Predicting LLM Planning Performance with Logistic Regression

Sanjay Chezhian, Robotics and Autonomous Systems with AI Specialization

Mentor: Dr. Lindsay Sanneman, Assistant Professor

School of Computing and Augmented Intelligence (SCAI), Ira A. Fulton Schools of Engineering, Arizona State University

Introduction

Research Question:

- To what extent can a **logistic-regression** classifier built on compact text embeddings predict, before inference, whether a specified **LLM** will produce a valid plan on a given **Blocksworld** instance?

Objective:

- Build a labeled dataset from **PlanBench / Blocksworld**: extract **init & goal states** from PDDL and form **natural-language prompts**.
- Generate plan candidates with **Llama-3 (8B, 70B)** and obtain **valid/invalid** labels via a validator.
- Encode text with **SBERT** and add simple **textual/structural features** (token count, goal literals, predicted steps, operator diversity, repeats).
- Train a **logistic regression** model, **calibrate** probabilities, and **select** τ(maximize F1 on validation).
- Evaluate **AUROC, F1, Brier score**, and cross-model **transfer** performance.

Impact:

- **Save tokens/time** by early rejecting low-probability cases and **auto-routing** to stronger prompts, bigger models, or classical planners.
- Provide **interpretable coefficients** for what makes a plan likely to succeed.
- Establish a **portable pre-execution gate** that can extend beyond Blocksworld to other planning domains.

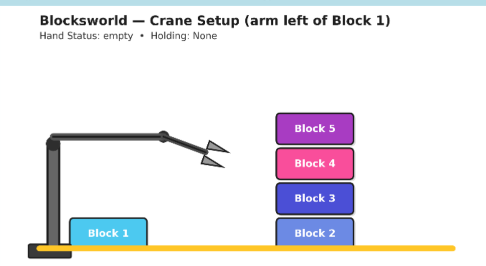

Figure 1.1: BlocksWorld Crane Setup

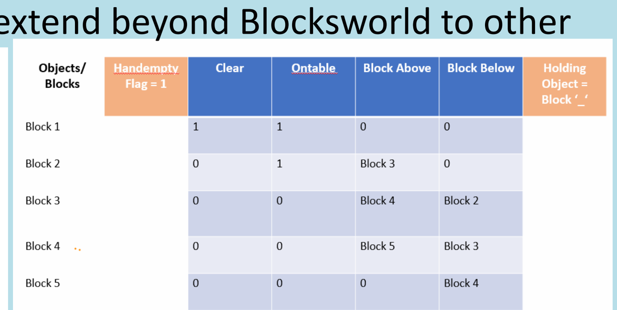

Figure 1.2: Blocksword state representation

Methodology

- **Parse PDDL → State JSON**
  Extract objects, predicates, Init, Goal. Save as JSON for templating.
- **Prompt construction**
  Blocksworld domain description + Init state NL + Goal state NL + instruction to produce a cost-optimal plan.
- **Plan generation**
  Run Llama-3-8B and Llama-3-70B to get a candidate plan for each instance.
- **Validator label (peval)**
  Validate the plan. peval ∈ {0,1}: 0 invalid, 1 valid. Store prompt, JSON, plan, peval.
- **Pre-execution features**
  SBERT embedding of the prompt, token counts, object and goal literal counts, simple structure stats.
- **Model, calibration, threshold**
  Logistic regression predicts P(valid). Calibrate probabilities. Choose τ on validation to maximize F1.
- **Deployment use**
  If P(valid) < τ: re-prompt, stronger LLM, or classical planner. Otherwise accept or verify once.

SBERT pooling (prompt → embedding):

$$\boxed{u = \frac{1}{T}\sum_{t=1}^{T} h_t,\ \tilde{u} = \frac{u}{\|u\|_2}}$$

Feature vector (embed + stats):

$$\boxed{x = [\tilde{u};\ s']\text{with } s' = \frac{s-\mu_{train}}{\sigma_{train}}}$$

Label (from validator):

$$\boxed{y = \text{peval} \in \{0, 1\}}$$

Logistic regression (0/1 output):

$$\boxed{z = w^{\top}x + b,\ \ \hat{y} = 1\,,\ \left[\frac{1}{1+e^{-z}} \geq 0.5\right]}$$

Training loss:

$$\boxed{L=\frac{1}{N}\sum_i -y_i log\sigma(z_i) - (1-y_i)\log(1-\sigma(z_i))] + \lambda\|\omega\|_2^2}$$

