TermDiffuSum: A Term-guided Diffusion Model for Extractive Summarization of Legal Documents

Xiangyun Dong, Wei Li*, Yuquan Le, Zhangyue Jiang, Junxi Zhong, Zhong Wang
College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
{dongxy, rj_wli, leyuquan, s2210w1095, zjx1038, zhongwang}@hnu.edu.cn

Abstract

Extractive summarization for legal documents aims to automatically extract key sentences from legal texts to form concise summaries. Recent studies have explored diffusion models for extractive summarization task, showcasing their remarkable capabilities. Despite these advancements, these models often fall short in effectively capturing and leveraging the specialized legal terminology crucial for accurate legal summarization. To address the limitation, this paper presents a novel term-guided diffusion model for extractive summarization of legal documents, named TermDiffuSum. It incorporates legal terminology into the diffusion model via a well-designed multifactor fusion noise weighting schedule, which allocates higher attention weight to sentences containing a higher concentration of legal terms during the diffusion process. Additionally, TermDiffuSum utilizes a re-ranking loss function to refine the model's selection of more relevant summaries by leveraging the relationship between the candidate summaries generated by the diffusion process and the reference summaries. Experimental results on a self-constructed legal summarization dataset reveal that TermDiffuSum outperforms existing diffusion-based summarization models, achieving improvements of 3.10 in ROUGE-1, 2.84 in ROUGE-2, and 2.89 in ROUGE-L. To further validate the generalizability of TermDiffuSum, we conduct experiments on three public datasets from news and social media domains, with results affirming the scalability of our approach.

1 Introduction

Legal artificial intelligence (LegalAI) (Zhong et al., 2020a; Le et al., 2024) aims to explore technologies such as natural language processing (Zhang et al., 2020) to assist with legal tasks in real-world scenarios. Legal tasks often involve the processing of legal documents. Most legal documents record numerous case details, making it difficult for legal practitioners to quickly obtain the important information they need (Jain et al., 2021b; Kanapala et al., 2019). Extractive summarization for legal documents aims to automatically extract essential information from these documents and has received extensive attention from both scholars and legal practitioners in recent years.

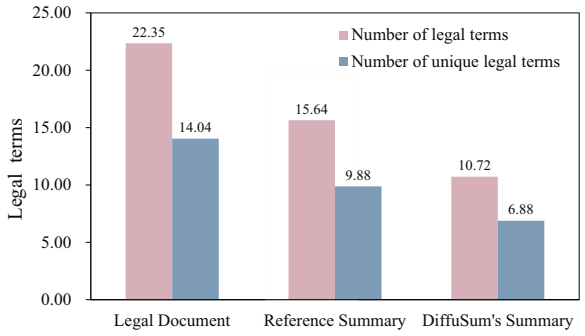


Figure 1: Statistical data on legal terms in the LegalAFSum Dataset and DiffuSum Summarization Model. "Number of legal terms" refers to the average number of terms contained in each text. "Number of unique legal terms" denotes the average number of distinct legal terms in each text.

Researchers usually view summarization of legal documents as an automatic text summarization problem (Shukla et al., 2022). Automatic text summarization approaches can be divided into abstractive summarization (Huang et al., 2020; Feijo and Moreira, 2019) and extractive summarization (Galgani et al., 2015; Anand and Wagh, 2022). Abstractive summarization methods may produce unexpected vocabulary, resulting in potentially risky summaries. Even recently developed Large Language Models (LLMs) like ChatGPT[1] exhibit hallucination problems in abstractive summarization (Deroy et al., 2023; Zhang et al., 2023b). Due to the high accuracy requirements for legal document summaries, most research methods (Polsley

*Corresponding author.

[1] https://chatgpt.com/

Proceedings of the 31st International Conference on Computational Linguistics, pages 3222–3235
January 19–24, 2025.

et al., 2016; Liu and Chen, 2019) focus on extractive summarization approaches, which extract key sentences from documents to construct summaries.

Recently, diffusion models have shown considerable potential in extractive text summarization (Zhang et al., 2023a). However, directly applying diffusion models to summarization of legal documents may result in an inadequate understanding of the legal terminology. Legal documents typically include numerous legal terms that reflect the core concepts of the documents. As shown in Figure 1, it presents the statistical information of our self-constructed legal summarization dataset LegalAFSum. Legal documents contain an average of 14.04 unique legal terms, and reference summaries contain an average of 9.88 unique legal terms. About 70.37% of legal terms in legal documents appear in reference summaries, indicating that these terms constitute crucial information for legal summaries. Lacking a comprehensive understanding of these legal terms, the generated summary may fail to meet practical requirements. Figure 1 shows that the diffusion-based model includes only 6.88 unique legal terms, suggesting that there is still significant room for improvement in its comprehension of legal terminology.

Spurred by the above observations, this paper proposes a term-guided diffusion model for extractive summarization of legal documents, named TermDiffuSum, which leverages legal terminology to guide the diffusion process in extracting summaries. To enhance the model's perception of legal terminology, TermDiffuSum devises a multifactor fusion noise weighting schedule that allocates greater attention to sentences containing a higher concentration of legal terms during the diffusion process. Additionally, TermDiffuSum constructs a re-ranking loss function based on the ROUGE scores of candidate summaries generated by diffusion process. This function optimizes the model's selection of more relevant summaries by exploiting the relationship between the candidate summaries and the reference summaries. The specific contributions of this paper are as follows:

- This paper proposes a novel term-guided diffusion model for extractive summarization of legal documents. It improves the diffusion model's sensitivity to legal terminology by incorporating a tailored multifactor fusion noise weighting schedule.
- We introduce a re-ranking module that enhances the model's ability to identify more relevant summaries by leveraging the relationship between candidate and reference summaries.
- We construct a new legal summarization dataset, called LegalAFSum. Extensive experiments on LegalAFSum validate the effectiveness of our method compared to competitive summarization models.
- We conduct experiments on three public datasets in the news and social media domains. The experimental results show the generality of our approach. We will release the code and dataset to facilitate research in this area [2].

2 Related Work

2.1 Diffusion Models on Text

Diffusion models first appear in continuous domain (Ho et al., 2020a; Ramesh et al., 2022) and have recently gained attention in natural language processing (Austin et al., 2021; He et al., 2022). Researchers have successfully applied diffusion models to tasks such as conditional generation (Gong et al., 2022; Yuan et al., 2022) and entity recognition (Shen et al., 2023) by employing techniques like embedding functions (Li et al., 2022) or designing noise schedules suitable for text (Wang et al., 2023). For summarization, DiffuSum (Zhang et al., 2023a) is the first to explore diffusion models for extractive summarization. Instead of generating text word by word, DiffuSum directly generates summary representations and selects sentences based on the matching of sentence representations. Additionally, DiffuSum introduces a contrastive sentence encoding module to learn sentence representations, further enhancing the performance of DiffuSum. Compared to DiffuSum, our model differs in the following ways: (1) We apply the diffusion model for the extractive summarization of legal documents and design a sentence-level noise schedule that utilizes legal terminology. (2) We incorporate a re-ranking module that improves the model's ability to discern more relevant summaries by ranking the candidate summaries and evaluating their alignment with reference summaries.

