Take Home Assignment 2
Member #1: Arya Bhavesh Shah
Roll No: 25290002
arya.shah@iitgn.ac.in

Designing and Evaluating Cognitive and Language Models

Question 1: Design a Continuous Interaction Model

Design a model that can interact continuously with a dynamic environment and learn object associations over time, basically going from a stationary model like LLMs to more real like human -like learning technique. The model should not just passively receive data, but actively explore, associate, and adapt based on its interactions.

- Describe the model structure in terms of units, architecture
- Specify how the model would learn.
- Explain how the model handles continuous interaction.

There are no "right" or "wrong" answers. The goal is to design a conceptually plausible system inspired by principles we have discussed in class so far.

During our lectures, we observed and identified key issues in static/stationary systems such as Vision Models, LLMs. With the goal to develop a model that can interact continuously with a dynamic environment and learn object associations over time, it is crucial to include elements from Neuroscience and understand how humans learn and try to replicate phenomena, and learning techniques.
I explored literature from Arxiv, Nature and ScienceDirect to support my claims related to the model designed by me for this problem statement. Here's the proposed model that can pave the way to better continuous interaction and learning reaching a step closer to AGI, perhaps.

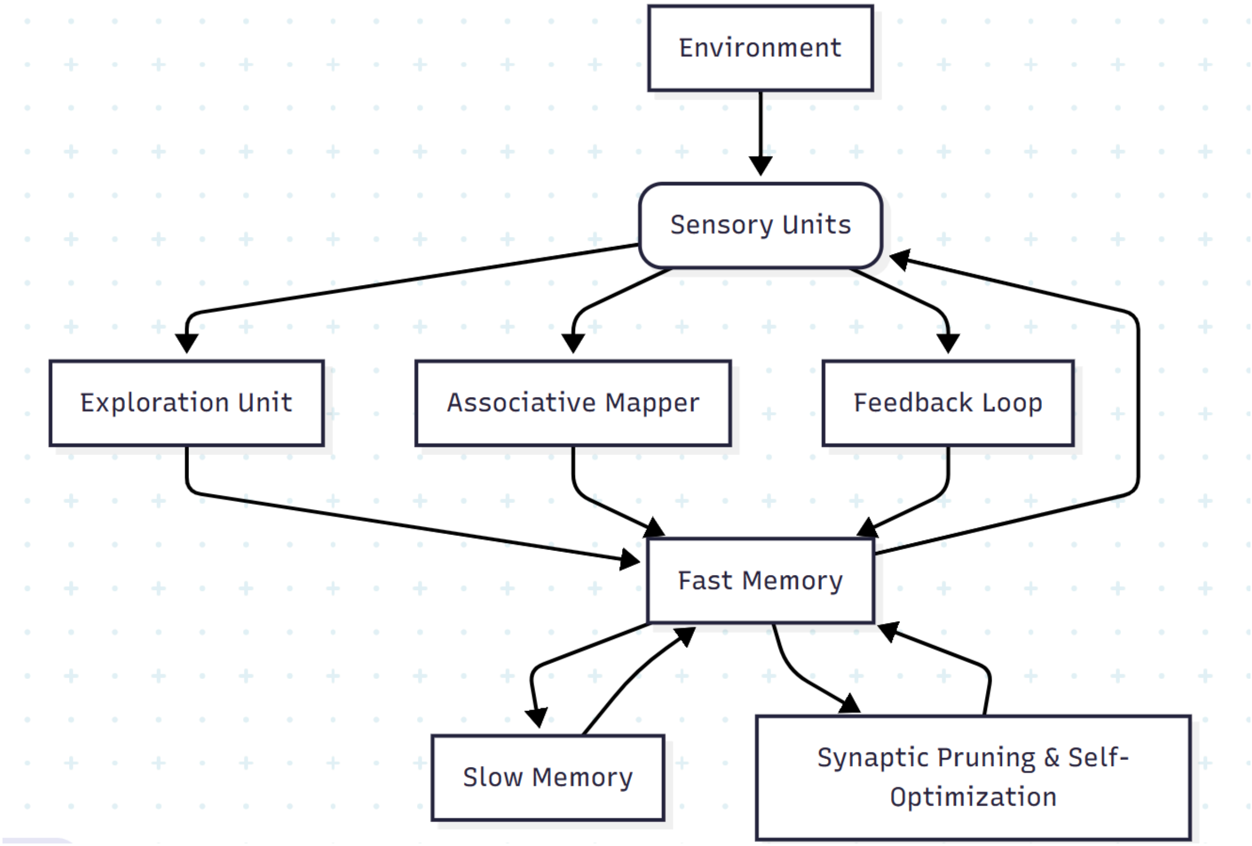
Model Overview: Units and Architecture

The model proposed can be called a Continuous Associative Interaction Network which is designed to interact, learn, and adapt within a dynamic environment, learning object associations over time. Unlike stationary models (e.g., LLMs), the proposed model is structured for continuous and active engagement, similar to how a human learns through exploration.

