HKN: Heterogeneous Knowledge-aware Network for Similar Legal Case Retrieval

No Author Given

No Institute Given

Abstract. Similar legal case retrieval aims to search for legal cases relevant to the target case. The legal documents of cases contain a variety of knowledge that helps the model observe multiple aspects of information and avoid model ambiguity. However, current similar legal case retrieval methods do not make good use of this knowledge. This paper proposes a multi-level heterogeneous knowledge-aware network (HKN) that integrates entity knowledge and event knowledge to achieve fine-grained retrieval of similar legal cases in Chinese. Results on multiple similar legal case retrieval datasets show that HKN outperforms other existing baselines.

Keywords: Similar legal case retrieval · Heterogeneous knowledge · Event representation.

1 Introduction

Similar legal case retrieval (SLCR) aims to search for legal cases similar to the target case. As an integral part of the legal assistance system, SLCR can help legal practitioners improve the efficiency of judicial decision-making and effectively promote the fairness of the judicial system. The SLCR task can be formally defined as follows: given a target case D and a candidate case set S, output cases in the candidate set S that are similar to the target case. We mainly research similar legal case retrieval in Chinese criminal cases. However, the research content and results also apply to similar legal case retrieval tasks in other languages or other jurisdictions' legal systems.

The documents of criminal cases mainly include four parts: litigant information, fact description, judgment result, and citation of legal provisions. The fact description part records the occurrence of the case in detail. It introduces critical information, such as the circumstances and elements of the crime, which plays a representative role in legal documents. Experts and scholars often measure the similarity of legal cases through factual similarity.

Unlike text matching in the general domain, legal case retrieval tasks require higher reliability, aiming to measure the factual similarity between legal documents rather than simply calculating the textual similarity. Legal documents are complex that even in entirely different cases, few differences can be found in the description and structure. Therefore, similar legal case retrieval is still a challenging task. Most early similar case retrieval methods[14] used explicit text

Fig. 1. An example of dissimilar cases A and B. Yellow words are entities, blue backgrounds are event triggers, and pink backgrounds are event arguments.

statistical features to measure text similarities, such as topic models[10] and TF-IDF methods[6]. These methods, although concise, do not consider the semantic information of the text at all. Current similar case retrieval methods mainly use a pre-trained language model to encode the contextual semantic features of the text and determine similar cases through the vector distance of the text representation. However, legal documents are generally longer than the limit of PLMs and filled with legal terminologies, making methods based on pre-trained models difficult to understand. Although researchers introduced legal features to enhance the model's understanding of legal documents, they only considered discrete knowledge at the entity level and ignored higher-level event knowledge. Events are usually used to describe an action and several attributes related to the action, such as the subject, object, and mediator of the action. The words that describe this action are trigger words, and the attributes are called arguments. Indeed, event knowledge is essential to describe the complete picture of legal facts. As shown in Fig.1, texts A and B are dissimilar cases with the same high-frequency entities (e.g. "drugs"). Judging by entity-level information alone, texts A and B can be mistaken for similarity. While at the event level, texts A and B contain significantly different event types and can be easily identified as dissimilar cases. This phenomenon indicates that high-frequency entity information may cause model bias, and complementing event information reduces the risk of misjudgment. In the legal domain, discrete legal entity knowledge can supplement event information to form a hierarchical knowledge structure and achieve more accurate retrieval of similar cases.

In order to solve the above problems, this paper proposes a multi-level heterogeneous knowledge-aware network (HKN). HKN integrates the information of the text layer, entity layer, and event layer to achieve fine-grained Chinese similar legal cases retrieval. Specifically, in addition to the legal text representation layer, we delicately design multi-layers of knowledge networks. For entity information, we combine the external knowledge base legalKG to construct an entity knowledge network to help the model understand the semantics of terms in the legal field. For event information, we observe that legal documents include an event sequence that summarizes legal facts in detail. Therefore, HKN constructs an event knowledge network to mine the potential features of event sequences. The event network enhances the interpretability of the prediction results and assists the model in obtaining global information about the text. In addition, a combination of legal entity and event knowledge is imperative. HKN aligns the entity and event knowledge to construct a heterogeneous knowledge network. Heterogeneous knowledge helps the model comprehensively consider the multi-levels of information and the interaction between them and strengthens the model's ability to identify subtle differences between legal texts.

The main contributions of this paper can be summarized as follows:

- We propose a method to construct a multi-level legal knowledge network, which can utilize the existing legal knowledge graph and augment it with the entity and event knowledge extracted from legal documents.
- We propose a multi-level heterogeneous knowledge-aware network representation architecture that integrates text representation, entity knowledge network representation and event knowledge network representation.
- We conduct extensive experiments on three datasets. Experimental results reveal that HKN achieves state-of-the-art performance on different truthful datasets.