2.2 Summarization of Legal Documents

Summarization methods for legal documents are primarily categorized into abstractive (Moro and Ragazzi, 2022) and extractive (Jain et al., 2021a;

[2] https://github.com/huaand/TermDiffuSum-

Shukla et al., 2022) approaches. Abstractive models offer flexibility but may produce content that is unfaithful to the document. In contrast, extractive models derive summaries directly from the documents, thus ensuring greater accuracy and consistency. This approach is more relevant to our work. Research in the area can be divided into unsupervised and supervised methods. Early works mainly focus on unsupervised methods. They use statistical features to generate summaries. For instance, LetSum (Farzindar and Lapalme., 2004) identifies sentence themes and selects key sentences based on thematic content and document structure. CaseSummarizer (Polsley et al., 2016) scores sentences based on word frequency and legal features. DELSumm (Bhattacharya et al., 2021) incorporates legal domain knowledge to enhance the algorithm for selecting informative sentences.

Recently, attention has shifted to supervised methods. DCESumm (Jain et al., 2024) uses BERT to predict sentence relevance scores and refines these scores with unsupervised clustering. Gist (Liu and Chen, 2019) employs three classifiers: Decision Tree, MLP, and LSTM, to evaluate sentence importance based on features such as legal and linguistic characteristics. These methods treat extractive summarization as a sequence labeling task. They select the important sentences to form the summaries but ignore the relationships among the sentences. In contrast, our approach is a summary-level framework for summarizing legal documents that employs a diffusion model to effectively model sentence relationships.

3 Model

This section provides a detailed examination of TermDiffuSum, as illustrated in Figure 2. TermDiffuSum mainly consists of a diffusion module and a re-ranking module. The diffusion module generates embeddings for the target summary, while the re-ranking module constructs candidate summaries from these embeddings and establishes a re-ranking loss by evaluating candidate summaries.

3.1 Problem Definition

Extractive summarization of legal documents aims to generate a subset of the legal documents that captures the essential content of the documents. Formally, given a legal document D, the objective is to generate a summary S:

$$S = F(D, \theta). \tag{1}$$

Specifically, D is the legal document and S is the desired summary. θ is the parameters of the summarization model F.

Since legal terms often contain important information, it is crucial for legal document summaries to accurately express the content of documents. We incorporate legal terms to assist the model in understanding legal documents:

$$S = F(D, T, \theta), \tag{2}$$

where T denotes the legal terms in document D.

3.2 Diffusion Module

3.2.1 Multifactor Fusion Noise Weighting Schedule

In legal documents, the importance of the sentence determines its probability of appearing in the summary. Naturally, the noise should vary according to the importance of the sentences. Unlike conventional noise schedules (Lin, 2004; Li et al., 2022) that add uniform noise to all tokens, we propose a multifactor fusion noise weighting schedule that assigns weights to sentences. This schedule aims to prioritize adding noise to sentences of higher importance, thereby enabling the model to focus more on critical sentences. For a sentence $s = \{w_1, w_2, \ldots, w_k\}$, the sentence weight $e(s)$ is calculated by considering the following aspects:

- **Word Information Entropy**: Higher information entropy (Bentz and Alikaniotis, 2016) generally indicates that a sentence contains richer information. Therefore, we use information entropy as a criterion for evaluating sentence importance. The entropy of sentence s is:

$$H_{\text{entropy}}(s) = -\sum_{i=1}^{k} p(w_i) \cdot \log(p(w_i)), \tag{3}$$

  where k is the number of words in sentence s, w_i denotes the i-th word, and $p(w_i)$ represents the probability of the word w_i.

- **Number of Legal Terms**: Legal documents typically contain numerous legal terms that reflect the core views of the document. Therefore, we use the number of legal terms to evaluate sentence importance. The formula is as follows:

$$H_{\text{key}}(s) = \text{Bool}(s) + \lambda_1 \cdot \text{Num}(s), \tag{4}$$

  where $\text{Bool}(\cdot)$ denotes a binary indicator function ($\text{Bool}(s) = 1$ if the sentence contains legal terms, and $\text{Bool}(s) = 0$ otherwise), $\text{Num}(s)$ is the number of legal terms in the sentence, and λ_1 is a weight parameter.

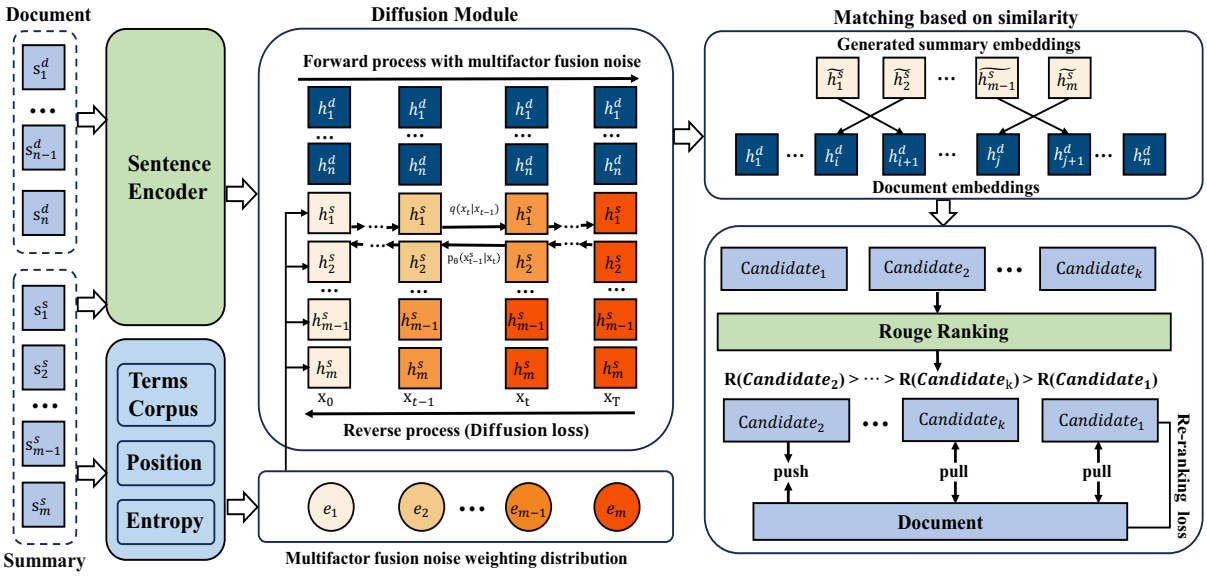


Figure 2: The overview architecture of TermDiffuSum, where e_i is the importance of sentence s_i^s, and $e_1 < e_2 < \ldots < e_{m-1} < e_m$. The multifactor fusion noise weighting schedule enables the model to prioritize adding noise to more important sentences.

