Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DocMidFuse-MN: Mongolian long-text output safety detection via middle semantic awareness and document-level risk fusion

Hao Wang
Inner Mongolia University of Technology
Gang Wang
wg@imut.edu.cn
Inner Mongolia University of Technology
Yan Li
Inner Mongolia University of Technology

Article

Keywords:

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10689691/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

DocMidFuse-MN: Mongolian long-text output safety detection via middle semantic awareness and document-level risk fusion

Hao Wang[1], Gang Wang[1*], Yan Li[1]

[1*]Inner Mongolia University of Technology, Hohhot, Inner Mongolia, China.

*Corresponding author(s). E-mail(s): wg@imut.edu.cn;

Abstract

With the expansion of large language models into cross-linguistic applications, safety detection for generated content has become important for trustworthy deployment. Experimental observations indicate that, under semantically equivalent instructions, Cyrillic Mongolian outputs are often longer than Chinese and English outputs, causing potential risk information to be dispersed across long texts. Existing safety detection methods are mainly designed for high-resource languages and have limited ability to model Mongolian word order and long-text risk distribution. To address this problem, this paper formulates safety detection for Cyrillic Mongolian long-text model outputs and proposes DocMidFuse-MN. The method first applies sentence-level segmentation to reduce truncation and risk signal dilution. It then introduces an SOV-inspired middle semantic enhancement mechanism to strengthen the representation of risk objects and key semantic information. Finally, it integrates local chunk-level risk, middle semantic risk, and document-level risk for overall safety prediction. A manually reviewed Mongolian output safety detection dataset is also constructed. Under the dataset and experimental protocol of this study, DocMidFuse-MN achieves strong long-text detection performance with the Mongolian-specific encoder relative to the evaluated no-segmentation, hierarchical modeling, and general safety detection baselines, while maintaining competitive performance on short texts.

Keywords: Large language model safety, Mongolian, Long-text safety detection, Low-resource languages, Content moderation, Document-level risk fusion

Introduction

In recent years, the capabilities of conversational artificial intelligence systems have improved rapidly. This progress has been mainly driven by the continuous scaling of autoregressive language modeling in terms of training data, model parameters, and computational resources [1, 2]. Conversational agents built upon large language models have been widely applied in scenarios such as chat assistants, intelligent question answering, text generation, and tool use, and have demonstrated strong capabilities in language understanding, commonsense reasoning, instruction following, and complex task processing [3, 4].

However, as large language models are increasingly deployed in real-world applications, the safety of their generated content has become a prominent concern. Existing studies have shown that pretrained language models may generate toxic, discriminatory, misleading, or otherwise potentially harmful content under specific prompts [5, 6]. Meanwhile, user inputs may also contain adversarial instructions, prompt injection, jailbreak inducement, or attack intentions aimed at bypassing safety policies [7, 8]. Therefore, safety guardrails for generative artificial intelligence applications have gradually become an important component of trustworthy large language model deployment. Relevant practical guidelines also recommend integrating content moderation, adversarial testing, and human oversight in real-world products to detect model inputs and outputs, thereby reducing the likelihood of risk content generation and strengthening system defenses against adversarial inputs and jailbreak attempts [9].

A direct approach to constructing safety guardrails for large language models is to reuse existing online content moderation tools, such as Perspective API [10], OpenAI Moderation API [11], Azure AI Content Safety API [12], and Llama Guard [13]. These tools can detect toxicity, hate speech, violence, sexual content, self-harm, illegal content, or other potentially risky content in text, and have been used in online content governance and human-machine dialogue safety protection scenarios. However, directly applying these general moderation tools to output safety detection for large language models still presents clear limitations. First, many moderation tools rely on fixed safety policies, closed APIs, or predefined risk taxonomies, making them difficult to adapt flexibly to specific languages, scenarios, and risk types. Second, most existing tools and general large language model safety detection methods are designed primarily for high-resource languages such as English and Chinese, while their support for low-resource languages such as Mongolian remains relatively limited. Existing multilingual safety studies have shown that large language models are more likely to suffer from safety degradation or harmful outputs in low-resource language scenarios [14, 15]. Evaluation studies on modern Mongolian have also indicated that current large language models still exhibit clear gaps in Mongolian semantic understanding, knowledge transfer, and reasoning ability [16].

It should be noted that the term Mongolian in this study mainly refers to modern Mongolian written in Cyrillic script, rather than traditional Mongolian script. This scope is adopted because the publicly available general large language models, safety detection models, and related pretrained resources used in current experimental settings still provide limited support for tokenization, encoding, generation, and semantic

understanding of traditional Mongolian script. If traditional Mongolian script were directly used without stable model support and a comparable experimental toolchain, the experimental results would be likely to reflect the limitations of script adaptation and base model coverage, rather than the effectiveness of the proposed long-text output safety detection method itself. Therefore, this study restricts its research object to Cyrillic Mongolian in order to ensure the reproducibility of the experimental setting, the fairness of model comparison, and the practical validity of method evaluation. Traditional Mongolian script remains an important research scenario, but it is more suitable for future extended studies supported by dedicated pretrained models, tokenization tools, and evaluation benchmarks.

As shown in Table 1, under the condition that the instruction semantics are kept consistent, the output token length of large language models in the Cyrillic Mongolian input setting is generally higher than that in Chinese and English. This indicates that, under the experimental setting of this study, Mongolian scenarios are more likely to produce long-text model outputs. As the output length increases, risk information becomes more dispersed throughout the text and may appear at different positions, including the beginning, middle, or end of the generated content.

Table 1 Token Statistics of Model Outputs across Request Types and Languages

Request Type	Language	Qwen2.5 -14B	DeepSeek -V4-pro	DeepSeek -R1-14B	Avg.
Normal	Chinese	187.16 / 1.20%	419.90 / 34.30%	696.70 / 64.10%	434.59 / 33.20%
	English	250.84 / 9.20%	359.55 / 26.50%	914.96 / 82.60%	508.45 / 39.43%
	Mongolian	387.64 / 22.30%	614.36 / 43.60%	705.61 / 63.10%	569.20 / 43.00%
Malicious	Chinese	128.82 / 0.20%	149.26 / 5.90%	540.19 / 44.90%	272.76 / 17.00%
	English	232.85 / 6.40%	205.02 / 9.90%	962.50 / 88.30%	466.79 / 34.87%
	Mongolian	491.65 / 46.90%	481.51 / 41.80%	904.38 / 83.70%	625.85 / 57.47%

Note: Each cell reports the average number of output tokens and the proportion of outputs exceeding 400 tokens.

This phenomenon can be attributed to three main factors, namely linguistic structure, tokenization mechanism, and model generation behavior. First, Cyrillic Mongolian has strong agglutinative characteristics. Grammatical relations such as case, tense, aspect, mood, and syntactic dependency are often expressed through the stacking of suffixes. As a result, the same semantic unit may be divided into more tokens after subword tokenization, thereby increasing the token length of model outputs. Second, Mongolian remains a relatively low-resource language in current large language model ecosystems. Existing large language models and their tokenizers are usually not primarily optimized for Mongolian. Their vocabulary coverage and subword segmentation quality may therefore be weaker than those for high-resource languages such as Chinese and English, leading to higher token consumption for

semantically equivalent content in Mongolian. Finally, because the semantic understanding and instruction alignment capability of large language models for Mongolian remain relatively unstable, models may tend to use more explanatory, supplementary, or repetitive expressions when generating Mongolian responses in order to maintain answer completeness. This behavior can further lengthen the generated output. Therefore, the higher output length of Mongolian shown in Table 1 does not simply indicate that the Mongolian outputs contain more semantic information. Rather, it reflects the combined effects of morphological complexity, differences in tokenization granularity, and generation instability in a low-resource language setting.

