IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 June 2024, accepted 3 August 2024, date of publication 7 August 2024, date of current version 19 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439775

RESEARCH ARTICLE

Deep Text Understanding Model for Similar Case Matching

JIE XIONG AND YIHUI QIU
School of Economics and Management, Xiamen University of Technology, Xiamen 361024, China
Corresponding author: Yihui Qiu (qiuyihui@xmut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China, Youth Program, under Grant 7180040248 and in part by the Natural Science Foundation of Fujian Province of China, under Grant 2022J011261.

ABSTRACT Natural Language Processing (NLP) technology is rapidly evolving, and various large language models have been widely applied in Legal Artificial Intelligence (AI). However, low accuracy in Similar Case Matching (SCM) persists in the most popular case recommendation systems. It hinders the practical application of case recommendations in Legal Judgment Prediction (LJP). Developing effective methods to extract features from long texts and improve the accuracy of SCM is an urgent matter that requires attention. Therefore, the paper proposes a SCM method based on deep text comprehension. A fine-tuned BERT model is used to extract text information, and a combination of global attention and self-attention is employed to represent the features of long texts deeply. A dual-channel similar text-matching approach is used after candidate texts are pre-encoded to reduce the SCM model's training time and improve accuracy. Experiments on the China AI and Law (CAIL) competition dataset show that the proposed method achieves the highest accuracy in SCM compared to the recent methods.

INDEX TERMS Similar case matching, text mining, similarity analysis, attention mechanism, feature extraction.

I. INTRODUCTION

Currently, Legal Artificial Intelligence (AI) is undergoing a profound reform, driven by big data, artificial intelligence, and information technologies. The application of large Natural Language Processing (NLP) models effectively enhances the deep analysis of legal texts. Rapidly recommending similar cases based on the characteristics of case documents and facts provides crucial support for intelligent case assistance systems used by judicial authorities. It significantly assists judges and mediators by reducing the workload of case analysis, facilitating decision-making, and increasing work efficiency. At the same time, it aids in minimizing discrepancies in judgments across similar cases, thereby promoting the efficient and equitable dispensation of justice. However, the current accuracy of Similar Case Matching (SCM) is hindered by factors such as the length and professionalism of legal documents, which limit the auxiliary effect of Legal Judgment Prediction (LJP) or judicial mediation.

A similar case shares similarities with the pending case regarding basic facts, points of dispute, legal issues, etc., and has been adjudicated and finalized by a people's court [1]. SCM involves the intelligent analysis of judicial documents using techniques such as deep learning to locate similar cases in a case database. Research on SCM increases the efficiency of matching large volumes of legal texts to improve the accuracy of LJP. In the field of Legal AI, there are three common methods of SCM: keyword-based case retrieval, label-based case retrieval, and fundamental NLP text-matching algorithms [2], [3], [4]. Existing methods focus on computing the similarity between two texts. Distinguished from basic word frequency statistical methods, scholars attempt to extract features from legal documents or improve them by converting the documents into embeddings using vector space models, and then assess the similarity between the extracted features or embeddings [5], [6]. These methods are highly efficient, but they only compare extracted features without incorporating full-text information or capturing contextual and local features. Xu et al. [7] proposed a context-aware similar case matching and recommendation model (CASCMR), which has significantly improved text feature extraction relative to traditional methods. However, there is still considerable room for improvement in accuracy and applicability to long legal texts. It faces challenges

The associate editor coordinating the review of this manuscript and approving it for publication was Fu Lee Wang.

such as insufficient representation of long legal texts and low accuracy, which limits its effectiveness in assisting LJP. Thus, this paper proposes an SCM method based on comprehensive text understanding. The method aims to enhance accuracy through supervised comparison of multiple texts. The ultimate goal is to achieve end-to-end similar case recommendations.

The main contributions are as follows:

1. In the field of Legal AI, an improved SCM model based on comprehensive text understanding has been proposed. The model effectively addresses the issue of low accuracy in SCM.
2. Effective representation of long text features is achieved by employing global attention and self-attention.
3. This model can also be used for other tasks that require matching both long texts and short texts.

II. RELATED WORK

SCM, as an important intersection of AI and judicial, has received widespread attention in recent years from both academia and industry [8]. The core idea is to automatically query and recommend court judgments from databases that are most similar to the current case in terms of factual circumstances and points of dispute. It is particularly useful in situations where guiding cases are scarce or identifying similar cases is challenging. The technology not only improves judicial fairness and accuracy but also saves significant human and material resources [9]. Deep mining and analysis of textual information from numerous cases facilitate the extraction of key features and patterns. These extracted features and patterns are subsequently used to construct mathematical models that measure case similarity. Although the three methods previously mentioned in the field of Legal AI have significantly improved operational efficiency, they fail to incorporate contextual and local key information from texts, resulting in insufficient accuracy. SCM is typically implemented using technologies from various domains, such as NLP, information extraction, and machine learning. These include Natural Language Inference (NLI), Information Retrieval (IR), and Question Answering (QA) [10], [11]. NLI seeks to determine whether a premise can be inferred from a hypothesis, while both case recommendation and NLI focus on text similarity. In NLI, texts may convey both related and identical meanings, whereas in SCM, cases meanings are distinct due to variations in parties and facts. Therefore, integrating methods from NLI into SCM is a significant challenge in this domain. SCM technology has been employed in certain judicial practices, such as the case recommend function in Legal AI systems, yet challenges remain [12]. This paper introduces the main tasks involved in SCM, such as text similarity matching and long text feature extraction.

