Chunk Attention-based Learning: Obtain Interpretability and Boost Performance in Content moderation

Anonymous ACL submission

Abstract

The advent of ChatGPT and DeepSeek has led to a widespread outbreak of content generation over the internet, amplifying the demand for reliable and scalable moderation solutions to monitor the sheer volume of generated content. Current approaches that rely on Deep Neural Networks(DNNs), fail to meet user expectations in transparency and reliability. Additionally, it is a common issue in which safe content is blocked or harm content is not. Rule-based approaches provide interpretability, but they are limited in scalability and fail to meet the dynamic moderation needs. In this paper, we present CAL (Chunk Attention-based Learning to Obtain Interpretability and Boost Performance in Content Moderation): a novel approach that simultaneously provide interpretability and enhance classification performance in content moderation. Experiments on 8+ golden multilingual datasets show that CAL outperforms traditional state-of-the-art approaches in interpretability and significantly improves the F1 score in text classification. Moreover, it achieves consistent gains across three different backbone models and three distinct taxonomy classification tasks. Furthermore, we validate CAL's practical scalability through seamless integration into a production-scale model, where it achieves millisecond latency while processing 3.5 billion daily requests.

1 Introduction

The launch of ChatGPT has paved the way towards AI-generated content (AIGC) and boosted rapid growth in LLM adoption across industries (Zhao et al., 2024b). OpenAI's annualized revenue has doubled to $3.4 Billion in the six months starting from late 2023 (Ghaffary, 2024). Recent launched model like DeepSeek-R1 achieved 10 million downloads and 1.8 million daily active users within weeks of release (of Apps, 2025). However, LLMs can generate harm content that violates ethical or social norms (Weidinger et al., 2021), and will greatly affect the economic value of AIGC. It is critical and urgent to ensure the safety of content generation (Halaweh, 2023; Deshpande et al., 2023). Similarly, social media platforms, such as RedNote, X (formerly Twitter) have billions of global users posting text and comments, face the same challenge to moderate content at scale while adhering to legal standards, ethical norms and social norms cross different cultures and regions (CNN, 2025; Singhal et al., 2023).

Therefore, major corporations such as Google and Microsoft have provided content moderation as an industry service. In terms of scalability and efficiency, DNNs are widely adopted to meet the increasing demand for complex modern content moderation (Malik et al., 2023). But there are limitations in DNNs. First, given the "black box" nature of DNNs, it's challenging to interpret the decision-making process (Räuker et al., 2023). Considering that Content Moderation is highly sensitive and frequently evolving to accommodate diverse cultures and regions (Singhal et al., 2023), it's especially hard for users to understand why certain content is blocked. Second, DNNs are data driven method and can inherit biases from the training data (Hall et al., 2022). Especially in the field of content moderation it's complex and difficult to collect balanced and unbiased data. One example is annotators can't possibly understand all ethical and social norms of different cultures and regions, producing biased training data (Singhal et al., 2023). Given its sensitive and subjective nature, content moderation is more vulnerable to multiple sources of bias.

Contrasting with DNNs, prior works such as (Clarke et al., 2024; Xin et al., 2024) explored Rule-based approaches such as blocklists and key phrases matching, which inherently expose the interpretable classification logic, but they face scalability limitations when adhering to evolving rules. Recent work in LLMs such as GPT-4, though capable of adapting to evolving rules and addressing

dynamic moderation needs (AlDahoul et al., 2024), lack the specialization for content moderation and may overlook unsafe content. Moreover, their computational overhead leads to prohibitive cost and latency, making LLMs impractical for real-time content moderation. Therefore, it remains an unresolved challenge to provide both interpretability and scalability in real-time content moderation.

To address this challenge, we present CAL (Chunk Attention-based Learning): a novel dual-task framework that unifies classification and interpretability in DNNs. CAL introduces an auxiliary interpretability task alongside different backbone models and extracts determining semantic segments for the classification decision. CAL is model-agnostic and can be plugged into any DNNs. In summary, the contributions of this paper are as follows:

- We propose a dual-task framework to unify classification and interpretability. CAL jointly optimizes interpretability alongside a main classification task, and incurs no additional computational cost or latency cost.
- We demonstrate how CAL can be plugged into backbone models. Using Turing and BERT as backbone models we improve the F1-score by up to 7%, and the gain is consistent across 8 golden multilingual datasets and 3 binary classification tasks.
- We demonstrate that CAL produces reliable interpretable rationales. It outperforms the traditional approach in terms of interpretability, across 8 golden multilingual datasets.
- We conduct an in-depth analysis of its gains in interpretability and classification, and we believe it will benefit future advancements in this area.

2 Related Work

Interpretability Current approaches to interpretability in text classification predominantly fall into three categories: gradient based techniques, quantifying feature contributions, and auxiliary explanatory models. While these approaches provide partial insights into model decisions, they are constrained by critical limitations. **(1) Gradient-based Attribution:** These methods compute the importance of each input feature via partial derivatives of model output with respect to the inputs, thereby identifying keywords that influence model decisions (Sundararajan et al., 2017). Despite their computational efficiency, they have some critical limitations. First, it lacks sensitivity (Zhao et al., 2024a). Second, it often produces discrete keywords leading to noisy and unreadable explanations. **(2) Post Hoc Mechanisms:** Frameworks like SHAP (SHapley Additive explanations) leverage Shapley values from game theory to quantify feature contributions to model output (Lundberg and Lee, 2017). While theoretically principled, they are limited by high computational latency and poor scalability (Chen et al., 2023). **(3) Auxiliary Explanatory Models:** Hybrid architectures that pair classifier model with auxiliary explanatory model (Rudin, 2019) have two issues. First, with two models involved it doubles DevOps cost. Second, it often produce inconsistent result between two models.

