A Writer–Editor–Critic Pipeline with Deterministic Fact Verification for Football Match Report Generation

Luís Relvas[1], Sérgio Nunes[1], André Restivo[2]
[1]INESC TEC, Faculdade de Engenharia, Universidade do Porto
[2]LIACC, Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

Abstract

Generating football match reports from structured data requires both factual accuracy and fluent prose: template-based systems favor accuracy, while large language models favor fluency but risk introducing unsupported content. We present a writer–editor–critic pipeline in which a writer verbalizes source triples as individual statements, an editor composes those statements into a report, and a critic uses an LLM to extract the report's facts for deterministic comparison with the source triples. Precision, recall, F1, and a structured error report then drive a Reflexion-style loop for up to five rounds or until F1 reaches 100%. We compare atomic encoding, which represents each attribute as a separate triple, with hyper-relational encoding, which groups related attributes as qualifiers on a compound triple. On a dataset of 50 *Liga Portugal* matches, the loop converges for every article within four of the five allowed rounds, averaging 0.84 correction rounds for hyper-relational encoding and 1.12 for atomic encoding, with no unsupported content detected by this metric. Hyper-relational encoding reduces the mean triple count from 187 to 68 and the mean input payload from 3,195 to 1,853 tokens. CheckEval scores show positive associations with human preferences for fluency and naturalness, but not coherence. On a CheckEval-stratified diagnostic subset of 13 matches, annotators prefer reports generated from both representations over the deployed template baseline in at least 81% of comparisons; hyper-relational is preferred to atomic in 64% of coherence and naturalness comparisons.

1 Introduction

Automated match reports can increase coverage of competitions that receive limited dedicated media attention. For such fixtures, the alternative to a generated report is not a human-written report, but no report at all. This context motivates us to explore approaches for generating comprehensive match reports from existing match data.

The *zerozero.pt* platform uses Prosebot (Fernandes et al., 2024), a deterministic template system for this task. Templates provide factual reliability but impose rigid, repetitive prose. LLMs offer fluency but can introduce plausible details that are not grounded in the source data. Reliable generation therefore requires factual claims to be grounded in the structured source data.

We present a writer–editor–critic pipeline: the writer verbalizes source triples, the editor composes a journalistic narrative, and the critic extracts facts from that narrative for source comparison. The key difference from prior self-correction approaches is the correction signal. Self-Refine (Madaan et al., 2023) and multi-agent D2T (Osuji et al., 2025b) use LLM-based error judgment; in our pipeline, the LLM extracts article facts but does not judge their correctness. Deterministic matching and scoring compare the extracted facts with the source triples and return precision, recall, F1, and a structured error report to drive a Reflexion-style loop (Shinn et al., 2023).

We evaluate two variants of the generation pipeline that differ only in how the source data are represented. In the atomic variant, each event attribute is encoded as a separate subject–predicate–object triple. In the hyper-relational variant, attributes belonging to the same event are encoded as qualifiers on a compound triple. Both representations are constructed from the same source information, but expose it to the editor at different levels of granularity.

This difference changes both the number of units that must be integrated during generation and the form of the feedback produced during correction. We therefore ask whether representation still matters once both variants achieve metric $\mathrm{F1} = 100\%$ under representation-specific matching.

Our contributions are:

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 1327–1347
October 17-21, 2026 ©2026 Association for Computational Linguistics

- a **writer–editor–critic pipeline** combining LLM fact extraction with deterministic matching and scoring, replacing LLM-based error judgment with a deterministic correction signal;
- an **atomic–hyper-relational comparison** under the same generation pipeline, measuring token overhead in each representation's native format and surface quality after self-correction converges;
- a **three-layer evaluation** combining triple matching, CheckEval (Lee et al., 2025), and pairwise preference, with alignment analysis between automated and human judgments; and
- a **parallel reference dataset** comprising atomic and hyper-relational annotations of 50 *Liga Portugal* matches for European Portuguese sports D2T (Relvas et al., 2026).

The remainder of the paper is organized as follows. Related work on data-to-text generation and self-correction is reviewed in Section 2. Section 3 describes the generation architecture and representation variants, Section 4 presents the evaluation framework, and Section 5 reports the experimental results. The main findings are summarized in Section 6, followed by a discussion of limitations.

2 Related Work

Data-to-text generation for sports. The *Rotowire* benchmark pairs NBA box scores with professional summaries and exposes the difficulty of jointly optimizing content selection and surface realization in sports D2T (Wiseman et al., 2017). Later systems separated these tasks through modular pipelines and explicit content planning (Puduppully et al., 2019). Football deployments still retain templates for factual reliability at the cost of stylistic flexibility (Gatt and Krahmer, 2018); Prosebot (Fernandes et al., 2024), our production baseline, follows this approach. Recent benchmarks compare LLMs, rule-based systems, and human-written texts on long triple-set inputs using LLM-based evaluation (Osuji et al., 2025a).

Triple representations for structured input. D2T work often treats attribute–value and RDF inputs as alternative syntactic forms with little effect on performance (Lin et al., 2024). Hyper-relational graphs expose a different choice: attach co-occurring event context as inline qualifiers to a base triple instead of using reified intermediate nodes (Galkin et al., 2020). This representation has mainly been studied for link prediction. To our knowledge, its downstream effect on D2T surface realization has not been quantified.

Hallucination and faithfulness. Factual unfaithfulness persists in neural generation from structured input: models exhibit intrinsic and extrinsic hallucinations, and confidence correlates poorly with factuality (Maynez et al., 2020). Source-grounded D2T metrics have compared extracted relations with input records (Wiseman et al., 2017); instead, we feed deterministic match labels back as correction signals, complementing constrained generation and post-generation verification (Schmidtová et al., 2024).

Iterative self-correction. Iterative refinement of an initial draft has a precedent in D2T: templates verbalize the data, and a trained editing model then fuses and improves the result (Kasner and Dušek, 2020). Self-Refine uses internal model critique without an external signal (Madaan et al., 2023), limiting correction when the model misses its own errors. Reflexion instead converts environmental feedback into natural-language critiques held in episodic memory (Shinn et al., 2023). Multi-agent D2T extends this loop with LLM guardrails (Osuji et al., 2025b), which may share the generator's blind spots. We retain an LLM extractor but replace guardrail judgment with deterministic triple matching that returns precision, recall, and F1 against the source at every iteration. Deterministic verification signals have been used before: VCP checks keyword presence with a fixed rule and converts the result into an error-indication prompt for regeneration (Ren and Liu, 2025). Our signal operates at the triple rather than at the token level and separates missing, partial, contradicting, and unsupported facts. The resulting error report tells the editor which repair to apply.

Evaluation of narrative generation quality. BLEU and ROUGE correlate poorly with human D2T judgments (Wiseman et al., 2017; Schmidtová et al., 2024). CheckEval decomposes quality dimensions into LLM-evaluated Boolean checklists (Lee et al., 2025), while comparative judgments use forced choices to reduce annotator variance (Kiritchenko and Mohammad, 2017). Our protocol presents two same-match texts and records

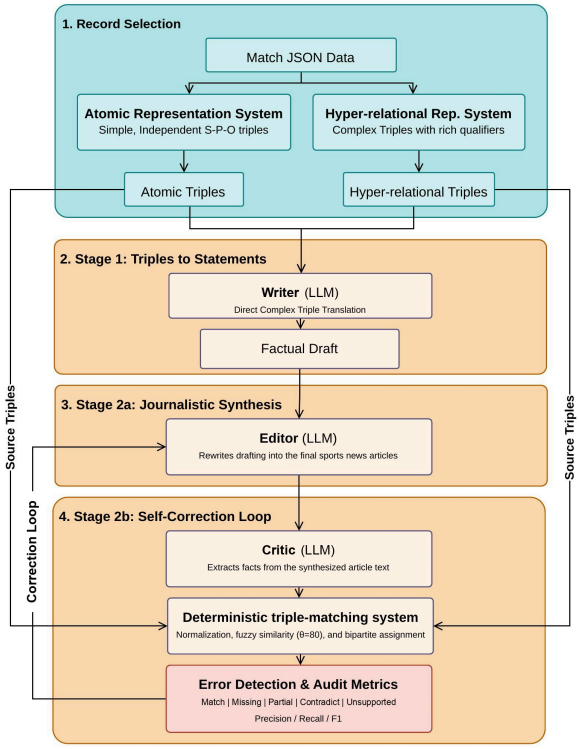


Figure 1: Writer–editor–critic pipeline. AMALIA verbalizes individual logical units; `gemini-2.5-flash` composes the report; and `gemini-2.5-pro` extracts facts for deterministic matching and iterative correction.

the preferred one, making it a pairwise preference rather than a multi-item Best–Worst Scaling. General-domain LLM-as-Judge findings do not establish reliability for European Portuguese sports journalism, so we report automated and human signals and their agreement.

3 System Architecture

Figure 1 shows the pipeline from selected records to a European Portuguese report. A record-selection stage chooses which records of the match feed to include without ranking facts by newsworthiness; every selected record is expected to appear in the output. Three model roles handle surface realization: a writer verbalizes each logical unit, an editor composes the statements into a coherent article, and a critic extracts facts from the article. Deterministic normalization, assignment, matching, and scoring, then compare extracted and source triples to drive correction.