A. TEXT SIMILARITY MATCHING

To resolve the issue of text similarity in SCM, various methods have been developed and applied. Effective methods can significantly improve the accuracy of SCM. Existing text similarity matching methods include keyword-based methods, syntax, and text structure-based methods, deep learning-based methods, and multidimensional perspective-based methods [6]. Keyword-based matching methods are the earliest approaches, including methods such as word frequency statistics and word graph networks. The core idea involves representing texts with keywords and calculating their similarity based on the weights of these keywords. However, keyword-based matching methods ignore text structure which can result in text matching errors. Methods based on syntax, semantics, and knowledge structures split text to effectively solve these issues [13]. In NLP, there are two types of deep learning frameworks for matching text similarity. One type is based on Siamese Networks, which encode sentence pairs separately using the same encoder and then compute similarity using features [14]. For example, Cao and Zhao [15] proposed a Siamese network for text similarity computation that uses a multi-head self-attention. The Siamese model performs well when using a bidirectional Gated Recurrent Unit (GRU) as the basis and combining with multi-head self-attention to extract deep semantic information from long texts. However, such methods have limitations in deep semantic interaction, which can lead to the loss of important information. Additionally, they require large amounts of labeled data and incur high computational costs. To address these shortcomings, scholars have proposed a new type of matching aggregation network that incorporates more interactions at the word and phrase levels. Chen et al. [16] proposed an advanced model known as Enhanced LSTM for NLI, which captures more local information between text pairs before performing global comparisons. It calculates the similarity between two cases more effectively. Other scholars have conducted additional research on this [10]. Many scholars have used this approach to process text data by mapping it to higher-dimensional spaces to extract features. As a result, they have achieved significant outcomes in text similarity matching. To comprehensively and accurately compare the similarities between the two texts, scholars have begun to study text matching from multidimensional perspectives. The approach effectively tackles the issues of feature sparsity and representation discrepancy in texts. However, after mapping texts to high-dimensional spaces, distance computation methods for determining text similarity become meaningless. Therefore, it is crucial to determine how to effectively reduce dimensionality by learning deep feature representations and selecting appropriate text similarity calculation methods. It is essential for improving the accuracy and effectiveness of text similarity.

B. FEATURE EXTRACTION

Feature extraction for long texts as a key technology in the field of NLP has made significant progress, but there are still challenges such as computational efficiency and feature extraction performance that must be addressed urgently. Long texts are rich in semantic information, so accurately

understanding their deep meaning is critical for feature extraction. There are various methods for extracting features from long texts. The most common methods are statistical approaches, deep learning techniques, and hybrid methods. The processing methods mainly include filtering, fusion, mapping, and clustering [17], [18], [19], [20], [21]. Currently, the most widely used method is deep learning-based feature extraction for long texts. Deep learning combines low-level features to form more abstract, higher-level attributes. It enables the rapid generation of new effective features from training data. In 2013, Mikolov et al. [22] proposed the Word2vec model, which significantly advanced feature extraction technology. The model transforms words in text space to vector space and represents text with low-dimensional vectors. It effectively resolves the issue of exploding text vector dimensions while also improving the accuracy of semantic expression in the original text. For example, Deng et al. [23] integrated Word2vec, Doc2vec, and Term Frequency-Inverse Document Frequency (TF-IDF) to compute case similarity, which improved the accuracy of SCM. Although such methods have produced satisfactory results, they have limitations such as incomplete keyword vectors and the absence of sentence vectors. In 2018, Google's AI Language team [5] proposed the BERT model, which trains on a vast amount of general corpus using a 12-layer Transformer and context from all encoding layers to train deep bidirectional representations. It provides the benefits of parallel computation while also addressing the polysemy problem. Subsequently, Tsinghua University's OpenClaP [24] trained BERT models specifically for feature extraction from Chinese civil and criminal legal cases. Hu et al. [25] proposed an SCM method based on legal facts (BERT-LF), which combines themes with legal entity facts to improve the applicability of document vectors to legal scenarios. The model encodes contextual semantic information and solves the problem of long text feature extraction by employing a BERT-based paragraph aggregation technique. Fang [26] proposed three data augmentation methods: truncation, dual loss, and prompting. These methods aim to achieve more effective learning in a simple and efficient manner. These models excel at analyzing word relationships and subtle contextual differences in legal texts through sequential modeling, thereby enhancing accuracy in complex SCM tasks. However, they are constrained by training costs and accuracy. For long text processing, researchers have made improvements based on BERT from aspects such as gating mechanisms, attention, hierarchical guidance, and Graph Neural Network (GNN) models [27], [28], [29], [30]. These advancements have effectively enhanced the utilization of long text features. The methods described above have produced significant results in the field of SCM, but they still have limitations, such as the inability to extract contextual and local key information from texts effectively. Moreover, there is a need to improve the accuracy of SCM, which reduces the effectiveness of supporting LJP. Using deep learning methods for feature extraction from long texts in SCM has benefits, but it also has drawbacks. These issues include problems such as gradient disappearance and explosion during long text information processing, as well as information loss during long text segmentation. Additionally, there are challenges in accurately and comprehensively representing the information contained in the text. These problems must be solved.

