ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Research on method of extracting legal document elements and predicting case types

Zhicheng Huang[1,*]

[1] School of Law, Case Western Reserve University, Cleveland, OH, 44106, US

SUMMARY: *In this paper, we propose a multi-task legal document analysis framework for feature extraction and case type prediction. Our method integrates Transformer text encoders, hierarchical feature aggregation, and two-branch supervision to model fact statements, procedural records, legal citations, and party information in legal documents. A corpus of 12,000 Chinese legal documents covering eight case categories was constructed, with 9600 samples for training, 1200 samples for validation, and 1200 samples for testing. Experimental results show that the proposed model achieves 93.4% micro f1 value and 91.8% macro f1 value in feature extraction, and achieves 92.6% accuracy and 91.1% macro f1 value in case type prediction. The average inference time per document was 38 ms. Under the type constraint correction, the boundary agreement rate of the system reaches 92.7%, and shows a good overall cross-class discrimination stability. The framework can provide highly accurate structured results for legal document indexing, content retrieval and case intelligent analysis, and can also support downstream applications such as evidence organization and semantic recommendation in information systems.*

KEYWORDS: *Legal document elements extraction; Case type prediction; Multi-task learning; Deep learning*

1 Introduction

With the continuous advancement of judicial informatization and electronic files, legal documents have become the core carrier of case facts expression, law application presentation and judgment logic solidification. The complaint, reply, judgment, award and court record contain many kinds of key information, such as the identity of the parties, the dispute focus, the complaint matter, the evidence point, the legal article reference and the judgment conclusion. These contents together form the data basis for subsequent retrieval, filing, statistics, recommendation and auxiliary analysis. In the face of the rapidly growing scale of judicial texts, it is difficult to rely only on manual reading and manual sorting to support the structured processing requirements in high-frequency call scenarios. Computer technology is introduced into the process of legal document analysis, and a stable method of element extraction and case type prediction is established, which can transform unstructured documents into computable, comparable and searchable semantic representations, and provide a unified entrance for similar case discovery, knowledge organization and intelligent assistance.

The task of legal document processing has distinct domain characteristics. First, the length of the document is long and the syntactic level is deep. Fact narration, procedure description

*wangjinxia.chn@foxmail.com
https://doi.org/10.65102/is2026738 Published: 30 April, 2026

and legal argument often intersect, so it is difficult for ordinary short text models to completely maintain the document-level dependencies. Secondly, the entity boundary in the document is obviously affected by the context, and the name of the subject, the expression of time, the amount involved in the case, the legal number and the behavior description often appear in the form of nested or deformed, so the fixed template is difficult to cover the real scene. Third, there are significant differences in element distribution, narrative focus and judgment expression among different case categories, and the semantic role of the same field in contract disputes, tort disputes and enforcement cases is not consistent. Fourth, case type judgment is not an isolated classification process, but is closely related to fact unit recognition, law semantic understanding and procedural information modeling, which puts forward higher requirements for feature representation hierarchy, task collaboration and result consistency.

Focusing on legal text computing, Chen et al. studied the performance difference between random forest and deep learning model in automatic legal text classification, indicating that deep representation has stronger adaptability in complex semantic discrimination [1]. Enamoto et al. proposed a multi-label legal text classification method combining BiLSTM and attention mechanism, which enhanced the ability of multi-category correlation expression [2]. Alcantara Francia and other systems sorted out the text mining technology in judicial judgment prediction, and provided a task spectrum and method reference for legal semantic computing [3]. Rabelo et al. summarized the evaluation framework of legal information extraction and entailment recognition around the COLIEE competition, which promoted the formation of a comparable experimental environment [4]. Cao et al. proposed CAILIE 1.0 dataset for legal information extraction, which provides an open basis for the research on feature level annotation [5]. Cui et al. reviewed the datasets, indicators and models of legal judgment prediction, and further clarified the technical boundaries of legal artificial intelligence [6]. Zhao et al. proposed a legal judgment prediction method combining seme-enhanced graph neural network and multi-graph fusion mechanism, which improved the structural relationship modeling ability [7]. Zahir studied the judgment prediction of Arabic grammar court documents based on deep learning, demonstrating the transferability of cross-lingual legal text computation [8].

Existing research has laid a solid foundation for legal text classification, legal information extraction and judgment prediction, but the integrated method for legal document element extraction and case type prediction still needs a closer task organization way. It is easy to ignore the constraint effect of case category prior on boundary determination and label assignment by using element identification alone, and it is difficult to make full use of the discrimination value of fine-grained fact units inside the document by using type classification alone. Based on this understanding, this paper takes the computational processing of legal documents as the main line, constructs a multi-layer feature representation and two-branch task modeling framework, jointly describes the discourse semantics, local entity clues and case category patterns in the shared coding space, and introduces a factor result correction mechanism under the constraint of case type discrimination to improve the integrity of the extraction results and the stability of the classification results. The research content includes legal document data construction, field labeling and preprocessing, multi-layer representation learning, dual-task collaborative modeling, and experimental verification. The goal is to form a set of computational methods suitable for legal information systems, judicial retrieval platforms, and intelligent auxiliary scenarios. This method emphasizes the establishment of interpretable connections between text structure, legal semantics and category priors, so that the model output can directly serve document governance, case classification and knowledge indexing.

2 Literature Review

2.1 Difficulties in modeling text features and elements of legal documents

Legal documents are one of the text forms with the highest information density in judicial activities. Case facts, procedures, evidence points, legal basis and judgment conclusions usually intersect in the same document. Compared with general news texts and short comment texts, legal documents have the characteristics of long length, deep nested sentence patterns, stable terms and more expression variants. The cause of action, the identity of the parties, the matter of claim, the time and place, the amount range, and the legal citations are often spread across different paragraphs, and the entity boundaries are constantly disturbed by quotes, brackets, numbers, and additional notes. Almuzaini et al. studied the judicial decision support system based on the deep learning framework, and pointed out that there are strong semantic dependencies and cross-paragraph correlations within legal sentences, and a single local feature is difficult to support stable recognition [9]. Lage-Freitas et al. studied the judgment prediction task of Brazilian courts and showed that there is an obvious document-level mapping relationship between factual narrative and outcome expression in legal texts [10]. Bi et al. proposed a numerical reasoning method integrating legal knowledge, revealing that quantitative elements such as amount, proportion and term are not isolated fields in judicial texts, but work together with the semantic of rules [11]. Sukanya et al. proposed an improved hierarchical attention network model, showing that the hierarchical structure modeling within long documents has a direct impact on legal semantic preservation [12].

From the perspective of computer modeling, the element extraction of legal documents is not a simple task of named entity labeling, but a composite processing process including discourse localization, semantic alignment and structure merging. Although different document templates are standardized in form, the coexistence of shorthand, paraphrasing, ellipsis and multiple rounds of reference is common in real data, which leads to the instability of similar elements on the syntactic surface. Similarly, case type prediction cannot be compressed into ordinary single-label classification, because factual strength, procedural stage, and law density in the document will jointly affect the class discrimination boundary. To achieve high-quality modeling, the model not only needs to maintain cross-sentence dependencies, but also needs to recognize the mapping relationship between multi-granularity semantic clues, and establish a shared representation space between field extraction and category discrimination. It can be seen that the complex combination of text features of legal documents determines that element modeling must simultaneously take into account context continuity, structural constraints and category sensitivity, which also constitutes the technical starting point for the design of subsequent methods. At the same time, the replacement of subjects with the same name, alias names of evidence and abbreviations of legal articles appear frequently in documents, which further raises the requirements of annotation consistency and automatic recognition accuracy.

2.2 Application status of artificial intelligence in legal text processing

The application of artificial intelligence in legal text processing has gradually shifted from the early rule matching and artificial feature engineering to the deep modeling path with pre-training representation, attention calculation and task collaborative learning as the core. This change directly promotes the analysis of legal documents from keyword retrieval to semantic understanding, so that the fact statement, responsibility attribution and law application in the document can be calculated in a unified representation space. Bertalan et al. studied the judicial outcome prediction based on attention method, and proved that attention weight can effectively capture the centralized distribution of judgment basis in the discourse [13]. Sun et al. proposed a knowledge prompt learning method for Chinese legal judgment prediction, indicating that the combination of external legal knowledge and text representation can enhance the ability of complex semantic discrimination [14]. Tong et al. proposed a legal judgment prediction method with constraint graph lifting, which introduced structural relationship constraints into the prediction process and improved the stability of category discrimination [15]. Castano et al. proposed a legal information extraction method ASKE based on context-aware technology, showing the collaborative computing path among entities, contexts and rules [16].

