# A Article extraction and validation

## A.1 Resources and hyperparameters

This section outlines the resources and parameters used in our experiments. Dataset generation with Open Source LLMs was conducted on an NVIDIA H100 PCIe GPU (80 GB * 2), while GPT was on smaller GPUs, as their API-based tasks required minimal computational resources. For first-step validation with the Qwen 2.5 32B instruct model, we used a temperature of 0.1 and top_p of 0.9.

## A.2 Match name formation

We initially used the Google News API to search for relevant articles with queries like: **"India v South Africa ODI 2023-11-05 before:2023-11-08 after:2023-11-02"** This query was designed to capture articles within a 3-day window before and after the match. However, the results were inconsistent and often failed to return the expected content. To improve the search process, we switched to the Google Search engine, using a more targeted query like: **"India v South Africa ODI 2023-11-05 articles"** We also leveraged Google's advanced search tools to precisely control the search window. This approach improved the search results by narrowing the focus to articles published within a defined time window. However, it remained unreliable for teams that frequently play against each other, as the search window would sometimes surface articles related to previous matches. To address this, we introduced an additional validation layer to filter out irrelevant articles. This involved cross-checking the retrieved articles with game-specific metadata to ensure they corresponded to the correct match. For validation, we provided a structured prompt ( 1) that included the article content and match metadata in the following format: **"India v South Africa ODI 2023-11-05"** This format captured key details - the sport, teams involved, and match date - enhancing the accuracy of article validation and reducing noise from unrelated matches. Comparative Performance of Open-Source LLMs in Validating Extracted Articles Based on Precision, Recall, and F1-Score Metrics can be seen from figure 2 .

# B Insight Generation

## B.1 Hyperparameters

This section outlines the resources and parameters used in our experiments. Dataset generation with Open Source LLMs was conducted on an NVIDIA H100 PCIe GPU (80 GB * 2), while GPT was on smaller GPUs, as their API-based tasks required minimal computational resources. We used temperature as 0.8 and top_p as 0.2 for the open-source models during the insight generation step but set the temperature to 0.15 and top_p to 0.8 for GPT-4o.

## B.2 Insight Generation Details and Prompt Design

Figure 3 illustrates the number of insights generated by each model across all sports domains. This comparison provides an overview of the insight generation capabilities and variability between different models. A higher count may indicate greater model expressiveness or verbosity, while a lower count may reflect more concise or filtered outputs.

To ensure consistency in evaluating model performance, we designed specific prompts tailored to each sport. These prompts are visualized in Figures 4, 5, and 6.

Figure 4 shows the prompt used for Soccer articles, structured to encourage the generation of analytical and context-aware insights regarding match results, player performances, and key moments.

Figure 5 presents the prompt crafted for Basketball articles, which emphasizes individual and team statistics, turning points in the game, and standout performances.

Figure 6 displays the prompt tailored for Baseball articles, with a focus on innings-based progress, pitcher vs. batter dynamics, and milestone achievements.

These prompts were systematically applied to ensure fair and domain-relevant generation across all models, supporting a controlled evaluation of the insight generation task.

# C Hallucination Detection

To evaluate the factual consistency of insights generated by different models, we employed two widely-used hallucination detection metrics: Fact-Score and Summac-Score. Figures 7 and 8 present the hallucination scores for each model based on these metrics.

Figure 7 displays the Fact-Score results, which measure factual alignment between the generated insight and the source document using entity and relation matching. Higher scores indicate better factual grounding and fewer hallucinations.

Figure 8 shows the Summac-Score evaluation, Structured prompt for sports specific article validation

*You are to carefully verify whether the article is about the given match information. Do not confuse it with other matches between the same teams. Use the match information to the best of your ability to ensure your verification is accurate and specific.*
*Return a one-word response:*
*'relevant' – if the article is about the specific match*
*'irrelevant' – if the article is not about the specific match.*

Figure 1: Prompt Employed in the Preliminary Stage of Article-Match Validation

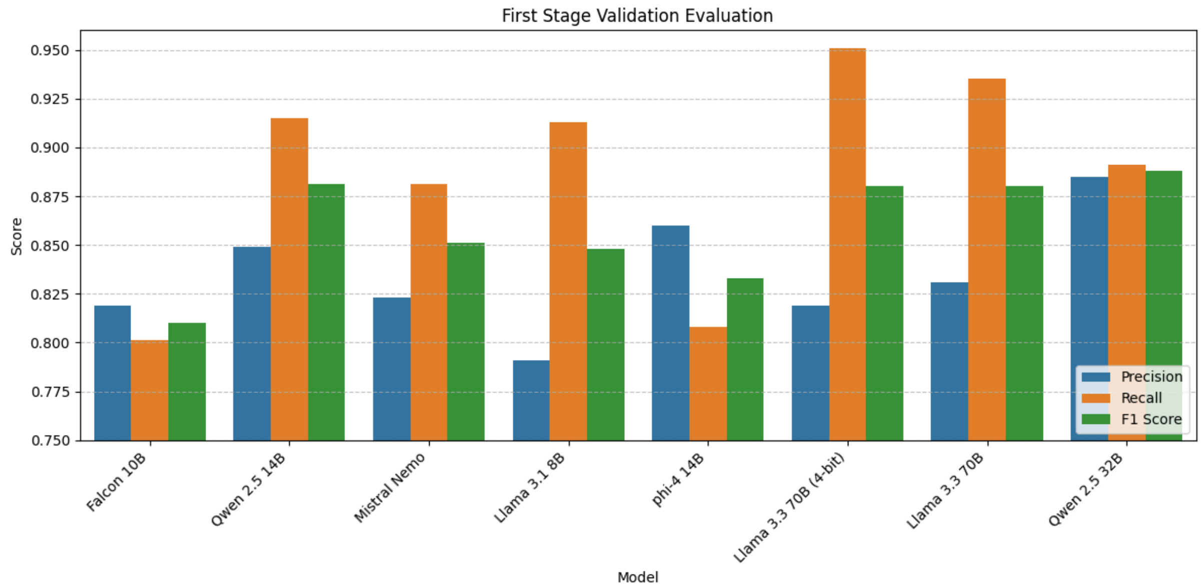


Figure 2: Evaluation of Model Performance Using a Gold-Standard Dataset of 996 Manually Labeled Sports Articles for Match Relevance Validation

which leverages a trained summarization consistency model to assess whether the generated insight can be inferred from the original article. As with Fact-Score, higher Summac-Scores suggest more reliable and faithful generation.

Together, these figures offer complementary perspectives on the hallucination tendencies of the models. They help identify which models produce factually sound content and which may require additional control mechanisms to reduce hallucinated information.

# D Insight Ranking Approaches

## D.1 Hyperparameters Settings

This section outlines the key hyperparameters and configurations used across the model training, PPO optimization, and evaluation components of our system.

### D.1.1 PPO Training Parameters

The Proximal Policy Optimization (PPO) algorithm was configured using the following parameters during reinforcement learning for ranking:

- `batch_size`: 1
- `mini_batch_size`: 1
- `learning_rate`: 2e-5
- `gradient_accumulation_steps`: 1
- `target_kl`: 0.2
- `cliprange`: 0.1
- `cliprange_value`: 0.1
- `max_grad_norm`: 0.5
- `seed`: 42

