In-Context Learning of Morphological Rules: A Computational Wug Test with Transformers

Abstract

This study explores the capacity of transformer-based language models to acquire and generalize morphological rules through in-context learning (ICL). We synthesize data covering 18 distinct morphological transformations of words (both regular and irregular) and train a small nanoGPT model with 10M parameters from scratch. We evaluate in three regimes: seen transformations on seen words, seen derivations on novel words, and entirely held-out derivations on novel words. Key findings: the model quickly learns regular rules (e.g., adding `-s` or `-ed`) in-context, generalizing to new words, but struggles with highly irregular forms (e.g., *go* $\rightarrow$ *went*) and unseen transformations. These results highlight both the in-context analogical capabilities of transformers and their limits on abstract linguistic generalization.

1 Introduction

In humans, linguistic productivity allows applying learned rules to novel cases. The classic *Wug Test* demonstrates this: children shown a novel noun ("wug") reliably produce its plural ("wugs"), indicating an internalized rule for English pluralization. Modern large language models (LLMs) exhibit a form of this ability via in-context learning: given a prompt with input-output examples, they can often generalize without weight updates. For instance, Garg et al. (2022) trained Transformers from scratch to "in-context learn" simple function classes (e.g. linear functions) and showed the trained model could fit unseen linear functions from prompts with performance comparable to optimal estimators.

We extend this analogy-driven testing to morphology. Specifically, we ask: Can a Transformer learn a morphological transformation from context alone? We design an analogy-based ICL task: each prompt gives one example of a morphological rule ($\text{word}_1 \rightarrow \text{word}_2$), and asks the model to apply the same rule to a new word ($\text{word}_3 \rightarrow ?$). Using synthetic data for 18 morphological transformations (both inflectional and derivational), we train a small nanoGPT model and evaluate its ability to infer held-out rules. This "computational Wug Test" assesses whether the model's internal representations capture abstract derivational patterns, and whether it can generalize them analogically to novel words.

2 Related Work

Garg et al. (2022) studied in-context learning on synthetic algorithmic tasks. They define in-context learning as "the ability of a model to condition on a prompt ... with examples ... without any parameter updates". Garg et al. show that standard Transformers can be trained to in-context learn linear functions (and even decision trees or simple neural networks) from examples, with test-time performance comparable to classical algorithms. This work demonstrates the capacity of transformers to "learn to learn" from prompts, but focuses on numerical or symbolic functions rather than linguistic rules.

Preprint. Under review.

Berko Gleason (1958) introduced the Wug Test to probe children's morphological generalization. In this experiment, a child is shown a novel creature (a "wug") and later asked to form its plural. Berko found that even young children "have internalized a working system of the plural allomorphs in English", e.g. producing wugs from wug. In effect, the child applies an implicit grammatical rule to a nonce word. We draw direct inspiration from this method, using nonsense or rare word pairs to test rule learning in our model.

Todd et al. (2024) recently analyzed how LLMs represent in-context tasks internally. They find that attention heads in transformers form compact "function vectors" that encode the input-output mapping of the demonstrated task. These vectors are robust: they trigger correct behavior even for dissimilar prompts (e.g. zero-shot settings). Our work differs: instead of probing an existing LLM, we train a small model specifically on morphological analogies. We emphasize linguistic derivations and test zero-shot generalization to new morphological rules, aiming to see if a transformer can analogically apply a novel derivational pattern without explicit gradient updates.