From the perspective of task spectrum, legal text processing has covered multiple directions such as element recognition, cause of action classification, law recommendation, judgment prediction and result interpretation. The most computationally valuable trend is that the model shifts from single-point output to multi-granularity collaborative output. The pre-trained language model can provide strong context representation, the graph structure method can supplement the law relationship and entity connection, the attention mechanism helps to highlight key fact fragments, and the context-aware extraction framework enhances the stability of field location. For legal document element extraction and case type prediction, these technologies are not parallel and stacked, but form complementary links: document coding is responsible for establishing the semantic base of the text, structural constraints are responsible for compressing unreasonable category drift, knowledge suggestion is responsible for supplementing the semantic gap of the domain, and context extraction is responsible for improving the accuracy of field recognition. The development of artificial intelligence in legal text processing shows obvious systematic characteristics, and also provides a reusable method basis for integrated task design. At the level of engineering implementation, the data noise in the legal scene does not mainly come from typo, but more from citation intersection, template difference and role transformation. Therefore, the model needs to preserve local boundary sensitivity and global semantic discrimination at the same time. For Chinese legal documents, the granularity of word segmentation, the form of legal article number and the proper name abbreviation will continue to affect the coding quality.

2.3 Analysis of existing element extraction and case type prediction systems

The existing feature extraction and case type prediction systems have gradually extended from a single text classification framework to a composite system for structured output, but the focus of different systems is not the same. One system emphasizes the management of judgment documents and entity organization, another emphasizes field extraction, label assignment and category discrimination, and still another pays more attention to the connection mode of the interface between knowledge retrieval and auxiliary analysis. Bellandi et al. proposed an entity-centric court decision management method based on natural language processing, indicating that the judicial text system has shifted from full-text storage to organizing information around entity units [17]. Lu et al. proposed a named entity recognition method for Chinese legal documents based on parallel instance query network, which improved the fine-grained expression ability of complex entity location [18]. Yulianti et al. constructed an Indonesian legal document dataset and studied a Transformer-based named entity recognition model, showing that the matching between domain data and coding architecture has a direct impact on the recognition effect [19]. Guimaraes et al. studied the method of segmentation and annotation of legal documents through named entity recognition, which promoted the transformation of legal documents from linear text to structural fragment

[20].

From the perspective of system analysis, existing studies have proved that entity extraction, fragment segmentation, and semantic annotation in legal texts can be used as the basic modules of case processing systems, but there is still room for integration between feature extraction and case type prediction. The entity-centric method is helpful to organize the information of people, institutions, laws and time, but it may not be able to complete the stable judgment of the cause of action simultaneously. The named entity recognition model can improve the accuracy of field extraction, but without category constraints, the field results are prone to boundary drift in cross-cause scenarios. Document segmentation and annotation methods can enhance the perception of document structure, but they still need to be linked with the type discrimination results to form a more complete legal information processing link. Therefore, the development direction of existing systems has shifted from single-module performance improvement to cross-module coupling design. The research on the method of legal document element extraction and case type prediction needs to establish a more stable connection between shared coding, branch output and result correction, so as to form a technical solution that takes into account accuracy, structure and deployment. This integrated system not only serves for document retrieval, but also supports case cataloging, case merging and rule discovery, and makes the output results more suitable for the subsequent calculation process of the judicial information platform.

3 Extraction of legal document elements and design of case type prediction model

3.1 Task definition and overall processing flow

Legal document element extraction and case type prediction are two-layer computing tasks in the same judicial text. The former requires the system to identify structural elements such as party information, cause of action expression, dispute facts, time node, amount field, law reference and judgment results from texts such as judgments, complaints, replies and rulings. The latter requires the system to complete the case type discrimination under the joint effect of full text semantics, procedural information and fact strength. Both tasks depend on document-level context, and form result objects with different granularities at the output level. Therefore, the whole process cannot be modeled independently from each other, but needs to be organized as a continuous processing chain based on unified input, shared state, and branch reasoning.

After the document enters the system, the format normalization, field cleaning and hierarchical segmentation are completed. The header, footer, seal noise, pagination number and repeated template sentences will be mapped uniformly, and the semi-structured content such as case number, date, trial level, amount, and law number will be jointly parsed through pattern matching and context judgment. Then, the system organizes the text into a three-level structure of "document-paragraph-sentence element", so that the semantic encoding can not only preserve local boundaries, but also continuously track inter-paragraph dependencies. This process is not a pure text chunking, but to establish a unified state propagation entry for subsequent dual-task computation. The sentence-element level representation undertakes local entity recognition and short-range context awareness, the paragraphlevel representation undertakes dispute focus and argument topic merging, and the document-level representation undertakes overall case semantic compression and type prior bearing. The three-level states together form the base input for subsequent models.

In order to make the original text, location cues and source information form a computable representation in the same space, it is necessary to establish a uniform underlying state for each sentence unit. Taking into account that the semantic role of the same word unit in legal documents can vary according to location, source and intra-sentence role, the initial representation is defined as follows.

$$z_i^{(0)} = E_w(x_i) + E_p(p_i) + E_s(s_i) + E_r(r_i) \quad (1)$$

where x_i represents the i lemma or subword unit, and $E_w(\cdot)$ is the word embedding mapping.p_i is the position index, and $E_p(\cdot)$ is used to store the order of the discourse.s_i represents the source type of the document, and $E_s(\cdot)$ is used to distinguish between different carriers such as judgments, complaints, replies and rulings. r_i represents the sentence element role label, and $E_r(\cdot)$ is used to label fact sentences, procedural sentences, legal sentences, and result sentences. This formula writes semantics, order, source and role into the underlying state at the same time, so that the subsequent aggregation process no longer only depends on word information, but can perceive the functional differences of the same expression in different positions of documents.

After the sentence-level initialization, the system needs to compress the sentence-element states in the same paragraph into a mesoscale representation to support the identification of dispute points, law concentration areas and program specification paragraphs. Paragraph states are aggregated in a weighted manner controlled by semantic gating, written as follows.

$$g_j = \sum_{i=1}^{n_j} \alpha_{ij} z_i^{(0)}, \quad \alpha_{ij} = \frac{\exp(q_j^{\top} W_g z_i^{(0)})}{\sum_{k=1}^{n_j} \exp(q_j^{\top} W_g z_k^{(0)})} \quad (2)$$

where g_j represents the semantic vector of the j paragraph and n_j is the number of sentence elements contained in the paragraph. q_j is the query vector generated by the paragraph title pattern, term density and numbering structure, W_g is the learnable parameter matrix. α_{ij} measures the contribution of the i sentence element to the paragraph representation. The function of this formula is to highlight the sentence elements inside the paragraph that really carry the judgment information of the case, rather than mechanically averaging the whole content. In this way, the generated paragraph representation not only preserves the local high-value semantics, but also compresses redundant descriptions and provides a more stable intermediate state for full-text reasoning.

To maintain cross-segment semantic continuity, the system continues to establish a document-level state transfer chain. The fact narration, dispute induction, law application and judgment conclusion in legal documents are distributed in different paragraphs. If the status of the full text is not updated, the subsequent case type determination is easy to be drawn by local fragments. Therefore, the document-level representation is updated in a gated manner:

$$c_t = \gamma_t \odot c_{t-1} + (1 - \gamma_t) \odot \tanh(W_c[g_t; u_t] + b_c) \quad (3)$$

Here, c_t represents the global state when the t paragraph is processed, and c_{t-1} is the document state at the previous time. g_t is the representation of the current paragraph, u_t is the auxiliary feature vector composed of time, amount, subject frequency and law statistics. Let γ_t be the gating coefficient and $\odot$ denote element-wise multiplication. The formula keeps the semantic connection between the preceding fact, the following argument and the conclusion, so that the global state not only reflects the local semantics, but also carries the overall trend of the case and the direction of the judgment.

Before entering the dual-task reasoning, the system needs to organize the states of different granularities into a shared input matrix to undertake the subsequent operations of the feature extraction branch and the case type prediction branch. To avoid multi-scale information being overlaid by a single representation as it enters the task layer, the system uses explicit concatenation to organize the input:

$$U = [Z^{(0)}; G; C; M] \tag{4}$$

where $Z^{(0)}$ is the sentence-level initial sequence, G is the paragraph state set, C is the document-level state trajectory, and M is the metadata matrix. This formula is not a common concatenation, but sends the information of different granularities into the subsequent double-branch module at the same time, so that the field boundary and the full-text category can be reasoned under the same framework. The shared input matrix constructed in this way not only retains the fine-grained features of the word level, but also retains the overall semantics of the paragraph level and document level, which can better adapt to the long text attributes and structural requirements of legal documents.