Model Units and Components

- Sensory Units: Receive multi-modal input (visual, auditory, proprioceptive) from environment and actions
- Exploration Unit: Selects actions to maximize novelty and reduce prediction errors
- Fast Memory System (Working, Episodic): Rapid storage and recall; highly plastic, responsible for short-term adaptation (like hippocampus)
- Slow Memory System (Semantic, Long-Term): Gradual consolidation, stores core associative structure (like neocortex)
- Associative Mapper: Binds object identity and context using positional and contextual information
- Self-Optimization/Synaptic Pruning: Prunes redundant connections, inspired by Hebbian rules and synaptic plasticity
- Feedback/Prediction Loop: Enables continuous learning from mismatches between predicted and actual outcomes

This can be depicted visually as follows:



Model Learning Mechanism

How does this model learn?

The Continuous Associative Interaction Network integrates active learning (selecting what to explore or ask next based on uncertainty), continuous real-time updating (on-the-fly model change), and associative binding (creating rich links between objects, context, and outcomes).

Key Learning Principles:

- Dual-Memory System: Inspired by neuroscience, Continuous Associative Interaction Network maintains a separation between episodic (fast, temporary) and semantic (slow, stable) memories, supporting lifelong adaptation. (Gupta et. al., Durstewitz et. al.)
- Active Learning and Neural Architecture Search: It actively chooses informative experiences/actions and adapts its neural architecture over time for each learning phase. (Tang et. al., Geifman et. al)
- Associative Binding & Contextual Learning: Objects are represented in networks that bind their identity with spatial/contextual info. Human-like associations are crucial for building robust, generalizable knowledge. (Quek et. al., Contier et. al)
- Neuroplasticity Mechanisms: Synaptic self-optimization and pruning avoid catastrophic forgetting and maintain efficiency. (Gupta et. al., Durstewitz et. al.)
- Hierarchical Predictive Coding: Continuous prediction and error-correcting feedback drive learning, with model layers updating at different speeds. (Dovrolis)

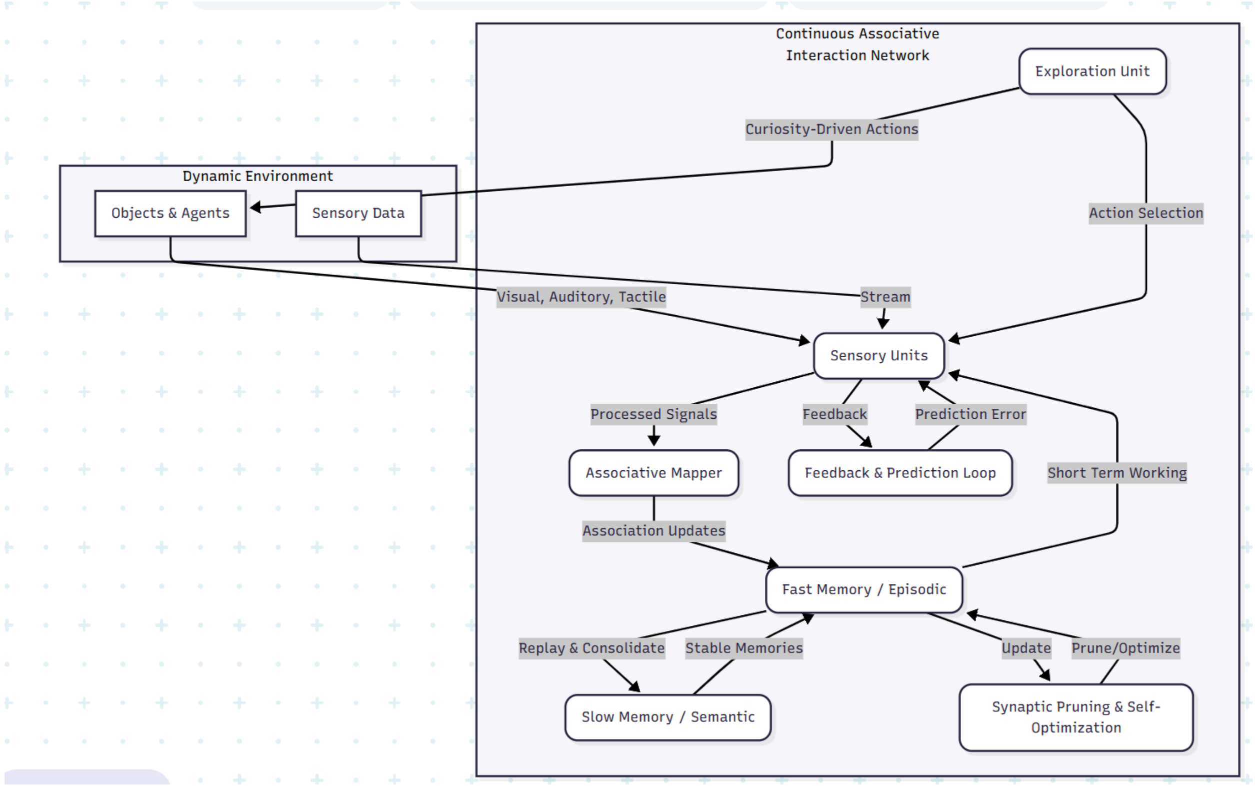
How does this model continuously interact?