3.1 Triple Representation Schemes

Atomic encoding represents each event attribute as a separate subject–predicate–object triple. A goal yields scorer, minute, assist, origin, zone, and body-part assertions, averaging 186.9 triples per match. This preserves predicate granularity at the cost of more independent units. One goal is encoded as:[1]

```
[TRIPLE] Goal_3 minute 71 [/TRIPLE];
[TRIPLE] Goal_3 scorer Evanilson [/TRIPLE];
[TRIPLE] Goal_3 assisted_by Pepê [/TRIPLE];
[TRIPLE] Goal_3 origin open_play [/TRIPLE];
[TRIPLE] Goal_3 zone
                inside_penalty_area [/TRIPLE];
[TRIPLE] Goal_3 body_part right_foot [/TRIPLE].
```

Hyper-relational encoding retains a base relation and consolidates related attributes as inline qualifiers, reducing the mean to 67.5 triples. In the deployed configuration, the mean input payloads are 3,195 tokens for atomic and 1,853 for hyper-relational encoding. The same goal becomes:

```
[TRIPLE] Evanilson scored_goal minute_71
          | zone:inside_penalty_area
          | body_part:right_foot
          | origin:open_play
          | assisted_by:Pepê [/TRIPLE].
```

Both encodings use the same source attributes but organize them differently. A compound triple bundles several attributes counted separately by the atomic scheme; raw counts therefore measure representation-specific granularity and workload, not common semantic facts. This distinction also affects correction: at Stage 2b, attributes represented as separate atomic triples must be reinserted individually, whereas one compound hyper-relational unit can restore an entire event.

3.2 Stage 1: Chunked Triple-to-Sentence Realization

Stage 1 converts each logical unit into a European Portuguese sentence in isolation. AMALIA (Simplício et al., 2026), a Portuguese-specialized 9B model, verbalizes one atomic or compound unit per call at $T = 0.1$. Chunk size remains one logical unit because two-unit calls produced qualifier crossover in pilot runs when adjacent triples shared predicates. Dynamic predicate retrieval supplies demonstrations for exactly the relations in that unit (Appendices B.1 and A), limiting interference between adjacent events. Qwen2.5-7B (Qwen et al., 2025) serves as a similarly sized control for Stage 1 fidelity. The output is locally factual but globally unstructured.

[1]The experiment used European Portuguese schemas, prompts, checklists, identifiers, and outputs; all such material shown in this English paper is author-translated rather than literal runtime text.

3.3 Stage 2a: Cohesive Rewrite

The `gemini-2.5-flash` editor rewrites the full Stage 1 draft into a title, subtitle, short summary, introduction, chronological body, lineups, and final debriefing/statistics paragraph (Appendix B.2). Its prompt requires every draft fact and forbids additions. The editor sees no source triples at this stage: the draft is its only source of truth, and omissions from dense inputs motivate correction.

3.4 Stage 2b: Self-Correction with Deterministic Matching

Stage 2b runs a Reflexion-style loop (Shinn et al., 2023) driven by deterministic matching and scoring (§4.2). At each iteration, `gemini-2.5-pro` extracts article triples in a single pass, and a deterministic post-processor normalizes them. The matcher reports precision, recall, and F1 and enumerates missing, partial, contradicting, or unsupported triples. Each category maps to one repair: add an omitted fact, complete missing qualifiers, correct a conflicting value, or remove unsupported content. Given the source triples, current article, and structured error report, the editor applies targeted corrections at the appropriate chronological positions (Appendix B.4). Matching and scoring are therefore deterministic, conditional on the LLM extraction step (§4.1).

A three-message critique memory carries persistent errors across iterations. The loop stops at zero detected errors or five rounds, retaining the highest-F1 article if it does not converge. Appendix F.2 illustrates one correction round in full.

4 Evaluation Framework

We evaluate factual coverage separately from prose quality. Triple matching quantifies coverage against source triples; CheckEval scores surface quality; pairwise preference anchors those dimensions to readers; and an alignment analysis compares automated and human signals at the segment and system level. Section 4.1 distinguishes the human-report annotations used to validate extraction from the source triples used to generate and score outputs.

Each evaluation layer has its own limitations. Triple matching is closed-loop: the same extraction and matching pipeline supplies the correction signal and the final score. It measures only facts recovered by the critic, so extraction misses are invisible. CheckEval reads the article outside that loop, but it is an LLM judge, instantiated with the same model as the critic and not validated for European Portuguese sports journalism. Pairwise preference is the only evaluation layer based on human judgments. Annotators saw no source data and judged only fluency, coherence, and naturalness, so no factuality check was added. No layer independently covers both factual accuracy and prose quality; §5.6 reports where they disagree.

4.1 Reference Triple Dataset and Extraction Validation

We distinguish two reference objects. G_{human} contains facts annotated from 50 professional reports and is used to estimate extractor accuracy. It serves only to validate extraction; no system is asked to reproduce it. T_{source} contains record-selection triples supplied to generation; self-correction and output scoring compare extracted article triples with this set, not G_{human}.

A single annotator created atomic G_{human} labels using a standardized guide; deterministic conversion, not independent annotation, produced the parallel hyper-relational version. Journalists select only part of the feed, so these sets average 47.6 atomic units (2,380 total) and 31.4 hyper-relational units (1,568), versus 186.9 atomic and 67.5 hyper-relational source units. Scored against G_{human}, record selection recovers 87.7% of annotated hyper-relational units and 82.0% of atomic units, at 35.9% and 18.6% precision. Because the selector supplies the full record while journalists report only a subset, high recall with low precision is expected rather than evidence of selection failure. Since a compound unit can encode several atomic attributes, neither these counts nor the raw missing-unit counts are expressed in a shared semantic space. Roughly 88–92% of annotations are directly derivable from structured JSON; the rest include observations absent from the feed, such as untimed attempts or tactical descriptions (Appendix D).

Against G_{human}, hyper-relational and atomic extraction reach 88.4% and 82.3% F1. These are observed accuracies on unconstrained professional prose, not extractor upper bounds. The gap between these accuracies and the near-perfect ($\text{F1} > 99.7\%$) recovery on formulaic Stage 1 drafts shows that extraction accuracy depends on the prose regime. Stage 1 fidelity alone does not validate extraction from final articles.

4.2 Triple-Matching Metric

The deterministic triple-matching system measures how closely extracted triples $\hat{T}$ match T_{source}. Each extracted triple is assigned to at most one source triple using the Hungarian algorithm over a pairwise string-similarity matrix (Kuhn, 1955). This avoids greedy assignment for that objective but optimizes similarity rather than the number of pairs that ultimately pass all validity checks (Appendix C.3 gives the cell-level computation).

Strings are lowercased and stripped of diacritics, domain aliases are resolved, and numeric event indices are retained to distinguish events. Exact or token-subset pairs score 100; other pairs use RapidFuzz edit similarity over sorted, deduplicated token bags (RapidFuzz, 2026).

A pair is a **Match** only above $\theta = 80\%$ with complete, consistent qualifiers. Pairs with missing qualifiers are **Partial**, those with conflicting values are **Contradict**, and unmatched sources are **Missing**; a secondary pass recovers contradictions that would otherwise be labeled missing. Partial applies only to the hyper-relational format, since atomic triples carry no qualifiers. Only Match contributes to the precision and recall numerators. Partial and Contradict therefore reduce precision, recall, and F1, but remain distinct from unsupported assertions in the hallucination rate. An extracted triple unmatched to any source is unsupported; a partially expressed or contradictory source fact is not reclassified as an unrelated invention.

Two design choices limit this objective. Order-insensitive token bags can overscore role-reversed asymmetric relations. Qualifiers are checked only after assignment, so an invalid high-similarity edge can displace a lower-scoring valid one. No order-sensitive, schema-aware admissible-edge rematching was run.

Precision, recall, and F1 are computed per match, with M the number of **Match** outcomes:

$$\text{P} = \frac{M}{|\hat{T}|}, \quad \text{R} = \frac{M}{|T_{\text{source}}|}, \quad \text{F1} = \frac{2PR}{P+R}.$$

Scores are expressed as percentages and averaged by match.

4.3 CheckEval: Checklist-Based Quality Assessment

CheckEval (Lee et al., 2025) decomposes each quality dimension into a checklist of binary questions evaluated by an LLM judge. We instantiate it using European Portuguese football checklists to assess faithfulness, factual consistency, fluency, coherence, and naturalness. Faithfulness tests unsupported entities and statistics; factual consistency tests attribution and temporal ordering; fluency tests grammatical correctness and readability; coherence tests discourse organization; and naturalness tests lexical variety and resemblance to human prose. The first two dimensions receive source triples, while the last three receive only the article. We use `gemini-2.5-pro` at $T = 0.0$, once per article and dimension; prompts contain no system identifier. Each score is the proportion of affirmative answers over the dimension's checklist, which contains 8 items for faithfulness and factual consistency, 16 for fluency, and 13 for coherence and naturalness (Appendix E). Questions are binary; their aggregate yields a graded score.