Fig. 1 shows the overall processing sequence of the task of legal document element extraction and case type prediction. After format normalization, field cleaning and hierarchical segmentation, the original document is converted into three levels of state representation: sentence level, paragraph level and document level. The three-layer states are fed into the shared input matrix along with the metadata and subsequently into the dual-task inference module. The system generates the feature label sequence and the case type label respectively, and then forms the final structured output through the result correction module.

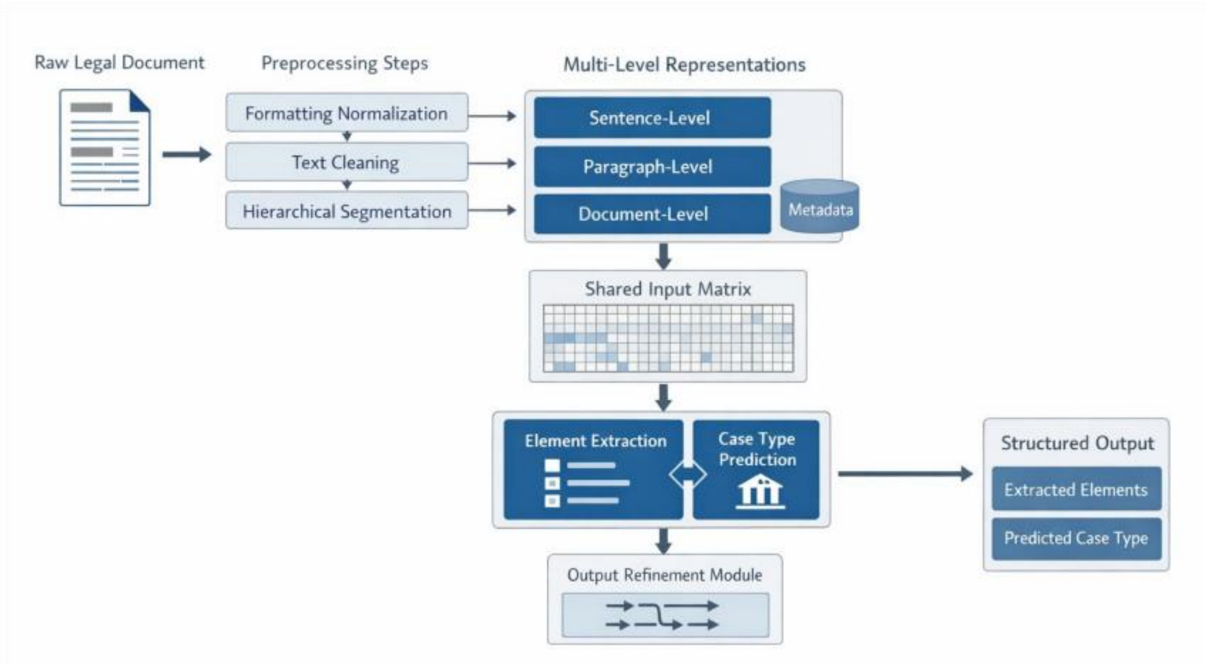

Figure 1: Legal document task definition and overall processing flow chart

Through the above process definition, legal documents are no longer regarded as natural language materials that are difficult to compute directly, but are reconstructed as judicial data objects that can be hierarchically represented, state propagated, and jointly reasoned. The whole process starts from the raw text access and ends with the structured field table, case type label and index record output, forming a complete cohesive chain. The direct result of this design is that the system can establish a consistent state foundation for field extraction and type discrimination under the premise of maintaining the semantic continuity of the whole text.

3.2 Multi-layer feature representation and two-branch task modeling of legal documents

After the overall process is determined, the key of model design turns to multi-layer feature representation and two-branch task modeling. The information in legal documents is not uniformly distributed: the word element level is suitable for capturing entity boundaries and legal article numbers, the sentence element level is suitable for expressing actions, facts and responsibilities, the paragraph level is suitable for organizing the dispute structure and argument density, and the document level is suitable for forming the overall judgment of cause of action and procedural properties. If the full text is directly compressed into a single vector, the fine-grained field recognition will be overwhelmed by the global semantics. If we only rely on local sequence labeling, it is difficult to obtain enough discourse evidence for case types. Based on this feature, the model adopts a shared coding layer to undertake multi-layer semantics, and then performs differentiated reasoning by the feature extraction branch and the case type prediction branch.

The shared encoding layer is responsible for mapping the sentential state, paragraph state and document state to the same semantic space, and keeping the local boundary updated with the global topic in the process of cross-layer propagation. The sharing here is not a simple superposition of parameter multiplexing, but the construction of alignable semantic tensors around the hierarchical structure of legal documents, so that different granularities of evidence remain compatible in the same model. The sentence element layer emphasizes field boundaries, the paragraph layer emphasizes dispute topics, and the document layer emphasizes the full picture of the case. These three types of features must form a stable upper and lower relationship in the shared representation, otherwise the optimization direction of one task branch will dilute the semantic clues relied on by the other branch.

In order to form an effective relationship between sentence units, paragraphs and document states, the model first constructs a cross-layer update mapping. This mapping enables each sentence unit to receive the local context, the topic of the paragraph it belongs to, and the semantic context of the full text at the same time when it is updated, which is defined as follows:

$$h_i^{(l+1)} = LN(h_i^{(l)} + W_1 \phi(h_i^{(l)}) + W_2 \psi(g_{\pi(i)}) + W_3 c) \tag{5}$$

where $h_i^{(l)}$ represents the i sentence state at the l layer, $g_{\pi(i)}$ represents the paragraph state, c represents the document-level vector, $\phi(\cdot)$ and $\psi(\cdot)$ are the local mapping function and paragraph mapping function, respectively, and LN is the layer normalization. This formula enables the shared encoding layer to continuously inject paragraph theme and full text background while preserving the details of sentence boundary, so it is more suitable for processing long, deep-level and strongly context-dependent texts such as legal documents.

In the feature extraction branch, the field label is not only dependent on the sentence element itself, but also restricted by the full text case topic. If only the local state is used to determine the label, the seemingly similar segments such as amount, time, subject and law are prone to role confusion. Therefore, the extraction branch uses a type-sensitive label distribution function:

$$P(y_i^e = k) = \frac{\exp(v_k^T \tanh(W_e h_i^{(L)} + U_e c + b_e))}{\sum_{m=1}^{K_e} \exp(v_m^T \tanh(W_e h_i^{(L)} + U_e c + b_e))} \tag{6}$$

where y_i^e represents the label of the i sentence unit in the feature extraction task, K_e is the number of feature labels, $h_i^{(L)}$ is the final state of the shared layer, and W_e, U_e, b_e, and v_k are the learnable parameters. This formula sends the local state and the global case semantics into the tag decision at the same time, so that the fields such as party, time, amount, law, and judgment result can still maintain semantic distinction under similar surface forms. The extraction branch is thus no longer a pure sequence labeling model, but a field reasoning module embedded with the overall semantic constraints of the case.

The case type prediction branch emphasizes the overall perception of the discourse structure, so the classifier after average pooling cannot be simply used. The category of a case in a legal instrument is often determined by a combination of multiple paragraphs of facts, the focus of the dispute, and the application of the statute, rather than by a single sentence. Based on this understanding, the classification branch uses gated aggregation to generate case representations:

$$r = \sum_{j=1}^{m} \beta_j g_j, \quad \beta_j = \frac{\exp(w_t^T \tanh(W_t g_j + U_t c))}{\sum_{q=1}^{m} \exp(w_t^T \tanh(W_t g_q + U_t c))} \tag{7}$$

Here, r represents the case-level semantic vector, g_j is the j paragraph representation, and β_j measures the contribution of different paragraphs to the decision of case type. This formula enables the model to focus on absorbing high-value evidence in fact statement segments, dispute focus segments and law application segments, and weakens the disturbance of procedural description on classification results. The case-level vector thus formed not only reflects the text topic, but also retains the structural differences at the paragraph level, which is more suitable for long document type classification.

