Memory-Guided Hard Data Augmentation for Multimodal Named Entity Recognition

Xinyu Liu[1], Kai Fu[1], Yinghan Shi[1], Quanyou Chu[1], Ming Du[1], Hongya Wang[1], Xiaojun Meng[2], Jiansheng Wei[2], Yanghua Xiao[3], Bo Xu[1]*

[1]School of Information and Intelligent Science, Donghua University
[2]Huawei Large Model Data Technology Lab [3]Shanghai Key Laboratory of Data Science, College of Computer Science and Artificial Intelligence, Fudan University
{2242850, 2242825, 2242829, 2242865}@mail.dhu.edu.cn
{duming, hywang}@dhu.edu.cn, {xiaojun.meng, weijiansheng}@huawei.com
shawyh@fudan.edu.cn, xubo@dhu.edu.cn

Abstract

Multimodal Named Entity Recognition relies on visual context to resolve textual ambiguities. To mitigate data scarcity, Data Augmentation (DA) has become a standard practice; however, existing methods predominantly adopt a one-size-fits-all and random perturbation paradigm, ignoring the internal state of the target model. In this paper, we first conduct a quantitative analysis, revealing that a significant portion of errors (over 30%) are model-specific, stemming from the unique biases of different architectures. To address this, we propose Memory-Guided Hard Data Augmentation, a framework designed to systematically repair these specific defects. First, we employ K-fold cross-validation to identify model-specific Hard Data. Second, we construct a Memory Tree and utilize Large Language Models (LLMs) with a clustering mechanism to induce macro-level error patterns from micro-level failures. This facilitates a paradigm shift from stateless instance-driven augmentation to a logical pattern-driven approach. Finally, we introduce an iterative augmentation mechanism that triggers recursive generation for stubborn instances that fail initial quality filters. Extensive experiments on Twitter-2015 and Twitter-2017 benchmarks demonstrate that our framework consistently yields significant performance gains across various MNER backbones.

1 Introduction

Named Entity Recognition (NER) is a fundamental task for identifying and classifying entities in unstructured text (Li et al., 2020; Huang et al., 2015; Ma and Hovy, 2016; Lample et al., 2016). However, traditional text-based systems (Devlin et al., 2019) often face semantic ambiguity (e.g., "Apple" as a brand vs. fruit) due to contextual sparsity. To mitigate this, Multimodal NER (MNER) incorporates visual context. Recent advancements focus on sophisticated cross-modal interactions, ranging from word-object alignment in UMT (Yu et al., 2020) and image-text matching in MAF (Xu et al., 2022) and ITA (Wang et al., 2022), to adaptive noise filtering via visual prefixes and mixture-of-experts in HVPNeT (Chen et al., 2022b) and VisualPT-MoE (Xu et al., 2023).

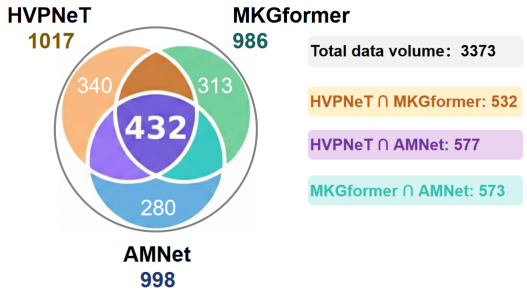


Figure 1: Venn diagram analysis of hard instances identified by HVPNeT, MKGformer, and AMNet. The distinct non-overlapping regions highlight the model-specificity of errors.

To mitigate data scarcity and high annotation costs, Data Augmentation (DA) has become a pivotal strategy for MNER (Wang et al., 2025; Ding et al., 2024). Existing techniques fall into three paradigms: heuristic-based perturbations involving simple unimodal transformations; cross-modal mixing, exemplified by MixGen (Hao et al., 2023) and AMIA (Xu et al., 2025), which construct synthetic training samples via linear interpolation or adaptive mixing of heterogeneous inputs; and generative synthesis, where methods like GMDA (Li et al., 2024) leverage LLMs and Diffusion Models to generate label-aware training samples.

Despite the progress in generative and mixing-based augmentation, existing MNER-DA frameworks still suffer from two fundamental limitations.

- First, current methods are predominantly **model-independent**, applying uniform trans-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 21376–21391
July 2-7, 2026

formations across the entire dataset without considering the specific feedback or architectural biases of the learner (Xu et al., 2025; Hao et al., 2023; Li et al., 2024). This one-size-fits-all paradigm ignores the fact that different models possess distinct blind spots. As illustrated by our Venn diagram analysis of three representative MNER backbones—HVPNeT, MKGformer (Chen et al., 2022a), and AMNet (Dai et al., 2023) (Figure 1)—approximately 30% of the identified Hard Data are unique to each specific architecture. Crucially, the common error intersection constitutes only a small fraction (e.g., less than 43% for HVPNeT) of each model's total failures. This Model-Specificity suggests that general data expansion fails to cover the private blind spots of a given model, necessitating a more targeted, model-aware optimization.

- Second, existing strategies are often **strategy-monolithic** (Hao et al., 2023; Li et al., 2024), relying on single-point random perturbations or global mixing rules that are decoupled from the underlying error logic. These methods treat all challenging instances as a monolithic group, failing to distinguish between diverse failure patterns such as visual occlusion, semantic polysemy, or boundary ambiguity. Without a mechanism to induce fine-grained error patterns and map them to specialized augmentation heuristics, the generated data often lacks the logical relevance required to systematically repair complex model defects.

Addressing the above challenges, this paper proposes a novel framework: **Memory-Guided Hard Data Augmentation**. Central to this framework is the construction of a **Memory Tree**, designed to specifically repair model-specific defects. Specifically, the process begins with Model-Aware Hard Data Mining to identify unique failure instances. These hard instances are dynamically updated into the Memory Tree, where we leverage Large Language Models (LLMs) to abstract micro-level hard instances into macro-level error patterns stored in memory. Guided by this Memory Tree, the framework synthesizes diverse strategies to generate augmented data. Finally, we employ a Data Filtering stage to ensure data quality, coupled with Iterative Augmentation to re-augment stubborn hard instances that fail the filtering stage.

