Tokenization Falling Short: On Subword Robustness in Large Language Models

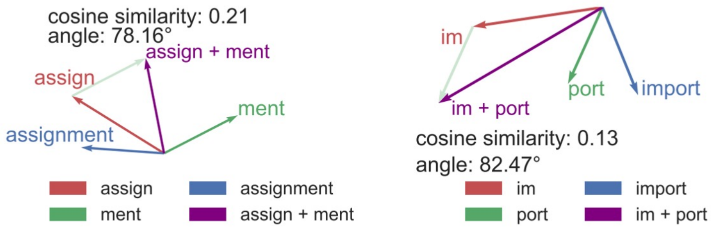
Yekun Chai*, Yewei Fang*, Qiwei Peng, Xuhong Li

Motivation

- Tokenization is a fundamental step in the preprocessing pipeline of LLMs

"lesson"	*lesson*
"racket"	*rack, ##et*
"vanquish"	*van, ##qui, ##sh*

- 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 Questions

To address these challenges, we conduct comprehensive study examining the limitations of current tokenization methods and their impact on LLM performance guided by three research questions:

1. **Complex Problem Solving**: Are LLMs capable of handling complex problems that are sensitive to tokenization?

2. **Token Structure Probing**: Do LLMs actually understand token structures, including intra-token and inter-token structures?

3. **Typographical Variation**: Are LLMs robust enough to typographical variations?

Contributions

1. We provide **a comprehensive analysis of the problem known as the curse of tokenization**, detailing its impact on large language model (Llama3, Mistral, and GPT-4) performance and introducing systematic evaluation benchmarks to assess these issues

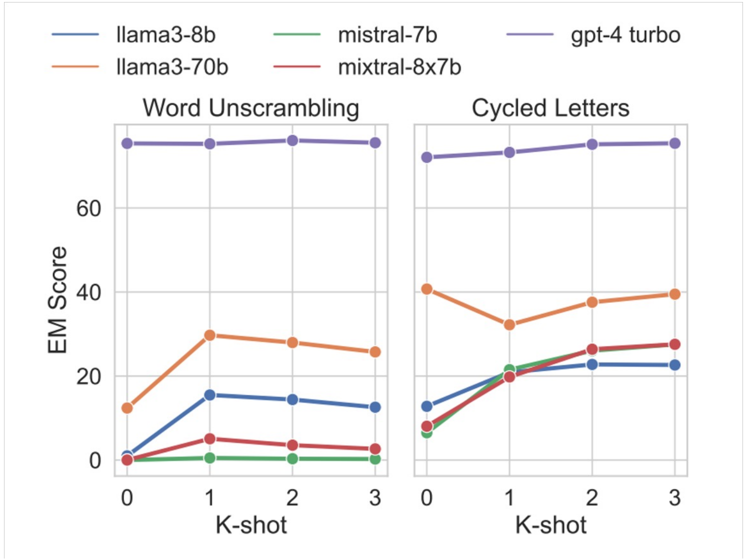
2. We demonstrate that **regularized tokenization approaches**, such as BPE-dropout with moderate dropout rates, can enhance the model's resilience to the discussed issues

Complex Problem Solving

- Anagram Task
  - Cycled Letters in Word (CL) (e.g., “remo” → “more”)
  - Word Unscrambling (WU) (e.g., “nad” → “and”)

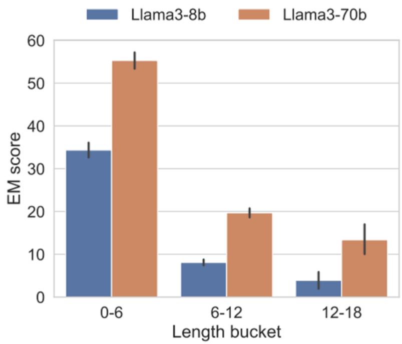
- Mathematical Language (LaTeX) Comprehension
  - Identify Math Theorems (IMT)

Results



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**:

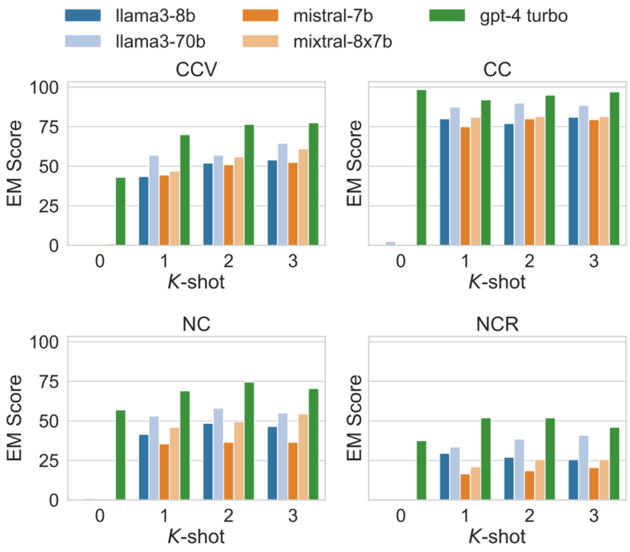
- Larger models generally perform better, while the relation between K-shot number and performance is not linear
- Simply increasing model size does not guarantee better performance on IMT

