applied sciences

MDPI

Article

Deep Learning-Based Automatic Summarization of Chinese Maritime Judgment Documents

Lin Zhang *, Yanan Li and Hongyu Zhang

School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China; jemmalee2019@outlook.com (Y.L.); yanghan@dlmu.edu.cn (H.Z.)
* Correspondence: zhanglin@dlmu.edu.cn

Abstract: In the context of China's accelerating maritime judicial digitization, automatic summarization of lengthy and terminology-rich judgment documents has become a critical need for improving legal efficiency. Focusing on the task of automatic summarization for Chinese maritime judgment documents, we propose HybridSumm, an "extraction–abstraction" hybrid summarization framework that integrates a maritime judgment lexicon to address the unique characteristics of maritime legal texts, including their extended length and dense domain-specific terminology. First, we construct a specialized maritime judgment lexicon to enhance the accuracy of legal term identification, specifically targeting the complexity of maritime terminology. Second, for long-text processing, we design an extractive summarization model that integrates the RoBERTa-wwm-ext pre-trained model with dilated convolutional networks and residual mechanisms. It can efficiently identify key sentences by capturing both local semantic features and global contextual relationships in lengthy judgments. Finally, the abstraction stage employs a Nezha-UniLM encoder–decoder architecture, augmented with a pointer–generator network (for out-of-vocabulary term handling) and a coverage mechanism (to reduce redundancy), ensuring that summaries are logically coherent and legally standardized. Experimental results show that HybridSumm's lexicon-guided two-stage framework significantly enhances the standardization of legal terminology and semantic coherence in long-text summaries, validating its practical value in advancing judicial intelligence development.

Keywords: maritime judgment lexicon; Chinese maritime judgment documents; extractive summarization; abstractive summarization; HybridSumm

check for updates

Academic Editors: Nadia Nedjah and Luiza de Macedo Mourelle

Received: 27 March 2025
Revised: 1 May 2025
Accepted: 7 May 2025
Published: 13 May 2025

Citation: Zhang, L.; Li, Y.; Zhang, H. Deep Learning-Based Automatic Summarization of Chinese Maritime Judgment Documents. *Appl. Sci.* **2025**, *15*, 5434. https://doi.org/10.3390/app15105434

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As a major maritime nation with extensive strategic interests, China has prioritized building a strong maritime power as a core national strategy, driven by the advancement of the Belt and Road Initiative and the rapid growth of the modern marine economy. Maritime trial, as a key judicial mechanism to safeguard this strategic development, has increasingly demanded specialized and intelligent updates. In recent years, China has been advancing the digital transformation of its judicial system, emphasizing the integration of cutting-edge technologies like artificial intelligence (AI) into trial processes to enhance efficiency and transparency. Within this context, the intelligent processing of maritime judgment documents has emerged as a critical task requiring urgent breakthroughs.

Maritime judgment documents (primarily judgments and rulings) exhibit distinct professional characteristics: excessive length (often spanning dozens of pages), dense legal terminology, and rigorous structure. Traditional manual summarization, plagued by low

efficiency and high costs, struggles to cope with rapidly growing caseloads. Consequently, developing automatic summarization methods for maritime judgment documents has emerged as a critical challenge.

Automatic text summarization, a core NLP technology that generates concise and coherent summaries capturing document essentials, has been widely used in news and academic domains. In recent years, this technology has extended to legal text processing, offering innovative pathways for the efficient extraction of legal information. However, existing approaches exhibit two critical limitations: first, they predominantly focus on surface-level linguistic patterns, failing to adequately model deep semantic relationships—such as implicit legal element logic and statutory provision dependencies—in legal documents; second, they are primarily tailored to civil and commercial judgment documents, leaving the specialized domain of maritime judgment documents under-researched. The automatic summarization of maritime judgment documents confronts dual technical barriers: (1) linguistic heterogeneity: judges' individualized writing styles amplify the unstructured nature of legal texts, complicating content parsing and key information extraction; and (2) semantic complexity: high-density domain terminology and complex case details demand advanced capabilities in both discourse-level semantic modeling and legal logical reasoning. To address these challenges, this paper leverages deep learning's prowess in long-text semantic analysis to bridge legal studies and artificial intelligence, exploring intelligent summarization techniques using the publicly available maritime judgment documents. The goal is to enhance judicial information processing efficiency, supporting the Belt and Road Initiative and maritime power strategy.

The main contributions of this paper are as follows: (1) to address the complexity and specialization of maritime judgment documents, we constructed a maritime judgment lexicon by extracting terms such as laws, regulations, and international conventions in real legal documents. This lexicon resolves terminological ambiguities caused by conventional tokenization tools and improves term consistency and legal compliance in the generated summaries. (2) Targeting the challenge of lengthy maritime judgments, we propose HybridSumm, a hybrid "extraction–abstraction" summarization framework based on the maritime judgment lexicon. The term "extraction–abstraction" refers to the framework's integration of the information filtering capability of extractive models with the semantic reorganization advantages of abstractive approaches. Specifically, in the extraction stage, the RoBERTa-DCNN-GR model (see Section 4.1) is employed to select key sentences from lengthy judgment documents. It uses a RoBERTa-wwm-ext-based semantic encoder, multi-layer dilated convolutional neural networks (DCNNs) and gated residual mechanisms (GR) for precise long-text compression to ensure complete retention of legally critical facts. In the abstraction stage, the Nezha-UniLM model (see Section 4.2) refines the semantic representation of these extracted key sentences. Nezha's encoder, equipped with relative positional encoding, systematically analyzes complex legal text structures to capture hierarchical dependencies, whereas the UniLM decoder generates coherent sequences through dynamic attention masking, ensuring contextual consistency. To address technical challenges such as out-of-vocabulary (OOV) terms and repetitive generation, a pointer–generator network is integrated, with its copy mechanisms directly handling OOV issues and a coverage mechanism actively suppressing redundancy. These components collectively transforms key sentence summaries into logically coherent and legally standardized outputs. As illustrated in Figure 1, this collaborative architecture effectively balances information integrity and expressive fluency. Detailed technical implementations of each component are elaborated in Section 4.

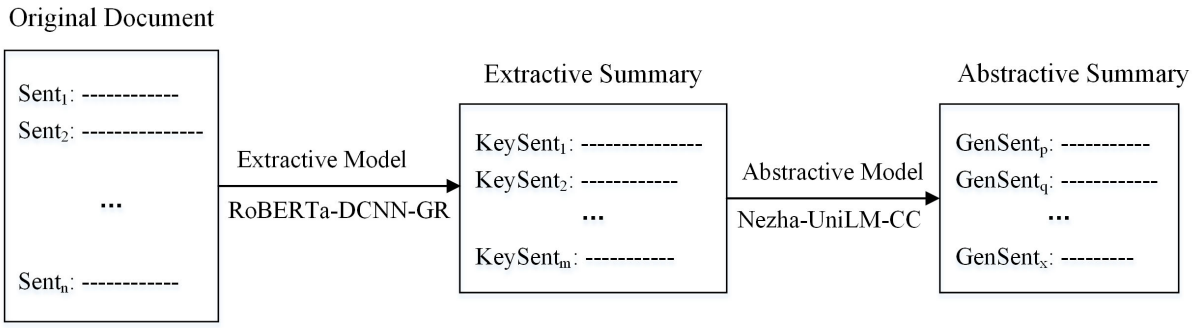


Figure 1. Extraction–abstraction two-stage framework. "-" represents a Chinese character in a sentence.

The rest of the paper is organized as follows. Section 2 describes the related work. Sections 3 and 4 present our methodology and design in detail. Section 5 shows the experiments, results and analysis. Section 6 concludes the paper.

2. Related Work

The pioneering work in automatic text summarization traces back to Luhn's statistical frequency approach in 1958 [1]. His algorithm operates through three steps: quantifying term significance via within-document frequency analysis, scoring sentences by combining the frequency and distribution of these salient terms, and selecting the highest-scoring sentences from technical documents to form summaries. Building on this foundation, summarization research has since evolved into two dominant paradigms: extractive and abstractive [2].

2.1. Extractive Summarization

Extractive summarization methods treat sentences as fundamental processing units, rank them through salience scoring mechanisms, and compile the top-ranked candidates into a summary [1].

Early studies on extractive summarization primarily employed statistical approaches that utilized features such as term frequency, sentence position, cue words, and document titles to compute sentence importance [3]. LEAD-3—one of the earliest statistical feature-based extractive methods—generates summaries by extracting the first three sentences of a document. Despite its simplicity, LEAD-3 demonstrates high effectiveness for news article summarization, where the opening sentences typically contain the most salient information [4]. Edmundson [5] expanded this paradigm by scoring sentences using four statistical criteria: cue words, key words, title and heading words, and sentence location, then selecting top-ranked sentences to form summaries. Lloret and Palomar [6] introduced sentence weighting metrics combining term frequency, noun phrase length, and content word frequency within noun phrases, achieving competitive performance in news document summarization. Building on these works, PadmaLahari et al. [7] proposed a hybrid method integrating statistical features (e.g., keywords' TF-IDF weights, term frequency, sentence position, word length, weighted parts-of-speech (POS) tags) with linguistic features (e.g., proper nouns, pronouns), employing a successive threshold to optimize high-weight sentence extraction.

Later, inspired by the manual document drafting process, where core topics are first identified and then expanded with relevant words, researchers developed topic model-based extractive summarization methods. These approaches map documents into a semantic space, uncover latent semantic relationships (e.g., topics) between documents and words through term frequency analysis, and select sentences that optimally represent these topics

as summaries [8]. Gong and Liu [9] proposed an extractive summarization method using the latent semantic analysis (LSA) topic model to identify semantically salient sentences. The method applies singular value decomposition (SVD) to the sentence–term matrix, producing singular vectors and corresponding index values. Each singular vector corresponds to a latent document topic, with the index value representing the optimal sentence score in that topic dimension. Sentences with the largest index values from those possessing the most significant singular vectors are selected to generate the summary. Kar et al. [10] proposed a method for automatically summarizing changes in dynamic text collections. Their approach uses the latent Dirichlet allocation (LDA) topic model to identify latent topic structures of changes, computes term-level and sentence-level latent topic scores, and selects sentences with maximum topic relevance for summary generation.