Multi-Task Learning Multi-task learning(MTL) enhances the generalization capabilities of DNNs through jointly optimization on multiple tasks (Ruder, 2017). By leveraging shared representations, MTL improves model robustness and generalization, thereby capable of addressing two critical challenges in content moderation: **(1) Interpretability**: By introducing auxiliary tasks MTL incentivizes DNNs to pay attention to semantically significant features across tasks, thereby provides interpretability by extracting text segments or phrases drive both the primary classification task and the auxiliary interpretability task. **(2) Data Bias**: MTL's noise-averaging effect suppresses divergent noise patterns across tasks. Shared representation through joint learning that reduce overfitting to task-specific biases. Prior studies have explored the application of MTL in Natural Language Processing (NLP) (Chen et al., 2024), its application to interpretability - particularly in the context of content moderation - remains underexplored. To the best of our knowledge, our work is the first to explore MTL in interpretable content moderation and we provide a comprehensive study of its efficacy across three DNN architectures (Turing, BERT, Zcode).

3 Framework

In this section, we present CAL framework, outline how CAL operates through two variants: CAL-standard and CAL-advanced.

As depicted in Figure 1, CAL incorporates a

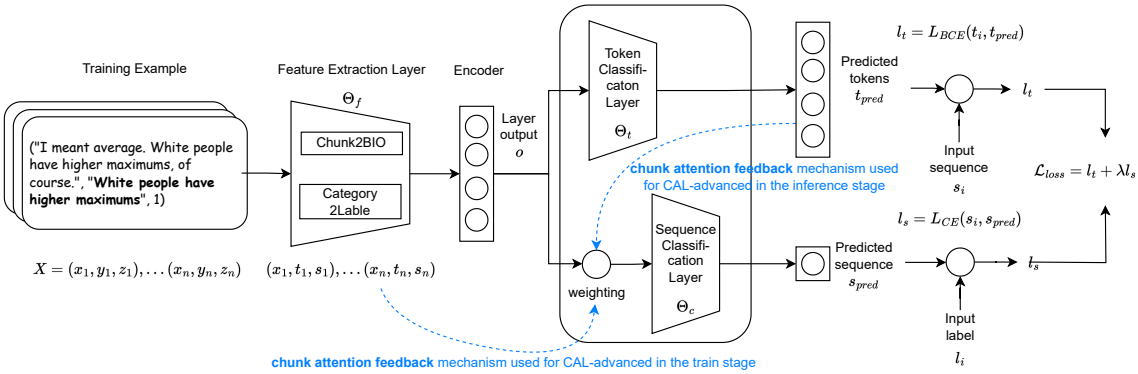

Figure 1: Chunk Attention-based Learning Framework: CAL is comprised of two downstream task, primary classification task and auxiliary interpretability task, which jointly learn semantic embedding representations

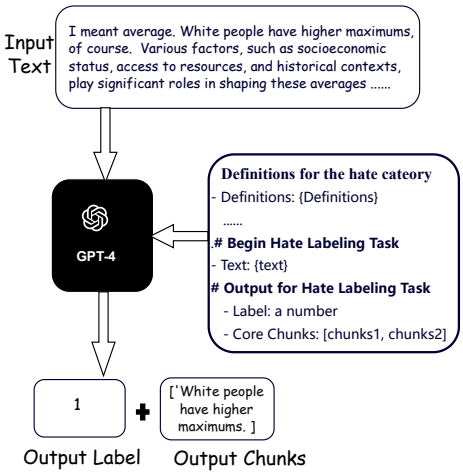

Figure 2: Auto chunk extraction pipeline

dual-task architecture, where two tasks share a backbone model with task-specific heads for joint optimization. **Primary Classification Task**: A sentence-level linear classifier predicts taxonomy labels. **Auxiliary Interpretability Task**: A token-level classifier predicts BIO tags for semantically critical text spans (core chunks).

As illustrated in Figure 2, we utilize GPT-4 to annotate taxonomy labels and core chunks using a single prompt to get the labeled training datasets.

By employing this dual-task architecture, CAL incentivizes model to focus on semantically critical text spans through chunk attention-based learning, thereby enhancing generalization for both classification and interpretability tasks.

3.1 CAL-standard

Next, let's delve into the data flow in the training stage within CAL-standard framework. When an input text, along with its core chunk label and taxonomy label, denoted as (x_i, y_i, z_i), is passed into CAL-standard framework, it undergoes tokenization and initial encoding via the **feature extraction layer**. This layer then converts the core chunk label y_i into a BIO tag format label t_i that model can easily process using the POS algorithm (Church, 1988), and project the taxonomy label z_i to the sequence label s_i. Subsequently, these encoded data are passed onto a shared heavy neural network (i.e. encoder) to generate semantic sentence embedding representations O_i. This sentence embedding is then passed to both a lightweight sentence classifier layer and a lightweight token classifier layer for further forward propagation calculations. Specifically, the lightweight sentence classifier layer is employed to output predicted taxonomy classification s_{pred}. This output s_{pred} is subsequently combined with the sentence label s_i from the training data to compute the Binary Cross-Entropy (BCE) loss over taxonomy labels. Similarly, the linear token classifier layer is designed to produce predicted token probabilities t_{pred}. This output t_{pred} is then juxtaposed with ti, which contains interpretable rationales information, to calculate the Cross-Entropy (CE) loss.

In implementation, a sentence classifier layer comprises several lightweight network structures, including a linear classifier, a dropout module, and a pooling layer. The token classification layer is constructed similarly. The final loss is calculated as the weighted sum of the two losses above.

The training loop is detailed in Algorithm 1.

3.2 CAL-advanced

CAL-advanced extends CAL-standard by introducing **chunk attention feedback** mechanism (see Figure 1, dotted lines). This mechanism dynamically refines sentence embeddings using the

Algorithm 1 CAL-standard Supervised Training

Require: Feature Extraction Layer Θ_f Token Classification Layer Θ_t Sequence Classification Layer Θ_s
Input: training data $X = \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\}$
Output: Updated parameters Θ_f Θ_t Θ_s

Initalize Θ_f Θ_t and Θ_s
while not concerged **do**
  Get mini-batch X_b
  for tokens x_i chunks y_i and category z_i in X_b **do**
    Get tokens label t_i = Chunk2BIO(y_i)
    Get sequence label s_i = Category2Label(z_i)
    Get $t_{pred} = \Theta_t(\Theta_f(x_i))$, $s_{pred} = \Theta_s(\Theta_f(x_i))$
    Compute $\mathcal{L} = L_{BCE}(t_i, t_{pred}) + \lambda L_{CE}(s_i, s_{pred})$
    Update parameters of Θ_f Θ_t and Θ_s
  end for
end while