2 Related Work

Similar legal case retrieval is a challenging task, and how to judge the similarity between legal cases is a problem worth exploring. Initial methods of determining similar legal cases rely on statistical results [14]. Winkels et al.[15] utilized a topic model to assign different topics to legal documents and performed similar legal case matching through topic similarity, while Kumar et al.[6] employed TF-IDF to collect term frequency-inverse document frequency information of words in different documents to generalize document-level features. Bhattacharya et al.[1] measured legal document similarity by mining features of citation information between legal documents. However, statistical-based methods concentrate on explicit textual features but fail to capture underlying semantic information.

Some research works utilized pre-trained models to embed documents into a low-dimensional continuous vector space and measured the relevance of legal documents through semantic similarity. Hong et al.[4] proposed the Legal Feature Enhanced Semantic Matching Network (LFESM) to capture subtle distinctions between legal document pairs. Shao et al.[12] proposed a legal case retrieval model BERT-PLI, which employed a pre-trained model to encode paragraph-level semantics and aggregated paragraph-level interaction information to deduce document relevance. In order to augment understanding of Chinese legal long-text documents, Xiao et al.[16] proposed a pre-trained language model Lawformer, which can handle long texts.

Actually, some problems exist in methods of similar Chinese legal cases retrieval based on pre-trained language models. Legal documents contain a large amount of domain-specific knowledge, which is difficult for those methods based on pre-trained models to understand. Li et al.[7] proposed the Interactive Attention Capsule Network (IACN), performing interpretable predictions by capturing fine-grained element-level similarity. Bi et al.[2] constructed a hybrid heterogeneous graph containing term network and external knowledge through legal documents and legal encyclopedia data, and utilized this graph to capture the intrinsic relationship between legal documents and legal domain knowledge. However, they failed to consider event-level information in legal documents. In contrast with discrete entity-level knowledge, event-level information summarizes the whole course of criminal fact in legal judgments.

3 Methodology

3.1 Overview

As is shown in Fig.2, the overall framework of HKN is organized into three steps. In step 1, we separately utilize entity-level knowledge and event-level knowledge to construct the entity and event knowledge networks. In step 2, we employ different encoders to get the semantic embeddings of the knowledge network in multi-levels and the heterogeneous knowledge network. In step 3, we concatenate the embeddings of text, entity information, event information, and hybrid heterogeneous information as the final representation of the legal document. A triplet margin loss layer is added to achieve better performance optimization.

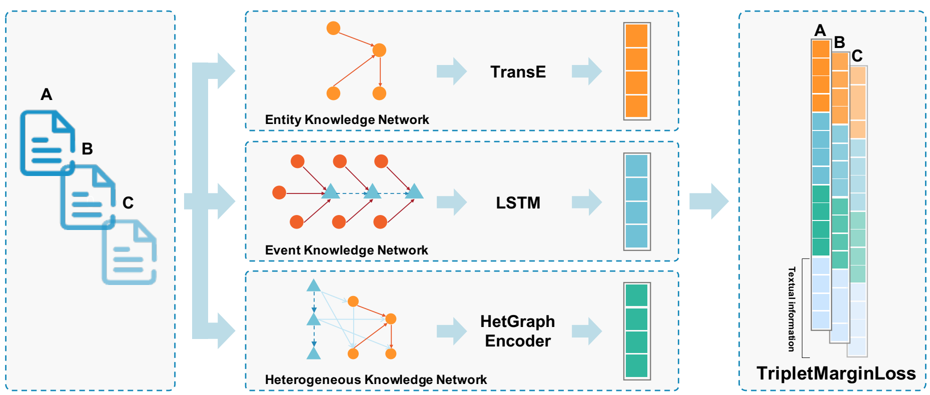

Fig. 2. Overall Framework of HKN.

3.2 Knowledge Network Construction

We follow the legal knowledge graph legalKG proposed by Bi et al.[2]. They extract entities from encyclopedias, legal documents and other related texts, and construct them according to the conceptual relationship of entities in encyclopedias. Given a piece of text D, we extract a set of legal-related entities E from the text and then perform entity alignment with legalKG. By connecting the entities in D according to the relationship between entities in legalKG, the entity knowledge network G_{ent} of the legal text can be obtained. Specifically, the entity vertex information is the name of the entity, and the directed edge between the legal entities t_1 and t_2 in Fig.3 means that t_1 quotes t_2 in the encyclopedia concept.

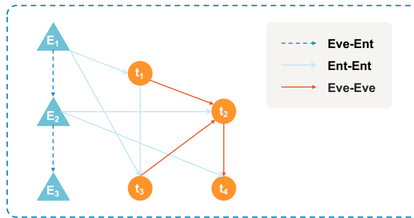

Fig. 3. Heterogeneous knowledge network of events and entities.

