Improving Number Tokenization in BPE by Numerically-Based Tokens

Student: Alireza Mohammadnezhad
Instructor: Prof. Assadollah Shahbahrami
Multimedia at Guilan University

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across various natural language processing tasks. However, their performance on numerical reasoning and arithmetic operations often lags behind their linguistic proficiency. This paper introduces a novel algorithm for parsing numbers within the Byte Pair Encoding (BPE) tokenization framework, designed to enhance numerical understanding in LLMs. Our method employs a hybrid approach, combining right-to-left parsing for integers, left-to-right parsing for fractions, and a maximum grouping of three digits. This technique aims to better capture the place value system and improve the model's ability to process and manipulate numerical data.

Introduction

Tokenization plays a crucial role in natural language processing (NLP) and, by extension, in the broader field of artificial intelligence (AI). In the contemporary AI landscape, efficient and effective text representation is paramount for various language tasks. Byte Pair Encoding (BPE), a data compression technique repurposed for NLP, has emerged as a particularly significant tokenization method. BPE's importance lies in its ability to balance the trade-off between vocabulary size and token length, effectively handling rare words and subword units. This approach allows for more nuanced representations of text, capturing morphological information and improving performance on tasks ranging from machine translation to text generation. The adaptability of BPE to different languages and domains has contributed to its widespread adoption in state-of-the-art language models, underscoring its pivotal role in advancing natural language understanding and generation capabilities in the age of AI.

Despite its widespread adoption, Byte Pair Encoding is not without limitations. One notable shortcoming is its inability to effectively handle out-of-vocabulary words, particularly in morphologically rich languages. Additionally, BPE's greedy nature can sometimes lead to suboptimal segmentations. To address these issues, researchers have proposed several enhancements. Notably, SentencePiece introduced a language-agnostic tokenization method that treats the input as a sequence of Unicode characters, eliminating the need for pre-tokenization. Another significant improvement is Scaffold BPE, which incorporates linguistic knowledge to guide the merging process, resulting in more semantically meaningful subword units. WordPiece, a variant used in BERT, modifies the BPE algorithm to choose the most probable subword unit instead of the most frequent pair. Unigram language model tokenization, employed in XLNet, offers a probabilistic approach to subword

segmentation. These advancements, while conceptually simple, have collectively enhanced the robustness and efficiency of tokenization in modern NLP systems, demonstrating the ongoing evolution of this fundamental preprocessing step in the AI pipeline.

Another problem of BPE and tokenization in general, is the exhibition of significant limitations when handling numerical data in Large Language Models (LLMs), impacting their performance in numerical operations. These shortcomings stem from BPE's tokenization approach, which often fails to preserve the semantic structure of numbers. Key issues include:

1. Loss of place value information
2. Inconsistent representation of similar numbers
3. Challenges with fractional numbers and decimal points
4. Issues in handling scientific notation
5. Difficulties in aligning numbers for arithmetic operations
6. Problematic representation of very large numbers
7. Struggles with cultural variations in number formatting

These limitations lead to reduced accuracy in calculations, misinterpretation of numerical scale, and difficulties in learning stable patterns for numerical reasoning. The impact extends to cross-lingual applications and fields requiring precise numerical manipulation, such as finance and scientific computing. The core issue we focus on here is issue number 5.

Traditionally, addressing these issues has involved two main steps: (1) using special tokens to invoke external tools like calculators, and (2) fine-tuning models to use such tools effectively. While these methods can improve numerical accuracy, they rely on external assistance rather than enhancing the model's inherent capabilities. “tokenomically” and computationally, this approach makes perfect sense, but model computational understanding and generality is in jeopardy in this case.

From a scientific perspective, it's crucial to explore ways to improve LLMs' numerical performance without relying on external tools. This approach aims to enhance the models' fundamental understanding and processing of numerical concepts. Potential strategies could include:

1. Developing specialized number-aware tokenization schemes that preserve numerical structure and place value.

2. Incorporating numerical reasoning tasks in pre-training objectives to encourage better number representation learning.
3. Designing architecture modifications that handle numerical data differently from textual data within the model.
4. Implementing number-specific attention mechanisms to better capture relationships between digits in different positions.
5. Exploring hybrid approaches that combine symbolic and neural methods for number processing within the model architecture.