In order to prevent the two branches from squeezing each other on the shared layer, the model introduces a task coupling loss in the training phase. The feature extraction branch is biased towards boundary sensitivity, and the case type prediction branch is biased towards discourse aggregation. If there is no coordination mechanism, the shared representation will be overly dominated by one branch. The joint loss is defined as follows.

$$\mathcal{L}_{mt} = \mathcal{L}_{ext} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 ||A_e A_t^T - I||_F^2 \tag{8}$$

Here, $\mathcal{L}_{ext}$ represents the feature extraction loss, $\mathcal{L}_{cls}$ represents the case type prediction loss, A_e and A_t are the attention weight matrices of the two branches, and λ_1 and λ_2 are the balance coefficients. This formula constrains the two branches to form reasonable cooperation in the shared representation space, avoids the excessive dominance of a single task in the coding direction, and also makes the extraction and classification maintain the same semantic direction on the same document.

Fig. 2 shows the internal connection relationship after the multi-layer feature representation enters the two-branch model. The sentence-level states, paragraphlevel states and document-level states first complete cross-layer interaction at the shared encoding layer, and then form a unified shared semantic representation. The feature extraction branch performs field boundary identification and label determination based on the representation, and the case type prediction branch performs discourse aggregation and category classification based on the same representation. The two branches jointly accept a joint loss constraint in the training phase to ensure the consistency of the shared representation and the task output.

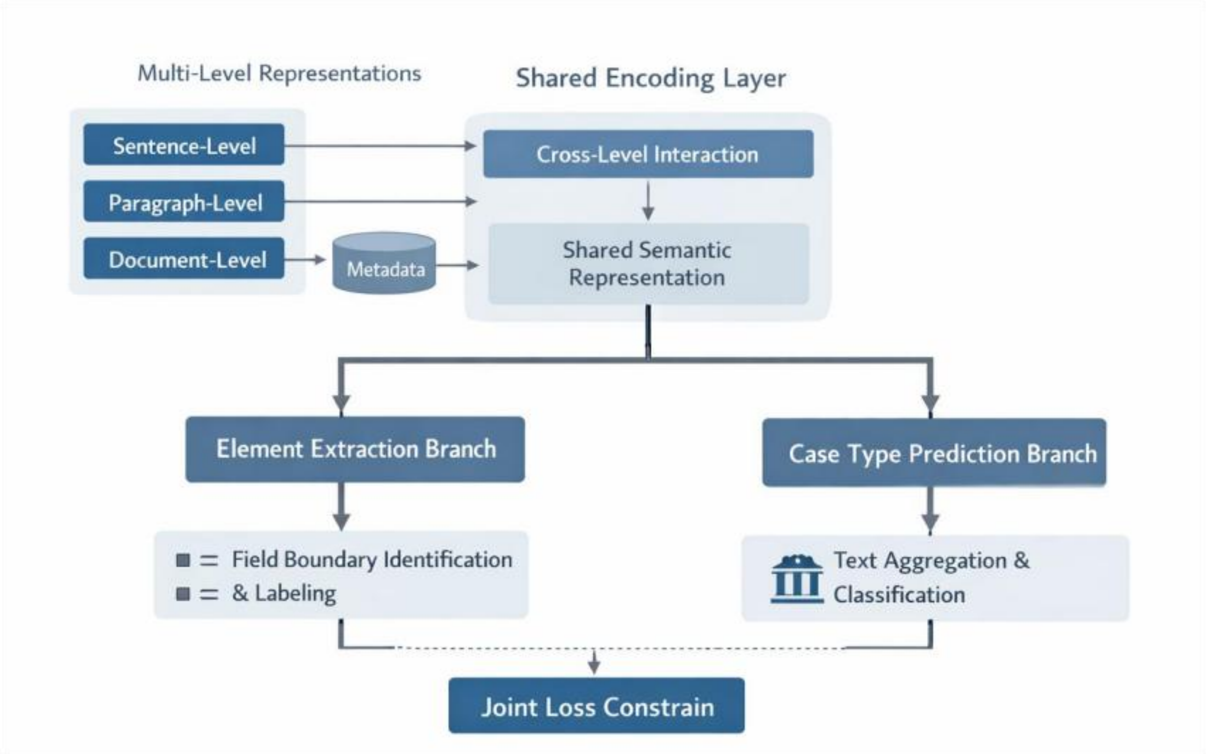

Figure 2: Multi-layer feature representation and two-branch task modeling diagram for legal documents

Through this modeling method, the field boundary, paragraph topic and discourse category are no longer calculated separately, but form a dual-branch structure with clear division of labor and consistent goals on a shared semantic base. The shared layer ensures the continuous accumulation of general legal semantics, the extraction branch ensures fine-grained field positioning, the classification branch ensures global case discrimination, and the coupling constraint ensures that the two keep a consistent interpretation direction on the same document.

3.3 Correction mechanism of element results under the constraint of case type discrimination

After the two-branch inference, the system has obtained the initial feature label sequence and the probability distribution of case types, but these two types of output still need to enter the unified correction stage. Many fields in legal documents cannot be finalized based on partial scores alone. The same amount may correspond to the claim amount, liquidated damages or the object of execution; The same time phrase may correspond to the signing time, the prosecution time or the judgment time. The same statute number may appear in the party's claim and may appear in the reason for judgment. If we only rely on local label probabilities, the extraction results are prone to boundary drift, role confusion and field overlap. The case type prediction provides a strong global prior, which can be written back into the feature space as a correction constraint, thereby improving the stability and structural consistency of the results.

The first correction comes from the type condition mapping. Different case types correspond to different field active modes. Contract disputes emphasize more on performance nodes and liability for breach of contract, labor disputes emphasize more on labor relations and wage composition, and enforcement cases emphasize more on the target amount and performance status. To turn this prior into a computable gain, we define the type condition correction term as follows:

$$\tilde{s}_{ik} = s_{ik} + \eta \sum_{c=1}^{C} P\,(c|D)\,\Omega_{ck} \tag{9}$$

Here,s_{ik} represents the original score of the i sentence element for the k element label, $P(c|D)$ represents the probability that document D belongs to the c type of case, Ω_{ck} is the statistical correlation matrix between the case type and the element label, η is the control coefficient. This equation directly writes the global prior of the output of the classification branch back into the label score space, so that the more common and stable field patterns in this class of cases are strengthened. Instead of replacing the original label score, type condition modification introduces global category support on top of the original local judgment, so it can improve the overall consistency of field labels without destroying the underlying discrimination ability.

To keep label paths locally coherent, the system further introduces a boundary consistency objective. The expression of subject name, legal article number and amount in legal documents often spans multiple sentence elements. If each position is decided independently, the boundary will be broken, which will affect the subsequent structured storage and retrieval call. For this purpose, the path score function is defined as follows:

$$\mathcal{B}(y) = \sum_{i=2}^{n} \Gamma_{y_{i-1},y_i} + \sum_{i=1}^{n} \tilde{s}_{i,y_i} \tag{10}$$

Here,y represents the whole label path, Γ is the label transition matrix, and $\tilde{s}_{i,y_i}$ are the modified single-point scores. The formula combines path continuity and single-point confidence into the same goal, which makes the system tend to generate field sequences with closed boundaries and continuous labels. In this way, the entity boundary is no longer determined by the local maximum value alone, but is jointly constrained by the consistency of the full path, which is more suitable for the identification of long fields and composite fields in legal documents.

Type constraints also need to be extended to the law compatibility layer. If a document is judged as a labor dispute, and the law with the highest score is concentrated in the context of corporate law or negotiable instruments law, the combination, although locally credible, does not match the overall category of the case. Therefore, the system sets the legal compatibility item:

$$\mathcal{C}_{\mathrm{law}} = \sum_{m=1}^{M} \sum_{c=1}^{C} P\,(c|D)\,\rho(a_m, c) \tag{11}$$

Here, a_m represents the m law candidate and $\rho(a_m, c)$ is the compatibility scoring function between the law and the case type. This formula corrects the set of legal articles through the type prior, so that the judgment keeps the same legal semantic direction according to the class field and the case category. The law compatibility constraint not only affects the law field itself, but also indirectly affects the semantic matching between the fact elements and the judgment result elements, so as to improve the internal consistency of the whole field set.