Building an event knowledge network is mainly divided into three steps: event extraction, event relation extraction, and network connection. First, we follow the Chinese event extraction method (PAJHEE) based on the Pedal Attention mechanism proposed by Shen et al.[13]. We utilize PAJHEE to extract the event sequence of "event description" in legal documents, including event trigger word, event type, and argument information. Then, we employ an event-causal extraction method based on derived prompt learning to extract causal relationships between events. It is worth noting that the events in the " event description" text are described in chronological order, so there is a temporal relationship between adjacent events by default. The event trigger word is used as the event vertex, the event type and argument information are used as the attribute vertex to connect the corresponding event vertex. Finally, the event vertices are arranged according to the text sequence in "event description" . If there is a causal relationship between adjacent event vertices, the two events are connected by a causal relationship edge; if there is no causal relationship, the two events are connected by a temporal relationship. The event knowledge network is represented as $G_{eve} = (V, E)$, where $V = \{V_e \cup V_a\}$ represents the set of event vertex V_e and argument vertex V_a, and $E = \{E_a \cup E_t \cup E_r\}$ represents the connecting edge between vertices. E_a stands for the edge between event vertices and argument vertices, E_t stands for the temporal edge between event vertices, and E_r stands for the causal edge between event vertices.

Law-related events or entities contain complex domain knowledge and concepts. Inspired by the legal hybrid knowledge network L-HetG [2], we construct a heterogeneous knowledge network by aligning event arguments and entities. Given an entity knowledge network G_{ent} and an event knowledge network G_{eve}, we first match the event argument vertices of G_{eve} with the legal entity vertices of G_{ent}, then fuse the aligned event argument vertices and legal entity vertices into a single vertex. At the same time, Unaligned argument vertices and legal entity vertices are discarded. We design three types of connection edges for the heterogeneous knowledge network $G_{het} = (V, E)$. The definitions of edges are as follows:

- $E_{eve-eve}$: Directed edges representing temporal relationships or causal relationships between event vertices, equivalent to $\{E_t \cup E_r\}$ in the event knowledge network.
- $E_{eve-ent}$: Directed edges between the event vertex and the entity vertex, equivalent to E_a in the event knowledge network.
- $E_{ent-ent}$: Directed edges between entity vertices, equivalent to those in the entity knowledge network.

3.3 Knowledge Network Representation

Text representation Text representation is a fundamental natural language processing task. HKN uses standard pre-trained language models to learn text representations, such as BERT and RoBERTa. Given a text D, input a pre-trained language model and output the text representation vector Z_D. To be more particular,

$$Z_D = PLM(D). \tag{1}$$

Entity Knowledge Network Representation Because single entity information is discrete and sparse, the representation of entity knowledge network mainly encodes the content information of entity knowledge. Therefore, HKN deliberately ignores the relationship between entities in the process of knowledge graph representation to improve efficiency and focuses on capturing critical entity semantics.

Given an entity knowledge network $G_{ent} = (V, E)$, where E represents the entity vertex, V represents the relationship edge between entities, and the entity knowledge network represents the mapping function that learns the graph G_{ent} to Z_{ent}. The model regards each vertex in the knowledge graph G_{ent} as discrete entity knowledge and randomly sorts the vertices into a sequence structure. First, a placeholder is added to the head of the entity sequence as the global embedding node of the network, and the entity vector is initialized using the TransE knowledge graph embedding method. Unlike language models, our model does not need to model the order of meaningless randomly arranged entities. Therefore, we remove the position encoding module. The initialized entity vector will pass through multiple stacked Transformer encoders to encode semantic information and deep interactions between entities based on a self-attention mechanism. To be more particular,

$$Z_{ent} = Transformer\{TransE(e_i)\}, \tag{2}$$

$TransE(\cdot)$ represents the TransE knowledge graph embedding method, and e_i represents the $i-th$ entity randomly arranged. In the output result, the representation vector of the global embedding node aggregates the knowledge features of all entities in the entity knowledge network.

Event Knowledge Network Representation Event knowledge network representation learning not only needs to mine latent semantic information but also needs to model the sequential information of events. HKN uses a sequence model to encode event-level semantic information. According to the relationship between events, multiple events are concatenated to form an event chain. First, we aggregate the argument information of events using linear tensor operations, and the semantic event feature of the i-th event vertex E_i is encoded as:

$$h_{E_i} = word2vec(E_i) \| \frac{1}{| N_i |} (\sum_{w_{ij} \in N_i} word2vec(w_{ij})), \tag{3}$$

where the text information of the event node E_i is the event trigger word. $N_i = \{w_{ij}\}$ represents the set of adjacent argument nodes of E_i, and $\|$ stands for concat operation. For the relationship edges E_t and E_r between events, HKN set two learnable vectors h_t and h_r and inserts them into the sequence of event vertex representations in order to obtain the initial vector sequence s of the event chain.In order to effectively mine the sequence features of the event knowledge network, we apply an LSTM to encode event chains. The corresponding initial vector on the event chain is input to the event chain coding module, and the final output hidden variable is used to represent the event knowledge network. The detailed calculation steps are as follows:

$$Z_{eve} = LSTM(s). \tag{4}$$

The encoding method of the event chain can be flexibly extended, and various sequence feature encoding methods can be well adapted to this task, such as RNN and Transformer encoder.