- Embodied Sensory Loops: Constantly receives feedback from actions (touch, vision, sound, etc.)
- Active Exploration: Decides what to observe or manipulate to maximize information gain, unlike LLMs that only passively receive text.
- Online Adaptation: Memories and model structure are updated on-the-fly based on experience, avoiding "catastrophic forgetting."
- Memory Replaying: Previously learned associations are periodically replayed and consolidated.

Now, we compare the models such as LLMs, RL and our own proposed model in the following table:

Feature / Model	LLM (Stationary)	Reinforcement Learning	Continuous Associative Interaction Network (Proposed)
Environment interaction	None (static data)	Episodic/goal-based	Continuous sensorimotor
Association learning	Fixed, via text tokens	Rewarded associations	Contextual, multi-modal
Adaptation	After offline retraining	Some, via episodic RL	Real-time, online, lifelong
Active exploration	No	Yes, goal-directed	Yes, curiosity-driven
Catastrophic forgetting	Yes, brittle	Frequent, mitigated by replay	Mitigated via dual-memory/pruning
Biological inspiration	Minimal	Partial	High (dual-memory, plasticity, etc.)

Detailed Architecture with Self-Critique?



Here’s the working of the model in greater and finer detail:

- Environment sends rich, multi-modal data to Continuous Associative Interaction Network
- Sensory Units process signals and trigger both exploration and association mapping
- Exploration Unit determines actions to maximize learning and novelty
- Fast memory allows quick adaptation to new associations; slow memory consolidates long-term structure
- Associative mapping binds sensory input to context and keeps learning robust
- Feedback and prediction loop drive learning by acting on surprises/prediction errors
- Self-optimization prunes inefficient pathways, ensuring scalability

Critique

1. The architecture is consistent with findings in dual-memory systems, plasticity, and predictive coding in neuroscience and is echoed in modern proposals for truly adaptive AI. (Contier et. al, Gupta et. al, Durstewitz et. al, Dovrolis)
2. Compared to typical AI, CAIN's continuous, curiosity-driven, and active learning resembles what is necessary for generalization and robust association learning in humans. (Quek et. al, Contier et. al)

3. Potential challenges: Online stability, computational/memory scaling, and transfer to physical robotic platforms; but mechanisms such as synaptic pruning and hierarchical coding are promising mitigations. (Gupta et. al, Dovrolis)
4. The model is easily extensible towards embodied AI for real-world robots, where environmental engagement and lifelong adaptation are necessary. (Liu et. al, Fung et. al)

Question 2: Ablation Study on LLAMA Model

Perform an ablation study on an LLM (LLAMA, GPT, others) model to investigate which components are critical for performance and where the model breaks.
You can choose a set of ablation techniques (e.g., removing attention heads, zeroing out layers, masking input embeddings). Define simple downstream tasks (e.g., next-word prediction, sentence completion, sentiment classification). Measure performance drop or failure point as more components are ablated.
Please highlight tables or graphs showing the effect of ablation on model performance. Also discuss the insights gained (e.g., redundancy in architecture, sensitivity of semantic vs. syntactic processing).