The main contributions of this paper are summarized as follows:

- To address the problem that risk information in Cyrillic Mongolian model outputs is more likely to be truncated or diluted due to longer output length, this paper proposes a sentence-level segmentation and local chunk risk modeling strategy. The model output is first divided into sentence-level chunks with relatively complete semantic units, which reduces the loss of risk information located in the middle or later parts of the text under fixed-length input constraints. Each chunk is then modeled separately for safety risk, allowing the detection model to capture local risk signals distributed at different positions in the generated output. This design provides a fine-grained chunk-level risk basis for subsequent middle semantic enhancement and document-level risk fusion.
- To address the limitation that general long-text modeling methods lack Mongolian word-order prior and may fail to sufficiently capture key semantics in the middle or later parts of a sentence, this paper designs an SOV word-order-inspired middle semantic enhancement mechanism. Considering that Cyrillic Mongolian follows a subject object verb word-order tendency, information related to risk objects, affected entities, and sensitive entities may appear in the middle or later regions of a text chunk. Therefore, this paper applies additional pooling to the middle region of each text chunk to enhance the representation of object-related risk semantics. This mechanism is not equivalent to explicit object extraction, nor does it rely on manually annotated syntax or dependency parsing tools. Instead, it serves as a lightweight positional inductive bias to complement the possible insufficient attention of global semantic representations to middle key semantics.
- To address the problem that risk prediction based on a single chunk is insufficient to reflect the safety status of the complete model output, this paper proposes DocMidFuse-MN, a document-level risk fusion method for safety detection. Based on chunk-level risk modeling and middle semantic enhancement, the proposed method further integrates local chunk risk, middle semantic risk, and document-level risk information. This enables the model to focus on local high-risk chunks while maintaining an overall judgment of the risk tendency of the full text. Experimental results show that, under the dataset and experimental protocol of this study, the proposed method achieves favorable effectiveness and stability in long-text detection, short-text evaluation, position sensitivity analysis, and efficiency evaluation.

Materials and methods

This paper selects bert-base-mongolian-cased [25] as the base semantic encoder of DocMidFuse-MN, mainly based on the following considerations. First, the object of this study is model output safety detection for large language models in Mongolian environments, while bert-base-mongolian-cased is a BERT model pretrained for Mongolian text. Compared with general multilingual models, its vocabulary and pretraining corpus are closer to the Mongolian language environment, enabling it to better represent Mongolian words, phrases, and contextual semantics. The public Mongolian BERT project states that it includes pretrained Mongolian BERT models and adopts SentencePiece tokenization with a vocabulary size of 32000 tokens [25]. The bert-base-mongolian-cased model on Hugging Face is provided as a pretrained Mongolian BERT model [25].

Second, the safety detection task is essentially a text semantic classification task, requiring the model to understand whether the context contains risky semantics such as illegal or non-compliant content, malicious guidance, hate, violence, and fraud. BERT [17] adopts a bidirectional Transformer encoding architecture, which can learn text representations by jointly incorporating left and right contexts, and it can be fine-tuned for downstream classification tasks by adding a classification layer. Therefore, bert-base-mongolian-cased is suitable as the chunk-level semantic encoder in the proposed method for extracting risk semantic representations from Mongolian output texts.

It should be noted that the design goal of DocMidFuse-MN is not to propose a new general network architecture that is completely independent of existing long-text classification techniques. Instead, it aims to combine the risk distribution characteristics of long texts with Mongolian word-order priors for the task of Mongolian long-text model output safety detection. Sentence-level segmentation, pooling representation, and document-level aggregation are common techniques in long-document modeling. However, in the scenario of Mongolian model output safety detection, risk information often exhibits position dispersion, normal semantic wrapping, and middle-hidden characteristics. Directly using full-text input or general pooling strategies makes it difficult to stably capture local risky chunks.

Therefore, the core contribution of the proposed method does not lie in the formal novelty of a single module, but in constructing a language-aware risk modeling framework for Mongolian long-text output safety detection. Specifically, sentence-level segmentation is used to alleviate long-text truncation and risk semantic disruption. Middle semantic pooling is based on the subject object verb word order of Mongolian and introduces focused awareness of potential object regions and risk object information in the sentence. Mean / max document-level fusion further models both the overall risk trend of the full text and local high-risk chunks. These three components work together to enhance the model's ability to capture risk information that is wrapped by normal content or dispersed across long texts.

The proposed framework combines sentence-level segmentation to reduce truncation, SOV-inspired middle semantic pooling to strengthen potential object-related semantics, and document-level mean/max fusion to aggregate global and local risk evidence. The overall framework is shown in Fig. 1.

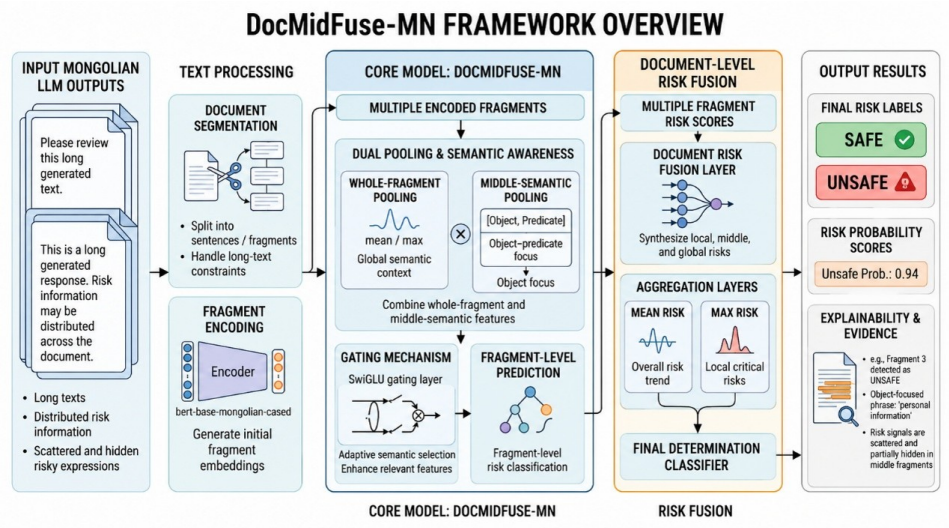


Fig. 1 Overall framework of DocMidFuse-MN. The framework combines sentence-level segmentation, Mongolian-aware middle semantic enhancement, chunk-level risk estimation, and document-level mean/max risk fusion to produce the final safety decision.

The contribution of individual components is further examined through ablation experiments described in the Experiments section.