In conclusion, the deep development of Legal AI faces several key technical challenges, which must be addressed to advance it. First, the effective extraction of features from legal texts is urgently needed. Legal texts typically have complex structures and rich semantics, so effectively extracting key features from long texts is critical for improving the speed and accuracy of information retrieval and processing. Second, a thorough understanding of legal expertise is required, as proper interpretation of legal texts is critical to ensuring the accuracy and fairness of Legal AI systems. Additionally, further research and optimization are necessary for processing text information and text dimensionality reduction. Finally, using appropriate text similarity measurement methods can effectively avoid the failure of high-dimensional space distance computations while also improving the accuracy and effectiveness of text similarity computations. Therefore, this paper proposes a SCM method based on deep contextual understanding. Specifically, by building on pre-trained language models for text feature extraction and combining global attention and self-attention, a deep understanding of long legal cases can be improved, with the goal of improving the accuracy and efficiency of SCM. It supports the further development of Legal AI and text-data mining.

经审理查明:2018年11月23日开始,被告人XXX乘坐由XXX和驾驶的车牌为云D 的白色现代轿车,车内放置了一套XXX购买的无线电发射设备。 由被告人XXX在汽车上操作设备,在南宁市市区中心一带发送诈骗、非法网站链接等违法短信。 2018年11月25日18时许,二人驾驶的车辆行至南宁市朝阳人民路口南华大厦前时被***查获.....

Upon investigation, it was found that starting from November 23, 2018, the defendant XXX was traveling in a white Hyundai sedan with the license plate Yun D, driven by XXX. Inside the car, a set of radio transmission equipment purchased by XXX was placed. The defendant XXX operated the equipment in the car and sent fraudulent and illegal website links via SMS in the central area of Nanning City..... **Query Text**

经审理查明:被告人XXX在天津市一网络平台找工作时,认识“张哥”(身份暂未查明),后其从“张哥”处获取一套发射短信的设备,其按照“张哥”的电话指示操作该设备发送违法信息,“张哥”承诺按照每月六千元左右的工资支付给被告人闫福宏报酬。 2018年11月7日16时许,在阜阳市颍州区格林豪泰酒店8518房间内,被告人闫福宏利用伪基站设备发送违法信息时被***抓获。 经安徽省阜阳无线电监测站鉴定,XXX使用发短信的设备为“伪基站”,被告人XXX使用该设备在天津、阜阳等地共发送510325条违法短信.....

Upon investigation, it was found that the defendant XXX, while looking for a job on an online platform in Tianjin, met "Brother Zhang" (identity yet to be determined). Subsequently, the defendant obtained a set of SMS transmitting equipment from "Brother Zhang" and operated this equipment to send illegal messages according to "Brother Zhang's" instructions. "Brother Zhang" promised to pay the defendant XXX a monthly salary of approximately 6,000 yuan..... **Candidate Text 1**

经审理查明:2013年至2017年11月,被告人XXXXXX以非法牟利为目的,在网络上非法出售能够破坏计算机信息系统,修改增加他人身份信息,逃避警方网络监管的“免刷身份证软件”,逐渐形成了以被告人***、XXX、XXX某明(已判刑)等人为销售渠道,面向全国的销售网络.....

Upon investigation, it was found that from 2013 to November 2017, the defendant XXXXXX, with the intent of illegal profit, illegally sold software on the internet that could damage computer information systems, modify and add others' identity information, and evade police network supervision, known as "ID Card Exemption Software." This led to the formation of a nationwide sales network with the defendant ***, XXX, XXX, and XXX (already sentenced) as sales channels..... **Candidate Text 2**

FIGURE 1. Here are three case document examples extracted from the CAIL2022 SCM data set. Query text is the document to be queried, candidate text1 is a document from the candidates that is similar to the query text, and candidate text2 is a document from the candidates that is dissimilar to the query text.

III. MODEL

The paper primarily investigates the problem of SCM in Legal AI. Based on the current case's description, type, and relevant legal provisions, it recommends similar cases from a case database. In the process of pre-training, given a set of triplets $\{t, d_{txt}, d_{cand}\}$, where d_{txt} represents the query text, d_{cand} is the candidate text, and $t \in \{0, 1\}$ represents the label.

If $t = 1, d_{cand}$ and d_{txt} are similar cases; if $t = 0$, these two texts are dissimilar cases. The input query text outputs the feature vector $X_{txt} \in R^n$ for the current query case and the feature vector $Y_{cand} \in R^m$ for the candidate case, where n and m are the number of features. The similarity between the query text and a set of $D = (Y_1, Y_2, \ldots, Y_a)$ candidate cases is measured as $sim(X_{txt}, Y_{cant}), Y_{cant} \in D$, and the top similar cases are recommended after ranking. Equation (1) represents the mathematical description. The model is trained using labeled data, which can also be used for unsupervised data labeling, as shown in Figure 2.

$$\mathrm{Top_n} = \underset{j=1,\ldots,n}{\overset{j}{\mathrm{arrange}}} \left(\underset{i=1,\ldots,m}{\overset{i}{sim}} (\mathrm{X}_{txt}, \mathrm{Y}_{\mathrm{cant}}) \right) \tag{1}$$

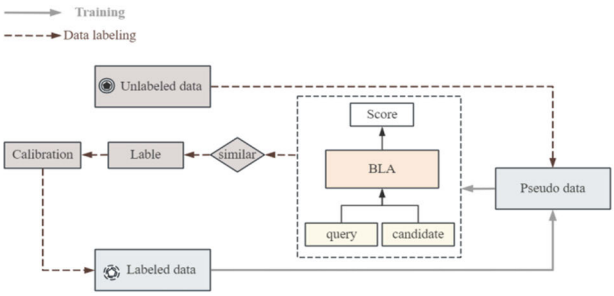

FIGURE 2. Overall logic diagram.

