Tokenization Falling Short: On Subword Robustness in Large Language Models

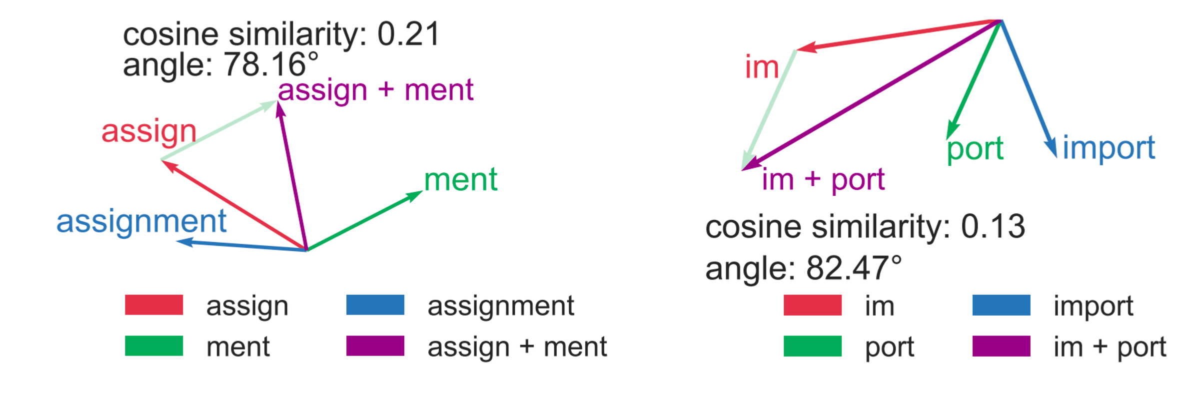
Yekun Chai[1], Yewei Fang[2], Qiwei Peng[3], Xuhong Li[1]

[1]Baidu Inc., [2]ModelBest, [3]University of Copenhagen

EMNLP 2024

Motivation

* Tokenization is a fundamental step in the preprocessing pipeline of LLMs;
* Challenges, such as **typographical errors**, **length variations**, **awareness of internal structure**, are observed to hinder the performance and robustness of LLMs.

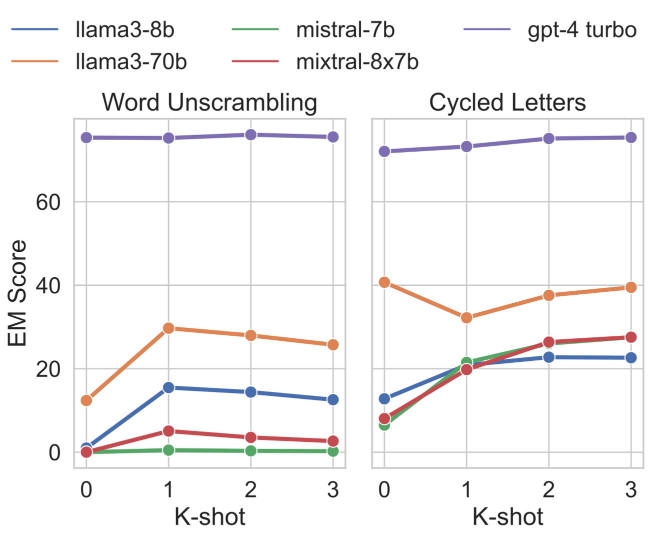


(a) cosine (“assignment”, “assign” + “ment”). (b) cosine(“import”, “im” + “port”).

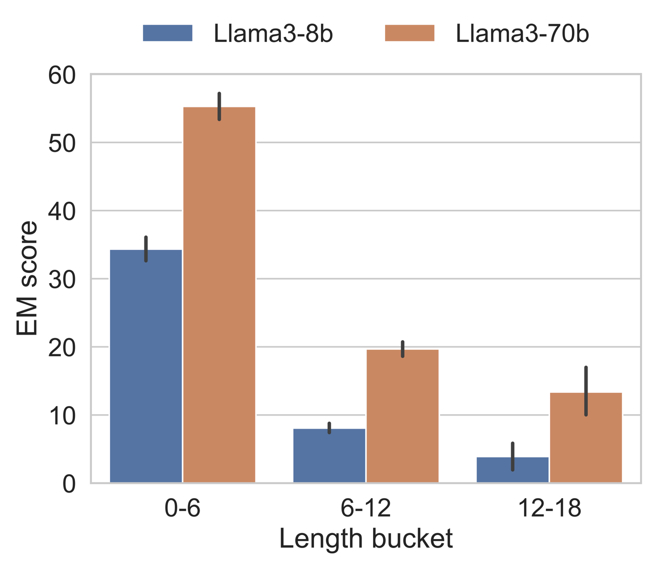
Research Question 1. Complex Problem Solving

