**Narrative-to-Box-Score:  
Evaluating LLMs on Structured Reasoning in Sports Analytics**

Submitted by: Lior Ben Sidi – 207490913, Ido Avital – 207850280  
This report summarizes the project at a non-technical level, focusing on problem framing, approach, and evaluation design.

**For more technical information, we recommend reading the "README.md" file.**

**Executive Summary**

This project evaluates how well a Large Language Model (LLM) can convert a natural-language, chronological play-by-play description of a basketball game into a complete, structured box-score in JSON format. The task stresses temporal reasoning (events over time), aggregation (rolling up plays to final statistics), and strict adherence to a predefined schema. The pipeline generates synthetic games, queries an LLM, repairs the model’s output to match the schema when needed, and scores the result with clear, interpretable metrics.

**Problem Statement & Motivation**

Sports analytics relies on structured data such as box scores and play-by-play logs. Automating the transformation from narrative descriptions to structured statistics is valuable for performance analysis and decision-making. However, it is can be challenging for language models because the output must be both faithful to the narrative, such that it will represent the true stats, and also consistent with a strict schema of JSON.  
This project provides a focused testbed for that challenge.

**Task Definition (What the Model Must Do)**

Given a chronological play-by-play of a single basketball game plus team/roster context, the model must produce a complete box-score: final score, per-team totals, and per-player statistics. The output must follow a fixed schema, be internally consistent, and reflect what actually happened in the narrative.

**Data & Simulation**

Games are synthetically simulated to produce two paired artifacts: (1) a natural-language play-by-play narrative with team metadata (rosters, starting lineup, participants), and (2) a ground-truth statistical report (team and per-player stats). The simulation enforces core safety invariants such as “attempts ≥ made” across all shot types and handles realistic basketball phenomena like substitutions, rebounds, fouls, and occasional video-review (VAR) adjustments that may overturn or modify recent plays while keeping the data consistent.

Rationale. Synthetic pairing lets us control difficulty and coverage while keeping a clean “source → target” mapping for auditing. Substitutions dynamically update the set of participants, testing entity tracking; rebounds and turnovers test possession flow; and VAR introduces limited retroactive edits to probe whether models can reconcile earlier and later statements without breaking invariants.

**Difficulty Levels**

To stress different aspects of reasoning, the simulator offers three presets. Internally, these presets adjust event mix, narrative variety, substitution/VAR rates, maximum passes before a shot, optional assist-wording ambiguity, and the target number of events (target game length). The wording does not change the underlying statistics, but it **can change** how hard the text is to interpret.

* **Basic** — Shorter games, simpler phrasing, few substitutions, no VAR.  
  **Why:** Isolates baseline skills: mapping narrative to schema, simple aggregation, and exact JSON formatting—without long-context drift or retroactive edits.
* **Medium** — Longer narratives, moderate substitutions, few VAR, richer wording.  
  **Why:** Adds paraphrase variety and more participants, along with occasional retroactive corrections, to test more robustness, while keeping complexity manageable..
* **Hard** — Many events, higher substitution/VAR rates, broader lexicon, and more adversarial wording (e.g., ambiguous pass verbs); often tighter possession windows (fewer allowed regular passes) to force quicker plays that affect the stats.  
  **Why:** Stresses long-context memory, entity consistency across many updates, and disambiguation under noisy wording - exactly where structured reasoning tends to break.
* **Note:** See "**Appendices**" in that report for more details on parameter value selection.

**Evaluation**

The evaluation compares the model’s structured box-score to the known ground truth, using two complementary views:

* Per-field accuracy: counts how many individual fields match exactly.
* Block-normalized accuracy: evaluates correctness within logical blocks (final score, team stats, player stats) so that each block contributes proportionally.

