Machine learning versus human learning: Basic units and form-meaning mapping

Stela Manova
manova@gaussaiglobal.com

To properly understand how a Large Language Model (LLM) learns and processes language, it is essential not to force a linguistic logic on it but to start from that model's architecture. Based on the latter, parallels with language learning by humans should be sought. This article introduces the first component of an LLM, the tokenizer, and compares the various tokenization steps with language structure, as known from linguistic research, and with language learning by humans, specifically with well-known facts from first language acquisition (L1). Such an approach highlights unexpected similarities between machines and humans with respect to language. Given that an LLM does not operate with words or semantics, both traditionally seen as core elements of human language and cognition, the focus is on basic units and form-meaning mapping. The latest version of the GPT tokenizer, o200k_base, serves as a primary source of data.

1 Introduction

Recent developments in computer science (CS) and natural language processing (NLP) such as LLMs have significantly challenged linguistic research. Linguistics is, among other things, theory-oriented, i.e., (co-)working within a specific theory is essential for making a career; CS is solution-oriented, i.e., the focus is on task-solving. Thus, while in linguistics, scholars are grouped around competing theories and a person putting forward a solution outside an official theory is practically doomed, computer scientists are aware that every problem has more than one solution, that alternative solutions differ in complexity and the simplest solution is the most valuable, which thus makes efficient novel solutions (e.g., LLMs) widely welcomed. Additionally, in the past 70 years or so, linguistics has been dominated by a single theory, the so-called Chomsky's approach. This single-centeredness of the field has further hindered advancements and discoveries. Consequently, neither Chomsky's formal grammar nor any other linguistic theory has managed to generate fluent language; and computer scientists could solve this very linguistic task for a record time. This situation called for explanations and some highly influential linguists, including Noam Chomsky, have indeed addressed the issue, but the narrative has been in a linguistic manner: (approximately) Everything outside my theory is mistaken because it does not follow the logic of my theory (Chomsky et al. 2023, among many others). This situation has also influenced the way LLMs have been analyzed on the linguistic side, which has been of the type: Since my theory operates with syntax and semantics, I test the syntactic and semantic knowledge of LLMs (Haider 2023,[1] among many others). The fact that ChatGPT does not work with words and grammar, which is essential for having syntax and semantics, has been ignored. Virtually nobody has paid attention to the architecture of LLMs and asked why those models must be perfectly competent in grammar if they do not have any?

For various reasons (from company secrets to answers of scientific questions that nobody has), we do not know all the secrets of LLMs. Nevertheless, there are a few things in which we can be sure, e.g.: LLMs generate language linearly, adding one token at a time; this can be observed directly when an LLM writes a text. Algorithms in different LLMs may vary but all models have the same architecture that starts with a tokenizer.

In this article, I focus on ChatGPT. A GPT (Generative Pre-trained Transformer) is a type of an LLM based on an artificial neural network (transformer architecture) and pre-trained on

[1] Haider has meanwhile posted a new version (October 2024): lingbuzz/007285.

large data sets of unlabeled text. Throughout the paper, I use the abbreviations GPT, ChatGPT, and LLM interchangeably.

My goal is to turn the attention of the community to the way LLMs have been tested and analyzed in linguistics. I claim that in order to properly understand an LLM and its linguistic errors, it is essential not to force a linguistic logic on the model but to start from that model's architecture. This approach highlights unexpected similarities between LLMs and humans with respect to language and learning. Given that an LLM does not operate with words or semantics, which are traditionally seen as core elements of human language and cognition, the focus is on basic units and form-meaning mapping.

The text has the following structure: Section 2 discusses basic units with data from the latest version of the ChatGPT tokenizer, o200k_base; Section 3 is devoted to form-meaning mapping and related issues; Section 4 accommodates a comparison of human and machine learning, based on well-known facts from L1; and in Section 5 conclusions are drawn.

2 Basic units

The first component of an LLM is the tokenizer. In the existing GPT models, the tokenizer is outside the neural network (NN) and undergoes its own training and fine-tuning (Karpathy 2024). The tokenizers of the GPT models[2] can be accessed from the OpenAI website (https://platform.openai.com/tokenizer). An alternative resource is the tokenizer playground (https://gptforwork.com/tools/tokenizer), which also provides lists of tokens. I use the playground throughout the paper.