Inspired by Google's PageRank algorithm for web page ranking [11], researchers proposed graph-based extractive summarization methods. These approaches model sentences as nodes and their relationships as edges in a graph, rank them iteratively based on connectivity, and select top-ranked sentences (basically centroids in the graph) from source documents to form summaries. These methods posit that a sentence's importance is determined by its interrelationships with others in the document (or document collection). The more a sentence is supported by others, the higher its importance [12]. TextRank [13] and LexRank [14] are two classic graph-based extractive summarization algorithms that differ primarily in their sentence similarity computation mechanisms. TextRank measures similarity through word overlap (i.e., the count of co-occurring words between sentences), while LexRank uses TF-IDF-weighted cosine similarity. Huang and Liu [15] proposed a weighted TextRank algorithm for Chinese single-document summarization by leveraging Word2Vec embeddings to generate sentence vectors. Their approach constructs inter-sentence influence weights through three criteria: cosine similarity between sentences, keyword coverage, and similarity between sentence and title, significantly improving the quality of generated summaries. Liu and Zhang [16] developed an improved Chinese single-document summarization method by integrating the LexRank algorithm with the vector space model (VSM), achieving improved summarization performance. This approach employs LexRank's graph-based centrality to compute inter-sentence similarity while leveraging VSM to measure inter-paragraph similarity. Sentence weights are computed by synthesizing multiple factors, including paragraph salience, sentence centrality, and cue words. Al-Khassawneh et al. [17] introduced a hybrid graph-based method for single-document extractive summarization. This method employs a hybrid similarity function—comprising cosine similarity, Jaccard similarity, word alignment-based similarity, and sliding window-based similarity—to calculate edge weight between sentence nodes. It combines sentence centrality (measured by node degree) with six sentence-level features (title words, sentence length, sentence position, numerical data, thematic words, and inter-sentence similarity) to compute sentence scores. Finally, triangle sub-graphs are constructed to optimize sentence selection for summary generation.

Summaries are typically concise. To enhance coverage while minimizing redundancy among selected sentences, researchers proposed constraint-based extractive summarization methods. The maximal marginal relevance (MMR) algorithm [18], a widely used technique for redundancy reduction and query-based summarization, prioritizes sentences that maximize relevance to user queries while minimizing similarity to previously selected summary content. Al-Khassawneh et al. [17] combined sentence scoring with the MMR algorithm during triangle sub-graph-based summary generation to minimize redundancy among extracted sentences. Pozzi et al. [19] proposed a domain-specific method integrating BERT sentence embeddings, agglomerative clustering algorithms, and MMR to summarize news related to cryptocurrency, blockchain, and decentralized finance. Yin and Pei [20]

proposed DivSelect, an optimization-driven sentence selection algorithm for extractive document summarization, which balances sentence salience and diversity by selecting sentences with high salience but dis-similar with each other to form summaries.

In the 1990s, with the application of machine learning methods—particularly supervised machine learning—to natural language processing (NLP), researchers proposed machine learning-based extractive summarization methods. These approaches formulate sentence selection for summaries as a binary classification problem, categorizing each sentence as either a summary or non-summary sentence [21]. Kupiec et al. [22] poineered the use of supervised machine learning algorithms in text summarization by introducing a naive Bayesian classifier that calculates sentence selection probabilities based on features such as sentence length cut-off, fixed phrases, paragraphs, thematic words, and uppercase words. Their method generates summaries of specified lengths by extracting sentences ranked by these probabilities. This work demonstrated machine learning's potential for extractive summarization and established foundational principles for subsequent research in machine learning-based approaches. Zhang et al. [23] developed a summarization framework for broadcast news and lecture speech by employing different feature combinations—acoustic, structural, and lexical features—to represent sentences. They utilized a support vector machine (SVM) classifier to assess sentence inclusion likelihood based on these combined representations. Conroy and O'Leary [24] employed a hidden Markov model (HMM) for probabilistic sentence selection, utilizing three features: sentence position, term frequency within the sentence, and the probability of a sentence term estimated from the document terms. In addition to SVM and HMM, other supervised machine learning approaches have also been applied to extractive summarization, including maximum entropy (MaxEnt) [25,26], logistic regression [27], and conditional random fields (CRF) [26].

With the exponential growth of online documents, deep learning models inspired by biological neural networks have emerged in machine learning and garnered significant attention across multiple domains in recent years, including text summarization. These deep learning-based extractive summarization methods automatically learn hierarchical semantic features of texts to evaluate sentence importance. In the aforementioned study by Yin and Pei [20], their convolutional neural network-based language model (CNNLM) integrates local contextual and global semantic information to generate context-sensitive sentence vectors (i.e., sentence semantic encodings), which provide foundational representations for subsequent high-value sentence selection through the information-divergence-based DivSelect algorithm. Nallapati et al. [28] developed SummaRuNNer, a sequence classifier built upon recurrent neural networks (RNNs). This model employs a gated recurrent unit (GRU)-enhanced RNN architecture to enable end-to-end summary sentence identification, incorporating multidimensional criteria including content richness, salience, novelty, as well as absolute and relative positional importance.

2.2. Abstractive Summarization

Abstractive summarization fundamentally differs from extractive methods. Unlike extractive approaches that select existing sentences, abstractive methods aim to semantically comprehend document content and generate novel expressions that concisely encapsulate core ideas, thereby producing human-readable summaries [29]. To achieve this, they employ advanced natural language generation (NLG) techniques such as sentence compression, sentence fusion, concept fusion and generalization, and lexical paraphrasing [30,31].

Embar et al. [32] developed a domain-specific template-based approach for abstractive summarization of Kannada language documents. They used abstraction schemes and information extraction rules to identify the relevant content, populated predefined

templates with the extracted data, and generated coherent sentences from these templates to produce the final summaries. Kurisinkel et al. [33] proposed a tree-based abstractive multi-document summarization approach. Their method involves parsing input text into a collection of syntactic dependency trees, employing a maximum density sub-graph cutting algorithm to select variable-sized partial dependency trees representing salient knowledge granules, and utilizing a transition-based syntactic linearization model to combine these granules into coherent summary sentences. Khan et al. [34] proposed a semantic graph-based abstractive multi-document summarization method. They first constructed a semantic graph with nodes defined as predicate argument structures (PASs) representing sentence-level semantics, and edges weighted by semantic similarity between PASs. Node relationships were augmented through PAS–document and PAS–document set relationships. A hybrid ranking strategy combining a graph-based ranking algorithm (e.g., PageRank) and the MMR algorithm prioritized salient PASs while reducing redundancy. Finally, NLG techniques transformed the top-ranked PASs into fluent summary sentences.

With the rapid advancement of computational capabilities, particularly through the widespread adoption of high-performance computing devices such as GPUs and TPUs, deep learning-based approaches for automatic sentence-level summarization have achieved substantial progress. Among these, the Seq2Seq (sequence-to-sequence) model, proposed by Cho et al. [35] and Sutskever et al. [36] for machine translation, catalyzed breakthroughs in abstractive text summarization techniques. The Seq2Seq model employs an encoder–decoder neural architecture: the encoder compresses the source text into a fixed-length context vector, which the decoder then iteratively decodes to generate the target output sequence.

Rush et al. [37] were the first to introduce the Seq2Seq model to abstractive summarization tasks in 2015 and proposed an attention-based summarization (ABS) model. ABS employed an attention-based encoder to generate context-aware semantic embeddings, paired with a feed-forward neural decoder using beam search for sentence generation. Building on this foundation, Chopra et al. [38] introduced the recurrent attentive summarizer (RAS) in 2016, which replaced the feed-forward decoder with an attentive recurrent neural network (RNN) and augmented the encoder with convolutional layers to capture local context dependencies, thereby achieving significant performance improvements. Nallapati et al. [39] proposed an RNN-based encoder–decoder architecture. The encoder incorporated linguistic features such as POS tags, named entity tags, and discretized TF and IDF values to enhance keyword recognition. For decoding, they introduced a switching generator–pointer model to handle OOV words. Additionally, a hierarchical attention mechanism was implemented to model document structure through word-level and sentence-level context interactions, thus improving document representation. Tan et al. [40] proposed a hierarchical long short-term memory (LSTM)-based encoder–decoder framework for abstractive summarization. The encoder processes words through a word-level LSTM, followed by a sentence-level LSTM to encode sentences of a document. To identify salience content, they introduced a graph-based attention mechanism inspired by PageRank, which identifies key sentences through graph centrality scoring. During decoding, a distraction mechanism [41] is applied to penalize redundant content generation. Additionally, a hierarchical decoder with a reference mechanism is designed to improve summary fluency and factual accuracy. The Google research team proposed PEGASUS [42], a pre-trained model based on the Transformer encoder–decoder architecture, specifically designed for abstractive text summarization. This model aims to enhance the ability to capture key information and generate fluent, human-like summaries. Pozzi et al. [43] developed an imitation learning-based knowledge distillation framework to mitigate exposure bias in sequence generation tasks.

Text representation is a critical step in automatic text summarization, as the quality of representation directly determines the accuracy and completeness of generated summaries. Traditional summarization methods typically rely on bag of words (BoW) frameworks—such as VSM and TF-IDF weighting schemes—to convert textual content into numerical vectors. While these BoW-based approaches can capture statistical lexical patterns, they inherently fail to model word order and contextual dependencies. This limitation leads to the loss of essential syntactic structures and semantic nuances present in the source document, thereby constraining the model's ability to produce summaries with contextual coherence and semantic precision.

The recent rapid advancement of deep learning technologies, particularly the emergence of pre-trained language models (PLMs), has revolutionized text representation and generation. PLMs are artificial intelligent models that learn deep linguistic features and contextualized representations through self-supervised pre-training on large-scale text corpora. Notable implementations such as GPT (generative pre-training) [44] and BERT (Bidirectional Encoder Representations from Transformers) [45] have demonstrated state-of-the-art performance across NLP tasks, including automatic text summarization. By learning generalizable language patterns, these models generate semantically coherent and context-aware outputs that significantly outperform traditional methods. In 2019, Liu [46] proposed BERTSUM, an extractive summarization model that builds on the BERT architecture to enable more precise sentence extraction. Subsequently, Liu and Lapata [47] further developed a unified framework integrating abstractive methods, significantly advancing the generalization capabilities of summarization techniques. In the same year, Radford et al. [48] introduced GPT-2, an enhanced version of GPT trained on large-scale text corpora with expanded parameters. The model demonstrated unprecedented linguistic generation capabilities and context-aware modeling, achieving state-of-the-art performance in open-domain text generation tasks. In 2020, Kieuvongngam et al. [49] employed the GPT-2 model to generate abstractive summaries for COVID-19-related medical research articles. By producing summaries with enhanced readability and sufficient information coverage, this work provided strong support for the World Health Organization (WHO) and medical communities in developing treatments and measures against COVID-19. This study demonstrated that deep learning-based abstractive summarization approaches are not only theoretically advanced, but also exhibit substantial potential in real-world applications.

2.3. Automatic Summarization of Judgment Documents

A judgment document, as a legal instrument, requires its summarization to not only extract the core content, but also strictly adhere to the professional and normative standards inherent in legal texts, ensuring the precise representation of legal provisions and uncompromising accuracy of statutory language.

Early studies on automatic summarization of judgment documents primarily employed discourse structure-based extractive methods. These approaches first divide a judgment document into several functionally distinct discourse units, then utilize statistical analysis or machine learning algorithms to identify key sentences within each unit. The selected sentences are subsequently recomposed into summaries while preserving the original logical structure [50]. Grover et al. [51] categorized sentences in UK House of Lords judgments into seven rhetorical roles: FACT, PROCEEDINGS, BACKGROUND, PROXIMATION, DISTANCING, FRAMING, and DISPOSAL. By analyzing linguistic features such as tense, POS, and cue phrases, they identified role-specific sentences for abstractive summary generation. Farzindar and Lapalme [52] segmented the discourse structure of judgments from the Federal Court of Canada into four thematic components: INTRODUCTION, CONTEXT, JURIDICAL ANALYSIS, and CONCLUSION. By integrating posi-

tional features with TF-IDF weighting, they developed an extraction framework to select theme-specific sentences, ultimately generating table-style summaries that preserve legal discourse coherence.