Results are recorded per game and summarized per difficulty, with transparent formulas so the accuracy can be audited rather than treated as a black box.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **basic** | | | | **medium** | | | | **hard** | | | | **General** | | | |
| **Method 1** | | **Method 2** | | **Method 1** | | **Method 2** | | **Method 1** | | **Method 2** | | **Method 1** | | **Method 2** | |
| **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Results of "Gemini-2.5-pro" on 150 examples (**50 on each difficulty level, that in "data"**)**(See the folder named "data-details" in folder "Appendices" for more details)Methods: 1 = "field", 2 = "fractional\_per\_block**" |** Metrices: A = Average, M = Median

**Analysis & Insights**

Initial observations show that LLMs handle short, simple games relatively well, often producing correct aggregations and valid JSON. As narratives become longer and more complex, models frequently struggle with misaligned team/player statistics or degenerate all-zero reports. The hardest cases expose significant long-context limitations. Detailed, per-game outputs enable inspection of where and why errors occur.

**Limitations**

The setup focuses on single-game narratives and a fixed schema; it does not evaluate multi-game aggregation, injury/time-on-court modeling, or retrieval from external databases. These choices keep the task well-controlled but limit generality to broader analytics workflows.

**Conclusion & Future Work**

We built a challenging dataset and evaluation framework to assess the structured reasoning of LLMs in sports analytics, generating diverse and automatic narratives, with strict and transparent evaluation. While modern LLMs can handle basic aggregation, they still struggle with long-context structured reasoning. **Future work** will explore:  
(1) Fine-tuning on structured sports data, that represents an even more realistic game.  
(2) Check for results of the LLM while integrating retrieval or tool-based reasoning.

**Appendices**

**Performance Comparison**

We compare multiple LLMs across providers on the same 15-example set per difficulty (Basic/Medium/Hard).  
Methods: 1 = "field" (strict per-field match), 2 = "fractional\_per\_block" (block-normalized: final score, team A/B, players A/B).  
Metrics: A = Average, M = Median.

How to read the grid:

* Rows are models within a provider;  
  moving down generally means a larger/stronger model.
* Columns move from Basic → Medium → Hard (left to right),  
  and from Method 1 → Method 2 (within each difficulty).
* **OOC (Out of credits): We exhausted Anthropic API credits mid-run.  
  Having already purchased $5 + 10$ top-ups for earlier experiments, we chose not to buy additional credits; therefore, Anthropic results in this comparison are incomplete.  
  \*\*This is a budget constraint, not a model issue.\*\*  
  In separate preliminary tests, all Anthropic models listed in this table gave valid responses, but those results are not included here due to the credit limit.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **basic** | | | | **medium** | | | | **hard** | | | | **General** | | | |
| **Method 1** | | **Method 2** | | **Method 1** | | **Method 2** | | **Method 1** | | **Method 2** | | **Method 1** | | **Method 2** | |
| **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** | **A** | **M** |
| **Gemini** (more details in "Gemini - 15 examples" folder, in " Performance Comparison" folder, in "Appendices" folder) | | | | | | | | | | | | | | | | |
| **gemini/gemini-1.5-flash** | **64.6** | **64.1** | **40.3** | **39.5** | **38.5** | **38.3** | **18.5** | **18** | **23.6** | **23** | **11** | **10.4** | **42.2** | **38.3** | **23.3** | **18** |
| **gemini/gemini-1.5-pro** | **71.1** | **68.9** | **43.8** | **44.4** | **48.6** | **46.5** | **23.3** | **22.9** | **32.0** | **32.4** | **14.6** | **14.8** | **50.6** | **46.5** | **27.2** | **22.9** |
| **gemini/gemini-2.5-flash** | **91.8** | **93.3** | **84** | **90.4** | **65.4** | **64.8** | **47.2** | **51.9** | **41.1** | **41.1** | **20.3** | **20.3** | **72.3** | **72.9** | **58** | **58.2** |
| **gemini/gemini-2.5-pro** | **94.4** | **96.9** | **87.6** | **95.3** | **80.5** | **81.9** | **48.8** | **41.9** | **59.3** | **59** | **27** | **26.8** | **78.1** | **81.9** | **54.4** | **41.9** |
| **OpenAI** (more details in "OpenAI - 15 examples" folder, in " Performance Comparison" folder, in "Appendices" folder) | | | | | | | | | | | | | | | | |
| **gpt-4o-mini** | **55.5** | **53.3** | **31.9** | **31.7** | **29.9** | **29.9** | **15.3** | **15.5** | **19.3** | **19.6** | **9.6** | **10** | **34.9** | **29.9** | **18.9** | **15.5** |
| **gpt-4o** | **71.8** | **72.2** | **55** | **59.1** | **41.9** | **43.5** | **20.5** | **22.3** | **26.5** | **26.6** | **13.2** | **13.4** | **46.7** | **43.5** | **29.6** | **22.3** |
| **Anthropic** (more details in "Anthropic- 15 examples" folder, in " Performance Comparison" folder, in "Appendices" folder) | | | | | | | | | | | | | | | | |
| **claude-sonnet -4-20250514** | **80.8** | **80** | **62.2** | **62.4** | **55.7** | **54.6** | **28.3** | **28.7** | **37.7** | **37.8** | **16.3** | **16.4** | **58** | **54.6** | **35.6** | **28.7** |
| **claude-opus -4-20250514** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **claude-opus -4-1-20250805** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |

Key takeaways:

1. **Difficulty gradient:**performance drops consistently from **Basic → Medium → Hard**. The Hard set stresses long context, more players, more retroactive edits (VAR), and more adversarial wording (e.g., ambiguous pass verbs), which increase block-level penalties.
2. **Metric effect:**  
   Method 2 is systematically lower than Method 1 because it averages **five blocks**:  
   (i) final score, (ii) team A, (iii) team B, (iv) players A, (v) players B,  
   So **each block carries ~20%** of the grade.  
   Because of that, a single error in the **final score** drags down ~20% of Method-2 accuracy, whereas in Method 1, it will only have a small weight, which is equal to other smaller mistakes.  
   The same applies similarly to the **team blocks**: each mistake results in 1/15%,  
   And if, for example, three stats are wrong, you lose another 1/3% at once.  
     
   **Also, mistakes in these blocks are more likely to occur because:**

* **Aggregation pressure:** Final score and team totals depend on aggregating many events; one misread (e.g., 3→2 after VAR) propagates to multiple fields and the final.
* **Retroactive edits (VAR):** Late corrections force the model to revise earlier assumptions; if it doesn’t, final and team totals drift.
* **Roster churn (participants/substitutions):** Who’s on court changes; if players aren’t tracked correctly, team blocks and player blocks desynchronize (missing players, all-zeros, or double-counting).
* **Invariants and dependencies:** Constraints like *attempts ≥ made* and “team totals = sum of players” are easy to violate under long context and paraphrased wording.
* **Partial/invalid outputs:** Truncated JSON or omitted sections count as a near-zero for that entire block in Method 2, but only penalize a subset of fields in Method 1.
* Using ambiguous verbs and a variety of templates can lead to incorrect attribution, which directly affects team and player blocks.

1. **Scaling helps:**  
   within each provider, upgrading to a more capable model almost always improves scores.
2. **Medians vs Averages:**  
   If **M >> A**, the model is competent but **unstable**;  
   if **A ≈ M**, performance is **steady** across the 15 games.

Bottom line:

Upgrading to stronger models improves both **accuracy** and **stability**, but **Hard** remains challenging due to long context, paraphrase variety, substitutions, and VAR

Method 2 (block-normalized) highlights these weaknesses more clearly.  
It emphasizes complete, end-to-end accuracy rather than just field-level matches.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Basic** | **Medium** | **Hard** | **Short Explanation** |
| **target\_events** | 150 | 600 | 900 | Controls how many events are generated. Low = short/simple logs, High = long/complex games. |
| **difficulty\_max\_passes** | 5 | 3 | 1 | Maximum passes before a shot. More passes create simpler logs (don't effect the stats), fewer passes make parsing harder. |
| **adversarial\_assist\_bias** | False | True | True | If True, uses ambiguous verbs for passes, making assist detection harder. |
| **substitution chance** | 5% | 10% | 15% | Probability of substitutions. More subs → more players appear, harder tracking of participants. |
| **VAR events chance** | Disabled | 5% | 10% | VAR cancels/changes plays. Adds complexity and requires the model to undo/reason backwards. |
| **narrative variety** | ¼ of phrases | ½ of phrases | All phrases | How many wording templates are sampled. Higher = more linguistic diversity, harder for LLMs. |
| **EVENT\_WEIGHTS** | Bias to misses & fouls | Balanced | Bias to made shots & opposite order of "passer", "shooter" in 2pt, 3pt made shots | Weighted distribution of event types shapes the overall game difficulty and ambiguity. More "important" statistical events and the use of opposite order of players in the event of made shots may cause the LLM to get confused between the roles of the player who made the assist and the player who scores. |