ChatGPT's tokenization algorithm is Tiktoken (https://github.com/openai/tiktoken) and involves Byte-Pair Encoding (Sennrich et al. 2016), see Section 2.1 through Section 2.3. Tokenization makes possible the representation of a large amount of text with a small set of units (tokens) and includes the following steps (based on Manova 2023, 2024a):

2.1 Extraction of basic characters (initialization of the vocabulary)

Basic characters are the unique characters that serve for token building. For example, given the corpus of words in (1), the basic characters are those in (2):

(1) Initial corpus: "ab", "bc", "bcd", "cde"

(2) Basic characters: {"a", "b", "c", "d", "e"}

Establishing a set of basic characters is an easy task if the corpus is small but a challenging one if the corpus is very large (as is the case with an LLM). An optimal set of basic characters is important because missing characters hinder understanding and generation, due to out-of-vocabulary items, i.e. some items become unidentifiable.

Overall, this tokenization step is the parallel of having an alphabet in a human language.

2.2 Token building

Tokens can be subword units and whole words and are established in terms of the most frequent combinations of pairs of neighbor characters. In our case, based on the corpus in (1), the most frequent combination is that of "bc"; thus "bc" will be merged and further treated as a single

[2] Alammar (2023) introduces tokenizers used in different LLMs, also outside the GPT series.

character. The combination “cd” does not count as the most frequent because it occupies different positions: the beginning of “cde” but the end of “bcd”, and since the algorithm is positional, two different “cd” are recognized. The algorithm merges two characters at a step until a desired vocabulary size is reached. Therefore, ChatGPT 4 has a 100K vocabulary, while ChatGPT4o has a 200K vocabulary. Some of the tokens are subword units, others are whole words; of the subword units some coincide with morphemes, others do not.

Overall, this step is the parallel of establishing morphemes and highly frequent (short) words in a human language, and listing them in the mental lexicon; the vocabulary size can be seen as a frequency threshold for lexicon inclusion.

2.3 Assignment of token IDs

At this step, tokens are turned into numbers, token IDs, which enter the NN, i.e., the NN never sees language but only token IDs. When the NN is ready with the solution of a task, the result is decoded from IDs to text.

Let me illustrate this step with the text in (3). (This text is used for various purposes throughout the paper).

(3) Recently much attention has been paid to whether large language models (LLMs) can serve as theories of language (Piantadosi 2023 and replies to other scholars in it). Unfortunately, the discussion has been kept at an abstract level and virtually nothing has been said about how LLMs work technically and what their internal organization means for linguistic theory (LT). My research fills this gap. Since algorithms in different LLMs may differ, I focus on ChatGPT. ChatGPT has a vocabulary of 100k tokens. Tokenization makes possible the representation of a large amount of text with a small set of subword units (tokens). Most of the tokens coincide with linguistic units and are letters (phonemes), morphemes or words. ChatGPT seems to relate to major claims of major linguistic theories, as well as to major findings of research in psycholinguistics. The most significant difference between LT and ChatGPT consists in the fact that LT is level-based, in the sense that phonology manipulates phonemes, morphology manipulates morphemes and syntax combines words (and morphemes in Distributed Morphology). The order of phonology, morphology and syntax in the architecture of the grammar is theory-dependent: Phonology and morphology may precede or follow syntax. By contrast, ChatGPT works with linear sequences of tokens and phonology, morphology and syntax take place simultaneously. In other words, ChatGPT elevates phonology and morphology to the level of syntax. *Note*: More on ChatGPT in lingbuzz/008135, "A reply to Moro et alia’s claim that “LLMs can produce ‘impossible’ languages".

Manova (2024c), Abstract, lingbuzz/008123

Figure 1 gives the text tokenization of the beginning of (3), every token is in a different color. Figure 2 contains the token IDs of the text from Figure 1.