Algorithm 2 CAL-advanced Supervised Training

Require: Feature Extraction Layer Θ_f Token Classification Layer Θ_t Sequence Classification Layer Θ_s
Input: training data $X = \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\}$
Output: Updated parameters Θ_f Θ_t Θ_s

Initalize Θ_f Θ_t and Θ_s
while not concerged **do**
  Get mini-batch X_b
  for tokens x_i chunks y_i and category z_i in X_b **do**
    Get tokens label t_i = Chunk2BIO(y_i)
    Get sequence label s_i = Category2Label(z_i)
    Get $t_{pred} = \Theta_t(\Theta_f(x_i))$
    Get weighted feature extraction result $r_w = [a_1 ... a_n \mid a_i = \begin{cases} \alpha\, \Theta_f(x_i)_j, & \text{if } t_{ij} \in [B, I] \\ 0, & \text{if } t_{ij} = O \end{cases}]$
    Get $s_{pred} = \Theta_s(r_w + \Theta_f(x_i))$
    Compute $\mathcal{L} = L_{BCE}(t_i, t_{pred}) + \lambda L_{CE}(s_i, s_{pred})$
    Update parameters of Θ_f Θ_t and Θ_s
  end for
end while

saliency of the output core chunks. The following sections detail this mechanism.

1. Core Chunk Identification: Identifies the core chunks by selecting tokens from the input sample that are labeled 'B' or 'I', indicating that these tokens form parts of the core chunks.

2. Saliency-Weighted Embedding Fusion: Retrieves the embedding values corresponding to these tokens and sums them. This sum is then multiplied by a constant factor for weight adjustment.

3. Embedding Adjustment: Combines this weighted sum with the original [CLS] token embedding values, and generate an updated final sentence-level embedding. This updated embedding is then used as the input for the sequence classification layer. In our implementation, we set the constant factor to 0.25.

Note that during the inference stage, BIO labels are unavailable. Instead, CAL-advanced utilizes predicted BIO labels as a substitute for the BIO labels used in the training stage.

Improvements

In the field of content moderation, traditional classifiers often fail to detect harmful content in long text input (e.g., 10k-20k characters), where semantically significant segments constitute only a small portion. This often results in overlooking of harm content.

CAL-standard addresses this by incentivizing model to focus on sematic critical chunks, however its performance degrades as attention weights attenuate in longer texts.

CAL-advanced introduces a chunk attention feedback mechanism that dynamically re-weights critical segments. By tuning attention distribution through iterative feedback, it further improves classification accuracy in long-text scenarios compared to CAL-standard.

4 Experiments Setup

We apply CAL in three backbone models - Turing, Bert, Zcode - to validate its efficacy across diverse DNN architectures (Zcode is presented in Appendix A):

1. Turing (Microsoft, 2022): A state-of-the-art model for multilingual tasks, serving as our high standard baseline.

2. Multilingual-distilBERT (Sanh et al., 2020): An open-source, lightweight model by distilling BERT, chosen to demonstrate CAL's adaptability across model scales.

3. Z-Code++ (He et al., 2023): A state-of-the-art pre-trained language model, selected as our production backbone model because of its exceptional text comprehension capabilities.

4.1 Baseline

We compare the performance of our baseline models **before and after** applying CAL, in terms of both classification and interpretability.

Classification: We use the Turing model (Microsoft, 2022) and Multilingual-distilBERT model

(Sanh et al., 2020) as our baseline models for classification tasks.

Interpretability: We set Turing model as the backbone model, given its demonstrated exceptional performance in text classification tasks. Additionally, we chose the gradient-based attribution (Sundararajan et al., 2017) and post hoc mechanism - SHAP (Lundberg and Lee, 2017) as our baseline for interpretability tasks.

4.2 Training parameters

In our experiment, all models were trained using AdamW optimizer with a weight decay of 0.01 applied to all training data. We utilized a learning rate of 1e-5, a batch size of 32, and set epochs to 20. Our models were trained on NVIDIA Tesla V100 32GB GPUs using Azure Machine Learning Studio. Each category was given its own training dataset comprising ∼100k data sampless. The training of all variations of CAL, as well as the baseline, across all three backbone models and three taxonomies, required an estimated total of 180 GPU hours.

4.3 Test Dataset

Given the scarcity, suboptimal quality, and limited multilingual support in the public dataset for content moderation(Mathew et al., 2021), we choose our proprietary, high-quality **RAI Golden Multilingual Dataset** for evaluation.

Coverage: Tier-1 major languages (English, Japanese, German, Spanish, French, Portuguese, Italian, Chinese), each language forms an independent test set consisting of ∼5k samples balanced across three taxonomies (sexual, hate, violence).

Quality labels: Taxonomy labels and core chunks annotated by GPT-4. We evaluated the consistency between GPT-4 and human labels surpassed 80% across all languages‘s dataset.

4.4 Training Dataset

Data Sources: Per-taxonomy datasets (∼100k samples each) were sampled from the production corpus. The production corpus originated from GPT-4 and public datasets. Many of these datasets, which originally contained taxonomy labels but lacked chunk labels, were relabeled by native speakers. **Dataset Distribution**: For each taxonomy dataset, the percentage of samples labeled as the taxonomy's respective category (violence, hate, or sexual) varies - 18.09% for violence, 16.52% for hate, and 23.07% for sexual. The rest of the samples were not representative of their respective taxonomy. **Average Text Length**: The average length of sample texts varies across taxonomies, with 40 characters for violence, 33 for hate, and 31 for sexual. **Collection Process**: Datasets were sampled from the production corpus and processed through chunk extraction pipeline mentioned before for label acquisition. Datasets were discarded if there was a discrepancy between GPT-4 predicted taxonomy label and the original label.

5 Results and Discussions

In this section, we present and analyze our experiment result from two aspects: text classification and model interpretability (see Appendix A.5 for comprehensive discussion regarding the computational efficiency of our framework).

5.1 Text Classification

Tables 1-3 present taxonomy-specific F1 scores across Tier-1 major languages. In this section, we detail the key findings for CAL-standard and CAL-advanced.

5.1.1 Taxonomy Violence

- **CAL-standard:** Positive gains across languages with +5.07% English, +3.79% French, +3.83% Spanish using **Turing** backbone. Gains persist with +4.12% French, +3.65% Italian using **DistilBERT**.