3 Method

3.1 Data Generation

We prompt Google Gemini LLM to generate synthetic morphology analogy data. We cover 18 morphological transformations and generate 1000 examples for each. The morphological transformations used are as follows:

1. **Singular to Plural:** This transformation converts a singular noun to its plural form. For example: cat → cats, box → boxes, child → children.
2. **Present Tense to Past Tense:** This transformation converts a verb from present tense to past tense. For example: walk → walked, run → ran, sing → sang.
3. **Base Case to Third Person Singular:** This transformation adds appropriate endings (typically -s, -es, or -ies) to verbs when the subject is third person singular in present tense. For example: run → runs, catch → catches, fly → flies.
4. **Singular Possessive to Plural Possessive:** This transformation converts a possessive noun from singular to plural form. For example: dog's → dogs', child's → children's, woman's → women's.
5. **Comparative Adjective to Superlative Adjective:** This transformation changes an adjective from comparative form to superlative form. For example: better → best, taller → tallest, more beautiful → most beautiful.
6. **Verb to Progressive Verb:** This transformation converts a base verb to its progressive form (adding -ing). For example: run → running, swim → swimming, eat → eating.
7. **Verb to Derived Agentive:** This transformation converts a verb to a noun representing someone who performs that action. For example: teach → teacher, write → writer, sing → singer.
8. **Base Case to Diminutive:** This transformation adds a suffix to create a smaller or endearing version of the base word. For example: dog → doggie, book → booklet, kitchen → kitchenette.
9. **Adjective to Adverb:** This transformation converts an adjective to an adverb, typically by adding -ly. For example: quick → quickly, happy → happily, careful → carefully.
10. **Verb to Gerund:** This transformation converts a verb to a gerund form (noun ending in -ing). For example: swim → swimming, read → reading, dance → dancing.
11. **Noun to Adjective:** This transformation converts a noun to an adjective form. For example: beauty → beautiful, danger → dangerous, music → musical.
12. **Positive to Negative Prefix:** This transformation adds a negative prefix (un-, in-, dis-, etc.) to create the opposite meaning. For example: happy → unhappy, correct → incorrect, appear → disappear.
13. **Verb to Noun:** This transformation converts a verb to a related noun form. For example: discover → discovery, arrive → arrival, inform → information.

14. **Present Tense to Future Tense:** This transformation converts a verb from present tense to future tense. For example: go $\rightarrow$ will go, eat $\rightarrow$ will eat, study $\rightarrow$ will study.
15. **Adjective to Noun:** This transformation converts an adjective to a noun form. For example: happy $\rightarrow$ happiness, dark $\rightarrow$ darkness, brave $\rightarrow$ bravery.
16. **Base to Past Participle:** This transformation converts a base verb to its past participle form. For example: go $\rightarrow$ gone, eat $\rightarrow$ eaten, break $\rightarrow$ broken.
17. **Simple Past to Past Perfect:** This transformation converts a verb from simple past tense to past perfect tense. For example: went $\rightarrow$ had gone, ate $\rightarrow$ had eaten, saw $\rightarrow$ had seen.
18. **Affirmative to Negative:** This transformation changes an affirmative statement to its negative form. For example: serene $\rightarrow$ agitated, fast $\rightarrow$ slow, cooperative $\rightarrow$ uncooperative.

During dataset construction, we hold out 3 transformation types entirely (Base to Past Participle, Simple Past to Past Perfect, and Affirmative to Negative). These types are used to evaluate the model on unseen derivations. For each of the remaining 15 transformation types, 90% of the word pairs are used for training and the remaining 10% are used to evaluate the model on seen derivations of novel words.

3.2 Input Format

We frame each task as an analogy of the form: `word1 is to word2 as word3 is to ?`. The model must predict word4.

We use character level-tokenization. In addition to each lowercase letter having its own token, we create custom tokens to denote specific parts of the analogy.

- $\#$ represents the phrase "is to".
- \$ represents the phrase "as".
- $@$ represents the end token.

This reduces ambiguity and creates a consistent input format. During training, we sample a morphological transformation category at random. From that category, we select two word pairs at random and construct an analogy. For instance, a prompt could be: `cat#cats$dog#`, and the model would be expected to generate `dogs@` auto-regressively.