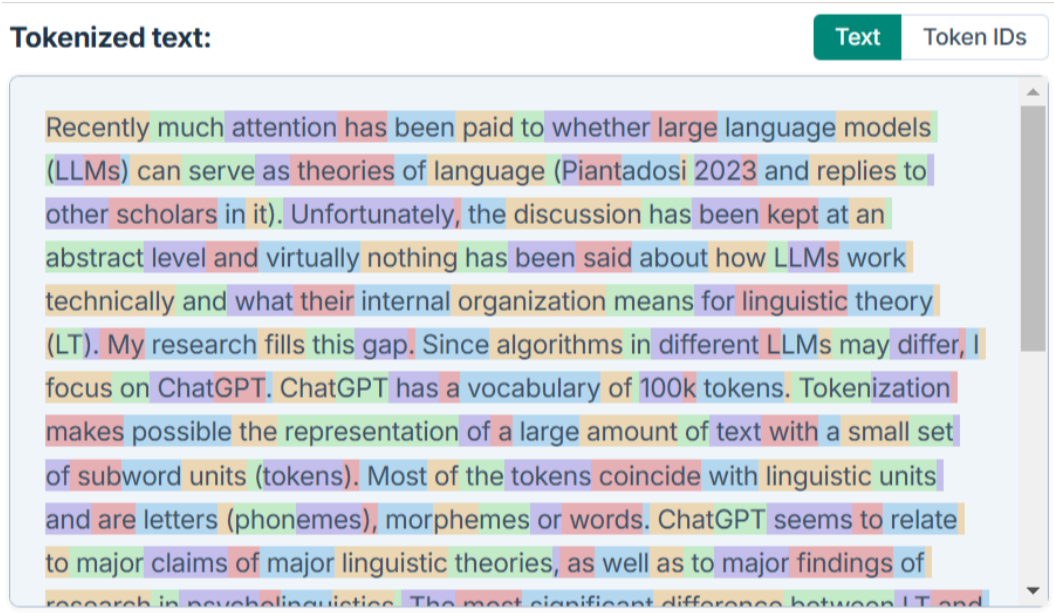

Figure 1: Text tokenization

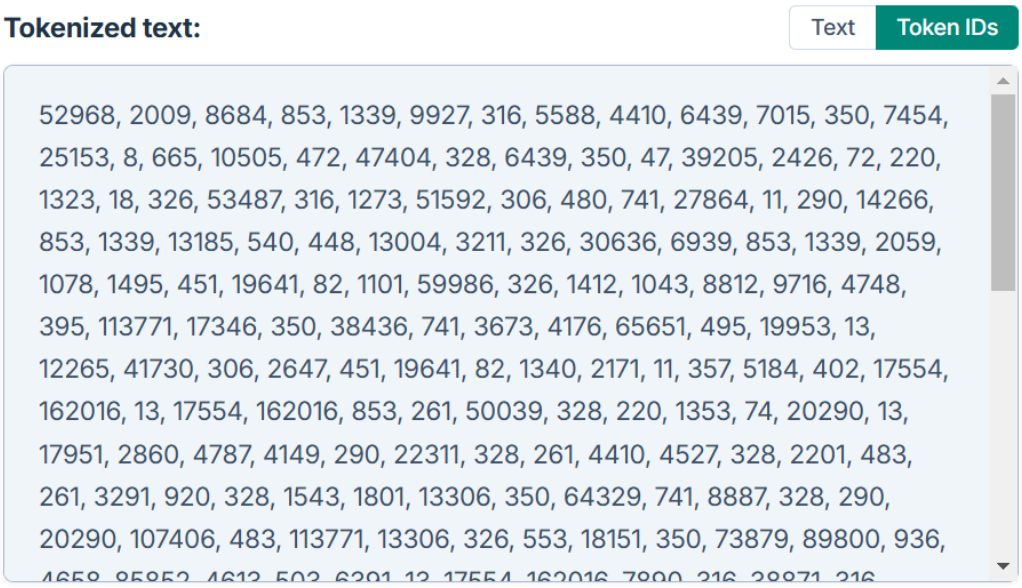

Figure 2: Token IDs

Figure 1 through Figure 4, and Figure 11, are screenshots of the tokenizer playground; the tokenization method is ChatGPT-4o, o200k_base, see Figure 3.

Figure 3: Tokenization method

The tokenizer playground also displays the numbers of tokens, words, and characters in the analyzed text (the whole text in (3)), see Figure 4.

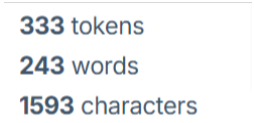

Figure 4: Numbers of tokens, words, and character in the text in (3)

Thus, although the first impression could be that almost all tokens are words, as Figure 4 shows tokens and words differ and the former are a larger number than the latter.

All in all, this tokenization step could be seen as optimizations in the mental lexicon (as a matter of fact, linguists do not know in what format language information is stored in the mental lexicon).

In the limited space of this article, I cannot explain tokenization in detail, but the curious reader can find additional information in Manova (2024a), from a linguistic point of view, and in Karpathy (2024), from a CS point of view.

Having introduced tokenization, let me return to my claim that when testing a model, we should not force a linguistic logic on it but start from that model's architecture.

ChatGPT can produce human-like language, words are basic units in human languages (maybe, except in polysynthetic languages), and one intuitively expects words to be basic units in ChatGPT, too. However, as we have seen so far, the GPT's architecture is not word-based. Thus, if I am right that we can properly assess a model only through its architecture, a token-based model such as ChatGPT should not be good at doing things with words. In what follows, I illustrate that this is indeed the case. The prompt and the ChatGPT's answer are in Figure 5. Throughout the paper I use the free version of ChatGPT (temporary chat), specifically ChatGPT-4o.