- **Positional Information**: Generally, sentences in specific positions (such as the beginning or end) are considered more important than those in other positions (such as the middle). Based on this, we incorporate position information when evaluating sentence importance. The positional information is computed by:

$$H_{\text{weight}}(s) = \exp\left(\left|p - \frac{\max_p}{2}\right| \Big/ \frac{\max_p}{2}\right), \quad (5)$$

where p denotes the position of sentence s in the text, $\max_p$ is the maximum position in the text.

Considering these aspects, the sentence weight $e(s)$ is defined as:

$$e(s) = \lambda_2 \cdot H_{\text{entropy}}(s) + H_{\text{key}}(s) + H_{\text{weight}}(s), \quad (6)$$

where $e(s) \in (0, 1)$, and λ_2 is a weight factor for adjusting the impact of H_{entropy}, which we set to 1.

3.2.2 Forward Diffusion

To apply continuous noise to discrete textual data, TermDiffuSum incorporates a sentence encoding module based on stacked transformer layers. This module converts the document $D = \{s_1^d, s_2^d, \ldots, s_n^d\}$ and the reference summary $S = \{s_1^s, s_2^s, \ldots, s_m^s\}$ into continuous representations $\mathbf{H}^{in} = \mathbf{H}^d \| \mathbf{H}^s \in \mathbb{R}^{(n+m)\times h}$. Afterwards, TermDiffuSum obtains the initial state $\mathbf{x}_0$ of input data through a one-step Markov transition:

$$\mathbf{H}^{in} = \text{MLP}(\text{Encoder}_s(D)) \| \text{MLP}(\text{Encoder}_s(S)), \quad (7)$$

$$\mathbf{x}_0 = \mathbf{x}_0^d \| \mathbf{x}_0^s \sim \mathcal{N}(\mathbf{H}^{in}, \beta_0 \mathbf{I}), \quad (8)$$

where $\text{Encoder}_s(\cdot)$ denotes the sentence encoder module, $\mathbf{x}_0^d \in \mathbb{R}^{n\times h}$ and $\mathbf{x}_0^s \in \mathbb{R}^{m\times h}$ are the initial states of the document and the reference summary, respectively. β_0 is the variance of Gaussian noise at diffusion step 0.

Inspired by DiffuSum (Zhang et al., 2023a), we optimize the sentence encoder module using the objective function $\mathcal{L}_{se}$. Besides, TermDiffuSum adopts the partial noise schedule proposed by DiffuSeq (Luo et al., 2023). During the forward process, Gaussian noise is injected into the summary sentence representations $\mathbf{x}_0^s$ based on sentence weight, while the document sentence representations $\mathbf{x}_0^d$ remain unchanged. After T diffusion steps, $\mathbf{x}_0^s$ becomes entirely noise, yielding a series of latent variables $\{\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots, \mathbf{x}_T^s\}$. The forward process is represented as follows:

$$\mathbf{x}_t = \mathbf{x}_0^d \| \mathcal{N}(\mathbf{x}_t^s; \sqrt{1-\beta_t^s}\mathbf{x}_{t-1}^s, \beta_t^s \mathbf{I}), \quad (9)$$

$$\beta_t^i = \sqrt{\frac{t}{T} + \gamma} + \lambda(t)\, e(s_i), \quad (10)$$

$$\lambda(t) = \lambda_w \sin\left(\frac{t}{T}\pi\right), \quad (11)$$

where $t \in \{1, 2, \ldots, T\}$, β_t^s is the variance of Gaussian noise at step t. $\sqrt{1-\beta_t^s}$ is the mean of Gaussian noise, and $e(s_i)$ denotes the importance of the i-th sentence in the summary. The constant γ corresponds to the initial noise level. Following Diffusionbert (He et al., 2022), $\lambda(t)$ is used

to adjust the impact of the noise at step t. For summary sentence representations $\mathbf{x}_t^{s_i}$ and $\mathbf{x}_t^{s_j}$, if $e(s_i) > e(s_j)$, then $\beta_t^i > \beta_t^j$ to encourage TermDiffuSum to prioritize adding noise to sentences with high importance.

3.2.3 Reverse Diffusion

After obtaining the noised representations $\mathbf{x}_t = \mathbf{x}_0^d \| \mathbf{x}_t^s$ at diffusion step t, the reverse process is performed. TermDiffuSum denoises $\mathbf{x}_t$ by predicting the noise distribution at step t-1. The reverse process is represented as:

$$p_\theta(\mathbf{x}_{t-1}^s | \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}^s; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta^2(t)\mathbf{I}\right), \quad (12)$$

where $\mu_\theta(\cdot)$ and $\sigma_\theta^2(\cdot)$ are the mean and variance of the predicted noise distribution. The diffusion module aims to minimize the difference between the predicted summary representations $\tilde{\mathbf{H}}_0^s = \left[\tilde{\mathbf{h}}_1^s, \tilde{\mathbf{h}}_2^s, \ldots, \tilde{\mathbf{h}}_m^s\right]$ and the reference summary $\mathbf{H}^s$. The objective function is:

$$\begin{aligned}\mathcal{L}_{\text{diffusion}} = &\sum_{t=2}^{T} \left\| \mathbf{x}_0 - \tilde{f}_\theta(\mathbf{x}_t, t) \right\|^2 \\ &+ \left\| \mathbf{H}^{in} - \tilde{f}_\theta(\mathbf{x}_1, t) \right\|^2 \\ &+ \mathcal{R}(\mathbf{x}_0),\end{aligned} \quad (13)$$

where $\tilde{f}_\theta(\mathbf{x}_t, t)$ represents the predicted text representation at step t and $\mathcal{R}(\mathbf{x}_0)$ is the L-2 regularization term of $\mathbf{x}_0$.