The contributions of this paper can be summarized as follows:

- We shift the focus from universal data augmentation to a Model-Aware approach, customizing augmentation strategies based on the specific error distributions and blind spots of different model architectures.
- We devise a Memory Tree structure. By inducing macro-level error patterns from micro-level hard instances stored in memory, we transform strategy generation from a random perturbation process into a logical, memory-guided repair mechanism.
- Extensive experiments on multiple mainstream MNER models demonstrate that our framework significantly outperforms existing data augmentation techniques. Rather than claiming absolute state-of-the-art performance on the general MNER task, our work establishes a robust methodological advancement specifically for multimodal data augmentation.

2 Related Work

2.1 Multimodal Named Entity Recognition

Existing MNER approaches typically evolve through three paradigms: interaction-based fusion, alignment-driven denoising, and prompt-based learning. Early interaction-based methods focused on deep fusion mechanisms, utilizing co-attention (Zhang et al., 2018; Lu et al., 2018; Sun et al., 2021; Zhang et al., 2021) or unified transformers (Yu et al., 2020) to capture word-object correspondences. To mitigate the noise from irrelevant visual contexts, **alignment-driven** approaches like MAF (Xu et al., 2022) and ITA (Wang et al., 2022; Zhou et al., 2024) were proposed to explicitly bridge the semantic gap through image-text matching. More recently, prompt-based methods have gained prominence; for instance, HVPNeT (Chen et al., 2022b) and VisualPT-MoE (Xu et al., 2023) leverage visual prefixes and mixture-of-experts to adaptively filter visual noise, enabling more robust parameter-efficient learning.

2.2 Data Augmentation for MNER

Data Augmentation (DA) techniques for MNER can be broadly categorized into heuristic perturbations, cross-modal mixing, and generative synthesis. Heuristic-based methods (Cubuk et al., 2019, 2020) apply simple unimodal transformations, such as synonym replacement for text or random cropping for images. Cross-modal mixing strategies,

derived from the foundational Mixup (Zhang et al., 2017) paradigm and represented by MixGen (Hao et al., 2023) and AMIA (Xu et al., 2025), construct synthetic training samples via linear interpolation or adaptive mixing to bridge modal inconsistencies. Generative synthesis approaches, such as the recently proposed GMDA (Li et al., 2024), utilize Large Language Models (LLMs) and Diffusion Models to generate label-aware multimodal instances. However, these methods are predominantly model-independent and strategy-monolithic.

Different from the aforementioned works, we propose the Memory-Guided Hard Data Augmentation framework. Instead of generic expansion, we identify model-specific hard instances and dynamically construct a Memory Tree to store and induce fine-grained error patterns. This mechanism facilitates a Memory-Guided iterative augmentation process that precisely repairs the unique logical defects of different MNER architectures.

3 Overview

3.1 Problem Definition

Formally, let $\mathcal{D}_{train} = \{(T_i, V_i, Y_i)\}_{i=1}^{N}$ denote a multimodal training dataset with N instances. For the i-th sample, $T_i = \{w_1, w_2, \ldots, w_L\}$ represents the input text sequence of length L, and V_i represents the associated visual image. $Y_i = \{y_1, y_2, \ldots, y_L\}$ corresponds to the sequence of entity labels (e.g., in BIO format), where $y_j \in \mathcal{Y}$ is the predefined label set. The goal of MNER is to learn a mapping function $f_\theta : (T, V) \rightarrow Y$ that accurately predicts the entity label sequence.

The task of Data Augmentation for MNER aims to construct a high-quality synthetic dataset $\mathcal{D}_{aug}$ derived from the original source $\mathcal{D}_{train}$ to expand the training distribution. Specifically, given the input training set $\mathcal{D}_{train}$, our objective is to generate an augmented output set $\mathcal{D}_{aug} = \{(T'_j, V'_j, Y'_j)\}_{j=1}^{M}$. Ultimately, the target model f_θ is trained on the combined dataset $\mathcal{D}_{final} = \mathcal{D}_{train} \cup \mathcal{D}_{aug}$ to minimize generalization error and enhance entity extraction performance on unseen data.

3.2 Framework

We propose Memory-Guided Hard Data Augmentation, a framework designed to repair model-specific defects. As illustrated in Figure 2, the workflow operates through a pipeline of five interconnected stages. Initially, Model-Aware Hard Data Mining takes the original dataset $\mathcal{D}_{train}$ and the target model as inputs to predict and output a subset of Hard Data ($\mathcal{D}_{hard}$). Subsequently, Memory Construction analyzes these instances to build a Memory Tree, which maps specific micro-level errors to abstract macro-level patterns stored in memory. Guided by this Memory Tree, Data Generation synthesizes targeted strategies to produce a Raw Augmented Dataset.

To ensure reliability, Data Filtering takes the raw candidates as input and subjects them to strict verification (Label, Text, and Text-Image validation) to filter out noise. The Iterative Augmentation mechanism directly processes stubborn instances that fail initial filtering by re-injecting them into the generation pipeline for secondary optimization. The framework ultimately outputs the high-quality Augmented Dataset ($\mathcal{D}_{aug}$), which serves as the final result of our generation process.

4 Method

This section details the proposed framework. As shown in Figure 2, the framework consists of five stages: Model-Aware Hard Data Mining, Memory Tree Construction, Data Generation, Data Filtering and Iterative Augmentation.

4.1 Stage 1: Model-Aware Hard Data Mining

Unlike random augmentation, we focus on instances where the model exhibits high uncertainty or error. We employ a K-fold cross-validation strategy on $\mathcal{D}_{train}$ to mine hard data. Specifically, we partition $\mathcal{D}_{train}$ into K disjoint subsets $\{\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(K)}\}$. For each fold k, we train a temporary model $f_\theta^{(k)}$ on $\mathcal{D}_{train} \setminus \mathcal{D}^{(k)}$ and perform inference on $\mathcal{D}^{(k)}$. Let $\hat{Y}_i = f_\theta^{(k)}(T_i, V_i)$ be the predicted label sequence for a sample i in the validation fold. We define the Hard Data set $\mathcal{D}_{hard}$ as:

$$\mathcal{D}_{hard} = \{(T_i, V_i, Y_i) \in \mathcal{D}_{train} \mid \hat{Y}_i \neq Y_i\} \quad (1)$$

where $\neq$ denotes any discrepancy between the predicted and ground-truth sequences (i.e., the Error Data branch in Figure 2).