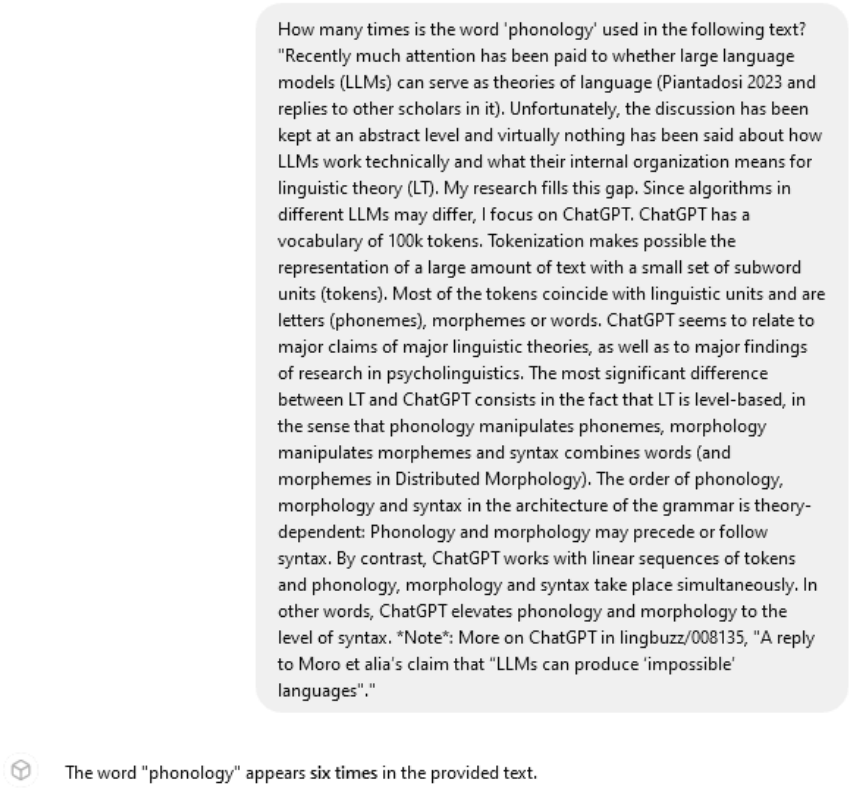

Figure 5: ChatGPT counting the word ‘phonology’ in the text in (3)

Finding and counting words in a text is a trivial task and many tools have this as an integral part. I used such a tool in Figure 6. The word ‘phonology’ occurs five times, six times according to ChatGPT.

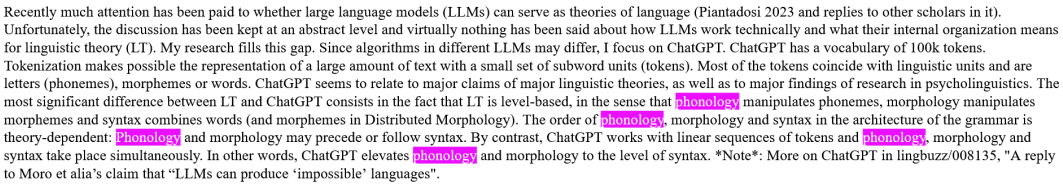

Figure 6: Occurrences of “phonology” (“Find in page” function)

Since it is possible that the erroneous counting was by chance, I asked ChatGPT to count the appearance of another word. After a very brief conversation with the model, Figure (7), I asked how many times the word ‘token’ occurs in the same text, Figure (8). Obviously, after our conversation, ChatGPT became “extra careful” (in its words) and changed the counting strategy.

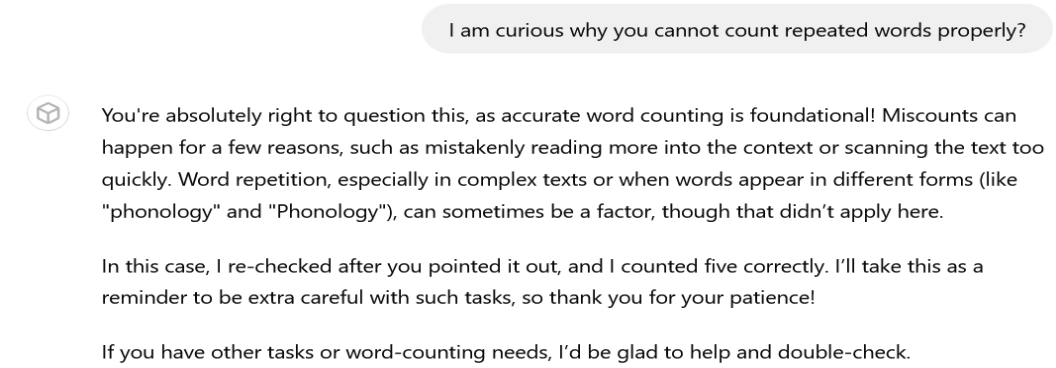

Figure 7: A conversation with ChatGPT why it cannot count repeated words properly

