ATNLP Assignment 3

Johan Palm
qkv218@alumni.ku.dk

1 Introduction

Humans possess the innate ability to combine known primitives into an infinite mixture of meanings, an ability known as compositional generalization. While AI has progressed significantly, systematic rule induction remains a challenge. Following the original evaluation of recursive neural network architectures on the **Simplified version of the CommAI Navigation (SCAN)** dataset (Lake and Baroni), we developed a Transformer encoder-decoder model with a custom tokenizer. We evaluate this reproduction across three benchmarks: **Percentage split**, **Length split**, and **Add primitive**. Furthermore, I investigate the potential of **Retrieval-Augmented Generation (RAG)** and **In-Context Learning (ICL)** using a pretrained T5 model to improve systematicity on the Percentage and Length splits.

1.1 SCAN Dataset

The SCAN benchmark evaluates whether an architecture can induce an underlying grammar rather than performing simple pattern recognition (Lake and Baroni). It is a sequence-to-sequence task mapping natural language commands comprising verbs (e.g., *run, jump*), modifiers (*twice, thrice*), and directions (*left, right*) into action sequences (e.g., `I_TURN_RIGHT I_JUMP`).

Percentage Split Using training subsets of 1, 2, 4, 8, 32, 64, and 100%, this split identifies the minimum data threshold required for a model to transition from memorizing commands to inducing general systematic rules.

Length Split This split evaluates a model's capacity for systematic productivity. The training set contains sequences of up to 22 actions, while the test set contains sequences between 24 and 48 actions. We conduct this experiment with and without an **oracle**. The oracle forces the generator to produce an output matching the exact length of the target bypassing the models normal termination logic.

Add Primitive Designed to test zero-shot generalization, this split provides a primitive verb (e.g., ”jump”) only in its simplest form during training. At test time, the model must compose this primitive with known modifiers (e.g., ”jump around right”). This determines if the model has learned the abstract functions of modifiers independently of specific verbs.

1.2 Experiment specifics

RAG was introduced as a mean to bridge models parametric knowledge with non-parametric knowledge (Lewis et al.). In the original article the non-parametric knowledge is a dense vector index of Wikipedia. This can be done by making an embedding of chunks of text and testing similarities with the input, thus retrieving the most similar text compare to the input. RAG consists of two components; the retriever that returns the top K results given the input, and the generator that takes the retrieved results combined with the input to generate the output.

The embeddings are made by the `all-MiniLM-L6-v2` model that is a fine-tuned MiniLM (Wang et al.) with 22m parameters included in sentence-transformers for Hugginface. It is tuned on multiple datasets with 1B sentence pairs toward the sentence embedding task. It searches through the embeddings with Facebook AI Similarity Search (FAISS) (Johnson et al.) that enables efficient nearest-neighbor retrieval by organizing high-dimensional vectors that can then be similarity searched with flat L2 norm.

In-context learning (ICL) is a paradigm in which a large language model learns to perform a new task by observing a few examples provided within its input prompt, without undergoing any updates to its internal weights (Dong et al.).

The Text-to-Text Transfer Transformer (T5) is a unified framework that reformulates all Natural Language Processing (NLP) tasks into a consistent text-to-text format (Raffel et al.). By treating every NLP problem as a sequence generation task, T5 enables the application of the same model architecture, loss function, and hyperparameters across diverse datasets.

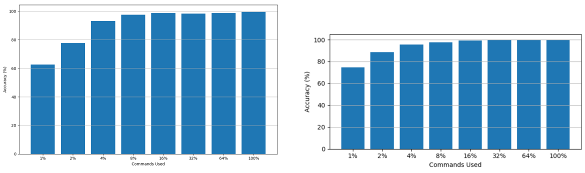
T5 utilizes a standard encoder-decoder Transformer structure. It is trained on the ”Colossal Clean Crawled Corpus” (C4) and is made up of 364,613,570 examples. T5 showed improvements over the previous best models when first introduced. Further, it has several sizes, including a ”small” with 60M parameters and is available through Huggingface as open weights.