In the field conflict layer, the system also needs to deal with the situation that different tags contend for the same segment. For example, the amount field and the target field may partially overlap, and the program time and the fact time may also share the expression interval. To avoid overwriting each other, we define the overlap penalty as follows:

$$\mathcal{R}_{\mathrm{ov}} = \sum_{p=1}^{P}\sum_{q=1}^{Q} 1\left(\mathrm{IoU}(e_p, e_q) > \tau\right) \cdot \kappa_{l_p, l_q} \tag{12}$$

where e_p and e_q are two candidate segments, $\mathrm{IoU}(\cdot)$ measures the degree of overlap, τ is the threshold, and κ_{l_p, l_q} represent the conflict cost between tag pairs. This formula does not eliminate all overlaps, but focuses on suppressing combinations of fields that are difficult to hold together in the same case type. This not only preserves the necessary semantic overlap, but also avoids the overlapping of structured results, which improves the readability and usability of subsequent field tables.

After completing condition correction, boundary consistency, law compatibility and conflict suppression, the system completes global decoding through a unified goal:

$$y^* = \arg\max_{y}\left(\mathcal{B}(y) + \lambda_3 \mathcal{C}_{\mathrm{law}} - \lambda_4 \mathcal{R}_{\mathrm{ov}}\right) \tag{13}$$

Here, y^* denotes the final label path, and λ_3 and λ_4 are the balance coefficients. This formula unified path consistency, law compatibility and conflict suppression into the same goal, so that the output result was no longer staying at the local best, but formed a globally consistent field set for the whole document. The significance of global decoding is that it really implements the prior constraints and path scoring to the final output stage, rather than staying in the middle scoring link.

In order to provide a unified basis for structured storage and manual review, the system also needs to output comprehensive confidence to support direct storage of high confidence results, manual review of medium confidence results, and reflux and reprocessing of low confidence results. The comprehensive confidence is defined as follows.

$$\omega = \sigma\left(\mu_1 \bar{s}_{\mathrm{ext}} + \mu_2 \bar{s}_{\mathrm{cls}} + \mu_3 \mathcal{C}_{\mathrm{law}} - \mu_4 \mathcal{R}_{\mathrm{ov}}\right) \tag{14}$$

where $\bar{s}_{\mathrm{ext}}$ is the average score of extraction results, $\bar{s}_{\mathrm{cls}}$ is the score of case type prediction, μ_1 to μ_4 are the weight parameters, and $\sigma(\cdot)$ is the Sigmoid function. The formula maps the field quality, category quality, law compatibility and conflict degree into a single reliability index, so that the system can adopt different processing strategies for the output of different risk levels and improve the overall deployment efficiency.

At this point, the case type is no longer just the final classification output, but becomes an important prior that drives field revision and structure screening. The result of feature extraction no longer stays at the local label matching level, but completes boundary closure, law screening, role adjustment and field conflict suppression under the guidance of global case semantics. The correction mechanism formed in this way makes Chapter 3 form a complete technical closed loop from task definition, to multi-layer modeling, and then to result writeback, and also makes legal document element extraction and case type prediction obtain higher structural consistency and engineering usability in the same system.

4 Case Evaluation

4.1 Experimental Design

The experimental design of this section focuses on the effectiveness of the legal document element extraction and case type prediction method, and aims to test the structural analysis ability and category discrimination ability of the model for long judicial texts in a unified data environment. The experimental corpus is from the China Judgment Document Network and the public law database, and 12,000 Chinese legal documents are formed after screening, covering eight types of cases: contract disputes, labor disputes, infringement liability, marriage and family, loan disputes, enforcement objections, administrative reconsideration and intellectual property rights. The training set, validation set and test set are 9600, 1200 and 1200, respectively, and the samples of each class maintain an approximately balanced distribution. The document sources include judgments, rulings and prosecution materials, which facilitates the test of the adaptation performance of the model in heterogeneous text scenarios.

The preprocessing stage completes garbled code cleaning, anonymous replacement, paragraph alignment, law number standardization, amount and date normalization, and chapter segmentation in turn. The elements are labeled by combining manual review and rule pre-labeling, and the label set includes parties, causes of action, dispute facts, time nodes, amount fields, legal provisions and judgment results. The case type labels are uniformly coded according to the main cause of the document. The experimental platform uses Python 3.11 and PyTorch 2.2, and the running environment is Intel Xeon Gold processor, RTX 4090 graphics card and 128GB memory.

The AdamW optimizer was used in the training phase, the initial learning rate was set to 2×10^{-5}, the batch size was 16, the number of training rounds was 18, the Dropout was set to 0.3, and the early stopping strategy was used to control the convergence process. The evaluation link sets indicators for the dual tasks respectively. Precision, Recall and Micro-F1 are used for the feature extraction task, and Accuracy and Macro-F1 are used for the case type prediction task. At the same time, the average reasoning time and video memory occupation of a single document are recorded to comprehensively investigate the feasibility of the model's deployment in the judicial information system.

To ensure the stability of the experimental conclusions, all the results were repeated three times with the same random seed and averaged. In the long document encoding stage, the text block length is set to 512, and the cross-block overlapping window is set to 128 to preserve the semantic continuity of adjacent paragraphs. The parameters of the shared coding layer are determined by grid search of the validation set, and the loss weights of the type branch and the feature branch are set to 1.0 and 1.2 to avoid the deviation of the field identification caused by the difference of category frequency.

4.2 Results

This section presents the experimental results from five levels: overall performance, category performance, error distribution, module contribution and running cost. The test set contains a total of 1200 legal documents, covering eight types of cases: contract disputes, labor disputes, tort liability, marriage and family, loan disputes, enforcement objections, administrative reconsideration and intellectual property rights. All the results are averaged after three repeated runs under the same hardware environment and random seed conditions to reduce the influence of accidental fluctuations on the conclusions. Since this paper performs the two tasks of feature extraction and case type prediction at the same time, the results section not only reports a single accuracy, but also synchronously presents field-level indicators, document-level indicators and system-level indicators, so as to more completely reflect the actual ability of the model in judicial text processing scenarios.

In order to first present the overall performance of the model on the test set, Table 1 summarizes the core indicators of the dual-task framework in three aspects: element extraction, case type prediction, and reasoning efficiency. This table corresponds to the overall results of the system level, which can directly reflect the comprehensive level of the model in structural analysis and category judgment.

Table 1: Overall performance results for dual tasks

Task	Precision / %	Recall / %	Micro-F1 / %	Macro-F1 / %	Other Metrics
Element Extraction	93.8	93.1	93.4	91.8	Boundary Consistency Rate: 92.7%
Case Type Prediction	92.6	91.4	92.0	91.1	Accuracy: 92.6%
System-Level Efficiency	—	—	—	—	Inference Time: 38 ms per document; GPU Memory Usage: 4.3 GB

Table 1 shows that the feature extraction task achieves 93.4% Micro-F1 and 91.8% Macro-F1 on the test set, indicating that the model maintains a relatively balanced recognition quality between high-frequency fields and low-frequency fields. The boundary agreement rate reaches 92.7%, which indicates that the system can form a stable closed boundary for fields with different spans such as party name, time expression, amount interval and law number. The Accuracy of case type prediction is 92.6%, and Macro-F1 is 91.1%, indicating that the double-branch structure has strong discrimination ability on eight types of cases, and there is no obvious class bias. The average reasoning time is controlled at 38 ms/ article, and the video memory occupation is stable at 4.3 GB, which shows that the proposed method has the conditions for online deployment in the judicial information system.

To further observe the performance differences under different case categories, Fig. 3 shows the distribution of the eight categories of cases in terms of F1 of feature extraction and accuracy of type prediction in a heatmap manner.