By focusing on these intrinsic improvements, we can push the boundaries of LLMs' numerical capabilities, potentially leading to more robust and generalizable numerical reasoning abilities without external dependencies. This approach not only enhances model performance but also deepens our understanding of how neural networks can learn to process and manipulate numerical information.

Proposed Solution

Our proposed tokenization method focuses on grouping digits in sets of two or three, parsing integers from right to left (RTL). Grouping numbers is not a new technique, all the leading LLMs use tokenizers which, de facto, use this approach. What differs my approach is the direction of parsing tokenizing numbers. This approach effectively creates a base-100 or base-1000 representation of numbers, aligning with human intuition for place value and magnitude. For fractional parts, we maintain a left-to-right (LTR) parsing to preserve the conventional decimal representation, which better matches the real magnitude of the number as well.

Key Features and Benefits

1. **Right-to-Left Integer Parsing**: By tokenizing integers RTL in groups of two or three digits, we preserve the significance of place value. For instance, 12345 would be tokenized as 12345 in base-1000, or 12345 in base-100. This representation allows the model to quickly assess the magnitude of a number by focusing on the leftmost (most significant) token. Meanwhile, most of the current tokenizers would tokenize 12345 as 12345, which ignores the exponential curve of digit growth, and make it harder for models to estimate the value they should return.
2. **Approximation and Scale Recognition**: This method facilitates easier approximation of multiplications. The model can possibly estimate the result's order of magnitude by primarily considering the highest value tokens. For

example, in multiplying 12300 by 45600, the model can quickly recognize that the result will be in the order of 10^11 by focusing on the [12] and [45] tokens.

3. **Consistent Fractional Representation**: For numbers with decimal points, the integer part is parsed RTL, while the fractional part is parsed LTR. This maintains consistency with standard decimal notation while benefiting from the RTL grouping for the integer portion.
4. **Improved Numerical Reasoning**: By providing a more structured representation of numbers, this method can probably, enhance the model's ability to perform arithmetic operations and comparisons. The consistent grouping allows for better alignment in addition and subtraction, while the base-100 or base-1000 representation simplifies certain multiplication and division operations.

Potential Challenges and Solutions

While this method offers significant advantages, it may introduce challenges in processing very small or very large numbers, or in handling numbers with many decimal places. To address these, we propose incorporating scientific notation for extreme values and allowing variable-length tokenization for extensive fractional parts. Additionally, special tokens for common mathematical constants (e.g., π, e) and operations can further enhance numerical processing capabilities. Another problem is varieties in international numerical character representations and how these languages group values, which can cause problems with lack of generality and need for a highly standard input, that is not a good perspective for an LLM or any NL application, e.g.: It might even lower understanding of the model on hexadecimal digits, unless computationally intensive methods to detect these non-standard form numbers be applied. On the contrary, these numbers are rare, and this might be a good trade-off, specially in case of hexadecimal digits which are often used, in representation of binary data, not to perform arithmetic calculations on.

This proposed tokenization method for handling numbers in Large Language Models is entirely theoretical at this point. It has never been tested in practice, and its effectiveness remains unproven. While the approach seems promising in theory, addressing many of the shortcomings we've identified with standard Byte Pair Encoding for numerical operations, it's important to emphasize that this is a novel concept without any empirical validation.

The potential benefits we've discussed – such as improved magnitude estimation, easier approximation of multiplications, and enhanced numerical reasoning – are hypothetical advantages that would need to be rigorously tested and verified through extensive experimentation. While the tokenization changes are not major,

implementing this method in practice would require significant resources for training an LLM from ground-up with this new approach.

Furthermore, the real-world performance of this method could reveal unforeseen challenges or limitations that are not apparent in the theoretical framework. It's possible that while solving some issues, this approach might introduce new complexities or trade-offs that could impact model performance in unexpected ways.

Given its untested nature, this method should be viewed as a starting point for research and experimentation rather than a proven solution. Any implementation would need to be accompanied by comprehensive comparative studies against existing methods, covering a wide range of numerical tasks and scenarios to truly assess its value and viability in improving LLMs' numerical processing capabilities.

Another probable shortcoming is, introduction of this specialized numerical tokenization method could potentially introduce significant challenges in terms of processing speed and complexity. Unlike standard Byte Pair Encoding, which treats numbers as regular text, this method requires a separate parsing mechanism for numerical strings. This dual-track tokenization process – one for text and another for numbers – could increase the computational overhead during the tokenization phase. The system would need to constantly switch between these two modes, identifying numerical strings, parsing them according to the new rules (right-to-left for integers, left-to-right for fractions), and then grouping them into appropriate base-100 or base-1000 tokens. This process is inherently more complex than the uniform approach of BPE, potentially leading to slower tokenization speeds, especially for text with a high density of numerical data. Additionally, implementing this method would require careful handling of edge cases, such as numbers in scientific notation or mixed alphanumeric strings, further complicating the tokenization logic and potentially impacting processing speed.