3.3 Re-ranking Module

Given the summary representations $\tilde{\mathbf{H}}_0^s = \left[\tilde{\mathbf{h}}_1^s, \tilde{\mathbf{h}}_2^s, \ldots, \tilde{\mathbf{h}}_m^s\right]$, TermDiffuSum selects sentences from the document that are most similar to $\tilde{\mathbf{h}}_i^s$ as candidate summary sentences, assuming $\text{Sents} = \left\{s_1^d, s_2^d, \ldots, s_m^d\right\}$. The model then combines the candidate summary sentences to obtain k candidate summaries $\text{Cands} = \{C_1, C_2, \ldots, C_k\}$, and C_i is a subset of the Sents. Candidate summaries are ranked in descending order based on the ROUGE metric. TermDiffuSum employs the evaluation function $f(S)$ to score the candidate summaries, where a higher score indicates a greater similarity to the document. The re-ranking loss is defined as:

$$\mathcal{L}_{\text{ctr}} = \sum_i \sum_{j>i} \max(0, f(C_j) - f(C_i) + \rho), \quad (14)$$

$$f(S) = \cos(\mathbf{H}^d, \mathbf{H}^s). \quad (15)$$

For a pair of candidate summaries (C_i, C_j), $i < j$, then $\text{ROUGE}(C_i) > \text{ROUGE}(C_j)$. Specifically, $\mathbf{H}^d$, $\mathbf{H}^{C_i}$ and $\mathbf{H}^{C_j}$ are the representations of D, C_i, and C_j respectively. ρ is the margin parameter. The re-ranking module allows TermDiffuSum to perceive the ROUGE scores of summaries.

Dataset	Domain	Train	Val	Test	#Doc	#Sum
LegalAFSum	Legal	629	110	107	1,380	487
CNN/DM	News	287,084	13,367	11,489	766	58
XSum	News	204,045	11,332	11,334	430	23
Reddit	Social Media	33,794	4,213	4,222	385	20

Table 1: Statistical data of the experimental datasets. #Doc and #Sum represent the average tokens in the document and summary.

3.4 Optimization and Prediction

The overall objective function is:

$$\mathcal{L} = \mathcal{L}_{\text{se}} + \mathcal{L}_{\text{diffusion}} + \mathcal{L}_{\text{ctr}}. \quad (16)$$

In the inference stage, which involves only the reverse process, TermDiffuSum first encodes the document D into $\mathbf{H}^d$ and performs a one-step Markov transition. The model then randomly samples m noise vectors from Gaussian noise to construct noised summary representations $\mathbf{x}_T^s \sim \mathcal{N}(0, \mathbf{I})$ at step T. During the reverse process, TermDiffuSum gradually denoises $\mathbf{x}_T = \mathbf{x}_0^d \| \mathbf{x}_T^s$ to obtain the predicted embeddings of the summary $\tilde{\mathbf{H}}_0^s = [\tilde{\mathbf{h}}_1^s, \tilde{\mathbf{h}}_2^s, \ldots, \tilde{\mathbf{h}}_m^s]$.

Finally, based on the similarity between $\tilde{\mathbf{h}}_i^s$ ($i \in [1, m]$) and document representations $\mathbf{H}^d$, TermDiffuSum maps $\tilde{\mathbf{h}}_i^s$ to the corresponding sentences in the document and selects the most closely matching sentences as summary sentences.

4 Experiments

In this section, we conduct experiments on multiple datasets to explore the following questions:

- **RQ1**: How does the performance of TermDiffuSum compare with existing extractive summarization methods?
- **RQ2**: How is the scalability and adaptability of TermDiffuSum?
- **RQ3**: How do the different modules influence the performance of TermDiffuSum?
- **RQ4**: How do the main hyperparameters affect TermDiffuSum?
- **RQ5**: Can the re-ranking module and multifactor fusion noise weighting schedule improve the quality of the generated summaries?

4.1 Experimental Setup

4.1.1 Datasets

Given the current scarcity of high-quality extractive legal summarization datasets (Bhattacharya et al., 2021; Shukla et al., 2022), we collect legal documents from the United Kingdom[3] and Singapore[4] to construct a dataset. We employ six law school undergraduates for annotation. According to guidelines from legal professionals (Giles, 2015; Pyle et al., 2017) and related research (Shukla et al., 2022; Zhong et al., 2019), legal documents contain various rhetorical segments and it is necessary to summarize each segment individually. This method makes the summary more user-friendly and better meets the needs of practitioners who focus on specific sections. Consequently, annotators divide the documents into five sections: *Analysis*, *Arguments*, *Facts*, *Judgments*, and *Statutes*. Given that the *Analysis* and *Facts* sections are particularly detailed and redundant, we select these sections for summarization. Therefore, the annotators are asked to annotate the summaries of these two parts. To this end, we construct an extractive legal summarization dataset comprising 846 document-summary pairs, named LegalAFSum. Additionally, to validate the generalizability of our model, we conduct experiments on three widely used datasets: CNN/DM (Hermann et al., 2015), XSum (Narayan et al., 2018), and Reddit (Kim et al., 2018). Detailed statistical information on these datasets is shown in Table 1. Further details about these datasets and the construction process of LegalAFSum can be found in Appendix A.3.

4.1.2 Baselines

We evaluate our model against a variety of baseline models, which are categorized into three groups: 1) General traditional methods: ORACLE, LEAD-K; 2) General deep learning-based methods: BERTSUM (Liu, 2019), MATCHSUM (Zhong et al., 2020b), CoLo (An et al., 2022), DiffuSum (Zhang et al., 2023a), and ChatGPT; 3) Legal domain-specific models: Gist (Liu and Chen, 2019). Details about the baselines are provided in Appendix A.4.

4.1.3 Implementation Details and Metrics

We employ Sentence-BERT (Reimers and Gurevych, 2019) as the initial encoder. The sentence encoder module is constructed using 8 stacked transformer encoders(Vaswani et al., 2017). We use another 12 stacked transformer encoders for the diffusion module. The hidden size of TermDiffuSum is 768. The maximum diffusion step T is set to 500, and weight parameters $\lambda_1 = 0.5$ and $\lambda_w = 0.05$. For optimization, we utilize the AdamW optimizer (Kingma and Ba, 2015) with a learning rate of $1e-5$ and a dropout rate of 0.1. The batch size is 32, and the number of candidate summaries is limited to 35. We randomly select five seeds for training our model and report the average score. The model is trained on GeForce RTX 3090 GPU. More details on the baselines are in Appendix A.5. We use keywords from publicly available datasets for experiments on CNN/DM, XSum, and Reddit. Detailed information about the keyword datasets is provided in Appendix A.6.

Following prior research (An et al., 2022; Liu et al., 2022), we select ROUGE[5] (Lin, 2004) as the evaluation metric. ROUGE-1, ROUGE-2, and ROUGE-L evaluate summaries by comparing the overlap of unigrams, bigrams, and the longest common subsequence with reference summaries.