Threat Model

The threat model studied in this paper mainly targets model output safety detection scenarios for large language models in Mongolian environments. In this scenario, attackers may construct inducing, evasive, or adversarial input prompts to guide large language models to generate Mongolian long-text outputs containing illegal or non-compliant content, malicious guidance, hate, violence, fraud, or other high-risk content. Unlike short-text risk detection, risk information in Mongolian long-text outputs is not necessarily concentrated at the beginning of the text. Instead, it may be distributed in the middle or tail of the document, or even wrapped by a large amount of normal explanation, background description, or harmless semantic content. This covert distribution of risk information weakens the ability of conventional detection methods to capture key risky chunks, thereby increasing the possibility of false negatives. As shown in Fig. 2, this study focuses on a prompt-controlled attack setting in which the attacker attempts to induce unsafe Mongolian long-text outputs while the detection pipeline remains active.

This paper assumes that attackers cannot directly access or modify the parameters, training data, or internal structure of the detection model, nor can they compromise the system deployment environment or bypass the detection pipeline. The main object that attackers can control is the input prompt, through which they induce and influence the output content of large language models. Their attack objective is to embed potential risk information into the generated Mongolian text while reducing

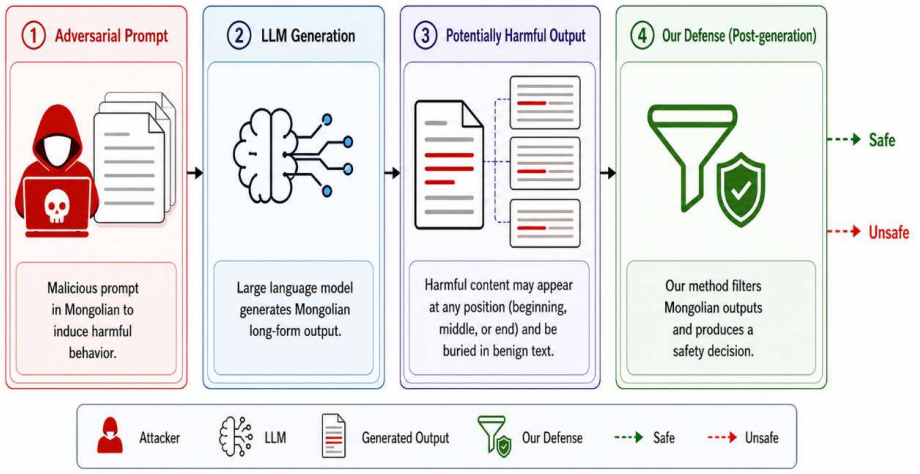


Fig. 2 Threat model of Mongolian long-text output safety detection. The attacker controls the input prompt to induce unsafe Mongolian long-text outputs, while the detector, its training data, and the deployment environment remain inaccessible to the attacker.

the recognition probability of the detection model by exploiting the long-text structure, normal semantic wrapping, and Mongolian syntactic features. This paper does not discuss stronger attack capabilities such as detection model parameter tampering, training data poisoning, or system-level bypassing. Instead, it focuses on how to effectively identify potential risk information concealed by long-text structure, normal semantic content, and Mongolian word-order features under a standard output detection pipeline.

SOV-guided middle semantic pooling mechanism

Mongolian has a relatively typical subject-object-verb word order in its basic syntactic structure. That is, a sentence is usually organized in the order of subject, object, and predicate, with the predicate often appearing at the end of the sentence. Unlike the subject verb object structure commonly observed in Chinese and English, the action target in Mongolian usually appears before the predicate and plays an important semantic role in the middle region of the sentence.

A Mongolian sentence can be abstractly represented as $s = (S, O, V)$, where S denotes the subject, O the object, and V the predicate. For safety detection, the risk attribute depends not only on the action expressed by the predicate but also on the object to which the action is directed. In other words, the model needs to understand both "what action is performed" and "what object the action is performed on." This relationship can be summarized as $r(s) = f_\theta(O, V, C)$, where C denotes contextual information. Compared with subject information, the object O often more directly carries information about the action target, affected entity, sensitive entity, or key event object. Compared with predicate information, the object can also help the model determine whether a certain action has practical safety risk. Therefore, object

semantics constitute an important factor affecting risk judgment in Mongolian model output safety detection.

For example, consider the following Mongolian sentence:

Тэр хувийн мэдээллийг нийтэлсэн.

Its approximate meaning is "He published personal information." Here, S = Тэр, O = хувийн мэдээллийг, and V = нийтэлсэн. хувийн мэдээллийг denotes the object "personal information," while нийтэлсэн denotes the action "published." If the model only focuses on the predicate "published," it may be difficult to determine whether the sentence involves risk. When the model further attends to the object "personal information," it can better identify the potential privacy leakage risk in the sentence. In Mongolian, nouns can express grammatical relations through case markers, and direct objects may take accusative markers in certain contexts. This also indicates that object components play an important role in syntactic and semantic understanding.

Based on the above characteristics, a Mongolian text chunk is represented as $p_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,L_i}]$, where L_i denotes the chunk length and $x_{i,j}$ the j-th token. Since the object in a Mongolian sentence usually appears after the subject and before the predicate, and is often close to the middle region of the text chunk, this paper selects a certain proportion of tokens in the middle of the chunk as the candidate key semantic region:

$$M_i = \{x_{i,j} \mid \lfloor \alpha L_i \rfloor \leq j \leq \lceil \beta L_i \rceil\}, \tag{1}$$

where M_i denotes the middle candidate object region of chunk p_i, and α and β denote the starting and ending ratios, with $0 < \alpha < \beta < 1$. In this paper, the central 20% token region is used as the default middle candidate semantic region, corresponding to $\alpha = 0.4$ and $\beta = 0.6$.

The main reason for selecting the middle region of a chunk is that Mongolian has a clear subject-object-verb structural tendency, and action targets, result descriptions, and restrictive modifiers closely related to risk judgment are more likely to be distributed in the core semantic region of the sentence. Based on this linguistic structural prior, this paper adopts a compact middle window for aggregation modeling, so as to avoid introducing excessive non-critical semantic components from an overly wide region while enhancing the model's awareness of potential risk information in the middle of the chunk. This setting does not depend on external syntactic parsing results. Instead, it serves as an empirical structural prior for improving the stability and interpretability of middle semantic representations.

Subsequently, this paper uses the encoder to extract contextual representations of the chunk and introduces a dual pooling structure at the pooling stage to capture both the overall semantics of the chunk and important semantic information in the middle region of Mongolian sentences. Specifically, the model first applies global average pooling to all valid token representations output by the encoder to obtain the overall semantic representation of the chunk:

$$\boldsymbol{g}_i = \frac{1}{|\mathcal{V}_i|} \sum_{j \in \mathcal{V}_i} \boldsymbol{h}_{i,j}, \tag{2}$$

where $\boldsymbol{g}_i$ denotes the global semantic representation of the i-th chunk, $\boldsymbol{h}_{i,j}$ denotes the contextual representation of the j-th token, and $\mathcal{V}_i$ denotes the set of valid token indices excluding special tokens and padding tokens. This representation is used to characterize the overall contextual information of the chunk.