4.2 Stage 2: Memory Tree Construction

To systematically analyze errors, we define an entity extraction function $\Phi(Y) \rightarrow \mathcal{E}$, which maps a label sequence to a set of entity tuples. Each entity $e \in \mathcal{E}$ is represented as $e = (c, s, e_{idx})$, denoting the category, start index, and end index.

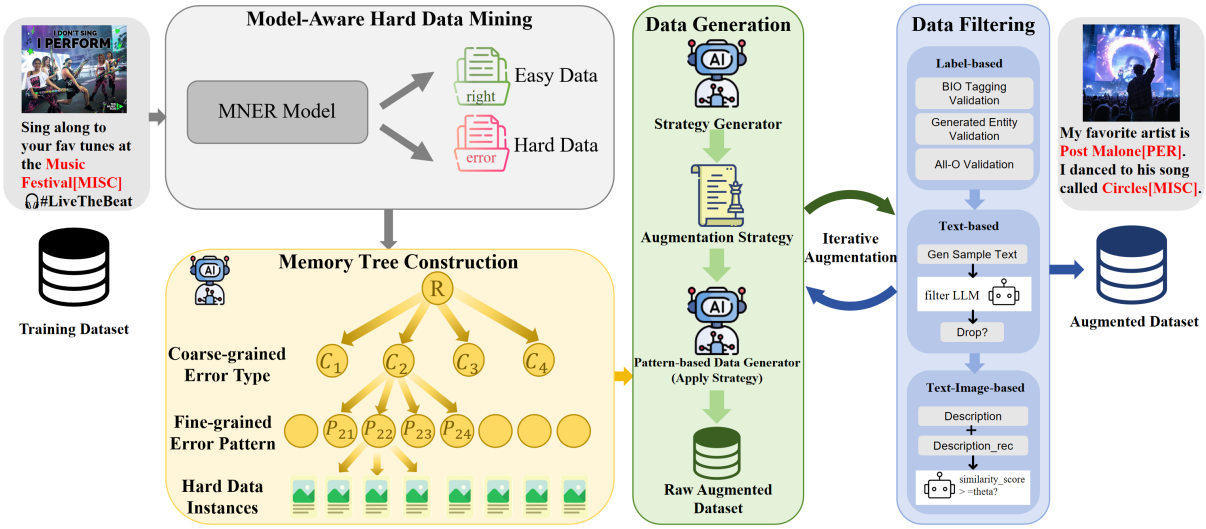


Figure 2: The overall architecture of the Memory-Guided Hard Data Augmentation framework. It consists of five stages: Model-Aware Hard Data Mining, Memory Tree Construction, Data Generation, Data Filtering, and Iterative Augmentation.

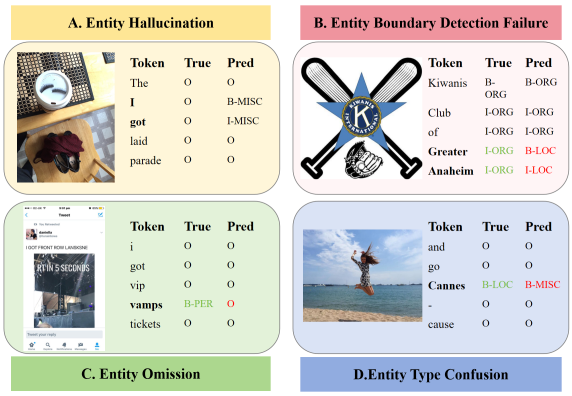


Figure 3: Specific Error Instances under Coarse-grained Classification.

Rule-Based Coarse-Grained Classification. For each hard instance, let the ground truth entity set be $\mathcal{E}_{gt}$ and the predicted set be $\mathcal{E}_{pred}$. As shown in Figure 3, we categorize errors into four explicit types based on the relationship between $\mathcal{E}_{gt}$ and $\mathcal{E}_{pred}$: A. Entity Hallucination, where the model predicts an entity that does not exist in the ground truth ($\exists e \in \mathcal{E}_{pred}, e \notin \mathcal{E}_{gt}$); B. Entity Boundary Detection Failure, where the predicted entity overlaps with the ground truth entity, but the start or end offsets do not match; C. Entity Omission, where an entity exists in the ground truth but is not predicted by the model ($\exists e \in \mathcal{E}_{gt}, e \notin \mathcal{E}_{pred}$); and D. Entity Type Confusion, where the boundaries of the predicted entity perfectly match the ground truth, but it is assigned an incorrect semantic category label.

Fine-Grained Analysis and Memory Tree Construction. Building upon the coarse-grained classification, we employ Large Language Models (LLMs) to conduct Root Cause Analysis. For a hard instance (T, V), we first generate a detailed visual description d_{vis} using an Image-to-Text model. The tuple $(T, d_{vis}, \{\text{Error Type}\})$ is then processed by the LLM to diagnose the root cause. As illustrated in the Memory Tree Construction module of Figure 2, we organize these insights into a Memory Tree: **Level 1 (Root)** is the set of all identified hard instances $\mathcal{D}_{hard}$; **Level 2 (Coarse-grained)** consists of the four error categories defined above; **Level 3 (Fine-grained)** contains compact error patterns derived via clustering; and **Level 4 (Instance)** comprises specific hard data instances mapped to Level 3.

Hierarchical Clustering. To construct the fine-grained level, we propose a two-phase mechanism: Step 1: Local Induction, where we process instances in batches and the LLM summarizes common error patterns $\mathcal{P}_{local}$; and Step 2: Global Merging, where we consolidate $\mathcal{P}_{local}$ into a unified global taxonomy $\mathcal{P}_{global}$ to resolve pattern fragmentation. As detailed in Algorithm 1, we use the LLM as a semantic discriminator to merge synonymous patterns.