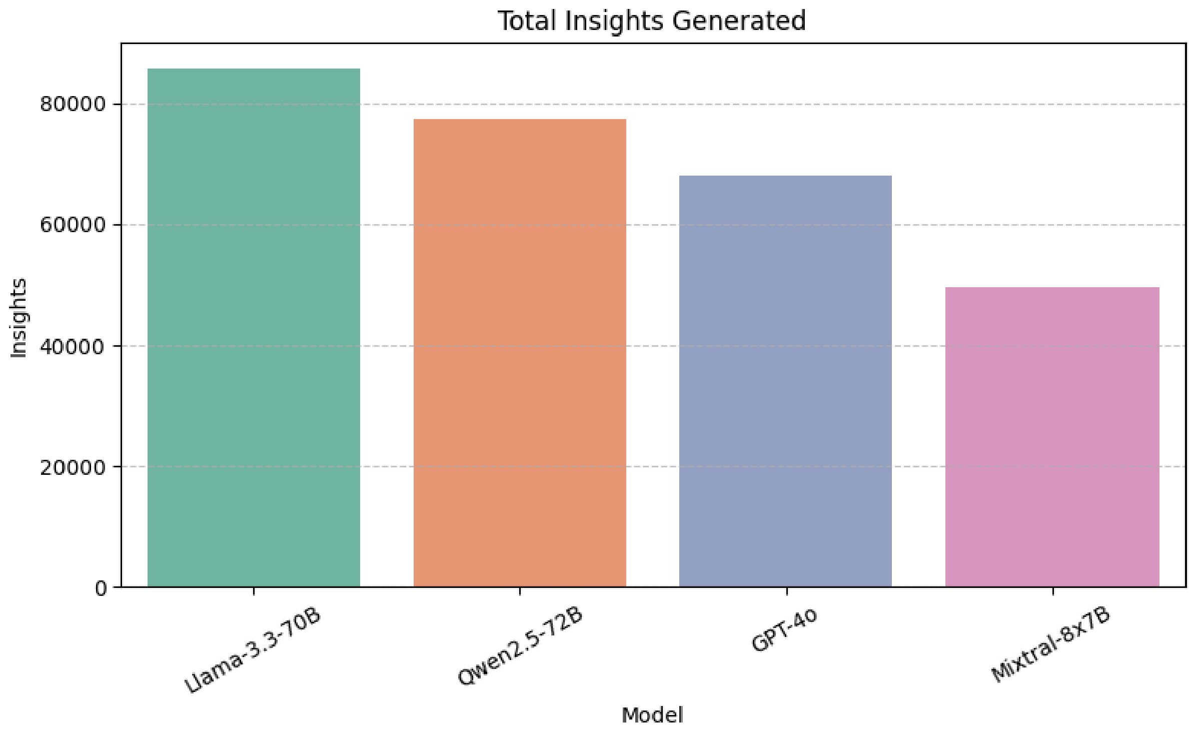


Figure 3: Quantitative Comparison of the Number of Insights Generated by Each Model During Sports Article Analysis

### D.1.2 ScoreNet Model Training

The ScoreNet model was trained using the following settings:

- Optimizer: Adam
- Learning Rate: 0.001
- Number of Epochs: 5
- Batch Size: 1
- Loss Function: ListNet-based with dynamic feature normalization

### D.1.3 Generation Parameters (T5-based Model Inference)

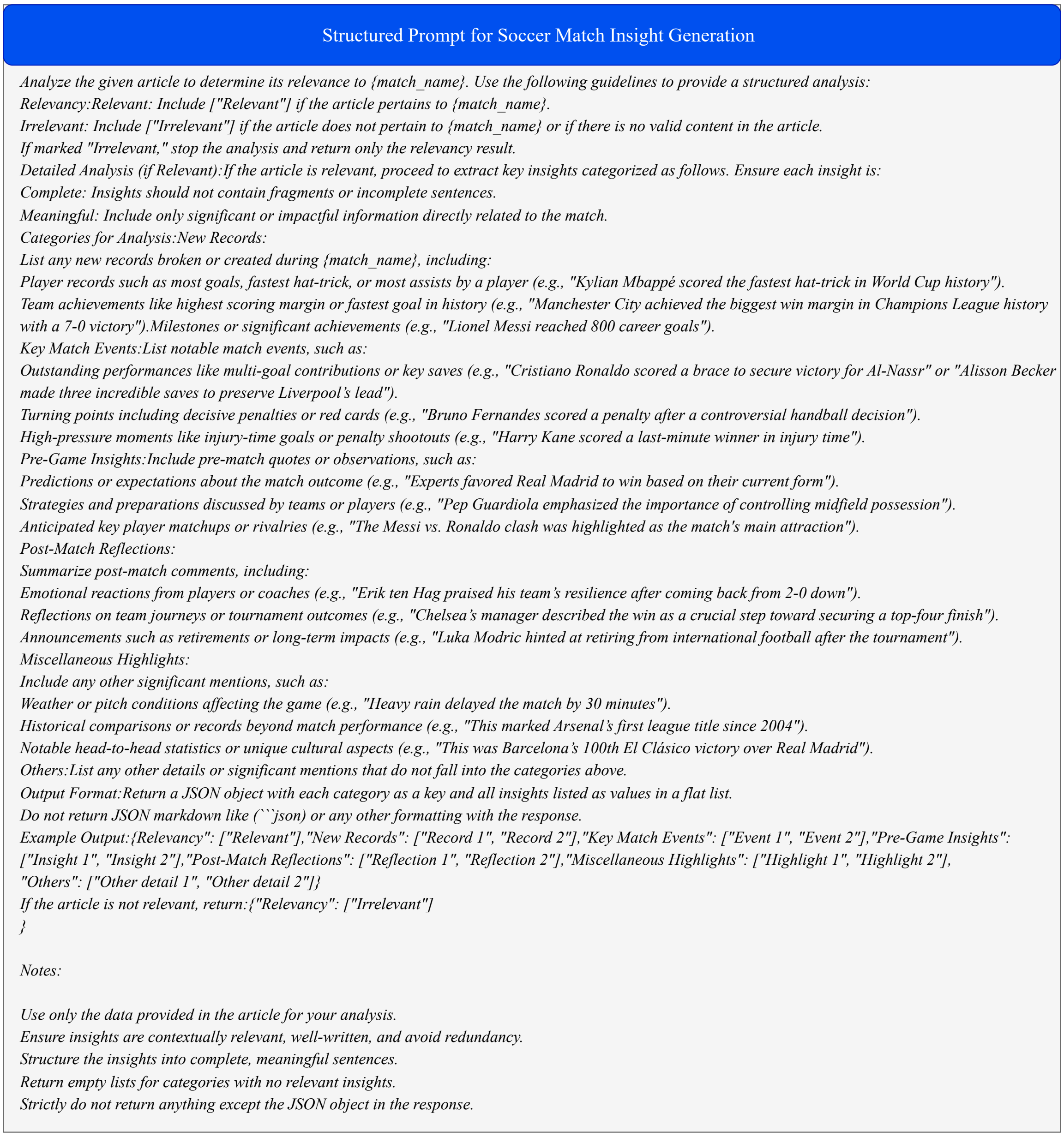
During evaluation using the pretrained model for sentence ranking, generation was performed with:

- `max_new_tokens`: $\min(100, 15 \times n\_sentences)$
- `do_sample`: True
- `top_p`: 0.9
- `num_beams`: 1
- `temperature`: 0.8
- `pad_token_id` and `eos_token_id`: set from tokenizer

These configurations were empirically chosen to balance performance and computational efficiency. Hyperparameter values were validated via experimental tuning on a development set.

## D.2 Computing Weights using Stochastic Gradient Descent

Determining optimal weights for aggregating the scores was challenging, as our training dataset has ranked insights validated by GPT-4o (Hurst et al., 2024) and human annotators without explicit numerical ground truths. After comparing multiple optimization strategies—including Ridge Regression and Neural Networks—we selected Stochastic Gradient Descent (SGD) (Robbins and Monro, 1951) for its efficiency and ability to manage high-dimensional features effectively. We began by assigning equal initial weights across all six features. These weights were iteratively optimized through SGD, employing a pairwise hinge loss function designed to minimize discrepancies between the computed rankings and the validated reference rankings provided by GPT-4o (Hurst et al., 2024) and human evaluation.