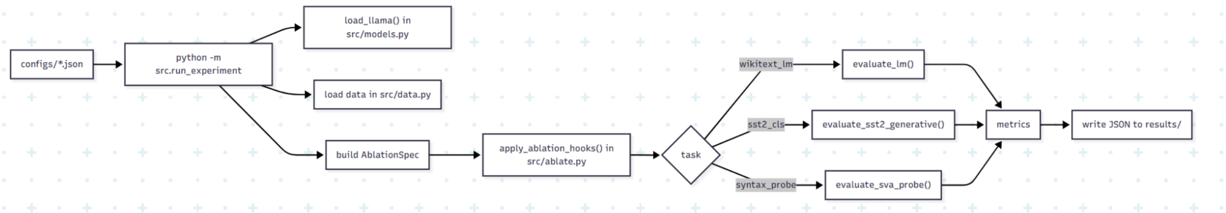
In this section we show how we conducted the ablation study on a LLaMA-family causal language model, covering model details, datasets and tasks, ablation mechanisms, implementation specifics, experimental design, and the rationale behind each ablation.

Model Choice, Tasks & Datasets

Model Name: TinyLlama/TinyLlama-1.1B-Chat-v1.0
We evaluated three representative tasks covering language modeling, classification, and syntax probing:

1. WikiText-2 (Next-Token Prediction)
    - Metric: Perplexity (exp of mean negative log-likelihood).
2. GLUE SST-2 (Sentiment Classification)
    - Metric: Accuracy on validation split.
    - Generative forced-choice between “positive” and “negative”.
3. Subject–Verb Agreement (Syntax Probe)
    - Metric: Accuracy and log-probability margin.
    - Minimal-pair evaluation of correct vs. incorrect verb forms.

Below is a High-Level Overview of the experiment suite for this task:



Ablations

Ablations were applied temporarily using forward hooks, ensuring the model itself remained unchanged.

1. Attention Head Ablation
- Removes contributions of selected attention heads at the projection layer.
- Tests redundancy at the head level.
2. Layer/Submodule Zeroing
- Targets entire transformer layers or specific components:
    - attn → attention output
    - mlp → MLP output
    - block → entire block output
    - Probes importance of interaction vs. transformation pathways.
3. Embedding Masking
- Zeroes out a fraction of embedding dimensions.
- Assesses representational redundancy at the input layer.
4. Baseline Control
- No ablation applied; serves as reference.

Rationale Behind Ablations

- Heads: Tests tolerance to head pruning and its threshold for collapse.
- Layers: Separates the role of attention, MLP, and full blocks in maintaining performance.
- Embeddings: Examines redundancy in input token representations.
- Syntax Probe: Assesses whether grammatical sensitivity relies on specific subcomponents.

Experiment Matrix

Ablation Type	Variables Tested	Tasks	Explanation
Heads (Attention Head Ablation)	Severities: [0.0, 0.25, 0.5, 0.75, 1.0]	LM, CLS	Heads are sub-parts of attention layers. Severity means the fraction of heads removed (e.g., 0.5 = 50% heads zeroed out). LM = Language Modeling (predicting next word, measured by perplexity). CLS = Classification (e.g., sentiment classification, measured by accuracy).

Layers (Layer Zeroing Ablation)	Components: {attn, mlp, block} at severities [0.0 … 1.0]	LM, CLS, Syntax	Layers are stacked blocks in the Transformer. Three parts tested separately: attn = attention sublayer, mlp = feed-forward sublayer, block = whole layer. Zeroing means setting outputs to zero. Syntax = Grammar probe (tests if the model still encodes grammar).
Embeddings (Embedding Masking)	Severities: [0.0, 0.25, 0.5, 0.75, 1.0]	LM, CLS	Embeddings are the vector representations of words. Masking means removing (zeroing out) a fraction of embedding dimensions. Higher severity = more dimensions masked.

Ablation Results

Baseline Performance

- Sentiment Classification (SST-2): Accuracy = 0.7741
- Syntax Probe (Subject–Verb Agreement): Accuracy = 1.0000
- Language Modeling (WikiText-2): Perplexity = 13.2298

These baselines serve as reference points for evaluating degradation under ablations.

Attention Head Ablations

- SST-2 (CLS): Accuracy declines steadily with severity, dropping from 0.77 → 0.51 at full ablation.
- WikiText-2 (LM): Perplexity rises sharply: ~18 at 25% severity but >1700 at full ablation, showing catastrophic loss of modeling quality.

Layer Zeroing Ablations

We tested three components: attention (attn), MLP, and the entire block.