Model	R-1	R-2	R-L
BERTSUM + LSTM	59.87	51.17	59.23
BERTSUM + Classifier	60.65	52.01	59.99
BERTSUM + Transformer	59.88	51.22	59.17
MATCHSUM (BERT-base)	62.15	51.32	62.68
MATCHSUM (RoBERTa-base)	62.10	51.31	62.63
CoLo$_{Ext}$	61.76	49.39	61.32
CoLo$_{Ext}$ + BERTScore	61.94	49.15	61.83
DiffuSum	61.47	49.88	60.21
ChatGPT	44.79	30.78	43.74
Gist	60.76	50.79	60.55
TermDiffuSum	**64.57**	**52.72**	**63.10**
w/o Re-ranking	62.08	50.51	60.71
w/o Multifactor fusion noise	63.83	51.53	62.21

Table 2: Experimental results on the LegalAFSum dataset. Bold indicates the best results. R-1/2/L represents ROUGE-1, ROUGE-2, and ROUGE-L.

4.2 Main Performance Comparison (RQ 1)

This section presents the performance of TermDiffuSum on the LegalAFSum dataset. Table 2 shows the results, as follows: 1) The first part of the table presents the performance of deep learning-based models. While BERTSUM excels in R-2, it falls short compared to MATCHSUM, CoLo, and DiffuSum in R-1 and R-L. This suggests that summary-level models may have advantages over

[3] https://www.supremecourt.uk/current-cases/

[4] https://www.elitigation.sg/

[5] https://pypi.org/project/rouge/

Model	CNN/DM			XSum			Reddit		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
ORACLE	52.59	31.23	48.87	25.62	7.62	18.72	29.10	11.80	23.10
LEAD-K	40.43	17.62	36.67	14.40	1.46	10.59	12.38	2.17	10.12
BERTSUM	43.25	20.24	39.63	22.86	4.48	17.16	23.86	5.85	19.11
MATCHSUM	44.41	20.86	40.55	24.86	4.66	18.41	25.09	6.17	20.13
CoLo	44.58	21.25	40.65	24.51	5.04	18.21	25.06	5.90	19.25
DiffuSum	44.83	**22.56**	40.56	24.00	**5.44**	18.01	25.17	5.40	20.41
TermDiffuSum	**46.18**	21.96	**42.28**	**25.18**	5.37	**18.51**	**26.63**	**6.22**	**21.77**

Table 3: Experimental results on the CNN/DM, XSum, and Reddit datasets. Bold indicates the best results.

sentence-level models and highlights the potential of diffusion models in summarization tasks. Notably, ChatGPT performs the worst, indicating that it is less suitable for extractive legal summarization. 2) TermDiffuSum achieves the best results across all metrics. Compared to the diffusion-based model DiffuSum, TermDiffuSum improves by 3.10 in R-1, 2.84 in R-2, and 2.89 in R-L. These results underscore the effectiveness of the re-ranking module and multifactor fusion noise weighting schedule, which enhance the model's ability to leverage ROUGE scores and adapt to the linguistic features of legal documents.

4.3 Performance on Other Datasets (RQ 2)

We evaluate TermDiffuSum on the CNN/DM, XSum, and Reddit datasets, as shown in Table 3. Our findings are: 1) When applied to shorter datasets like CNN/DM and XSum, TermDiffuSum exhibits a slightly lower R-2 score compared to DiffuSum. The results indicate that the binary matching performance of TermDiffuSum is suboptimal. 2) Across the three datasets, TermDiffuSum shows significant improvements over all baselines. Notably, on XSum, its ROUGE scores are nearly equivalent to those of ORACLE, highlighting the model's efficacy in handling texts of different lengths and domains.

4.4 Ablation Study (RQ 3)

Table 2 also presents the results of the ablation study. The corresponding findings are as follows: 1) When the multifactor fusion noise weighting schedule is replaced with the sqrt noise schedule, the performance of TermDiffuSum exhibits a declining trend. The decline is more pronounced in the absence of the re-ranking module. The results suggest that the re-ranking module contributes more significantly to TermDiffuSum. 2) When both the re-ranking module and multifactor fusion noise weighting schedule are removed, TermDiffuSum reverts to DiffuSum, suffering the most drastic performance decline. This confirms that both components are complementary.

4.5 Impact of Hyper-parameters (RQ 4)

In this section, we will further explore the impact of hyper-parameters on the model.

4.5.1 Impact of Diffusion Steps

This section compares TermDiffuSum with different diffusion steps, as shown in Figure 4a. The conclusions are: 1) When $T < 500$, TermDiffuSum's performance improves with increasing T. This could be because smaller diffusion steps result in sparser information density, making it harder for TermDiffuSum to distinguish subtle differences in the data. 2) When $T > 500$, TermDiffuSum's performance slightly declines with increasing T. This decrease is likely attributed to excessively large diffusion steps, which introduce excessive noise and hinder the recovery of data features.

4.5.2 Impact of Noise Weight

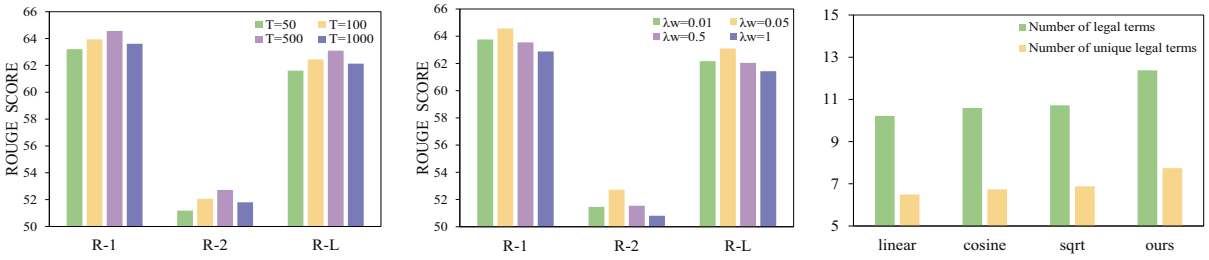
We evaluate TermDiffuSum with λ_w values of 0.01, 0.05, 0.5, and 1, as shown in Figure 4b. The findings are: 1) Performance drops significantly when λ_w deviates from 0.05. For $\lambda_w = 1$, the R-1 score decreases by 1.69 points, likely due to excessive initial noise that obscures text features and impairs the model's ability to process textual information effectively. 2) When λ_w is small, the multifactor fusion noise weighting schedule degrades to the sqrt noise schedule, resulting in reduced performance. This indicates that our schedule benefits the model performance.