A. PRE-TRAINING AND INPUT

The pre-training data for this study is divided into two groups. The first group contains long text data extracted from criminal data provided for an SCM task in the China AI and Law Competition (CAIL) in 2019, which includes 6,018 labeled training data, 1,012 validation data, and 1,012 test data. The second group consists of civil data extracted from the CAIL2022 competition's interpretable SCM task, which includes 15,306 labeled training data, 1,000 validation data, and 1,000 test data. During the pre-training process, data is input using a concatenation method of [label, input], where both the label and input items are enclosed by special markers [t], similar to the method described in the paper by Devlin et al. The input is a single sentence, and the label is the following sentence in the text. Each input token is composed of three embeddings: token embedding, position embedding, and segment embedding. The segment for the input token is 0, while the segment for the label token is 1. During training, the ratio of positive to negative samples is 1:1, but during testing, it is 1:9.

During pre-training, the data takes the form of a set of legal text data, denoted as t, d_{txt}, d_{cand}. In the pre-training strategy, the legal BERT model used is pre-trained by Tsinghua OpenClaP. The training strategy alternates between masked language modeling and next sentence prediction tasks, which is similar to the method described by Humeau et al [31]. The Adam optimizer is used with a learning rate of 5e-3, $\beta_1 = 0.9, \beta_2 = 0.98$, no L_2 weight decay, linear learning rate warmup, and inverse square root learning rate decay. A dropout rate of 0.1 is applied to all layers. The training batch size is 16, and further fine-tuning is performed based on downstream tasks.

B. SCM MODEL

The objective of SCM is to find the most similar case texts from the legal text corpus D based on a given query text d_{txt}. The architecture of the proposed SCM model based on deep text understanding is composed of three layers: input layer, encoding layer, and output layer. The input layer uses Bert word embedding to map the text to its corresponding vector. The encoding layer extracts both local information features and deep understanding features across the entire text. The model is trained on a dataset composed of triplets, each containing a label, a query, and a candidate. To train the model to distinguish the relevance of two texts, the candidate includes both positive samples (cases similar to the query) and negative samples (cases not similar to the query). When processing text data, the input layer first converts the vocabulary into high-dimensional word embeddings that contain semantic and contextual information about the words. The encoding layer then processes these word embeddings, using the attention to encode local features within the text. The encoding layer not only captures the local features of the query but also integrates a comprehensive understanding of the candidate. It ensures the model can effectively capture the intricate interactions between the query and the candidate, thereby enhancing its ability to accurately determine their similarity. Finally, the model's output is evaluated and corrected using a loss function. The parameters are adjusted based on the gradient of the loss function to optimize classification performance and achieve precise distinction between similar and dissimilar cases. Figure 3 shows the architecture of a SCM model based on deep text understanding.

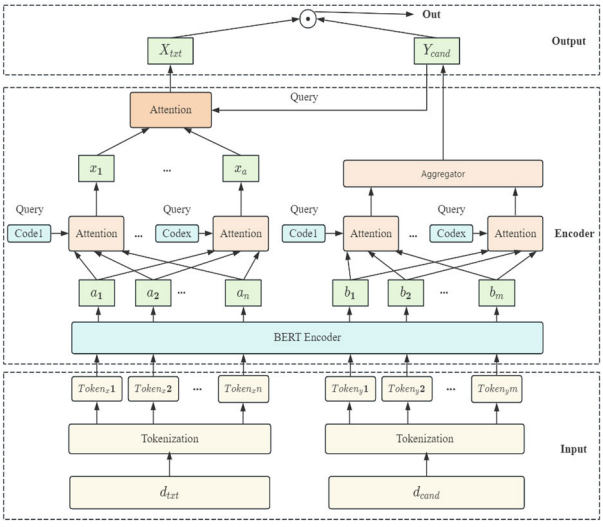

FIGURE 3. Schematic diagram of the architecture of the SCM model based on global deep understanding.

C. ENCODING

BERT is a bidirectional encoder, which means it can take into account both preceding and following words, allowing it to capture and comprehend the contextual information of the entire sentence more accurately than other encoders. BERT, which is trained using masked language modeling and next sentence prediction tasks, is fine-tuned based on legal text characteristics to better adapt to downstream tasks. In this paper, a trained legal BERT with 12 hidden layers, a hidden size of 768, and 12 attention heads is used to encode the representation of words, segments, and positions in legal texts. The input encoding of BERT is illustrated in Figure 4. Pretraining is conducted through alternating training in masked language modeling and next sentence prediction tasks.

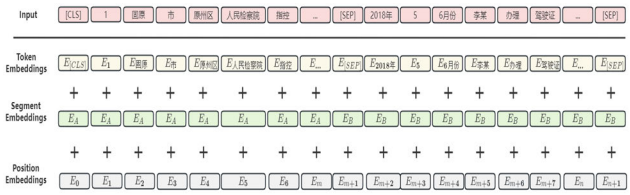

FIGURE 4. The schematic diagram of input encoding.