- SST-2:
  - *Attn/MLP:* Performance drops toward chance level by 50–75%.
  - *Block:* Accuracy falls immediately to ~0.49 even at mild ablation.
- Syntax Probe:
  - *Attn:* Gradual decline (1.0 → 0.0 across severities).
  - *MLP:* Severe degradation beyond 50% (to ~0.2 at full ablation).
  - *Block:* Accuracy collapses to 0 even at 25% ablation.
- WikiText-2:

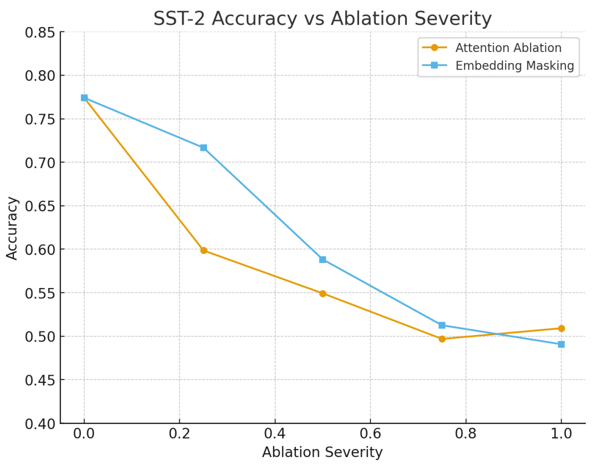
  - *Attn:* Perplexity grows from 13 → ~1768 by full ablation.
    MLP: Rapid blow-up (≈10k at 50%, ≈258k at full).
  - *Block:* Extreme brittleness, perplexity explodes to >31k at just 25%.

Embedding Masking Ablations

- SST-2: Accuracy shows resilience up to 25% masking (0.72), but steep drops appear beyond 50% (0.49 at full).
- Suggested embeddings have some redundancy, but representational capacity collapses once most dimensions are zeroed.

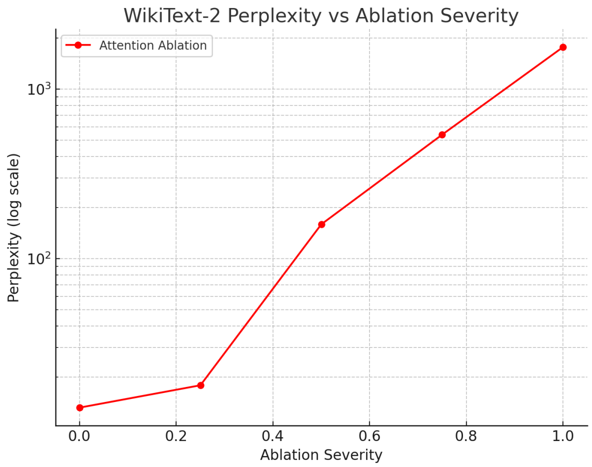
Key Insights

1. Redundancy vs. Fragility: Attention heads tolerate mild pruning (≤25%), but LM performance collapses beyond 50%.
2. Layer Importance: CLS and Syntax tasks reveal strong dependence on block integrity. Attn and MLP contribute differently: attn ablation impairs grammar, while MLP ablation destabilizes semantics.
3. Embedding Robustness: Input representations have limited redundancy; masking ≥50% dimensions sharply reduces accuracy.
4. Syntax Sensitivity: Syntax probe is extremely brittle to block ablations, confirming reliance on integrated pathways for grammatical agreement.
5. Seed Variability: At low severities, outcome variance across random seeds suggests non-uniform importance across layers.

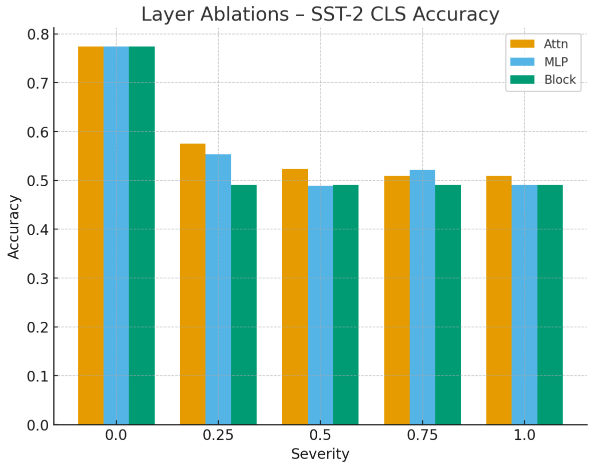


- Shows how sentiment classification accuracy drops as we increase the ablation severity.