As research continues, a number of scholars believe that legal knowledge plays an important role in the summarization of legal documents. Galgani et al. [53] developed a knowledge base comprising 23 manually crafted rules and integrated it with automatic summarization techniques to generate summaries for case reports from the Australasian Legal Information Institute (AustLII). Duan et al. [54] proposed a controversy focus-based extractive summarization method for civil trial court debate records. They found that controversy focuses constitute the core arguments between parties from different camps (plaintiffs, defendants, witness, judge, etc.) during trials, with debate dialogues organized around these controversy focuses. To generate summaries, their method first categorizes sentences in debate records based on their alignment with identified controversial focuses, and then selects salient sentences from each category to construct summaries. Bhattacharya et al. [55] proposed DELSumm, an unsupervised extractive summarization algorithm for legal case documents from the Indian Supreme Court. The method incorporates domain knowledge from legal experts to assign weights to thematic segments, sentences, and content words within case documents. Using integer linear programming (ILP), DELSumm optimizes sentence selection for maximal informal coverage while integrating representations from all rhetorical segments to minimize redundancy in generated summaries.

With the rapid advancement of AI technology in recent years, researchers have begun employing deep learning methods to perform automatic summarization of legal documents. At the 2020 Legal AI Challenge (LAIC2020), Su et al. [56] proposed a BERT-based hybrid summarization model integrating extractive and abstractive approaches. The model achieved the highest ROUGE score of 61.43% on the LAIC2020 dataset, which comprises 9848 first-instance civil judgment documents. Zhou et al. [57] proposed BASR, an abstractive summarization model that integrates BERT, attention mechanisms, and reinforcement learning, tailored for civil judgment documents related to small claims cases and summary procedures. Anand and Wagh [58] proposed an extractive summarization method for judgment documents from Indian Supreme Court. They labeled the importance of body sentences (1 for important, 0 for unimportant) by computing their similarity to headnote sentences. Subsequently, they utilized word embeddings and sentence embeddings to capture semantic features, and separately employed two classifiers—a feed-forward neural network (FFNN) and a LSTM—to extract salient sentences for summaries. Wei et al. [59] proposed a legal provision-enhanced abstractive summarization method for judgment documents. The method employs a bidirectional LSTM (BiLSTM) to encode legal provisions and incorporates the encoded legal context as external knowledge into an encoder–decoder framework, optimizing the preservation of legally critical elements while generating linguistically coherent summaries. Huang et al. [60] proposed a two-stage summarization model for judgment documents. Specifically, an LSTM network extracts key information in the first stage, while the second stage integrates the UniLM pre-trained model, copy mechanism, and attention mechanism to generate summaries. Experimental results demonstrate that this model achieves significant improvements in summarization performance on the CAIL2020 and LCRD datasets.

2.4. Literature Comments

Extractive summarization provides a faster and more straightforward approach to producing summaries compared to abstractive summarization. Extractive models generate summaries by directly selecting salient sentences from source texts, offering advantages in grammatical compliance and output controllability. However, they are susceptible to

redundant information in source texts, and the mechanical concatenation of extracted sentences often compromises summary coherence, thereby adversely affecting readability and information density. In contrast, abstractive models synthesize human-like summaries through semantic reconstruction, but face challenges in long-text processing—particularly in effectively capturing deep syntactic structures and word order patterns. These limitations may lead to semantic misinterpretations, grammatical errors, logical discontinuities, and information loss, ultimately constraining comprehensive quality improvement of abstractive summaries. Typical models such as BERTSUM (extractive) and PEGASUS (abstractive), while mature in general domains, are, respectively, constrained by the inherent limitations of extractive and abstractive paradigms, and neither has been tailored for optimization in domain-specific tasks such as legal document summarization.

As outlined in Section 2.3, a review of automatic summarization research on judgment documents reveals that international scholars predominately employ extractive summarization methods based on discourse structures or legal knowledge. These approaches, constrained by surface-level feature matching, often fail to capture underlying semantic relationships. While domestic research has achieved progress in summarizing civil and commercial judgment documents, automatic summarization for maritime judgments remains in an exploratory stage. This gap primarily stems from the domain-specific complexity of maritime cases. Maritime judgments exhibits the intricate structure, rigorous logic, and terminological density common to legal texts, while additionally involving cross-border jurisdictional elements such as the invocation of multinational legal provisions and applications of international conventions. Furthermore, cross-border maritime cases impose heightened demands on factual precision, legal argument completeness, and interpretive adaptability. These specialized requirements significantly increase the complexity of accurately comprehending and distilling key information from maritime judgments by automatic summarization models.

Based on these existing studies, we focus on maritime judgment documents publicly released by 11 maritime courts across China, addressing their characteristic challenges of excessive length, dense professional terminology, and complex logical structures. By integrating the strengths of extractive and abstractive summarization methods, we develop a deep learning-based framework for automatic summarization of Chinese maritime judgment documents, aiming to produce concise and coherent summaries that adhere to judicial norms.

3. Construction of Domain-Specific Lexicon for Maritime Judgment Documents Via Feature Analysis

3.1. Structural and Content Characteristics of Maritime Judgment Documents

Taking a first-instance judgment document for a dispute over contract of freight forwarding by sea as an example, we categorize the discourse structure of maritime judgments into six core segments through systematic analysis: cause of action, basic information, litigation proceedings, judicial decision, judicial personnel, and appendices.

Cause of action defines the legal nature and dispute type of the case. Based on 2023–2024 data from the China Maritime Trial Network (https://cmt.court.gov.cn/, accessed on 12 March 2025), Table 1 lists the top nine causes of action in maritime cases.

Basic information includes court details (court name, document type, case number), litigation parties (names/types/quantities of plaintiffs and defendants), and representation (law firms and attorneys involved).

Litigation proceedings records adversarial claims and defenses, including plaintiff's claims and evidentiary materials, defendant's rebuttals and count-evidence, and key disputed issues.

Table 1. Top nine causes of action in maritime court cases during 2023–2024.

Ranking	Cause of Action	Number of Cases
1	Dispute over contract of freight forwarding by sea or by waters leading to the sea	9002
2	Disputes over contract of seaman service	6515
3	Disputes over contract of carriage of goods by sea or by waters leading to the sea	5892
4	Disputes over liability for personal injury at sea or at waters leading to the sea	2155
5	Admiralty and maritime disputes	1935
6	Disputes over contract of supply of ship stores and spares	1617
7	Disputes over contract of sale and purchase of ship	1118
8	Disputes over contract of insurance by sea or by waters leading to the sea	1107
9	Charter party disputes	1073

Judicial decision comprises three components. Fact-finding reconstructs factual chains through evidentiary examination, including timelines, contractual obligations, and breach behaviors. Legal application cites applicable maritime laws, international conventions, and judicial interpretations to establish legal ground. The operative part of judgment specifies liability allocation, compensation calculations, and enforcement requirements.

Judicial personnel lists presiding judges, collegiate panel members, court clerks, and judgment date.

Appendices attach full texts of invoked laws and regulations, and evidentiary exhibits.

It is found that the litigation proceedings and judicial decision, as the core content of judgment documents, serve as the primary sources for summary generation; meanwhile, the legal application in the judicial decision and the legal provisions in the appendices provide a foundational corpus for extracting and incorporating legal terminology related to statutory clauses into the maritime judgment lexicon.

3.2. Construction of the Maritime Judgment Lexicon

A domain lexicon is a structured collection of domain-specific terminology and core concepts, providing a standardized lexical foundation for communication, research, and practical activities among professionals. In the task of automatic summarization for maritime judgment documents, constructing such a lexicon is critical for enhancing model performance. Firstly, the inherent lack of explicit word delimiters in Chinese text amplifies the difficulty of identifying term boundaries. Secondly, the jurisprudential specificity and maritime particularity of these documents demand a higher level of accuracy in generating legally precise terminology. Thus, constructing a maritime judgment lexicon significantly improves the model's ability to recognize domain-specific terms, ensuring that generated summaries accurately reflect the core legal logic of the original documents.

This paper utilizes 10,153 maritime judgment documents publicly available on China Judgments Online (https://wenshu.court.gov.cn/, accessed on 13 March 2023) as the data source, covering cases from 11 maritime courts, including those in Dalian, Tianjin, and Shanghai. The textual length distribution of the corpus is shown in Figure 2, where the horizontal axis represents the number of characters in the judgment documents, and the vertical axis denotes the number of documents with the corresponding character count. It shows that the corpus exhibits a pronounced long-text characteristic: the average document length is 3006 characters, with 18.77% (1906 documents) exceeding 5000 characters.

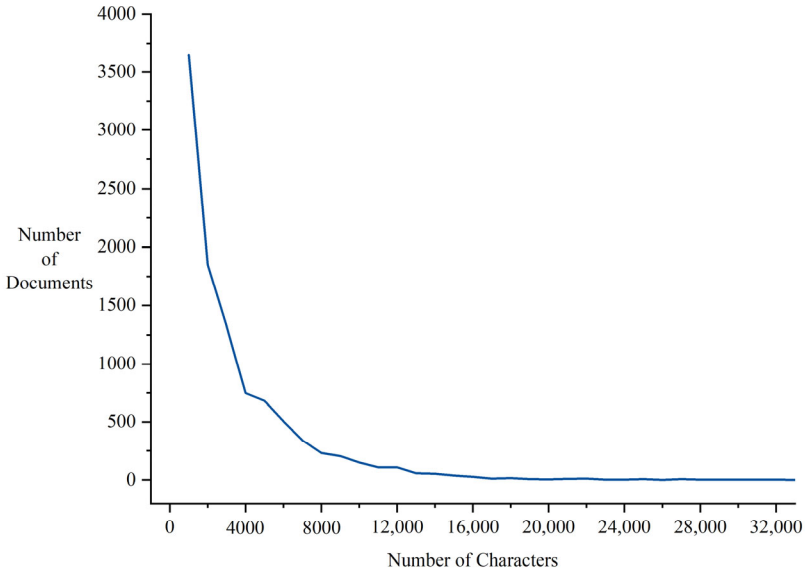


Figure 2. Length distribution of maritime judgment documents.

To construct the maritime judgment lexicon, this paper first employs regular expressions to develop three predefined domain-specific rule patterns for extracting legal terms from the corpus: law and regulation identification rules, case type recognition rules, and international treaty detection rules. Second, non-domain-specific terms are subsequently filtered out by combining a stopword list with manual rules (e.g., frequency thresholds). Finally, candidate legal and regulatory terms are validated against the National Database of Law and Regulation of China (https://flk.npc.gov.cn/, accessed on 22 February 2024) to ensure the lexicon's accuracy and authority. Through this process, an initial maritime judgment lexicon with 636 core terms is developed. To enhance domain-specific coverage, specialized vocabulary related to judicial trial is selectively incorporated. The final maritime judgment lexicon consists of 3005 standardized entries.