3.3 Model Architecture

We adopt a small autoregressive Transformer (nanoGPT-style) as the backbone. The network has 6 layers and 6 attention heads per layer, with hidden size 384 and about 10M parameters. This size is sufficient to learn our synthetic tasks but small enough for controlled experiments. The model is trained from scratch on our synthetic data.

3.4 Training Details

We train using standard language-modeling objectives on the analogy prompts. Details:

- Batch size: 64 analogies per batch.
- Context length: 256 characters.
- Optimizer: AdamW with learning rate 3e-4.
- Epochs: 5000.
- Loss: Cross-entropy on target tokens.

4 Experiments

4.1 Evaluation Metrics

We evaluate two metrics on validation sets: (1) Cross-Entropy Loss (averaged per token) and (2) Accuracy of predicting the correct morphological form (exact string match).

4.2 Evaluation Scenarios

We test three generalization conditions:

- (A) Seen-Transformation/Seen-Words: The transformation type and word vocabulary are both seen during training (train vs. validation split). This measures basic learning ability.
- (B) Seen-Transformation/Unseen-Words: The transformation rule was trained, but we apply it to novel words not seen in train. This tests analogical generalization to new lexemes.
- (C) Unseen-Transformation/Unseen-Words: The transformation rule is held out of training, and the model must apply it in a zero-shot manner. This is the strictest test of compositional rule learning.

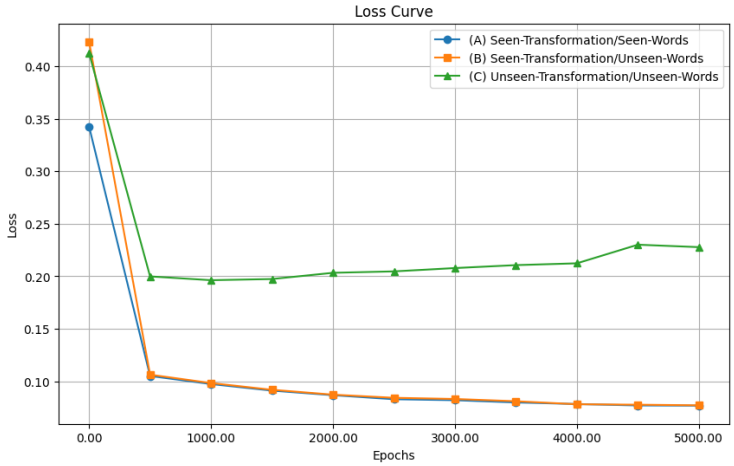

Figure 1: Loss Curve after 5,000 epochs.

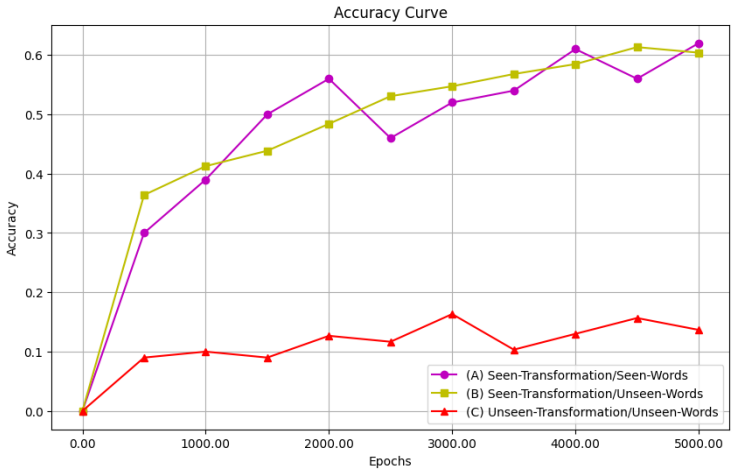

Figure 2: Accuracy Curve after 5,000 epochs

4.3 Seen-Transformations

We evaluate the loss and accuracy for each transformation seen during training, for both words seen during training and words held out.