Structured Prompt for Soccer Match Insight Generation

*Analyze the given article to determine its relevance to {match_name}. Use the following guidelines to provide a structured analysis:*
*Relevancy:Relevant: Include ["Relevant"] if the article pertains to {match_name}.*
*Irrelevant: Include ["Irrelevant"] if the article does not pertain to {match_name} or if there is no valid content in the article.*
*If marked "Irrelevant," stop the analysis and return only the relevancy result.*
*Detailed Analysis (if Relevant):If the article is relevant, proceed to extract key insights categorized as follows. Ensure each insight is:*
*Complete: Insights should not contain fragments or incomplete sentences.*
*Meaningful: Include only significant or impactful information directly related to the match.*
*Categories for Analysis:New Records:*
*List any new records broken or created during {match_name}, including:*
*Player records such as most goals, fastest hat-trick, or most assists by a player (e.g., "Kylian Mbappé scored the fastest hat-trick in World Cup history").*
*Team achievements like highest scoring margin or fastest goal in history (e.g., "Manchester City achieved the biggest win margin in Champions League history with a 7-0 victory").Milestones or significant achievements (e.g., "Lionel Messi reached 800 career goals").*
*Key Match Events:List notable match events, such as:*
*Outstanding performances like multi-goal contributions or key saves (e.g., "Cristiano Ronaldo scored a brace to secure victory for Al-Nassr" or "Alisson Becker made three incredible saves to preserve Liverpool's lead").*
*Turning points including decisive penalties or red cards (e.g., "Bruno Fernandes scored a penalty after a controversial handball decision").*
*High-pressure moments like injury-time goals or penalty shootouts (e.g., "Harry Kane scored a last-minute winner in injury time").*
*Pre-Game Insights:Include pre-match quotes or observations, such as:*
*Predictions or expectations about the match outcome (e.g., "Experts favored Real Madrid to win based on their current form").*
*Strategies and preparations discussed by teams or players (e.g., "Pep Guardiola emphasized the importance of controlling midfield possession").*
*Anticipated key player matchups or rivalries (e.g., "The Messi vs. Ronaldo clash was highlighted as the match's main attraction").*
*Post-Match Reflections:*
*Summarize post-match comments, including:*
*Emotional reactions from players or coaches (e.g., "Erik ten Hag praised his team's resilience after coming back from 2-0 down").*
*Reflections on team journeys or tournament outcomes (e.g., "Chelsea's manager described the win as a crucial step toward securing a top-four finish").*
*Announcements such as retirements or long-term impacts (e.g., "Luka Modric hinted at retiring from international football after the tournament").*
*Miscellaneous Highlights:*
*Include any other significant mentions, such as:*
*Weather or pitch conditions affecting the game (e.g., "Heavy rain delayed the match by 30 minutes").*
*Historical comparisons or records beyond match performance (e.g., "This marked Arsenal's first league title since 2004").*
*Notable head-to-head statistics or unique cultural aspects (e.g., "This was Barcelona's 100th El Clásico victory over Real Madrid").*
*Others:List any other details or significant mentions that do not fall into the categories above.*
*Output Format:Return a JSON object with each category as a key and all insights listed as values in a flat list.*
*Do not return JSON markdown like (```json) or any other formatting with the response.*
*Example Output:{Relevancy": ["Relevant"],"New Records": ["Record 1", "Record 2"],"Key Match Events": ["Event 1", "Event 2"],"Pre-Game Insights": ["Insight 1", "Insight 2"],"Post-Match Reflections": ["Reflection 1", "Reflection 2"],"Miscellaneous Highlights": ["Highlight 1", "Highlight 2"], "Others": ["Other detail 1", "Other detail 2"]}*
*If the article is not relevant, return:{"Relevancy": ["Irrelevant"]*
*}*

*Notes:*

*Use only the data provided in the article for your analysis.*
*Ensure insights are contextually relevant, well-written, and avoid redundancy.*
*Structure the insights into complete, meaningful sentences.*
*Return empty lists for categories with no relevant insights.*
*Strictly do not return anything except the JSON object in the response.*

Figure 4: Comprehensive Prompt for Relevance Classification and Structured Insight Extraction from Soccer Sports Articles Using Match Metadata

We consider a ranked set of sentences produced by GPT4o (Hurst et al., 2024). Let $a_1, a_2, \ldots, a_n$ denote these sentences in descending order of rank with $a_1$ as the highest scoring one. Each sentence $a_i$ is represented by a feature vector $\mathbf{x}_i = [F_{i1}, F_{i2}, F_{i3}, F_{i4}, F_{i5}, F_{i6}]$ corresponding, respectively, to Semantic Score, Emotional Intensity Score, TF-IDF Score, Buzzword Score, NER People Score and Sarcasm Score. We define a scoring function as the weighted sum of the features:

$$s(a_i) = \mathbf{w}^\top \mathbf{x}_i = \sum_{j=1}^{6} w_j \, x_{ij}, \qquad (1)$$

where $\mathbf{w} = [w_1, w_2, w_3, w_4, w_5, w_6]$ is the weight vector that we wish to learn. For the ranking to be correct, if $a_i$ is ranked higher than $a_j$ (i.e., $i < j$), then the scores should satisfy

$$s(a_i) - s(a_j) \geq \Delta, \qquad (2)$$

Structured Prompt for Basketball Match Insight Generation

*Analyze the given article to determine its relevance to {match_name}. Use the following guidelines to provide a structured analysis:*

*Relevancy:*

*Relevant: Include ["Relevant"] if the article pertains to {match_name}.*

*Irrelevant: Include ["Irrelevant"] if the article does not pertain to {match_name} or if there is no valid content in the article.*

*If marked "Irrelevant," stop the analysis and return only the relevancy result.*

*Detailed Analysis (if Relevant):*

*If the article is relevant, proceed to extract key insights categorized as follows. Ensure each insight is:*

*Complete: Insights should not contain fragments or incomplete sentences.*

*Meaningful: Include only significant or impactful information directly related to the match.*

*Categories for Analysis:*

*New Records:*

*List any new records broken or created during {match_name}, including:*

*Player Records such as most points scored in a game, fastest triple-double, highest three-point percentage in a match (e.g., "LeBron James scored a career-high 61 points").*

*Team Records like largest victory margin, most three-pointers made in a game (e.g., "The Golden State Warriors set a new record with 30 three-pointers in a single game").*

*Milestones or significant achievements (e.g., "Stephen Curry reached 3,000 career three-pointers").*

*Key Match Events:*

*List notable match events, such as:*

*Outstanding performances like multi-point contributions or key defensive plays (e.g., "Nikola Jokic recorded a triple-double with 35 points, 15 rebounds, and 12 assists").*

*Turning points including game-winning shots, critical fouls, or momentum-shifting plays (e.g., "Damian Lillard hit a game-winning three-pointer at the buzzer").*

*High-pressure moments like overtime thrillers, last-minute defensive stops, or clutch free throws (e.g., "Kawhi Leonard made two clutch free throws to seal the win in overtime").*

*Pre-Game Insights:*

*Include pre-match quotes or observations, such as:*

*Predictions or expectations about the match outcome (e.g., "Experts favored the Lakers to win based on their recent performance").*

*Strategies and preparations discussed by teams or players (e.g., "Coach Steve Kerr emphasized the importance of strong perimeter defense").*

*Anticipated key player matchups or rivalries (e.g., "The duel between Kevin Durant and Giannis Antetokounmpo was highlighted as the key matchup").*

*Post-Match Reflections:*

*Summarize post-match comments, including:*

*Emotional reactions from players or coaches (e.g., "LeBron James expressed pride in his team's effort after the comeback win").*

*Reflections on team journeys or tournament outcomes (e.g., "The coach highlighted the victory as a crucial step toward the championship").*

*Announcements such as player retirements or long-term impacts (e.g., "Vince Carter announced his retirement after the game").*

*Miscellaneous Highlights:*

*Include any other significant mentions, such as:*

*Arena conditions or unusual events affecting the match (e.g., "A power outage delayed the game by 30 minutes").*

*Historical comparisons or records beyond match performance (e.g., "The team ended a 20-year championship drought").*

*Notable head-to-head statistics or unique historical aspects (e.g., "This was the 100th meeting between the two rival teams").*

*Others:List any other details or significant mentions that do not fall into the categories above.*

*Output Format:Return a JSON object with each category as a key and all insights listed as values in a flat list.*

*Do not return JSON markdown like (```json) or any other formatting with the response.*