**More Detailed Parameter Value Selection**

**Detailed Explanation of Parameters**

* **target\_events:**

This parameter determines the total number of simulated play-by-play events per game. A lower value (150 in Basic) results in short and simple logs, while higher values (600 in Medium, 900 in Hard) produce longer narratives. From an NLP perspective, longer sequences exacerbate challenges in sequence modeling, as highlighted in the course when discussing vanishing gradients in RNNs and the bottleneck problem in sequence-to-sequence models. Transformers alleviate some of these issues via self-attention, but context length remains a practical limitation.

* **difficulty\_max\_passes:**

This parameter sets the maximum number of passes allowed before a shot. Basic games allow up to 5 passes, Medium 3, and Hard only 1. Fewer passes increase ambiguity by forcing quicker offensive plays. Theoretically, this connects to sparse context windows in early n-gram models, where limited history reduces predictive certainty and forces reliance on distributional generalization.

* **adversarial\_assist\_bias:**

When enabled (Medium, Hard), neutral pass verbs are replaced with ambiguous alternatives such as 'feeds' or 'delivers'. This complicates assist recognition. In NLP terms, this resembles lexical variation in distributional semantics: word embeddings (e.g., Word2Vec, GloVe) place synonyms in nearby regions of the vector space, but rare or misleading lexical choices may increase confusion. Thus, the bias directly stresses a model’s robustness to paraphrasing and lexical variability.

* **substitution chance:**

Controls how often substitutions occur: 5% in Basic, 10% in Medium, 15% in Hard. Higher substitution rates expand the set of active participants, requiring long-context tracking. This parallels sequence-labeling tasks in NLP (like NER), where entities may change mid-sequence. RNNs struggle with long dependencies due to vanishing gradients, while Transformers handle longer contexts better but still face quadratic scaling issues.

* **VAR events:**

Retroactive changes (e.g., canceling baskets, downgrading shots) are disabled in Basic, appear at 5% in Medium, and 10% in Hard. These events force the model to revise prior states, akin to structured prediction with retroactive constraints. From the course, this relates to limitations of left-to-right generative models, which cannot easily revise earlier predictions, highlighting the need for architectures with bidirectional context (e.g., BERT) or explicit constraint handling.

* **narrative variety:**

This parameter determines how many phrasing templates are used: one-quarter in Basic, half in Medium, and all templates in Hard. Higher variety increases linguistic diversity and reduces reliance on memorized surface patterns. This reflects the motivation for dense word embeddings: distributional models generalize across varied contexts, but extreme variability stresses a model’s ability to maintain semantic equivalence across forms, a challenge noted in discussions of Zipf’s law and OOV handling.

* **EVENT\_WEIGHTS:**

Defines the probability distribution over event types (shots, misses, turnovers, fouls). Basic is biased toward misses and fouls, Medium is balanced, and Hard favors made shots and turnovers. This reshapes the statistical prior of the game log, echoing the role of prior probabilities in probabilistic language models. As in n-gram or log-linear models, skewed distributions lead to systematic biases in prediction, requiring models to adjust expectations dynamically.

**Check the ability of a real human to create reports for 3 games**

While we just think of the idea of our task, we wanted to check if a real human could be able to constant real Report according to sequence of game events (those are small games).

We asked a friend to do that, and these are his results:

For "hard\_game\_5" – 100% accuracy:  
A screenshot of a document

AI-generated content may be incorrect.

For "hard\_game\_6" – 100% accuracy:A screenshot of a document

AI-generated content may be incorrect.

For "hard\_game\_7" – 100% accuracy:  
A white sheet with black text

AI-generated content may be incorrect.