4. HybridSumm: A Hybrid Summarization Model for Maritime Judgment Documents

4.1. RoBERTa-DCNN-GR: An Extractive Summarization Model

To address the challenge of long maritime judgment documents, this paper models the summarization of these documents as a binary classification task distinguishing key sentences from non-key sentences. By integrating a domain-specific maritime judgment lexicon, the RoBERTa-wwm-ext Chinese pre-trained language model, dilated convolutional neural network (DCNN), gated linear unit (GLU), and residual mechanisms, we propose an extractive summarization model that synergistically combines context-aware feature extraction and long-range dependency modeling. The model dynamically filters key sentences, preserving core semantics while reducing redundancy.

4.1.1. RoBERTa-wwm-ext Pre-Trained Model

RoBERTa [61] is a robustly optimized BERT pre-training approach developed by researchers from Facebook AI and the University of Washington in 2019. Compared with BERT, RoBERTa introduces dynamic masking to dynamically alter masked tokens during pre-training, which increases the randomness and diversity of the input data, enabling it to learn deeper linguistic representations. Furthermore, RoBERTa employs substantially larger training corpora and extends pre-training durations, allowing the model to learn intricate linguistic patterns and subtleties while significantly improving its ability in context-aware semantic granularity recognition.

RoBERTa-wwm-ext [62], developed by the joint laboratory of HIT and iFLYTEK Research (HFL), is a Chinese pre-trained language model based on RoBERTa. The "wwm" in its name stands for whole word masking—a strategy that masks the entire word instead of individual characters during pre-training, thereby improving the model's understanding of the whole word. The "ext" suffix indicates an expanded corpus. Compared to RoBERTa, RoBERTa-wwm-ext extends its pre-training dataset beyond Chinese Wikipedia to encyclopedia, news, and Q&A web, totaling 5.4 billion tokens. This enriched data coverage significantly enhances the model's semantic precision and its cross-domain generalization capabilities.

Given the superior capability of RoBERTa-wwm-ext in Chinese text processing, this paper employs the model to generate contextualized sentence embeddings from Chinese maritime judgment documents, thereby providing a foundation for downstream tasks including sentence classification and extractive summarization for judicial texts.

4.1.2. Vector Representations of Sentences

To obtain vector representations of sentences in maritime judgment documents, the raw text must be converted into the standard input format of RoBERTa-wwm-ext, which involves four processing stages (as shown in Figure 3):

(1) Text serialization. Following BERT's sentence vector generation paradigm, we prepend the [CLS] token and append the [SEP] token to each sentence. The final-layer hidden state of the [CLS] token is extracted as the sentence vector. Subsequently, the text is tokenized into subwords using RoBERTa-wwm-ext's WordPiece tokenizer, mapped to token ID sequences via its pre-trained vocabulary, and processed with padding/truncation to a fixed length.

(2) Token embeddings. Each subword token is converted into a vector representation through a pre-trained 768-dimensional embedding matrix. These vectors are then ordered according to the original token sequence to form the input representation.

(3) Segment embeddings. Segment embeddings are primarily designed to distinguish between multiple sentences in the input text. However, they are scarcely used in RoBERTa-wwm-ext due to pre-training optimizations like dynamic masking. Following the model's default configuration, all inputs employ all-zero segment embeddings.

(4) Position embeddings. Position embeddings encode sequential information of tokens. These embeddings share the same dimensionality as token embeddings (768 dimensions). We employ learnable absolute positional embeddings, where each position index is associated with a unique trainable vector. During training, these embeddings are optimized through backpropagation to better capture positional dependencies between tokens, thereby enhancing contextual semantic modeling.

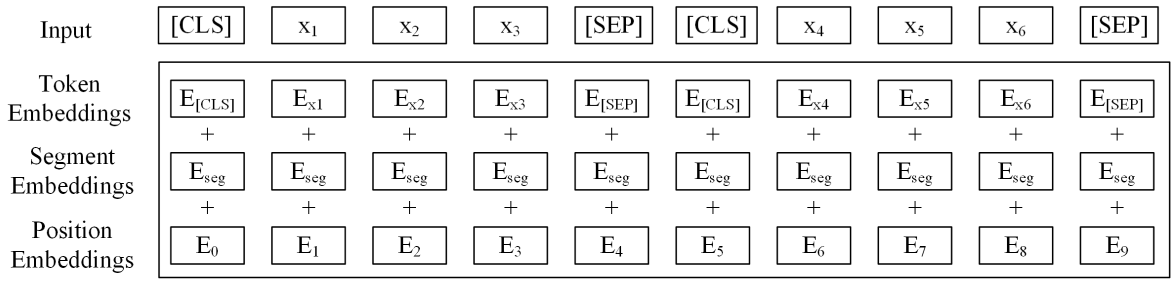


Figure 3. Input representation.

After generating token embeddings, segment embeddings, and position embeddings through preprocessing, these embeddings are fused via element-wise summation to form

the input representation for RoBERTa-wwm-ext. To derive sentence vector representations, we adopt a strategy distinct from BERT: performing global average pooling over all output token vectors (including [CLS] and [SEP]), as mathematically formalized in Equation (1):

$$\mathbf{h}_{sentence} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i \tag{1}$$

where $\mathbf{h}_i$ represents the hidden state of the i-th token, and n is the sequence length.

4.1.3. Dilated Convolutional Neural Network (DCNN)

CNN is a type of feed-forward neural network typically composed of alternately stacked convolutional layers, pooling layers, and fully connected layers. Its core strength lies in extracting local features through convolutional operations and compressing feature dimensions via pooling to enhance generalization, while optimizing network weights through backpropagation. However, CNN exhibits limitations when processing long text sequences: first, the limited receptive field of traditional convolutions struggles to model global information; second, pooling operations may discard critical details during dimensionality reduction, weakening the ability to capture long-range dependencies.

To address this issue, Yu and Koltun [63] proposed DCNN in 2016. Its core innovation involves introducing a dilation rate to expand the receptive field of convolutional kernels without increasing the number of parameters. Specifically, the dilation rate controls the spacing between convolutional kernel elements (rather than directly adjusting kernel size), enabling the exponential expansion of the receptive field (e.g., stacking multiple dilated convolutions progressively enlarges coverage). This design significantly enhances the model's capability to capture long-range contextual features while avoiding information loss from pooling operations, thereby more efficiently extracting multi-level textual information.

Figure 4 illustrates the comparison between CNN and DCNN, both with 3 × 3 kernels. In this figure, yellow nodes represent the input layer, orange represent denote the hidden layers, green nodes represent the output layer, and the direction of the arrows illustrates the flow path of information through the network. As can be seen from Figure 4:

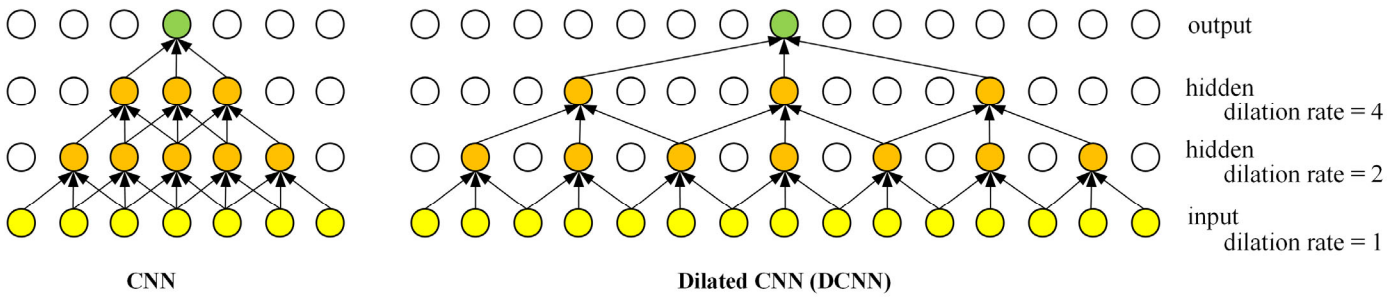


Figure 4. Receptive field expansion in CNN vs. DCNN with 3 × 3 kernels: layer-wise contextual span visualization.

(1) In the CNN architecture, each of the three convolutional layers employs a 3 × 3 kernel, where the output features are computed from a local 3 × 3 region of the previous layer. By stacking these three layers, the final receptive field expands to 7 (calculated as $1 + (3 - 1) \times 3 = 7$), enabling the model to capture contextual information from 7 consecutive characters in the input text. However, this limited receptive field restricts its ability to model long-range dependencies in long sequences.

(2) In the DCNN architecture, the first layer uses a 3 × 3 kernel with a dilation rate of 1 (equivalent to standard convolution), yielding a receptive field of 3. The second layer applies a dilation rate of 2, effectively covering a 5 × 5 region (calcu-

lated as $1 + (3-1) \times 2 = 5$). The third layer further scales the dilation rate to 4, expanding coverage to a 9×9 region (calculated as $1 + (3-1) \times 4 = 9$). By stacking these layers, DCNN achieves a final receptive field of 15 (calculated as $1 + (3-1) \times (1+2+4) = 15$), enabling the capture of global dependencies across 15 characters in the input text.

(3) DCNN expands the receptive field by introducing dilated spacing between kernel elements instead of increasing kernel size, and it maintains the same number of parameters as CNN (with nine weights per layer). This design enables the efficient capture of long-range features, allowing DCNN to significantly outperform CNN in processing long texts, such as summarizing maritime judgment documents.

4.1.4. Gated Linear Unit (GLU) and Residual Mechanism

GLU [64] employs a dual-branch parallel architecture to construct a dynamic gating mechanism through differentiated linear transformations, regulating the flow intensity of feature information. By adopting a feature filtering strategy, this mechanism suppresses redundant signals while effectively preserving gradient propagation paths. Such a design not only mitigates the gradient vanishing problem, but also significantly enhances the model's ability to model long-range dependencies within sequential data.

Based on this, we propose a GLU-integrated dilated convolution optimization module to capture long-distance features in maritime judgment documents. The module takes sentence embeddings $\mathbf{X}$ generated by RoBERTa-wwm-ext as input and processes features in parallel via dual-branch dilated convolutions, as shown in Equation (2).

$$\mathbf{Y} = Conv1d_1(\mathbf{X}) \otimes \sigma(Conv1d_2(\mathbf{X})) \tag{2}$$

where $conv1d_1$ performs a linear mapping on the input to preserve basic semantic features, while $conv1d_2$ generates a context-aware gating weight matrix through non-linear transformation and the Sigmoid function (σ). The two branches are dynamically fused via element-wise product ($\otimes$), producing the optimized feature representation $\mathbf{Y}$.