*Example Output:{"Relevancy": ["Relevant"],*
*"New Records": ["Record 1", "Record 2"],*
*"Key Match Events": ["Event 1", "Event 2"],*
*"Pre-Game Insights": ["Insight 1", "Insight 2"],*
*"Post-Match Reflections": ["Reflection 1", "Reflection 2"],*
*"Miscellaneous Highlights": ["Highlight 1", "Highlight 2"],*
*"Others": ["Other detail 1", "Other detail 2"]}*

*If the article is not relevant, return:{"Relevancy": ["Irrelevant"]}*

*Notes:*

*Use only the data provided in the article for your analysis.*

*Ensure insights are contextually relevant, well-written, and avoid redundancy.*

*Structure the insights into complete, meaningful sentences.*

*Return empty lists for categories with no relevant insights.*

*Strictly do not return anything except the JSON object in the response*

Figure 5: Structured prompt framework used for extracting comprehensive insights from basketball sports articles, including relevancy assessment and categorization of key match details such as records, events, player milestones, and reflections.

with a margin $\Delta > 0$ (taking, $\Delta = 1$). To enforce this condition, we define the pairwise hinge loss for a consecutive pair $(a_i, a_{i+1})$ as

$$L(a_i, a_{i+1}) = \max\{0, \Delta - (s(a_i) - s(a_{i+1}))\}. \tag{3}$$

The overall loss for one ranked observation is then

$$L = \sum_{i=1}^{n-1} L(a_i, a_{i+1}) = \sum_{i=1}^{n-1} \max\{0,\ \Delta - (s(a_i) - s(a_{i+1}))\}. \tag{4}$$

When $L(a_i, a_{i+1}) > 0$, the gradient with respect to $\mathbf{w}$ is

$$\frac{\partial L(a_i, a_{i+1})}{\partial \mathbf{w}} = -\mathbf{x}_i + \mathbf{x}_{i+1}, \tag{5}$$

and we update the weights using stochastic gradient descent (SGD) as follows:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \left(-\mathbf{x}_i + \mathbf{x}_{i+1}\right), \tag{6}$$

where $\eta$ is the learning rate. After each update, we normalize the weight vector to ensure that

$$\sum_{j=1}^{6} w_j = 1, \quad \text{i.e.} \quad \mathbf{w} \leftarrow \frac{\mathbf{w}}{\sum_{j=1}^{6} w_j}. \tag{7}$$

This iterative process is repeated for all observations and over multiple epochs.

### D.3 Computing Weights using Single-Neuron Regression Model

While our first approach in D.2 directly learns a weight vector $\mathbf{w}$ via stochastic gradient descent with hinge-loss ranking, we also explore a neural-style model. Each sentence $a_i$ is still represented by a feature vector $\mathbf{x}_i = [x_{i1}, \ldots, x_{i6}]$. Instead of learning the weights $\mathbf{w} = [w_1, \ldots, w_6]$ directly, we introduce trainable logits $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_6]$. A softmax transformation converts these logits into nonnegative weights that sum to one:

$$w_j = \frac{e^{\theta_j}}{\sum_{k=1}^{6} e^{\theta_k}},$$

ensuring $\sum_{j=1}^{6} w_j = 1$.
Given an input $\mathbf{x}_i$, the model assigns a score

$$s(a_i) = \mathbf{w}^\top \mathbf{x}_i = \sum_{j=1}^{6} w_j\, x_{ij}.$$

We train this model using Mean Squared Error (MSE) loss. Let $y_i$ be the target score for $a_i$; then the per-sentence loss is

$$L = \big(s(a_i) - y_i\big)^2,$$

and we minimize the average loss over all training samples. The logits $\boldsymbol{\theta}$ are updated via backpropagation and Adam, automatically enforcing valid (nonnegative, normalized) weights.

Compared to the hinge-loss ranking approach, this single-neuron model uses explicit score labels $\{y_i\}$ and employs a probabilistic interpretation of the weights. The Adam optimizer typically simplifies hyperparameter tuning, and softmax normalization provides a convenient constraint, but explicit numerical targets are required to guide the MSE objective.

### D.4 Results and Discussion

We selected articles from 35 matches across four sports for the insight ranking process. To ensure accuracy and consistency, these articles underwent three stages, as outlined in Section **??**: data validation, insight generation, and hallucination detection. Using data from these 35 matches, we constructed a robust ground truth ranking dataset by leveraging GPT-4o (Hurst et al., 2024) with human observation. The rankings generated through this process served as the ground truth for optimizing the weight parameters of our ranking module. To evaluate the performance of our ranking model, we employed widely recognized ranking metrics, including NDCG@K (Järvelin and Kekäläinen, 2002) and Recall@K (Buckley and Voorhees, 2017). The results, summarized in Table 1, demonstrate the model's capability to produce more accurate and contextually relevant rankings, validating the effectiveness of our approach.

### D.5 Error Analysis

We analyzed errors in our insight-ranking module to identify challenges affecting insight prioritization. Key issues included occasional misclassification of sarcasm and emotional intensity, leading to incorrect insight prioritization. Additionally, the TF-IDF component occasionally overemphasized frequent but less impactful terms, and the NER module sometimes introduced biases toward high-profile entities. Future enhancements could address these limitations by refining sarcasm detection, optimizing semantic embeddings, and recalibrating feature weights with expanded ground-truth data.

## E SUMMIR

### E.1 Preliminaries and Notation

Consider a ranking instance

$$\mathcal{S} = \{s_0, \ldots, s_{n-1}\} \quad \text{of} \quad n > 1 \text{ candidate sentences.}$$

Table 1: Comparison of Top 3 samples based on NDCG@10 with their average metrics, while the weights are computed using the Stochastic Gradient Descent approach and Single-Neuron Regression Model

| | SGD (Robbins and Monro, 1951) | | | | Single-Neuron Regression Model (Bishop, 1995) | | | |
|---|---|---|---|---|---|---|---|---|
| Metric/Sample | Top 1 | Top 2 | Top 3 | Avg. | Top 1 | Top 2 | Top 3 | Avg. |
| NDCG@5 | 0.9577 | 0.6590 | 0.7492 | 0.7887 | 0.6767 | 0.7786 | 0.7474 | 0.7342 |
| NDCG@10 | 0.9620 | 0.8246 | 0.7592 | 0.8486 | 0.8250 | 0.7780 | 0.7374 | 0.7801 |
| Recall@5 | 0.6000 | 0.2000 | 0.4000 | 0.4000 | 0.4000 | 0.8000 | 0.4000 | 0.5333 |
| Recall@10 | 0.7000 | 0.5000 | 0.6000 | 0.6000 | 0.4000 | 0.7000 | 0.3000 | 0.4667 |

Each sentence $s_i$ is associated with a pre–computed, bounded feature vector

$$\mathbf{x}_i = (x_{i1}, \ldots, x_{i6})^\top \in [0,1]^6. \tag{8}$$

We denote by

$$\pi_\theta\big(\mathbf{y} \,\big|\, \mathcal{S}\big)$$

the autoregressive policy implemented by a 1 B-parameter LLaMA model with trainable parameters $\theta$; and by $\pi_{\theta_{\text{ref}}}$ a frozen reference copy of the same architecture.

Whenever $\pi_\theta$ is prompted with $\mathcal{S}$, it emits a sequence of indices

$$\begin{gathered} \mathbf{p} = (p_0, \ldots, p_{n-1}) \in \mathfrak{S}_n, \\ p_0 \text{ (“best”)} \longrightarrow p_{n-1} \text{ (“worst”)}. \end{gathered} \tag{9}$$

Human supervisors supply a gold permutation

$$\mathbf{g} = (g_0, \ldots, g_{n-1}) \in \mathfrak{S}_n. \tag{10}$$

—

## E.2 ScoreNet: A Differentiable Relevance Prior

A light-weight fully differentiable scoring function

$$f_{\boldsymbol{\ell}} \colon [0,1]^6 \to \mathbb{R}$$

**ScoreNet** as we call it ( it basically gives the score n is same as the perceptron used in prev work)

$$\mathbf{w} = \operatorname{softmax}(\boldsymbol{\ell}), \quad f_{\boldsymbol{\ell}}(\mathbf{x}) = \sum_{j=1}^{6} w_j \, x_j \tag{11}$$

where $\boldsymbol{\ell} \in \mathbb{R}^6$ are trainable logits. Given the six-tuple of sentence features, ScoreNet yields continuous relevance scores

$$s_i := f_{\boldsymbol{\ell}}(\mathbf{x}_i), \quad i = 0, \ldots, n-1. \tag{12}$$

Although $\boldsymbol{\ell}$ is pre-optimised offline, Eq. (5) furnishes a differentiable *priory ranking* $\mathbf{p}^{\text{SN}}$.