The query text and candidate text are tokenized with the Word Piece tokenizer and then fed into BERT. After encoding, they each obtain their own vectors, as shown in Equations (2) and (3).

$$a_{\omega}^{\mathrm{n}} = BERT(\mathrm{d}_{txt}, n), a_{\omega}^{\mathrm{n}} \in R^{1\times k}, \forall \mathrm{n} \in \{1, \ldots, N_{\mathrm{txt}}\} \quad (2)$$

$$\mathrm{b}_{\theta}^{m} = BERT(\mathrm{d}_{cand}, m), \mathrm{b}_{\theta}^{m} \in R^{1\times l}, \forall \mathrm{m} \in \{1, \ldots, M_{cand}\} \quad (3)$$

where a_{ω}^{n} represents the word embedding of the nth word in query text ω, N_{txt} represents the number of words in text txt, and k represents the embedding dimension. b_{θ}^{m} represents the word embedding vector of the mth word in candidate text θ, N_{cand} denotes the number of words in text $cand$, and l represents the embedding dimension. The query and candidate text obtain text word embedding vectors through the same method using BERT's word embedding model, as shown in Equations (4) and (5).

$$\mathrm{A}_{\omega} = [a_{\omega}^{1}, \ldots, a_{\omega}^{n}] \in R^{n\times k} \quad (4)$$

$$\mathrm{B}_{\theta} = [b_{\theta}^{1}, \ldots, b_{\theta}^{m}] \in R^{m\times l} \quad (5)$$

D. KEY INFORMATION EXTRACTION

Due to the complexity and professionalism of legal long texts, traditional feature extraction methods are insufficient to capture all the important and detailed information. Therefore, new approaches are necessary to confront these challenges. Additionally, BERT has a maximum positional encoding limit when embedding long texts. While segmenting long texts is an effective approach, it can still result in the loss of text information and contextual coherence. Therefore, this paper introduces an attention that extracts key information from the text itself, improving expertise understanding. A schematic diagram of the local key information extraction method is shown in Figure 5.

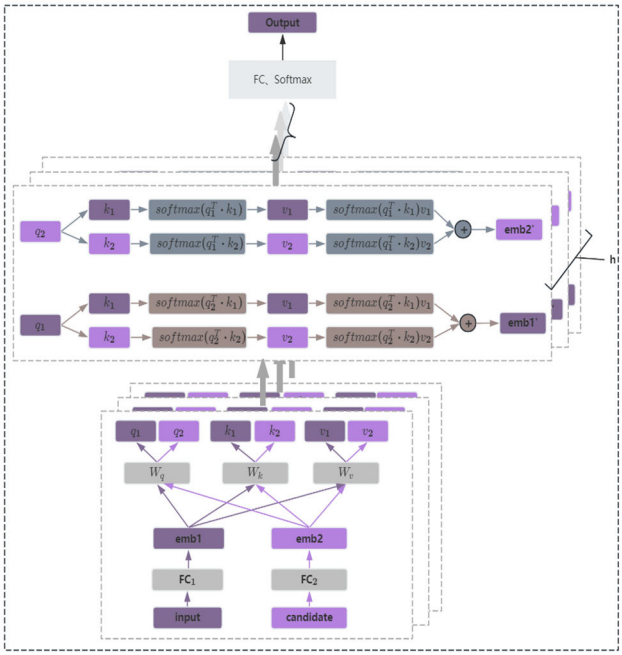

FIGURE 5. Schematic diagram of key information extraction.

The initially encoded text vectors of the query and candidate items are processed to optimize the representation of key information. The computation formulas are shown in Equations (6) to (9)

$$Attention(Q_a, K_a, V_a) = \mathrm{soft\,max}\left(\frac{\mathrm{Q}_a\mathrm{K}_a^{T}}{\sqrt{d_k}}\right) \quad (6)$$

$$x_{\omega} = soft\ \mathrm{max}\left(\frac{\left(a_{\omega}W_{Q}^{a}\right)\left(a_{\omega}W_{K}^{a}\right)^{T}}{\sqrt{d_k}}\right)\cdot\left(a_{\omega}W_{v}^{a}\right) \quad (7)$$

$$Attention(Q_{\mathrm{b}}, K_b, V_b) = \mathrm{soft\,max}\left(\frac{\mathrm{Q}_b\mathrm{K}_b^{T}}{\sqrt{d_l}}\right) \quad (8)$$

$$\mathrm{y}_{\theta}' = soft\ \mathrm{max}\left(\frac{\left(\mathrm{b}_{\theta}W_{Q}^{\mathrm{b}}\right)\left(\mathrm{b}_{\theta}W_{K}^{b}\right)^{T}}{\sqrt{d_l}}\right)\cdot\left(\mathrm{b}_{\theta}W_{v}^{b}\right) \quad (9)$$

where Q, K, and V represent the Query, Key, and Value matrices, respectively, and d denotes the variance. The subscripts a and b represent the query and candidate items, respectively. These are obtained by multiplying the input vector a by the corresponding weight matrix W. The weight matrices are randomly initialized and adjusted during the training process. Then, the candidate text is aggregated into a single vector, as shown in Equation (10).

$$\mathrm{Y}_{\mathrm{cand}} = red\left(y_{\theta}^{\mathrm{m}}\right) \quad (10)$$

where *red*() is a function that selects the first output of the encoding layer, simplifying the vector sequence into a single vector.

E. KEY INFORMATION EXTRACTION FROM QUERY BASED ON CANDIDATE ITEMS

To gain a better understanding of the text and obtain more concise representations of the query text, the query items' global context features are processed. The candidate features are used as queries to handle the relationship between the query and the candidate items, as shown in Equations (11) and (12).

$$\mathrm{X_{txt}} = \sum_{\mathrm{j}} w_j^{x\omega} X_{txt}^{i} \tag{11}$$

$$(\mathrm{w}_1, \ldots, \mathrm{w}_n) = soft\max(\mathrm{y_{cand}} \cdot x_{txt}^{1}, \ldots, y_{cand} \cdot x_{txt}^{\omega}) \tag{12}$$

Finally, the obtained query and candidate item features are dot-products to compute the similarity score between them, recommending the Top_n cases with the highest scores, as shown in Equation (13).

$$\mathrm{S(txt, cand)} = \mathrm{X_{txt}} \cdot Y_{cand} \tag{13}$$

The pseudo-code for the main computation process is shown in Algorithm 1.