Heterogeneous Knowledge Network Representation Given a heterogeneous knowledge network $G_{het} = (V, E)$, where $V = \{V_{ent} \cup V_{eve}\}$ represents the set of entity vertex V_{ent} and event vertex V_{eve}, $E = \{E_{eve-eve} \cup E_{eve-ent} \cup E_{ent-ent}\}$ represents the connecting edge between vertices. HKN first defines several different meta-paths on the heterogeneous graph and unifies the representation space of different types of vertices; then calculates the representation of the neighbor vertices of the target vertex on a certain meta-path; finally uses the attention mechanism to aggregate the target vertices' representation vectors on different meta-paths. Further, we design meta-paths $P\#1(EVE - ENT)$ and $P\#2(EVE - ENT - ENT)$ suitable for heterogeneous knowledge networks, considering the first-order and second-order entity neighborhood knowledge of event vertices, respectively. Eventually, HKN aggregates the information on these meta-paths and yields the final representation of the event vertex.

Before encoding a heterogeneous knowledge network, the input vertices need to be initialized. The event vertex employs the initialization vector in the event-knowledge network representation module. The entity vertex employs the TransE method to initialize the entity vector. The learnable edge vertex representation is randomly initialized. Due to different encoding methods, different vertices may be projected into different semantic spaces. Therefore, we transform various types of vertices into a unified vector space to better aggregate semantic information in heterogeneous networks.

The vertex transformation formula is as follows: For a vertex or edge x of type O, its initialization vector is h_x, and the feature information of the vertex is mapped to a unified representation space through the corresponding mapping matrix W_O, and the representation vector is $h_x^{'}$.

$$h_x^{'} = W_O \cdot h_x, \tag{5}$$

where O represents the two types of vertices and three types of edges contained in G_{het}, $h_x \in R^{d_O}$, $W_O \in R^{d*d_O}$, $h_x^{'} \in R^d$. After applying vertex vector transformation, entity vertices, event vertices, and their connected edges share a unified representation space, facilitating the subsequent aggregation of heterogeneous vertices information on meta-paths.

As shown in Fig.4, in order to comprehensively consider the information of event vertices and entity vertices, we elaborately designed two meta-paths $P\#1(EVE - ENT)$ and $P\#2(EVE - ENT - ENT)$. By considering the interaction information between the event vertex and its first-order and second-order neighborhood entity vertices, HKN updates the representation vector of the event vertex. In particular, a vertex sequence that conforms to the schema defined by P is called an instance of P. For example, $E_1 - t_1 - t_2$ in Fig.4 is an instance of the meta-path $P\#2$. First, HKN performs intra-metapath aggregation on all instances of meta-path starting from the vertex of the target event. Then HKN aggregates these different meta-paths attentively to obtain the final representation of the target event vertex. The specific operation process can be defined as follows.

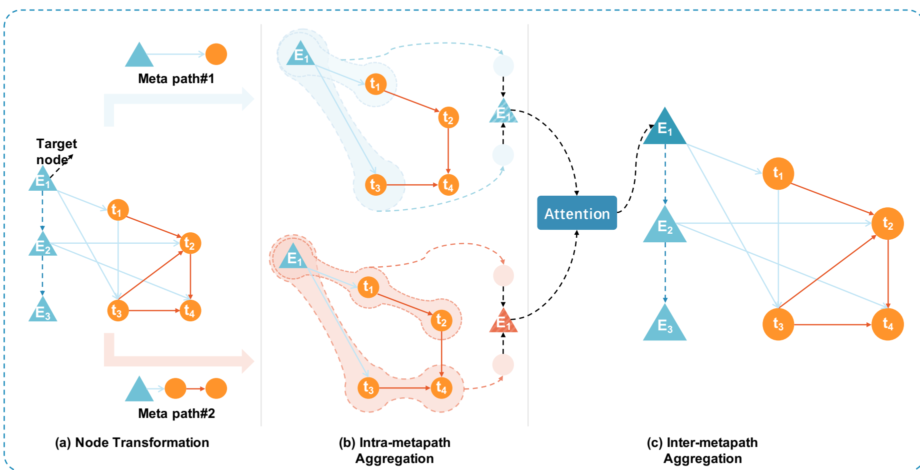

Fig. 4. Meta-path aggregation process in heterogeneous knowledge networks.

Given a meta-path P, HKN first encodes an instance of P to learn the structure and semantic information of the target vertex, the meta path-based neighbor vertex, as well as the context vertices in between. Let $P(v, u) = \{n_0, n_1, n_2, \ldots, n_M\}$ be an instance of meta-path connecting the target event vertex $v \in V_{eve}$ and the meta path-based entity neighbor $u \in N_v^P$, where $n_0 = u$, $n_M = v$. N_v^P represents all the entity neighbors of the node v based on the meta path P. In intra-metapath aggregation of metapath P, HKN aggregates all the