- **CAL-advanced:** Gains surpass in most languages with +7.28% English, +5.22% French, +4.42 % Italian, +4.59% Portuguese, +4.07% Spanish over using **Turing** backbone. Most languages show degradations (e.g., –6.41% German, –3.75% French) using **DistilBERT**.

5.1.2 Taxonomy Hate

- **CAL-standard:** Improvements are consistent (e.g., +3.62% French, +3.41% German) using **Turing** backbone. Gains remain positive but are modest (ranging roughly from +0.22% to +2.11%) using **DistilBERT**.

- **CAL-advanced:** Gains are slightly lower than CAL-standard (e.g., English +1.47% vs. +1.68% for CAL-standard) and even a slight drop is observed in Japanese (–0.27%) using **Turing**. Significant degradations in most languages (e.g., English –3.93%, French -5.40%, German –4.82%, Spanish –5.93%) using **DistilBERT**.

Model	F1 score on 8 multilingual datasets							
	Chinese	English	French	German	Italian	Japanese	Portuguese	Spanish
Baseline^{+}	0.649	0.608	0.627	0.634	0.604	0.659	0.702	0.580
CAL-standard^{+}	0.655 **0.59%**	0.659 **5.07%**	0.665 **3.79%**	0.641 **0.66%**	0.614 **0.99%**	0.663 **0.38%**	0.721 **1.94%**	0.618 **3.83%**
CAL-advanced^{+}	0.667 **1.78%**	0.681 **7.28%**	0.679 **5.22%**	0.636 **0.16%**	0.648 **4.42%**	0.674 **1.48%**	0.748 **4.59%**	0.620 **4.07%**
Baseline$^{\triangle}$	0.580	0.522	0.490	0.545	0.423	0.532	0.558	0.466
CAL-standard$^{\triangle}$	0.581 **0.16%**	0.539 **1.77%**	0.532 **4.12%**	0.550 **0.56%**	0.460 **3.65%**	0.538 **0.55%**	0.574 **1.63%**	0.467 **0.07%**
CAL-advanced$^{\triangle}$	0.585 0.51%	0.506 -1.56%	0.453 -3.75%	0.480 -6.41%	0.445 2.17%	0.523 -0.95%	0.548 -0.97%	0.440 -2.59%

Table 1: Taxonomy violence, $^{+}$Uses Turing as backbone model.$^{\triangle}$Uses multilingual-distilBERT as backbone model.

Model	F1 score on 8 multilingual datasets							
	Chinese	English	French	German	Italian	Japanese	Portuguese	Spanish
Baseline^{+}	0.613	0.600	0.650	0.536	0.418	0.641	0.501	0.673
CAL-standard^{+}	0.632 **1.91%**	0.617 **1.68%**	0.686 **3.62%**	0.570 **3.41%**	0.446 **2.75%**	0.641 **0.07%**	0.508 **0.76%**	0.687 **1.39%**
CAL-advanced^{+}	0.619 **0.69%**	0.615 **1.47%**	0.676 **2.56%**	0.551 **1.46%**	0.448 **2.91%**	0.638 -0.27%	0.513 **1.27%**	0.674 **0.13%**
Baseline$^{\triangle}$	0.501	0.534	0.578	0.421	0.295	0.541	0.424	0.586
CAL-standard$^{\triangle}$	0.522 **2.11%**	0.540 **0.60%**	0.584 **0.52%**	0.423 **0.22%**	0.313 **1.77%**	0.541 -0.03%	0.442 **1.72%**	0.601 **1.50%**
CAL-advanced$^{\triangle}$	0.503 **0.19%**	0.495 -3.93%	0.524 -5.40%	0.373 -4.82%	0.299 **0.37%**	0.503 -3.87%	0.372 -5.19%	0.527 -5.93%

Table 2: Taxonomy hate, $^{+}$Uses Turing as backbone model.$^{\triangle}$Uses multilingual-distilBERT as backbone model.

Model	F1 score on 8 multilingual datasets							
	Chinese	English	French	German	Italian	Japanese	Portuguese	Spanish
Baseline^{+}	0.822	0.855	0.824	0.794	0.854	0.716	0.879	0.840
CAL-standard^{+}	0.849 **2.72%**	0.867 **1.21%**	0.831 **0.71%**	0.803 **0.94%**	0.861 **0.70%**	0.758 **4.16%**	0.886 **0.66%**	0.840 -0.01%
CAL-advanced^{+}	0.849 **2.69%**	0.864 **0.88%**	0.824 0.00%	0.805 **1.14%**	0.852 -0.15%	0.764 **4.79%**	0.880 **0.10%**	0.844 **0.38%**
Baseline$^{\triangle}$	0.735	0.734	0.760	0.676	0.775	0.566	0.764	0.718
CAL-standard$^{\triangle}$	0.764 **2.92%**	0.780 **4.60%**	0.775 **1.50%**	0.710 **3.45%**	0.785 **1.06%**	0.607 **4.17%**	0.795 **3.11%**	0.745 **2.74%**
CAL-advanced$^{\triangle}$	0.703 -3.15%	0.706 -2.76%	0.749 -1.16%	0.667 -0.91%	0.739 -3.62%	0.589 **2.37%**	0.762 -0.22%	0.687 -3.04%

Table 3: Taxonomy sexual, $^{+}$Uses Turing as backbone model.$^{\triangle}$Uses multilingual-distilBERT as backbone model.

5.1.3 Taxonomy Sexual

- **CAL-standard:** Gains are consistent with a notable +4.16% Japanese using **Turing** backbone. Consistent gains across all languages with +4.60% using **DistilBERT**.

- **CAL-advanced:** Gains are similar to CAL-standard using **Turing**. Significant degradations in most languages (e.g., Chinese –3.15%, English –2.76%, French –1.16%, Italian –3.62%, Spanish –3.04%)) using **DistilBERT**.

5.1.4 Result Analysis

Below, we outline our observations and findings.