- Task:
  - Anagram Task
    - Cycled Letters in Word (CL) (e.g., "Please unscramble the letters into a word Q: uald A: " -> "dual")
    - Word Unscrambling (WU) (e.g., "The word hte is a scrambled version of the English word " -> “the”)
  - Mathematical Language (LaTeX) Comprehension
    - Identify Math Theorems (IMT)

- Result



- K-shot performance on **WU and CL anagram tasks**:
  - Increasing k number does not consistently enhance the performance
  - Models with larger parameter sizes generally perform better



- Larger models tend to have better performance on anagram tasks
- Models tend to correctly reorder anagrams of shorter lengths, while struggling with longer ones

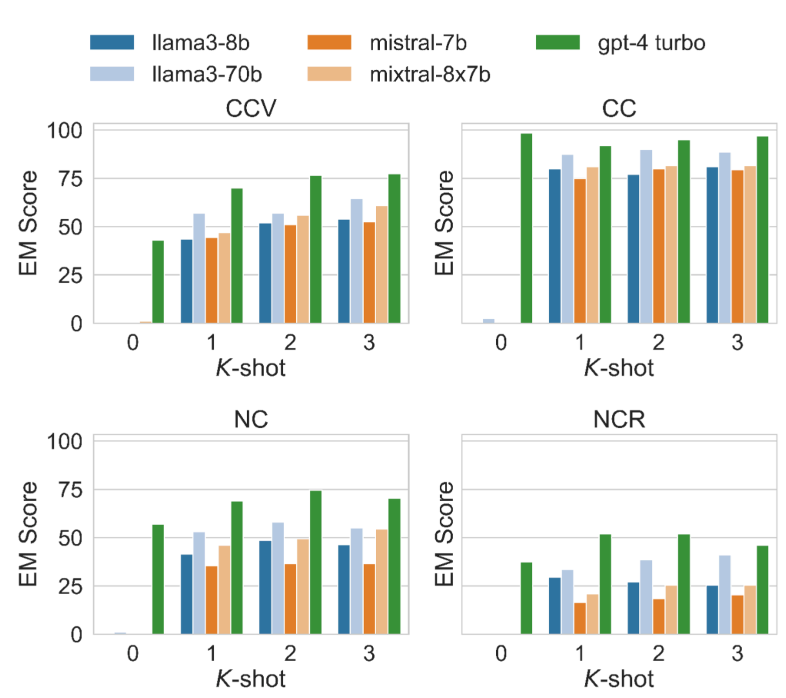
Setting	0-Shot	1-Shot	2-Shot	3-Shot
GPT-3 (6B)[a]	33.96	28.30	33.96	28.30
GPT-3 (200B)[a]	32.08	30.19	33.96	30.19
Llama2-7b	37.70	34.00	35.80	37.70
Llama3-8b	41.51	45.28	45.28	35.85
Llama3-70b	**62.26**	**79.25**	**69.81**	**71.70**
Mistral-7b	47.20	43.40	37.70	37.70
Mixtral-8x7b	49.10	56.60	64.20	62.30

- On **IMT tasks**:
  - Larger models generally perform better, while the relation between K-shot number and performance is not linear
  - Simply increasing model size does not guarantee better performance on IMT

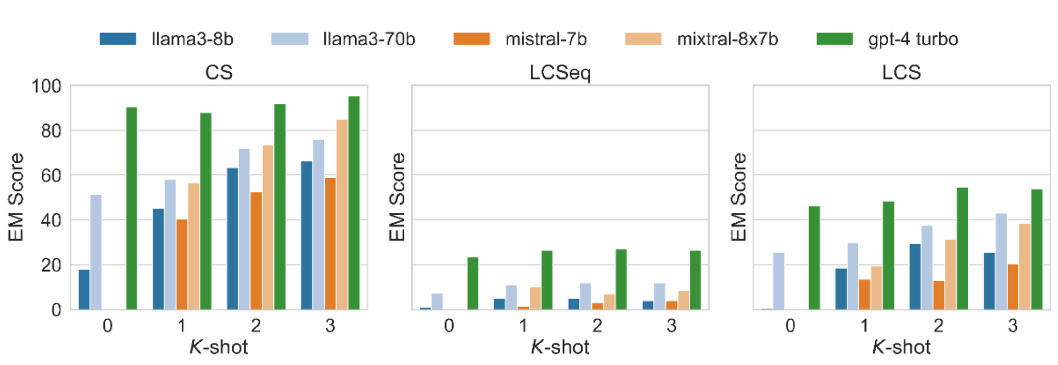
Research Question 2. Token Structure Probing