P-based instance representations $\{h_{P(v,u)} \mid u \in N_v^P\}$ into a single vector h_v^P. Inspired by knowledge graph embedding, the aggregation information q_i of node n_i on $P(v,u)$ can be formulated as:

$$q_i = h_{n_i}^{'} + q_{i-1} \odot r_i, \tag{6}$$

where $q_0 = h_u^{'}$, $r_i \in R^d$ means the representation vector of the edge between adjacent vertices n_i and n_{i-1}. Further, the representation of the metapath instance $P(v,u)$ is obtained as

$$h_{P(v,u)} = \frac{q_M}{M+1}. \tag{7}$$

Then, based on the attention mechanism, HKN calculates the attention weight of each P-based instance representation $h_{P(v,u)}$:

$$e_{vu}^P = LEAKYRELU\left(a_P^T \cdot \left[h_v^{'} \parallel h_{P(v,u)}\right]\right), \tag{8}$$

where $a_P \in R^{2d}$ is a learnable parameter vector. $\parallel$ means concat operation. e_{vu}^P is then normalized to the weight coefficient α_{vu}^P:

$$\alpha_{vu}^P = SOFTMAX_{u \in N_v^P}(e_{vu}^P). \tag{9}$$

Finally, the representation h_v^P of the target event vertex v on the meta-path P is obtained:

$$h_v^P = RELU(\sum_{u \in N_v^P} \alpha_{vu}^P \cdot h_{P(v,u)}). \tag{10}$$

Next HKN aggregates the representations of the target event vertex v in different meta paths. Firstly, HKN calculates the average of all event vertex representations on the meta path P_i to get the representation:

$$h_{P_i} = \frac{1}{\mid V_{eve} \mid} \sum_{v \in V_{eve}} tanh(W \cdot h_v^{P_i} + b), \tag{11}$$

where $W \in R^{d_N \times d}$ and $b \in R^{d_N}$ are learnable parameters. $\mid V_{eve} \mid$ represents the number of target event vertexs. Then, HKN employs the attention mechanism to calculate the normalized weight of P_i:

$$e_{P_i} = a_O^T \cdot h_{P_i}, \tag{12}$$

$$\beta_{P_i} = \underset{P_i}{SOFTMAX}(e_{P_i}), \tag{13}$$

where $a_O \in R^{d_N}$ is a learnable parameter. Finally, the representation of the target event vertex v is calculated as:

$$h_v = \sum_{P_i} \beta_{P_i} \cdot h_v^{P_i}. \tag{14}$$

Similar to the event knowledge network embedding, after the representation of each event vertex is obtained, HKN employed LSTM to encode the sequence information of the event chain. The output is used to be the representation of heterogeneous knowledge graph:

$$Z_{het} = LSTM(\{h_{v_i} \mid v_i \in V_{eve}\}). \tag{15}$$

3.4 Multi-level Knowledge Fusion

The representation fusion module fuses text representation, entity knowledge network representation, event knowledge network representation, and heterogeneous knowledge network representation. Because various levels of information are encoded in different ways, the features of different semantic spaces can't be forcibly fused to avoid introducing unnecessary noise.Through layer normalization, HKN maps the representation vectors of different levels of information into a unified space, and realizes the fusion representation of multi-level information of legal documents. The detailed fusion process can be expressed as follows:

$$Z = Z^*_{ent} \parallel Z^*_{eve} \parallel Z^*_{het} \parallel Z^*_{D}, \tag{16}$$

$$Z^*_i = tanh(W_i \cdot Z_i + b_i), Z_i \in \{Z_{ent}, Z_{eve}, Z_{het}, Z_D\}, \tag{17}$$

where $W \in R^{d' \times d_i}$ and $b \in R^{d'}$ are learnable parameters.

3.5 Training

HKN calculates the loss function based on the pairwise mode in the model training phase and optimizes the ranking problem by similarity. Each training dataset contains legal text triples (A, B, C), where A is the Query legal text. Each legal query document will match a positive document (similar document) B and a negative document (no similar instruments) C. The specific process is as follows:

$$\mathcal{L} = \sum_{(A,B,C) \in T_{batch}} [\gamma - Sim(Z_A, Z_B) + Sim(Z_A, Z_C)]_+ \tag{18}$$

where γ is margin, and Sim(·) represents the similarity calculation function between feature vectors.

4 Experiment

4.1 Datasets and Evaluation Indicators

We evaluate our models on three similar legal case datasets in Chinese, CAIL2019-SCM, LeCaRD2K, and LDC2K to demonstrate the effectiveness of incorporating event-level knowledge into text representations.

CAIL2019SCM is derived from the similar case matching task of the Challenge of AI in Law held by Xiao et al.[17]. It contains 8964 similar case triples, and the triples consist of one inquiry legal text and two candidate legal texts.

LeCaRD2K: Ma et al.[11] released a Chinese legal case retrieval dataset LeCaRD. It contains 107 query legal documents, each query document has 100 candidate legal documents, of which only a small number of candidate documents are similar to the query documents. Based on the LeCaRD dataset, we manually constructed 2000 high-quality similar case triples by combining expert opinions and evaluation criteria of legal similarity.

LDC2K: We collected 271,295 legal documents in seven common charges from China Judgments Online, and conducted preliminary screening and structured processing of each legal document. In addition, we manually constructed 2,000 high-quality similar case triples based on key elements of party information and judgment results, combined with document TF-IDF similarity and legal expert opinions.