At the same time, considering that Mongolian follows the subject-object-verb word order, the object usually appears in the sentence middle region, and this region often contains key information that is discriminative for action, intent, and risk attributes. Therefore, this paper further applies average pooling to a certain proportion of valid token representations in the middle region of the chunk to obtain the middle semantic representation:

$$\boldsymbol{m}_i = \frac{1}{|\mathcal{M}_i|} \sum_{j \in \mathcal{M}_i} \boldsymbol{h}_{i,j}, \tag{3}$$

where $\boldsymbol{m}_i$ denotes the middle semantic representation of the i-th chunk, and $\mathcal{M}_i$ denotes the valid token index set corresponding to the middle candidate semantic region. This representation is used to enhance the model's attention to objects, action results, and risk-related semantics in Mongolian sentences.

Finally, this paper concatenates the global semantic representation and the middle semantic representation to form a dual pooling chunk representation:

$$\boldsymbol{z}_i = [\boldsymbol{g}_i; \boldsymbol{m}_i], \tag{4}$$

where $[\cdot;\cdot]$ denotes vector concatenation, and $\boldsymbol{z}_i$ denotes the fused chunk semantic representation. Through this dual pooling structure, the model can preserve the overall contextual information of the chunk while strengthening key semantics related to action and risk judgment in the middle region of Mongolian sentences.

Therefore, the proposed semantic awareness mechanism does not simply rely on a single pooling result for classification. Instead, it jointly models global semantics and Mongolian syntactic key regions through a dual pooling structure, thereby establishing complementary representations between the overall semantic context and the middle syntactic semantic region. This provides richer chunk-level risk features for the subsequent SwiGLU risk classification head.

It should be noted that the subject-object-verb word order in Mongolian does not mean that the object or risk target necessarily falls within the central 20% token region of a text chunk. SOV describes the relative order of syntactic components, whereas the middle window adopted in this paper is a token-position-level modeling approximation. There is no strict one-to-one correspondence between the two. Therefore, this paper does not regard the middle window as a substitute for explicit object extraction or dependency parsing. Instead, it treats it as a lightweight positional inductive bias for enhancing the model's representation of object-related semantics in the middle region of a chunk without relying on external syntactic tools.

From the perspective of safety detection, risk judgment usually depends on multiple types of semantic information, including action predicates, action targets, sensitive entities, restrictive conditions, and contextual explanations. For the Mongolian SOV structure, action targets usually appear before the predicate, which provides linguistic

motivation for focusing on the middle or middle-to-late region of the sentence. However, in real generated texts, the position of the risk object may still be affected by modifier length, clause structure, topicalization, cross-sentence expression, and generation style. Therefore, this paper only sets the central 20% tokens as a compact candidate semantic region, rather than claiming that this region necessarily covers all risk objects.

To examine whether the proposed middle window captures object-related semantic information in actual Mongolian outputs, this paper conducts a manual analysis on 300 stratified text chunks, including 100 long-text unsafe, 80 long-text safe, 80 short-text unsafe, and 40 short-text safe samples. For each chunk, the central 20% token region is examined to determine whether it covers the main object, affected entity, action target, sensitive entity, or other semantic unit relevant to risk judgment.

Two reviewers with Mongolian reading ability independently assign each sample to one of three categories: full coverage, partial coverage, or no coverage. Full coverage indicates that the complete object-related semantic unit is contained in the middle window, while partial coverage indicates that only its core component is covered. Samples without relevant semantic information in the middle window are classified as no coverage. Disagreements are resolved through joint review. The independent annotations achieve a Cohen's κ of 0.84, indicating high inter-annotator agreement. Informed consent was obtained from all the experts/reviewers.

Table 2 Manual Analysis of the Middle-window Prior

Sample Type	Number of Samples	Full Coverage	Partial Coverage	No Coverage	Coverage Rate
Long-text unsafe	100	76	18	6	94.0%
Long-text safe	80	54	20	6	92.5%
Short-text unsafe	80	58	17	5	93.8%
Short-text safe	40	30	6	4	90.0%
Total	300	218	61	21	93.0%

Note: Coverage rate is calculated by treating both full coverage and partial coverage as effective coverage of object-related or risk-related semantic units.

As shown in Table 2, the middle window provides full or partial coverage of object-related semantic units in 93.0% of the sampled chunks. Similar coverage rates are observed across both safe and unsafe subsets, indicating that middle-window coverage itself is not a discriminative safety rule. Instead, the result supports the use of the middle window as an auxiliary representation mechanism for strengthening object-related semantics. The final safety decision therefore remains dependent on the joint modeling of global semantics, middle semantic representations, chunk-level risk, and document-level risk fusion.

Mongolian long-text segmentation and local risk modeling

To reduce truncation while preserving sentence-level semantics, each Mongolian model output is first normalized by removing zero-width characters and byte order marks, merging consecutive whitespace, and replacing empty normalized text with `[EMPTY]`. Sentence boundaries are identified using common Mongolian and multilingual sentence-ending punctuation. This procedure relies only on punctuation and length information and does not use risk keywords, category labels, or manually defined safety rules.

Each segmented sentence is treated as an independent chunk. Because BERT accepts at most 512 tokens and reserves positions for [CLS] and [SEP], the effective content limit is 510 tokens. Sentences within this limit are encoded directly. If a sentence exceeds 510 tokens, token-level sliding windows of 510 tokens with an overlap of 64 tokens are used, giving a step size of $\Delta = 446$. For an overlength token sequence $T = [t_1, \ldots, t_L]$, the k-th window is

$$w_k = [t_{1+k\Delta}, \ldots, t_{\min(510+k\Delta, L)}], \tag{5}$$

for $k = 0, 1, \ldots$ while $1 + k\Delta \leq L$. All chunks are encoded with `max_length=512`, `padding="max_length"`, and `truncation=True`. During batching, chunks are flattened for encoder execution while document assignments and chunk counts are retained for document-level fusion.

The chunk-level local risk module consists of bert-base-mongolian-cased, the dual global/middle representation described above, a SwiGLU gated transformation, and a local risk classifier. The model structure is shown in Fig. 3.

This paper introduces a SwiGLU gated learning mechanism after the bert-base-mongolian-cased encoder mainly based on the following considerations. First, SwiGLU is a variant of the GLU gating structure. It implements feature selection through the element-wise product of two linear transformations, one of which passes through a Swish nonlinear transformation, thereby enabling adaptive filtering of different semantic dimensions in the input representation. Existing studies have shown that GLU and its variants can achieve better modeling performance in Transformer feed-forward networks than traditional ReLU or GELU activation functions. Second, in recent years, multiple large language models have adopted SwiGLU as the nonlinear activation structure in feed-forward networks, indicating that this gated activation method has good representational capacity and applicability in large-scale language modeling tasks. Since the detection target of this paper is Mongolian output text generated by large language models, such text often has characteristics of generative models, including long-context dependency, semantic buildup, and covert expression of risk content. Therefore, introducing SwiGLU after the output representation of bert-base-mongolian-cased helps the model further filter key semantic dimensions related to risk judgment and enhance its ability to represent covert risk features in large language model outputs. It should be noted that this paper does not claim that using the same activation function can directly simulate the generation mechanism of large language models. Instead, it uses the gating property of SwiGLU to improve the detection