Conclusion

The proposed method of tokenizing numbers in Large Language Models (LLMs) by grouping digits in sets of two or three and parsing integers right-to-left presents an intriguing avenue for scientific study. This approach, while untested, offers a theoretically sound foundation for addressing the numerical processing limitations currently observed in LLMs. By aligning tokenization more closely with human numerical intuition and mathematical principles, it has the potential to enhance models' ability to perform calculations, estimations, and numerical reasoning tasks.

The scientific value of this proposal lies in its potential to bridge the gap between symbolic mathematical reasoning and the distributed representations used in neural networks. By encoding numerical magnitude and place value more explicitly in the tokenization process, we may enable LLMs to develop a more robust understanding of numerical relationships. This could lead to improved performance across a wide range of tasks, from basic arithmetic to complex mathematical modeling, without relying on external tools or calculators.

When coupled with techniques like chain-of-thought prompting, this tokenization method could significantly enhance an LLM's reasoning capabilities.
Chain-of-thought prompting encourages models to break down complex problems into step-by-step solutions. With a more intuitive numerical representation, models might be better equipped to articulate and follow logical steps in mathematical reasoning, potentially leading to more accurate and explainable outcomes in numerical tasks.

Moreover, this approach opens up exciting possibilities for leveraging the model's own understanding to further refine numerical representations. As LLMs trained with this method develop more sophisticated numerical reasoning, we could potentially use their insights to iteratively improve the tokenization scheme. This could lead to a co-evolution of model architecture and numerical representation, potentially yielding more powerful and efficient systems for mathematical processing. Analyzing these models with monosemantic models introduced by anthropic may unveil new horizons into understanding how models learn basic arithmetic operations.

However, it is crucial to approach this method with caution and rigorous scientific scrutiny. As an untested theoretical concept, its practical implementation may reveal unforeseen challenges or limitations. The transition from standard tokenization methods to this specialized numerical approach could introduce complexities in model training and potential trade-offs in performance across different types of tasks. Extensive empirical testing would be necessary to validate its effectiveness and understand its impact on various aspects of model behavior.

Furthermore, while this method aims to improve numerical processing, we must be cautious about potential biases or limitations it might introduce. For instance, the focus on base-100 or base-1000 representations might inadvertently bias the model towards certain number systems or mathematical conventions (e.g.: Base 10), potentially impacting its performance in cross-cultural (e.g.: Roman Numerals) or interdisciplinary contexts. Rigorous testing across diverse mathematical tasks and cultural numerical representations would be essential.

In conclusion, this novel tokenization method presents a promising direction for enhancing numerical capabilities in LLMs, warranting further research and

experimentation. Its potential to improve model reasoning, coupled with the possibility of iterative refinement based on model understanding, makes it an exciting prospect for advancing the field of AI and mathematical processing. However, as with any new approach in AI, it should be pursued with a balance of enthusiasm and caution, ensuring thorough validation and consideration of its broader implications on model behavior and applicability across diverse contexts.

References

[1] Sennrich, R., Haddow, B., & Birch, A. (2015). Neural Machine Translation of Rare Words with Subword Units. Retrieved from https://arxiv.org/abs/1508.07909

[2] Lian, H., Xiong, Y., Niu, J., Mo, S., Su, Z., Lin, Z., Liu, P., Chen, H., & Ding, G. (2024). Scaffold-BPE: Enhancing Byte Pair Encoding with Simple and Effective Scaffold Token Removal. Retrieved from https://arxiv.org/abs/2404.17808

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. (2020). Language Models are Few-Shot Learners. Retrieved from https://arxiv.org/abs/2005.14165

[4] Taku Kudo, John Richardson. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Retrieved from https://arxiv.org/abs/1808.0622

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, Denny Zhou (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Retrieved from https://arxiv.org/abs/2201.11903

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from https://arxiv.org/abs/1810.04805

[7] Anthropic Team (2024). Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Retrieved from https://transformer-circuits.pub/2023/monosemantic-features/index.html