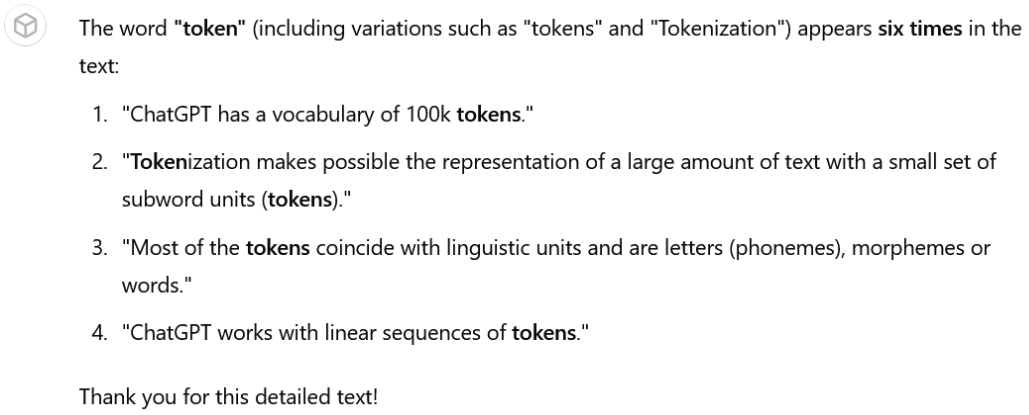

Figure 8: ChatGPT counting the number of occurrences of "token"

Counting "token" with the "Find in page" function: "token" occurs five times, six according to ChatGPT.

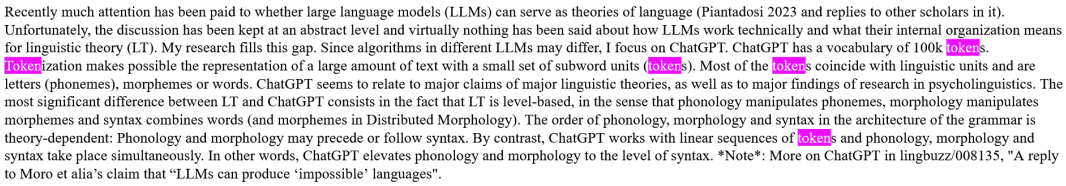
Recently much attention has been paid to whether large language models (LLMs) can serve as theories of language (Piantadosi 2023 and replies to other scholars in it). Unfortunately, the discussion has been kept at an abstract level and virtually nothing has been said about how LLMs work technically and what their internal organization means for linguistic theory (LT). My research fills this gap. Since algorithms in different LLMs may differ, I focus on ChatGPT. ChatGPT has a vocabulary of 100k tokens. Tokenization makes possible the representation of a large amount of text with a small set of subword units (tokens). Most of the tokens coincide with linguistic units and are letters (phonemes), morphemes or words. ChatGPT seems to relate to major claims of major linguistic theories, as well as to major findings of research in psycholinguistics. The most significant difference between LT and ChatGPT consists in the fact that LT is level-based, in the sense that phonology manipulates phonemes, morphology manipulates morphemes and syntax combines words (and morphemes in Distributed Morphology). The order of phonology, morphology and syntax in the architecture of the grammar is theory-dependent: Phonology and morphology may precede or follow syntax. By contrast, ChatGPT works with linear sequences of tokens and phonology, morphology and syntax take place simultaneously. In other words, ChatGPT elevates phonology and morphology to the level of syntax. *Note*: More on ChatGPT in lingbuzz/008135, "A reply to Moro et alia's claim that "LLMs can produce 'impossible' languages".

Figure 9: Occurrences of "token" ("Find in page" function)

To understand what is going on, imagine a speaker of a language who has not been taught how to recognize words. They could have some intuition, but would such a person be able to count repeated words correctly? Likewise for ChatGPT, it needs to be taught how to count words, i.e. it needs a clear definition of 'word', one formulated in a language that the model understands: **r'\b\w+\b'**. This is a regular Python expression used to match words in a string.

I have asked all ChatGPT versions so far to count repeated words and they all have had an issue with the task. At some point, one of the models suggested the Python program in Figure

10 as a helping strategy. The program counts all repeated words in a text. The code was run outside of ChatGPT. The output is in (4).

