Beyond tokens: A new way to teach AI models to read

The Innovation Informatics Editorial Team*
*Correspondence: informatics@the-innovation.org

Received: October 12, 2026; Accepted: October 20, 2026; Published Online: October 23, 2026; https://doi.org/10.59717/j.xinn-inform.2026.100071
© 2026 The Author(s). This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).
Citation: The Innovation Informatics Editorial Team (2026). Beyond tokens: A new way to teach AI models to read. *The Innovation Informatics* **2**:100071.

HOW MANY TOKENS HAVE YOU BURNED THROUGH TODAY?

As Large Language Models (LLMs) become increasingly accessible, the term token is finding its way into everyday conversations. From asking a chatbot to generate a cartoon to using AI to analyze code, people are consuming tokens often without realizing it though.

Tokens are the basic units of text that most LLMs process. Rather than reading and generating text strictly letter by letter, these models typically divide each word into smaller pieces, then generate responses by predicting the next token based on the preceding sequence and statistical patterns learned from oceans of training data. Their ability to make these predictions underlies everything from answering simple questions to producing sophisticated scientific analyses.

Contrary to what many people might assume, LLMs do not divide words into tokens based on syllables. For example, the word approach occupies eight bytes in standard UTF-8 encoding, one for each English letter. Yet a tokenizer might represent it using just two tokens, namely "appro" and "ach", substantially shortening the sequence the model must process.

For frequently encountered English words, the compression can be even greater. A word such as international, containing 13 letters and therefore 13 UTF-8 bytes, may be represented by a single token according to the online token vocabulary of OpenAI.

The reduction matters. Shorter sequences generally allow the computationally expensive Transformer architectures underlying modern LLMs to process text more efficiently, which is a main reason why subword tokenization has become the dominant approach in language modeling.

As a trade-off, when several characters are represented by a single token, their individual identities are not explicitly presented to the model as separate input units. This can make seemingly elementary character-level tasks unexpectedly challenging — for example, an LLM may generate complex computer code yet stumble when asked to count the letters in a word, reverse a string or identify a character at a particular position.

A study published in Nature on Oct 7 suggests a possible solution to this.[1] Titled "Retrofitting language models to operate over bytes", the study introduces a method for converting existing, pretrained token-based LLMs into systems capable of processing text at the byte level, without sacrificing most of their original capabilities.

The research was conducted by Benjamin Minixhofer, Tyler Murray, Tomasz Limisiewicz, Anna Korhonen, Luke Zettlemoyer, Noah A. Smith, Edoardo M. Ponti, Luca Soldaini and Valentin Hofmann, affiliated with institutions including the Allen Institute for AI, the University of Cambridge, the University of Washington and Imperial College London.

Several previous LLMs, including the Byte Latent Transformer (BLT), H-Net and Dynamic Token Pooling (DTP), have explored ways to process text directly from bytes while combining neighboring bytes into larger computa-



Figure 1. Task-dependent dynamic token granularity.

tional units to improve efficiency.

However, these byte-level models have generally struggled to match the overall performance of leading subword-based LLMs, for which one important reason is that earlier byte-level models were typically trained from scratch. Meanwhile, conventional token-based models have benefited from enormous investments in pretraining, increasingly sophisticated architectures and extensive post-training improvements.

According to the study, instead of building another byte-level model from the ground up, the researchers developed a technique called byteification, which retrofits an existing subword-based LLM to operate on bytes (Figure 1).

The approach seeks to preserve the knowledge and computational capabilities already encoded in the pretrained model's Transformer while redesigning the mechanisms through which it receives and produces text. The researchers applied the technique to three established models: Qwen3 8B, Llama 3 8B and Olmo 3 7B, producing byte-level counterparts named Bwen 8B, Blama 8B and Bolmo 7B.

According to the study, the byteification process used approximately 49.1 billion tokens of training data, measured using the original tokenization scheme. The researchers estimated that the additional training required less than 1 percent of a typical large-scale pretraining budget.

The results showed that the converted models retained much of their predecessors' overall performance. On the researchers' aggregate evaluation suite, Bwen 8B scored 69.8, compared with 70.3 for the original Qwen3 8B; Blama 8B achieved 59.8, against 60.7 for Llama 3 8B; Bolmo 7B scored 62.5, slightly exceeding the 62.3 achieved by its source model, Olmo 3 7B.

These results do not establish that byte-level models are universally superior. Performance varied across individual tasks, and some capabilities, including certain mathematical and coding tasks, still showed declines.

Nevertheless, the narrow overall performance gaps represent an important development, while the converted models also demonstrated substantial improvements in character-level understanding, an area where conventional subword models can be disadvantaged by their input representation.

RETHINKING THE WAY TO ENHANCE EFFICIENCY

While performance is essential, efficiency remains a fundamental challenge for byte-level language modeling. Take the above-mentioned "international" again — A conventional tokenizer may encode the entire word as one token, whereas a direct byte-level representation contains 13 bytes. If a large Transformer had to process every byte independently, the longer sequence could substantially increase computational demands.

Researchers have therefore explored an intermediate solution known as latent tokenization. Instead of passing every byte separately through the most expensive layers of the model, these systems group neighboring bytes into larger segments, or patches; The Transformer then operates on the shorter sequence of patches, while smaller components handle the underlying byte-level information.

This architecture seeks to combine the flexibility of byte-level processing with the efficiency traditionally associated with tokenization.

Conventional subword models can shorten sequences by expanding their vocabularies, but larger vocabularies bring additional costs, including larger embedding and output layers. Byte-level approaches make it possible to maintain a small basic symbol space while allowing the model to learn how bytes should be grouped internally, and the resulting boundaries don't need to correspond to those established by a fixed tokenizer before training.

The study published on Nature does not demonstrate that every future language model should operate directly on bytes, but points out that existing token-based models can be converted into competitive byte-level systems while retaining much of the knowledge acquired through expensive pretraining.

Rather than taking tokenization as an immutable foundation of a language model, researchers may increasingly regard the interface between raw text and the Transformer as a component that can be redesigned, adapted and learned.

Future models could still group text into token-like units, but those units might be determined dynamically according to the content being processed and the computational resources required. Some passages might be handled in relatively large segments, while others could be divided more finely when precise character-level information matters.

Such flexibility could eventually reshape how models balance linguistic understanding, accuracy and efficiency. For now, fixed tokenizers remain central to most widely used LLMs. But the new research suggests that their dominance may reflect the history of model development as much as an unavoidable technical necessity.

And if this line of research continues, the next time you visit an online tokenizer to see how international is divided, there may be more than one meaningful answer.

REFERENCES

1. Minixhofer B., Murray T., Limisiewicz T., et al. (2026). Retrofitting language models to operate over bytes. *Nature*. DOI:10.1038/s41586-026-11111-4

DECLARATION OF INTERESTS

The author declares no competing interests.