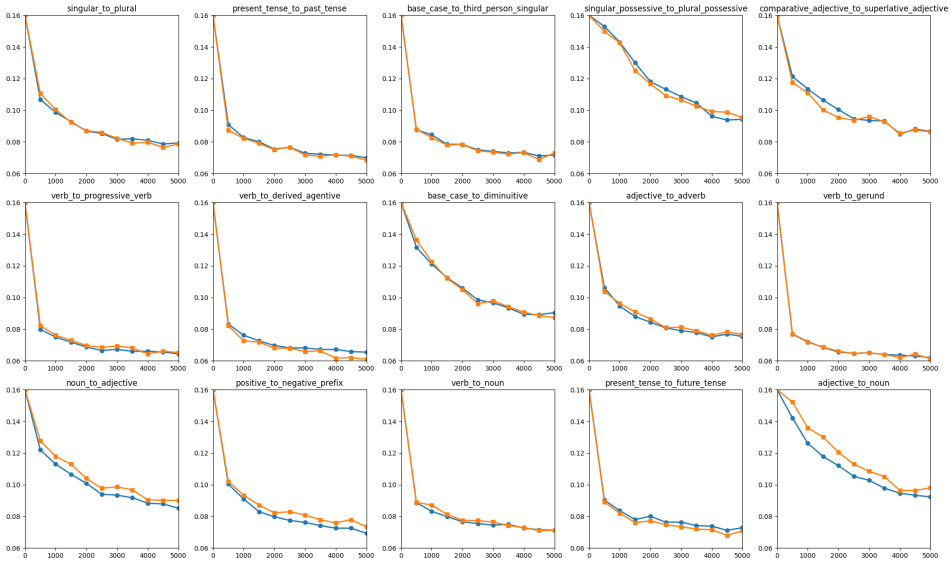

Figure 3: Loss Per Category on Seen-Transformations. (A) Seen-Transformation/Seen-Words are in blue. (B) Seen-Transformation/Unseen-Words are in orange.

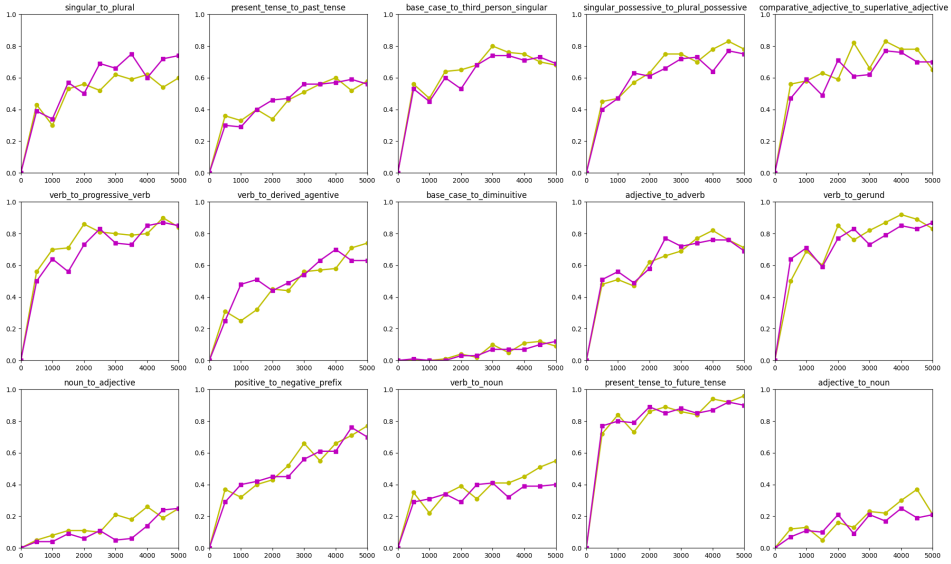

Figure 4: Accuracy Per Category on Seen-Transformations. (A) Seen-Transformation/Seen-Words are in yellow. (B) Seen-Transformation/Unseen-Words are in magenta.