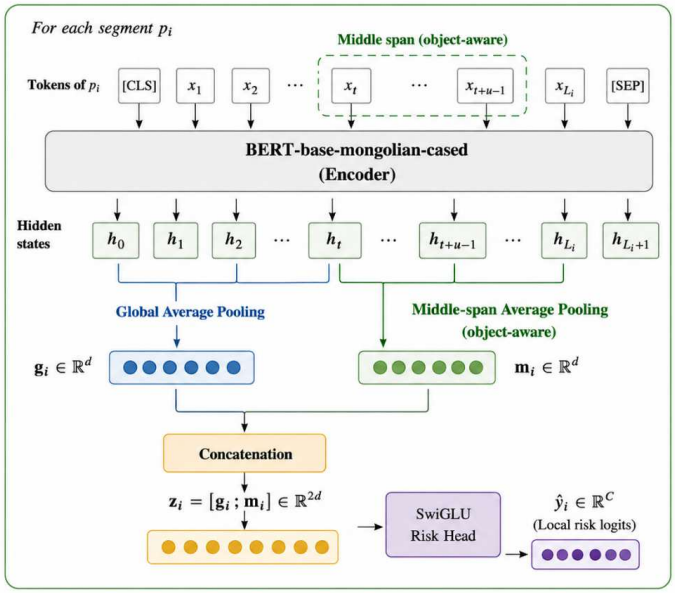


Fig. 3 Chunk-level model structure. Each Mongolian text chunk is encoded by bert-base-mongolian-cased, enhanced through the SwiGLU module, and passed to the local risk classifier to estimate chunk-level risk.

model's adaptability to complex semantic patterns in generated text, thereby providing more effective chunk representations for subsequent local risk judgment and document-level risk fusion.

For the fused representation $\boldsymbol{z}_i$, the SwiGLU gated transformation is defined as

$$\mathrm{SwiGLU}(\boldsymbol{z}_i) = (\boldsymbol{z}_i \boldsymbol{W}_1 + \boldsymbol{b}_1) \odot \mathrm{Swish}(\boldsymbol{z}_i \boldsymbol{W}_2 + \boldsymbol{b}_2), \tag{6}$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, and $\boldsymbol{b}_2$ are learnable parameters. The enhanced representation $\boldsymbol{u}_i$ is mapped to the local unsafe probability

$$r_i = \sigma(\boldsymbol{W}_r \boldsymbol{u}_i + b_r), \tag{7}$$

where $\sigma(\cdot)$ denotes the Sigmoid function. These chunk-level probabilities provide local risk evidence for subsequent document-level fusion.

Document-level risk fusion

Long-text setting

After chunk-level modeling, DocMidFuse-MN aggregates local unsafe probabilities $r_1, \ldots, r_m$ into a document-level score. The mean risk score captures the overall risk

tendency,

$$R_{\text{mean}}(D) = \frac{1}{m} \sum_{i=1}^{m} r_i, \tag{8}$$

while the maximum risk score preserves sensitivity to a locally concentrated high-risk chunk,

$$R_{\max}(D) = \max_{1 \le i \le m} r_i. \tag{9}$$

The final document risk score is obtained by weighted fusion:

$$R(D) = \lambda_{\text{mean}} R_{\text{mean}}(D) + \lambda_{\max} R_{\max}(D), \tag{10}$$

where λ_{mean} and $\lambda_{\max}$ are selected on the validation set. Mean aggregation reflects the overall risk tendency across chunks, whereas max aggregation preserves sensitivity to localized risk. Their combination reduces both local-risk dilution and overreaction to isolated chunk-level fluctuations.

Short-text setting

In addition, this paper does not construct two different model architectures for the short-text and long-text settings. Instead, separate short-text and long-text branches are trained under the unified DocMidFuse-MN framework, with training and fusion parameters matched to their respective data distributions. The two settings share the same chunk-level risk modeling process and document-level risk fusion form. That is, both use sentence-level segmentation, chunk risk prediction, and weighted fusion of mean risk and maximum risk. The difference is that short texts and long texts adopt fusion weights and decision thresholds matched to their text characteristics because they differ in length distribution and risk presentation.

It should be noted that although short texts are relatively short, this paper does not adopt simple whole-text binary classification training. Instead, it still uses the same sentence-level segmentation, chunk risk prediction, and document-level risk fusion training strategy as in the long-text setting. The reason is that short text is not equivalent to a single risk semantic unit. Even when the overall text is short, it may still contain multiple sentences or multiple semantic chunks, and different chunks may contain both normal explanations and risk expressions. If the short text is directly input into the model as a whole for binary classification, the model can only learn one overall label and cannot explicitly distinguish the contribution of different internal chunks to the final risk judgment.

By contrast, using the same document-level fusion training strategy as in the long-text setting allows the short-text model to also learn the relationship between local chunk risk and overall document risk, thereby improving the model's ability to identify local risk, semantic shifts, and risk-wrapped expressions within short texts. At the same time, this consistent training structure ensures that the short-text model and the long-text model have the same risk scoring form during inference. That is, both obtain the final document-level risk score through the fusion of mean chunk risk and maximum chunk risk.

Length-conditioned routing inference

During inference, a deterministic length-conditioned rule selects the short-text or long-text model according to the token count $N(D)$:

$$\mathcal{M}(D) = \begin{cases} \mathcal{M}_{\text{short}}, & N(D) < \tau_{\text{len}}, \\ \mathcal{M}_{\text{long}}, & N(D) \geq \tau_{\text{len}}. \end{cases} \tag{11}$$

The routing threshold τ_{len} is selected exclusively on the validation set from $\{256, 320, 384, 400, 512, 640\}$ by jointly considering Accuracy, Precision, unsafe-class Recall, F1, and MCC, with priority given to avoiding a clear reduction in unsafe Recall. The test set is not used for threshold selection.

After routing, the selected branch follows the previously described chunk-level modeling and document-level fusion procedure. The final prediction is

$$\hat{y} = \begin{cases} \texttt{unsafe}, & R(D) \geq \tau_{\text{doc}}, \\ \texttt{safe}, & R(D) < \tau_{\text{doc}}. \end{cases} \tag{12}$$

Ethics statement

The study did not involve experiments on human subjects or the collection of personal, clinical, biometric, or personally identifiable human data. Human involvement was limited to the manual review and annotation of Mongolian text samples for linguistic and safety assessment. Informed consent was obtained from all the experts/reviewers.

Code availability

The source code and scripts required to reproduce the main computational workflow, model execution, and evaluation procedures reported in this study are provided as Supplementary Information. The supplementary code package contains the publication release version of the code associated with this manuscript. The package includes the core model implementation, preprocessing scripts, configuration templates, inference scripts, evaluation utilities, dependency information, and documentation.

The released code package is intended to support transparency and reproducibility of the main computational workflow. It does not include obsolete development files, temporary debugging scripts, private local paths, unrelated intermediate experiments, or sensitive configuration files. Users should modify the configuration templates according to their own data paths, model paths, and computing environments before running the code.

Experiments

Detailed dataset statistics and split information are provided in Supplementary Table S1. For the long-text experiments, the complete dataset contains 895 Baize-derived unsafe samples and 1030 Databricks-derived safe samples. A total of 1015 samples

form the candidate training pool, while the independent test set contains 910 samples, including 395 Baize-derived unsafe outputs and 515 Databricks-derived safe outputs. The candidate training pool is further divided into training and validation subsets using the fixed 80/20 procedure described in the Supplementary Information. The test set is not involved in training, hyperparameter tuning, threshold selection, or model selection. Hyperparameter settings and random-seed stability results are reported in Supplementary Tables S2 and S3, respectively. Detailed procedures for long-text dataset construction, generation settings, output construction, Mongolianization, manual review, and leakage control are provided in Supplementary Tables S9 to S13.