1. CAL-standard consistently boosts F1-score across models and datasets.

- **MTL Framework:** Incentivizes the model

Method	8 multilingual datasets	segment-level	token-level	char-level	Variance
		F1	F1	F1	
SHAP	English	0.180	0.165	0.469	0.020
gradient		0.247	0.230	0.540	0.020
original classifier		0.203	0.199	0.443	0.013
CAL		**0.616**	**0.597**	**0.736**	**0.004**
SHAP	German	0.104	0.087	0.254	0.006
gradient		0.209	0.191	0.416	0.010
original classifier		0.088	0.086	0.231	0.005
CAL		**0.472**	**0.457**	**0.574**	**0.003**
SHAP	Japanese	0.128	0.136	0.468	0.025
gradient		0.302	0.293	0.457	0.006
original classifier		0.157	0.133	0.293	0.005
CAL		**0.553**	**0.558**	**0.631**	**0.001**
SHAP	Spanish	0.164	0.153	0.392	0.012
gradient		0.286	0.256	0.539	0.016
original classifier		0.130	0.120	0.356	0.012
CAL		**0.533**	**0.511**	**0.652**	**0.004**
SHAP	Chinese	0.300	0.289	0.586	0.019
gradient		0.335	0.322	0.473	0.005
original classifier		0.188	0.167	0.292	0.003
CAL		**0.588**	**0.596**	**0.622**	**0.000**
SHAP	Italian	0.159	0.156	0.304	0.005
gradient		0.259	0.231	0.448	0.009
original classifier		0.098	0.089	0.260	0.006
CAL		**0.514**	**0.503**	**0.601**	**0.002**
SHAP	French	0.122	0.109	0.293	0.007
gradient		0.233	0.211	0.459	0.013
original classifier		0.130	0.124	0.295	0.006
CAL		**0.531**	**0.522**	**0.637**	**0.003**
SHAP	Portuguese	0.091	0.080	0.255	0.006
gradient		0.298	0.279	0.524	0.012
original classifier		0.102	0.094	0.289	0.008
CAL		**0.571**	**0.564**	**0.665**	**0.002**

Table 4: Comparison of Chunk Extraction Quality Across Different Levels.

to focus on semantically decisive features for both classification and interpretability. Additionally, different tasks have different noise patterns, through joint learning it suppresses divergent noise patterns across tasks, thereby reducing overfitting to task-specific biases. These benefits contribute to better generalization and enhanced performance for both tasks.

Additionally, the successful application of MTL depends on two key prerequisites: the tasks must be closely related to each other, and the training datasets for both tasks must be of high quality with reliable labels. CAL framework meets these criteria by employing **Chunk Attention-based Learning** to closely associate the tasks and using **GPT annotation** to generate reliable labels for auxiliary task.

- **Chunk Attention-based Learning:** CAL applying Chunk Attention-based Learning to tightly connect the two tasks. Within CAL framework, the extraction of core chunks that align with a specific taxonomy provides compelling evidence for the relevance; conversely, the absence of such core chunks indicates a weak or non-existent relevance. This mutual reinforcement — where attention-based chunk extraction reinforces classification decisions and, in turn, classification outcomes guide the focus of the attention mechanism — ultimately enhances feature representation and overall classification performance.

- **Quality Annotation:** Without high-quality annotations, CAL framework would struggle to maintain data consistency between two tasks, reducing the effectiveness of **Dual-Task Framework** and **Chunk Attention-based Learning**, and ultimately affect the overall classification performance. By leveraging GPT-4 to generate high-quality annotations, it ensures the mutual reinforcement between classification and chunk extraction.

2. CAL-advanced shows Inconsistent Performance.

- **Performance Boost in Long-Text:** We observe that it's mainly consisting of long texts in violence taxonomy, where semantically significant segments may be obscured in long texts, and chunk attention feedback's additive saliency forces attention on these segments and improve the performance.

- **Performance Variance across Models:** We observe overall gains over Turing, but overall degradation over DistillBERT. It seems Turing's 17B parameters absorb feedback noise, while DistillBERT's 66M parameters may overfit to chunk re-weighting.

- **Hard Mode Limitation:** We apply chunk attention feedback in hard mode, re-weighting the embedding information of tokens corresponding to chunk labels by a fixed constant. It doesn't generalize well across models and data, we will evaluate adaptive weighting as future work.

5.2 Interpretability

For focused analysis, we report results on the *violence* taxonomy across eight languages in the RAI Golden Multilingual Dataset, omitting CAL-advanced due to space constraints.

5.2.1 Evaluation

CAL generates interpretable rationales by exacting semantically critical text spans (chunks). We evaluate its performance by measuring the alignment between predicted chunks and golden chunks, similar to the evaluation process of sequence tagging. However traditional evaluation metrics used

in sequence tagging (e.g., exact match) are limited to manage the fuzzy boundaries and contextual continuity of chunks.

To address these limitations, we propose a noval evaluation mechanism that captures both the **accuracy** and **completeness** of chunk extraction. Our proposed mechanism employs a multi-granular analysis approach to ensure a more comprehensive evaluation process.

5.2.2 Accuracy Assessment

We assess the accuracy of core chunks' at three different granularities: character, token, and segment (parts of a sentence, separated by punctuation marks). At each of these levels, we use the following method to measure accuracy. Let's explore this using the token-level as an example:

Given the i-th test sample as input, $N_{P,i}$ represents the number of predicted core chunks segmented at the token level, $N_{T,i}$ represents the number of target core chunks segmented at the token level, and $N_{T \cap P,i}$ stands for the number of segments that overlap between the predicted and target core chunks, both segmented at the token level.

$$p_i = \frac{N_{T \cap P,i}}{N_{T,i}} \quad (1) \qquad r_i = \frac{N_{T \cap P,i}}{N_{P,i}} \quad (2)$$

Using Equation (1), we first calculate the proportion of segments that overlap between the target and predicted segments, represented as p_i. Similarly, Equation (2) allows us to find the proportion of segments that overlap between the target and the predicted segments, denotes as r_i.

$$P = \frac{1}{n}\sum_{i=1}^{n} p_i \quad (3) \quad R = \frac{1}{n}\sum_{i=1}^{n} r_i \quad (4) \quad F1 = 2 \times \frac{P \times R}{P + R} \quad (5)$$

Upon evaluating all test samples, we compute the arithmetic mean of all p_i and r_i values to produce the overall indicators, P and R, which quantify the relationship between the predicted and target core chunks in all samples. Lastly, the harmonic mean of P and R, referred to as the F1 score, serves as the final measure of accuracy in the relationship between the predicted and the target core chunks, as illustrated in Equations (3-5).