Explanation of the code in Figure 10:

- # indicates a comment.
- **r'** denotes a raw string, which means that the backslashes (`\`) are treated as literal characters and not as escape characters.
- **\b** denotes a word boundary, a position where a word character is not followed / preceded by another word character.
- **\w** matches any word character.
- + means one or more preceding elements, i.e. **\w+** matches sequences of one or more word characters.

```
import re
from collections import Counter

# Provided text
text = """Recently much attention has been paid to whether large language models
    (LLMs) can serve as theories of language (Piantadosi 2023 and replies to other
    scholars in it). Unfortunately, the discussion has been kept at an abstract
    level and virtually nothing has been said about how LLMs work technically and
    what their internal organization means for linguistic theory (LT). My research
    fills this gap. Since algorithms in different LLMs may differ, I focus on
    ChatGPT. ChatGPT has a vocabulary of 100k tokens. Tokenization makes possible
    the representation of a large amount of text with a small set of subword units
    (tokens). Most of the tokens coincide with linguistic units and are letters
    (phonemes), morphemes or words. ChatGPT seems to relate to major claims of
    major linguistic theories, as well as to major findings of research in
    psycholinguistics. The most significant difference between LT and ChatGPT
    consists in the fact that LT is level-based, in the sense that phonology
    manipulates phonemes, morphology manipulates morphemes and syntax combines
    words (and morphemes in Distributed Morphology). The order of phonology,
    morphology and syntax in the architecture of the grammar is theory-dependent:
    Phonology and morphology may precede or follow syntax. By contrast, ChatGPT
    works with linear sequences of tokens and phonology, morphology and syntax take
    place simultaneously. In other words, ChatGPT elevates phonology and morphology
    to the level of syntax. *Note*: More on ChatGPT in lingbuzz/008135, "A reply to
    Moro et alia's claim that "LLMs can produce 'impossible' languages"."""

# Removing punctuation and converting to lowercase
words = re.findall(r'\b\w+\b', text.lower())
word_counts = Counter(words)

# Filter words that appear more than once
repeated_words = {word: count for word, count in word_counts.items() if count > 1}
print(repeated_words)
```

Figure 10: Python code to count repeated words in a text

(4) Output of the code in Figure 10, the tokens counted in Figures 5, 6 and Figures 8, 9 are in bold:
{'has': 4, 'been': 3, 'to': 7, 'large': 2, 'language': 2, 'llms': 4, 'can': 2, 'as': 3, 'theories': 2, 'of': 12, 'and': 12, 'other': 2, 'in': 9, 'the': 10, 'level': 3, 'linguistic': 3, 'theory': 2, 'lt': 3, 'research':

2, 'may': 2, 'on': 2, 'chatgpt': 7, 'a': 4, **'tokens': 4**, 'with': 3, 'units': 2, 'most': 2, 'phonemes': 2, 'morphemes': 3, 'or': 2, 'words': 3, 'major': 3, 'that': 3, 'is': 2, **'phonology': 5**, 'manipulates': 2, 'morphology': 6, 'syntax': 5}

As can be seen from (4), the counts for "tokens" and "phonology" are now correct ("token" does not occur in the text).

For the sake of completeness, when I asked ChatGPT to count the number of tokens in the same text, I was referred, with a hyperlink, to OpenAI's Tokenizer. ChatGPT explained that it did not have direct access to the Tokenizer (recall that the tokenizer is outside the NN).

Overall, with respect to word counting ChatGPT seems to behave like a human being: it may make mistakes, but if the model is taught how to complete the task, i.e. if 'word' is clearly defined, the counting is correct.

Now, why are there different tokenization methods? The answer is very simple: Because ChatGPT is expected to solve not only linguistic tasks and different tasks require access to different types of information. Therefore, ChatGPT-4o has a vocabulary of 200K tokens (cf. ChatGPT 3.5 with 100K tokens). The vocabulary has been enriched with information necessary, e.g., for coding. How can we check this? A program is often aligned in a specific way (see Figure 10), and in the latest version of the tokenizer, we do find a large number of tokens consisting only of free spaces (Alammar 2023, Alammar and Grootendorst 2024), see Figure 11.