2 Reproduction

As a group of four members we implemented the transformer block from the ground up using the assigned skeleton code. There after we conducted the three experiments included below:

2.1 Results

Percentage Split Experiment As illustrated in Figure 1, our implementation exhibits a notable sample-efficiency gap at the lowest data splits. For the 1% and 2% splits, the reproduced results trail the expected baseline by approximately 12%. However, this performance gap fades at the 4% mark and beyond, where the model achieves near-perfect token and sequence-level fidelity.

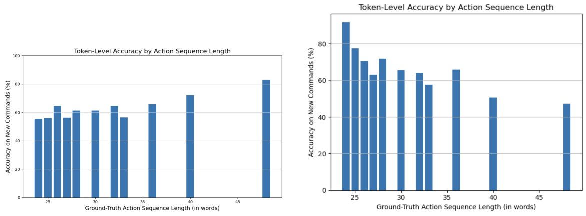


(a) Reproduced

(b) Expected

Figure 1: Comparison of sequence accuracy Percentage split with expected and reproduced results.

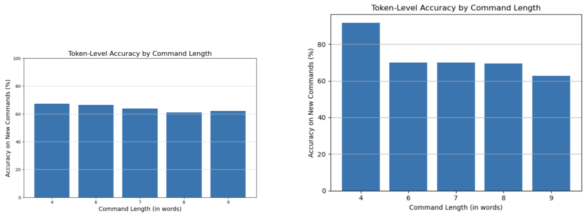
Length split: Without Oracle Without an oracle, sequence-level accuracy for both reproduction and expected results is 0.00 thus only token level accuracy is shown. For action length (Figure 2a), the reproduced results show a mirroring effect to the expected results (Figure 2b). While our reproduction demonstrates lower initial accuracy at length 24, it exhibits a higher token-level extrapolation at extreme lengths (Length 48). This trend is corroborated by Figure 3, where the reproduction holds a flattened performance plateau compared to the baseline's steep decay.



(a) Reproduced

(b) Expected

Figure 2: Comparison of Token-level accuracy by action length without oracle.

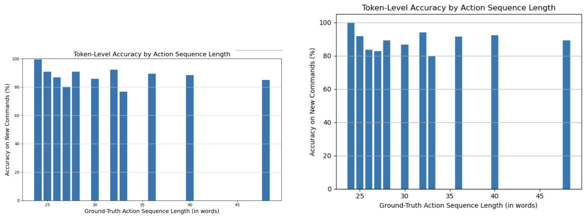


(a) Reproduced

(b) Expected

Figure 3: Comparison of Token-level accuracy by command length without oracle

Length split: With Oracle When the sequence length is provided via the oracle, the model maintains high token-level fidelity across all lengths (Figures 4a and 5a). Despite this, Figure 6 shows that reproduced sequence-level accuracy values are generally lower compared to the expected baseline. In Figure 7, the reproduced model's sequence accuracy is lower for 4-command inputs by approximately 10% but higher for 6 and 7 commands by roughly 10%.

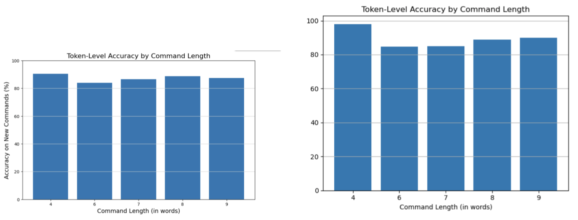


(a) Reproduced

(b) Expected

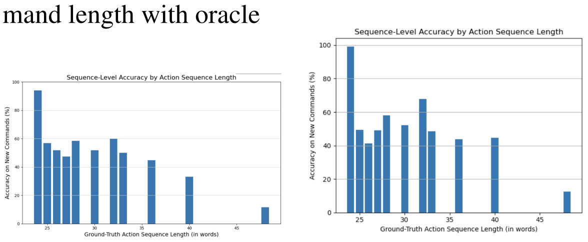
Figure 4: Comparison of Token-level accuracy by action length with oracle.

