Report – NLP Final Project

**Submitted by:** Lior Ben Sidi (207490913), Ido Avital (207850280)

# 1) Introduction

We evaluate Large Language Models (LLMs) on a structured reasoning task: converting a chronological basketball play-by-play narrative into a complete JSON box score. The task stresses temporal reasoning, aggregation across events, long-context robustness, and strict schema compliance.

# 2) Background

While LLMs show impressive capabilities, measuring their performance on domain-specific, schema-bound outputs is still challenging;

# 3) Methodology

## 3.1 Data Generation (generate\_data.py)

The simulator produces (a) a natural-language play\_by\_play log plus team metadata and (b) a ground-truth report with team and per-player stats, including participants. Outputs are written as interleaved pairs to data/examples.jsonl (example → true\_report). Schemas (matchup/teams/play\_by\_play; final\_score/final\_stats) are fixed and consistent between example and ground truth. VAR events can undo/modify shots (e.g., 3PT↔2PT) while keeping invariants like attempts ≥ made. Difficulty presets (basic, medium, hard) tune event mix, target length, substitutions, VAR rate, lexicon breadth, and assist-leaning wording via adversarial\_assist\_bias.

## 3.2 Model Evaluation (run\_eval.py)

For each game: build an instruction+context prompt, call the selected model, then repair & rebuild the model's JSON to the exact ground-truth shape and types. The runner detects JSON-mode support once; on a failure it falls back to plain text plus local JSON repair (strip fences/comments, remove trailing commas, balance braces/brackets). Artifacts saved per difficulty: raw text, rebuilt JSON, optional per-game \_\_details.json, and a consolidated summary.json.

## 3.3 Scoring (evaluation.py)

Evaluation is single-pass: the evaluator emits per-check contributions and from the same list computes two accuracy modes:

* Mode A — field: strict per-field counting (each check weight = 1.0).
* Mode B — fractional\_per\_block: block-normalized; the five logical blocks (final score; team A stats; team B stats; players A; players B) each sum to ~1.0.

Checks cover: top-level final\_score, team-level stats for each team, all stats for each participant, and an all-zeros check for every non-participant on the roster. If an entire team block is missing in the model’s report, all checks for that team are counted incorrect (denominator preserved). formula\_vars in outputs expose the exact numerators/denominators and block weights for transparency.

# 4) Results

Data volume recorded: the main set uses 150 examples for gemini/gemini-2.5-pro; other models have 15 examples in the comparison table.

Illustrative results (from the current table):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metric | Basic | Medium | Hard |
| gemini/gemini-1.5-pro | field (avg) | 71.1 | 48.6 | 32.0 |
|  | fractional\_per\_block (avg) | 43.8 | 23.3 | 14.6 |
| gemini/gemini-1.5-flash | field (avg) | 64.6 | 38.5 | 23.6 |
|  | fractional\_per\_block (avg) | 40.3 | 18.5 | 11.0 |

(Medians are also recorded in the table and will populate for additional models as runs complete.)

# 5) Analysis & Insights

Short, simple games tend to be handled well (correct aggregations, valid JSON). As length/complexity increases, models more often produce malformed JSON, misaligned team/player stats, or degenerate all-zero reports. The hardest cases expose long-context limitations. The pipeline’s JSON-mode detection, local repair, and the all-zeros guard mitigate—but don’t fully eliminate—these errors.

# 6) Conclusion & Future Work

We present a dataset and end-to-end evaluation framework for narrative→box-score conversion that surfaces structured-reasoning gaps under strict schemas. Future directions in the notes include fine-tuning on structured sports data, adding retrieval/tool-based reasoning, and stricter schema validation.

# Appendix (operational notes)

• Reproducibility: set a fixed seed at the top of the generator for deterministic runs.

• Outputs: llm\_responses/<difficulty>/{text,json} per game and summary.json with per-game accuracy\_pct, formula, and formula\_vars.