- Task:
  - Intra-Token Probing
    - Character Count (CC) (e.g., "Which character appears 2 times in the word 'fleet'?" -> "e")
    - N-th Character (NC) (e.g., "What is the 4th character of the word 'fleet'?" -> "e")
    - N-th Character Reverse (NCR) (e.g., "What is the 1st character from the end of the word 'fleet'?" -> "t")
    - Case Conversion (CCV) (e.g., "Convert the 4th character of the word 'correlate' to uppercase:" -> "R")
  - Inter-Token Probing
    - Common Substrings (CS) (e.g., "What are the common substrings of 'cover' and 'correlate'?" -> ["c", "o", "e", "co", "r"])
    - Longest Common Substrings (LCS) (e.g., "What are the longest common substrings of 'control' and 'count'?" -> ["co", "nt"])
    - Longest Common Subsequences (LCSeq) (e.g., "What are the longest common subsequences of 'control' and 'count'?" -> ["cont"])

- Result



- K-shot performance on **intra-token probing tasks**:
  - Increasing k number from zero-shot to one-shot results in large improvements, with performance stabilizing thereafter
  - Models with larger parameter sizes generally perform better
  - GPT-4 turbo achieves decent and the best performance among all tested models

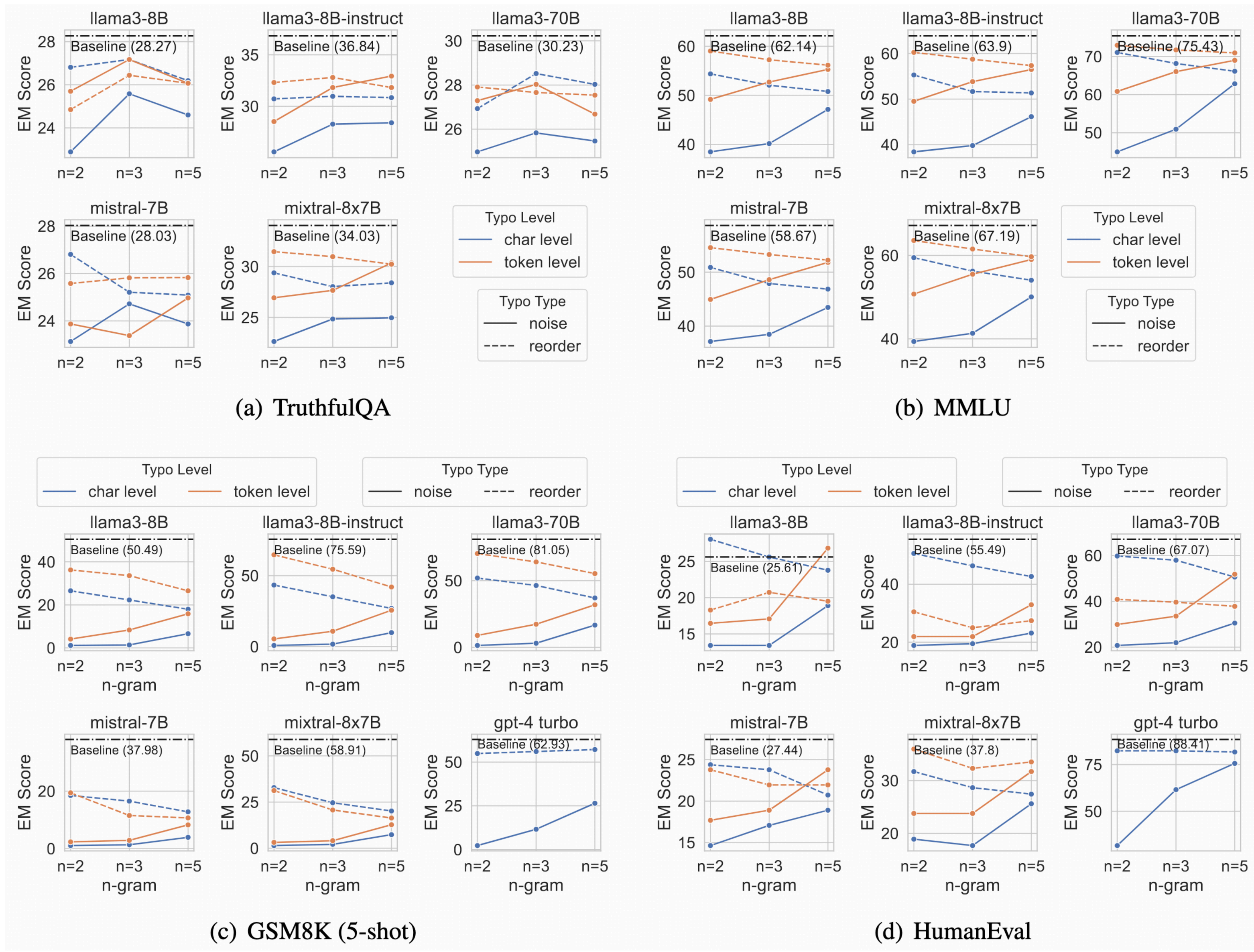


- On **inter-token probing tasks**:
  - Models with larger parameter sizes generally perform better