Match-paired t-tests compare outputs from the same match, reducing between-match variation in content volume and event complexity. Prosebot comparisons use the same design on a 13-match subset of Prosebot-derived content.

4.4 Human Evaluation: Pairwise Preference

The diagnostic sample contains 39 texts, one per system, for 13 matches. Stratification on the mean CheckEval score across the two neural configurations selects the three highest- and three lowest-scoring matches, four interquartile cases, and three with the largest between-configuration divergence. The last group targets cases where the configurations disagree most. Because selection deliberately covers extremes and divergences, it does not estimate population preference over all 50 matches.

The three systems produce 39 unique same-match text pairs. We collected 163 comparison records across these pairs, each recording preferences for fluency, coherence, and naturalness; every pair received at least three records. Annotators were recruited by direct invitation, informed of the study's purpose, and free to abstain from any comparison. Reliability uses Krippendorff's α (Hayes and Krippendorff, 2007).

Comparing texts for the same match keeps the underlying source information approximately constant within each judgment. The protocol records preference rather than an absolute rating, reducing scale-calibration differences between annotators, but repeated observations within matches and annotators introduce statistical dependence.

4.5 Automated–Human Alignment

Within the same selected subset, segment-level ($n = 39$) alignment pairs each article's dimension-specific CheckEval score with its preference win rate. System-level alignment ($n = 3$) uses system means only to compare aggregate rankings. Both use Spearman's ρ and Kendall's τ_b; 95% percentile intervals come from 200 bootstrap resamples. CheckEval-based selection makes these associations diagnostic rather than unbiased population estimates.

5 Results

Both representations reach F1 $= 100\%$ under the extraction-and-matching metric. Both neural configurations are preferred over Prosebot in the diagnostic subset. On the full 50-match set, CheckEval shows a descriptive 3.4-point coherence difference favoring hyper-relational encoding over atomic.

5.1 Stage 1: Triple-to-Statement Fidelity

Across four writer–representation configurations ($n = 50$ each), Stage 1 F1 ranges from 99.7% to 99.9%, with negligible hallucination. AMALIA scores 99.8% in both formats, while Qwen2.5-7B scores 99.9% with hyper-relational input and 99.7% with atomic input. These near-identical scores make Stage 1 unlikely to explain larger downstream differences. However, the design does not independently vary the model, prompting, and temperature.

5.2 Self-Correction Convergence

At round 0, the cohesive rewrite already reaches mean F1 scores of 97.83% with hyper-relational input and 98.14% with atomic input (Table 1). Unsupported units are rare (0.06%/0.08% hallucination rates). Error categories differ by representation. The hyper-relational representation has more Partial than Missing or Contradict units, whereas the atomic representation has more Missing than Contradict units. Atomic triples carry no qualifiers, so Partial is undefined for that format, and its count is zero by construction. Since hyper-relational units bundle attributes counted separately by the atomic representation, these counts describe correction workload in each representation's native space, not effects on common semantic facts.

The round distribution supports this workload interpretation. The hyper-relational configuration begins with complete measured coverage more often and has fewer cases requiring two or more rounds.

Table 1: Self-correction loop convergence statistics ($n = 50$). Round 0 denotes the initial cohesive rewrite before any correction iteration. Hallucination rate is the proportion of extracted triples with no support in the source triple set. Partial is defined only for the hyper-relational format.

Metric	Hyper-Rel.	Atomic
Initial rewrite (round 0)		
Mean F1 (%)	97.83 ± 2.73	98.14 ± 3.39
Minimum F1 (%)	88.14	85.13
Mean hallucination rate (%)	0.06	0.08
Mean missing triples	0.26	2.78
Mean partial triples	0.82	0.00
Mean contradict triples	0.50	1.94
Convergence to metric F1 $= 100\%$		
Round 0	17/50 (34%)	11/50 (22%)
Round 1	29/50 (58%)	28/50 (56%)
Round ≥ 2	4/50 (8%)	11/50 (22%)
Mean rounds per match	0.84	1.12
Maximum rounds used	4	4
Matches not converged	0/50	0/50
Final article		
Mean F1 (%)	100.0 ± 0.0	100.0 ± 0.0
Mean hallucination rate (%)	0.00	0.00

This does not show that one representation preserves more semantic facts because the formats were not mapped to a shared fact ontology.

Every match reaches metric F1 $= 100\%$ within four of five allowed rounds. The hyper-relational configuration converges at round 0 for 34% of matches and at round 1 for another 58%, averaging 0.84 rounds versus 1.12 for the atomic configuration. Because the same extraction-and-matching pipeline supplies both correction feedback and the final metric, convergence demonstrates successful optimization of this verification signal rather than independent factual completeness. No unsupported triples are detected in the final articles by that signal.

5.3 Automated Quality: Atomic vs. Hyper-Relational

Paired tests find no representation difference in faithfulness ($p = 1.000$), factual consistency ($p = 0.322$), fluency ($p = 0.270$), or naturalness ($p = 0.829$; Table 2).

CheckEval coherence is 3.4 points higher for hyper-relational than for atomic outputs (80.8% vs. 77.4%). The difference has a nominal $p =$

Table 2: CheckEval scores on the 50-match evaluation set ($n = 50$; mean $\pm$ std dev, %).

Dimension	Hyper-Rel.	Atomic
Faithfulness	98.5 ± 4.1	98.5 ± 4.8
Factual Consistency	99.0 ± 3.4	98.2 ± 4.4
Fluency	89.9 ± 11.0	87.5 ± 10.4
Coherence	80.8 ± 11.0	77.4 ± 12.2
Naturalness	43.1 ± 11.8	42.6 ± 11.8

Table 3: CheckEval scores on the 13-match Prosebot-derived content condition (%). Fl. = fluency; Co. = coherence; Na. = naturalness.

System	Fl.	Co.	Na.
Hyper-Rel. Pipeline	97.6	89.9	89.3
Atomic Pipeline	97.1	92.3	79.9
Prosebot	83.7	84.6	68.0

0.049 ($t(49) = 2.019$), but does not survive Holm correction across five dimensions (adjusted $p = 0.245$), so we treat it as descriptive. CheckEval faithfulness and factual consistency also remain below 100% after metric convergence, indicating that the two layers are not interchangeable.

Naturalness is low in both formats (43.1%/42.6%), consistent with the factuality–variation tension in neural D2T (Schmidtová et al., 2024). Low temperature and high event volume may contribute to repeated syntactic frames, but neither was ablated.

5.4 Comparison with the Prosebot Baseline

In the 13-match Prosebot-derived content condition, neural inputs contain facts recovered from each Prosebot report and therefore approximately match its content scope. Because these triples are extracted from Prosebot's output rather than sourced from the raw match feed, extractor omissions may leave the neural configurations with strictly fewer facts than Prosebot expressed. Both configurations reach metric F1 = 100%. We therefore use Table 3 to compare surface quality rather than factual coverage. Table 4 shows excerpts for one match in this condition, and Appendix F.1 gives the three full articles.

Both configurations improve fluency over Prosebot ($p < 0.001$ hyper-relational; $p = 0.012$ atomic) and do not differ significantly from each other. Coherence differences are nonsignificant. Hyper-relational naturalness (89.3%) exceeds atomic naturalness (79.9%) and Prosebot naturalness (68.0%). The difference is significant relative to Prosebot ($t = 3.860$, $p = 0.002$), but not to atomic ($t = 1.891$, $p = 0.083$). Consolidated attributes and the subset's lower content volume are plausible but untested explanations. The excerpt illustrates these differences. The Prosebot article reports the goal and the sending-off in two separate one-sentence paragraphs. Both neural articles combine them into a single passage and explicitly mark the time gap. A single example cannot, however, corroborate the differences in scores.

Because the Prosebot-derived condition contains less information than the full 50-match setting, its substantially higher naturalness scores should not be interpreted as a direct representation-only effect. Content volume, representation, and mandatory article structure are not independently ablated.

5.5 Human Evaluation (Pairwise Preference)

Table 5 reports the diagnostic subset's pairwise preferences. Reliability is moderate and consistent ($\alpha = 0.552$–0.585; agreement near 71%). Both neural configurations are preferred over Prosebot in at least 81.1% of comparisons across all dimensions. Hyper-relational is preferred over atomic in 58.5% of fluency comparisons and 64.2% of coherence and naturalness comparisons. These selected-subset rates are descriptive; we report neither confidence intervals nor a clustered model accounting for repeated observations within matches and annotators.

5.6 Automated–Human Agreement

Figure 2 compares CheckEval and pairwise-preference scores in the selected subset. Segment-level association is positive for fluency ($\rho = 0.476$, $p = 0.002$) and naturalness ($\rho = 0.385$, $p = 0.016$), but not coherence ($\rho = 0.106$, $p = 0.519$); Kendall's τ_b gives the same pattern. Because CheckEval informed selection, these values characterize diagnostic cases rather than the full population.

The lack of a coherence correlation does not by itself imply metric failure. Similar coherence and naturalness preference rates (64.2% in the neural head-to-head; 81.1%/86.8% against Prosebot) may reflect a broader readability judgment, although aggregate rates do not establish dimension conflation. Paired dimension-level decisions would be required.