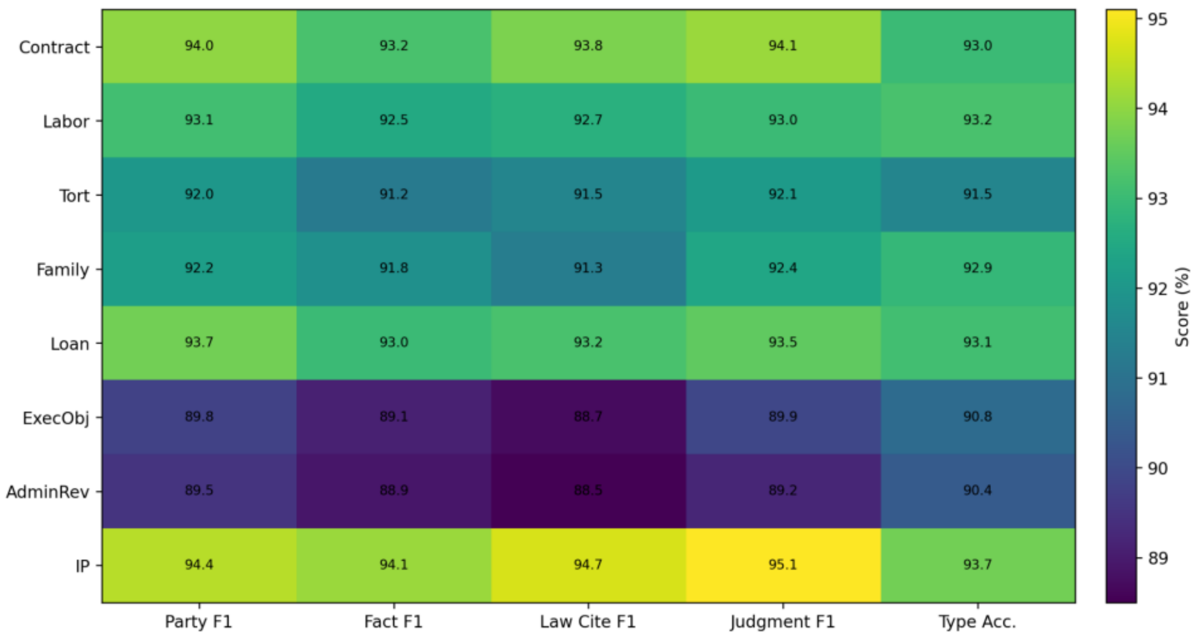

Figure 3: Heatmap of element extraction F1 versus type prediction accuracy for eight categories of cases

Fig. 3 shows that the element extraction effect of contract disputes, loan disputes and intellectual property cases is high, and the F1 of intellectual property cases in the two fields of legal article reference and judgment result reaches 94.7% and 95.1%, respectively. The fact expression of labor disputes and marriage and family documents is more concentrated, and the accuracy of type prediction reaches 93.2% and 92.9%, respectively. Administrative reconsideration and execution objection documents contain more procedural descriptions and cross-paragraph references, and the field recognition accuracy is slightly lower, but the overall F1 is still above 89%. Overall, the quality of feature extraction is not significantly damaged by category switching, indicating that the shared coding layer and the type constraint correction mechanism can maintain good transfer stability between different causes.

After the class-level results, it is also necessary to observe the error structure inside the classification output. Fig. 4 shows the confusion matrix of case type prediction, which is used to characterize the direction of misjudgment between various types of cases and the boundary clarity between the nearest neighbor categories.

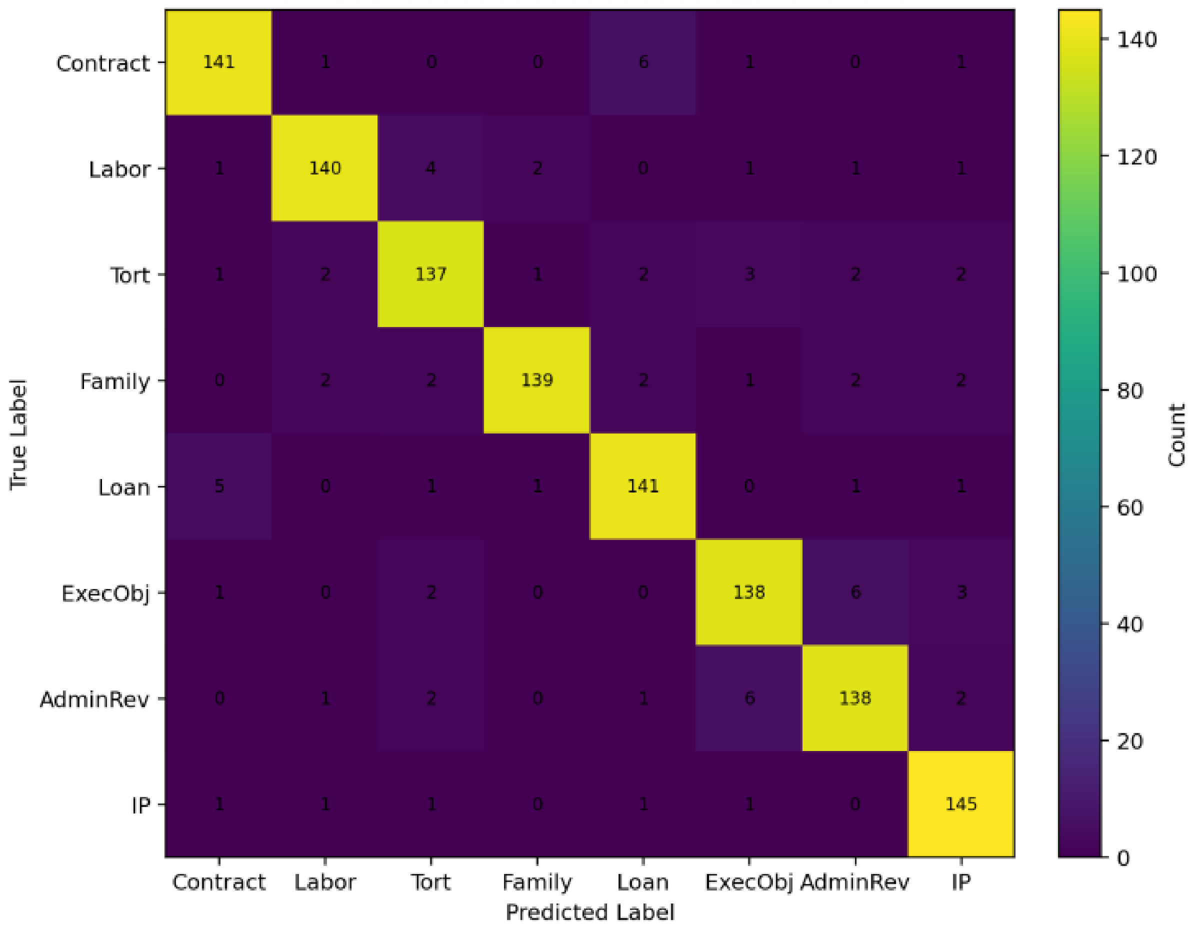

Figure 4: Confusion matrix for case type prediction

Fig. 4 shows that the main diagonal keeps a high value, indicating that most of the documents can be accurately mapped to the correct cause of action. The errors are mainly concentrated between two similar groups: contract disputes and loan disputes, execution objections and administrative reconsideration. The former group contains high overlapping fields such as creditor's rights and debts, repayment agreement and performance responsibility, while the latter group shares procedural expressions such as execution procedures, objection reasons and review basis, so it is easier to form partial confusion. Even so, the proportion of contract disputes misjudged as loan disputes is still controlled at 3.8%, and the proportion of enforcement objections misjudged as administrative reconsideration is 4.1%. The cross error between other categories is relatively small, especially the boundary between intellectual property and marriage and family cases is the clearest. This result indicates that the model has a strong ability to distinguish the nearest neighbor categories while maintaining high overall accuracy.

In order to further observe the reasoning stability of the model under different document lengths, Fig. 5 shows the distribution of the single-article reasoning time for the three length interval samples in the test set. This distribution can be used to determine whether the model has significant delay jitter under the condition of long text, and whether the inference cost gets out of control with the growth of length.