4.5.3 Impact of Noise Schedule

This section evaluates the number of legal terms and unique legal terms generated by TermDiffuSum using linear, cosine, sqrt, and multifactor fusion noise weighting schedules. The results are shown in Figure 4c. We find that: 1) The sqrt noise

System	Summary
Reference	Mr. Gopi and Mdm. Sartha were married on 11 September. They were divorced in 2004 on the ground that they had been separated for more than 4 years. The decree absolute was granted on 22 July. At the material time, Mr. Gopi and Mdm. Sartha had children, and the latter was granted custody, care and control of all their children. Mr. Gopi appealed against DJ Tan's decision on maintenance and division of the matrimonial property. The appeal was heard in May 2004 by VK Rajah JC, as he then was, who dismissed the appeal and ruled that DJ Tan's orders on maintenance and division of the matrimonial property were to stand. Subsequently, Mr. Gopi sold the matrimonial property and purchased another flat for himself without paying Mdm. Sartha her 25% share of the nett proceeds of sale of the matrimonial property. DJ Tan also ordered Mr. Gopi to pay Mdm. Sartha arrears in maintenance fees amounting to $6. Mr. Gopi appealed against DJ Tan's decision.
DiffuSum	Mr. Gopi and Mdm. Sartha were married on 11 September. They were divorced in 2004 on the ground that they had been separated for more than 4 years. The decree absolute was granted on 22 July. At the material time, Mr. Gopi and Mdm. Sartha had 3 children, and the latter was granted custody, care and control of all their children. Mr. Gopi was to pay Mdm. Sartha $50 a month for her maintenance, $300 a month for her maintenance of their second child, and $250 a month for the maintenance of their third child. Mr. Gopi was required to put back into his own Central Provident Fund ("CPF") account the amount utilized from the said account for the purchase of the matrimonial property. Subsequently, Mr. Gopi sold the matrimonial property and purchased another flat for himself without paying Mdm. Sartha her 25% share of the nett proceeds of sale of the matrimonial property. DJ Tan also ordered Mr. Gopi to pay Mdm. Sartha arrears in maintenance fees amounting to $6.8. Mr. Gopi appealed against DJ Tan's decision.
Ours	Mr. Gopi and Mdm. Sartha were married on 11 September. They were divorced in 2004 on the ground that they had been separated for more than 4 years. The decree absolute was granted on 22 July. At the material time, Mr. Gopi and Mdm. Sartha had 3 children, and the latter was granted custody, care and control of all their children. Mr. Gopi was to pay Mdm. Sartha $50 a month for her maintenance, $300 a month for her maintenance of their second child, and $250 a month for the maintenance of their third child. Mr. Gopi appealed against DJ Tan's decision on maintenance and division of the matrimonial property. The appeal was heard in May 2004 by VK Rajah JC, as he then was, who dismissed the appeal and ruled that DJ Tan's orders on maintenance and division of the matrimonial property were to stand. Subsequently, Mr. Gopi sold the matrimonial property and purchased another flat for himself without paying Mdm. Sartha her 25% share of the nett proceeds of sale of the matrimonial property. DJ Tan also ordered Mr. Gopi to pay Mdm. Sartha arrears in maintenance fees amounting to $6. Mr. Gopi appealed against DJ Tan's decision.

Figure 3: Case study analysis on the LegalAFSum dataset. Green text indicates sentences that appear in the reference summary, while red text denotes legal terminology.



(a) Model performance with different diffusion steps T.

(b) Model performance with different noise weights.

(c) Model performance with different noise schedules.

Figure 4: Experimental results on the legal dataset.

schedule outperforms linear and cosine schedules. This is because the sqrt schedule introduces greater noise in the initial stage and is more suitable for text data which is less sensitive to noise. 2) When using the multifactor fusion noise weighting schedule, both the number of legal terms and unique legal terms increase. This indicates that the schedule helps the model understand the semantic information of legal documents.

4.6 Case Study (RQ 5)

To address RQ5, we visualize a case to evaluate the summaries generated by TermDiffuSum and DiffuSum, as shown in Figure 3. We observe that: 1) DiffuSum fails to recognize the fifth and sixth sentences in the reference summary, whereas TermDiffuSum identifies all these sentences. Additionally, while DiffuSum identifies only seven unique legal terms, TermDiffuSum successfully captures all the legal terms present in the reference. This discrepancy may be because the fifth and sixth sentences contain three and five legal terms, respectively. These sentences attract more attention from TermDiffuSum. 2) The fifth and sixth sentences in DiffuSum's summary are not present in the reference. TermDiffuSum correctly excludes the sixth sentence but includes the fifth sentence, which contains three legal terms. We argue that not all legal terms are equally important. The multifactor fusion noise weighting schedule also directs TermDiffuSum's attention to legal terms that are not present in the reference, which may lead to a decline in model performance.

5 Conclusions

This paper introduces TermDiffuSum, a novel diffusion-based model for the extractive summarization of legal documents. We propose a multifactor fusion noise weighting schedule that directs the model's attention to sentences with legal terms. Additionally, we introduce a re-ranking module to en-

hance the model's ability to perceive more relevant summaries. We construct a legal summarization dataset, LegaAFSum. Experiments on LegaAFSum and other datasets validate the effectiveness of our approach. Future research will focus on refining noise schedules and further optimizing diffusion models for extractive summarization.

Limitations

Our study has the following limitations. Firstly, due to the high cost of constructing a legal extractive summarization dataset, our dataset is relatively small, which may impact model performance. Secondly, our model focuses on summarizing the *Analysis* and *Facts* sections of legal documents. In practical applications, this approach may require additional role recognition models to ensure comprehensive summaries.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (2022YFC3303400).

References

Chenxin An, Ming Zhong, Zhiyong Wu, Qin Zhu, Xuanjing Huang, and Xipeng Qiu. 2022. Colo: A contrastive learning based re-ranking framework for one-stage summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5783–5793. International Committee on Computational Linguistics.

Deepa Anand and Rupali Wagh. 2022. Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University-Computer and Information Sciences*, 34(5):2141–2150.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.

Christian Bentz and Dimitrios Alikaniotis. 2016. The word entropy of natural languages. *arXiv preprint arXiv:1606.06996*.

Paheli Bhattacharya, Soham Poddar, Koustav Rudra, Kripabandhu Ghosh, and Saptarshi Ghosh. 2021. Incorporating domain knowledge for extractive summarization of legal case documents. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 22–31.

Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2023. How ready are pre-trained abstractive models and llms for legal case judgement summarization? *arXiv preprint arXiv:2306.01248*.

Atefeh Farzindar and Guy Lapalme. 2004. Letsum, an automatic legal text summarizing system. *Proc. Legal knowledge and information systems (JURIX)*.

Diego Feijo and Viviane Moreira. 2019. Summarizing legal rulings: Comparative experiments. In *proceedings of the international conference on recent advances in natural language processing (RANLP 2019)*, pages 313–322.