At $n = 3$, the system rankings coincide for fluency and naturalness ($\rho = 1.000$), but this agreement is descriptive. Coherence reverses the order-

Table 4: Body-section excerpt covering the same two events, the 9th-minute goal and the 81st-minute sending-off, under the three systems. Text is verbatim system output, including the comma error in the atomic article. Inputs come from the 13-match Prosebot-derived condition. Appendix F.1 gives the three full articles.

Hyper-relational pipeline	**Atomic pipeline**	**Prosebot**
Original European Portuguese		
O único golo da partida surgiu cedo, ao minuto 9, quando Evanilson, com o pé esquerdo e dentro da área, inaugurou o marcador. Já na fase final do encontro, ao minuto 81, David Carmo, do FC Porto, viu o segundo cartão amarelo e, consequentemente, o cartão vermelho, deixando a sua equipa em inferioridade numérica até ao apito final.	O momento decisivo do encontro aconteceu logo ao minuto 9, quando Evanilson, no interior da área, finalizou com o pé esquerdo e inaugurou o marcador. Aos 81 minutos, a equipa da casa ficou reduzida a dez elementos, depois de David Carmo ter visto o segundo cartão amarelo e, ter sido expulso com cartão vermelho.	Evanilson marcou um golo solitário para o FC Porto ao minuto 9, de dentro da área e com o pé esquerdo. Aos 81 minutos, os Dragões viram David Carmo ser expulso devido à acumulação de amarelos.
Author translation		
The only goal of the match came early, in the 9th minute, when Evanilson, with his left foot and inside the area, opened the scoring. In the closing stage of the match, in the 81st minute, FC Porto's David Carmo received a second yellow card and, consequently, a red card, leaving his team a man down until the final whistle.	The decisive moment came as early as the 9th minute, when Evanilson, inside the area, finished with his left foot and opened the scoring. In the 81st minute, the home side were reduced to ten men, after David Carmo had received a second yellow card and, been sent off with a red card.	Evanilson scored a solitary goal for FC Porto in the 9th minute, from inside the area and with his left foot. In the 81st minute, the Dragons saw David Carmo sent off for accumulating yellow cards.

Table 5: Preferred-system rates from 163 comparison records over 39 unique text pairs in the 13-match diagnostic subset. Fl. = fluency; Co. = coherence; Na. = naturalness.

Comparison	**Fl.**	**Co.**	**Na.**
Hyper-Rel. over Prosebot	88.7%	81.1%	86.8%
Atomic over Prosebot	84.2%	82.5%	86.0%
Hyper-Rel. over Atomic	58.5%	64.2%	64.2%

ing of the two configurations ($\rho = 0.500$). CheckEval scores atomic above hyper-relational (92.3% vs. 89.9%), while humans prefer hyper-relational in 64.2% of comparisons. The aggregate results cannot explain the reversal.

Segment- and system-level analyses answer different questions. Segment-level association asks whether higher-scoring individual articles tend to receive more preference, whereas system-level association compares only three aggregate means and cannot support general inference despite apparently perfect ranks.

6 Conclusion

We presented a writer–editor–critic pipeline that separates fact extraction from error judgment: an LLM extracts article facts, while deterministic matching and scoring identify discrepancies and provide targeted correction feedback to the editor. Across 50 matches, every article reaches $\mathrm{F1} = 100\%$ under this verification signal within four correction rounds. Hyper-relational encoding reduces the mean number of source triples from 186.9 to 67.5 and requires fewer correction rounds on average than atomic encoding (0.84 vs. 1.12). These results show an operational advantage for representing related event attributes together, although the evaluation does not establish cross-representation factual equivalence or statistically significant differences in surface quality.

The evaluation also shows why factual verification and prose-quality assessment should be considered separately. In the selected Prosebot-derived condition, readers prefer both neural configurations over Prosebot in at least 81% of comparisons. CheckEval scores are positively associated with human preferences for fluency and naturalness but not coherence. At the descriptive $n = 3$ level, coherence shows a rank reversal. Together, these findings suggest that deterministic verification can provide an effective corrective signal without making an LLM responsible for judging its own factual accuracy, while human evaluation remains necessary for assessing the quality of the resulting prose. We also release parallel atomic and hyper-relational reference annotations for 50 *Liga Portugal* matches to support further work on European Portuguese sports D2T.

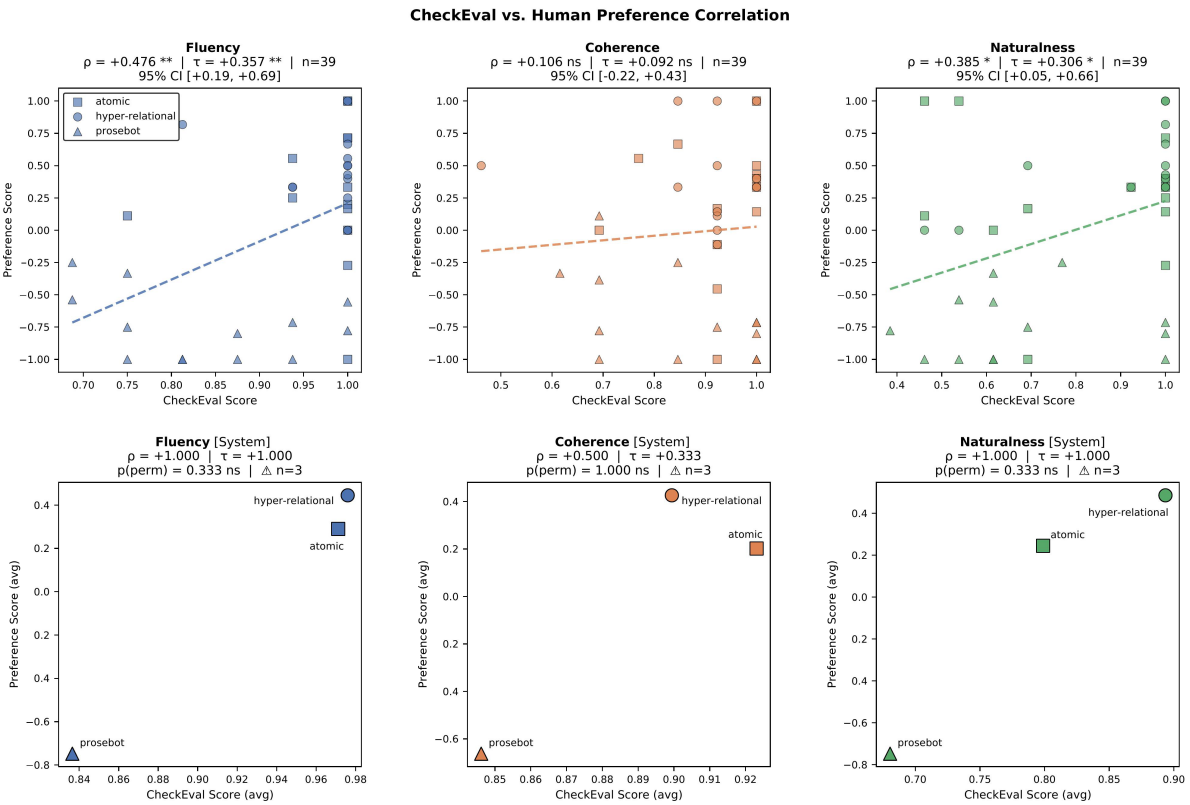


Figure 2: CheckEval–human association in the selected 13-match subset: segment level ($n = 39$, top) and descriptive system level ($n = 3$, bottom), with rank correlations and 95% bootstrap intervals.

Future work should independently audit the factuality of final articles, test whether temperature scheduling improves naturalness, and examine whether the findings generalize across languages, competitions, and generation models.

Limitations

The study is limited to one language, one competition, and 50 matches. Human evaluation uses a CheckEval-stratified diagnostic subset of 13 matches and 39 texts, with 163 comparison records clustered by match and annotator. Annotators judged fluency, coherence, and naturalness without access to source data, so the study provides no independent human check of factuality. Because selection targets score extremes and between-configuration divergences, preferences and correlations do not estimate population-level effects. We report no clustered inference or win-rate intervals and did not retain the number of unique annotators, assignment procedure, presentation order, masking settings, or verbatim interface instructions.

Metric F1 $= 100\%$ is a closed-loop result because the same extraction-and-matching pipeline supplies correction feedback and final scores. Extractor F1 on professional prose is 88.4% for hyper-relational and 82.3% for atomic encoding, and no independent claim-level audit covers final Stage 2 articles. Extraction misses can therefore remain invisible to both correction and final scoring. CheckEval factuality remains below 100%, further showing that the two signals are not interchangeable. The reference annotations also lack inter-annotator agreement: one annotator created the atomic set before deterministic conversion to the hyper-relational format.

Atomic and hyper-relational outputs are evaluated in their respective unit systems, so equal F1 and raw error counts do not establish factual equivalence at a common semantic granularity. Results are also conditional on the post-processing, similarity, assignment, and qualifier rules detailed in Appendix C. The 3.4-point CheckEval coherence difference is descriptive because it does not survive Holm correction.