To further improve gradient propagation stability and enhance the model's capability for complex semantic modeling, we introduce a residual mechanism [65] into the gated dilated convolution module, as shown in Equation (3).

$$\mathbf{Y} = \mathbf{X} + Conv1d_1(\mathbf{X}) \otimes \sigma(Conv1d_2(\mathbf{X})) \tag{3}$$

Since $conv1d_1$ uses no activation function (only linear transformation), Equation (3) can be equivalently reformulated as Equation (4) [66].

$$\mathbf{Y} = \mathbf{X} \otimes [1 - \sigma(Conv1d_2(\mathbf{X}))] + Conv1d_1(\mathbf{X}) \otimes \sigma(Conv1d_2(\mathbf{X})) \tag{4}$$

Let $\alpha = \sigma(Conv1d_2(X))$ denote the spatially adaptive dynamic weight matrix that controls feature fusion ratios. Equation (4) is then simplified to Equation (5).

$$\mathbf{Y} = \mathbf{X} \otimes (1 - \boldsymbol{\alpha}) + Conv1d_1(\mathbf{X}) \otimes \boldsymbol{\alpha} \tag{5}$$

As shown in Equation (5), the input sequence $\mathbf{X}$ undergoes dual-branch parallel processing. In the residual branch, the original input is element-wise multiplied by the dynamic weight $1—\boldsymbol{\alpha}$ to preserve shallow-layer features. Meanwhile, in the transformation branch, $\mathbf{X}$ is first processed through a dilated convolutional layer ($conv1d_1$) to extract high-order semantic features, which are then element-wise multiplied by the dynamic gating weight $\boldsymbol{\alpha}$ to enable feature selection and enhancement. The final output $\mathbf{Y}$ is generated by direct summation of both branch results.

4.2. Nezha-UniLM-CC: An Abstractive Summarization Model

The primary goal of abstractive text summarization is to produce summaries that are semantically complete and human readable. However, this task faces three critical challenges due to textual semantic complexity and domain-specific characteristics: (1) long-range dependency modeling limitations, resulting the loss of key information; (2) OOV generation failures, limiting coverage of domain-specific terms; and (3) repetitive word generation, degrading textual coherence. To address these challenges, we propose Nezha-UniLM-CC, a domain-adaptive abstractive summarization model tailored for maritime judgment documents. This model integrates the Nezha encoder, UniLM decoder, copy mechanism from a pointer–generator network, coverage mechanism, beam search optimization, and a domain-specific lexicon.

4.2.1. Nezha Pre-Trained Model

Nezha (neural contextualized representation for Chinese language understanding), proposed by Huawei Noah's Ark Lab in 2019, is a BERT-based pre-trained language model for Chinese NLP tasks [67]. It achieved state-of-the-art performance across multiple Chinese NLP benchmarks at the time. Similar to the contemporaneous RoBERTa-wwm-ext model, Nezha adopts the whole word masking strategy to enhance semantic modeling of Chinese lexical units. Additionally, Nezha optimizes the original BERT architecture in the following three aspects:

(1) Functional relative positional encoding. To address the limitations of BERT's absolute positional encoding in handling long sequences (e.g., exceeding the 512-token length constraints during pre-training), Nezha introduces a novel sinusoidal-based relative positional encoding mechanism. By mapping relative positional relationships between tokens into a fixed-dimensional vector space using periodic sinusoidal functions—where each dimension corresponds to a specific sinusoidal frequency—the model dynamically generates positional embeddings. This approach not only overcomes sequence length limitations, but also effectively captures long-range dependencies through its periodic characteristics, significantly improving the modeling capability for ultra-long texts.

(2) Mixed-precision training acceleration. To enhance training efficiency, Nezha employs an FP16/FP32 mixed-precision training strategy [68]. This method maintains FP32-format master weights as the parameter baseline while converting them to FP16 for forward and backward computations. Gradients are then updated back to the FP32 master weights. By leveraging the computational advantages of GPU tensor cores for half-precision operations while ensuring numerical stability, this strategy achieves a 1.5–3x training speed improvement.

(3) Large-scale distributed training optimization. To mitigate the performance degradation of traditional optimizers under large-batch training, Nezha pioneers the use of the LAMB (layer-wise adaptive moments optimizer for batch training) optimizer. LAMB enables stable convergence even when scaling batch sizes to tens of thousands through layer-wise adaptive learning rate adjustments, providing critical technical support for distributed training of ultra-large language models.

In the maritime judgment document summarization task, where input texts frequently exceed 512 tokens in length, we employ Nezha as the base encoder. By leveraging its optimized relative positional encoding mechanism, Nezha effectively extracts deep semantic features from long documents. The resulting dynamic contextualized vectors are then fed into the unified language model (UniLM) decoder to generate abstractive summaries.

4.2.2. UniLM Generative Model

UniLM, proposed by Microsoft Research Asia, is a pre-trained Transformer model based on the BERT architecture [69]. By integrating three pre-training paradigms—unidirectional, bidirectional, and Seq2Seq language modeling—UniLM achieves excellent performance in both natural language understanding and generation tasks. Its core innovation lies in a dynamic attention masking mechanism that enables multi-task compatibility. Unidirectional modeling predicts masked tokens using left-to-right or right-to-left contexts to capture local dependencies within text sequences. Bidirectional modeling leverages both the left and right contexts to predict masked tokens, capturing global semantic features. Seq2Seq modeling employs an encoder–decoder framework to learn complex mappings between input and output sequences. This multi-paradigm joint pre-training strategy allows UniLM to dynamically generate diverse context-aware representations, laying the foundation for downstream NLG tasks.

For the maritime judgment document summarization task, we construct a fine-tuned Seq2Seq architecture based on UniLM. Specifically, the source text and target summary are concatenated into a joint input sequence $X = w_1 w_2 \ldots w_{\mathrm{n}}$ (w_i denotes a Chinese character), with dynamic masking applied to the summary portion to simulate the "detailed-to-concise" summarization process. The input sequence X is first converted into character-level embedding vectors via the Nezha model and then fed into an L-layer Transformer encoder. Each Transformer layer computes contextual weights using a multi-head attention mechanism, as shown in Equation (6). After L layers of iterative encoding, the model outputs deep semantic representations $\mathbf{H}^L = \left[\mathbf{h}_1^L, \mathbf{h}_2^L, \cdots, \mathbf{h}_n^L\right]$, where $\mathbf{h}_j^L$ represents the semantic encoding of the i-th input unit w_i. Finally, the decoder generates the summary autoregressively based on $\mathbf{H}^L$.

$$\mathbf{A}_l = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V} \tag{6}$$

where $\mathbf{Q} = \mathbf{H}^{l-1}\mathbf{W}_l^Q$, $\mathbf{K} = \mathbf{H}^{l-1}\mathbf{W}_l^K$, and $\mathbf{V} = \mathbf{H}^{l-1}\mathbf{W}_l^V$ represent the query, key, and value matrices, respectively. $\mathbf{H}^{l-1}$ denotes the output of the previous layer, $\mathbf{W}_l^Q$, $\mathbf{W}_l^K$, and $\mathbf{W}_l^V$ are parameter matrices for the l-th attention head, d_k is the dimension scaling factor, and M is the mask matrix controlling attention visibility between token pairs, as defined in Equation (7).

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{position } j \text{ is visible to } i \\ -\infty, & \text{position } j \text{ is masked from } i \end{cases} \tag{7}$$

4.2.3. Decoding Optimization Strategies

The decoding phase faces two critical challenges: (1) the OOV problem, where the decoder fails to generate words absent from the predefined vocabulary; and (2) repetitive word generation, where the decoder redundantly attends to specific parts of the source text, leading to duplicated content in the summaries. To address these issues, this paper integrates the copy mechanism from the pointer–generator network with a coverage mechanism, synergistically optimizing the generation process.

(1) Copy Mechanism

The copy mechanism [70] enables the decoder to dynamically choose between generating new words from the vocabulary or copying words directly from the source text. The probability distribution for generating target word w at time step t is defined in Equation (8).

$$p(w) = p_{gen} p_{vocab}(w) + \left(1 - p_{gen}\right) \sum_{i:w_i = w} a_i^t \tag{8}$$

where $p_{gen} \in [0, 1]$ controls the probability of generating w from the vocabulary, $p_{vocab}(w)$ is the probability distribution over the vocabulary (enhanced with the maritime judgment lexicon from Section 3.2), a_i^t denotes the attention weight on source word w_i at step t, and $\sum_{i:w_i=w} a_i^t$ sums the attention weights for all occurrences of w in the source text. If w does not appear in the vocabulary, $p_{vocab}(w) = 0$, forcing w to be copied from the source text. Conversely, if w does not appear in the source text, $\sum_{i:w_i=w} a_i^t = 0$, forcing w to be generated from the vocabulary.

(2) Coverage Mechanism

To suppress repetitive generation, the coverage mechanism [70] introduces a coverage vector $\mathbf{c}^t$, which accumulates historical attention distributions to guide the decoding focus, as shown in Equation (9).

$$\mathbf{c}^t = \sum_{t'=0}^{t-1} \mathbf{a}^{t'} \tag{9}$$

where $\mathbf{a}^{t'}$ represents the attention distribution over source tokens at decoding step t'. Initially ($t = 0$), $\mathbf{c}^0$ is a zero vector, as no source tokens have been covered. During decoding, $\mathbf{c}^t$ is iteratively updated to track previously attended source positions. By incorporating $\mathbf{c}^t$ into the current attention calculation, this mechanism penalizes focus on already covered positions, thereby reducing redundancy. This dynamic prioritization ensures that the decoder progressively shifts focus to uncovered content, enhancing summary coherence.

(3) Beam Search Optimization

To balance summary quality and computational efficiency, the decoding process employs a beam search algorithm [71] with beam width K. At each time step, the algorithm retains the top-K candidate sequences based on their cumulative log probabilities. The final summary is selected as the sequence with the highest global probability, ensuring coherence and relevance. This approach efficiently explores the search space while maintaining high-quality output.

4.3. HybridSumm: A Hybrid Summarization Model for Maritime Judgment Documents

Based on the maritime judgment lexicon constructed in Section 3.2, the extractive summarization model from Section 4.1, and the abstractive summarization model from Section 4.2, this paper proposes HybridSumm, a hybrid summarization model that integrates the maritime judgment lexicon, as illustrated in Figure 5.

The model adopts an "extraction–abstraction" two-stage collaborative architecture, with the maritime judgment lexicon as its core foundation, and achieves efficient summarization through the following mechanisms:

Extraction stage: to address the lengthy nature of maritime judgment documents, the RoBERTa-wwm-ext pre-trained model is employed for sentence-level vector encoding, constructing high-dimensional sentence representations via average pooling. A multi-level feature processing framework is designed to capture key semantic features in long texts: First, DCNNs are utilized to capture long-distance dependencies in text sequences, leveraging their expanded receptive fields to process extended semantic correlations. Then, GLUs are employed to perform feature filtering, combined with residual connections for information enhancement. Finally, a deep feature space is constructed by stacking multi-layer DCNNs to classify key sentences versus non-key sentences. Through this stage, the concatenation of key sentences forms an extractive summary, achieving preliminary semantic compression from verbose text to concise summary.