Algorithm 1 Iterative Training for SCM

Input: Query text, Candidate text
function EncodeAndRetrieve ()
Using Bert for word embedding encoding, input the query text and candidate text separately into the BERT model for word embedding encoding.
1. Encode embeddings
   q_embedding ← BERT(query_text)
   cand_embedding ← BERT(candidate_text)
   # Using the attention mechanism to extract key information, compute the attention weight matrix between the query text and the candidate text.
2. Extract key information using the attention mechanism
   q_atten ← Attention(q_embedding, q_embedding,q_embedding)
   cand_atten ← Attention(cand_embedding,cand_embedding, cand_embedding)
   q_feature ← Reduce(q_embedding × q_attention)
   cand_feature ← Reduce(cand_embedding × cand_attention)

Aggregate the encoded features of the query text and candidate text based on the attention weights.
3. Extract key information of query using candidate
   q_feature ← cand_feature × q_embedding
Calculate the similarity score between the query text and the candidate text.
4. Calculate similarity scores
   similarity_scores ← q_feature × cand_feature

Rank and recommend based on the similarity scores.
5. Sort and recommend
   sorted_candidates ← Sort(similarity_scores, descending=True)
   top_n_candidates ← sorted_candidates[:N]
   return top_n_candidates
end function0

IV. EXPERIMENTAL RESULTS AND ANALYSIS

To validate the effectiveness of the proposed method, it is applied to the CAIL2019 and CAIL2022 datasets for SCM.

A. DATASETS

To improve the model's efficiency in processing professional legal texts, we trained it on civil and criminal case datasets. The datasets are the CAIL 2019 SCM Dataset and the CAIL2022 Second Phase SCM Competition Dataset. These datasets respectively consist of a three-dimensional array $\{t, d_{txt}, d_{cand}\}$, containing 8,042 criminal cases published by the Supreme People's Court of China and 17,306 similar civil cases. Table 1 shows how the dataset is divided. Each criminal case in the dataset is derived from a corpus of Chinese judicial documents and typically includes information about the plaintiff and defendant, a basic description of the case facts, relevant evidence and laws, and the judgment result. Each civil case contains information about the plaintiff and defendant, the plaintiff's claims, a statement of facts, and the court's decision.

TABLE 1. Dataset structure details.

Dataset	Criminal Case Dataset	Civil Case Dataset
Training Set	15306	6018
#Query Cases	5102	2006
#Candidate Cases	10204	4012
Validation Set	1000	1012
#Query Cases	100	92
#Candidate Cases	900	920
Test Set	1000	1012

B. EVALUATION METRICS AND MODEL PARAMETERS

Because the SCM dataset does not include similarity labels for case names, laws, case descriptions, and so on, but instead directly labels whether two cases are similar or not, this study employs Recall@1/10, Recall@2/10, and Recall@5/10 to assess the model's ability to predict similar cases from 10 given candidates for clearly labeled cases in both the proposed and baseline models. When compared to other advanced models, accuracy ensures that the results are consistent across models. Fine-tuning experiments in the BERT encoding layer are carried out with Google's BERT_base, Chinese BERT_WWM, and a legal-specific BERT. The BERT model, pre-trained by Open Clap on 6.63 million documents, is chosen as the baseline model for criminal SCM. The default hyperparameters from the BERT model are used. For civil SCM, the baseline model

TABLE 2. Comparison of different similarity computation methods.

Model	Criminal Case Dataset			Civil Case Dataset		
	R@1/10	R@2/10	R@5/10	R@1/10	R@2/10	R@5/10
Cos	0.705	0.788	0.871	0.756	0.855	0.916
L1	0.122	0.263	0.462	0.102	0.225	0.50
Jaccard	0.712	0.724	0.776	0.454	0.659	874
Dot Product	**0.843**	0.981	1.0	**0.897**	0.898	0.917

TABLE 3. Experimental results of the models.

	Model	Criminal Case Dataset			Civil Case Dataset		
		R@1/10	R@2/10	R@5/10	R@1/10	R@2/10	R@5/10
base	Attention	0.406	0.519	0.781	0.430	0.787	0.969
	BERT	0.306	0.506	0.825	0.517	0.639	0.866
Sensitivity Analysis	BERT+Attention1	0.688	0.688	0.794	0.532	0.537	0.676
	BERT+Attention2	0.581	0.919	1.0	0.602	0.837	0.884
Ours	Ours	**0.843**	0.981	1.0	**0.897**	0.898	0.917
	Ours_turn	**0.895**	0.895	0.895	**0.902**	0.990	0.991

is the pre-trained BERT model from Open Clap, which is trained on 26.54 million documents and uses default hyperparameters. The similarity score is computed by the dot-product method, which sums the products of corresponding elements in two vectors to measure their overall similarity. As shown in Table 2, experiments are conducted to compare cosine similarity, Manhattan similarity, Jaccard similarity, and dot product similarity. The experimental results and analysis consistently show that the dot-product method best captures the overall similarity between two vectors, resulting in the best performance.

C. EXPERIMENTAL RESULTS

In this experiment, the models are trained on case datasets. During training, the ratio of positive to negative cases is 1:1, which means that each query item is associated with one matching and one non-matching case. During validation, the ratio of positive to negative cases is 1 to 9. The experimental results are shown in Table 3.

The experimental results show that, when compared to baseline models such as BERT and Attention, the proposed model's prediction metrics increased by 15% to 54%, respectively. Additionally, as illustrated in Figure 6, the model's accuracy gradually improved while its loss decreased during the training process. The proposed model can better represent long texts, thereby performing downstream tasks more effectively. It demonstrates the superiority of the proposed model in the SCM task.