## E.3 Reward Shaping via Normalised DCG

### E.3.1 Relevance Vectors

Gold and ScoreNet relevance values are defined, respectively, as

$$r_i^{\text{gold}} = n - \operatorname{rank}_{\mathbf{g}}(i), \quad r_i^{\text{SN}} = s_i. \tag{13}$$

### E.3.2 DCG and NDCG

For a permutation $\mathbf{p}$ and relevance vector $\mathbf{r}$,

$$\begin{gathered} \operatorname{DCG}_k(\mathbf{p}, \mathbf{r}) = \sum_{t=1}^{k} \frac{2^{\,r_{p_t}} - 1}{\log_2(t+1)}, \\ \operatorname{IDCG}_k(\mathbf{r}) = \operatorname{DCG}_k\big(\operatorname*{argsort}_i(-r_i), \, \mathbf{r}\big) \end{gathered} \tag{14}$$

$$\operatorname{NDCG}_k(\mathbf{p}, \mathbf{r}) = \frac{\operatorname{DCG}_k(\mathbf{p}, \mathbf{r})}{\operatorname{IDCG}_k(\mathbf{r})}. \tag{15}$$

We employ $k = \max\!\big(1, \lfloor n/2 \rfloor\big)$.

### E.3.3 Scalar Reward

Two NDCG measurements are computed for the policy output $\mathbf{p}$:

$$\begin{aligned} N_{\text{gold}} &= \operatorname{NDCG}_k(\mathbf{p}, \mathbf{r}^{\text{gold}}), \\ N_{\text{SN}} &= \operatorname{NDCG}_k(\mathbf{p}, \mathbf{r}^{\text{SN}}) \end{aligned} \tag{16}$$

A convex combination $(\lambda_1 = 0.7, \ \lambda_2 = 0.3)$ is then mapped to $(0,1)$ via a sigmoid:

$$\begin{aligned} \hat{R} &= \lambda_1 \, N_{\text{gold}} + \lambda_2 \, N_{\text{SN}}, \\ R &= \sigma(\hat{R}) = \frac{1}{1 + e^{-\hat{R}}} \end{aligned} \tag{17}$$

$R$ acts as the sole environment reward.

## E.4 Policy Optimisation with PPO

Let $\mathbf{y} = (y_1, \ldots, y_T)$ be the full token sequence produced by $\pi_\theta$. We adopt the clipped PPO objective (Schulman *et al.*, 2017).

### E.4.1 Advantage Estimate

$$A = R - V_\phi(\mathbf{y}), \tag{18}$$

where $V_\phi$ is the value head with parameters $\phi$.

#### E.4.2 Probability Ratio per Token

$$r_t(\theta) = \frac{\pi_\theta\big(y_t \mid \mathbf{y}_{<t}, \mathcal{S}\big)}{\pi_{\theta_{\text{old}}}\big(y_t \mid \mathbf{y}_{<t}, \mathcal{S}\big)}, \tag{19}$$

#### E.4.3 Surrogate Loss

$$\begin{aligned}\mathcal{L}_{\text{clip}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \min\!\big(r_t(\theta)\, A_t,\\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\, A_t\big)\end{aligned} \tag{20}$$

#### E.4.4 Auxiliary Terms and Full Objective

$$\mathcal{L}_V(\phi) = \tfrac{1}{2}\big(V_\phi(\mathbf{y}) - R\big)^2, \tag{21}$$

$$\mathcal{L}_H(\theta) = -\beta\, \mathcal{H}[\pi_\theta], \tag{22}$$

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{clip}}(\theta) + c_V\, \mathcal{L}_V(\phi) + c_H\, \mathcal{L}_H(\theta), \tag{15}$$

$$\boxed{c_V = 1,\ c_H = 0.01,\ \beta > 0}$$

Gradients are clipped to $\|\nabla\mathcal{L}\|_2 \le 0.5$, and updates proceed until $\text{KL}\big(\pi_\theta \,\|\, \pi_{\theta_{\text{ref}}}\big) > 0.2$.

### E.5 Training Procedure

For each epoch $e = 1, \ldots, E$:

1. **Sample** a ranking instance $\mathcal{S}$ and construct the prompt containing ScoreNet scores $\{s_i\}$.

2. **Generate** permutation $\mathbf{p} \sim \pi_\theta(\,\cdot\,|\mathcal{S})$ via nucleus sampling ($p = 0.9, T = 0.7$).

3. **Compute reward** $R$ according to Eqs. (6)–(10).

4. **Optimise** $\theta, \phi$ by stochastic gradient descent on Eq. (15).

5. **Checkpoint** $\pi_\theta^{(e)}$, the tokenizer, and $\boldsymbol{\ell}$ (if updated).

#### E.5.1 Insight Ranking Prompt

To ensure consistent and interpretable ranking of sports-related sentences, we designed a structured prompt (Figure 9) that guides a language model to evaluate and order candidate insights.

#### E.5.2 Sample Datapoint for Insight Ranking Framework

#### E.5.3 Training Prompt

### E.6 Detailed Ranking Results

To better understand the output behavior of our ranking models under various reward configurations, we present detailed top-5 permutation results for each setting. These tables report performance metrics including NDCG@2/5/10 and Recall@2/5/10, across models of different sizes (e.g., LLaMA-3.2-1B and LLaMA-3.2-3B) and reward formulations (e.g., multi-metric, NDCG-only, Recall-only).

By analyzing both individual permutation scores and their mean, this breakdown enables a more granular comparison of ranking quality and robustness across experiments. The results also highlight consistency patterns and variance introduced by different objective designs. These insights complement the main findings and offer practical guidance for future reward shaping or model selection.

## F Human Annotation

To ensure the reliability of our ranking and validation process, we initially conducted human annotation on a subset of 996 articles. This initial validation was performed solely by the authors of the paper, without the involvement of any external annotators. After establishing a reliable annotation framework, the majority of the subsequent annotations were generated using large language models (LLMs), specifically LLaMA, under continuous human supervision by the authors. All automated annotations were carefully reviewed and verified by the authors to ensure consistency and accuracy. No additional human annotators beyond the authors themselves were employed at any stage of this study.

## G Frequently Asked Questions

**Q1: Why did you use multiple reward metrics (6-metric) instead of relying solely on NDCG?**

A1: While NDCG effectively captures ranking quality, we found that combining it with complementary signals (like semantic meaning, emotional intensity, sarcasm, and entity presence) encourages more diverse and human-aligned rankings. This multi-faceted reward better reflects real-world editorial preferences in sports content.

**Q2: What is the role of the training-time prompt in Figure 11?**

A2: The training-time prompt is designed to provide a structured and interpretable scoring signal. It includes per-sentence breakdowns across six dimensions, along with a final score. This improves transparency for the model and aligns the reward model more closely with domain-specific editorial heuristics.