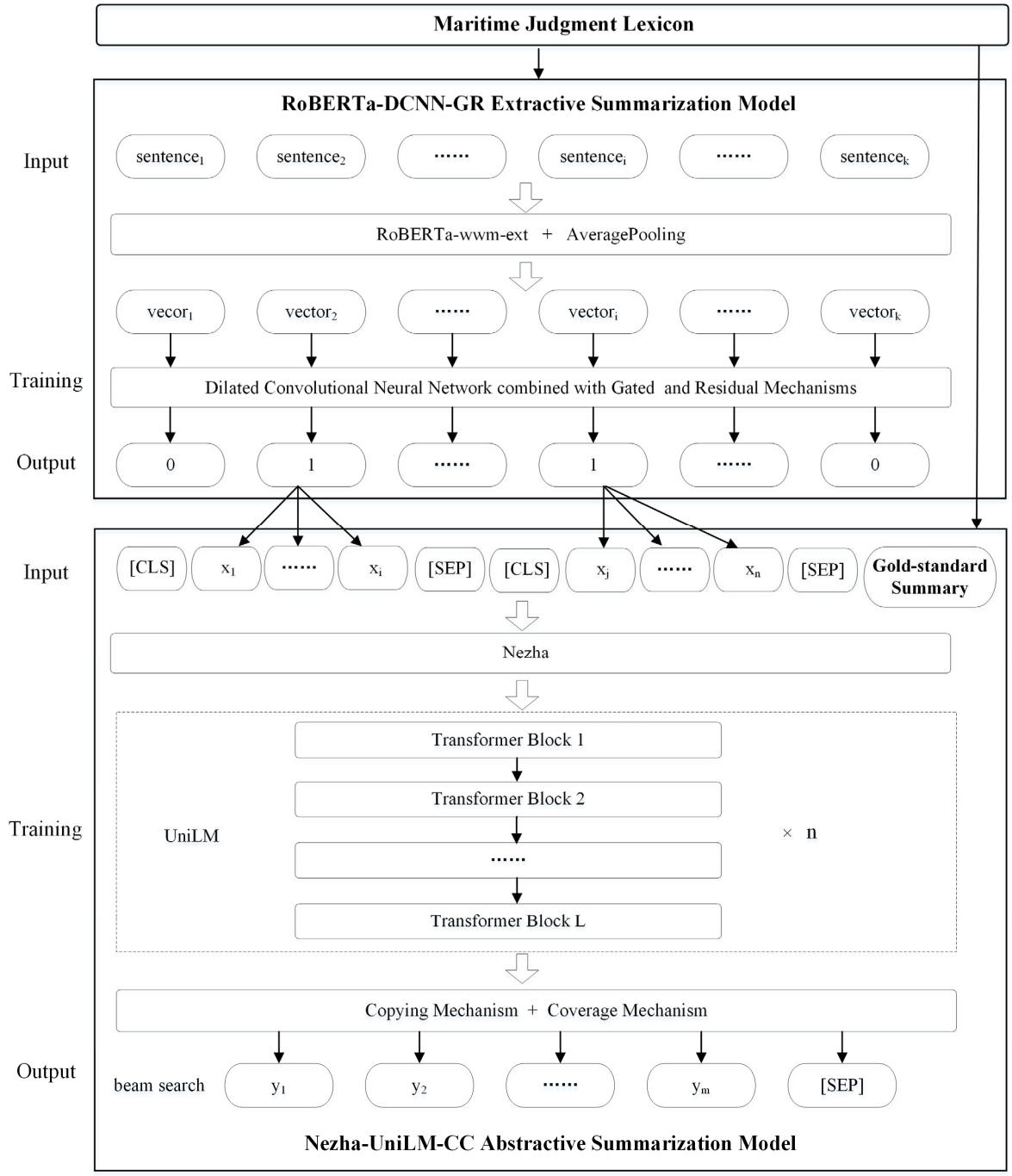


Figure 5. HybridSumm architecture: maritime judgment summarization via extractive–abstractive pipeline with maritime domain-specific lexicon integration. The arrows represent the flow and processing direction of information or data.

Abstraction stage: the extractive summary serves as input to a Nezha-based encoder for constructing contextualized embeddings, while the sequence generation task is fined-tuned through a UniLM-adapted decoder architecture. To enhance generation quality, this paper introduces a dual-mechanism collaborative optimization strategy. On the one hand, through the copy mechanism of a pointer–generator network, the vocabulary is dynamically expanded to alleviate OOV issues with domain-specific terminology, ensuring accurate generation of maritime legal terms. On the other hand, the coverage mechanism is employed to track historical attention distributions, suppressing repetitive word generation through attention weight decay. This dual guarantee mechanism achieves complementary

optimization in domain adaptability and linguistic fluency, significantly improving the professionalism and readability of generated summaries.

The HybridSumm model achieves triple synergistic advantages through the organic integration of the maritime judgment lexicon and the hybrid architecture: ensuring accurate expression of domain-specific terminology via specialized lexicon, preserving key legal elements through the extractive model, and improving linguistic fluency and structural standardization via the abstractive model. Addressing three core dimensions—semantic accuracy, content completeness, and expression standardization—HybridSumm comprehensively enhances the quality and efficiency of automatic summarization of maritime judgment documents, providing an innovative solution for intelligent processing of judicial texts.

5. Experiment and Discussion

5.1. Experimental Dataset

To validate the effectiveness of the proposed HybridSumm model based on the maritime judgment lexicon, we randomly selected 1800 maritime judgment documents from the corpus of 10,153 documents described in Section 3.2 as the experimental dataset. Gold-standard summaries for these documents were manually curated. The dataset was divided into a 5:1 ratio, with 1500 documents allocated to the training and validation sets using 15-fold cross-validation, and the remaining 300 documents reserved as the test set to ensure robust evaluation.

Table 2 presents the length statistics of original maritime judgment documents and their corresponding gold-standard summaries in the experimental dataset. The data show that the original documents have an average length of 2790 characters, with 17 documents exceeding 10,000 characters. This significant length increase is primarily due to embedded raw case evidence materials and structured tabular data. After preprocessing to remove redundant attachments and tables, the effective character count of the core text was reduced to the average range, closely aligning with the length distribution of most documents.

Table 2. Character count statistics in maritime judgment documents and gold-standard summaries in the experimental dataset.

	Maximum	Minimum	Average	Standard Deviation	Median
Judgment documents	50,104	252	2784	2782	2605
Gold-standard summaries	988	39	273	111	262

5.2. Measurement for Evaluation

5.2.1. Single-Document Summary Evaluation Metrics

To assess the quality of single-document summaries, we employ two metrics: F_1-N (N = 1, 2) based on n-gram matching and F_1-L based on the longest common subsequence (LCS).

The F_1-N metric, derived from ROUGE-N framework, quantifies n-gram overlap between the generated summary and the reference summary through Equation (10). This metric comprises two components: PRECISION-N and RECALL-N. PRECISION-N, as defined in Equation (11), measures the accuracy of generated content by calculating the proportion of n-grams in the generated summary that match the reference summary. RECALL-N, specified in Equation (12), evaluates the completeness of key information reproduction by quantifying the coverage of reference n-grams in the generated summary.

$$F_1 - N = \frac{2 \times PRECISION - N \times RECALL - N}{PRECISION - N + RECALL - N} \; N \; = \; 1, \; 2 \tag{10}$$

$$PRECISION - N = \frac{\sum_{s \in \{GeneratedSummary\}} \sum_{gram_n \in s} Count_{match}(gram_n)}{\sum_{s \in \{GeneratedSummary\}} \sum_{gram_n \in s} Count_s(gram_n)} \; N = 1, 2 \tag{11}$$

$$RECALL - N = \frac{\sum_{s' \in \{Ref Summary\}} \sum_{gram_n \in s'} Count_{match}(gram_n)}{\sum_{s' \in \{Ref Summary\}} \sum_{gram_n \in s'} Count_{s'}(gram_n)} \; N = 1, 2 \tag{12}$$

The F_1-L metric, based on ROUGE-L, assesses semantic coherence using LCS through Equation (13). It integrates two sub-metrics: PRECISION-L (Equation (14)) and RECALL-L (Equation (15)). PRECISION-L calculates the merged LCS length relative to the generated summary's word count (n), while RECALL-L computes this ratio against the reference summary's word count (m). Here, $LCS \cup (s_i, GeneratedSummary)$ represents the merged LCS length between reference sentence s_i and all generated sentences, u denotes the total number of reference sentences, and m and n indicate the word counts of the reference and generated summaries, respectively.

$$F_1 - L = \frac{2 \times PRECISION - L \times RECALL - L}{PRECISION - L + RECALL - L} \tag{13}$$

$$PRECISION - L = \frac{\sum_{i=1}^{u} LCS(s_i, GeneratedSummary)}{n} \tag{14}$$

$$RECALL - L = \frac{\sum_{i=1}^{u} LCS(s_i, GeneratedSummary)}{m} \tag{15}$$

Both F_1-N and F_1-L values positively correlate with summary quality, with higher values indicating superior performance.

5.2.2. Multi-Document Summary Evaluation Metrics

For a dataset containing r maritime judgment documents, the model's comprehensive performance is evaluated in two stages. First, compute the F_1-N (N = 1, 2) and F_1-L values for each generated summary using Equations (10) and (13). Then, derive the global metrics AF_1-N (N = 1, 2) and AF_1-L by calculating the arithmetic mean of these values across all r summaries using Equations (16) and (17). This approach ensures both fine-grained quality assessment of individual summaries and holistic evaluation of the model's stability across the dataset.

$$AF_1 - N = \frac{1}{r} \sum_{i=1}^{r} F_1 - N_i \; N = 1, 2 \tag{16}$$

$$AF_1 - L = \frac{1}{r} \sum_{i=1}^{r} F_1 - L_i \tag{17}$$

5.3. Results and Analysis

5.3.1. Validation of Maritime Judgment Lexicon

To validate the effectiveness of the maritime judgment lexicon constructed in Section 3.2 in the automatic summarization of maritime judgment documents, this paper conducts a comparison of the quality of summaries generated by the summarization models under two scenarios: with and without the provision of the maritime judgment lexicon, as shown in Table 3. RoBERT-DCNN-GR, Nezha-UniLM-CC, and HybridSumm in Table 3 refer to the extractive summarization model (see Section 4.1), the abstractive summarization model (see Section 4.2), and the hybrid (see Section 4.3) summarization model based on the maritime judgment lexicon, respectively. Moreover, "—GMJT" denotes the removal of the maritime judgment lexicon from the aforementioned three models.

Table 3. Effectiveness comparison of maritime judgment lexicon using ROUGE AF_1-N and AF_1-L.

Model	AF_1-1%	AF_1-2%	AF_1-L%
RoBERT-DCNN-GR	53.88	39.88	52.14
—GMJT	52.08	38.38	50.53
Nezha-UniLM-CC	59.47	44.57	57.01
—GMJT	56.40	40.32	53.79
HybridSumm	69.95	59.64	68.66
—GMJT	68.21	56.52	66.59

As shown in Table 3, the absence of the maritime judgment lexicon leads to performance degradation across all three models. Specifically, the AF_1-1, AF_1-2 and AF_1-L metrics of the RoBERT-DCNN-GR model decrease by 1.80%, 1.50%, and 1.61%, respectively; the Nezha-UniLM-CC model exhibits reductions of 3.07%, 4.25%, and 3.22% for the same metrics; while the HybridSumm model shows declines of 1.74%, 3.12%, and 2.07%, respectively. From these results, it can be concluded that:

(1) RoBERTa-DCNN-GR exhibits the lowest dependence on the maritime judgment lexicon, primarily due to its importance-driven extraction strategy. By directly selecting key sentences from the source text, this model prioritizes high-importance content, even if technical terms are incompletely identified. This approach inherently reduces reliance on precise term segmentation, thereby mitigating the impact of segmentation errors. Consequently, the exclusion of the lexicon results in only limited performance degradation.