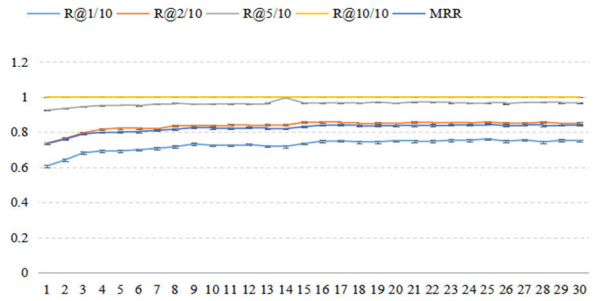

FIGURE 6. Model training trend graph.

D. SENSITIVITY ANALYSIS

To validate the effectiveness of the model modules, this study conducted experiments contrasting them with the baseline BERT [28], BERT+Attention1, and BERT+Attention2. The experimental results are presented in Figures 7 and 8. From the results in Table 3, it is evident that adding Attention Mechanism 1 and Attention Mechanism 2 to the

BERT model significantly improves experimental outcomes. Upon evaluation, our proposed model exhibits substantial superiority in evaluation metrics compared to these baseline models, highlighting the effectiveness of each module in downstream SCM tasks. Further improvement is achieved by integrating both mechanisms into the model. Therefore, each component of the model has been proven effective and indispensable. Additionally, this study employed a fine-tuning approach that simultaneously encodes query and candidate items, as detailed in Table 3. Experimental results show that fine-tuning enhances the model's accuracy in case recommendation tasks relative to the original model. However, fine-tuning prolongs training time compared to our original model presented in this paper.

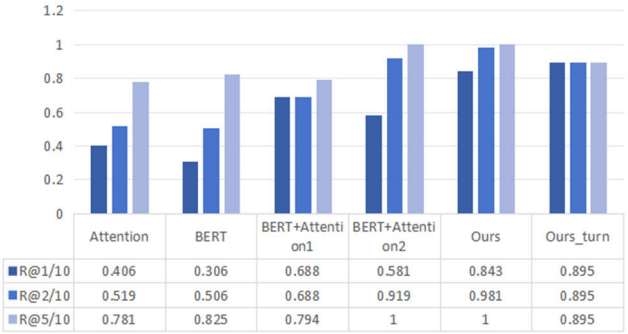

FIGURE 7. Comparison of predicted results for criminal cases.

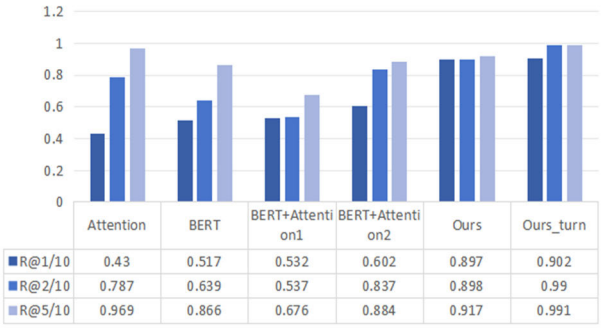

FIGURE 8. Comparison of the predicted results for civil cases.

E. ANALYSIS OF SCM MODEL RESULTS

By transforming case descriptions, types, legal provisions, and other information into feature vectors, the model captures key information from multiple perspectives. It allows a better understanding of the underlying information conveyed in the text and improves case-matching accuracy. The model is based on a BERT encoder and includes attention that considers both local and global information in the text enabling it to handle longer texts more effectively. Extracting key information from both local and global perspectives significantly improves the model's ability to extract critical information. As shown in Table 4, the model proposed in this paper outperforms existing advanced models by more than 10%. The experimental results confirm the model's effectiveness, demonstrating its strong performance in SCM. Sensitivity analysis involves detailed verification of both overall performance and module effectiveness, demonstrating the model's reliability and stability. The model has the potential to be used not only in the legal domain but also in other fields for text matching. Based on the SCM model proposed in this paper, it aids in LJP, legal consulting services, and other Legal AI construction efforts, while also offering inspiration for improving other text-matching tasks.

TABLE 4. Comparison of evaluation results with advanced models.

	Model	Civil Case Dataset Accuracy
Best	The-Siamese	0.530
	CASCMR	0.738
	LFESM	0.742
Ours	Ours	**0.844**
	Ours_turn	**0.901**

V. CONCLUSION

To address the issue of low accuracy in case recommendation, this paper proposes a SCM model based on deep text understanding. By capturing both local and global key information, the model achieves a better representation of long texts, thereby improving the accuracy of SCM. Experimental results show that the proposed model not only performs well in SCM but can also be applied to other dialogue systems or recommendation tasks. During the model fine-tuning process, we found that increasing the number of attention heads improves model performance. If computational power is sufficient, using more attention heads can further enhance the model's effectiveness. Additionally, we observed that appropriately adjusting other hyperparameters, such as learning rate and batch size, can significantly impact model performance. These findings provide valuable insights for future research and applications.

Despite the outstanding performance of the proposed model in various aspects, there are still some limitations. Handling extremely long texts requires high computational resources, which may affect the practical application of the model. Furthermore, while the study primarily focuses on SCM tasks, future research could explore the potential application of this model to other legal text analysis tasks.

REFERENCES

[1] *Guiding Opinions of the Supreme People's Court on Unifying the Application of Laws and Strengthening the Search of Similar Cases (Trial Implementation)*, Supreme People's Court, People's Court Daily, Beijing, China, 2020.