4.3 Stage 3: Data Generation

Based on the constructed Memory Tree, we formulate augmentation strategies at the pattern level.

Algorithm 1 LLM-driven Global Pattern Merging

Require: Local set $\mathcal{P}_{local}$, LLM function LLM$(\cdot)$
Ensure: Global set $\mathcal{P}_{global}$

$\mathcal{P}_{global} \leftarrow \emptyset$
for each $p_{new} \in \mathcal{P}_{local}$ **do**
  matched $\leftarrow$ False
  for each $p_{exist} \in \mathcal{P}_{global}$ **do**
    is_same $\leftarrow$ LLM$(p_{new}.d, p_{exist}.d, \tau = 0)$
    if *is_same* **then**
      $p_{exist}.\mathcal{C} \leftarrow p_{exist}.\mathcal{C} \cup p_{new}.\mathcal{C}$ ▷ Merge
      matched $\leftarrow$ True, **break**
    end if
  end for
  if not *matched* **then**
    $\mathcal{P}_{global} \leftarrow \mathcal{P}_{global} \cup \{p_{new}\}$ ▷ Append
  end if
end for
return $\mathcal{P}_{global}$

For each error pattern $p_i \in \mathcal{P}_{global}$, we employ a Strategy Generator to synthesize a generalized augmentation strategy S_i tailored to the specific failure pattern. Subsequently, the Pattern-based Data Generator applies S_i to the instances in cluster $\mathcal{C}_i$, generating raw augmented pairs (T_{aug}, V_{aug}).

4.4 Stage 4: Data Filtering

Since generative models may introduce noise, we design a rigorous Quality Filtering process.

We implement three levels of filtering to screen the generated pairs: (1) Label-Based Filtering: We perform structural validation on the generated label sequence Y_{aug}, including BIO Tagging Validation to ensure standard formatting, Generated Entity Validation to check for undefined categories, and All-O Validation to discard instances containing only "O" tags. (2) Text-Based Filtering: To verify logical consistency, we employ a fine-tuned discriminator (filter LLM) that assesses whether the predicted labels are supported by the text context, outputting a decision to retain or drop the sample. (3) Text-Image-Based Filtering: To ensure semantic consistency, we use an Image-to-Text model to transcribe V_{aug} into a description d_{rec} and calculate the cosine similarity between the original augmentation description and d_{rec}. Only instances with a score $\geq \theta$ are retained.

4.5 Stage 5: Iterative Augmentation

As shown by the loop in Figure 2, if all candidates for a specific instance are filtered out (count = 0), the system triggers the Iterative Augmentation mechanism. Specifically, this mechanism reinvokes the data augmentation strategy corresponding to the instance's specific fine-grained error pattern to regenerate samples, which are subsequently passed through the Data Filtering stage again. This ensures that even stubborn hard instances are effectively augmented.

5 Experiment

In this section, we empirically evaluate the proposed Memory-Guided Hard Data Augmentation framework. The experiments aim to answer three core research questions: (1) Can our framework achieve consistent performance improvements across different MNER architectures compared to existing DA techniques? (2) Does Model-Aware hard data identification contribute more to performance than generic data augmentation? (3) How does the scale of the foundation model (LLM) affect the quality of augmentation and final performance?

5.1 Dataset and Metrics

We conduct experiments on two standard benchmarks for Multimodal Named Entity Recognition (MNER): Twitter-2015 and Twitter-2017. These datasets comprise image-text pairs collected from Twitter, annotated with four entity types: Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). Following standard protocols in the field, we utilize the fixed splits for training, validation, and testing. We adopt the standard BIO tagging scheme and employ Precision (P), Recall (R), and F1-score (F_1) as the primary evaluation metrics.

5.2 Implementation Details

To ensure high-quality data generation and filtering, we utilize the following foundation models and parameter settings, organized according to the five stages of our framework:

- Stage 1: Model-Aware Hard Data Mining. We employ a K-fold cross-validation ($K = 10$) strategy to systematically identify hard instances from the training set.
- Stage 2: Memory Tree Construction. We primarily employ the **Qwen3-235B-A22B**[1] model as the core reasoning engine for complex root cause analysis and high-level strategy induction. To evaluate the impact of model scale, we also conduct comparisons using the lighter **Qwen3-30B-A3B**[2].

[1] https://huggingface.co/Qwen/Qwen3-235B-A22B
[2] https://huggingface.co/Qwen/Qwen3-30B-A3B

- Stage 3: Data Generation. Guided by the formulated strategies, we utilize **Stable-Diffusion-3.5-Large**[3] (Rombach et al., 2022) to execute multimodal synthesis.
- Stage 4: Data Filtering. We enforce strict quality control with a semantic similarity threshold θ set to 0.5. For text-based logical filtering, we fine-tune a **Meta-Llama-3-8B-Instruct**[4] model. For text-image filtering, we employ **Qwen2.5-VL-72B-Instruct**[5] to generate detailed captions for the synthesized images.
- Stage 5: Iterative Augmentation. This mechanism recursively triggers the generation and filtering models for stubborn instances.

To balance computational efficiency and accessibility, we adopted a hybrid deployment strategy. The fine-tuning of the filtering model (Meta-Llama-3-8B-Instruct) and the inference of the image synthesis model (Stable-Diffusion-3.5-Large) were conducted locally on a single NVIDIA H100 GPU. Conversely, the ultra-large-scale foundation models (e.g., Qwen3-235B-A22B, Qwen2.5-VL-72B-Instruct and Qwen3-30B-A3B) were accessed via API endpoints.

5.3 Backbones

To verify the universality of our framework, we apply it to diverse MNER backbones:

HVPNeT (Chen et al., 2022b): A network that treats visual features as pluggable hierarchical prefixes to guide textual representation, utilizing dynamic gating to aggregate multi-scale visual information.

AMNet (Dai et al., 2023): An alignment-focused network that bridges the semantic gap by performing fine-grained matching between visual regions and textual tokens at multiple granularities.