(2) Nezha-UniLM-CC shows the strongest dependence on the lexicon, directly related to its "pure generative" architecture design. As an abstractive summarization model, it requires explicit access to domain-specific terminology through a lexicon; otherwise, it is prone to producing ambiguous terms or incorrect citations of laws and regulations. The integration of the lexicon enables the model's pointer–generator network to copy standardized legal terms during decoding, ensuring precise expression of legal concepts. Furthermore, the predefined keywords of laws and regulations in the lexicon standardize the citation logic of legal bases, enabling the model to accurately cite valid legal sources. Consequently, the maritime judgment lexicon has the greatest impact on Nezha-UniLM-CC's performance.

(3) HybridSumm demonstrates intermediate dependence on the lexicon, primarily due to its "extraction–abstraction" hybrid architecture. The extraction stage preserves key sentences containing domain terms, while the abstraction stage reorganizes content for fluency. Without the lexicon, the abstraction stage still encounters term precision challenges; however, the retained sentences from the extraction stage partially mitigate this issues, resulting in performance declines between purely extractive and purely abstractive models.

(4) The maritime judgment lexicon has enhanced the performance of all three models—RoBERT-DCNN-GR, Nezha-UniLM-CC, and HybridSumm—to varying degrees, confirming the non-negligible role of domain-specific terminology in maritime legal summarization. By explicitly encoding professional terms and their semantic relationships, the lexicon provides interpretable domain-specific prior constraints for automatic summarization models, thereby improving the accuracy, coherence, and judicial logical validity of generated summaries.

5.3.2. Validation of the HybridSumm Model

To validate the effectiveness of the proposed HybridSumm model for maritime judgment document summarization, we conduct a comprehensive comparison among seven summarization algorithms, six of which are commonly used baseline algorithms, and one is the HybridSumm model.

(1) M1: the LexRank algorithm [14].
(2) M2: the Word2Vec-based LexRank algorithm. In contrast to M1, which employs TF-IDF-based VSM for sentence vectorization, M2 constructs contextualized sentence representations through Word2Vec embeddings. Consider a sentence s with q feature terms. Let $\mathbf{e}_i \in \mathbb{R}^d$ $(1 \leq i \leq q)$ represent the d-dimensional embedding of the i-th term. The corresponding sentence vector $\mathbf{v}_s$ is computed as the arithmetic mean of all q term embeddings, as shown in Equation (18). The embeddings are generated by a 300-dimensional Word2Vec model trained on a 9.86 GB Chinese Wikipedia corpus using the Gensim library.

$$\mathbf{v}_s = \frac{\sum_{i=1}^{q} \mathbf{e}_i}{q} \tag{18}$$

(3) M3: the MacBERT-attention algorithm. MacBERT [72], developed by HFL in November 2020, is a BERT-based Chinese pre-trained language model. In this paper, the base version of the MacBERT model [73] with a multi-head attention mechanism is employed to summarize maritime judgment documents. Specifically, based on the attention matrices produced by each head in the last layer of MacBERT's multi-head attention mechanism, the average attention scores of all heads for each token within the input sequence are computed, with the score of the [CLS] token taken as the sentence score. Subsequently, the top-scoring sentences are extracted to form a summary.
(4) M4: the RoBERTa-DCNN-GR algorithm, which is based on the maritime judgment lexicon, as proposed in Section 4.1.
(5) M5: the Chinese BART model, a Chinese-specific linguistically enhanced adaptation of the BART model [74] originally developed by Facebook AI in 2019. By combining BERT's bidirectional contextual encoding with GPT-style autoregressive decoding, it achieves exceptional performance in both text generation and comprehension tasks. In this paper, we employ the base version of the Chinese BART [75], fine-tuned on a dataset of 1500 documents, to generate summaries of maritime judgment documents.
(6) M6: the Nezha-UniLM-CC algorithm, which is based on the maritime judgment lexicon, as proposed in Section 4.2.
(7) M7: the HybridSumm algorithm, which is based on the maritime judgment lexicon, as proposed in Section 4.3. Unlike M6 that takes raw maritime judgment documents as input, M7 uses the pre-summary generated by M4 as the input for M6, and then utilizes M6 to produce the final summary.

The above seven algorithms are categorized into three groups: M1–M4 as extractive summarization algorithms; M5–M6 as abstractive summarization algorithms; and M7 as a hybrid summarization algorithm. The parameter settings for baseline algorithms M3–M6 are shown in Table 4.

To objectively quantify their performance, the ROUGE-based automatic metrics AF1-N (Equation (10)) and AF1-L (Equation (13)) are employed to evaluate the quality of summaries generated from the test set of maritime judgment documents. The comparison results are presented in Table 5.

Table 4. Parameter settings for baseline algorithms M3–M6.

Parameter	M3	M4	M5	M6
batch_size	32	64	3	4
epochs	30	20	40	50
max_length	512	512	512	512
hidden_size	768	384	768	768
learning_rate	5×10^{-5}	0.001	5×10^{-5}	2×10^{-5}
warmup_steps	200	—	1000	—
dropout_rate	0.1	0.1	0.1	0.1

Table 5. Performance comparison of M1–M7 using ROUGE AF_1-N and AF_1-L.

Type	Method	AF_1-1%	AF_1-2%	AF_1-L%
Extractive	M1	34.31	16.54	28.81
	M2	38.31	21.29	33.59
	M3	36.75	21.20	33.46
	M4	53.88	39.88	52.14
Abstractive	M5	57.70	41.98	54.32
	M6	59.47	44.57	57.01
Hybrid	M7	69.95	59.64	68.66

The following points can be determined from Table 5.

(1) Among the three groups of summarization methods, the hybrid method exhibits the best performance, followed by the abstractive methods, while the extractive methods rank last. This is mainly due to the differences in their capabilities for deep semantic understanding and information restructuring mechanisms. The hybrid summarization method integrates extraction and generation techniques through a two-stage processing framework (first extracting key sentences, then performing semantic reconstruction) and dynamic balancing mechanisms (e.g., attention mechanisms). This ensures the accuracy of facts while improving linguistic fluency, thus producing the highest quality summaries. The abstractive summarization methods, though capable of transcending text structures to generate new sentences through semantic reorganization, are constrained by semantic drift risks (e.g., long-range dependency bias of the Transformer decoders) and high data dependency, resulting in suboptimal performance compared to the hybrid method. The extractive summarization methods, confined to text structures, face information density bottlenecks and logical discontinuity issues, ultimately exhibiting the weakest adaptability and lowest metrics, and thus ranked last among the three groups of methods.

(2) Among the four extractive summarization methods, M4 shows significant superiority over M1, M2, and M3. This result strongly proves the effectiveness of our proposed extractive method M4. Specifically, M4 employs the RoBERTa-www-ext model to obtain sentence embeddings, utilizes stacked DCNNs for sentence classification, and further optimizes the architecture through GLUs and residual mechanisms. This innovative combination enables the model to comprehensively capture global contextual semantics in long-text maritime judgment documents, thus generating high-quality summaries.

(3) Between the two abstractive summarization methods, M6 shows moderately superior performance compared to M5, with AF_1-1, AF_1-2, and AF_1-L metrics improving by 1.77%, 2.59%, and 2.69%, respectively. This result validates the effectiveness of our proposed abstractive method M6 for Chinese maritime judgment document summarization. Specifically, by integrating the Nezha model, the UniLM model, the

copy mechanism from a pointer–generator network, the coverage mechanism, and beam search, M6 can effectively capture domain-specific semantic patterns while maintaining structural coherence, thus exhibiting good performance.

(4) The hybrid summarization method M7, which integrates extractive method M4 and abstractive method M6, outperforms both individual methods. Quantitative analysis shows that M7 achieves improvements in AF_1-1, AF_1-2, and AF_1-L by 16.07%, 19.76%, and 16.52%, respectively, compared to M4, and by 10.48%, 15.07%, and 11.65%, respectively, compared to M6. This demonstrates that in the task of maritime judgment document summarization, using the extractive summaries generated by M4 as the input to M6 can effectively overcome the limitations of individual methods, and significantly enhance summary quality.

(5) Among the seven summarization methods (M1–M7), the hybrid method M7 proposed in this paper demonstrates superior performance in generating summaries for Chinese maritime judgment documents. This result strongly validates the effectiveness of M7 and highlights its exceptional capability in the task of maritime judgment document summarization.

In addition, the following points can also be determined from Table 5.

(1) Compared to M1, M2 achieves improvements of 4.00%, 4.75%, and 4.78% in AF_1-1, AF_1-2, and AF_1-L metrics, respectively. This demonstrates that the Word2Vec model, which accounts for semantic relationships and syntactic dependencies between terms, provides superior text representation compared to traditional VSM. By more accurately capturing inter-sentence semantic similarities, the enhanced LexRank algorithm empowered by Word2Vec significantly improves summary quality in maritime judgment document summarization tasks.

(2) The summaries generated by M3 achieve improvements of 2.44%, 4.66%, and 4.65% in AF_1-1, AF_1-2, and AF_1-L metrics compared to M1, while exhibiting decreases of 1.56%, 0.09%, and 0.13% relative to M2. The reason for M3's superior performance over M1 mainly lies in MacBERT's advanced linguistic comprehension and context-aware representation capabilities. Compared to VSM, MacBERT captures richer semantic patterns—including complex inter-word semantic relationships and contextual dependencies—enabling more precise text interpretation and key information extraction during summarization. As for the reason why M3 underperforms M2, we consider that it mainly results from MacBERT's input length constraint (maximum 512 tokens). When processing long-text maritime judgment documents, this limitation leads to partial text truncation, hindering the model's ability to resolve long-range dependencies and consequently impairing its capacity to capture key document information.

6. Conclusions

To address the characteristics of maritime judgment documents such as excessive length, complex terminology, and specialized content, this paper introduces HybridSumm, an "extraction–abstraction" hybrid summarization framework integrated with a domain-specific maritime judgment lexicon. Experimental results demonstrate that both the constructed lexicon and the HybridSumm model effectively enhance the quality of generated summaries for maritime judgment documents.

However, this paper has two key limitations: First, while comparative experiments confirm the performance improvement of combining DCNN, GLU, and the pointer–generator mechanism, component-level ablation studies to quantify individual module contributions are lacking. Second, although the overall effectiveness of the maritime judgment lexicon is empirically validated, deeper analysis of core term influence on information extraction and

generation processes remains incomplete. Future work will focus on addressing these gaps through systematic ablation frameworks and term impact quantification methodologies.