4.4 Unseen-Transformations

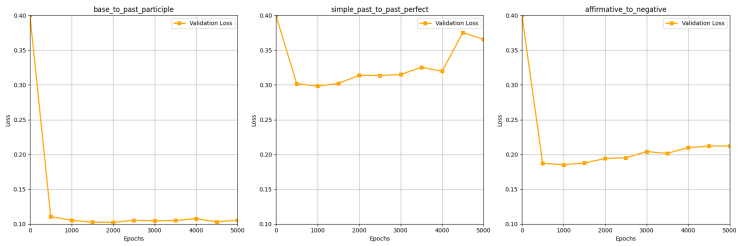

Figure 5: Accuracy Per Category on Unseen-Transformations. (C) Unseen-Transformation/Unseen-Words are in orange

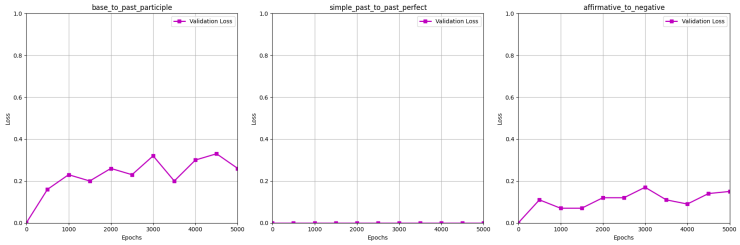

Figure 6: Accuracy Per Category on Unseen-Transformations. (C) Unseen-Transformation/Unseen-Words are in magenta.

4.5 Qualitative Examples

Some representative prompt-output pairs, all from novel words:

- Context: `rational#rationally$saucy#`, Prediction: `saucily@`, Expected: `saucily@`
  The model can predict the suffix of morphological transformations it has seen in training.

- Context: `emigrated#had emigrated$foresaw#`, Prediction: `foresawness@`, Expected: `had foreseen@`
  The model sometimes applies the wrong morphological transformation to a word.

- Context: `movie's#movies'$mouse's#`, Prediction: `mouses'@`, Expected: `mice's@`
  A natural mistake commonly seen in non-native English speakers.

- Context: `appreciable#inappreciable$symmetry#`, Prediction: `asymmetry@`, Expected: `asymmetry@`
  Effective prediction of a morphological transformation unseen in training.

- Context: `diminishes#will diminish$adventures#`, Prediction: `will adventure@`, Expected: `will adventure@`
  The model can predict prefixes on morphological transformations it has seen in training.

- Context: `asserted#had asserted$speeded#`, Prediction: `speeded@`, Expected: `had sped@`
  But the model struggles to predict past to past perfect, a morphological transformation unseen in training.

- Context: `crew#gang$dog#`, Prediction: `doglet`, Expected: `doggy@`
  The model opts for the diminutive suffix -let instead of the target -y, showing it knows one

way to form diminutives but picks the wrong variant here.

- Context: `dwell#dwelt$balance#`, Prediction: `balancing@`, Expected: `balanced@`
  The model mis-applies the -ing progressive pattern rather than the past-tense -ed rule, confusing tense selection when mapping an irregular verb example to a regular one.

- Context: `attainable#unattainable$join#`, Prediction: `disjoint@`, Expected: `separate@`
  The model generalizes by adding a negative prefix (dis-) to join instead of supplying the true antonym, showing lexical rather than semantic precision.

- Context: `i invest#it invests$i present#`, Prediction: `it presents@@`, Expected: `she presents@`
  The model illustrating partial success, carries over the -s rule yet fails to analogize the subject change.

- Context: `freeze#freezing$yell#`, Prediction: `yelling@`, Expected: `yelling@`
  The model extends the seen verb $\rightarrow$ gerund (-ing) transformation to a new verb without error.