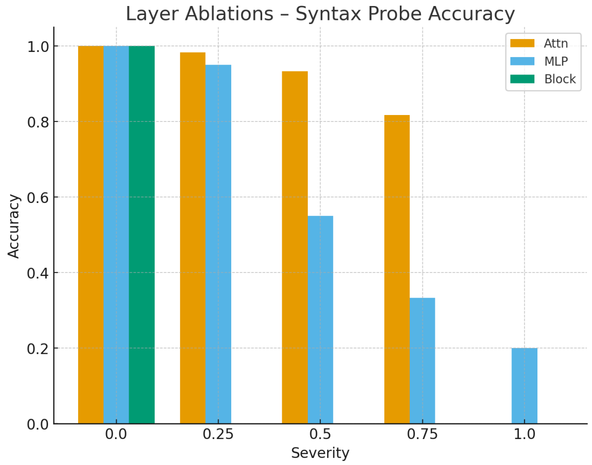
- Both attention head ablation and embedding masking reduce performance, but attention ablation harms accuracy much faster



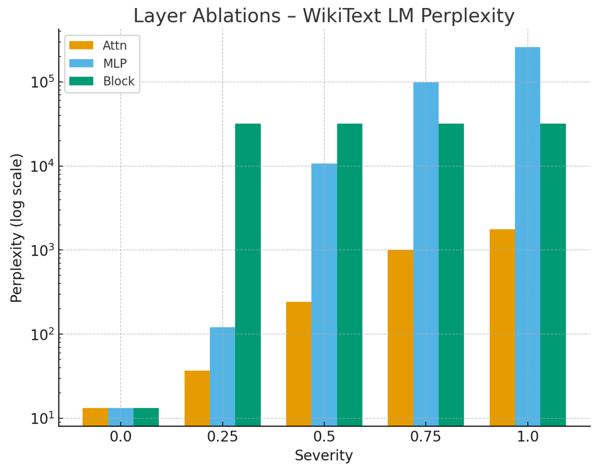
- Shows how perplexity (lower is better) increases with more attention heads removed.
- The rise is exponential, which is why we plotted it on a log scale.
- This means that language modeling is extremely sensitive to attention head removal.



- Accuracy drops sharply as severity increases.
- Block zeroing kills performance fastest (stuck near ~49%).
- Attn and MLP degrade more gradually, but still fall below chance level.



- Block zeroing → accuracy instantly collapses to 0 even at 25% severity.
- MLP → accuracy drops steadily, showing syntax depends partly on feed-forward layers.
- Attn → gradual decline, but total collapse only at 100% severity.



- Block zeroing explodes perplexity to ~32,000 immediately.
- MLP → grows exponentially (10^5+ by 0.75 severity).
- Attn → rises fast but less catastrophic compared to block/MLP.

These visuals make it clear that:

- CLS task is most sensitive to block zeroing.
- Syntax is brittle to block ablation, resilient only with partial attn/MLP.
- Language modeling is extremely fragile—block/MLP ablations cause exponential loss.

Question 3: Syntax vs. Semantics Analysis in Open-Source Language Models (Optional - Extra Credit)