This hierarchy addresses the fuzzy boundary problem: a near-hit chunk missing one character (e.g. "I hate you" vs. "I hate yo") would retain high character-level F1 but low token-level F1, reflecting partial correctness while penalizing fragmentation.

5.2.3 Completeness Quantification

As demonstrated in Equation (6), we employ the concept of **variance** as a metric to quantify the completeness of the core chunks, where $F1_{char}$, $F1_{token}$, $F1_{segment}$ represent the F1 scores at the "character, token, segment" levels respectively.

$$variance = \frac{1}{3}\sum_{i=1}^{3}(X_i - \overline{X})^2, \quad \overline{X} = \frac{X_1 + X_2 + X_3}{3} \quad (6)$$

The variance is computed from these F1 scores, providing a quantitative measure of the completeness of the core chunks.

- High variance indicates fragmented pieces (e.g. core chunks = ["I", "hate", "you"]).
- Low variance indicates contiguous spans (e.g. core chunks = ["I hate you"]).).

This metric penalizes systems that sporadically highlight individual tokens (common in gradient-based attribution) while rewarding holistic reasoning (e.g. contextual continuity of semantically significant chunks).

5.2.4 Result Analysis

As shown in Table 4, CAL-standard outperforms all the three baselines in F1 scores across levels and languages. Specifically its performance generally improves from character-level to segment-level, showing that CAL are better to capture larger, semantic segments. Additionally CAL achieves the most balanced performance across levels - the highest varience score, demonstrating its capability of capturing contextual continuity of semantically significant chunks.

6 Conclusion

In this paper, we present CAL, a novel Dual-Task Learning framework that enhances text classification with interpretability in content moderation without introducing computational overhead. Through jointly optimizing an auxiliary interpretability task CAL achieves two critical advancements. **(1) Performance Gains:** Experiments result on 8+ golden multilingual datasets demonstrate that CAL improves classification F1 by up to 7% over single-task baselines. **(2) Interpretability:** Extract interpretable rationales via chunk attention, significantly outperforming gradient-based attribution and SHAP across languages.

Deployed in our production content moderation platform, CAL achieves milliseconds latency while processing 3.5 billion daily requests demonstrating its practicality in serving real-time content moderation. Though evaluated on moderation tasks, CAL's architecture is domain-agnostic, offering a blueprint for interpretable text classification in healthcare and other domains.

7 Limitations

7.1 Dependence on High-Quality Chunk Annotations

The CAL approach relies heavily on core chunks as interpretability task labels, implying that the quality of these annotated chunks by GPT-4 (or any other LLMs) could significantly impact the quality of the subsequent interpretability task. If the extracted core chunk labels are of inferior quality, it could negatively affect the quality of the interpretability task. Furthermore, the shared hidden layer in our multitask setup can create a domino effect. A reduction in the quality of one task can potentially degrade the performance of another task, such as text classification. This could potentially lead not only to the CAL framework's inability to provide reliable interpretable features but also a decrease in the original text classifier's performance. A possible future research could be to examine the impact of the quality of annotated core chunk labels on the performance of the CAL framework.

7.2 Fixed Attention re-weighting in CAL-advanced

In CAL-advanced, we introduce a chunk attention feedback mechanism aimed at enhancing performance on long texts by re-weighting the embeddings corresponding to chunk labels. In the current implementation, this re-weighting is performed in a hard mode by multiplying a fixed constant. However, this fixed approach shows poor generalization and inconsistent performance across different backbone models and datasets.

In future work, we plan to evaluate adaptive weighting strategies that dynamically adjust the re-weighting factor based on the estimated saliency of each chunk, with the goal of achieving more consistent and robust performance across a wider range of models and datasets.

7.3 Dependence on DNN-based backbone model

Our CAL approach can be applied to any backbone model, but it necessitates a DNN-based backbone model. It remains uncertain if the backbone model is not a DNN-based model, the multitask framework that CAL depends upon may not function optimally, as the employment of two lightweight classifiers sharing a heavy-duty network to complete various downstream tasks might not be feasible. Further researches may be needed to directly validate the corresponding outcomes for backbone models that are not DNN-based backbone model.

8 Ethics

The data used in this work is a proprietary asset of the company. The collection process adheres to established ethical guidelines and regulatory standards, and fully complies with the strict compliance requirements set by our company. To the best of our knowledge, there are no ethical risks associated with the data or its use in this work.

References

Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024. Advancing content moderation: Evaluating large language models for detecting sensitive content across text, images, and videos. *Preprint*, arXiv:2411.17123.

Hugh Chen, Ian C Covert, Scott M Lundberg, and Su-In Lee. 2023. Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, 5(6):590–601.

Shijie Chen, Yu Zhang, and Qiang Yang. 2024. Multi-task learning in natural language processing: An overview. *Preprint*, arXiv:2109.09138.

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA. Association for Computational Linguistics.

Christopher Clarke et al. 2024. Rule by example: Harnessing logical rules for explainable hate speech detection. *ArXiv*, abs/2307.12935.

CNN. 2025. As us tiktok users move to rednote, some are encountering chinese-style censorship for the first time. *CNN*.

Wenliang Dai, Tiezheng Yu, Zihan Liu, and Pascale Fung. 2020. Kungfupanda at semeval-2020 task 12: Bert-based multi-task learning for offensive language detection. *Preprint*, arXiv:2004.13432.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1236–1270.

Shirin Ghaffary. 2024. Openai doubles annualized revenue to $3.4billion$. *Bloomberg*.

Mohanad Halaweh. 2023. Chatgpt in education: Strategies for responsible implementation. *Contemporary Educational Technology*.

Melissa Hall, Laurens van der Maaten, Laura Gustafson, Maxwell Jones, and Aaron Adcock. 2022. A systematic study of bias amplification. *arXiv preprint arXiv:2201.11706*.