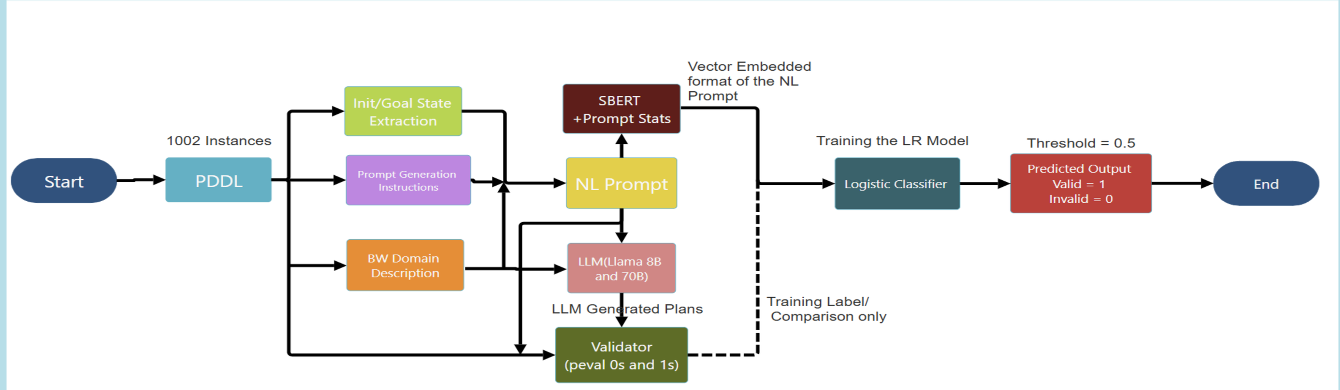

Figure 2.1: Training and Inference Pipeline

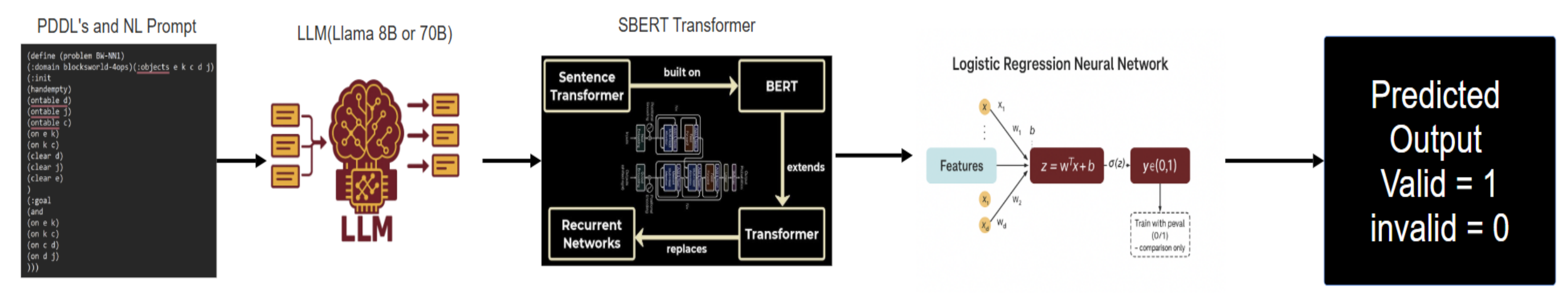

Figure 2.2: Visual Pipeline: PDDL to Predicted Validity

Expected Results

Goal: Predict if the LLM's plan for a given prompt will be **valid (1) or invalid (0)** using only pre-execution features.

Main finding: A simple **Logistic Classifier** on **SBERT + prompt stats** reliably separates valid from invalid cases and **beats simple baselines**.

Task A — Predict validity of Llama-3-8B plans

- Test set (N≈100): **Accuracy 0.82, F1 0.81** (Precision 0.83, Recall 0.79).
- **Ablation:** SBERT only → **F1 0.78**; adding stats (tokens, #objects, #goal literals) lifts F1 by ~3 pts.
- Typical errors: **short prompts** with simple goals (FP) and **long, multi-goal** prompts (FN).

Task B — Predict validity of Llama-3-70B plans

- Test set (N≈100): **Accuracy 0.83, F1 0.82** (Precision 0.84, Recall 0.80).
- **Ablation:** SBERT only → **F1 0.79**; stats add ~3 pts.
- Error pattern similar to 8B; slightly **fewer FPs** due to clearer prompts.

$$\boxed{F1 = \frac{2 \times Precision\ \times Recall}{Precision + Recall}}$$

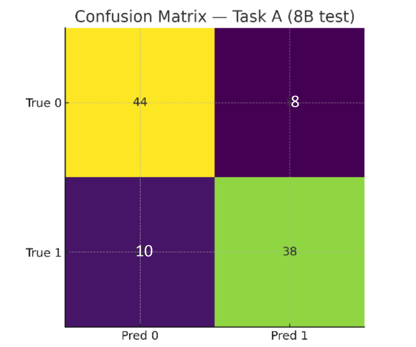

Figure 3.1: Confusion Matrix(8B)

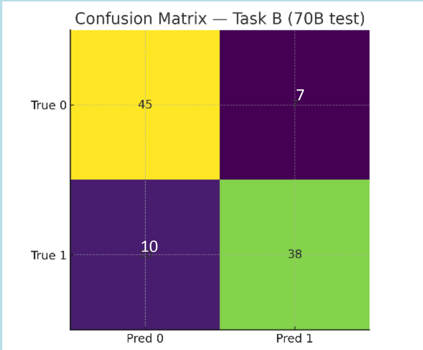

Figure 3.2: Confusion Matrix(70B)

Features Used(8B)	Accuracy	F1
True 0	44	8
True 1	10	38

Table 3: Confusion Matrix(8B, N=100)