MKGformer (Chen et al., 2022a): A hybrid Transformer architecture employing multi-level fusion. Originally designed for knowledge graphs, we adapt its encoder to MNER to test augmentation effectiveness across different architectures.

5.4 Baselines

We compare our framework with state-of-the-art augmentation methods:

[3] https://huggingface.co/stabilityai/stable-diffusion-3.5-large

[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[5] https://huggingface.co/Qwen/Qwen2.5-VL-72B-Instruct

Table 1: Time cost breakdown on Twitter-2017 (HVPNeT backbone). The process acts as an automated "diagnosis and repair" phase.

Pipeline Stage	Key Operation	Time
1. Hard Data Mining	Model Inference ($K = 10$)	$\sim$ 15 min
2. Memory Construction	LLM Analysis	$\sim$ 20 min
3. Data Generation	**Diffusion Synthesis**	$\sim$ **5 h**
4. Data Filtering	Logic Check	$\sim$ 30 min
Total Offline Cost	**Automated Repair**	$\sim$ **6 h**

Vanilla LLM Augmentation: A simple generative approach based on Large Language Models (LLMs). It randomly selects instances from the training set to generate text variants and employs Stable-Diffusion-3.5 to synthesize corresponding visual contexts.

GMDA (Li et al., 2024): A two-stage generative multimodal data augmentation framework. It first utilizes a label-aware Multimodal Large Language Model (LMLLM) to generate synthetic text sequences with fine-grained entity annotations, and subsequently employs a Stable Diffusion model to synthesize corresponding images conditioned on the generated text and the original image.

AMIA (Xu et al., 2025): An adaptive mixup image augmentation framework. It utilizes Stable-Diffusion to generate candidate images, which are then dynamically blended with the original images to construct synthetic training samples, thereby mitigating modality mismatches.

5.5 Efficiency Analysis and Trade-offs

We address potential concerns regarding computational overhead. As shown in Table 1, the primary time cost ($\sim$5 hours) stems from Stable-Diffusion-3.5 image synthesis. Crucially, since state-of-the-art baselines (e.g., Vanilla LLM Augmentation and AMIA) also rely on Stable Diffusion for fair visual comparison, our method does not incur significantly higher costs. Our framework's unique components (Mining and Memory Tree Construction) add negligible overhead ($\sim$35 mins). Furthermore, this is a one-time, offline process that automates labor-intensive manual error analysis, justifying the investment without affecting final training or inference latency.

Table 2: Performance comparison (%) on Twitter-2015 and Twitter-2017 datasets. The best results are highlighted in **bold**.

Method	Twitter-2015			Twitter-2017		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Backbone: HVPNeT						
Original HVPNeT	74.19	75.35	74.76	85.24	85.94	85.59
+ Vanilla	73.87	76.82	75.32	86.08	86.97	86.52
+ AMIA	**75.41**	75.16	75.28	85.64	86.35	85.99
+ GMDA	74.00	71.84	72.90	85.43	85.94	85.68
+ Ours	75.27	**76.09**	**75.68**	**87.58**	**86.68**	**87.13**
Backbone: AMNet						
Original AMNet	74.28	**76.69**	75.47	84.76	86.45	85.60
+ Vanilla	73.19	76.48	74.80	84.81	**87.69**	86.22
+ AMIA	75.15	75.14	75.15	85.63	86.90	86.26
+ GMDA	73.43	72.45	72.93	85.38	86.90	86.13
+ Ours	**76.48**	74.76	**75.61**	**87.79**	87.27	**87.53**
Backbone: MKGformer						
Original MKGformer	74.50	75.33	74.91	86.89	87.34	87.12
+ Vanilla	73.87	75.56	74.70	85.50	**89.05**	87.24
+ AMIA	74.52	75.50	75.01	**87.40**	88.30	**87.56**
+ GMDA	**74.67**	72.27	73.45	85.40	87.49	86.44
+ Ours	74.07	**76.21**	**75.12**	86.98	88.01	87.49

5.6 Main Results

Table 2 presents the comparison between our framework and state-of-the-art methods on the Twitter-2015 and Twitter-2017 datasets.

From the results, we can observe that our Memory-Guided framework consistently outperforms both the original baselines and other augmentation strategies. Specifically, on the HVPNeT backbone, our method improves the F1-score by 0.92% on Twitter-2015 and 1.54% on Twitter-2017. This demonstrates that identifying and repairing model-specific blind spots is superior to generic random augmentation or mixing strategies.

5.7 Impact of Foundation Model Scale

To investigate the impact of the LLM size used in our augmentation module, we compared the performance of using Qwen3-30B-A3B versus Qwen3-235B-A22B on the HVPNeT backbone (Twitter-2017).

As shown in Table 3, the smaller 30B model achieves a remarkable F1-score of 87.02%, which not only significantly improves over the baseline (85.59%) but also outperforms other state-of-the-art augmentation methods (e.g., Vanilla and AMIA). Crucially, the performance gap between the 30B model and the massive 235B model (87.13%) is marginal (only 0.11%). This demonstrates that our Memory-Guided framework is highly effective even with smaller-scale foundation models, striking an excellent balance between performance and computational efficiency without heavy reliance on ultra-large parameters.

Table 3: Comparison of different Foundation Models for Augmentation (Backbone: HVPNeT, Dataset: Twitter-2017).

Foundation Model	Prec.	Rec.	F1
Ours (w/ Qwen3-30B)	86.48	87.56	87.02
Ours (w/ Qwen3-235B)	87.58	86.68	**87.13**

5.8 Effectiveness of the Filtering Model

To ensure the reliability of the Text-Based Filtering mechanism in Stage 4, we perform Supervised Fine-Tuning (SFT) on the Meta-Llama-3-8B-

(a) Original Hard Instance (b) Augmented Data

Figure 4: Case Study. (a) The baseline model misclassifies "Madison" due to textual ambiguity ("Madison's one direction") and visual sparsity. (b) Our framework generates augmented samples—City Sign and Map for LOC vs. Sports Team and School for ORG—to refine the decision boundary.