In order to verify the performance of the model in similar legal case retrieval, we adopt similar legal case matching(SCM) and similar legal case ranking(SCR) as the evaluation tasks of model performance.

SCM: For similar case triples (A, B, C), A represents the query text, and B, C represent candidate texts. The purpose of the SCM task is to infer candidate text that is more similar to the query text. It is worth noting that we did not use evaluation metrics such as recall or F1 because the number of positive samples in the test dataset is much lower than the actual value.

SCR: Unlike the SCM task, the SCR task is closer to the actual application scenario, and this evaluation strategy can better reflect the real performance of the model in practice. For a pair of similar cases (A, B), the evaluation metric Mean Rank(MR) refers to the similarity score ranking of ground-truth B among all candidate texts.

4.2 Baseline

In this subsection, we select some widely used methods in text matching tasks as baselines.To better understand our proposed methods, we also compare within the performance of our methods using different configurations. Details of the baselines are listed below:

BERT[3] is a pre-trained language model with bidirectionally encoded representations. In particular, we select the BERT trained on the legal corpus released by Zhong et al.[19].

RoBERTa[9] improves the pre-training task designed on the basis of Bert, and is trained with larger batches on a larger corpus.

Lawformer[16] is a long-text pre-trained model based on Longformer to introduce legal domain knowledge.

AlphaCourt is the champion solution of the SCM task of 2019 Challenge of AI in Law, but the main body of the model integrates five sub-modules based on pre-trained language models, making the training process very time-consuming.

LFESM[4] is a legal feature enhanced semantic matching network that utilizes a Siamese network framework to capture subtle feature differences between query text and candidate text.

HetGNN[18] constructs a legal document-legal entity heterogeneous graph, and obtains the semantic representation vector of a legal document that incorporates external legal knowledge through graph encoding.

TextCNN[5] utilizes a convolutional neural network to capture local semantic features of text.

TextRNN[8] utilizes a recurrent neural network to model text sequence features and mine the underlying semantic information in the text.

4.3 Model Configuration

Considering the obvious differences in parameter distribution and model structure of multiple sub-modules of HKN, we set separate hyperparameters and optimization strategies for each sub-module.For the pre-trained language model of the text representation module, the text embedding vector remains 768-dimensional and the learning rate is 2e-6. The event knowledge network, entity knowledge network and knowledge-aware heterogeneous network remains a representation vector of 256-dimensional, and the learning rate is set to 5e-5. Furthermore, the triplet loss in the fusion stage is computed with a margin of 1. For those hyperparameters not mentioned by the model, keep consistent with the default parameters provided by Pytorch.

4.4 Evaluation Results

We examined the ability of HKN as well as other benchmarks to retrieve similar cases. Table 1 shows the performance of each model on the SCM task and the SCR task. Overall, we observed that HKN based on heterogeneous knowledge-aware networks has better performance on different datasets and evaluation metrics.

Model	**CAIL2019SCM**		**LeCaRD2K**		**LDC2K**	
	Acc.	**MR**	**Acc.**	**MR**	**Acc.**	**MR**
Bert	70.29	211.37	76.03	59.66	74.86	49.35
RoBERTa	71.79	196.63	76.21	50.14	75.84	43.97
LFESM	72.35	424.80	77.20	185.27	76.61	192.93
AlphaCourt	72.44	355.42	73.46	137.16	77.20	138.80
Lawformer	72.36	208.96	75.84	43.62	75.28	41.18
HetGNN	73.18	-	-	-	-	-
TextCNN	67.40	346.18	73.85	102.57	72.24	117.83
TextRNN	68.77	359.65	73.80	96.49	73.43	114.96
HKN	**75.70**	**195.55**	**81.30**	**37.85**	**77.90**	**35.27**

Table 1. Results on the SCM task and SCR task in different datasets.

In the SCM task, any model can easily identify the candidate cases that are more similar to the query case, even TextCNN which has the worst performance still achieves 67.43% on the CAIL2019-SCM dataset. This indicates the SCM task is not that complicated. First, the SCM task approximates binary classification. Second, the optimization method of the loss function is highly compatible with the SCM task. The performance of HetGNN, a heterogeneous graph neural network that fuses text and entity knowledge, is preceded only by HKN. This fully demonstrates that additional domain knowledge can help the model obtain richer semantic information. Besides, HKN that fuses legal entity knowledge and event knowledge outperforms models that only focus on entity knowledge. Entity knowledge and event knowledge belong to heterogeneous and different levels of knowledge information, both of which can help HKN learn stronger textual feature representations.

In the SCR task, we observed several different experimental results. The results present that LFESM and AlphaCourt are not suitable for handling this task. The sub-models of AlphaCourt are integrated roughly, and some noise may be introduced in the process. During the experiments, we found that HetGNN could not complete the SCR task, which indicated the unsupervised graph-based neural network had poor scalability and could not solve the OOV problem. In addition, the performance improvement of the complete HKN model is not obvious, which demonstrates that the SCR task is more challenging than the SCM task.

