eighth international conference on Information and knowledge management*, 1999, pp. 316–321.

[137] F. Sovrano, M. Palmirani, and F. Vitali, "Legal knowledge extraction for knowledge graph based question-answering," in *Legal Knowledge and Information Systems*. IOS Press, 2020, pp. 143–153.

[138] K. Sugathadasa, B. Ayesha, N. de Silva, A. S. Perera, V. Jayawardana, D. Lakmal, and M. Perera, “Legal document retrieval using document vector embeddings and deep learning,” in *Science and Information Conference*. Springer, 2018, pp. 160–175.

[139] O.-M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, and J. Van Genabith, “Exploring the use of text classification in the legal domain,” *arXiv preprint arXiv:1710.09306*, 2017.

[140] R. E. Susskind, “Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning,” *The modern law review*, vol. 49, no. 2, pp. 168–194, 1986.

[141] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1556–1566.

[142] M. Tang, C. Su, H. Chen, J. Qu, and J. Ding, “Salkg: a semantic annotation system for building a high-quality legal knowledge graph,” in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 2153–2159.

[143] N. H. Thanh, B. M. Quan, C. Nguyen, T. Le, N. M. Phuong, D. T. Binh, V. T. H. Yen, T. Racharak, N. Le Minh, T. D. Vu *et al.*, “A summary of the alqac 2021 competition,” in *2021 13th international conference on knowledge and systems engineering (kse)*. IEEE, 2021, pp. 1–5.

[144] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[145] V. Tran, M. Le Nguyen, S. Tojo, and K. Satoh, “Encoded summarization: summarizing documents into continuous vector space for legal case retrieval,” *Artificial Intelligence and Law*, vol. 28, pp. 441–467, 2020.

[146] S. S. Ulmer, “Quantitative analysis of judicial processes: Some practical and theoretical applications,” *Law and Contemporary Problems*, vol. 28, no. 1, pp. 164–184, 1963.

[147] M. Van Opijnen and C. Santos, “On the concept of relevance in legal information retrieval,” *Artificial Intelligence and Law*, vol. 25, no. 1, pp. 65–87, 2017.

[148] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[149] T. Vu, D. Q. Nguyen, M. D. Dai Quoc Nguyen, and M. Johnson, “Vncorenlp: A vietnamese natural language processing toolkit,” *NAACL HLT 2018*, p. 56, 2018.

[150] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, “A deep architecture for semantic matching with multiple positional sentence representations,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16. AAAI Press, 2016, p. 2835–2841.

[151] Y. Wang, W. Wang, Y. Liang, Y. Cai, J. Liu, and B. Hooi, “Nodeaug: Semi-supervised node classification with data augmentation,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 207–217.

[152] S. Wehnert, V. Sudhi, S. Dureja, L. Kutty, S. Shahania, and E. W. De Luca, “Legal norm retrieval with variations of the bert model combined with tf-idf vectorization,” in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ser. ICAIL ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 285–294. [Online]. Available: https://doi.org/10.1145/3462757.3466104

[153] S. Wehnert, L. Kutty, and E. W. De Luca, “Using textbook knowledge for statute retrieval and entailment classification,” in *New Frontiers in Artificial Intelligence*, Y. Takama, K. Yada, K. Satoh, and S. Arai, Eds. Cham: Springer Nature Switzerland, 2023, pp. 125–137.

[154] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao, “Adapting boosting for information retrieval measures,” *Information Retrieval*, vol. 13, no. 3, pp. 254–270, 2010.

[155] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang *et al.*, “Cail2018: A large-scale legal dataset for judgment prediction,” *arXiv preprint arXiv:1807.02478*, 2018.

[156] S. Xiao, S. Wang, Y. Dai, and W. Guo, "Graph neural networks in node classification: survey and evaluation," *Machine Vision and Applications*, vol. 33, pp. 1–19, 2022.

[157] J. Yang, W. Ma, M. Zhang, X. Zhou, Y. Liu, and S. Ma, "Legalgnn: Legal information enhanced graph neural network for recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 2, pp. 1–29, 2021.

[158] Z. A. Yilmaz, S. Wang, W. Yang, H. Zhang, and J. Lin, "Applying BERT to document retrieval with birch," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 2019, pp. 19–24.

[159] Z. A. Yilmaz, W. Yang, H. Zhang, and J. Lin, "Cross-domain modeling of sentence-level evidence for document retrieval," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3481–3487.

[160] M. Yoshioka, Y. Aoki, and Y. Suzuki, "Bert-based ensemble methods with data augmentation for legal textual entailment in coliee statute law task," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021, pp. 278–284.

[161] M. Yoshioka, Y. Suzuki, and Y. Aoki, "Hukb at the coliee 2022 statute law task," in *JSAI International Symposium on Artificial Intelligence*. Springer, 2022, pp. 109–124.

[162] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, "Big bird: Transformers for longer sequences," *Advances in neural information processing systems*, vol. 33, pp. 17 283–17 297, 2020.

[163] Y. Zeng, R. Wang, J. Zeleznikow, and E. Kemp, "Knowledge representation for the intelligent legal case retrieval," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2005, pp. 339–345.

[164] C. Zhai and J. Lafferty, “A study of smoothing methods for language models applied to ad hoc information retrieval,” in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 268–276.

[165] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, and D. E. Ho, “When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings,” in *Proceedings of the eighteenth international conference on artificial intelligence and law*, 2021, pp. 159–168.

[166] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and S. Maosong, “Jec-qa: a legal-domain question answering dataset,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9701–9708.

[167] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, “How does nlp benefit legal system: A summary of legal artificial intelligence,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5218–5230.