CheckEval and the critic both use `gemini-2.5-pro`, creating potential model-family dependence between correction and automated evaluation. This may affect faithfulness and factual consistency in particular, although the

judge receives no system identifier, and human preferences independently favor both neural configurations over Prosebot on the dimensions evaluated. CheckEval and human preference nevertheless diverge on coherence. We did not evaluate a judge from another model family or repeated-judge reliability. CheckEval also weights all checklist items equally, regardless of error severity. The editor and critic rely on proprietary Gemini APIs, so changes to service or versions may affect replication.

Finally, the evaluation lacks a single-pass frontier-model baseline and therefore does not establish an advantage of the pipeline over direct generation from source triples at comparable cost. We also do not measure latency or monetary cost against Prosebot; Stage 1 requires one model call per representation unit and therefore trades additional inference workload for linguistic flexibility.

Data Availability

The parallel 50-match reference-triple files (atomic and hyper-relational) used in this work are published under CC BY-NC 4.0 at https://doi.org/10.25747/J6D2-J785 (Relvas et al., 2026). The professionally authored reference reports are not redistributed with the dataset but remain available on the zerozero.pt platform. Direct participant identifiers, generation and evaluation code, CheckEval outputs, and human-evaluation materials are not publicly released.

Ethical Considerations

The match data and professionally authored reference reports are provided under a research collaboration with ZOS Lda., operator of the zerozero.pt platform. Every reference article was written by a professional journalist. The sports corpus contains no sensitive personal data; person-related information is limited to names and professional roles already reported in the sports press.

The pairwise preference study recruited annotators by direct invitation and collected name, email address, age, and professional occupation for post hoc response identification. Annotators were informed of the study's purpose. We make no claim of institutional review or a separately archived consent procedure beyond this information-and-abstention process. The work targets additional coverage of lower-tier fixtures and is framed as extending, rather than replacing, existing journalistic labor.

We used AI-assisted tooling in the preparation of this manuscript. All experimental design, data collection, analysis, and empirical claims are the authors' own, and every quantitative result reported was verified against the underlying data.

Acknowledgments

This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the support UID/50014/2025 (https://doi.org/10.54499/UID/50014/2025). We thank ZOS Lda., operator of the zerozero.pt platform, for providing the match data and the professionally authored reference reports used in this work.

References

Pedro Fernandes, Sérgio Nunes, and Luís Santos. 2024. A community-driven data-to-text platform for football match summaries. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 164–173, Torino, Italia. ELRA and ICCL.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message passing for hyper-relational knowledge graphs. *Preprint*, arXiv:2009.10847.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170.

Andrew F. Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1):77–89.

Zdeněk Kasner and Ondřej Dušek. 2020. Data-to-text generation with iterative text editing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 60–67, Dublin, Ireland. Association for Computational Linguistics.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Harold W. Kuhn. 1955. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. Checkeval: A reliable llm-as-a-judge framework for evaluating text generation using checklists. *Preprint*, arXiv:2403.18771.

Yupian Lin, Tong Ruan, Jingping Liu, and Haofen Wang. 2024. A survey on neural data-to-text generation. *IEEE Transactions on Knowledge and Data Engineering*, 36(4):1431–1449.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1685–1701. Association for Computational Linguistics.

Chinonso Cynthia Osuji, Simon Mille, Ornait O'Connell, Thiago Castro Ferreira, Anya Belz, and Brian Davis. 2025a. Scaling up data-to-text generation to longer sequences: A new dataset and benchmark results for generation from large triple sets. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 810–822, Hanoi, Vietnam. Association for Computational Linguistics.

Chinonso Cynthia Osuji, Brian Timoney, Mark Andrade, Thiago Castro Ferreira, and Brian Davis. 2025b. Are multi-agents the new pipeline architecture for data-to-text systems? In *Proceedings of the 18th International Natural Language Generation Conference*, pages 542–553, Hanoi, Vietnam. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

RapidFuzz. 2026. RapidFuzz: Fast and accurate fuzzy string matching library. https://rapidfuzz.com/. Accessed 18 July 2026.

Luís Relvas, Sérgio Nunes, and André Restivo. 2026. Parallel atomic and hyper-relational reference triple annotations for European Portuguese football match reports. Dataset, INESC TEC. Available at https://doi.org/10.25747/J6D2-J785.

Xuan Ren and Lingqiao Liu. 2025. You can generate it again: Data-to-text generation with verification and correction prompting. In *Proceedings of the 7th ACM International Conference on Multimedia in Asia*. Association for Computing Machinery.

Patrícia Schmidtová, Saad Mahamood, Simone Balloccu, Ondřej Dušek, Albert Gatt, Dimitra Gkatzia, David M. Howcroft, Ondřej Plátek, and Adarsa Sivaprasad. 2024. Automatic Metrics in Natural Language Generation: A Survey of Current Evaluation Practices. In *Proceedings of the 17th International Natural Language Generation Conference (INLG)*. Available at https://aclanthology.org/2024.inlg-main.44.pdf.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Afonso Simplício, Gonçalo Vinagre Martins, Miguel Moura Ramos, Diogo Tavares, Rafael Ferreira, Giuseppe Attanasio, Duarte M. Alves, Inês Calvo, Inês Vieira, Rui Guerra, James Furtado, Beatriz Canaverde, Iago Paulo, Vasco Ramos, Diogo Glória-Silva, Miguel Faria, Marcos V. Treviso, Daniel Gomes, Pedro Gomes, and 3 others. 2026. AMALIA technical report: A fully open source large language model for european portuguese. *CoRR*, abs/2603.26511.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. *Preprint*, arXiv:1707.08052.

A Prompt Bank: Representative Examples

Table 6 presents one representative entry per predicate family from the prompt bank used for dynamic in-context prompting at Stage 1 (§3.2). For a given logical unit, the predicates present determine which family-matched examples are inserted into the writer prompt.

B Surface Realization Prompts

The surface realization pipeline (§3) uses four prompts: one at Stage 1, one at Stage 2a, and two at Stage 2b.

B.1 Stage 1: Triple Verbalization Prompt (Writer)

The following prompt is passed to AMALIA at Stage 1. It instructs the model to convert each

Table 6: Representative entries from the hyper-relational example bank, one per predicate family.

Family	Example input	Target output
Match info	`[TRIPLE] Match round 18 [/TRIPLE]`	The match was part of round 18.
Coach	`[TRIPLE] Liverpool coached_by Arne_Slot [/TRIPLE]`	Liverpool was coached by Arne Slot.
Lineup	`[TRIPLE] Alisson belongs_to Liverpool | position:goalkeeper | status:starter [/TRIPLE]`	Alisson Becker plays for Liverpool as a goalkeeper and started the match.
Goal	`[TRIPLE] Haaland scored_goal minute_39 | assisted_by:De_Bruyne | zone:inside_penalty_area | body_part:right_foot | origin:open_play [/TRIPLE]`	Erling Haaland scored with his right foot in the 39th minute from inside the penalty area, with an assist from Kevin De Bruyne.
Classification	`[TRIPLE] Manchester_City position_after_match 1 | points_before_match:69 | points_after_match:72 | position_before_match:1 [/TRIPLE]`	Before the match, Manchester City was first with 69 points. After the match, it remained first with 72 points.
Stats	`[TRIPLE] Arsenal performance match | total_shots:15 | shots_on_target:7 | possession_percentage:60 [/TRIPLE]`	Arsenal recorded 15 shots, 7 shots on target, and 60% possession.
Substitution	`[TRIPLE] Trossard replaced Martinelli | minute:46 [/TRIPLE]`	Trossard replaced Martinelli in the 46th minute.
Season stats	`[TRIPLE] Essende season_statistics season | goals_in_season:15 | scoring_streak_matches:3 [/TRIPLE]`	Essende reached 15 goals for the season, extending his scoring streak to three matches.

structured triple into a single factual sentence in European Portuguese while maintaining a strict one-to-one correspondence between input triples and output sentences. The conditional clause in braces is included only when the atomic triple representation is active; it instructs the model to suppress the numeric identifiers (e.g., `Goal_1`) that the atomic representation attaches to event subjects for grouping purposes.

```
Act as a strict formatting system. I will
    provide [TRIPLE] blocks containing match
    data. Convert each triple into a single
    sentence by following a deterministic
    template in Portuguese. Convert numeric
    positions to their ordinal form (1st, 2nd, 3
    rd, etc.). Use team and player names exactly
     as provided.

[ATOMIC ONLY] The numbers in the subjects (
    Goal_1, Card_5, Substitution_2) are unique
    identifiers used only to group events of the
     same type; do not include them in the
    output.

Do not include explanations, variations, or
    additional text. Maintain a one-to-one
    correspondence: one input entry equals one
    output sentence.
```

Listing 1: Stage 1 triple verbalization system prompt.