  - Increasing K number is effective
  - The task of LCSeq is extremely challenging

Research Question 3. Typographical Variation

- Dataset: MMLU, TruthfulQA, GSM8K, HumanEval
- Typographical variation
  - Character-Level Permutation (e.g., "find the largest number" -> "fdin teh raglets number")
  - Character-Level Noise (adding, deleting, replacing with p) (e.g., "find the largest number" -> "faind the lwrgest number")
  - Token-Level Permutation (e.g., [369, 17954, 1475, 6693, 323, 293] -> [369, 17954, 1475, 6693, 293, 323])
  - Token-Level Noise (adding, deleting, replacing with p) (e.g., [369, 17954, 1475, 6693, 323, 293] -> [369, ~~17954~~, 1475, 6693, 323, 4124, 293])

- Result

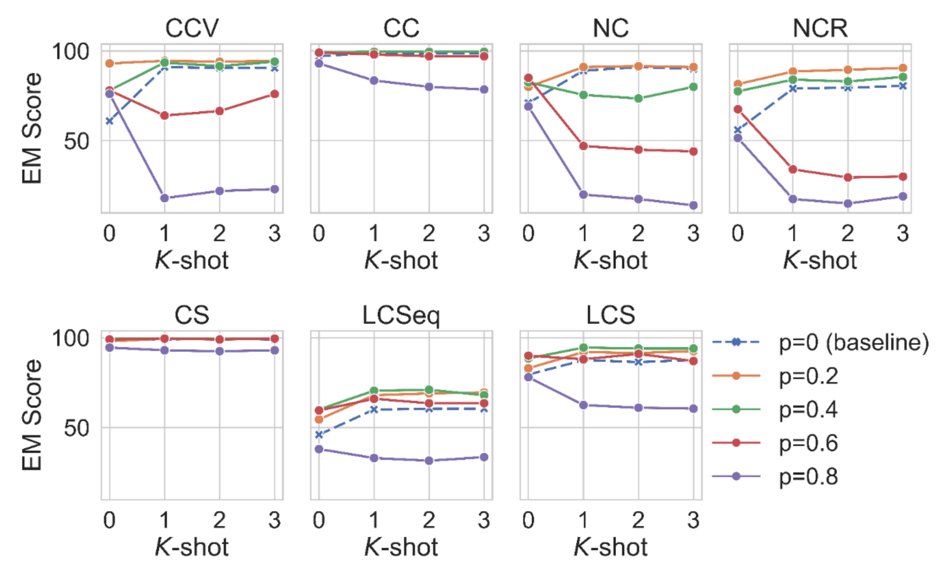


(a) TruthfulQA (b) MMLU (c) GSM8K (5-shot) (d) HumanEval

- Models with larger parameter sizes generally perform better.
- LLMs are much more sensitive to noise (solid lines) than to reordering (dashed lines).
- Degradation is observed on all models regardless of the parameter size and types, highlighting their sensitivity to typographical noises.
- Models generally perform better with token-level noise compared to character-level noises, suggesting token-level errors may be less disruptive to overall.semantics of the input.

Is BPE-dropout helpful?