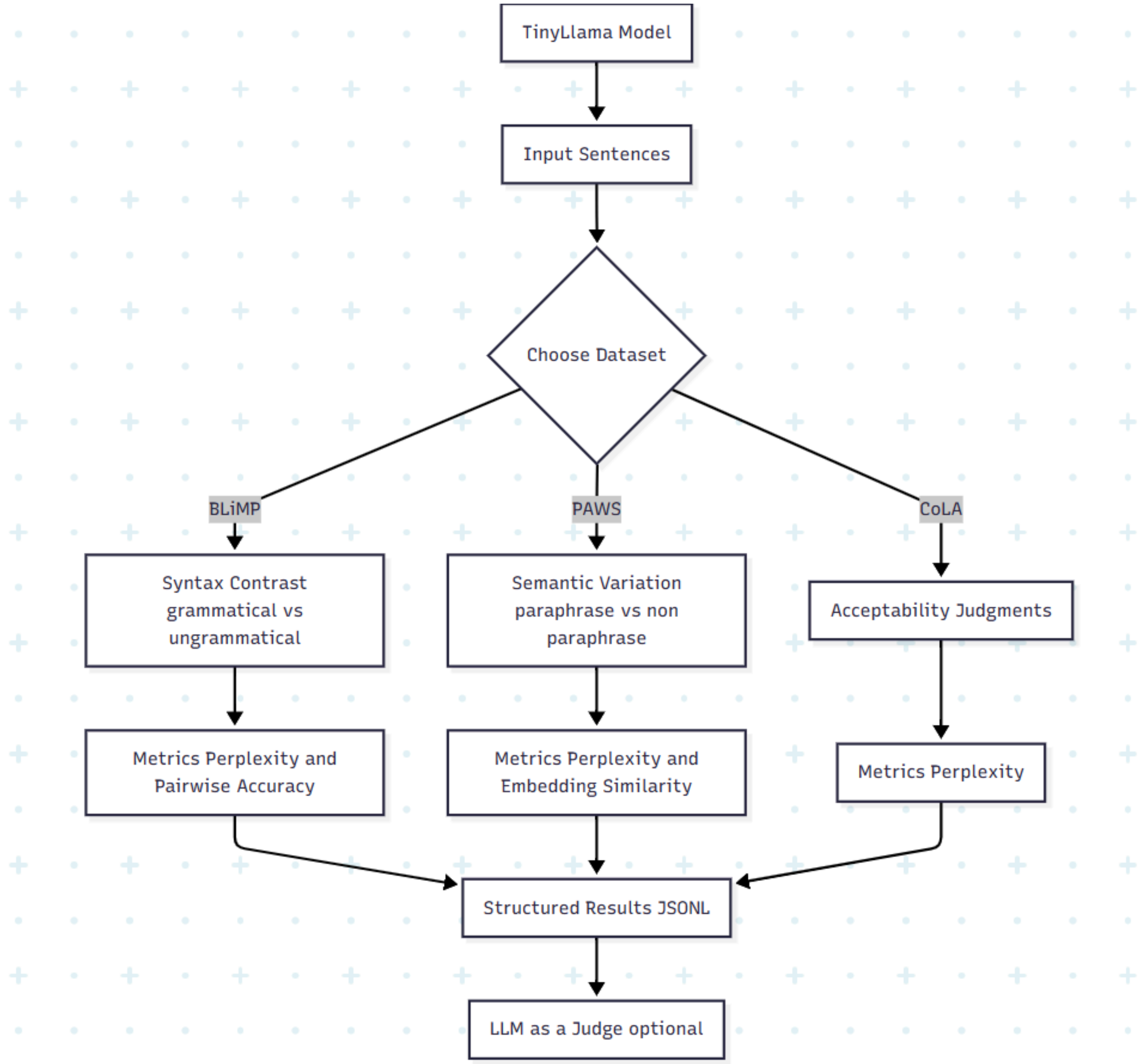
Choose an open-source language model (e.g., LLAMA, GPT-2, GPT-J, BLOOM) and design an experiment to test how well it differentiates between syntactic structure and semantic meaning.

You can create or identify datasets that systematically vary syntax while keeping meaning constant, and vice versa (e.g., syntactically correct but semantically nonsensical sentences). Propose a methodology to evaluate the model's sensitivity to syntax and semantics (e.g., perplexity scores, prediction accuracy, embedding similarity).

A brief report explaining your experimental design, expected outcomes, and interpretation of how syntax and semantics are represented in the model.

In this section, we document a controlled evaluation of TinyLlama for distinguishing syntactic structure from semantic meaning. The study uses three complementary datasets (BLiMP, PAWS, CoLA) and three metrics (perplexity, pairwise probability accuracy, and embedding similarity). All raw results are written to structured JSONL files, and a lightweight LLM-as-a-Judge is provided using a Hugging Face instruction-tuned model.

Below is the depiction of the overall experiment flow:


TinyLlama Model
Input Sentences
Choose Dataset
BLiMP
PAWS
CoLA
Syntax Contrast grammatical vs ungrammatical
Semantic Variation paraphrase vs non paraphrase
Acceptability Judgments
Metrics Perplexity and Pairwise Accuracy
Metrics Perplexity and Embedding Similarity
Metrics Perplexity
Structured Results JSONL
LLM as a Judge optional

Why choose TinyLlama: TinyLlama is small enough for reproducible experiments, while still powerful enough to demonstrate modern language model behavior.

Datasets used

250 samples each of:

1. BLiMP (Benchmark of Linguistic Minimal Pairs)
    a. Focus: syntax. Each pair has a grammatical vs. ungrammatical sentence. The model should give higher probability to the grammatical option.
2. PAWS (Paraphrase Adversaries from Word Scrambling)

   a. Focus: semantics under syntactic variation. Sentences may look very similar (high word overlap) but differ in meaning. The task is to tell apart true paraphrases from non-paraphrases.