Table 2: Llama-3.2-1B (6-metric reward) output: top-5 permutations and their mean.

| Rank | NDCG@2 | NDCG@5 | NDCG@10 | Recall@2 | Recall@5 | Recall@10 |
|---|---|---|---|---|---|---|
| #1 | 1.0000 | 0.9982 | 0.9980 | 1.0000 | 1.0000 | 0.9000 |
| #2 | 1.0000 | 0.9982 | 0.9976 | 1.0000 | 1.0000 | 1.0000 |
| #3 | 0.8800 | 0.9379 | 0.9712 | 0.5000 | 0.6000 | 1.0000 |
| #4 | 1.0000 | 0.9128 | 0.9741 | 1.0000 | 0.6000 | 0.9000 |
| #5 | 0.4099 | 0.6937 | 0.7730 | 0.5000 | 0.6000 | 1.0000 |
| **Average** | **0.8580** | **0.9082** | **0.9428** | **0.8000** | **0.7600** | **0.9600** |

Table 3: RECALL ONLY USING Llama-3.2-1B (top-5 permutations and their mean).

| Rank | NDCG@2 | NDCG@5 | NDCG@10 | Recall@2 | Recall@5 | Recall@10 |
|---|---|---|---|---|---|---|
| #1 | 1.0000 | 0.9494 | 0.9886 | 1.0000 | 0.6000 | 0.9000 |
| #2 | 0.8800 | 0.9379 | 0.9712 | 0.5000 | 0.6000 | 1.0000 |
| #3 | 0.8597 | 0.8237 | 0.8139 | 1.0000 | 0.6000 | 0.5000 |
| #4 | 0.7750 | 0.7877 | 0.9099 | 0.5000 | 0.6000 | 0.9000 |
| #5 | 0.7610 | 0.6944 | 0.8653 | 0.5000 | 0.4000 | 1.0000 |
| **Average** | **0.8552** | **0.8386** | **0.9098** | **0.7000** | **0.5600** | **0.8600** |

Table 4: NDCG ONLY USING Llama-3.2-1B (top-5 permutations and their mean).

| Rank | NDCG@2 | NDCG@5 | NDCG@10 | Recall@2 | Recall@5 | Recall@10 |
|---|---|---|---|---|---|---|
| #1 | 1.0000 | 0.9683 | 0.9671 | 1.0000 | 0.8000 | 0.9000 |
| #2 | 0.8800 | 0.9463 | 0.9534 | 0.5000 | 0.8000 | 0.9000 |
| #3 | 0.8200 | 0.9262 | 0.9315 | 0.5000 | 0.8000 | 0.9000 |
| #4 | 0.8597 | 0.7854 | 0.8417 | 1.0000 | 0.6000 | 0.9000 |
| #5 | 0.7675 | 0.7019 | 0.8599 | 0.5000 | 0.4000 | 1.0000 |
| **Average** | **0.8655** | **0.8656** | **0.9107** | **0.7000** | **0.6800** | **0.9200** |

**Q3: How were the rankings evaluated during inference?**

A3: We evaluated rankings using NDCG@2/5/10 and Recall@2/5/10 against gold-standard rankings. This setup measures how much the model retrieves and how well it prioritises what it retrieves. The three cut-offs capture performance in the immediate top slots and across a broader range.

**Q4: Could this ranking framework generalize to domains beyond sports?**

A4: Yes. While our prompts are sports-specific, the modular design of the reward function and ranking policy can generalize to domains like finance, entertainment, or politics by adjusting the feature set and prompt criteria.

**Q5: Why were $\lambda_1 = 0.7$ (Gold) and $\lambda_2 = 0.3$ (SUMMIR) selected as the weighting coefficients in the ranking objective?**

A5: These coefficients were chosen to introduce a explicit "interesting" factor using the ranking generated by ScoreNet while not straying away from the Gold Ranking hence the $\lambda_1 = 0.7$ and $\lambda_2 = 0.3$.

## References

Christopher M Bishop. 1995. *Neural networks for pattern recognition*. Oxford university press.

Chris Buckley and Ellen M Voorhees. 2017. Evaluating evaluation measure stability. In *ACM SIGIR Forum*, volume 51, pages 235–242. ACM New York, NY, USA.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

**Structured Prompt for Baseball Match Insight Generation**

*Analyze the given article to determine its relevance to {match_name}. Use the following guidelines to provide a structured analysis:*

*Relevancy:*

*Relevant: Include ["Relevant"] if the article pertains to {match_name}.*

*Irrelevant: Include ["Irrelevant"] if the article does not pertain to {match_name} or if there is no valid content in the article.*

*If marked "Irrelevant," stop the analysis and return only the relevancy result.*

*Detailed Analysis (if Relevant):*

*If the article is relevant, proceed to extract key insights categorized as follows. Ensure each insight is:*

*Complete: Insights should not contain fragments or incomplete sentences.*

*Meaningful: Include only significant or impactful information directly related to the match.*

*Categories for Analysis:*

*New Records:*

*List any new records broken or created during {match_name}, including:*

*Player Records such as most home runs in a game, fastest pitch speed recorded, longest hitting streak (e.g., "Shohei Ohtani hit the fastest pitch recorded in MLB history").*

*Team Records like largest winning margin, most runs scored in an inning, highest fielding percentage (e.g., "The New York Yankees achieved their highest scoring game with a 22-1 victory").*

*Milestones or significant achievements (e.g., "Miguel Cabrera reached 3,000 career hits").*

*Key Match Events:*

*List notable match events, such as:*

*Outstanding performances like pitching a no-hitter, hitting multiple home runs, or making game-saving plays (e.g., "Max Scherzer pitched a perfect game with 20 strikeouts").*

*Turning points including grand slams, critical double plays, clutch hits in the final innings (e.g., "Bryce Harper hit a walk-off grand slam in the ninth inning").*

*High-pressure moments like walk-off home runs, extra-inning thrillers, dramatic strikeouts to end the game (e.g., "The game went into 15 innings before a decisive home run by Mike Trout").*

*Pre-Game Insights:*

*Include pre-match quotes or observations, such as:*

*Predictions or expectations about the match outcome (e.g., "Analysts predicted a close game between the two rival teams").*

*Strategies and preparations discussed by teams or players (e.g., "The manager emphasized the importance of strong bullpen performance").*

*Anticipated key player matchups or rivalries (e.g., "The showdown between Clayton Kershaw and Mookie Betts was highly anticipated").*

*Post-Match Reflections:*

*Summarize post-match comments, including:*

*Emotional reactions from players or managers (e.g., "Aaron Judge expressed pride in the team's resilience after the comeback win").*

*Reflections on team journeys or tournament outcomes (e.g., "The coach highlighted the victory as a pivotal moment in their playoff push").*

*Announcements such as player retirements or long-term impacts (e.g., "Albert Pujols announced his retirement at the end of the season").*

*Miscellaneous Highlights:*

*Include any other significant mentions, such as:*

*Weather or field conditions affecting the match (e.g., "Rain delays led to the game being postponed until the next day").*

*Historical comparisons or records beyond match performance (e.g., "The team ended a 20-year championship drought").*

*Notable head-to-head statistics or unique cultural aspects (e.g., "This was the 100th meeting between the two teams").*

*Others:List any other details or significant mentions that do not fall into the categories above.*

*Output Format:Return a JSON object with each category as a key and all insights listed as values in a flat list.*

*Do not return JSON markdown like (```json) or any other formatting with the response.*

*Example Output:*

*{"Relevancy": ["Relevant"],*

*"New Records": ["Record 1", "Record 2"],*

*"Key Match Events": ["Event 1", "Event 2"],*

*"Pre-Game Insights": ["Insight 1", "Insight 2"],*

*"Post-Match Reflections": ["Reflection 1", "Reflection 2"],*

*"Miscellaneous Highlights": ["Highlight 1", "Highlight 2"],*

*"Others": ["Other detail 1", "Other detail 2"]}*

*If the article is not relevant, return:{"Relevancy": ["Irrelevant"]}*

*Notes:*

*Use only the data provided in the article for your analysis.*

*Ensure insights are contextually relevant, well-written, and avoid redundancy.*

*Structure the insights into complete, meaningful sentences.*

*Return empty lists for categories with no relevant insights.*

*Strictly do not return anything except the JSON object in the response.*

Figure 6: Structured prompt framework used for extracting detailed insights from baseball sports articles, including relevance determination and categorization of match-related highlights such as player records, game events, strategic insights, and post-game reflections.