Comparison experiments

To comprehensively evaluate the effectiveness of DocMidFuse-MN in Mongolian model output safety detection, this paper further conducts comparison experiments from two perspectives: segmentation methods and detection modes. The segmentation method comparison is mainly used to verify the role of sentence-level chunk segmentation in long-text risk detection and to examine whether it can better preserve risk semantics than no-segmentation or simple truncation strategies.

The detection mode comparison is used to compare the performance difference between the proposed method and general-purpose large language model safety detection methods, and to analyze whether a detector specifically designed for Mongolian long-text outputs is more suitable for low-resource language scenarios and long-text risk detection. Through these comparisons, this paper further evaluates the advantages of the proposed method in detection performance, long-text adaptability, and practical deployment efficiency. Complete efficiency and effectiveness statistics, including inference latency, throughput, peak GPU memory usage, average number of chunks, and total inference time, are reported in Supplementary Table S8.

In the comparison experiments, this paper sets up segmentation method baselines and detection model baselines according to different experimental objectives. For the segmentation method comparison, this paper selects Hierarchical BERT (HB) and HiPool (HP) as the main baselines for long-text segmentation and modeling. These two methods are selected because HB and HP are representative hierarchical modeling methods for long-document classification tasks. They can mitigate the input-length limitation of long texts through sentence-level representations, chunk aggregation, or hierarchical pooling. Compared with simple no-segmentation methods, HB and HP explicitly model the internal chunk structure of documents, and are therefore suitable as comparison baselines for evaluating the long-text modeling capability of DocMidFuse-MN.

For the detection model comparison, this paper selects Llama-Guard-4-12B, Qwen3Guard, DeepSeek-V4, and GPT-5.5 as general-purpose detection model baselines. Among them, Llama-Guard-4-12B and Qwen3Guard are large language models fine-tuned for safety detection tasks and can directly make safe/unsafe judgments on input texts. Therefore, this paper treats them as strong baselines among detection models.

The reason for selecting Llama-Guard-4-12B and Qwen3Guard as strong baselines is that both models are multilingual safety detection models optimized for content safety, risk identification, or model output moderation, and can directly classify input texts as safe or unsafe. Compared with ordinary general-purpose large language models, these models are closer to the usage mode of practical safety guardrail systems and can represent a relatively strong general-purpose safety detection paradigm. Meanwhile, since both models have certain multilingual understanding ability, they can be used to judge the safety of texts in low-resource languages such as Mongolian. Therefore, they are suitable as strong baselines in the detection mode comparison experiment.

DeepSeek-V4 and GPT-5.5 have strong contextual semantic understanding and complex reasoning capabilities, and can perform zero-shot safety judgment through prompts. They are therefore used to further evaluate the ability of general-purpose large language models to directly detect risky Mongolian outputs. By comparing with the above baseline methods, this paper evaluates the effectiveness of DocMidFuse-MN in Mongolian long-text output safety detection from the perspectives of long-text segmentation strategies and detection model paradigms.

It should be noted that this paper does not take the translate-then-detect strategy as a main comparison baseline. This does not deny its potential usability in engineering systems, but rather because this strategy changes the core research question of this paper. This paper focuses on model output safety detection in Mongolian low-resource language environments. That is, when the detector directly faces Mongolian text, the key question is whether it can identify risky content that is wrapped by normal semantics, dispersed across long texts, or hidden in sentence-internal regions. If Mongolian outputs are first translated into English or Chinese and then judged by large language models or high-resource-language safety detectors, the actual evaluation target becomes a combination of machine translation quality and high-resource-language detection ability, rather than low-resource language safety detection ability itself.

Such a setting would weaken the focus of the research question. One key issue in low-resource language jailbreaks and cross-lingual safety risks is that existing safety models may suffer from insufficient understanding, unstable risk-semantic transfer, and degraded long-text risk detection ability when directly processing low-resource languages. If Mongolian text is translated into English or Chinese before detection, the detection process effectively bypasses the representation challenge of the low-resource language itself, and it becomes difficult to determine whether the model truly has Mongolian safety detection ability. Therefore, translate-then-detect is more suitable as an additional engineering deployment strategy, rather than as a main baseline for evaluating direct Mongolian safety detection ability.

Segmentation methods

To further analyze the effects of different base encoders and long-text modeling strategies on Mongolian long-text safety detection performance, this paper conducts an encoder and long-text modeling comparison experiment. In this experiment, mDeBERTa-v3-base, mmBERT, XLM-R-base, and bert-base-mongolian-cased are

selected as base semantic encoders, and different long-text processing strategies are compared on the same training and test sets.

For long-text processing strategies, this paper compares four methods: Our, HB [18], HP [19], and None [20]. Our denotes the proposed DocMidFuse-MN method. This method adopts sentence-level chunk segmentation to divide Mongolian long texts into multiple semantically relatively complete chunks, and combines middle semantic awareness with document-level mean/max risk fusion to obtain the final detection result.

HB denotes the Hierarchical BERT method. This method first divides a document into an ordered sentence sequence, then uses BERT to encode each sentence, and further applies a sentence-level Transformer to model sentence representations at the document level, finally producing the document-level classification result.

HP denotes the HiPool method. This method first divides the text into several sentence chunks and groups a fixed number of sentences into a segment. It then uses BERT to obtain semantic representations of each segment, and further aggregates segment representations through graph-structure modeling and hierarchical pooling to obtain the document-level classification result.

None denotes the no-segmentation baseline method. This method does not perform sentence-level chunk segmentation, nor does it introduce hierarchical modeling or local semantic enhancement. Instead, it directly feeds the complete text into the encoder for detection. When the text length exceeds the maximum input length of the encoder, the excessive part is truncated.

Longformer, BigBird, and LED are not used as primary baselines because commonly available checkpoints are predominantly pretrained with English-centered vocabularies and corpora. Direct application to Cyrillic Mongolian would therefore confound long-text architecture with language-representation quality. HB and HP can instead share the same Mongolian base encoder as DocMidFuse-MN, enabling a more controlled comparison of long-text processing strategies. Constructing strictly comparable Mongolian Longformer, BigBird, or LED models would require additional pretraining and introduce new experimental variables beyond the scope of this study.

The detailed comparison results across encoders and long-text modeling strategies are shown in Table 3.

Detection methods

To further verify the effectiveness of the proposed method compared with general-purpose large language model-based detection methods, this paper conducts a detection mode comparison experiment. A mainstream approach for model output safety detection is to directly use the contextual understanding and safety judgment capabilities of large language models to classify the target text as safe or unsafe. This type of method does not require retraining a classifier on a specific dataset. Instead, it guides large language models through prompts to understand the text content and determine whether it contains unsafe information such as illegal or non-compliant content, malicious guidance, hate, violence, privacy leakage, or fraud.