Add Primitive Experiment As shown in Figure 8, token-level accuracy remains similar between the reproduction and expected baseline. However, for sequence-level accuracy (Figure 9), they differ largely. At the 32-command mark, the expected



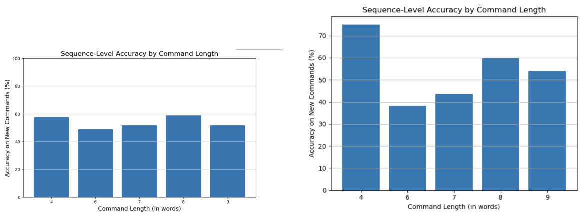
(a) Reproduced (b) Expected

Figure 5: Comparison of Token-level accuracy by command length with oracle



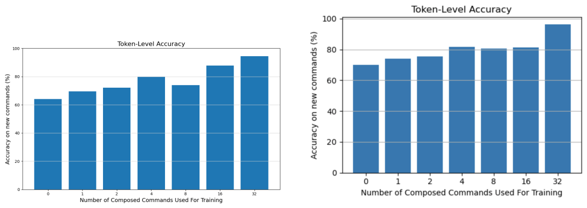
(a) Reproduced (b) Expected

Figure 6: Comparison of sequence-level accuracy by action length with oracle.



(a) Reproduced (b) Expected

Figure 7: Comparison of sequence-level accuracy by command length with oracle



(a) Reproduced (b) Expected

Figure 8: Comparison of token-level accuracy on the add primitive split

model achieves approximately 55% sequence accuracy, whereas our reproduction plateaus at roughly 36%.

2.2 Discussion

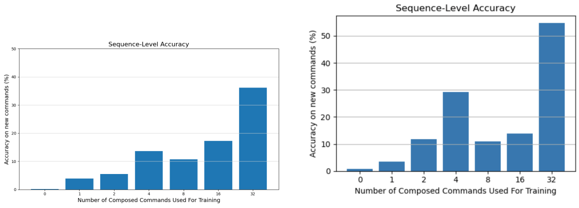
Positional Encodings and Sample Efficiency The 12% gap at the 1% and 2% splits in the percentage experiment suggests that our possible mistaken implementation of absolute positional embeddings requires more observations to optimize learned vectors compared to relative embeddings. The 'cold-start' phenomenon suggests that once the attention mechanism understands the modifiers, the specific position of the data becomes less important to the model's overall performance. However, the model's reliance on seeing nearly all combinations to achieve high accuracy differs from human zero-shot understanding. The success is from statistical interpolation across a known coordinate space rather than length invariant reasoning.

Termination Logic and the Positional Horizon The 0.00 sequence-level accuracy observed without the oracle highlights a fundamental discrepancy between human compositional reasoning and Transformer-based models. While the high token-level accuracy at extreme lengths (e.g., length 48) suggests the attention mechanism successfully induced abstract grammatical rules for modifiers like `around` and `thrice`, the failure to precisely emit the `<eos>` token at unlearned positional indices confirms a localized failure in termination logic. The machine's understanding appears bounded by a ”positional horizon”; once the sequence length exceeds training limits, the model struggles to maintain global structural integrity regardless of its ability to predict individual action tokens correctly.

As evidenced by the results with the oracle (Figures 4 through 7), bypasses this horizon by providing the target length as an external structural constraint. This allows the model to recover significant performance, indicating that the underlying attention mechanism relies more on learned weights than on fixed positions to produce the correct actions once the termination logic is given.

However, even with the oracle, sequence-level accuracy remains below perfect fidelity and fluctuates compared to the baseline. This reveals that the model's bottleneck that it still suffers from categorical errors in action selection. The oracle effectively masks the termination failure, but the remaining gap in sequence accuracy confirms that the model still struggles to synthesize recursive commands into long-form action chains.