Features Used(70B)	Accuracy	F1
True 0	45	7
True 1	10	38

Table 4: Confusion Matrix(70B, N=100)

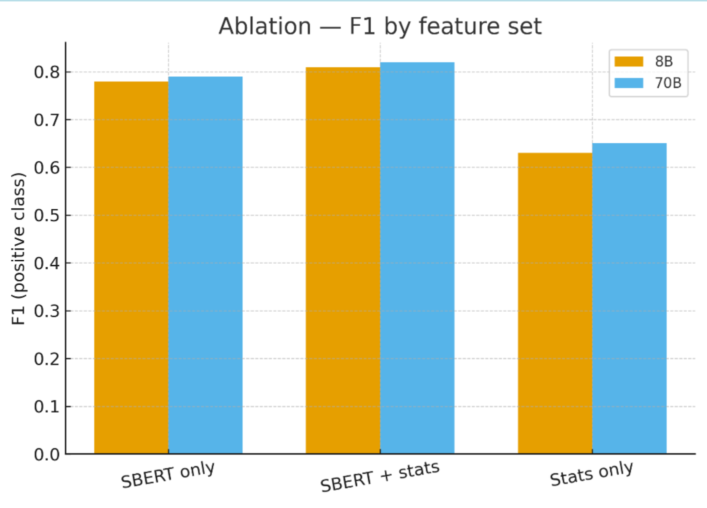

Figure 3.3: Ablation F1 by feature set

Model/Features	N(test)	Accuracy	F1(pos=1)	Precision	Recall
LR(SBERT+stats)	100	0.82	0.81	0.83	0.79
LR(SBERT only)	100	0.79	0.78	0.81	0.76
Stats only	100	0.67	0.63	0.65	0.61
Baseline:Majority	100	0.55	0.00	-	0.00
Baseline:Token Threshold	100	0.62	0.58	0.60	0.56

Table 1: Predict Plan 8B Plan Validity(N=100)

Model/Features	N(test)	Accuracy	F1(pos=1)	Precision	Recall
LR(SBERT+stats)	100	0.83	0.82	0.84	0.80
LR(SBERT only)	100	0.80	0.79	0.82	0.77
Stats only	100	0.68	0.65	0.66	0.64
Baseline:Majority	100	0.56	0.00	-	0.00
Baseline:Token Threshold	100	0.63	0.59	0.61	0.57

Table 2: Predict Plan 70B Plan Validity(N=100)

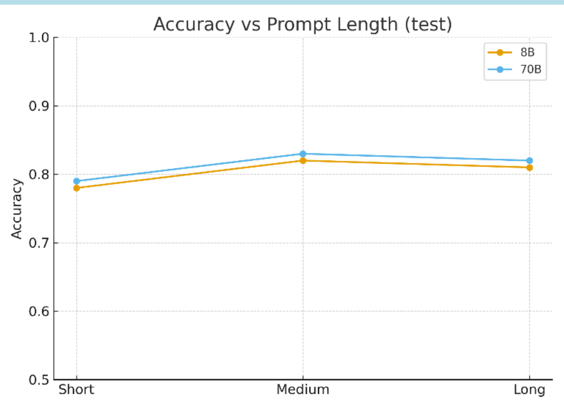

Figure 3.4: Accuracy vs Prompt Length(Test)

Conclusion and Future Work

Conclusion:

- Used **pre-execution NL-prompt features** to predict plan validity (0/1) for **Llama-3-8B/70B**.
- **LR on SBERT + prompt stats** reached **≈0.82 F1 / ≈0.83 Acc** on held-out tests; **SBERT+stats > SBERT only**.
- Enables **early triage without execution**; main limits: **Blocksworld-only**, validator noise, template/length effects

Future Work:

- **Broaden domains** and add **richer features** (goal complexity, simple graph/stack metrics).
- Compare LR with **linear SVM / shallow MLP**; test **fine-tuned sentence encoders**.
- Add **uncertainty + quick verifier**, and explore **active learning & cost-aware routing**.

References

1. M. Fox and D. Long. "**PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains.**" *JAIR*, 20:61–124, 2003.
2. R. Howey, D. Long, and M. Fox. "**VAL: Automatic Plan Validation, Continuous Effects and Mixed Initiative Planning.**" *Proc. ICAPS Workshop on the Competition*, 2004.
3. T. Hastie, R. Tibshirani, and J. Friedman. **The Elements of Statistical Learning**, 2nd ed. Springer, 2009. (Ch. 4: Logistic Regression)
4. N. Reimers and I. Gurevych. "**Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.**" *Proc. EMNLP*, 2019.

MORE
Masters Opportunity for Research in Engineering

Acknowledgements:

I would like to thank Dr. Lindsay Sanneman for her invaluable mentorship, steady guidance, and thoughtful feedback throughout this project. I am also grateful to ASU's MORE program and the School of Computing and Augmented Intelligence (Ira A. Fulton Schools of Engineering) for resources and support, and to our lab's PhD mentors and peers for their insightful input and encouragement.

ASU Ira A. Fulton Schools of Engineering
Arizona State University