Instruct model. Specifically, we format the MNER training set into multi-turn JSON dialogues to train a robust NER validator. During the inference phase of data filtering, we prompt this model using the identical system instruction (detailed in Figure 12 in the Appendix) to extract entities from the generated text. If the predicted entity set does not strictly match the predetermined labels assigned during generation, the synthesized instance is intercepted and discarded.

To validate the filtering capability of this module, we evaluated its performance on the validation set. As shown in Table 4, the fine-tuned validator demonstrates highly robust performance across all metrics. This confirms its efficacy in accurately intercepting and filtering out conspicuous textual noise, ensuring that only high-quality texts or more complex cross-modal mismatches are passed to the subsequent model for further verification.

Table 4: Validation performance of the fine-tuned Meta-Llama-3-8B-Instruct filtering model.

Model	Prec. (%)	Rec. (%)	F1 (%)
Filter LLM (Llama-3-8B)	83.70	83.01	83.35

Table 5: Ablation study on the Twitter-2017 dataset (Backbone: HVPNeT).

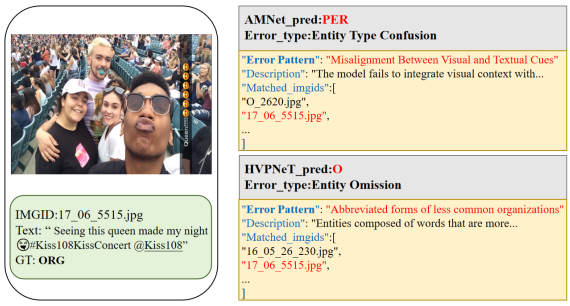
Variant	Prec.	Rec.	F1
Full Framework	87.58	86.68	**87.13**
w/o Model-Aware Mining	86.08	86.97	86.52
w/o Memory Tree Construction	86.32	87.34	86.83
w/o Iterative Augmentation	85.72	87.12	86.42
w/o Data Filtering	83.49	87.19	85.30

5.9 Ablation Study

To investigate the contribution of each component, we conducted ablation studies on the Twitter-2017 dataset using the HVPNeT backbone. The results are summarized in Table 5.

Impact of Model-Aware Mining. Replacing our specific hard data mining with random hard sample selection ("w/o Model-Aware Mining") leads to a performance drop. This indicates that targeting the specific weaknesses of the architecture is crucial for effective augmentation.

Effectiveness of Memory Tree Construction. Removing the Memory Tree ("w/o Memory Tree Construction") results in decreased stability. This



(a) Micro-level Hard Instance (b) Macro-level Error Patterns

Figure 5: A case study of error abstraction. It illustrates how model-specific failures on the same sample are diagnosed and mapped to distinct macro-level error patterns.

suggests that the clustering mechanism helps the model generalize macro-level error patterns, avoiding overfitting to micro-level noise.

Necessity of Data Filtering and Re-augmentation. The removal of the filtering ("w/o Data Filtering") introduces noise, degrading Precision significantly. Meanwhile, disabling the recursive generation ("w/o Iterative Augmentation") hurts the overall F1, as stubborn hard instances are not effectively reinforced.

5.10 Case Study I: Error Abstraction

To further elucidate how our framework abstracts micro-level errors into macro-level patterns, we present a detailed analysis of a representative instance (17_06_5515.jpg). As shown in Figure 5, the input text contains the phrase "Seeing this queen made my night #Kiss108KissConcert @Kiss108" where the target entity "Kiss108" is categorized as ORG.

Despite facing the same sample, different backbones exhibit distinct internal defects. For AMNet, the model exhibits Entity Type Confusion, misclassifying the ORG as PER. Our LLM-driven root cause analysis indicates that AMNet is overly influenced by the prominent visual features of the crowd in the image and the misleading word "queen" in the text, failing to properly integrate the visual context with the actual organizational entity. Consequently, this failure is clustered into the macro-level pattern: Misalignment Between Visual and Textual Cues. Conversely, HVPNeT suffers from Entity Omission, predicting the entity tokens as O. The diagnostic reasoning reveals that HVPNeT struggles to recognize unconventional or non-standard naming conventions (e.g., alphanumeric handles like "Kiss108"). This leads to its abstraction into the pattern: Abbreviated forms of less common organizations. This case study demonstrates that our framework does not merely expand data but provides model-specific, pattern-driven augmentation strategies by diagnosing the unique cognitive vulnerabilities of different architectures.

5.11 Case Study II: Logic-Driven Repair

Figure 4 illustrates a representative repair process on the Twitter-2017 training dataset. In the original instance (a), the model misclassifies "Madison" (Ground Truth: LOC) as PER, misled by the possessive phrasing "Madison’s one direction" and a low-light image containing a person. To resolve this, our framework generates augmented samples shown in (b). It synthesizes distinct visual contexts: clear city signs and maps for LOC (GEN_1, GEN_3) versus sports teams and school settings for ORG (GEN_2). This explicit visual disambiguation enables the model to learn clearer decision boundaries, successfully correcting the prediction to LOC. Furthermore, additional examples of augmented data generated in weakly-aligned scenarios are provided in Appendix C.

6 Conclusion

In this work, we addressed the limitations of existing model-independent and strategy-monolithic data augmentation methods, which often treat challenging instances homogeneously and fail to adequately rectify specific architectural blind spots. To effectively bridge this gap, we proposed the Memory-Guided Hard Data Augmentation framework as a comprehensive solution. By constructing a dynamic Memory Tree, we successfully transition from generic data expansion to a highly targeted, model-aware repair mechanism that abstracts micro-level hard instances into macro-level error patterns. Synergistically coupled with our Iterative Augmentation strategy, our framework consistently improves the baseline performance of multiple MNER architectures. Ultimately, our core contribution lies in introducing a novel methodological advancement in the field of data augmentation, rather than pursuing absolute state-of-the-art performance through model architecture innovations. We clearly demonstrate that shifting from random expansion to pattern-driven repair is a crucial step towards building more reliable and robust multimodal systems.

Limitations

Our current experiments are conducted primarily on social media datasets (Twitter), where image-text relationships are often loosely coupled. The effectiveness of our Pattern-Driven strategies in more specialized or strictly aligned domains (e.g., medical imaging reports) remains to be verified in future work.