Structural Bias and Contextual Systematicity The results from the Add Primitive experiment demonstrate that while implementation differences did not significantly impact token-level mapping, they created a critical bottleneck for sequence-level accuracy. It is clear that the standard Transformer architecture lacks the intrinsic capability to see an action and place it into context as humans do. Because the core attention mechanism is permutation-invariant (Dufter et al.), the model's understanding of context is strictly bound to the specific sequence



(a) Reproduced

(b) Expected

Figure 9: Comparison of sequence-level accuracy on the add primitive split

configurations encountered during training. Without a structural bias, the Transformer struggles to maintain global structural integrity for new primitives, even within known contexts.

2.3 Conclusion

The experiments on the SCAN benchmark reveals a significant gap between the Transformer's ability to map individual tokens and its capacity for global structural integrity. The results demonstrate that while the attention mechanism can successfully induce abstract rules for modifiers, as evidenced by high token fidelity under oracle conditions, the model remains constrained by a "positional horizon" that prevents autonomous sequence termination on out-of-distribution lengths.

3 Experiment on pre-trained network

The **Percentage split** and **Length split** experiments will be performed using the `T5-small` model using RAG to perform ICL. RAG is performed by `all-MiniLM-L6-v2` for embeddings and Facebook AI Similarity Search (FAISS) for the datastore with flat L2 similarity search.

3.1 Motivation and Research Objectives

The primary motivation for implementing a RAG framework that combines a `FAISS` vector index, the `all-MiniLM-L6-v2` encoder, and a `T5-Small` generator is to address the inherent systematicity gaps in standard Transformer architectures. By separating the storage of compositional logic from the generative process, I can evaluate whether ICL and external knowledge retrieval can overcome the length and data-availability "barrier" without the need for expensive retraining or fine-tuning.

Furthermore, this architecture uses the pretraining of the `T5` model on English corpora to determine if linguistic knowledge can be transferred to the recursive grammar of the SCAN dataset, especially in the percentage split. A multilingual model could bring advantages, however, it is not used due to the commands and actions being in English. Additionally, the RAG pipeline provides a "glass-box" diagnostic environment. This setup allows me to isolate failures: I can specifically determine if an incorrect output originates from the retriever's inability to find relevant analogs or the generator's failure to synthesize those parts into coherent actions.

Research Objectives My experiment seeks to address the following core objectives:

1. **Quantification of the Retrieval Penalty:** To analyze the impact of retrieval noise on systematicity, specifically examining how the `T5` attention mechanism handles the retrieval pool increasing in size.

2. **Evaluation of Zero-Shot Structural Synthesis:** To determine if a pretrained generator can synthesize action sequences for length split commands by using shorter retrieved training commands.

3. **Oracle-Bound Benchmarking:** To establish a performance upper bound by utilizing a length constraint Oracle. Testing the recursive ability of the model without the possible noise of termination logic.

System architecture `FAISS` was chosen for its high-speed, high-dimensional similarity searches that bridge the gap between a raw dataset and actionable context (Johnson et al.). L2 distance calculation, providing the the model with the most similar training examples available. K (=5) examples are retrieved and added to the prompt.

`all-MiniLM-L6-v2` is chosen as the embedding as it provides high-fidelity embeddings without the computational overhead of larger encoders and is available through the Huggingface interface where it is the most downloaded sentence-embedder. The model is specifically tuned on 1B sentence pairs which might positively impact the command to action sequence. The training set is embedded in the datastore.

The `T5-Small` model is motivated by its "text-to-text" paradigm, where it is trained for different downstream tasks. It's small 60M parameters keeps the computational cost down for the experiment. A purely English trained model is used. It's availability as a open weight with Huggingface is also an advantage. The prompt for ICL was designed as following: ”Translate SCAN to actions” followed with a new line and ”command $->$ action” times K and the test command is added as: ”command $->$”.

3.2 Results

Percentage split For all percentage splits, the sequence accuracy is 0.00. As can be seen in Figure 10 the token level accuracy starts at 14.4% at 1% and slowly degrades hitting its lowest at 10.9% for 100%.