Horizontally comparing the performance results of each model on different datasets, we observe that LeCaRD2K and LDC2K are of comparable quality to CAIL2019SCM and have richer event information.HKN can achieve significant performance improvements on these two datasets.In contrast, AlphaCourt uses feature engineering for CAIL2019SCM, which does not have enough scalability and generalization.

4.5 Ablation Analysis

HKN contains different sub-modules that extract different levels of semantic information in the text. We performed a basic ablation test better to understand the impact of HKN heterogeneous content and submodules. HKN[L]:HKN with a specific submodule, where L=$\{Txt, Ent, Eve, Het\}$, Txt donated text module, Ent stands for entity knowledge network module, Eve stands for event knowledge network module, and Het stands for heterogeneous knowledge network module.

Method	**Acc.**	**MR**
HKN	**75.70**	195.55
HKN[Eve]	63.57	262.82
HKN[Ent]	68.49	359.77
HKN[Het]	71.86	201.88
HKN[$Eve + Ent$]	69.66	289.65
HKN[$Eve + Txt$]	73.46	208.86
HKN[$Ent + Txt$]	74.30	256.34
HKN[$Het + Txt$]	74.80	**173.92**

Table 2. Ablation analysis results of HKN on the CAIL2019SCM dataset.

The results of the ablation experiments are shown in Table 2, and we observe that any sub-module of HKN can help matching. Due to the lack of event information in the CAIL2019SCM dataset, the matching accuracy of HKN[Eve] only reached 63.57%. HKN[$Het+Txt$], HKN[$Ent+Txt$], and HKN[$Eve+Txt$] have apparent performance improvement, which shows that both entity knowledge and event knowledge can help the model obtain richer semantic features. The complete HKN matching accuracy reaches 75.40%, proving that entity knowledge and event knowledge belong to different levels of semantic information, and there is an implicit semantic correlation between them.

4.6 Analysis and Discussion

Crime analysis In a similar legal case retrieval task, the case information of different crimes is obviously distinct, thus it is easy to judge the similarity of cases. However, cases of the same crime may have similar essential circumstances or evidence information, making it difficult to distinguish the similarities between them. Moreover, in collecting datasets, we found that the legal documents of different charges have evident long tail distribution, which may significantly impact the experimental results. Therefore, we selected three representative crimes for analysis, namely, crime of larceny, crime of trafficking in drugs, and crime against homeland security. We selected the experimental data of these three crimes respectively, and the statistical results are shown in the Table 3. It illustrates that (1) the HKN model is superior to the baseline model in the performance of the crime of larceny due to the event information providing critical information to simplify the huge collection of candidate documents for LCR of high-frequency crimes. (2) There is a big gap between the performance of other baseline models and HKN in drug crimes. The reason may be that event information alleviates the misleading caused by high-frequency entity information. (3) Due to the lack of sufficient marked corpus, the overall performance of the experimental model in the crime of endangering national security is not as high as other crimes, but HKN still shows a satisfactory advantage. This is because the lack of label information leads to insufficient reference standards for low-frequency crimes, while HKN introduced multi-level knowledge information to ease this problem.

Model	Larceny		Trafficking in Drugs		Against Homeland Security	
	Acc.	MR	Acc.	MR	Acc.	MR
Bert	68.69	82.27	67.83	86.46	61.37	234.87
RoBERTa	69.98	77.36	68.33	811.38	62.46	213.46
LFESM	70.23	220.83	69.24	210.64	64.74	445.64
AlphaCourt	70.35	163.97	68.95	165.34	65.72	373.83
Lawformer	70.21	74.84	68.83	80.15	63.58	233.94
HKN	**72.60**	**58.86**	**71.20**	**43.59**	**67.47**	**129.74**

Table 3. Results on the SCM task and SCR task in different crimes.

Meta-path analysis Designing meta-path plays a key role in heterogeneous graph representation. In order to illustrate its effectiveness, we use a heat map to show the weight distribution of different neighborhood information. Especially, we added a meta-path $P\#3(EVE-ENT-ENT-ENT)$ to capture the third-order neighborhood information, and retrained the model. As shown in Fig.5, entity vertices $t_1 \sim t_9$ provide rich neighborhood information for event vertices

$E_1 \sim E_3$. The entity vertices t_6 and t_9 are the third-order neighbors of the event vertex E_2, and t_9 also acts as the third-order neighbor of the event vertex E_3. Through the heat diagram, it can be observed that (1) even under the same meta path, the weights of different instances are different, such as $(E_1, t_1)$and (E_1, t_2). This is because different neighbor vertices play different roles in the event. Thus, it's unreasonable to replace the attention mechanism with equal distribution weight. (2) The attention weight of the second-order neighborhood information is obviously lower than that of the first-order, while the third-order neighborhood information is hardly assigned to the weight, such as (E_2, t_4), $(E_2, t_5)$and(E_2, t_6), which shows that HKN has learned enough relevant entity information by encoding the first-order and second-order neighborhood. Time and space consumption can also be limited to an acceptable scope. In addition, we have also verified through experiments that considering the third-order neighborhood information and using the average instead of the attention mechanism to calculate the weight both damage the performance of HKN, which also verifies the rationality of our method.