Pengcheng He, Baolin Peng, Liyang Lu, Song Wang, Jie Mei, Yang Liu, Ruochen Xu, Hany Hassan Awadalla, Yu Shi, Chenguang Zhu, Wayne Xiong, Michael Zeng, Jianfeng Gao, and Xuedong Huang. 2023. Z-code++: A pre-trained language model optimized for abstractive summarization. *ArXiv*.

Vivian Lai, Samuel Carton, Rajat Bhatnagar, Q Vera Liao, Yunfeng Zhang, and Chenhao Tan. 2022. Human-ai collaboration via conditional delegation: A case study of content moderation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from large language models make small reasoners better. *Preprint*, arXiv:2210.06726.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Jitendra Singh Malik, Hezhe Qiao, Guansong Pang, and Anton van den Hengel. 2023. Deep learning for hate speech detection: A comparative study. *Preprint*, arXiv:2202.09517.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14867–14875.

Microsoft. 2022. Microsoft turing universal language representation model, t-ulrv6, tops both xtreme and glue leaderboards with a single model. *Microsoft Bing Blogs*.

Business of Apps. 2025. Deepseek revenue and usage statistics (2025). *Business of Apps*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *ArXiv*, abs/1706.05098.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Preprint*, arXiv:1811.10154.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *Preprint*, arXiv:2207.13243.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Mohit Singhal, Chen Ling, Pujan Paudel, Poojitha Thota, Nihal Kumarswamy, Gianluca Stringhini, and Shirin Nilizadeh. 2023. Sok: Content moderation in social media, from guidelines to enforcement, and research to practice. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroSamp;P)*, page 868–895. IEEE.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *Preprint*, arXiv:2112.04359.

Wangjiaxuan Xin, Kanlun Wang, Zhe Fu, and Lina Zhou. 2024. Let community rules be reflected in online content moderation. *Preprint*, arXiv:2408.12035.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *Preprint*, arXiv:2006.07235.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024a. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024b. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Extension and practical usage

We've utilized the CAL framework in practical online content moderation services. In the following discussion, we provided a comprehensive overview of this online real traffic application, covering the testing data we used, our experimental setup, and the results we achieved.

A.1 Experiments setting

To address a variety of user needs, our online text classifier for is not a mere binary classifier, but a complex tool designed for multi-label, multi-severity, and multi-lingual text classification. The 'multi-label' aspect allows it to simultaneously identify harmful content across different categories like hate speech, sexual violence, and self-harm. The 'multi-severity' aspect ensures we not only categorize the input text but also assess its specific severity level within those categories.

A.1.1 Baseline

Given a consistent training set, we utilize Zcode ++(He et al., 2023) as backbone model to identify four taxonomies, supporting multi-class, multi-severity, and multi-lingual text classification.

A.1.2 Our proposed CAL

Maintaining the same conditions as the baseline, including a consistent backbone model and training dataset, we implement CAL framework to Zcode ++(He et al., 2023) as new approach.

A.2 Dataset

We have used several datasets for our evaluation. some of these datasets are publicly available, while others are our internal dataset, these internal datasets will be made publicly accessible in the future. Each dataset possesses specific taxonomy labels for classification. Using our Auto-chunk extraction module, we've supplemented these with interpretable chunk labels, enhancing their utility for interpretability evaluation.

A.2.1 Internal Eval

This dataset consists of 1996 samples that have been annotated using our internal taxonomy Guidance. It encompasses multiple categories, including hate, sexual, violence, and self-harm.

A.2.2 OpenAI Eval

It is an evaluation set from OpenAI, which comprises 979 samples and includes content across four harm categories.

Taxonomy	Datasets	Baseline	CAL-standard	Delta
		Binary F1		
Violence	Internal Eval	76.4	77.7	+1.3
	OpenAI Eval	57.8	60.3	+2.5
	Average	74.4	74.4	**+1.9**
SelfHarm	Internal Eval	78.7	84.2	+5.5
	Suicide Watch	98.3	98.3	+0.0
	Average	85.8	86.0	**+2.8**
Sexual	Internal Eval	87.1	88.5	+1.4
	OpenAI Eval	91.1	90.9	-0.2
	Average	89.1	90.0	**+0.8**
Hate	Internal Eval	76.1	76.2	+0.1
	ETHOS	76.8	81.5	+4.7
	Average	75.6	74.4	+2.4
		Multiclass F1 (>4)		
Violence	RAI Golden Datasets	55.4	57.0	**+1.6**
SelfHarm		58.5	61.8	**+3.3**
Sexual		74.8	74.8	**0.0**
Hate		40.9	43.3	**+2.4**
		Multiclass F1 (>2)		
Violence	RAI Golden Datasets	70.2	70.7	**+0.5**
SelfHarm		78.2	80.5	**+2.3**
Sexual		87.0	87.8	**+0.5**
Hate		64.1	64.5	**+0.4**

Table 5: Comparison of the performance of online multi-lingual multi-task and multi-level models after applying the REAVEAL method.

A.2.3 ETHOS

ETHOS dataset is from the 2020 paper . Human annotated text from YouTube and Reddit comments. A threshold of 0.5 is applied when converting to a binary label. Note: Many of the examples with label scores below 0.5 are quite hateful and fall within hate category when applying our taxonomy's guidance definition.

A.2.4 SuicideWatch

Text taken from SuicideWatch subreddit and weakly labeled.

A.2.5 RAI Golden Dataset

The RAI Golden Dataset is a benchmark dataset, labeled by native speakers of eight major Tier-1 languages: English, Japanese, German, Spanish, French, Portuguese, Italian, and Chinese. Each language forms an independent test set, divided into four subcategories: sexual, hate, and violence, self-harm. Each subcategory includes approximately 5k samples, labeled according to three dimensions: "is_sexual", "is_hate", and "is_violence", each assigned a severity value ranging from 0 to 7.