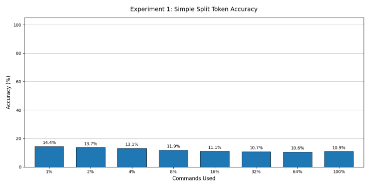
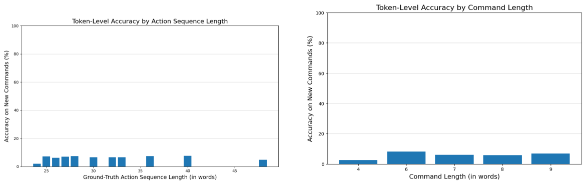


Figure 10: Token-level accuracy for percentage split with RAG solution

Length split Without oracle, the sequence accuracy is 0.00. As can be seen in Figure 11a the token level accuracies are around $5-10\%$ for the different action lengths. The same can be seen for Figure 11b for command length.



(a) Action length (b) Command length

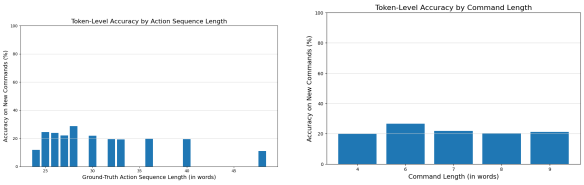
Figure 11: Token-level accuracy for command and sequence length on the length split without oracle

The oracle was implemented per batch by generating until the longest target length and then truncating to the individual target length if needed.

As can be seen in Figure 12b the oracle vastly improved the performance where the command length for all lengths is over 20% and for length 6 closer to 30%. For action sequence lengths, as seen in Figure 12a, only for length 24 and 48 are far below 20% while the rest are around 20% with some going closer to 30%. Sequence level accuracy is however still 0 even with the oracle.

3.3 Discussion

Analysis of output sequences across experiments reveals a significant retrieval penalty caused by demonstration over-reliance. As seen in the examples, frequent substitution errors (e.g., I_TURN_RIGHT for I_TURN_LEFT) often correlate with the most common tokens in the retrieved ICL examples. This suggests the attention mechanism prioritizes the few-shot context over the input command. Future experiment would be to test with different number of retrieved sentences to see if the problem persists. Furthermore, ”task-specification leakage” where the model repeats prompt metadata like *”SCAN to actions:”* highlights a failure to separate instructions from logical synthesis.



(a) Action length (b) Command length

Figure 12: Token-level accuracy for command and sequence length on the length split with oracle

As the FAISS index grows, the vector space becomes increasingly dense, causing the retriever to struggle with ”semantic crowding.” At 100% capacity, it frequently fetches logically incorrect distracting examples that share high semantic similarity with the query, interfering with the generator’s mapping. In the length split, the model consistently under-generates sequence lengths, failing to use parametric knowledge to extrapolate. While the oracle recovers performance by bypassing this termination bottleneck, the persistent categorical errors and prompt hallucinations confirm that the ICL method is still limited by contextual interference.

3.4 Conclusion

My evaluation confirms that while the RAG-ICL pipeline identifies relevant primitives, it fails to achieve full systematicity due to contextual noise and structural limitations. The model’s tendency to mirror retrieved demonstrations rather than the input command prevents successful zero-shot structural synthesis.

Furthermore, the Oracle-bound benchmarking proves that while the generator possesses the underlying mapping rules, it lacks the autonomous logic to manage sequence termination or filter distractors. Ultimately, unlike human recursive logic, which independently decouples an action’s meaning from its repetition, the standard Transformer architecture requires stronger structural biases to overcome the systematicity gaps present in the recursive SCAN grammar.

References

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128. Association for Computational Linguistics.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. 48(3):733–763.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

A Appendix: Experiment data

Given in the file structure that is handed in is also a folder containing the outputs from the different experiments, each labeled with the experiment number.