Table 3 Performance Comparison across Encoders and Long-text Modeling Strategies

Strategy	Accuracy	Precision	Recall	F1	PR-AUC	ROC-AUC	MCC
Panel A. mDeBERTa-v3-base							
Our	0.7943	0.6799	0.9949	0.8078	**0.9706**	**0.9822**	0.6542
HB	0.8240	0.7117	**1.0000**	0.8316	0.9160	0.9646	0.7001
HP	**0.9769**	**0.9495**	**1.0000**	**0.9741**	0.9868	**0.9936**	**0.9543**
None	0.9637	0.9229	**1.0000**	0.9599	0.9777	0.9889	0.9293
Panel B. mmBERT							
Our	0.7448	0.6347	**0.9722**	0.7680	0.9213	0.9363	0.5697
HB	**0.9615**	0.9688	0.9418	**0.9551**	**0.9753**	**0.9889**	**0.9271**
HP	0.8999	**0.9691**	0.7949	0.8734	0.9598	0.9680	0.8026
None	0.8405	0.8511	0.7671	0.8069	0.9055	0.9228	0.6743
Panel C. XLM-R-base							
Our	**0.9593**	**0.9475**	0.9595	**0.9535**	**0.9844**	0.9888	**0.9174**
HB	0.9461	0.8896	**1.0000**	0.9416	0.9743	0.9897	0.8971
HP	0.8933	0.9257	0.8203	0.8698	0.9565	0.9563	0.7841
None	0.9516	0.9016	0.9975	0.9471	0.9754	**0.9902**	0.9066
Panel D. bert-base-mongolian-cased							
Our	**0.9747**	0.9745	0.9671	**0.9708**	**0.9908**	**0.9947**	**0.9485**
HB	0.9582	0.9496	0.9544	0.9520	0.9844	0.9902	0.9105
HP	0.9362	**0.9884**	0.8633	0.9216	0.9854	0.9927	0.8739
None	0.5171	0.4520	0.5241	0.4853	0.4796	0.5443	0.0354

Note: The table reports the full metric results across different base encoders and long-text modeling strategies on the 910-sample long-text test set. The best value in each encoder panel is highlighted in bold.

Therefore, this paper selects Llama-Guard-4-12B [21], Qwen3Guard-Gen-8B [22], DeepSeek-V4-pro [23], and GPT-5.5 [24] as representative general-purpose large language model-based detection methods. These models are selected for the following reasons. On the one hand, Llama-Guard-4-12B and Qwen3Guard-Gen-8B are representative safety detection models that can directly identify risks in input texts. On the other hand, DeepSeek-V4-pro and GPT-5.5 have strong contextual semantic understanding and complex reasoning capabilities, and are commonly used for zero-shot or few-shot text judgment tasks. Since Mongolian is a low-resource language, many general-purpose large language models may have limited understanding of Mongolian texts and may fail to accurately identify implicit risk semantics. Therefore, this paper selects the above models, which have strong reasoning ability and certain multilingual understanding ability, as comparison methods to evaluate the effectiveness of the general-purpose large language model-based detection paradigm in Mongolian model output safety detection.

In this experiment, the short-text setting is used to examine the ability of different models to identify locally concentrated risk expressions, while the long-text setting is used to examine detection ability when risk information is dispersed, wrapped by normal semantics, or hidden in the middle of the text. For large language model outputs that contain additional explanations or cannot be directly parsed into binary labels, this paper normalizes them into safe or unsafe according to predefined rules. If the output cannot be determined, it is treated as a detection failure and counted into the corresponding error category. The complete API/LLM baseline evaluation

protocol, including model checkpoints or API identifiers, decoding parameters, input prompts, label mapping rules, output parsing rules, and retry strategy, is provided in Supplementary Table S14.

Table 4 Comparison with General-purpose LLM-based Safety Detectors

Detection Method	Accuracy	Precision	Recall	F1	MCC
Panel A. Short-text test set					
Llama-Guard-4-12B	0.6325	0.6788	0.7217	0.6996	0.2283
Qwen3Guard-Gen-8B	0.7043	0.7281	0.8000	0.7623	0.3762
DeepSeek-V4-pro	0.7110	**0.8879**	0.5867	0.7066	0.4818
GPT-5.5	0.5768	0.7247	0.4614	0.5638	0.2090
Our	**0.7800**	0.8183	**0.8084**	**0.8133**	**0.5456**
Panel B. Long-text test set ($n = 910$)					
Llama-Guard-4-12B	0.6315	0.6705	0.2987	0.4133	0.2332
Qwen3Guard-Gen-8B	0.7294	0.7123	0.6329	0.6702	0.4443
DeepSeek-V4-pro	0.6964	0.9281	0.3266	0.4831	0.4230
GPT-5.5	0.6341	0.9697	0.1620	0.2777	0.2879
Our	**0.9747**	**0.9745**	**0.9671**	**0.9708**	**0.9485**

Note: The table compares general-purpose LLM-based safety detectors with the proposed method on the short-text test set and the 910-sample long-text test set. The best value in each panel is highlighted in bold.

As shown in Table 4, DocMidFuse-MN achieves the strongest overall performance in both settings. On the short-text test set, it achieves an Accuracy of 0.7800, a Precision of 0.8183, a Recall of 0.8084, an F1 of 0.8133, and an MCC of 0.5456. Although DeepSeek-V4-pro obtains a higher Precision of 0.8879, its Recall is substantially lower at 0.5867; the proposed method therefore provides the best F1 and MCC among the evaluated detectors in the short-text setting. The performance gap becomes substantially larger on long texts: the general-purpose detectors show marked recall degradation, whereas DocMidFuse-MN maintains an F1 of 0.9708 and an MCC of 0.9485. This pattern indicates that direct LLM-based safety judgment is less stable when Mongolian risk information is dispersed or wrapped by normal semantics, while sentence-level segmentation, middle semantic awareness, and document-level fusion provide more reliable long-text risk detection.

Ablation study

To examine whether the performance gains of DocMidFuse-MN arise from its individual design components rather than from the overall model capacity alone, we conduct complementary ablation experiments at the architecture, training, feature-transformation, and inference levels. Specifically, we evaluate model architecture variants, separate versus joint training strategies, alternative feed-forward modules, and document-level inference and risk-fusion strategies. The complete results are provided in Supplementary Tables S4 to S7, respectively.

Since middle semantic awareness is one of the central components of DocMidFuse-MN, we further report the position sensitivity experiment in the main text to directly examine how the location of risk information affects detection performance.

Position sensitivity experiment

Table 5 Position Sensitivity Analysis of Risk Detection

Risk Position	Accuracy	Precision	Recall	F1	PR-AUC	ROC-AUC	MCC
Panel A. Short-text test set							
Beginning	**0.7850**	**0.8191**	0.8181	**0.8186**	**0.9046**	**0.8613**	**0.5548**
Middle	0.7800	0.8183	0.8084	0.8133	0.8998	0.8555	0.5456
Ending	0.7700	0.7985	**0.8187**	0.8084	0.8986	0.8537	0.5211
Panel B. Long-text test set ($n = 910$)							
Beginning	0.8214	0.8132	0.7968	0.8049	0.8427	0.8583	0.6185
Middle	**0.9747**	0.9745	**0.9671**	**0.9708**	**0.9908**	**0.9947**	**0.9485**
Ending	0.8768	**0.9764**	0.7342	0.8382	0.9776	0.9836	0.7615

Note: The table reports the full metric results under different risk positions. The best value in each panel is highlighted in bold.