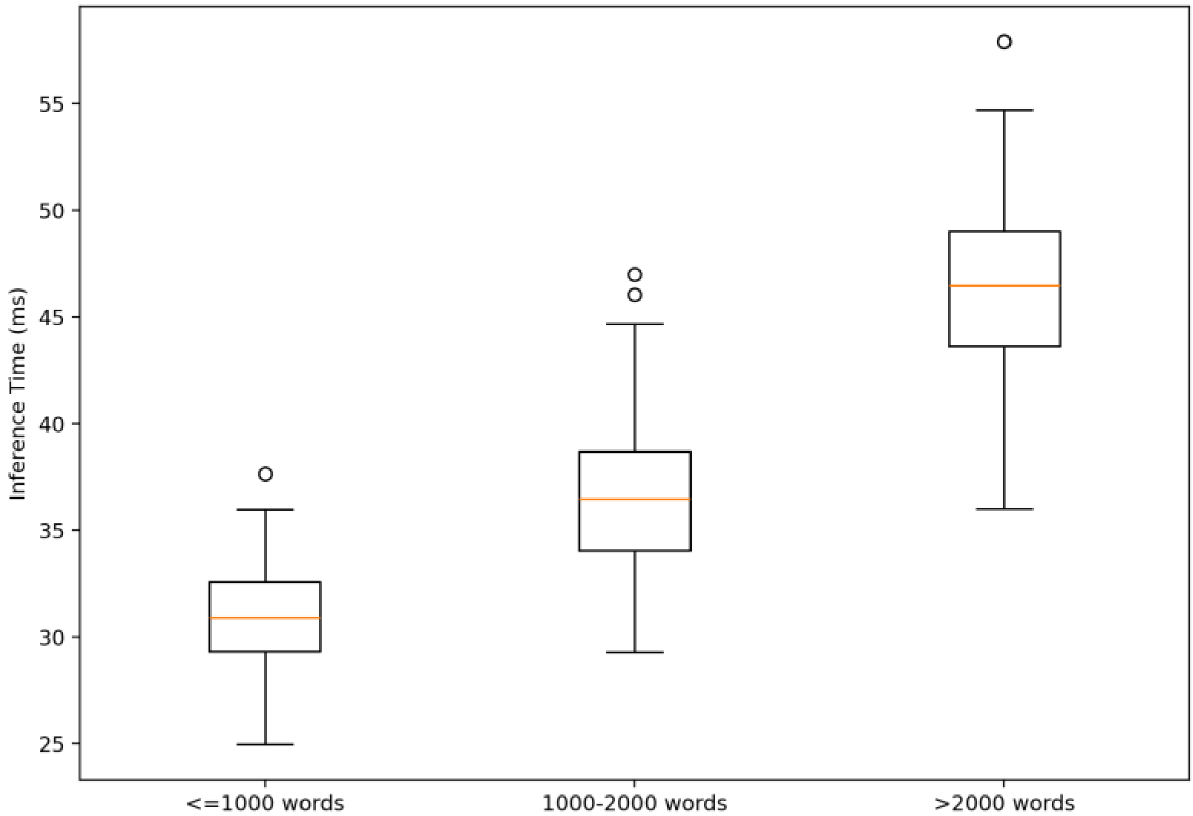

Figure 5: Distribution of interval inference times for different instrument lengths

Fig. 5 shows that the median inference time is 31 ms for samples with length up to 1000 words; When the length is between 1000 and 2000 words, the median time rises to 37 ms. The median time for long documents with more than 2000 words was 46 ms. The interquartile ranges of the three groups of samples are relatively compact, and the number of outliers is small, indicating that the reasoning overhead of the model changes smoothly with the growth of the text length, and there is no obvious delay diffusion under the condition of long documents. This result is consistent with the overall efficiency metrics in Table 1, indicating that shared encoding with two-branch inference still has good time controllability in long text scenarios. For judicial information systems, this stationary distribution means that the model can adapt to multiple business links such as batch archiving, online retrieval and auxiliary labeling, and the overall processing rhythm will not be slowed down by a single long document.

After the efficiency analysis, it is necessary to further confirm which modules the performance improvement comes from. Table 2 presents the ablation experiment results of the model, which is used to compare the contribution of cross-layer shared encoding, dual-branch cooperation, and type constraint correction to the final output quality. This table focuses on differences in the role of the internal structure of the system and complements the previous overall and category results.

Table 2: Results of ablation experiments

Model Configuration	Element Extraction Micro-F1 / %	Element Extraction Macro-F1 / %	Boundary Consistency Rate / %	Type Accuracy / %	Type Macro-F1 / %
Full Model	93.4	91.8	92.7	92.6	91.1
Without Type-Constrained Correction	91.6	89.9	90.8	91.3	89.8
Without Cross-Layer Shared Encoding	90.9	88.7	89.5	90.8	88.9
Extraction Branch Only	91.1	89.1	88.7	—	—
Classification Branch Only	—	—	—	89.8	88.1

Table 2 shows that after removing the type constraint correction module, Micro-F1 of feature extraction decreases from 93.4% to 91.6%, and the boundary consistency rate decreases to 90.8%, indicating that case type prior writeback has a direct effect on field boundary stability and label consistency. After removing the cross-layer shared encoding, Macro-F1 decreases the most, indicating that the collaboration between the three-level information of sentence unit, paragraph and document is the main source of model performance improvement. If only the single-task classification branch was retained, the Accuracy of case type was reduced to 89.8%. If only the single task extraction branch is retained, the boundary agreement rate drops to 88.7%. These changes show that the advantages of the proposed method come from the continuous linkage among multi-layer representation, two-branch reasoning and result revision, rather than relying on a single local module.

In order to compare the performance structure between different methods from a more comprehensive perspective, Fig. 6 uses a radar chart to show the relative performance of the proposed model, BiLSTM-CRF, Legal-BERT and single-task Transformer in four dimensions of field F1, type Accuracy, boundary agreement rate and inference efficiency.

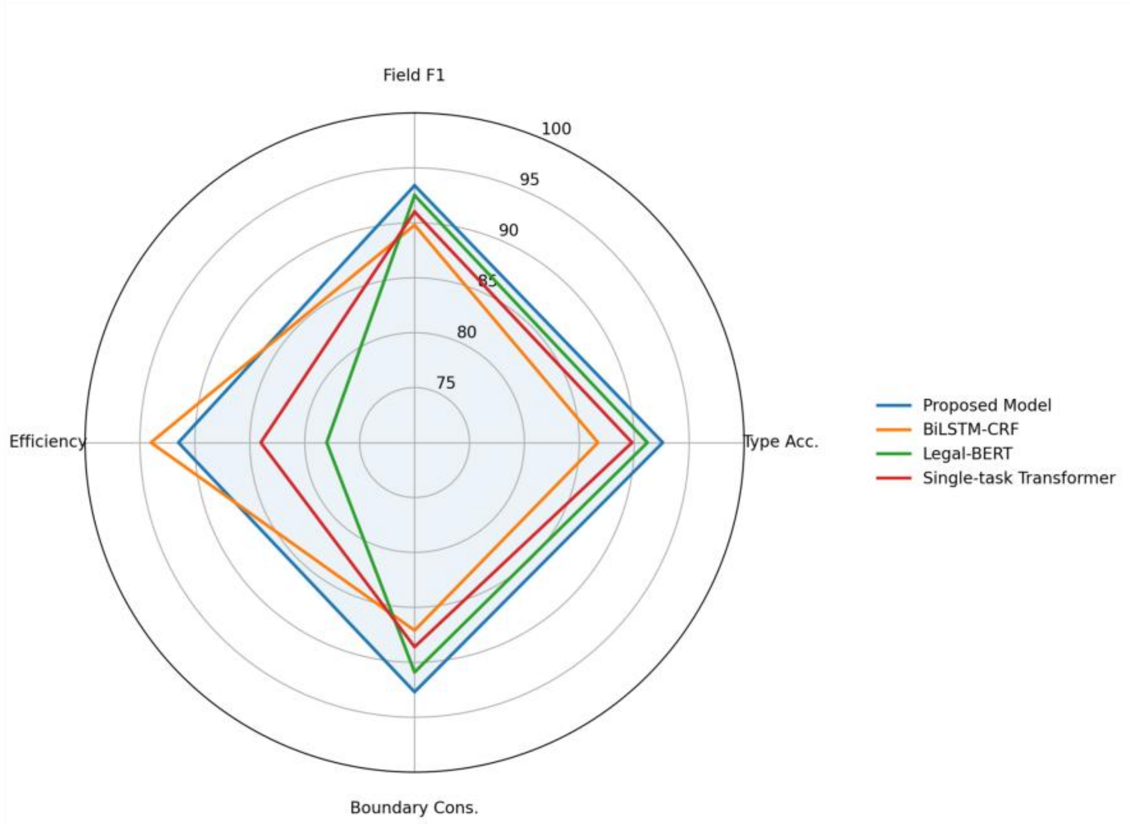

Figure 6: Comparison of comprehensive performance of different methods

Fig. 6 shows that the proposed model forms the most lateral contour in all four dimensions, and the advantages are particularly obvious in the boundary agreement rate and type Accuracy. Legal-BERT is close to the proposed model in field recognition ability, but slightly weaker in inference efficiency and category stability. BiLSTM-CRF has fast inference speed, but due to the lack of document-level modeling, its category discrimination ability is insufficient under the condition of long documents. The single-task Transformer has a certain foundation in type prediction, but due to the lack of fine-grained supervision brought by field branches, the structured output quality is significantly lower than that of the dual-task model. Fig. 6 shows that the proposed method does not only achieve local gains in a single index, but forms a more balanced comprehensive performance among field extraction, category discrimination, boundary control and system efficiency.

After the comparison of performance, architecture and method, it is necessary to evaluate the training and testing cost from the perspective of system scaling. Table 3 records the change of training time, single test time and video memory occupancy when the sample size is gradually expanded, which is used to observe the running trend of the model under the condition of increasing data amount.