[2] Z. Wang, "Exploration of artificial intelligence application in case recommendation," *Legal System Soc.*, no. 8, pp. 207–208, Mar. 2020.
[3] J. Mo, J. Zheng, and D. Bi, "Establishment of a copyright regulations knowledge base and development of a case recommendation system utilizing transformer and graph convolutional network," *IEEE Access*, vol. 12, pp. 32849–32858, 2024.
[4] Y. Liu, T.-P. Tan, and X. Zhan, "Iterative self-supervised learning for legal similar case retrieval," *IEEE Access*, vol. 12, pp. 17231–17241, 2024.
[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minneapolis, MI, USA, 2019, pp. 4171–4186.
[6] R. Dutt, M. Basu, K. Ghosh, and S. Ghosh, "Utilizing microblogs for assisting post-disaster relief operations via matching resource needs and availabilities," *Inf. Process. Manage.*, vol. 56, no. 5, pp. 1680–1697, Sep. 2019.
[7] Z. Xu, B. Huang, and W. Pan, "A new context-aware case matching and recommendation method," *J. Taiyuan Univ. Technol.*, vol. 53, no. 1, pp. 80–88, Jan. 2022.
[8] R. E. Susskind, "Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning," *Modern Law Rev.*, vol. 49, no. 2, pp. 168–194, Mar. 1986.
[9] C. Sansone and G. Sperlí, "Legal information retrieval systems: State-of-the-art and open issues," *Inf. Syst.*, vol. 106, May 2022, Art. no. 101967.
[10] Z. Hong, Q. Zhou, R. Zhang, W. Li, and T. Mo, "Legal feature enhanced semantic matching network for similar case matching," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., Jul. 2020, pp. 1–8.
[11] Z. Liu, M. Zhang, and R. Zhen, "Multi-task learning model for legal judgment prediction with crime keywords," *J. Tsinghua Univ. Sci. Technol.*, vol. 59, no. 7, pp. 497–504, Apr. 2019.
[12] C. Wang and X. Jin, "Study on the multi-task model for legal judgment prediction," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Dalian, China, Jun. 2020, pp. 309–313.
[13] Z. Jiang, L. Li, and D. Huang, "Word vector model based on word relationships," *J. Chin. Inf. Process.*, vol. 31, no. 3, pp. 25–31, May 2017.
[14] K. Raghav, P. K. Reddy, and V. B. Reddy, "Analyzing the extraction of relevant legal judgments using paragraph-level and citation information," in *Proc. ECAI*, Hague, The Netherland, 2016, pp. 30–37.
[15] X. Cao and K. Zhou, "Text similarity calculation method based on multi-head self-attention mechanism Siamese network," *Microelectron. Comput.*, vol. 38, no. 10, pp. 15–20, Sep. 2021.
[16] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Vancouver, BC, Canada, 2017, pp. 1657–1668.
[17] K. K. Bharti and P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3105–3114, Apr. 2015.
[18] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2016, pp. 1480–1489.
[19] B. Jang, M. Kim, G. Harerimana, S.-U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2Vec CNN and attention mechanism," *Appl. Sci.*, vol. 10, no. 17, p. 5841, Aug. 2020.
[20] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[21] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Montreal, QC, Canada, Jul. 2005, pp. 729–734.
[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR*, Scottsdale, AZ, USA, 2013, pp. 1–12.
[23] W. Deng, "Research on judicial intelligence based on deep learning," M.S. thesis, Harbin Inst. Technol., Heilongjiang, China, 2017.
[24] H. Zhong, Z. Zhang, and Z. Liu. (Jul. 2, 2019). *Open Chinese Language Pretrained Model Zoo*. Tsinghua Univ., Beijing, China. [Online]. Available: http://zoo.thunlp.org
[25] W. Hu, S. Zhao, Q. Zhao, H. Sun, X. Hu, R. Guo, Y. Li, Y. Cui, and L. Ma, "BERT_LF: A similar case retrieval method based on legal facts," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–9, Apr. 2022.
[26] F. Fang, X. Li, and Y. Liu, "Low-resource similar case matching in legal domain," in *Proc. ICANN*, Bristol, U.K., 2022, pp. 570–582.
[27] K. Chen and H. Liu, "Chinese text classification method based on improved BiGRU-CNN," *Comput. Eng.*, vol. 48, no. 5, pp. 59–66, Jul. 2022.
[28] Y. Jin, X. Yang, and Y. Zhang, "Attention-guided multimodal fusion for RGB-D image segmentation," *Comput. Eng. Design*, vol. 43, no. 12, pp. 3453–3460, Dec. 2022.
[29] F. Xu, J. Liu, Q. Lin, Y. Pan, and L. Zhang, "Logiformer: A two-branch graph transformer network for interpretable logical reasoning," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Madrid, Spain, Jul. 2022, pp. 1055–1065.
[30] J. Kim, A. Lamb, S. Woodhead, S. Peyton Jones, C. Zhang, and M. Allamanis, "CoRGi: Content-rich graph neural networks with attention," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2022, pp. 773–783.
[31] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, "Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring," 2019, *arXiv:1905.01969*.

JIE XIONG is currently pursuing the master's degree in management science and Engineering with Xiamen University of Technology. Her research interests include natural language processing, data mining, intelligent optimization, machine learning, and artificial intelligence with a focus on their practical applications.

YIHUI QIU received the Ph.D. degree in systems engineering from Xiamen University, China. She is currently a Professor with the School of Economics and Management, Xiamen University of Technology. Her research interests include data mining, pattern recognition, deep learning, and intelligent manufacturing with a focus on their practical applications.