As shown in Table 5, the short-text setting exhibits only limited variation across beginning, middle, and ending risk positions, suggesting that positional effects are relatively weak when risk semantics are concentrated within shorter outputs.

In contrast, position sensitivity becomes more pronounced in the long-text setting. The middle-risk condition achieves the strongest overall performance, with an F1 of 0.9708 and an MCC of 0.9485, whereas performance decreases when risk information is located at the beginning or ending of the text. This pattern is consistent with the motivation for introducing additional middle-region representation in DocMidFuse-MN, particularly when risk-related objects, affected entities, or action targets are embedded within longer Mongolian outputs.

However, these results should be interpreted as evidence of positional sensitivity rather than direct causal evidence that SOV syntax alone determines the location of risk information. The middle semantic mechanism therefore serves as a positional inductive bias that complements global semantic modeling rather than as an explicit syntactic rule.

Conclusion and future work

This paper proposes DocMidFuse-MN for Mongolian model output safety detection, targeting dispersed risk information, semantic wrapping, and middle-hidden risks in long texts. The method combines sentence-level segmentation, an SOV-inspired middle semantic awareness mechanism, and document-level mean/max risk fusion to strengthen local risk representation while preserving overall document-level judgment.

Experiments on both short-text and long-text settings show that DocMidFuse-MN achieves stable detection performance under the dataset and evaluation protocol of this study. Ablation results confirm the contributions of sentence-level segmentation, middle semantic enhancement, and document-level risk fusion, while comparisons with general-purpose large language model-based detectors indicate better long-text robustness and lower inference cost.

Future work will extend the current safe/unsafe binary task to finer-grained risk categories, including illegal guidance, hate and discrimination, violent harm, privacy leakage, fraud inducement, and regulatory evasion. More explicit linguistic modeling, such as dependency analysis and entity recognition, may also improve the localization and interpretation of risk objects and behavioral predicates. Finally, the method will be evaluated on other low-resource languages and multilingual safety detection settings to examine its cross-lingual generalizability and practical deployment potential.

Limitations

The middle semantic awareness mechanism in this paper is mainly based on the subject-object-verb word order characteristics of Mongolian, assuming that the middle region of a sentence is more likely to contain key information such as objects, action targets, or risk targets. This design can improve the model's ability to identify middle-hidden risks, but it is still an approximate position-based modeling method and does not explicitly introduce more fine-grained linguistic tools such as dependency parsing, entity recognition, or semantic role labeling. Therefore, for texts with complex sentence structures, shifted object positions, or cross-sentence risk semantics, the model may still have insufficient recognition ability.

Funding

This paper was supported by Inner Mongolia "YingCaiXingMeng" Talent Team Project (Category II) "Inner Mongolia Cyberspace Security Innovation Talent Team" (Project No. 2025TGL03).

Data availability

The dataset generated and used in this study contains manually reviewed Mongolian model outputs, including potentially harmful and actionable content. To reduce the risk of misuse, the complete raw dataset is not publicly available.

Data necessary to verify the results of this study may be requested from the corresponding author, Gang Wang (wg@imut.edu.cn), for legitimate academic research and research-verification purposes. Requests will be reviewed according to the proposed research purpose and applicable safety requirements. Approved access may be subject to restrictions on redistribution and the misuse of potentially harmful content.

Data format descriptions, label definitions, data construction procedures, and reproduction instructions are provided in the Supplementary Information.

References

[1] Kaplan, J. et al. Scaling laws for neural language models. *arXiv* 2001.08361 (2020).

[2] Brown, T. B. et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* **33**, 1877–1901 (2020).

[3] Ouyang, L. et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems* **35**, 27730–27744 (2022).

[4] Bubeck, S. et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv* 2303.12712 (2023).

[5] Gehman, S., Gururangan, S., Sap, M., Choi, Y. & Smith, N. A. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3356–3369 (2020). doi:10.18653/v1/2020.findings-emnlp.301.

[6] Weidinger, L. et al. Ethical and social risks of harm from language models. *arXiv* 2112.04359 (2021).

[7] Liu, Y. et al. Prompt injection attack against LLM-integrated applications. *arXiv* 2306.05499 (2023).

[8] Yi, S. et al. Jailbreak attacks and defenses against large language models: A survey. *arXiv* 2407.04295 (2024).

[9] OpenAI. Safety best practices. OpenAI API Documentation (2026). Available at: https://developers.openai.com/api/docs/guides/safety-best-practices.

[10] Perspective API. Perspective API documentation (2026). Available at: https://developers.perspectiveapi.com/s/docs.

[11] OpenAI. Moderation. OpenAI API Documentation (2026). Available at: https://developers.openai.com/api/docs/guides/moderation.

[12] Microsoft. Azure AI Content Safety documentation (2026). Available at: https://learn.microsoft.com/en-us/azure/ai-services/content-safety/.

[13] Inan, H. et al. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv* 2312.06674 (2023).

[14] Shen, L. et al. The language barrier: Dissecting safety challenges of LLMs in multilingual contexts. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2668–2680 (2024). doi:10.18653/v1/2024.findings-acl.156.

[15] Jain, D., Kumar, P., Gehman, S., Zhou, X., Hartvigsen, T. & Sap, M. Polyglotoxicityprompts: Multilingual evaluation of neural toxic degeneration in large language models. *arXiv* 2405.09373 (2024).

[16] Zhang, M., Wang, R., Xia, B., Sun, Y. & Zhao, X. MM-Eval: A hierarchical benchmark for modern Mongolian evaluation in LLMs. *arXiv* 2411.09492 (2024).

[17] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, 4171–4186 (2019). doi:10.18653/v1/N19-1423.

[18] Lu, J., Henchion, M., Bacher, I. & Mac Namee, B. A sentence-level hierarchical BERT model for document classification with limited labelled data. *arXiv* 2106.06738 (2021).

[19] Li, I., Feng, A., Radev, D. & Ying, R. HiPool: Modeling long documents using graph neural networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*, 161–171 (2023). doi:10.18653/v1/2023.acl-short.16.

[20] Park, H. H., Vyas, Y. & Shah, K. Efficient classification of long documents using transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*, 702–709 (2022). doi:10.18653/v1/2022.acl-short.79.

[21] Meta. Llama Guard 4 model card: meta-llama/Llama-Guard-4-12B (2025). Available at: https://huggingface.co/meta-llama/Llama-Guard-4-12B.

[22] Zhao, H. et al. Qwen3Guard technical report. *arXiv* 2510.14276 (2025).

[23] DeepSeek-AI. DeepSeek-V4-Pro. Hugging Face model card and technical report (2026). Available at: https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro.

[24] OpenAI. Introducing GPT-5.5 (2026). Available at: https://openai.com/index/introducing-gpt-5-5/.

[25] Tuguldur, E.-O., Gunchinish, S. & Bataa, E. BERT pretrained models on Mongolian datasets. GitHub repository (2019). Available at: https://github.com/tugstugi/mongolian-bert/.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementarymaterial.pdf
- DocMidFuseMNSupplementaryCode.zip