The framework's effectiveness is inevitably bottlenecked by the hallucination issues inherent in Generative AI. While our data filtering stage mitigates low-quality synthesis, subtle semantic inconsistencies—such as an LLM misinterpreting a complex metaphor or the diffusion model generating text-irrelevant visual artifacts—may still propagate noise into the augmented dataset. Future work will focus on knowledge distillation techniques to reduce resource dependency and more fine-grained verification modules.

References

Xiang Chen, Ningyu Zhang, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, and Huajun Chen. 2022a. Hybrid transformer with multi-level fusion for multimodal knowledge graph completion. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 904–915.

Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Good visual guidance makes a better extractor: Hierarchical visual prefix for multimodal entity and relation extraction. *arXiv preprint arXiv:2205.03521*.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703.

Yinglong Dai, Feng Gao, and Daojian Zeng. 2023. An alignment and matching network with hierarchical visual features for multimodal named entity and relation extraction. In *International Conference on Neural Information Processing*, pages 298–310. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using large language models: Data perspectives, learning paradigms and challenges. *arXiv preprint arXiv:2403.02990*.

Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. 2023. Mixgen: A new multi-modal data augmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 379–389.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arxiv 2015. *arXiv preprint arXiv:1508.01991*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Ziyan Li, Jianfei Yu, Jia Yang, Wenya Wang, Li Yang, and Rui Xia. 2024. Generative multimodal data augmentation for low-resource multimodal named entity recognition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7336–7345.

Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1990–1999.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Lin Sun, Jiquan Wang, Kai Zhang, Yindu Su, and Fangsheng Weng. 2021. Rpbert: a text-image relation propagation-based bert model for multimodal ner. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13860–13868.

Xinyu Wang, Min Gui, Yong Jiang, Zixia Jia, Nguyen Bach, Tao Wang, Zhongqiang Huang, and Kewei Tu. 2022. Ita: Image-text alignments for multi-modal named entity recognition. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 3176–3189.

Zaitian Wang, Pengfei Wang, Kunpeng Liu, Pengyang Wang, Yanjie Fu, Chang-Tien Lu, Charu C Aggarwal, Jian Pei, and Yuanchun Zhou. 2025. A comprehensive survey on data augmentation. *IEEE Transactions on Knowledge and Data Engineering*.

Bo Xu, Shizhou Huang, Ming Du, Hongya Wang, Hui Song, Yanghua Xiao, and Xin Lin. 2023. A unified visual prompt tuning framework with mixture-of-experts for multimodal information extraction. In *International Conference on Database Systems for Advanced Applications*, pages 544–554. Springer.

Bo Xu, Shizhou Huang, Chaofeng Sha, and Hongya Wang. 2022. Maf: A general matching and alignment framework for multimodal named entity recognition. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1215–1223.

Bo Xu, Haiqi Jiang, Jie Wei, Hongyu Jing, Ming Du, Hui Song, Hongya Wang, and Yanghua Xiao. 2025. Enhancing multimodal named entity recognition through adaptive mixup image augmentation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1802–1812.

Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. Association for Computational Linguistics.

Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. 2021. Multi-modal graph fusion for named entity recognition with targeted visual guidance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14347–14355.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Xiaoying Zhou, Yijia Zhang, Zhuang Wang, Mingyu Lu, and Xiaoxia Liu. 2024. Mafn: multi-level attention fusion network for multimodal named entity recognition. *Multimedia Tools and Applications*, 83(15):45047–45058.

Appendix

A Detailed Implementation Settings

A.1 Dataset Statistics

Table 6 details the statistics of the Twitter-2015 and Twitter-2017 datasets used in our experiments. Both datasets are standard benchmarks for Multimodal Named Entity Recognition (MNER).

Table 6: The basic statistics of the Twitter-2015 and Twitter-2017 datasets.

	Twitter-2015			Twitter-2017		
Entity Type	**Train**	**Dev**	**Test**	**Train**	**Dev**	**Test**
Person	2,217	552	1,816	2,943	626	621
Location	2,091	522	1,697	731	173	178
Organization	928	247	839	1,674	375	395
Miscellaneous	940	225	726	701	150	157
Total	**6,176**	**1,546**	**5,078**	**6,049**	**1,324**	**1,351**
Num of Tweets	**4,000**	**1,000**	**3,257**	**3,373**	**723**	**723**

B Prompt Templates for LLMs

To facilitate reproducibility, we provide the exact prompt templates used in our framework. We use standard Jinja2 formatting where double curly braces '{{}}' denote input variables. Below, we present each prompt template as a separate figure for clarity.

B.1 Image Captioning Prompt

```
Model: qwen2.5-vl-72b-instruct
Prompt: "Please provide a detailed and accurate description of this image."
```

Figure 6: Prompt template for Image Captioning.

B.2 Root Cause Analysis (System Prompt)

Model: qwen3-235b-a22b
System Prompt:
You are an expert in the field of Multimodal Named Entity Recognition (MNER). Your core task is to diagnose the root causes of model prediction failures. Your analysis must be precise, insightful, and strictly follow the professional framework outlined below.
Error Type Definitions: To ensure consistency and professionalism in analysis, you must first understand and apply the following error classification system.

1. **Entity Boundary Detection Failure**: The entity type is correct, but the entity boundaries are inaccurately defined—either too long or too short.
   * Example: B-MISC, I-MISC -> B-MISC, O (The model only recognized "Ross" but missed "Torff," though it correctly identified "Ross" as MISC type).
2. **Entity Omission**: The entities that should have been present in the text were completely unrecognized by the model.
   * Example: B-MISC -> O (The model fails to recognize any entity).
3. **Entity Hallucination**: The model fabricates an entity where none exists.
   * Example: O -> B-PER (Mistakenly recognizing a common word as an entity).
4. **Entity Type Confusion**: The entity boundaries are perfectly correct, but the assigned entity category is wrong.
   * Example: B-ORG -> B-PER (An organization name is mislabeled as a person's name).