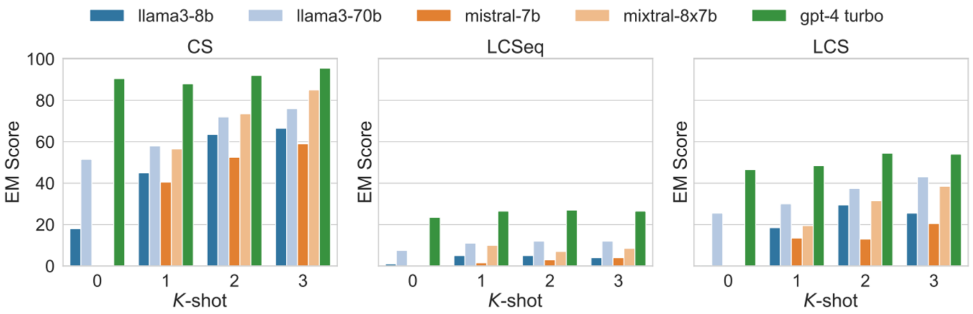
Token Structure Probe

- Intra-Token Probing
  - Character Count (CC)
  - N-th Character (NC)
  - N-th Character Reverse (NCR)
  - Case Conversion (CCV)

- Inter-Token Probing
  - Common Substrings (CS)
  - Longest Common Substrings (LCS)
  - Longest Common Subsequences (LCSeq)

Results





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

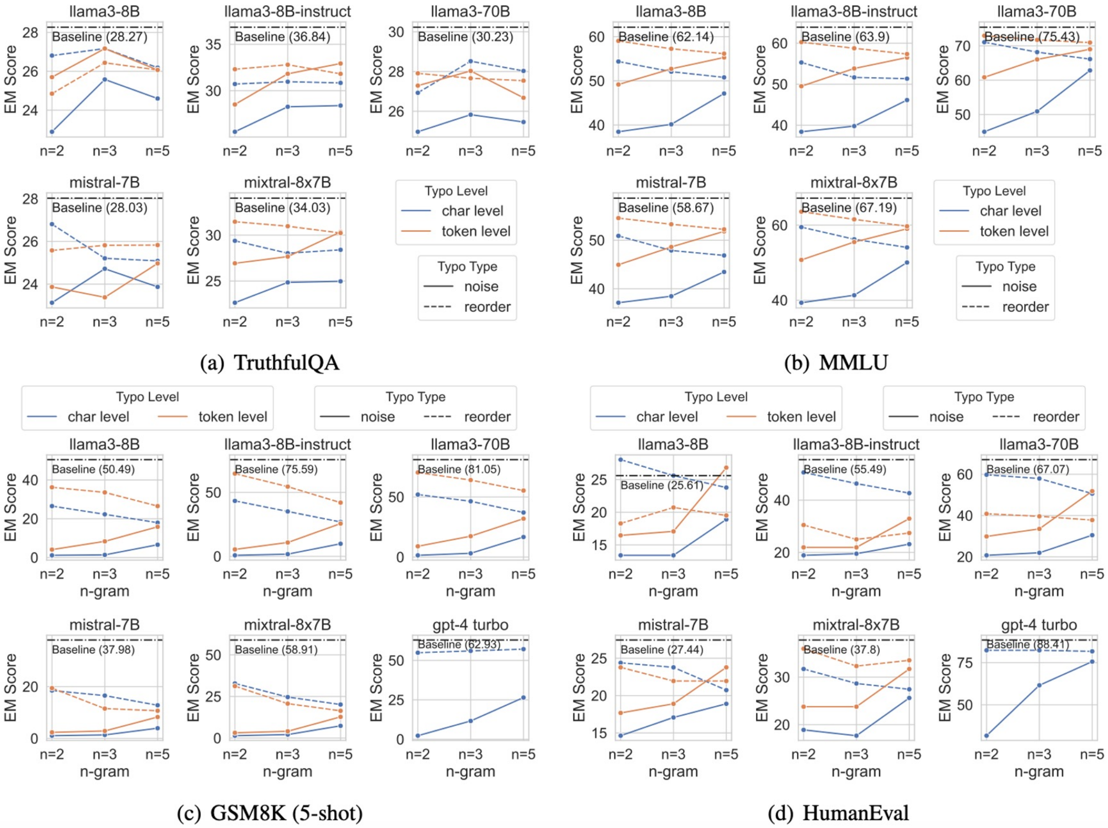
Performance on Tasks When Typographical Variations Introduced

- MMLU
- TruthfulQA
- GSM8K
- HumanEval

Typographical Variation:

- Character-Level Permutation
- Character-Level Noise (adding, deleting, replacing with p)
- Token-Level Permutation
- Token-Level Noise (adding, deleting, replacing with p)

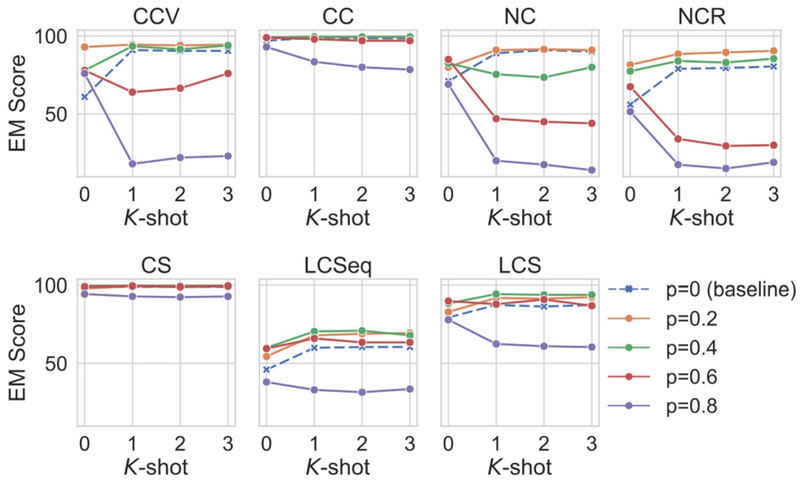
Results



- 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.



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

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