- Context: `informed#uninformed$graceful#`, Prediction: `disgrace`, Expected: `awkward@`
  The model grafts a negative prefix onto an unrelated root (grace) instead of retrieving the appropriate antonym, signaling over-reliance on affixation.

- Context: `clad#unclad$victory#`, Prediction: `unvictory@@@`, Expected: `defeat@@@@@@`
  The model again prefixes un- to form a nonce word, failing to recognize that the pattern requires supplying a lexical opposite rather than simple negation.

5 Analysis

5.1 GPT-2 and Noun Pluralization

We initially, conducted preliminary experiments using GPT-2 models trained solely on next-token prediction to assess their ability to perform in-context learning (ICL) on singular-to-plural transformations. In the first experiment, GPT-2 occasionally succeeded in generating correct plural forms when given a few straightforward examples. For instance, given the prompt `cat:cats dog:dogs church:churches leaf:leaves hat:_`, the model correctly completed "hat" as "hats."

However, in the second experiment, where the model was exposed to a wider variety of test cases, it frequently failed. It sometimes repeated the input word (e.g., "bus" $\rightarrow$ "bus") or produced completely unrelated outputs (e.g., "box" $\rightarrow$ "bookcase"), suggesting limited generalization ability in more complex or less common scenarios.

Despite these limitations, two notable findings emerged. First, GPT-2 was able to infer the underlying task—pluralization—even when specific morphological rules (such as changing "-y" to "-ies") were not explicitly shown in the prompt. For example, the model successfully pluralized "city" as "cities" after being shown examples like `toy:toys` and `bus:buses`, indicating some capacity for rule abstraction from contextual patterns.

Second, we observed a counterintuitive trend: longer prompts often led to worse performance. In cases where several examples were included—such as `toy:toys bus:buses boy:boys baby:babies city:_`—the model incorrectly returned "city" instead of "cities." This suggests that increasing prompt length may introduce cognitive overload or dilute the signal necessary for successful ICL in smaller models like GPT-2.

5.2 Irregular Verb Conjugation

We extended our evaluation of ICL by GPT-2 to a different morphological domain—namely, verb inflections—focusing on irregular verb conjugations such as *present* → *past* → *past participle*. Prompts included sequences like `awake,awoke,awoken#begin,began,begun#bear,bore,` to assess whether the model could infer the third form based on patterns in the earlier pairs.

Unlike the partial success observed with regular noun pluralization, GPT-2 performed significantly worse in this domain. It almost never produced the correct third form of the verb, even when clear patterns were provided in the prompt. This poor performance highlights a limitation in GPT-2's ability to generalize more abstract or irregular morphological transformations through in-context examples.

5.3 Domain-Specific Pretraining

To explore whether targeted pretraining improves ICL, we trained a new model exclusively on singular-to-plural transformations, omitting verbs and other morphological forms entirely. This model was never exposed to irregular inflections during training.

Despite its limited exposure, the model exhibited modest generalization. For example, it correctly inferred `cleans` from `jump#jumps#jumped?clean#`, but performance degraded rapidly as input complexity increased. For longer prompts like `jump#jumps#jumped$clean#cleans#cleaned$wash#washes#`, the model produced malformed completions like `cles$ples$ples@`.

In simpler cases such as `play#plays#played`, the model occasionally generalized correctly. However, additional contextual hints often led to repetition or errors (e.g., `paintes` instead of `painted`), suggesting a superficial rule-based pattern matching rather than robust morphological reasoning.

5.4 Morphological Transformations

We trained our model exclusively on singular plurals and tested our model on other transformations including:

- **Regular**: base to past (e.g., walk → walked), third-person present (run → runs), plural possessive (dog's → dogs').
- **Challenging**: comparative to superlative (fast → fastest), base to progressive (run → running), derivational (confuse → confusing), and diminutive (dog → doggy).

While the model succeeded on common and regular inflections, it struggled with less productive or irregular derivations. These require more abstract morphological manipulation, which the model failed to capture.