Additionally, each dataset sample includes label data for evaluating text interpretability capabilities, with GPT annotations providing the ground truth for these labels. We further validated its quality by manually labelling 100 samples from each language dataset. The consistency between GPT and

Method	Eval datasets	segment-level			token-level			char-level			Variance
		P	R	F1	P	R	F1	P	R	F1	
CAL	English dataset	**0.864**	**0.778**	**0.789**	**0.859**	**0.775**	**0.785**	**0.874**	**0.851**	**0.838**	**0.0006**
CAL	German dataset	**0.654**	**0.638**	**0.623**	**0.655**	**0.633**	**0.620**	**0.693**	**0.701**	**0.679**	**0.0007**
CAL	Japanese dataset	**0.805**	**0.772**	**0.770**	**0.807**	**0.775**	**0.775**	**0.834**	**0.800**	**0.803**	**0.0002**
CAL	Spanish dataset	**0.728**	**0.694**	**0.687**	**0.721**	**0.687**	**0.678**	**0.759**	**0.787**	**0.755**	**0.0012**
CAL	Chinese dataset	**0.825**	**0.766**	**0.773**	**0.819**	**0.759**	**0.766**	**0.825**	**0.784**	**0.804**	**0.0002**
CAL	Italian dataset	**0.736**	**0.719**	**0.706**	**0.731**	**0.715**	**0.700**	**0.757**	**0.776**	**0.749**	**0.0004**
CAL	French dataset	**0.717**	**0.691**	**0.683**	**0.715**	**0.690**	**0.680**	**0.750**	**0.744**	**0.731**	**0.0005**
CAL	Portuguese dataset	**0.783**	**0.719**	**0.742**	**0.780**	**0.742**	**0.739**	**0.811**	**0.818**	**0.800**	**0.0008**

Table 6: Comparison of Chunk Extraction Quality Across Different Levels

a text sample		**Batchsize**	**1000 reqs took seconds**		**Delta(CAL - Baseline)**	**%Delta(CAL - Baseline)**
content type	**Length**		**Baseline**	**CAL**		
safe content	10k chars	1	22.511	22.610	0.000	0.44%
		5	35.810	36.247	0.438	1.22%
		10	49.809	51.792	1.984	3.98%
safe and harmful content	10k chars	1	22.868	23.025	0.157	0.68%
		5	36.101	38.138	2.037	5.64%
		10	50.854	53.237	2.383	4.69%
harmful content	10k chars	1	28.437	29.036	0.599	2.11%
		5	41.338	43.796	2.457	5.94%
		10	60.611	62.594	1.984	3.27%

Table 7: Latency Comparison results

human labels surpassed 80% across all languages, demonstrating the reliability of the interpret labels.

A.3 Text classification result

Table 1 reports our classification evaluation results for our online model on binary dataset and multi-class datasets.

(1)Result on binary evaluation datasets Our observations on binary evaluation datasets indicate that the CAL-standard method consistently improves the F1-score across all taxonomies. Notably, the method boosts the F1-score by an average of 1.9%, 2.8%, 0.8% and 2.4% for violence, self-harm, sexual, and hate taxonomies, respectively. In some instances, the F1 gain even peaks at 5.5%.

(2) Result on multi-class evaluation datasets: For the multi-class evaluation datasets, we utilized two distinct thresholds to evaluate the model's quality at various severity levels. The multi-class evaluation data shown in the table above is derived from the average F1 values across eight languages. The findings suggest that irrespective of whether the severity level is set to 2 or 4, the CAL-advanced method invariably enhances the model's F1 score.

Note: As our online models are trained on a vast amount of data, the baseline performance is already impressive. We were pleasantly to find that our proposed CAL-standard method can further enhance the model's quality. Moreover, unlike previous experiments, the backbone of the online model is based on Zcode++, a different backbone, once again demonstrating the robustness and practicality of our method.

A.4 Text interpretablity result

Table 6 presents the interpretability evaluation results of our online model after applying the CAL-standard framework. We observe that the utilization of the CAL method results in high F1 scores under three granularity levels for the online model, exceeding 0.7 across all eight multilingual datasets. In many instances, the chunk-level F1 score even surpasses 0.8, indicating high accuracy in chunk extraction.

As discussed before, the final chunk F1 and token F1 values can measure the completeness of our chunk extraction. The minimal difference observed in the last column of Table 2 across all eight multilingual datasets further confirms the high completeness of the chunks.

Note:We have not included a baseline result using the same data to train the initial text classifier because the classifier wasn't trained with interpretability information, leading to low F1 scores across all three granularity levels. Additionally, we've omitted results from the shape gradient method as a comparison baseline due to the time-consuming nature of these methods, particularly

the gradient method. We haven't yet gathered all evaluation outcomes for it, though we have previously tested both methods on smaller data sets.

A.5 Latency Comparison results

As depicted in Table 7, we conducted extensive latency tests for the CAL and baseline models using different test samples. The results are as follows:

- For "safe content", the CAL model exhibits a slight latency increase of 0.44%, 1.22%, and 3.98% for batch sizes of 1, 5, and 10, respectively.

- For "safe and harmful content", the CAL model consistently shows a negligible increase in latency compared to the baseline model across all batch sizes. It shows an increase of 0.68%, 5.64%, and 4.69% for batch sizes of 1, 5, and 10, respectively.

- For "harmful content", the CAL model consistently shows a slight increase in latency compared to the baseline model, with improvements of 2.11%, 5.94%, and 3.27% for batch sizes of 1, 5, and 10, respectively.

From these results, it's clear that regardless of the content type or the batch size, the latency increase with the CAL model compared to the baseline model is minimal and can be disregarded. This minor increase is due to the time needed to process the BIO tags predicted by the token classifier into interpretable core chunks as features.

Furthermore, we observed that the latency is lower for safe content compared to harmful content. Considering that in real-world scenarios, safe content makes up a large proportion, far exceeding 95%, this further indicates that CAL is highly suitable for practical production services. Therefore, the CAL model's efficiency and scalability in processing all types of data sets make it a valuable tool for real-world applications where handling large volumes of data quickly is critical.

In conclusion, although the CAL model has a very small increase in latency compared to the baseline model (less than 5%, almost negligible), it provides better classification results and transparency for users. Overall, the CAL model is an excellent framework for practical scenarios, with almost no compute or latency cost increase compared to baseline.

A.6 Conclusion

Differing from the experiments previously carried out and mentioned in the main text,we applied the CAL framework to a larger scale training dataset and introduced a new backbone model. Even in a non-simplistic classifier scenario, our CAL framework consistently achieved stellar performance in both classification and interpretability. These results powerfully demonstrate the robustness and practicality of our proposed CAL framework, suggesting that our approach could potentially be applied to any classification model based on neural networks.