Author Contributions: Conceptualization, L.Z. and Y.L.; methodology, L.Z. and Y.L.; software, Y.L. and L.Z.; validation, L.Z. and Y.L.; formal analysis, L.Z. and Y.L.; data curation, Y.L. and H.Z.; writing—original draft preparation, Y.L. and L.Z.; writing—review and editing, L.Z.; visualization, Y.L. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Dalian Social Science Research Project (Grant No. 2025dlskzd020) of 2025 from the Dalian Federation of Social Science Associations, and Liaoning Economic and Social Development Research Project (Grant No. 2025lslybwzzkt-062) of 2025 from Liaoning Federation of Social Science Associations.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data supporting the findings of this study are openly available at https://wenshu.court.gov.cn/ (accessed on 13 March 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Luhn, H.P. The automatic creation of literature abstracts. *IBM J. Res. Dev.* **1958**, *2*, 159–165. [CrossRef]
2. Gambhir, M.; Gupta, V. Recent automatic text summarization techniques: A survey. *Artif. Intell. Rev.* **2017**, *47*, 1–66. [CrossRef]
3. Liu, J.; Zou, Y. A review of automatic text summarization research in recent 70 years. *Inf. Sci.* **2017**, *35*, 154–161. (In Chinese)
4. Ishikawa, K.; Ando, S.; Okumura, A. Hybrid text summarization method based on the TF method and the LEAD method. In Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization, Tokyo, Japan, 7–9 March 2001; pp. 325–330.
5. Edmundson, H.P. New methods in automatic extracting. *J. ACM* **1969**, *16*, 264–285. [CrossRef]
6. Lloret, E.; Palomar, M. A gradual combination of features for building automatic summarisation systems. In Proceedings of the 12th International Conference on Text, Speech and Dialogue, Pilsen, Czech Republic, 13–17 September 2009; pp. 16–23.
7. PadmaLahari, E.; Kumar, D.V.N.S.; Prasad, S. Automatic text summarization with statistical and linguistic features using successive thresholds. In Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 8–10 May 2014; pp. 1519–1524.
8. Luo, F.; Wang, J.; He, D. Research on automatic text summarization combining topic feature. *Appl. Res. Comput.* **2021**, *38*, 129–133. (In Chinese)
9. Gong, Y.; Liu, X. Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, USA, 9–13 September 2001; pp. 19–25.
10. Kar, M.; Nunes, S.; Ribeiro, C. Summarization of changes in dynamic text collections using Latent Dirichlet Allocation model. *Inf. Process. Manag.* **2015**, *51*, 809–833. [CrossRef]
11. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report; Stanford Infolab: Stanford, CA, USA, 1998.
12. Zong, Q.; Xia, R.; Zhang, J. *Text Data Mining*; Springer Nature Singapore Pte Ltd.: Singapore, 2021; pp. 287–299.
13. Mihalcea, R.; Tarau, P. TextRank: Bringing order into texts. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
14. Erkan, G.; Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [CrossRef]
15. Huang, B.; Liu, C. Chinese automatic text summarization based on weighted TextRank. *Appl. Res. Comput.* **2020**, *37*, 407–410. (In Chinese)
16. Liu, H.; Zhang, Y. Chinese Single Document Summarization Based on LexRank. *J. Ordnance Equip. Eng.* **2017**, *38*, 85–89. (In Chinese)
17. AL-Khassawneh, Y.A.; Salim, N.; Jarrah, M. Improving triangle-graph based text summarization using hybrid similarity function. *Indian J. Sci. Technol.* **2017**, *10*, 1–15. [CrossRef]

18. Carbonell, J.; Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual ACM SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia, 24–28 August 1998; pp. 335–336.
19. Pozzi, A.; Barbierato, E.; Toti, D. Cryptoblend: An AI-powered tool for aggregation and summarization of cryptocurrency news. *Informatics* **2023**, *10*, 5. [CrossRef]
20. Yin, W.; Pei, Y. Optimizing sentence modeling and selection for document summarization. Proceedings of 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1383–1389.
21. Nenkova, A.; McKeown, K. A Survey of Text Summarization Techniques. In *Mining Text Data*; Aggarwal, C.C., Zhai, C., Eds.; Springer Science+Business Media: New York, NY, USA, 2012; pp. 43–76.
22. Kupiec, J.; Pedersen, J.; Chen, F. A trainable document summarizer. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 9–13 July 1995; pp. 68–73.
23. Zhang, J.; Chan, H.Y.; Fung, P.; Cao, L. A comparative study on speech summarization of broadcast news and lecture speech. Proceedings of 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 27–31 August 2007; pp. 2781–2784.
24. Conroy, J.; O'leary, D. Text summarization via hidden Markov models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, USA, 9–13 September 2001; pp. 406–407.
25. Osborne, M. Using maximum entropy for sentence extraction. In Proceedings of the ACL-02 Workshop on Automatic Summarization, Phildadelphia, PA, USA, 11–12 July 2002; pp. 1–8.
26. Chali, Y.; Hasan, S.A. Query-focused multi-document summarization: Automatic data annotations and supervised learning approaches. *Nat. Lang. Eng.* **2012**, *18*, 109–145. [CrossRef]
27. Louis, A.; Joshi, A.; Nenkova, A. Discourse indicators for content selection in summarization. In Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Tokyo, Japan, 24–25 September 2010; pp. 147–156.
28. Nallapati, R.; Zhai, F.; Zhou, B. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of document. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3075–3081.
29. Rachabathuni, P.K. A survey on abstractive summarization techniques. In Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23–24 November 2017; pp. 762–765.
30. Belkebir, R.; Guessoum, A. Concept generalization and fusion for abstractive sentence generation. *Expert Syst. Appl.* **2016**, *53*, 43–56. [CrossRef]
31. Nayeem, M.T.; Chali, Y. Paraphrastic fusion for abstractive multi-sentence compression generation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 2223–2226.
32. Embar, V.R.; Deshpande, S.R.; Vaishnavi, A.K.; Jain, V.; Kallimani, J.S. sArAmsha—A Kannada abstractive summarizer. In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 22–25 August 2013; pp. 540–544.
33. Kurisinkel, L.J.; Zhang, Y.; Varma, V. Abstractive multi-document summarization by partial tree extraction, recombination and linearization. In Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Taipei, Taiwan, 27 November–1 December 2017; pp. 812–821.
34. Khan, A.; Salim, N.; Farman, H.; Khan, M.; Jan, B.; Ahmad, A.; Paul, A. Abstractive text summarization based on improved semantic graph approach. *Int. J. Parallel Prog.* **2018**, *46*, 992–1016. [CrossRef]
35. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
36. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 2, pp. 3104–3112.
37. Rush, A.M.; Chopra, S.; Weston, J. A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 379–389.
38. Chopra, S.; Auli, M.; Rush, A.M. Abstractive sentence summarization with attentive recurrent neural networks. In Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 93–98.
39. Nallapati, R.; Zhou, B.; dos Santos, C.; Gulcehre, C.; Xiang, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 280–290.

40. Tan, J.; Wan, X.; Xiao, J. Abstractive document summarization with a graph-based attentional neural model. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1171–1181.
41. Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H. Distraction-based neural networks for document summarization. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16), New York, NY, USA, 9–15 July 2016; pp. 2754–2760.
42. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020; pp. 11328–11339.
43. Pozzi, A.; Incremona, A.; Tessera, D.; Toti, D. Mitigating exposure bias in large language model distillation: An imitation learning approach. *Neural. Comput. Appl.* **2025**. [CrossRef]
44. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 16 June 2024).
45. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
46. Liu, Y. Fine-tune BERT for extractive summarization. *arXiv* **2019**, arXiv:1903.10318. [CrossRef]
47. Liu, Y.; Lapata, M. Text summarization with pretrained encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3730–3740.
48. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. 2019. Available online: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (accessed on 16 June 2024).
49. Kieuvongngam, V.; Tan, B.; Niu, Y. Automatic text summarization of COVID-19 medical research articles using bert and gpt-2. *arXiv* **2020**, arXiv:2006.01997. [CrossRef]
50. An, Z.; Lai, Y.; Feng, Y. Natural language understanding for legal text: A review. *J. Chin. Inf. Process.* **2022**, *36*, 1–11. (In Chinese)
51. Grover, C.; Hachey, B.; Hughson, I. Automatic summarisation of legal documents. In Proceedings of the 9th International Conference on Artificial Intelligence and Law, Scotland, UK, 24–28 June 2003; pp. 243–251.
52. Farzindar, A.; Lapalme, G. LetSum, an automatic legal text summarizing system. In Proceedings of the 17th Annual Conference on Legal Knowledge and Information Systems, Berlin, Germany, 8–10 December 2004; pp. 11–18.
53. Galgani, F.; Compton, P.; Hoffmann, A. Combining different summarization techniques for legal text. In Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, Avignon, France, 23 April 2012; pp. 115–123.
54. Duan, X.; Zhang, Y.; Yuan, L. Legal summarization for multi-role debate dialogue via controversy focus mining and multi-task learning. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1361–1370.
55. Bhattacharya, P.; Poddar, S.; Rudra, K.; Ghosh, K.; Ghosh, S. Incorporating domain knowledge for extractive summarization of legal case documents. In Proceedings of the 18th International Conference on Artificial Intelligence and Law, São Paulo, Brazil, 21–25 June 2021; pp. 22–31.
56. Su, J. "SPACES: Extraction-Abstraction"-Based Long-Text Summarization (LAIC Summary). 2021. Available online: https://spaces.ac.cn/archives/8046 (accessed on 27 July 2024). (In Chinese).
57. Zhou, W.; Wang, Z.; Wei, B. Abstractive automatic summarizing model for legal judgment documents. *Comput. Sci.* **2021**, *48*, 331–336. (In Chinese)
58. Anand, D.; Wagh, R. Effective deep learning approaches for summarization of legal texts. *J. King Saud Univ.-Com.* **2022**, *34*, 2141–2150. [CrossRef]
59. Wei, X.; Qin, Y.; Tang, X.; Huang, R.; Chen, Y. Method of abstracting judgement document combined with law. *Comput. Eng. Des.* **2023**, *44*, 2844–2850. (In Chinese)
60. Huang, Y.; Sun, L.; Han, C.; Guo, J. A High-Precision Two-Stage Legal Judgment Summarization. *Mathematics* **2023**, *11*, 1320. [CrossRef]
61. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692. [CrossRef]
62. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-training with whole word masking for Chinese bert. *IEEE/ACM T. Audio Spe.* **2021**, *29*, 3504–3514. [CrossRef]
63. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
64. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning—Volume 70, Sydney, Australia, 6–11 August 2017; pp. 933–941.

65. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
66. Su, J. DGCNN: A Reading Comprehension-Based Question Answering Model Using CNN. 2018. Available online: https://spaces.ac.cn/archives/5409 (accessed on 29 July 2024). (In Chinese)
67. Wei, J.; Ren, X.; Li, X.; Huang, W.; Liao, Y.; Wang, Y.; Lin, J.; Jiang, X.; Chen, X.; Liu, Q. NEZHA: Neural contextualized representation for Chinese language understanding. *arXiv* **2019**, arXiv:1909.00204. [CrossRef]
68. Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; et al. Mixed precision training. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
69. Li, D.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.W. Unified language model pre-training for natural language understanding and generation. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 13063–13075.
70. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1073–1083.
71. Wiseman, S.; Rush, A.M. Sequence-to-sequence learning as beam-search optimization. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1296–1306.
72. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting pre-trained models for Chinese natural language processing. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 657–668.
73. Available online: https://github.com/ymcui/MacBERT (accessed on 27 June 2024).
74. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
75. Available online: https://huggingface.co/uer/bart-base-chinese-cluecorpussmall (accessed on 21 April 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.