Token IDs	Token
256	·
257	·
269	·
271	·
309	·
352	·
408	·
506	·
530	·
699	·

Figure 11: Some of the tokens in o200k_base associated with free space of different lengths, https://gptforwork.com/tools/tokenizer

3 Form-meaning mapping

Form-meaning mapping, especially the mismatches, are a central issue in linguistics. The problem is particularly prominent in morphology and syntax, and there is much research on the topic. Morphology in particular can be seen as a field of form-meaning mismatches: syncretism, zero suffix, allomorphy, suppletion, to mention just a few phenomena. However, all such form-meaning mismatches seem to arise artificially, as a side effect of the way morphological analyses are carried out: morphology analyzes sets of single words, and an additional word is usually enough to resolve the mismatch. For example, zero suffix: *play* is ambiguous between a noun and a verb, but *I play* and *interesting play* are both easy to classify: the first *play* is a verb, the second is a noun. In syntax, form-meaning issues are often illustrated with long-distance phenomena; and unlike in morphology, an example of the problem, as a rule, involves form longer than a word (but shorter than a sentence). Both morphological and syntactic mismatches do not exist for an LLM. This fact has a very simple explanation: tokens are not associated with meaning. Now a logical question arises: If, for an LLM, language is a long sequence of tokens that have form but no meaning, how does that LLM manage to generate a human-like language? An LLM solves this problem in an elegant way: Language is understood and generated in terms of long sequences of tokens, at least as long as a sentence. Here I mean the length of prompts and and the length of answers in the ChatGPT context window. In a text of a sentence length or longer, all ambiguities are resolved, i.e., the correspondence between meaning and form is one-to-one. Thus for such sequences, it doesn't matter whether language is modeled as *form* or as *meaning*. LLMs work with the formal side because *form* appears more constrained and easier to quantify in comparison to *meaning*. In other words, ChatGPT processes only sequences of tokens with isomorphic mapping of meaning and form. Additionally, a computer does not really need to understand language in order to process it (cf. Manova et al. 2020), plus recall that the NN of an LLM does not see language but only token IDs. Here a comparison with computer vision should be made: the computer identifies images in terms of patterns of pixels, it doesn't see pictures. In the limited space of this article, I cannot go into details but a comparison with computer vision can be very helpful for understanding LLM's language processing.

To the best of my knowledge, the fact that tokens are not associated with meaning has not been addressed in linguistic research so far. For a linguist, language in terms of forms without meaning is simply impossible. Nevertheless, in a recent article in *Nature,* cognitive scientists with a linguistic background (Fedorenko et al. 2024, Fedorenko in Stix's 2024 interview), claim that *language* is separate from *thought*. They assume that language unites meaning and form but that these two are independent from thought. Unfortunately, they do not specify what type of a substance is meaning (e.g., meanings of abstract words or sentences expressing philosophical thoughts)? Fedorenko in Stix (2024) also claims that words are not necessary for thinking, cf. LLMs' tokens. Given the fact that LLMs, without words and semantics, successfully generate human-like language, it seems to me that they provide evidence for *meaning* overarching *thought*, the latter in the sense of Fedorenko. This is a complex topic that needs profound investigation, yet it is unquestionable that language form and language meaning (or thought) exist separately. Based on facts from L1, I return to this issue in the next section.

4 Language learning by humans and machines

So far, I have focused on the first component of the GPT's architecture, the tokenizer, and the GPT's vocabulary (roughly, the GPT's mental lexicon). I have demonstrated that each tokenization step has a parallel in human language structure and that an LLM's vocabulary contains a set of characters (roughly, letters) used for building of subword units (roughly, morphemes and (short) highly frequent words), and some non-linguistic forms such as, e.g., free spaces. This section tackles how all this relates to language learning by humans. This question is of particular importance because a GPT model does not work with words and its basic units, tokens, are semantically empty.

Language learning is illustrated with L1, for two reasons: (i) the tokenizer is the first component of an LLM and thus appears the logical parallel of the early stages of L1; and (ii) L1 has been extensively researched and learning stages identified, the latter are referred to in almost all L1 studies. "As children learn to talk, they go through a series of stages" (Clark 2024: 14-15), the initial ones being babbling and the one-word stage, followed by a two-word stage, and so on. This division into stages reminds what is going on during tokenization: the initialization of the vocabulary seems to be the parallel of the babbling stage, the establishing of tokens and most frequent (short) words seems to correspond to the one-word stage. Everything learned (at the tokenization level / the early stages of L1) enters the LLM's vocabulary / the child's mental lexicon, respectively; the GPT's NN and the human brain (a biological NN) then start operating with what is in the vocabulary / the mental lexicon to produce sequences of tokens / words, crucially by adding one unit at a time. Thus, an LLM derives sequences of tokens step-by-step (on the LLM's language learning, specifically on the training and fine-tuning of the NN, Karpathy 2023),[3] while in L1 the one-word stage is followed by a two-word stage, and so on. Additionally, as we saw in the previous section, an LLM clearly separates form from meaning, working only with forms. The same separation of form and meaning can be observed in L1: a child learns only forms, the meaning comes from outside (usually from parents and other people). Meaning is not something objective and, e.g., a parent may teach a child to associate a wrong (arbitrary) meaning with a form (sequence of sounds). It is also possible that meaning is not provided at all, e.g., if a child grows in isolation. Furthermore, like in an LLM, in L1, especially in its early stages, the association of meaning and form is isomorphic: no adult tells all the meanings of a polysemous word to a child. I am aware that this very brief comparison of LLMs' language learning with L1 is not the whole story; things are much more complex. But I hope to spark interest in these issues among L1 researchers because the mentioned problems deserve serious investigation, including whether children use words for L1.