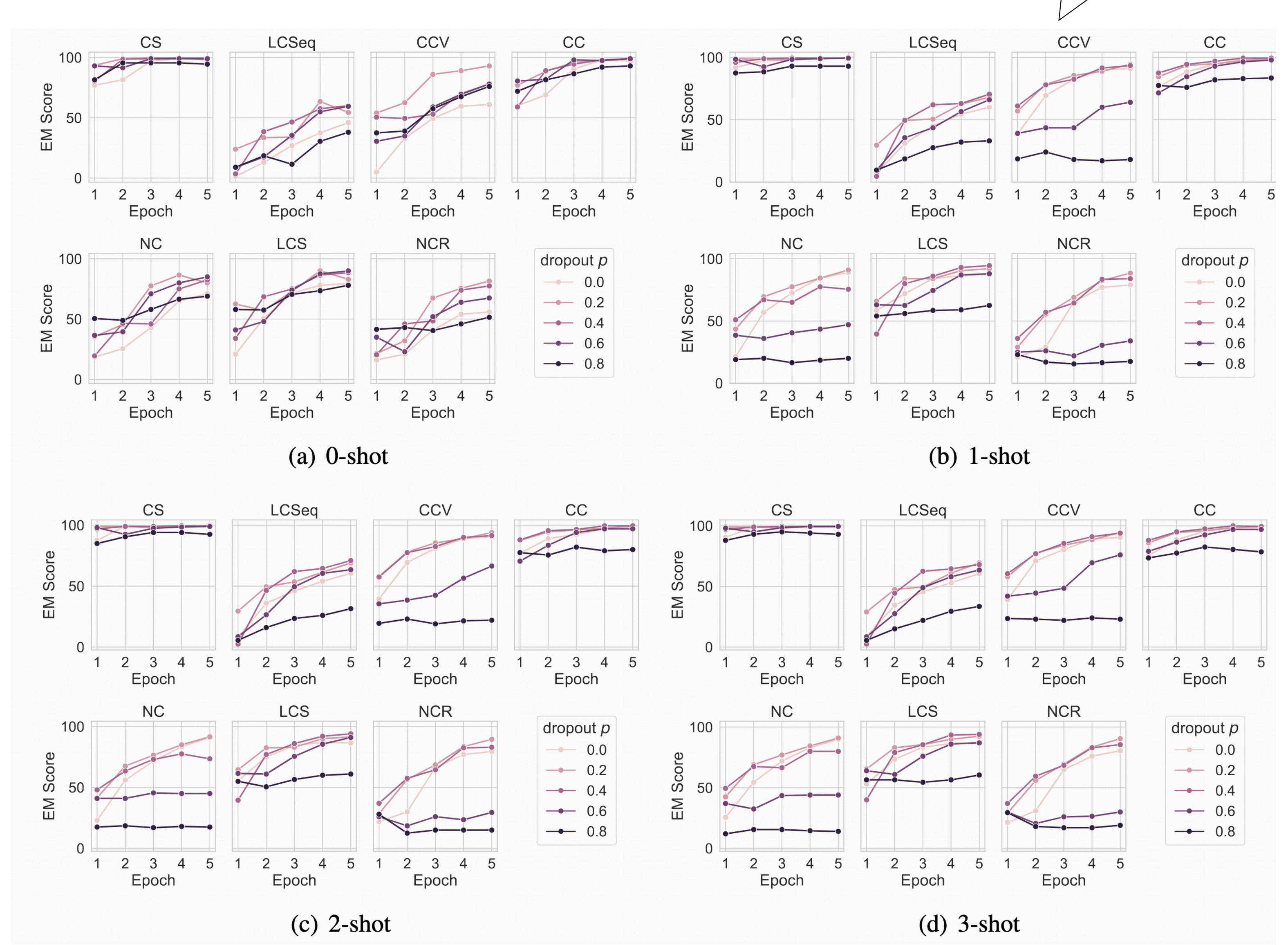
We post-train the Mistral-7B model with BPE-dropout for 5 epochs, with different rate of p value and experiment with token structure probe tasks.



Key findings:

- Introducing **a moderate (e.g., p=0.2) amount of variability** during tokenization improves the model’s understanding to token structures.

The test-set performance across seven tasks over the course of BPE-dropout fine-tuning



(a) 0-shot (b) 1-shot (c) 2-shot (d) 3-shot

Conclusion

- We comprehensively evaluate mainstream LLMs across 13 tasks that are sensitive to subwod tokenization
- Our findings reveal that while larger models and increased k-shot can partially mitigate these issues, LLMs still struggle with understanding internal structures of tokens
- We further demonstrate that moderate BPE-dropout can alleviate such issues and increase robustness