Filippo Galgani, Paul Compton, and Achim Hoffmann. 2015. Summarization based on bi-directional citation analysis. *Information processing & management*, 51(1):1–24.

Jessica Giles. 2015. Writing case notes and case comments. Last accessed: July 28, 2024.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2022. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020a. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020b. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Yuxin Huang, Zhengtao Yu, Junjun Guo, Zhiqiang Yu, and Yantuan Xian. 2020. Legal public opinion news abstractive summarization by incorporating topic information. *International Journal of Machine Learning and Cybernetics*, 11:2039–2050.

Srinivasan Iyer, Sewon Min, Yashar Mehdad, and Wentau Yih. 2021. Reconsider: improved re-ranking using span-focused cross-attention for open domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1280–1287.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2021a. Automatic summarization of legal bills: A comparative analysis of classical extractive approaches. In *2021 international conference on computing, communication, and intelligent systems (ICCCIS)*, pages 394–400. IEEE.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2021b. Summarization of legal documents: Where are we now and the way forward. *Computer Science Review*, 40:100388.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2024. A sentence is known by the company it keeps: Improving legal document summarization using deep clustering. *Artificial Intelligence and Law*, 32(1):165–200.

Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review*, 51:371–402.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2018. Abstractive summarization of reddit posts with multi-level memory networks. 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)*, pages 2519–2531. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yuquan Le, Sheng Xiao, Zheng Xiao, and Kenli Li. 2024. Topology-aware multi-task learning framework for civil case judgment prediction. *Expert Systems with Applications*, 238:122103.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chao-Lin Liu and Kuan-Chun Chen. 2019. Extracting the gist of chinese judgments of the supreme court. In *proceedings of the seventeenth international conference on artificial intelligence and law*, pages 73–82.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903. Association for Computational Linguistics.

Yuanzhen Luo, Qingyu Zhou, and Feng Zhou. 2023. Enhancing phrase representation by information bottleneck guided text diffusion process for keyphrase extraction. *arXiv preprint arXiv:2308.08739*.

Luís Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P Neto. 2013. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. *arXiv preprint arXiv:1306.4886*.

Gianluca Moro and Luca Ragazzi. 2022. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11085–11093.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. Association for Computational Linguistics.

Vinay Pandramish and Dipti Misra Sharma. 2020. Checkpoint reranking: An approach to select better hypothesis for neural machine translation systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 286–291.

Seth Polsley, Pooja Jhunjhunwala, and Ruihong Huang. 2016. Casesummarizer: a system for automated summarization of legal texts. In *Proceedings of COLING 2016, the 26th international conference on Computational Linguistics: System Demonstrations*, pages 258–262.

C. Pyle, K. Killoran, and M. Richards. 2017. How to brief a case. Online. Created by C. Pyle, 1982. Revised by K. Killoran, 1999 and by M. Richards, 2017.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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)*, pages 3982–3992. Association for Computational Linguistics.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Diffusionner: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890. Association for Computational Linguistics.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. *arXiv preprint arXiv:2210.07544*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Renzhi Wang, Jing Li, and Piji Li. 2023. Infodiffusion: Information entropy aware diffusion process for non-autoregressive text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13757–13770. Association for Computational Linguistics.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. *arXiv preprint arXiv:1911.02671*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Diffusum: Generation enhanced extractive summarization with diffusion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13089–13100. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023b. Extractive summarization via chatgpt for faithful summary generation. *arXiv preprint arXiv:2304.04193*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020a. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

Linwu Zhong, Ziyi Zhong, Zinian Zhao, Siyuan Wang, Kevin D Ashley, and Matthias Grabmair. 2019. Automatic summarization of legal decisions using iterative masking of predictive sentences. In *Proceedings of the seventeenth international conference on artificial intelligence and law*, pages 163–172.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020b. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208. Association for Computational Linguistics.

A Appendix

A.1 Related Work of Re-ranking

Re-ranking is widely applied in natural language processing tasks such as machine translation (Pandramish and Sharma, 2020), question answering (Iyer et al., 2021) and summarization (Liu and Liu, 2021; Liu et al., 2022). In extractive summarization, MATCHSUM (Zhong et al., 2020b) evaluates candidate summaries based on semantic matching between the candidates and the original text. CoLo (An et al., 2022) implements a one-stage re-ranking framework for summarization. Unlike these works, we apply re-ranking to diffusion models. TermDiffuSum integrates evaluation metrics into the objective function through re-ranking, enabling the model to exploit the relationship between candidate summaries and reference summaries.

A.2 Background of Diffusion Models

Diffusion models are a class of latent variable models that include a forward process and a reverse process (Ho et al., 2020b; Sohl-Dickstein et al., 2015). The forward process gradually introduces Gaussian noise to corrupt input data, while the reverse process trains the model to recover the original data from the Gaussian noise.

In the forward process, given the input data distribution $\mathbf{x}_0 \sim q(x)$, the forward process gradually adds Gaussian noise to $\mathbf{x}_0$, producing a Markov chain of latent variables $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \quad (17)$$

where $\beta_t \in (0, 1)$ controls the amount of noise at each time step, and T is the total number of diffusion steps. Eventually, $\mathbf{x}_T$ becomes a Gaussian distribution.

After obtaining a series of noised data, the reverse process involves recovering the original data using a learned parameterized model $p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$:

$$p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_\theta^2\left(t\right)\mathbf{I}\right), \quad (18)$$

where $\mu_\theta(\cdot)$ and $\sigma_\theta^2(\cdot)$ denote the predicted mean and variance, which can be implemented by a neural network like U-Net (Ronneberger et al., 2015) or Transformer (Vaswani et al., 2017).

The diffusion model is trained by optimizing a variational upper bound of $-\log p_\theta(\mathbf{x}_0)$. In this paper, to enhance training stability, we adopt the simplified loss function proposed by (Ho et al., 2020b):

$$\mathcal{L}_{\text{simple}} = \sum_{t=1}^{T} \|\mathbf{x}_0 - \tilde{f}_\theta(\mathbf{x}_t, t)\|^2, \quad (19)$$

where $\tilde{f}_\theta(\mathbf{x}_t, t)$ is the predicted $\mathbf{x}_0$ at step t.

A.3 Dataset Details

A.3.1 LegalAFSum Dataset

To the best of our knowledge, there is currently a lack of high-quality legal extractive summarization datasets (Bhattacharya et al., 2021; Shukla et al., 2022). Most publicly available datasets are of small size, such as the IN-Ext dataset (Shukla et al., 2022), which consists of only 50 samples. To address this gap, we construct a new legal extractive dataset. The annotation process is as follows:

- **Data collection and preparation**: We collect 468 legal documents from two countries: the United Kingdom[6] and Singapore[7]. Then, we hire six law school undergraduates to annotate the dataset.