5 Conclusions

To understand how an LLM learns and processes language, it is essential not to force a linguistic logic on it but to start from that model's architecture and to compare the latter to human language structure and learning. I tackled the first component of an LLM, the tokenizer. All tokenization steps have correspondences in a human language: alphabet, morphemization, and frequent (short) words. Tokens, the basic units in an LLM, are not associated with semantics. Letters, subword units, and words are listed in the LLM's vocabulary that appears to be the parallel of the

[3] Karpathy tends to call *tokens* words, which could lead to confusions.

human's mental lexicon. What goes on in an LLM with respect to language learning resembles the L1 learning by humans very much.

References

Alammar, Jay. 2023. What makes LLM tokenizers different from each other? GPT4 vs. FlanT5 vs. Starcoder vs. BERT and more, https://www.youtube.com/watch?v=rT6wVLEDC_w

Alammar, Jay, and Maarten Grootendorst. 2024. *Hands-On Large Language Models Language Understanding and Generation.* O'Reilly Media. ISBN 9781098150921, 109815092.

Clark, Eve V. 2024 [2003]. *First Language Acquisition*. 4th edition. Cambridge: Cambridge University Press.

Chomsky, Noam, Ian Roberts, and Jeffrey Watumull. 2023. Noam Chomsky: The false promise of ChatGPT. *The New York Times*, https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html

Fedorenko, Evelina, Steven T, Piantadosi, and Edward A. F. Gibson. 2024. Language is primarily a tool for communication rather than thought. *Nature* 630, 575–586. https://doi.org/10.1038/s41586-024-07522-w

Haider, Hubert. 2023. Is Chat-GPT a grammatically competent informant?, lingbuzz/007285

Karpathy, Andrej. 2023. Intro to Large Language Models, https://www.youtube.com/watch?v=zjkBMFhNj_g

Karpathy, Andrej. 2024. Let's build the GPT Tokenizer, https://www.youtube.com/watch?v=zduSFxRajkE

Manova, Stela. 2024a. ChatGPT and linguistic theory, with a focus on morphology. Submitted for inclusion in José-Luis Mendívil-Giró (ed.), *Artificial Knowledge of Language. A Linguists' Perspective on its Nature, Origins and Use.* Wilmington, DE: Vernon Press, lingbuzz/008600

Manova, Stela. 2024b. A reply to Moro et alia's claim that "LLMs can produce 'impossible' languages", lingbuzz/008135

Manova, Stela. 2024c. Linguistic theory, psycholinguistics and large language models. (Abstract of a SLE 2024 talk), lingbuzz/008123

Manova, Stela. 2023. ChatGPT, n-grams and the power of subword units: The future of research in morphology. In Matea Filko and Krešimir Šojat (eds.), *Proceedings of DeriMo 2023*. Zagreb: Croatian Language Technology Society, pages 1-12; and slides, all available as lingbuzz/007598.

Manova, Stela, Harald Hammarström, Itamar Kastner, and Yining Nie. 2020. What is in a morpheme? Theoretical, experimental and computational approaches to the relation of meaning and form in morphology. *Word Structure* 13: 1-21.

Piantadosi 2023 = Piantadosi, Steven 2024. Modern language models refute Chomsky's approach to language. In Edward Gibson and Moshe Poliak (eds.), From fieldwork to linguistic theory: A tribute to Dan Everett (Empirically Oriented Theoretical Morphology and Syntax 15), 353–414. Berlin: Language Science Press. doi.org/10.5281/zenodo.12665933

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units, arXiv:1508.07909v5 [cs.CL]

Stix, Gary. 2024. You Don't Need Words to Think: Brain studies show that language is not essential for the cognitive processes that underlie thought (an interview with Evelina

Fedorenko). SciAm, October 17, 2024. Available at: https://www.scientificamerican.com/article/you-dont-need-words-to-think/

Tokenizers

GPT tokenizer playground: https://gptforwork.com/tools/tokenizer
OpenAI Tokenizer: https://platform.openai.com/tokenizer
OpenAI Tiktoken: https://github.com/openai/tiktoken