3. CoLA (Corpus of Linguistic Acceptability)
   a. Focus: acceptability judgments. Each sentence is labeled as acceptable or not. Lower perplexity should align with acceptability.

Metrics used

1. Perplexity (PPL)
   a. Measures how confidently the model predicts a sentence. Lower perplexity = more confident and usually more grammatical.
2. Pairwise probability accuracy
   a. For sentence pairs (e.g., BLiMP), checks whether the model assigns higher probability to the correct/grammatical option.
3. Embedding similarity
   a. Compares sentence embeddings using cosine similarity. High similarity means the model views the sentences as meaningfully close.

Experiment Design

- Syntax-only contrasts (BLiMP)
  - Tests if the model consistently prefers grammatical sentences across phenomena.
- Semantics under syntax variation (PAWS non-paraphrases)
  - Sentences with similar wording but different meanings. A good model should recognize the difference.
  - Meaning preserved, syntax varied (PAWS paraphrases)
  - Sentences that look different but mean the same. Embeddings should remain similar, showing semantic robustness.
- Acceptability (CoLA)
  - Checks whether lower perplexity aligns with grammatical acceptability.

Results & Insights

- BLiMP: TinyLlama shows sensitivity to syntax, generally favoring grammatical sentences.
- PAWS: The model can separate paraphrases from non-paraphrases, though high word overlap sometimes confuses it.
- CoLA: Acceptable sentences tend to have lower perplexity, but the signal is imperfect.
- Syntax vs. semantics tradeoff: The model shows stronger performance on surface acceptability (syntax) than on subtle semantic distinctions.
- Redundancy vs. brittleness: TinyLlama's embeddings capture meaning, but are brittle when syntax disrupts word order.
- Practical implications: While useful for grammaticality judgments, the model is less reliable when distinguishing nuanced meanings.

BLIMP Dataset Examples

Example ID	Input A	Input B	Model Decision	Model Correct	Judge Rating	Judge Rationale
adjunct_island:228	Who did Emily leave without curing Alan?	Who did Emily leave Alan without curing?	b	0	semantics	The second sentence has a clearer semantic structure with fewer syntactic ambiguities.
adjunct_island:69	Who has Rebecca fled from without listening to Kirsten?	Who has Rebecca fled from Kirsten without listening to?	b	0	semantics	The correct answer should have a subject-verb-object structure, which is present in B but not in A.
adjunct_island:67	Who would Jesus hug without discovering Bradley?	Who would Jesus hug Bradley without discovering?	b	0	semantics	The second sentence has a clearer semantic flow with the phrase 'without discovering' placed logically.

CoLA Dataset Examples

Example ID	Input Sentence	True Label	NLL Score	PPL Score	Judge Rating	Judge Rationale
cola:validation:163	Jessica loaded boxes on the wagon.	1 (Grammatical)	4.638	103.298	syntax	The score difference between NLL and ppl suggests the model focused more on syntax.
cola:validation:159	You've really lived it up.	1 (Grammatical)	4.076	58.914	semantics	The phrase is ambiguous and could be interpreted in various ways without clear syntax.
cola:validation:31	The tub leaked water.	1 (Grammatical)	6.268	527.402	semantics	The word 'tub' is a common noun and does not require complex syntax to understand its meaning.

PAWS Dataset Examples

Example ID	Input A	Input B	True Label	Model Decision	Embedding Cosine	Judge Rating	Judge Rationale

paws:50	It was founded in 1982 by Samuel Lipman, a former art critic...	It was founded in 1982 by Hilton Kramer, former art critic...	0 (Not Paraphrases)	b	0.9739	syntax	Input B has consistent order of names and roles, while Input A has mixed roles which is less clear.
paws:115	Windows XP Mode runs Windows XP on a separate computer...	Windows XP Mode runs Windows XP in a separate machine...	1 (Paraphrases)	a	0.9347	syntax	The second sentence has clearer syntax with correct word order and fewer grammatical errors.
paws:166	It was an integral part of Assyria from perhaps the 25th century BC...	It was an integral part of Assyria from perhaps the 7th century BCE...	0 (Not Paraphrases)	a	0.9896	syntax	The year range in Input B is reversed compared to Input A, which changes the historical context significantly.

- BLIMP: Model struggled with syntax and fell back on semantic reasoning (all wrong)
- CoLA: Model used mixed strategies depending on the sentence type
- PAWS: Model relied heavily on surface-level syntactic features rather than true meaning

Accompanying Code Present Here: Google Drive Link
THANK YOU :)