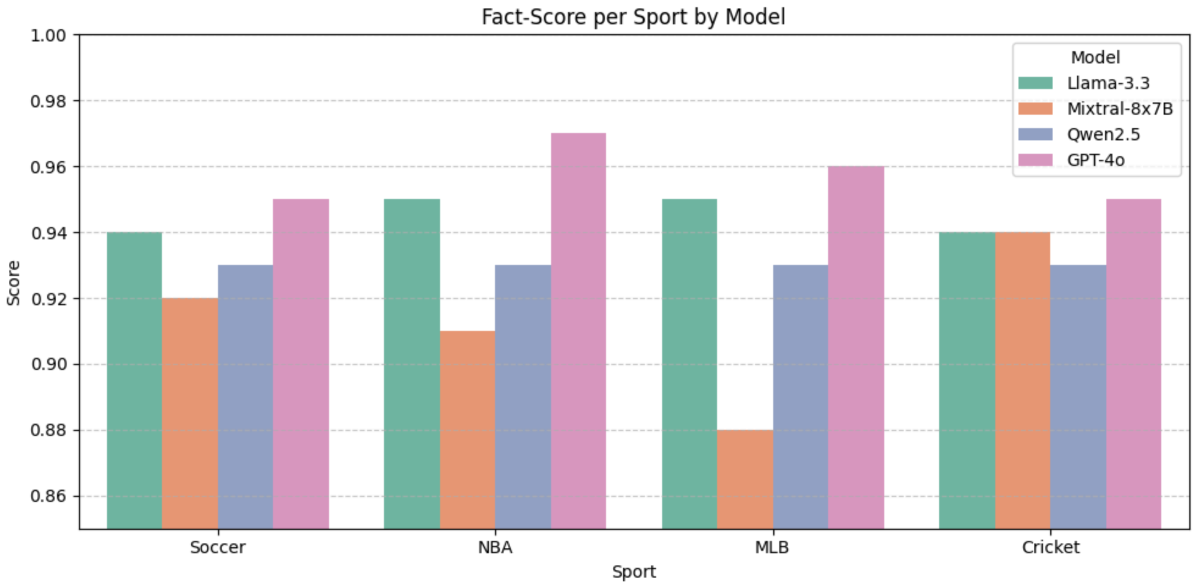


Figure 7: Comparative visualization of hallucination scores across different language models, evaluated using the Fact-Score metric to assess factual consistency in generated outputs.

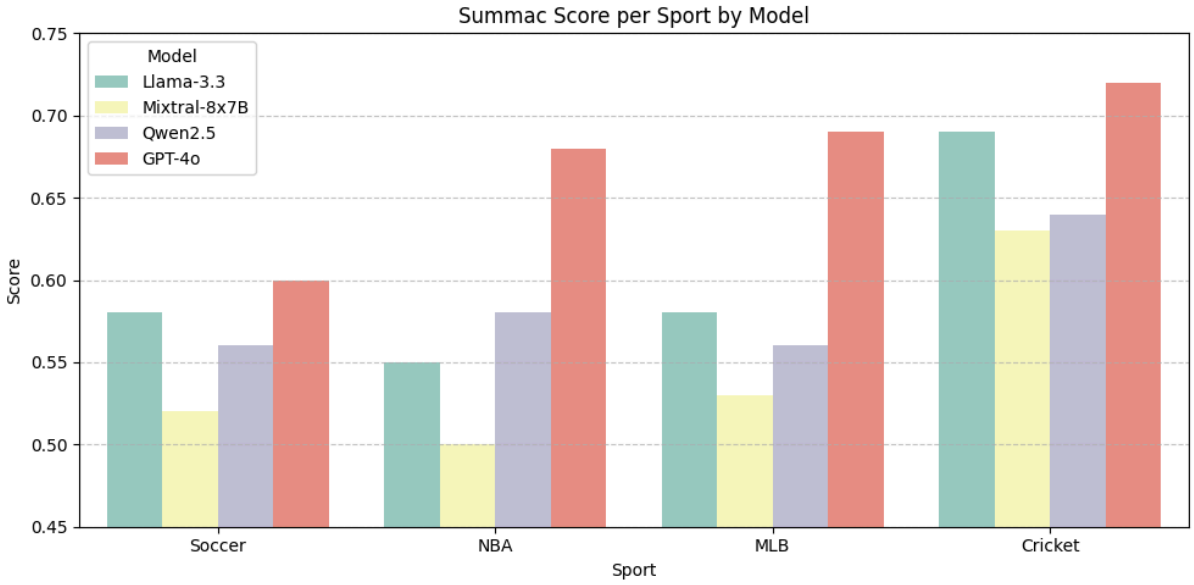
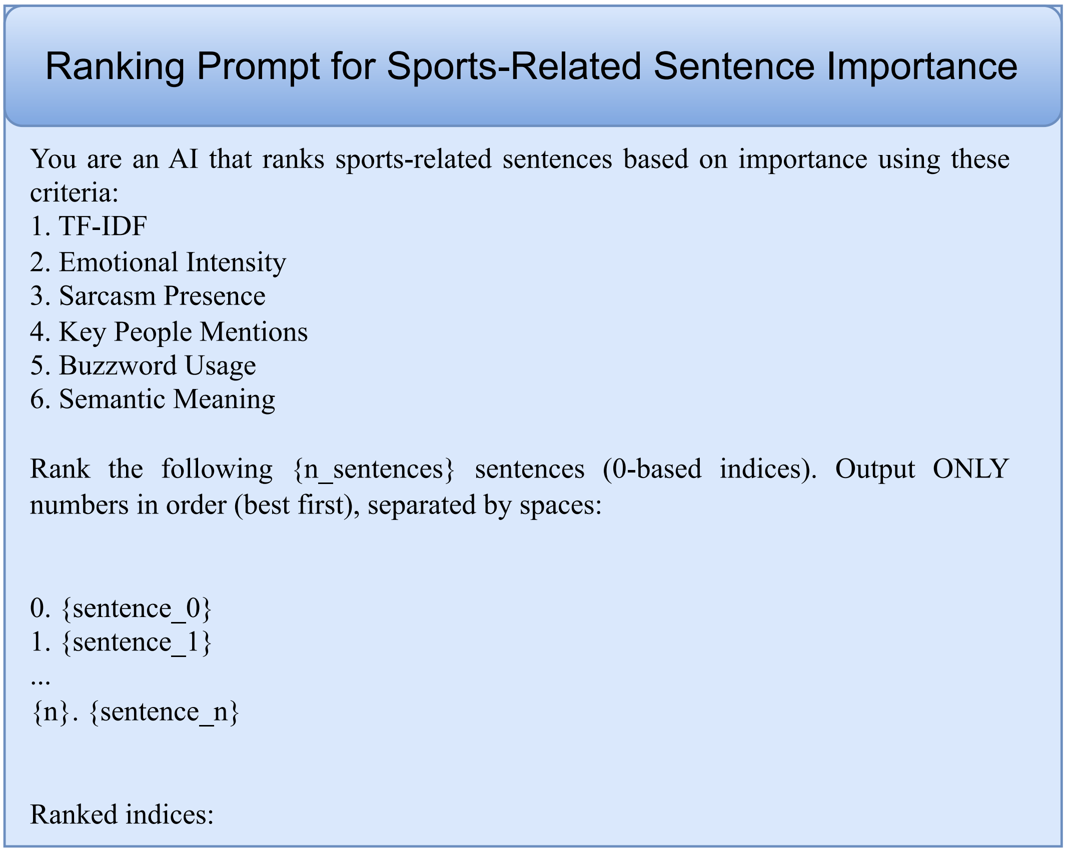


Figure 8: Comparative visualization of hallucination scores across multiple language models, evaluated using the Summac-Score metric to measure summary faithfulness and factual alignment.mani



Ranking Prompt for Sports-Related Sentence Importance

You are an AI that ranks sports-related sentences based on importance using these criteria:

1. TF-IDF
2. Emotional Intensity
3. Sarcasm Presence
4. Key People Mentions
5. Buzzword Usage
6. Semantic Meaning

Rank the following {n_sentences} sentences (0-based indices). Output ONLY numbers in order (best first), separated by spaces:

