Tokenization in the Theory of Knowledge

Subjects: Computer Science, Artificial Intelligence
Contributor: Robert Friedman

Tokenization is a procedure for recovering the elements of interest in a sequence of data. This term is commonly used to describe an initial step in the processing of programming languages, and also for the preparation of input data in the case of artificial neural networks; however, it is a generalizable concept that applies to reducing a complex form to its basic elements, whether in the context of computer science or in natural processes. In this entry, the general concept of a token and its attributes are defined, along with its role in different contexts, such as deep learning methods. Included here are suggestions for further theoretical and empirical analysis of tokenization, particularly regarding its use in deep learning, as it is a rate-limiting step and a possible bottleneck when the results do not meet expectations.

Keywords: token ; tokenization ; neural network ; deep learning ; cognition ; atomism ; natural language

In computer science, a token is an element of a programming language, such as the expression of a multiplication operation or a reserved keyword to exit a block of code. In a more general context, including from other areas of study, tokenization and tokens are useful to represent objects in general. In the case of a common computer program, tokenization is a procedure that creates tokens from a data source that consists of human readable text in the form of instructions (computer code). Given that the source code is a target to be compiled as a program, the idealized procedure is to begin with a tokenizer that transforms the text to a sequence of tokens, followed by delivery of the individual tokens to the subsequent step. Next, a parsing procedure transforms the tokenized sequence to a hierarchical data structure, along with validation and further processing of the code. Lastly, the machine code, a set of instructions as expected by a computer processing unit, is generated [1].

In an artificial environment, such as machine learning, there are other examples of tokenization apart from the idealized computer compiler, including the multi-layer artificial neural network and its weighted connections, in itself a type of computer program [2][3]. This network and its input data are likewise dependent on tokenization to create a sequence of tokens. As a neural network is expanded by adding layers of artificial neurons, this architecture may be referred to as deep learning.

In Nature, such as in the case of human speech, words are also divisible into a set of smaller elements, the phonemes, which are a type of set of tokens [4]. These phonemes are categorized by their occurrence along the natural boundaries of spoken words, emerging from the mechanics of speech and the cognitive processes. These elements, the phonemes, are further transformed, processed, and potentially routed to the other pathways of cognition. From an abstract perspective, these corresponding pathways encapsulate the internal representational forms, a type of intermediate language that originates with the tokenization process; these forms are expected to emerge from the cognitive processing of the phonemes. Otherwise, there is an expectation that the phonemes are unprocessed and uninterpretable along the downstream pathways of cognition.

Tokens are also the essential elements of text as observed in common documents. In this case, tokens are typically constrained in number, presumably a limitation of any human readable source that relies on the limits of human thought. However, in all cases, there is a potential for the generation of combinations and recombinations of tokens, and therefore the formation of new objects based on prior information, a process that corresponds to the mechanics of building knowledge. In deep learning, such as that implemented by the transformer architecture [5], the power is in collecting a large number of tokens as the essential elements for training a neural network [6][7][8]—a graph composed of artificial neurons (nodes) and their interconnections (edges) with weight value assignments [2][3].

References

1. Wirth, N. Compiler Construction; Addison Wesley Longman Publishing, Co.: Harlow, UK, 1996.
2. Hinton, G.E. Connectionist learning procedures. Artif. Intell. 1989, 40, 185–234.

3. Schmidhuber, J. Deep learning in neural networks: An overview. Neural Netw. 2015, 61, 85–117.
4. Michaelis, H.; Jones, D. A Phonetic Dictionary of the English Language; Collins, B., Mees, I.M., Eds.; Daniel Jones: Selected Works; Routledge: London, UK, 2002.
5. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing System, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
6. Zand, J.; Roberts, S. Mixture Density Conditional Generative Adversarial Network Models (MD-CGAN). Signals 2021, 2, 559–569.
7. Mena, F.; Olivares, P.; Bugueño, M.; Molina, G.; Araya, M. On the Quality of Deep Representations for Kepler Light Curves Using Variational Auto-Encoders. Signals 2021, 2, 706–728.
8. Saqib, M.; Anwar, A.; Anwar, S.; Petersson, L.; Sharma, N.; Blumenstein, M. COVID-19 Detection from Radiographs: Is Deep Learning Able to Handle the Crisis? Signals 2022, 3, 296–312.