5.5 Generalization to Held-Out Morphological Transformations

We trained our model withheld three transformations entirely during training to assess out-of-distribution generalization via ICL:

Base to Past Participle. The model occasionally produced regular participles (e.g., `cleaned`) from analogical prompts. Irregular forms (e.g., `gone`) led to noise or malformed tokens.

Simple Past to Past Perfect. Tasks involving auxiliary verbs (e.g., `had eaten`) produced only partial success, with malformed completions like `cleaneded@e` or syntactically incorrect phrases.

Affirmative to Negative. This was the most difficult transformation. Neither GPT-2 nor our model generated valid negatives (e.g., "did not eat"), instead producing repetition or nonsensical tokens like "i eat@@@@."

Discussion. The model generalized better for regular suffix-based tasks, but failed to induce abstract, irregular, or syntactically complex transformations. GPT-2, in contrast, defaulted to repetition

across all held-out tasks. These findings suggest that small transformer models can acquire inductive biases through targeted morphological pretraining, but their generalization is limited by both task complexity and model capacity.

6 Results

When compared directly to GPT-2 using morphological analogy-style prompts, our specialized model demonstrated a modest but measurable capacity for morphological generalization through in-context learning (ICL). For regular transformations such as Base to Past Participle, it occasionally produced correct forms like `cleaned`, while GPT-2 often repeated input tokens (e.g., `jump#jumps`) or generated unrelated fragments (e.g., `cleansed#cle`).

The contrast was especially pronounced in structurally complex tasks. For instance, in the Affirmative to Negative transformation, our model at least gestured toward syntactic structure (e.g., "will" or "i eat"), while GPT-2 failed to recognize the transformation pattern entirely. Though far from fully correct, these attempts indicate some emergent syntactic reasoning capabilities in our model.

These comparative observations suggest that pretraining explicitly on morphological transformations—even without exposure to the final target tasks—can improve in-context generalization relative to general next-token prediction. Although performance on irregular or syntactic transformations remained weak, our model consistently outperformed GPT-2 across held-out evaluations.

Consequently, while next-token prediction remains effective for general-purpose language modeling, it may not be optimal for learning structured morphological rules. Even small models trained on focused morphological transformations develop useful inductive biases. Scaling model size and expanding the training corpus could further amplify these early signs of generalization.

7 Limitations and Future Work

7.1 Limitations

Although our nano-sized transformer ($\approx$10M params) learns regular suffix-based rules, its training regime of 5k epochs on 18k LLM-generated analogies exposes only a sliver of English morphology and none of the syntactic cues needed for harder phenomena. As a result, accuracy collapses on irregular or structurally complex transformations (e.g., comparative $\rightarrow$ superlative, affirmative $\rightarrow$ negative) and even on regular tasks when prompt length grows. This produces repetitions or nonce strings like "foresawness". The study is further limited to English analogies, so cross-lingual generalization and open-text generation remain untested.

7.2 Future Work

Future work should therefore (i) scale model capacity and context window, (ii) replace or augment the synthetic dataset with attested, multi-lingual corpora that cover richer inflectional and derivational paradigms, and (iii) integrate morphological tags or lightweight syntactic signals during training; an approach our preliminary domain-specific pre-training results suggest could substantially improve robustness. Probing the resulting attention patterns against larger pretrained LLMs may also reveal transferable inductive biases that bridge the gap between artificial and human-like morphological reasoning.

References

[1] Jean Berko. The child's learning of English morphology. *Word*, 1958.

[2] S. Garg, Y. Wu, M. L. Littman, et al. What can transformers learn in-context? A case study of simple function classes. *arXiv preprint arXiv:2208.01066*, 2022.

[3] Eric Todd and Millicent L. Li and Arnab Sen Sharma and Aaron Mueller and Byron C. Wallace and David Bau English Morphological Dataset (2023). `https://huggingface.co/datasets`