B.2 Stage 2a: Initial Article Synthesis System Prompt (Editor)

The following system prompt governs the single initial pass of Stage 2, in which the editor model receives the full set of Stage 1 sentences and synthesizes them into a structured article. This pass is executed once per match and is not part of the self-correction loop. The prompt specifies the mandatory article structure, section order, content rules for each section, and style constraints governing the output.

```
You are an experienced sports journalism editor.
     You have received a draft sports report
    composed of independent sections. Your only
    task is to rewrite it as a coherent, fluent,
     and journalistically natural article in
    European Portuguese. The draft is your ONLY
    source of truth. Do not introduce external
    knowledge about the teams, players, or
    competition.

MANDATORY STRUCTURE (FIXED ORDER):
1. Title -- a very short sentence summarizing
    the match outcome (win/draw) with the team
    names. It MUST be enclosed in <h1>...</h1>.
2. Subtitle -- a single line characterizing the
    result based only on data from the draft. It
     MUST be enclosed in <h2>...</h2>.
The following sections must be separated by <p
    >...</p>.
3. Short summary -- 1 to 2 sentences indicating
    the final result, teams, principal scorers,
    and any expulsions.
4. Introduction (Context) -- 1 short paragraph
    framing the match. Include the round, date
```

```
    and time, competition, and name of each team
    's coach. If the draft contains pre-match
    position and pre-match points values,
    mention them here. NEVER use post-match
    position or post-match points in this
    section.
5. Body -- describes all goals and relevant
    plays of the match (cards and substitutions)
     in strict chronological order.
6. Lineups (Starters and Substitutes) --
    introduce with a transition sentence. List
    ALL players from each team without exception
    . Grouping players by tactical sector is
    forbidden.
7. Debriefing, statistics, and season trivia --
    a final paragraph covering the impact on
    league standing (post-match values only),
    match statistics, trivia, form, and calendar
    .

PRESERVATION RULES (NON-NEGOTIABLE):
P1. Include ALL facts, players, statistics,
    goals, cards, and substitutions present in
    the draft.
P2. It is strictly forbidden to add any
    information not present in the draft.
P3. Never group players by position or tactical
    sector.
P4. In the Body, always preserve the temporal
    sequence of events.
P5. Pre-match values belong exclusively to
    section 4. Post-match values belong
    exclusively to section 7.
P6. Never omit data from previous or upcoming
    matches if present in the draft.
P7. The order of the sections must be as stated
    above.

STYLE:
- Vary sentence construction to avoid mechanical
     repetition.
- The article must be continuous prose, without
    bulleted lists.
```

Listing 2: Stage 2a initial article synthesis system prompt.

B.3 Stage 2b: Self-Correction Loop System Prompt (Editor)

From the second iteration onward, the editor model receives a revised system prompt that reorients its task from initial synthesis to targeted factual correction. The prompt defines two explicit stages: a factual correction stage, in which the editor inserts missing facts or removes hallucinated ones at their correct chronological positions, and a style revision stage, restricted to the modified sections only.

```
You are an extremely rigorous sports journalism
    editor.

STAGE 1 -- FACTUAL CORRECTION:
   Identify missing facts and insert them at the
     correct chronological position. Never omit
    Form data (previous/upcoming matches) if
    they are in the draft.

STAGE 2 -- STYLE REVISION:
   Ensure the final result sounds
    journalistically natural. The original text
    contains HTML markup (<h1>, <h2>, and <p>)
    that MUST be preserved.
```

Listing 3: Stage 2b self-correction system prompt (iterations 2–5).

B.4 Stage 2b: Self-Correction Loop User Prompt (Editor)

The user prompt passed at each self-correction iteration provides three structured inputs to the editor model: the source triple set, the article from the preceding iteration, and the critic's structured error report (§C.5). Placeholder variables (`{reference_triples}`, `{text}`, `{critique_history}`) denote values injected programmatically at runtime.

```
You have three blocks of information:
1. <DRAFT>: The absolute source of truth.
    Contains all triples used to generate the
    text.
2. <CURRENT_VERSION>: The current text, with the
     following section structure that you may
    NOT alter:
   1. Title (inside <h1>)
   2. Subtitle (inside <h2>)
   3. Short summary
   4. Introduction (Context)
   5. Body (goals, plays, cards, substitutions
    in chronological order)
   6. Lineups (Starters and Substitutes,
    including positions when mentioned in the
    draft)
   7. Debriefing, statistics, and trivia
   (Sections 3 to 7 must be separated by <p
    >...</p>)
3. <ERROR_HISTORY>: Missing or incorrect facts
    that need to be added or corrected in <
    CURRENT_VERSION>.

<DRAFT>
{reference_triples}
</DRAFT>

<CURRENT_VERSION>
{text}
</CURRENT_VERSION>

<ERROR_HISTORY>
{critique_history}
</ERROR_HISTORY>

INSTRUCTION: Insert missing facts or remove
    extra facts (both listed in <ERROR_HISTORY>)
     in the correct section and chronological
    position. Do not reorder existing sections.
    Do not shorten the text unless necessary to
    remove extra facts. HTML tags <h1>, <h2>,
    and <p> must be preserved. Then, perform a
    style revision ONLY on the modified sections
     to ensure journalistic flow. Do not use
    greetings. Return only the corrected text.
```

Listing 4: Stage 2b self-correction user prompt.

C Fact Extraction and Triple-Matching Details

The self-correction loop (§3.4) and the triple-matching metric (§4.2) both depend on the same fact-extraction and comparison components. This appendix documents both components, emphasizing details needed to reimplement the deterministic pipeline.

C.1 Fact Extractor

The critic (`gemini-2.5-pro`) receives the current article and returns a set of triples in `[TRIPLE] subject predicate object [/TRIPLE]` delimiters, matching the format of the source triple set. Extraction follows two conventions. First, extracted facts follow the same predicate vocabulary used by record selection (Appendix D), so the downstream matching engine operates on comparable strings. Second, the prompt instructs the model to prefix any inferential, rather than directly asserted, fact with the reserved subject token `deduction`; these facts are flagged so the post-processor can discard them before matching. Extraction uses one model call per evaluated article version at $T = 0.1$, with separate original Portuguese prompts for the atomic and hyper-relational schemas.

C.2 Deterministic Post-Processor

The raw extractor output passes through six deterministic steps in a fixed sequence:

1. **Delimiter parsing.** All content enclosed within `[TRIPLE]...[/TRIPLE]` is extracted; any surrounding model reasoning or narrative framing is discarded.

2. **Inference filtering.** Triples whose subject is the reserved token `deduction` are dropped; the extractor flags these triples as inferential rather than directly asserted, so they are excluded from evaluation.

3. **Alias normalization.** A fixed alias map replaces common malformed predicates with their canonical equivalents (e.g., `goalkeeper-of` → `goalkeeper_of`) and resolves generic team-reference terms against the article's own match metadata: `visitors` is replaced with the away team name recorded in `away_team`, and `hosts` with the home team name recorded in `home_team`.

4. **Entity locking.** For each player entity appearing as the subject of a `belongs_to` predicate, the associated team object is recorded. Any subsequent triple assigning the same player to a different team is discarded under the locking rule. A symmetric lock is applied to coach triples governed by `coached_by`.

5. **Match-metadata deduplication.** For triples with the subject `Match`, each predicate is allowed at most one occurrence; subsequent metadata triples for that predicate are dropped to prevent double-counting of match-level facts.

6. **Exact-string deduplication.** Any remaining identical triples are collapsed to a single occurrence.

The order matters: alias normalization must precede entity locking so that player–team assignments are compared against canonical team names, and both must precede deduplication so that a triple normalized into an existing form is dropped rather than kept. These rules assume that later conflicting affiliations and repeated metadata are extraction noise. If the generated article itself contains two contradictory assertions, however, entity locking or metadata deduplication can discard the second extracted assertion before scoring and thereby hide a genuine contradiction. Discarded triples were not independently audited, so the reported error counts are conditional on this filtering behavior.

C.3 Deterministic Triple-Matching System

This appendix gives the cell-level computation behind the matcher described in §4.2. The engine constructs a pairwise similarity matrix $S \in \mathbb{R}^{|\hat{T}| \times |T_{\text{source}}|}$ and computes a similarity-optimal one-to-one assignment over it via the Hungarian algorithm.

Triple strings are first normalized through lowercasing, Unicode diacritic stripping, and resolution against the domain alias dictionary (Appendix C.4). A priority-ordered procedure then assigns each cell S_{ij}:

1. If the two normalized strings are identical, assign a similarity of 100.

Table 7: Inline normalizations in the post-processor.

Source form	Canonical form
Predicate normalizations	
goalkeeper-of	goalkeeper_of
central-midfielder-of	central_midfielder_of
centre-back-of	centre_back_of
Generic-reference resolutions	
visitors	value of away_team
away side	value of away_team
home side	value of home_team
hosts	value of home_team

2. Otherwise, if one string's whitespace-separated token set is a strict subset of the other's, assign a similarity of 100. This case handles partial references such as surname-only mentions of players.

3. Otherwise, compute similarity as the normalized edit distance over sorted, deduplicated token bags using RapidFuzz (RapidFuzz, 2026).

The threshold, qualifier check, and outcome labels are as defined in §4.2. Two properties of this construction bear on the labels it produces. The token-bag fallback scores the normalized triple as a whole rather than enforcing separate compatibility for subject, predicate, object, and qualifiers, so it can overscore role-reversed asymmetric relations. Qualifier validity is absent from S and checked only after assignment, and the secondary contradiction pass does not make the assignment objective equivalent to maximizing valid matches.