Table 3: Training and testing overhead with data size scaling

Data Scale / Samples	Training Time / h	Inference Time per Sample / ms	GPU Memory Usage / GB
12000	9.6	38	4.3
18000	13.7	41	4.6
24000	18.1	43	4.9

Table 3 shows that when the sample size increases from 12000 to 24000, the training time increases from 9.6 hours to 18.1 hours, the single test time only increases from 38 ms to 43 ms, and the video memory occupation increases from 4.3 GB to 4.9 GB. The training cost increases with the increase of sample size, which is consistent with the general law of deep models in multiple rounds of parameter updates. The delay increase in the testing phase remains in a small range, indicating that the model complexity is mainly concentrated in the training end, while the online inference cost control is relatively stable. This overhead structure is more suitable for practical application scenarios such as legal document filing, cause of action screening and assisted retrieval, because the system can update parameters through offline training and then support high-frequency calls with low latency.

The comprehensive results show that the proposed method has formed a relatively stable performance structure between long text processing, field extraction, case classification and operation efficiency of legal documents. The overall metrics, category performance, error distribution, module contribution, and expansion cost are consistent with each other, and no single metric stands out while other metrics are significantly unbalanced. The results show that the three parts of multi-layer feature representation, two-branch task modeling and type constraint correction form an effective linkage in the system, so that the model can not only generate high quality structured fields, but also complete stable case type prediction, and maintain an acceptable delay level in the inference stage.

4.3 Discussion

Experiments show that the extraction of legal document elements and case type prediction is not a simple series relationship, the quality of field recognition will directly affect the semantic compression effect of the text, and the result of category discrimination will reverse the constraint on the closure of the boundary of key fields such as amount, law, and judgment result. The complete model maintained a relatively stable performance in eight types of cases, indicating that the multi-layer feature representation can absorb the three-level information of sentence elements, paragraphs and documents, and avoid the traction of local high-frequency words in long texts on the overall judgment. After the modification of type constraints, the cross offsets between the close neighbor categories such as contract disputes and loan disputes, enforcement objections and administrative reconsideration are significantly contracted, indicating that the category prior has a corrective effect on the structured output. From the perspective of system application, the value of this method is not only reflected in the improvement of indicators, but also reflected in the output form more suitable for legal information system calls. Field tables, category labels, and confidence scores can be directly fed into archiving, retrieval, screening, and auxiliary labeling processes, reducing duplicate entry and manual curation costs. The reasoning time remains stable with the increase of the length of the document, which indicates that shared encoding and two-branch reasoning have better scalability in engineering implementation. On the whole, the method in this paper establishes a relatively stable technical connection between structured parsing, discourse classification and result consistency, which provides a landing calculation path for the intelligent processing of judicial texts. At the same time, the heat map, confusion matrix and ablation results are consistent, indicating that the source of performance improvement is clear, and the model behavior has a good interpretable basis.

5 Conclusion

Focusing on the task of legal document element extraction and case type prediction, this paper constructs a computational framework combining multi-layer feature representation, two-branch task modeling and type constraint modification. Experimental results show that the proposed method can maintain high accuracy of field identification and stability of case type discrimination in long judicial texts, and meet the requirements of legal information system in reasoning efficiency. The shared coding layer enhances the semantic preservation ability of the discourse, the dual-task structure strengthens the linkage between local fields and global categories, and the type constraint correction further improves the boundary closure, law compatibility and label consistency, making the structured output more suitable for archivating, retrieval and auxiliary analysis processes. There are still several limitations in this paper. First, the data sources mainly focus on public Chinese legal documents, and the ability to transfer across regions, institutions and languages still needs to be expanded. Second, the distribution of few-sample fields in complex cases is still insufficient, and the stability of low-frequency tags in long-tail scenarios still has room for improvement. Third, current revision mechanisms are mainly based on statistical correlation and discourse semantics, and fine-grained law networks and program rule graphs have not been incorporated into unified reasoning. Subsequent research will focus on multilingual legal document adaptation, knowledge graph enhanced reasoning, lightweight deployment, and human-machine collaborative verification, so as to improve the generalization ability, interpretability, and engineering usability of the model. At the same time, confidence hierarchical scheduling and manual review feedback reflux mechanism will be introduced in the future to enhance the operation stability and deployment adaptability in high-risk document scenarios.

Author's Profile

Zhicheng Huang, a native of Wuhan, China, is currently pursuing a JD degree at Case Western Reserve University School of Law in the United States and has obtained the California Attorney Qualification. Previously, I served as a partner at Wuhan Hua Ping Law Firm, where I have been deeply engaged in legal practice for many years, handling numerous criminal, civil and commercial, and litigation cases, accumulating solid professional experience and industry resources.

References

[1] Chen H, Wu L, Chen J, et al. A comparative study of automated legal text classification using random forests and deep learning[J]. Information Processing & Management, 2022, 59(2): 102798.

[2] Enamoto L, Santos A R A S, Maia R, et al. Multi-label legal text classification with BiLSTM and attention[J]. International Journal of Computer Applications in Technology, 2022, 68(4): 369-378.

[3] Alcántara Francia O A, Nunez-del-Prado M, Alatrista-Salas H. Survey of text mining techniques applied to judicial decisions prediction[J]. Applied Sciences, 2022, 12(20): 10200.

[4] Rabelo J, Goebel R, Kim M Y, et al. Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021[J]. The Review of Socionetwork Strategies, 2022, 16(1): 111-133.

[5] Cao Y, Sun Y, Xu C, et al. CAILIE 1.0: A dataset for Challenge of AI in Law-Information Extraction V1. 0[J]. AI Open, 2022, 3: 208-212.

[6] Cui J, Shen X, Wen S. A survey on legal judgment prediction: Datasets, metrics, models and challenges[J]. IEEE Access, 2023, 11: 102050-102071.

[7] Zhao Q, Gao T, Guo N. LA-MGFM: A legal judgment prediction method via sememe-enhanced graph neural networks and multi-graph fusion mechanism[J]. Information Processing & Management, 2023, 60(5): 103455.

[8] Zahir J. Prediction of court decision from a rabic documents using deep learning[J]. Expert Systems, 2023, 40(6): e13236.

[9] Almuzaini H A, Azmi A M. TaSbeeb: A judicial decision support system based on deep learning framework[J]. Journal of King Saud University-Computer and Information Sciences, 2023, 35(8): 101695.

[10] Lage-Freitas A, Allende-Cid H, Santana O, et al. Predicting Brazilian court decisions[J]. PeerJ Computer Science, 2022, 8: e904.

[11] Bi S, Zhou Z, Pan L, et al. Judicial knowledge-enhanced magnitude-aware reasoning for numerical legal judgment prediction[J]. Artificial Intelligence and Law, 2023, 31(4): 773-806.

[12] Sukanya G, Priyadarshini J. Modified Hierarchical-Attention Network model for legal judgment predictions[J]. Data & knowledge engineering, 2023, 147: 102203.

[13] Bertalan V G F, Ruiz E E S. Using attention methods to predict judicial outcomes[J]. Artificial Intelligence and Law, 2024, 32(1): 87-115.

[14] Sun J, Huang S, Wei C. Chinese legal judgment prediction via knowledgeable prompt learning[J]. Expert Systems with Applications, 2024, 238: 122177.

[15] Tong S, Yuan J, Zhang P, et al. Legal Judgment Prediction via graph boosting with constraints[J]. Information Processing & Management, 2024, 61(3): 103663.

[16] Castano S, Ferrara A, Furiosi E, et al. Enforcing legal information extraction through context-aware techniques: The ASKE approach[J]. Computer Law & Security Review, 2024, 52: 105903.

[17] Bellandi V, Bernasconi C, Lodi F, et al. An entity-centric approach to manage court judgments based on natural language processing[J]. Computer Law & Security Review, 2024, 52: 105904.

[18] Lu R, Li L. Named entity recognition method of Chinese legal documents based on parallel instance query network[J]. International Journal of Digital Crime and Forensics (IJDCF), 2024, 16(1): 1-19.

[19] Yulianti E, Bhary N, Abdurrohman J, et al. Named entity recognition on Indonesian legal documents: a dataset and study using transformer-based models[J]. International Journal of Electrical & Computer Engineering (2088-8708), 2024, 14(5).

[20] Guimarães G M C, Silva F X B, Macêdo L A B, et al. Legal document segmentation and labeling through named entity recognition approaches[J]. Journal of Information and Data Management-JIDM, 2024, 15(1): 123-131.