Now, based on the definitions provided above, please conduct an in-depth root cause analysis of the following MNER samples.
Data to be analyzed:
Input Details
1. All Tokens and Corresponding Labels:
The following data contains all the tokens from the dataset. Each row is in the format "token true_label predicted_label":
{{ all_tokens_context }}
2. Description of the Image Associated with the Tokens:
{{ description }}
3. Initial Error Judgment Provided:
{{ initial_error_judgment }}
Output Requirements: Please strictly follow the structure below, ensuring the analysis is concise, professional, and in-depth. The total word count should not exceed 400 words.
1. **Root Cause Analysis**: Expand and deepen the analysis of preliminary misclassifications based on sentence semantics and corresponding image descriptions.
2. **Exploration of Underlying Causes**: Do not merely describe the phenomenon. Infer **why** the model made these errors. Consider, but are not limited to, the following perspectives:

- **Contextual Ambiguity**: Does the text itself contain ambiguity making it hard for the model to decide?
- **Knowledge/Common Sense Deficiency**: Is the entity a rare word, emerging term, or domain-specific proper noun outside the model's pretraining knowledge?
- **Pattern Interference**: Is there a linguistic pattern in the text that conflicts with common patterns the model has learned from training data?
- **Sample Representation Deficiency**: Does this issue reflect the scarcity of such samples in the training data?

Figure 7: System prompt for Root Cause Analysis.

B.3.1 Pattern Clustering (System Prompt)

Role: MNER Error Analyst
I will provide a batch of error cases (ID and Reason) belonging to the category: "{{ coarse_type }}".
Your Task:
1. Analyze the root causes of these errors.
2. Group them into distinct sub-patterns based on intrinsic connections.
3. Output a JSON list where each object represents a sub-pattern.
JSON Format:

```
[
  {
    "pattern_name": "Name of the sub-pattern",
    "description": "Summary of the error logic",
    "matched_imgids": ["ID1", "ID2", ...]
  }
]
```

Important: Every provided ID in this batch must be assigned to exactly one pattern. Output MUST be a valid JSON list.

Figure 8: Prompt template for Pattern Clustering.

B.3.2 Pattern Merging (System Prompt)

Role: Pattern Deduplication Expert
I will provide you with a "New Pattern" and a list of "Existing Patterns".
Your Task: Determine if the "New Pattern" describes essentially the SAME root cause as any of the "Existing Patterns".
• If yes, return the 'pattern_name' of the existing pattern.
• If no, return "None".
Input Format:
New Pattern: {{ new_name }} - {{ new_desc }}
Existing List:
1. {{ name_1 }}: {{ desc_1 }} ...
Output: ONLY the pattern_name or "None".

Figure 9: Prompt template for Pattern Merging.

B.3.3 Strategy Generation (System Prompt)

Role: Data Augmentation Strategist
I will provide:
1. A specific error pattern name and description.
2. {{ k }} representative error reasons from actual cases.
Your Task: Analyze these reasons and formulate a specific data augmentation strategy to address this pattern. Output ONLY the strategy text.

Figure 10: Prompt template for Strategy Generation.

B.3.4 Data Augmentation (System Prompt)

Role: Top-tier Data Scientist in MNER
Based on the provided Strategy and the Reference Case tokens, generate 5 brand-new, grammatically correct, and diverse augmented samples.
Requirements:
1. **Strategic Augmentation:** Each generated sample must be based on the provided augmentation strategy.
2. **Accurate Annotation:** This is the most critical rule. The BIO labels must be perfectly accurate.
   * Entity label definitions: PER (Person), ORG (Organization), LOC (Location), MISC (Miscellaneous), O (Outside).
   * BIO format: Any I- label (e.g., I-PER) MUST immediately follow a B- or I- label of the SAME entity type. An I- label must never follow an O label.
3. **Multimodal Support:** For each sample, generate a concise and descriptive image caption that provides relevant visual context.
Output Format:
IMGID: GEN_{{ base_id }}_1
this O
road O
...
Description: An expert...

Figure 11: Prompt template for Data Augmentation.

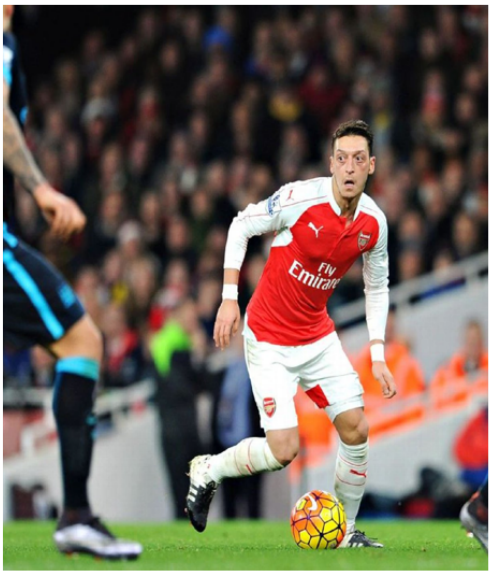
B.4 Quality Filtering (Filter LLM)

Model: Meta-Llama-3-8B-Instruct (SFT)
Prompt: "You are a precise named entity recognition assistant. Please extract the specified entities from the text and return them in a JSON list format."

Figure 12: Prompt template for the Logic Discriminator (Filter LLM).

C Examples of Weakly-Aligned Augmented Data

As shown in Figure 13, our proposed framework is capable of generating not only strongly-aligned multimodal pairs but also diverse weakly-aligned (weakly-matched) augmented data. This demonstrates the dynamism and flexibility of our generation strategy in exploring various contextual perturbations.



Text: " ...battle for #ChampionsLeague.."
Pred: O (Fail)
GT: B-ORG
Cause: Common Sense Deficiency

(a) Original Hard Instance

GEN_1 (ORG): "...to # ChampionsLeague..."

GEN_2 (ORG):" The # ChampionsLeague is..."

GEN_3 (ORG): "...in # ChampionsLeague ..."

GEN_4 (ORG): "the # ChampionsLeague is ..."

Analysis: This framework generates numerous augmentation cases where images are irrelevant to the entity, all containing the entity 'ChampionsLeague' in different contexts.

(b) Augmented Data

Figure 13: Examples of weakly-aligned augmented data generated by our framework.