C.4 Alias Map Excerpt

Table 7 shows the inline alias rules implemented by the post-processor. The implementation can additionally load optional entity dictionaries named `data/aliases.json` and `data/aliases_sr.json`; those files are not part of the released artifact (see Data Availability).

C.5 Critique Message and Episodic Memory

After post-processing and matching, the critic assembles the structured error report (§3.4) and passes it to the editor as the `<ERROR_HISTORY>` block of the Stage 2b user prompt (Appendix B.4). When all four lists are empty, the critic emits the sentinel string `NO_PROBLEMS`, and the loop terminates.

An episodic memory buffer maintains the three most recent critique messages. Errors that appear in more than one of these messages are promoted to a high-priority block prepended to the next critique, directing the editor's attention to persistent failures. We use this three-message window to bound context growth while retaining recent errors. Most matches converge by round 1 (92% hyper-relational; 78% atomic; Table 1), so the buffer rarely reaches capacity.

D Reference Triple Corpus

The human-report reference corpus contains parallel annotations for the same 50 matches. Table 8 reports representation-specific units: a hyper-relational unit may contain several qualifier attributes that appear as separate atomic triples, so the totals are not counts in a common semantic space. The two corpus files contain the match identifier, subject, predicate, object, source-text reference, and a flag indicating whether the annotation is directly extractable from the structured JSON.

The remaining 183 hyper-relational and 194 atomic annotations describe information not directly derivable from the structured match feed. Examples include untimed shot attempts represented with predicates such as `attempted_shot` and the value `unknown_minute`, as well as tactical descriptions supplied by the professional report. Representative predicate-to-text demonstrations used by the writer appear in Appendix A.

E CheckEval Framework

The evaluator uses one of two prompt templates. Faithfulness and factual consistency receive the source triples followed by the generated text; fluency, coherence, and naturalness receive the generated text alone. In both cases the dimension definition and its ordered checklist follow the text. The `gemini-2.5-pro` evaluator returns one reasoning statement and one binary answer per question in the form `Reasoning N: ... / QN: Yes|No`. Evaluations use $T = 0.0$, once per article and dimension, without a repeated-judge reliability run.

E.1 Template A: Source-dependent dimensions

```
### Task Overview ###
Your task is to read the structured Input
    Triples (the source data) and the
Generated Text (the report), then answer 'Yes'
    or 'No' to specific questions
about the generated text.

### Aspect Definition ###
<aspect_name> - <definition>
```

Table 8: Reference-corpus statistics by representation. Directly derivable counts refer to annotations marked as extractable from the structured JSON.

Representation	Total	Mean	SD	Min.	Max.	Predicates	Directly derivable
Hyper-relational	1,568	31.4	7.1	17	49	97	1,385 (88.3%)
Atomic	2,380	47.6	14.3	19	76	105	2,186 (91.8%)

```
### Instructions ###
1. Read these instructions carefully.
2. Read the Input Triples and Generated Text
    carefully.
3. Answer each question with 'Yes' or 'No',
    based on clear reasoning.
4. Follow the specified response format.

### Response Format ###
Reasoning 1: [Your reasoning]
Q1: [Your answer]
...

# Input Triples #
"<source>"

# Generated Text #
"<summary>"

# Questions #
<questions>

# Response
```

Listing 5: CheckEval prompt template for faithfulness and factual consistency.

E.2 Template B: Source-independent dimensions

```
### Task Overview ###
Your task is to read the Generated Text and
    answer 'Yes' or 'No' to specific
questions about its linguistic and structural
    quality.

### Aspect Definition ###
<aspect_name> - <definition>

### Instructions ###
1. Read these instructions carefully.
2. Read the Generated Text carefully.
3. Answer each question with 'Yes' or 'No',
    based on clear reasoning.
4. Follow the specified response format.

### Response Format ###
Reasoning 1: [Your reasoning]
Q1: [Your answer]
...

# Generated Text #
"<summary>"

# Questions #
<questions>

# Response
```

Listing 6: CheckEval prompt template for fluency, coherence, and naturalness.

The evaluation prompts contain no system name. Articles were processed once in dataset order; no randomized repeat-judge schedule was used.

E.3 Faithfulness

Definition: The generated text must not contain facts, entities, or events that are absent from the supplied triples. All information in the text must be verifiable from the data source. Direct logical and mathematical deductions, such as determining the winner from the goal totals, are valid and do not constitute hallucinations. This dimension assesses only the presence of elements unsupported by the triples.

Entities

- Does the text mention only teams explicitly present in the supplied triples?
- Does the text mention only players, coaches, and referees explicitly present in the supplied triples?

External Context

- Does the text refrain from referring to information absent from the supplied triples?
- Does the text avoid introducing external context or prior knowledge absent from the supplied triples?

Inference and Causality

- Does the text avoid superlative adjectives or emotional descriptions (e.g., “fantastic goal”) that cannot be verified from the supplied triples?
- Does the text avoid attributing unsupported emotional states, intentions, or causal relationships (e.g., claiming that a substitution occurred because of an earlier yellow card)?

Match Metadata

- Does the text mention only dates, stadiums, and locations present in the supplied triples, without adding external geographical information?
- Does the text mention only the correct competition and round or knockout stage specified in the supplied triples?

E.4 Factual Consistency

Definition: This dimension assesses the alignment between the report's statements and the source data (triples). A consistent text correctly represents the values, attributions, and chronology of events in the triples, without distortions, attribution errors, or internal contradictions.

Event Attribution

- Does the text correctly attribute goals to the players specified in the supplied triples?
- Does the text correctly attribute assists and cards to the players specified in the supplied triples?

Chronology and Temporal Structure

- Does the text report the minutes of goals, cards, and substitutions using the exact values in the supplied triples?
- Does the main body strictly follow match chronology, describing events from the first half to the second without temporal inversions?

League-Table Information

- Does the introduction use ONLY pre-match position and point totals, without mentioning the post-match point total in that section?
- Does the final debriefing paragraph use ONLY post-match position and point totals, without repeating the initial values?

Internal Consistency

- Does the final score stated in the title and short summary exactly match the goals described throughout the body?
- Do the final statistics (shots and possession) match the exact numerical values in the supplied triples?

E.5 Fluency

Definition: The grammatical and formal quality of the text. A fluent text follows Portuguese grammar and spelling, avoids literal translations from English, and sounds human-written. High scores require error-free text typical of European Portuguese.

Spelling and Grammar

- Are all words in the text free of spelling errors?
- Is the text free of gender-agreement errors (masculine/feminine)?
- Is the text free of number-agreement errors (singular/plural)?
- Is punctuation, including commas and full stops, used correctly throughout the text?
- Do shifts between verb tenses (past, present, future) make logical sense in context?
- Are there no abrupt and unjustified tense shifts within the same sentence?

Idiomatic European Portuguese Syntax

- Is the vocabulary clearly typical of European Portuguese?
- Is the text free of vocabulary or expressions exclusive to Brazilian Portuguese?
- Does the sentence construction avoid literal translations from English?
- Does pronoun use follow the usual rules of European Portuguese?
- Where applicable, does the text favor the infinitive construction (e.g., "is playing" in the European Portuguese form) over the Brazilian Portuguese gerund construction?

Articulation and Reading Flow

- Is capitalization fully correct?
- Does the text avoid excessive repetition of the same word within a sentence?
- Do paragraph lengths support continuous reading?
- Is the text free of incomplete or fragmented sentences?
- Do the sentences avoid resembling lists of isolated data separated by commas?

E.6 Coherence

Definition: The logical structure of the text. A coherent text combines information naturally, avoids repetition, and creates effective links between ideas. Individual sentences must not merely make sense in isolation; together they must build a clear narrative from beginning to end.

Fact Aggregation and Chronology

- Are match events reported in an easy-to-follow chronological order?
- Are events that occurred at the same time, or in the same passage of play, described together?
- Does the text avoid listing facts mechanically, one per sentence?
- Are substitutions and cards integrated into the match narrative rather than listed at the end?
- Is the progression of match time (start, half-time, end) clear to the reader?

Cohesion and Transition Mechanisms

- Are transition words (e.g., "however", "therefore", "in addition") used correctly and appropriately?
- Does the text vary its transition words and expressions rather than repeatedly using the same ones?
- Do transitions between paragraphs flow naturally without abrupt topic shifts?
- When a new topic is introduced (e.g., the league table), is the transition from the match summary smooth?

Topic and Redundancy Management

- Is the central idea (the result or key match event) consistent across all paragraphs?
- Does the text focus on important details without digressing into irrelevant topics?
- Is each specific match event (e.g., a goal) described only once, without later repetition as though it were new?
- Does the text have a perceptible beginning (context), middle (action), and end (conclusion/consequences)?

E.7 Naturalness

Definition: The degree to which the text resembles human writing. A text that merely states what happened, even without errors, appears artificial if it repeatedly uses the same words. A strong text reads naturally, uses journalistic language, and does not reveal its computational origin.