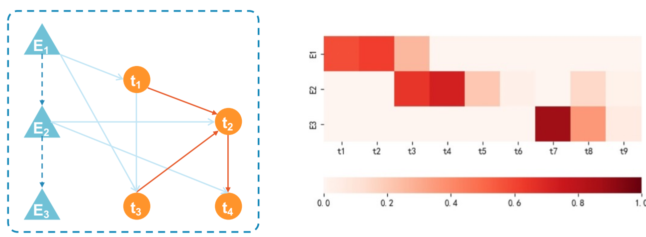

Fig. 5. The influence weight distribution of different neighborhood information in the meta-path.

5 Conclusion

This paper designs a multi-level heterogeneous knowledge-aware network that fuses text-level, entity-level, and event-level knowledge to achieve fine-grained similarity legal case retrieval in Chinese. HKN combines entity knowledge and event knowledge networks to construct a heterogeneous knowledge perception network. Experimental results on multiple legal similar case retrieval datasets show that HKN can utilize knowledge to effectively encode the latent semantic features of legal texts and achieve accurate similar legal case retrieval results.

Our future research will focus on two directions: (1) Modeling more complex logical connections between events to obtain more comprehensive event information; (2) Current similar legal case retrieval methods are over-reliant on labeled data, and we will explore the solutions in few-shot scenarios.

References

1. Bhattacharya, P., Ghosh, K., Pal, A.e.a.: Hier-spcnet: a legal statute hierarchy-based heterogeneous network for computing legal case document similar-

ity. In: Proceedings of the 43th ACM SIGIR. pp. 1657–1660 (2020). https://doi.org/10.1145/3397271.3401191
2. Bi, S., Huang, Y., Cheng, X.e.a.: Building chinese legal hybrid knowledge network. In: KSEM. pp. 628–639. Springer (2019). https://doi.org/10.1007/978-3-030-29551-656
3. Devlin, J., Chang, M.W., Lee, K.e.a.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Hong, Z., Zhou, Q., Zhang, R.e.a.: Legal feature enhanced semantic matching network for similar case matching. In: IJCNN. pp. 1–8. IEEE (2020). https://doi.org/10.1109/IJCNN48605.2020.9207528
5. Kim, Y.: Convolutional neural networks for sentence classification. Proceedings of the 2014 EMNLP pp. 1746–1751 (08 2014). https://doi.org/10.3115/v1/D14-1181
6. Kumar, S., Reddy, P.K., Reddy, V.B.e.a.: Finding similar legal judgements under common law system. In: International workshop on databases in networked information systems. pp. 103–116. Springer (2013). https://doi.org/10.1007/978-3-642-37134-99
7. Li, H., Lu, J., Le, Y.e.a.: Iacn: Interactive attention capsule network for similar case matching. Intelligent Data Analysis **26**(2), 525–541 (2022)
8. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. In: Proceedings of the 25th IJCAI. pp. 2873–2879 (2016)
9. Liu, Y., Ott, M., Goyal, N.e.a.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
10. Lu, Q., Conrad, J.G.: Bringing order to legal documents-an issue-based recommendation system via cluster association. In: International Conference on Knowledge Engineering and Ontology Development. vol. 2, pp. 76–88. SCITEPRESS (2012). https://doi.org/10.13140/2.1.1594.0486
11. Ma, Y., Shao, Y., Wu, Y.e.a.: Lecard: A legal case retrieval dataset for chinese law system. In: Proceedings of the 44th ACM SIGIR. pp. 2342–2348 (2021). https://doi.org/10.1145/3404835.3463250
12. Shao, Y., Mao, J., Liu, Y.e.a.: Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In: Proceedings of the 29th IJCAI. pp. 3501–3507 (2020). https://doi.org/10.24963/ijcai.2020/480
13. Shen, S., Qi, G., Li, Z.e.a.: Hierarchical chinese legal event extraction via pedal attention mechanism
14. Trivedi, A., Trivedi, A., Varshney, S.e.a.: Similarity analysis of legal documents: A survey. In: ICT Analysis and Applications, pp. 497–506. Springer (2021). https://doi.org/10.1007/978-981-15-8354-449
15. Winkels, R., Boer, A., Vredebregt, B.e.a.: Towards a legal recommender system. In: Proceedings of the 27th ICLKIS (2014). https://doi.org/10.3233/978-1-61499-468-8-169
16. Xiao, C., Hu, X., Liu, Z.e.a.: Lawformer: A pre-trained language model for chinese legal long documents. AI Open **2**, 79–84 (2021)
17. Xiao, C., Zhong, H., Guo, Z.e.a.: Cail2019-scm: A dataset of similar case matching in legal domain. arXiv preprint arXiv:1911.08962 (2019)
18. Zhang, C., Song, D., Huang, C.e.a.: Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD. pp. 793–803 (2019). https://doi.org/10.1145/3292500.3330961
19. Zhong, H., Zhang, Z., Liu, Z.e.a.: Open chinese language pre-trained model zoo. Tech. rep. (2019), `https://github.com/thunlp/openclap`