0. {sentence_0}<br>
1. {sentence_1}<br>
...<br>
{n}. {sentence_n}

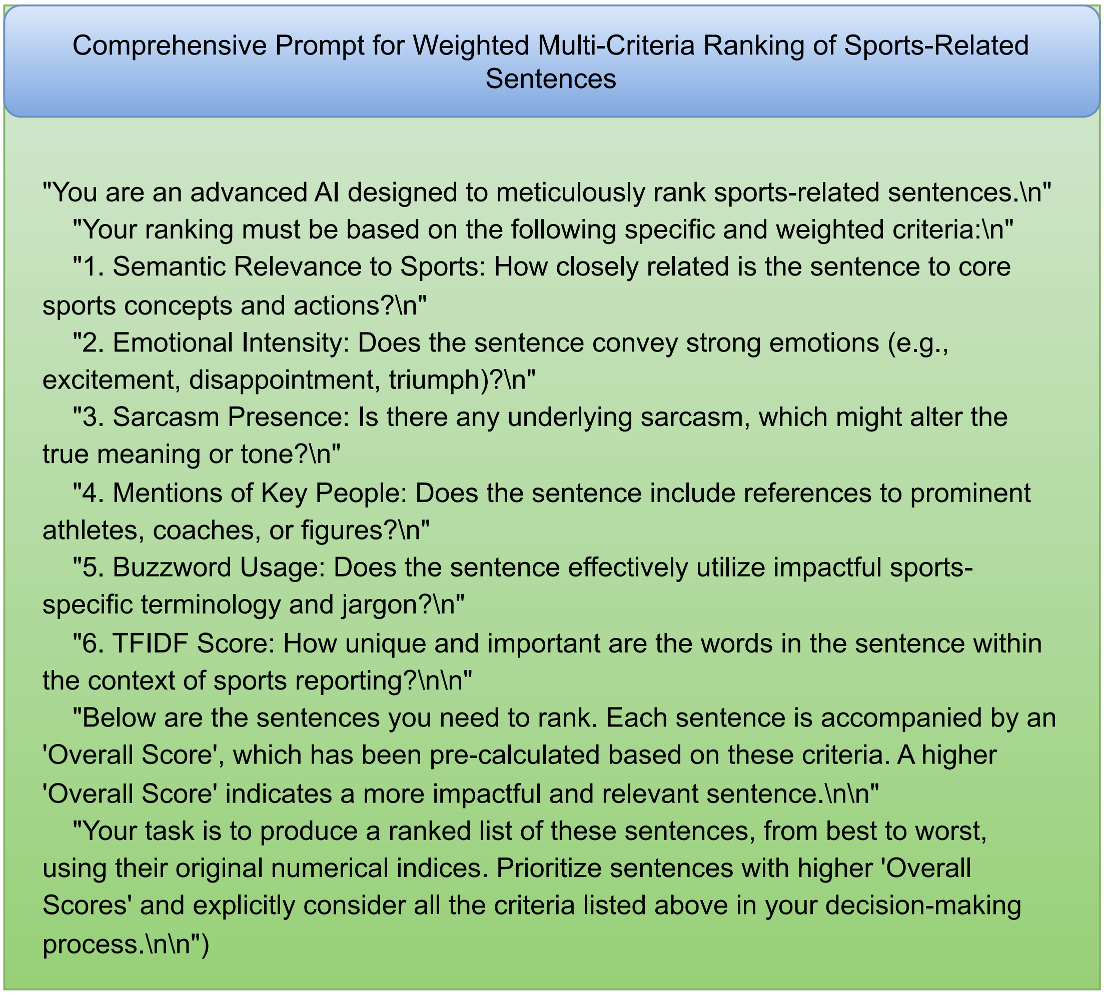
Ranked indices:

Figure 9: Structured prompt template used for sports-related insight ranking. The system instructs an AI to rank multiple candidate sentences by importance using five key criteria: (1) Sports relevance, (2) Emotional intensity, (3) Sarcasm detection, (4) Key player/entity mentions, and (5) Buzzword usage. This prompt enables consistent ranking of extracted insights from articles.

**Illustrative Datapoint from the Gold-Ranked Training Set**

{"candidates":
["Eoin Morgan commented on England's underperformance, suggesting something is unsettled within the team.", "Chris Woakes expressed disappointment and mentioned the lack of confidence in the team.", "Jos Buttler acknowledged the team's poor performance and the need to play for Champions Trophy qualification.", "Jasprit Bumrah mentioned the team's enjoyment of the tournament and their practice in evening conditions.", "Rohit Sharma, adjudged the player of the match, emphasized the importance of his partnership with KL Rahul and taking the game deep.", "Rohit Sharma felt India were 30-40 runs short but was pleased with the overall performance.", "India's victory will galvanize the team as they head into the business end of the tournament.", "England's captain Jos Buttler will have much to contemplate as their chances of progression look even more scant.", "Rohit Sharma was named Player of the Match for his 87 runs.", "India earned 2 points from the match.", "Rohit Sharma expressed pride in the team's character and experience, noting the challenge of batting first and the effectiveness of their bowling attack.", "Jos Buttler expressed disappointment in England's performance, particularly their batting, despite a good start by the bowlers.", "India captain Rohit Sharma: 'This was a game that showed a lot of character in our squad. When times were tough, our experienced players stood up at the right time and fought for us.'", "England captain Jos Buttler: 'Very disappointing. At the halfway stage chasing 230 we fancied ourselves. But it's the same old story.'", "Player of the match Rohit Sharma: 'Looking at where we were after the first 10 overs of our batting, it was important to put on a partnership like myself and KL Rahul did. It was a challenging pitch to start with but it got easier the longer you spent in the middle. We are very happy with the performance.'"],
"ranking": [12, 11, 10, 15, 8, 9, 5, 6, 14, 13, 4, 3, 2, 1, 7]}

Figure 10: Illustrative example from the gold-ranked training set. The list shows 15 candidate highlight sentences from a sports commentary scenario. These sentences include post-match reflections by players (e.g., Rohit Sharma, Jos Buttler), performance summaries, and factual outcomes. The ground-truth ranking on the right orders the candidates by perceived relevance or salience, with lower ranks indicating higher importance. This example highlights the nature of the ranking task and the complexity of modeling both subjective judgments (like emotional emphasis) and objective details (like scores and awards).



Comprehensive Prompt for Weighted Multi-Criteria Ranking of Sports-Related Sentences

"You are an advanced AI designed to meticulously rank sports-related sentences.\n"

"Your ranking must be based on the following specific and weighted criteria:\n"

"1. Semantic Relevance to Sports: How closely related is the sentence to core sports concepts and actions?\n"

"2. Emotional Intensity: Does the sentence convey strong emotions (e.g., excitement, disappointment, triumph)?\n"

"3. Sarcasm Presence: Is there any underlying sarcasm, which might alter the true meaning or tone?\n"

"4. Mentions of Key People: Does the sentence include references to prominent athletes, coaches, or figures?\n"

"5. Buzzword Usage: Does the sentence effectively utilize impactful sports-specific terminology and jargon?\n"

"6. TFIDF Score: How unique and important are the words in the sentence within the context of sports reporting?\n\n"

"Below are the sentences you need to rank. Each sentence is accompanied by an 'Overall Score', which has been pre-calculated based on these criteria. A higher 'Overall Score' indicates a more impactful and relevant sentence.\n\n"

"Your task is to produce a ranked list of these sentences, from best to worst, using their original numerical indices. Prioritize sentences with higher 'Overall Scores' and explicitly consider all the criteria listed above in your decision-making process.\n\n")

Figure 11: Comprehensive training-time prompt used to guide the AI in ranking sports-related sentences. The prompt explicitly instructs the model to consider six weighted criteria—semantic relevance, emotional intensity, sarcasm detection, key people mentions, buzzword usage, and TF-IDF importance—alongside precomputed overall scores. This structured input enables consistent alignment with training objectives and facilitates reinforcement learning using reward signals derived from these criteria.