Sentence-Structure Variety

- Do sentence lengths vary throughout the text (short and long)?
- Does the text vary how sentences begin, avoiding repeated openings with a player or team name?
- Does the text avoid a repetitive "Subject + Verb + Complement" structure in every sentence?
- Does the text avoid repeatedly using the rigid pattern "In minute X, player Y did Z"?

Lexical Richness and Journalistic Style

- Does the text use rich and varied vocabulary appropriate to sports reporting?
- Does the text use suitable synonyms for the same action (e.g., kick, shoot, strike)?
- Does the text avoid excessive use of generic verbs such as "be", "have", or "do"?
- Does the writing style reflect the energy typical of a sports report or match story?
- Does the text sound like professional journalism rather than a bureaucratic statistical report?

Human-Writing Imitation

- When read in full, does the text sound like something that could have appeared in a sports newspaper?
- Does the text sound natural enough to conceal that it was generated by artificial intelligence?
- Does the tone (serious, enthusiastic, informative) remain consistent from beginning to end?
- Does the vocabulary avoid expressions that, while correct, sound overly formal or artificial in football reporting?

F Generated Output Examples

F.1 Three-System Comparison

Table 9 shows the final article produced by each system for the same match, and Table 10 gives author translations of the same three texts. The example is drawn from the 13-match Prosebot-derived content condition (§5.4), so the neural inputs contain only facts recovered from the Prosebot report. Both neural articles therefore omit the lineups section and the coach names that the Stage 2a prompt (Appendix B.2) marks as mandatory; in the full 50-match condition both are present. Prosebot names both coaches only through possessive phrases such as *a equipa de Sérgio Conceição* ("Sérgio Conceição's team"). The extractor recovered no coached_by triples from these, illustrating the recovery gap noted in §5.4.

F.2 Worked Self-Correction Example

This example is drawn from match 9429630 in the stored hyper-relational refinement record; as elsewhere in the paper, identifiers and predicates are shown in author-translated English. At round 0, the source contained:

```
[TRIPLE] Godfried_Frimpong substituted
         Pedro_Amador | minute:90 [/TRIPLE]
[TRIPLE] Ruben_Ismael substituted
         Lawrence_Ofori | minute:90 [/TRIPLE]
```

The critic extracted both substitutions but returned the minute qualifier as unknown. The matcher therefore labeled the units **Partial**, not Missing or Unsupported, and the error report led the editor to restore the minute. On re-extraction, both matched with complete qualifiers: precision, recall, and F1 moved from 97.01% at round 0 to 100% at round 1, with no error category remaining. The example shows category-specific deterministic feedback; the final score remains conditional on extracting the revised article.

Table 9: Final articles for the same match under the three systems, in the original European Portuguese. HTML section markup is rendered typographically; the text is otherwise verbatim system output, including the comma error in the atomic article (*cartão amarelo e, ter sido expulso*). Inputs come from the 13-match Prosebot-derived condition, so lineups and coaches are absent from both neural articles.

Hyper-relational pipeline	**Atomic pipeline**	**Prosebot**
FC Porto vence Portimonense *Golo madrugador e expulsão tardia decidem* O FC Porto venceu o Portimonense pela margem mínima de 1-0, num jogo decidido por um golo de Evanilson. A equipa da casa terminou a partida reduzida a dez unidades após a expulsão de David Carmo. O encontro, a contar para a 8.ª jornada da competição, opôs o FC Porto à equipa do Portimonense. O único golo da partida surgiu cedo, ao minuto 9, quando Evanilson, com o pé esquerdo e dentro da área, inaugurou o marcador. Já na fase final do encontro, ao minuto 81, David Carmo, do FC Porto, viu o segundo cartão amarelo e, consequentemente, o cartão vermelho, deixando a sua equipa em inferioridade numérica até ao apito final. Com este resultado, o FC Porto fixou-se na 3.ª posição com 19 pontos, enquanto o Portimonense se posicionou no 12.º lugar com 8 pontos. No que diz respeito às estatísticas do jogo, o FC Porto dominou a posse de bola com 70%, contra 30% do Portimonense. Quanto à forma recente, o FC Porto vinha de uma derrota e no próximo compromisso defronta o FC Vizela. Por sua vez, o Portimonense, que vinha de uma vitória, terá pela frente o Estoril Praia.	**FC Porto vence Portimonense** *Vitória pela margem mínima decidida cedo* O FC Porto derrotou o Portimonense por 1-0, num jogo resolvido por um golo solitário de Evanilson e que ficou ainda marcado pela expulsão de David Carmo perto do final. A partida, referente à 8.ª jornada, colocou frente a frente o FC Porto, a jogar em casa, e o Portimonense. O momento decisivo do encontro aconteceu logo ao minuto 9, quando Evanilson, no interior da área, finalizou com o pé esquerdo e inaugurou o marcador. Aos 81 minutos, a equipa da casa ficou reduzida a dez elementos, depois de David Carmo ter visto o segundo cartão amarelo e, ter sido expulso com cartão vermelho. Com este resultado, o FC Porto fixou-se na 3.ª posição com 19 pontos, enquanto o Portimonense se posicionou no 12.º posto com 8 pontos. As estatísticas do jogo revelaram um claro domínio dos anfitriões na posse de bola, com 70% contra 30% da equipa visitante. No que toca à forma recente, o FC Porto vinha de uma derrota e o seu próximo compromisso será contra o FC Vizela. Já o Portimonense, que vinha de uma vitória, defrontará o Estoril Praia na próxima jornada.	**FC Porto derrotou Portimonense por 1x0** *Vitória tangencial em casa* Na oitava jornada, o FC Porto bateu o Portimonense, 1x0, no domingo. Nesta competição, a equipa de Porto vinha de uma derrota, e a equipa de Portimão chegava de uma vitória. Evanilson marcou um golo solitário para o FC Porto ao minuto 9, de dentro da área e com o pé esquerdo. Aos 81 minutos, os Dragões viram David Carmo ser expulso devido à acumulação de amarelos. Somando à vitória, o FC Porto obteve o melhor registo de posse de bola, com 70%. Somando à derrota, o Portimonense registou a pior posse de bola da época com 30%. Depois deste resultado a equipa de Sérgio Conceição ocupa o terceiro lugar na classificação geral, 19 pontos, com a equipa de Portimão a encontrar-se na décima segunda posição, 8 pontos. Quanto à próxima jornada, a equipa de Porto visita o terreno do FC Vizela. Por sua vez, a equipa de Paulo Sérgio joga em casa frente ao Estoril Praia.

Table 10: Author translations of the three articles in Table 9. The translations follow the source closely, including the comma error in the atomic article, and are provided for readers of English; all reported results were computed on the Portuguese originals.

Hyper-relational pipeline	Atomic pipeline	Prosebot
FC Porto beat Portimonense *Early goal and late sending-off decide it* FC Porto beat Portimonense by the narrowest margin, 1-0, in a match decided by a goal from Evanilson. The home side finished the match down to ten men after David Carmo was sent off. The fixture, counting towards round 8 of the competition, pitted FC Porto against Portimonense. The only goal of the match came early, in the 9th minute, when Evanilson, with his left foot and inside the area, opened the scoring. In the closing stage of the match, in the 81st minute, FC Porto's David Carmo received a second yellow card and, consequently, a red card, leaving his team a man down until the final whistle. With this result, FC Porto settled into 3rd place with 19 points, while Portimonense stood in 12th with 8 points. As for the match statistics, FC Porto dominated possession with 70%, against Portimonense's 30%. On recent form, FC Porto were coming off a defeat and face FC Vizela in their next fixture. Portimonense, in turn, coming off a win, will face Estoril Praia.	**FC Porto beat Portimonense** *A narrow win settled early* FC Porto defeated Portimonense 1-0, in a match settled by a solitary goal from Evanilson and also marked by David Carmo's sending-off near the end. The match, in round 8, brought together FC Porto, playing at home, and Portimonense. The decisive moment came as early as the 9th minute, when Evanilson, inside the area, finished with his left foot and opened the scoring. In the 81st minute, the home side were reduced to ten men, after David Carmo had received a second yellow card and, been sent off with a red card. With this result, FC Porto settled into 3rd place with 19 points, while Portimonense stood in 12th with 8 points. The match statistics showed clear home dominance in possession, 70% against the visitors' 30%. On recent form, FC Porto were coming off a defeat and their next fixture will be against FC Vizela. Portimonense, coming off a win, will face Estoril Praia in the next round.	**FC Porto defeated Portimonense by 1x0** *Narrow home win* In round eight, FC Porto beat Portimonense, 1x0, on Sunday. In this competition, the team from Porto were coming off a defeat, and the team from Portimão arrived from a win. Evanilson scored a solitary goal for FC Porto in the 9th minute, from inside the area and with his left foot. In the 81st minute, the Dragons saw David Carmo sent off for accumulating yellow cards. Adding to the win, FC Porto recorded the best possession figure, 70%. Adding to the defeat, Portimonense recorded the season's worst possession with 30%. After this result Sérgio Conceição's team occupy third place in the general standings, 19 points, with the team from Portimão in twelfth position, 8 points. As for the next round, the team from Porto visit FC Vizela's ground. In turn, Paulo Sérgio's team play at home against Estoril Praia.