- **Case structure split**: According to the guidelines for legal document summaries by legal professionals (Giles, 2015; Pyle et al., 2017) and other research (Shukla et al., 2022; Zhong et al., 2019), legal documents have various rhetorical segments and it is necessary to summarize each segment separately. This method enhances the clarity and organization of the summary by providing detailed information across different levels. Furthermore, it effectively meets the needs of legal practitioners who only focus on specific sections of the summary (such as the arguments or statutes). Therefore, annotators split legal texts into five parts based on their thematic structure: *Analysis*, *Arguments*, *Facts*, *Judgments*, and *Statutes*. *Analysis* refer to the judge's examination of legal issues. *Arguments* are the arguments of the contending parties or lawyers. *Facts* are statements of the case details. *Judgments* represent the final rulings. *Statutes* are the established laws involved in the document.

- **Extractive summaries**: The *Arguments*, *Judgments*, and *Statutes* are typically brief (Bhattacharya et al., 2021). Conversely, the *Analysis* and *Facts* sections are more extensive and redundant, prompting this work to focus on these parts. Based on this preference, the six undergraduates summarize *Analysis* and *Facts* separately by selecting important sentences to compose summaries.

- **Extract legal terms**: The six undergraduates identify legal terms from legal documents, forming the basis of our legal terms dataset.

In this way, we construct the Legal Analysis and Facts summarization dataset, named LegalAFSum, which contains 846 document-summary pairs.

A.3.2 Other Datasets

In addition, we also use three public datasets in the news and social media domains to verify the generalizability of our model. The details of these datasets are as follows:

- **CNN/DM**[8] (Hermann et al., 2015) comprises 93k articles from CNN and 220k from the Daily Mail. In this study, the non-anonymous version is used.

- **XSum**[9] (Narayan et al., 2018) is a highly abstractive article dataset comprising 227k articles from the British Broadcasting Corporation (BBC).

- **Reddit**[10] (Kim et al., 2018) comprises 120k posts from social media platforms. We employ the TIFU-long version for analysis.

[6] https://www.supremecourt.uk/current-cases/
[7] https://www.elitigation.sg/
[8] https://cs.nyu.edu/~kcho/DMQA/
[9] https://github.com/EdinburghNLP/XSum
[10] https://github.com/ctr4si/MMN

Model	TermDiffuSum			DiffuSum		
	R-1	R-2	R-L	R-1	R-2	R-L
seed=100	64.32	52.42	62.81	61.46	49.62	60.11
seed=101	64.57	52.61	63.25	61.72	50.08	60.34
seed=201	64.89	52.98	63.33	61.50	50.13	60.35
seed=901	64.15	52.35	62.67	61.18	49.38	59.94
seed=1001	64.92	53.25	63.42	61.48	50.21	60.29
Avg	64.57	52.72	63.10	61.47	49.88	60.21
Variance	0.0928	0.1175	0.0893	0.0295	0.1058	0.0251

Table 4: Performance comparison of TermDiffuSum and DiffuSum with various random seeds.

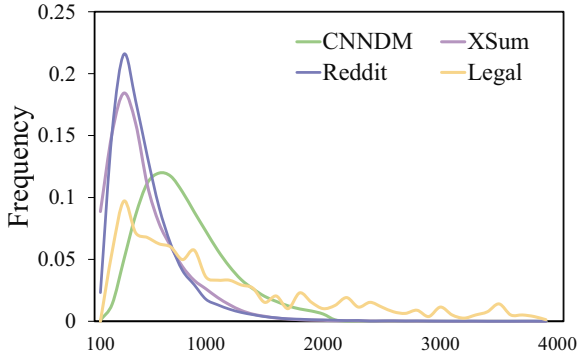


Figure 5: Description of dataset length distribution.

The distribution of data lengths across all datasets is presented in Figure 5.

A.4 Baselines

We choose a variety of models as baselines, which can be summarized into three groups.

- **General traditional methods**, including ORACLE, which obtains summaries by maximizing the ROUGE scores through a greedy algorithm. ORACLE represents the upper bound of extractive summarization; LEAD-K is an unsupervised method that selects the first K sentences of a document as summary.

- **General deep learning based methods**, including BERTSUM (Liu, 2019), a sentence-level extractive model that utilizes BERT to score and select sentences; MATCHSUM (Zhong et al., 2020b), a summary-level summarization model that formulates extractive summarization as a semantic matching task; CoLo (An et al., 2022), a summary-level extractive summarization model that incorporates a re-ranking module; DiffuSum (Zhang et al., 2023a), an extractive summarization model based on diffusion model; ChatGPT, a large language model for generating and understanding natural language text.

- **Legal domain-specific models**, including Gist (Liu and Chen, 2019), which is a supervised extractive summarization model of legal documents. Gist categorizes sentences using classifiers based on machine learning and deep learning.

A.5 Implementation Details for Baselines

In this research, we employ publicly available implementations of the baseline models, BERTSUM[11], MATCHSUM[12], CoLo[13], and DiffuSum[14]. For the Gist model, given the unavailability of an official implementation, we utilize the code provided by Shukla et al. (2022)[15].

For CNN/DM, we use the data from the original papers. For XSum and Reddit, if there is no relevant data in the original paper, we follow the settings in the source code to obtain the evaluation results.

A.6 Keyword Datasets for Experiments

As mentioned in Appendix A.3, while annotating the LegalAFSum, we also annotated legal terms. For experiments on LegalAFSum, we train TermDiffuSum using this legal terms dataset. For experiments on CNN/DM and XSum, we use keywords from the 500N-KPCrowd dataset (Marujo et al., 2013); for experiments on Reddit, we use keywords from the OpenKP dataset (Xiong et al., 2019). Both the 500N-KPCrowd dataset and the OpenKP dataset are publicly available, ensuring

[11] https://github.com/nlpyang/BertSum
[12] https://github.com/maszhongming/MatchSum
[13] https://github.com/ChenxinAn-fdu/CoLo
[14] https://github.com/hpzhang94/DiffuSum
[15] https://github.com/Law-AI/summarization

that our model is not constrained by specific keyword datasets.

A.7 Random Seed Sensitivity Analysis

To assess model stability, we train DiffuSum and TermDiffuSum on the LegalAFSum dataset using five randomly selected seeds (100, 101, 201, 901, 1001). The results are presented in Table 4.

The performance varies with different random seeds for both models. Despite some fluctuations, both models demonstrate a high level of accuracy across all tested seeds. Notably, DiffuSum exhibits lower variance than TermDiffuSum, indicating higher stability.