

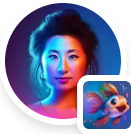
“always” are its own token, whereas longer or less common words such as “atrocious”, “precocious”, and “supercalifragilisticexpialidocious” are broken into smaller subwords.



GPT-3 Codex

It's supercalifragilisticexpialidocious
Even though the sound of it is something quite atrocious
If you say it loud enough you'll always sound precocious

Clear Show example

Tokens	Characters
70	457



Discover more from  **art fish intelligence** 

♥ stories told with ~ art fish ~ intelligence ♥ code · data · art · AI

Launched 6 months ago

Type your email...

Subscribe

Continue reading >

Sign in

AI's tokenizer

leading to disparities in the
rent languages. For
more tokens than a

This
num
exa
sim

Language	ISO	Text	↑ Num Tokens
English	en-US	what will the weather be next week	7
Spanish	es-ES	qué tiempo hará la semana que viene	8
Korean	ko-KR	다음 주 날씨 어때	12
Burmese	my-MM	နောက်တစ်ပတ် ရာသီဥတုဘယ်လိုရှိမလဲ	61
Amharic	am-ET	በሚቀጥለው ሳምንት ሳምንት አየሩ ምንድን	69

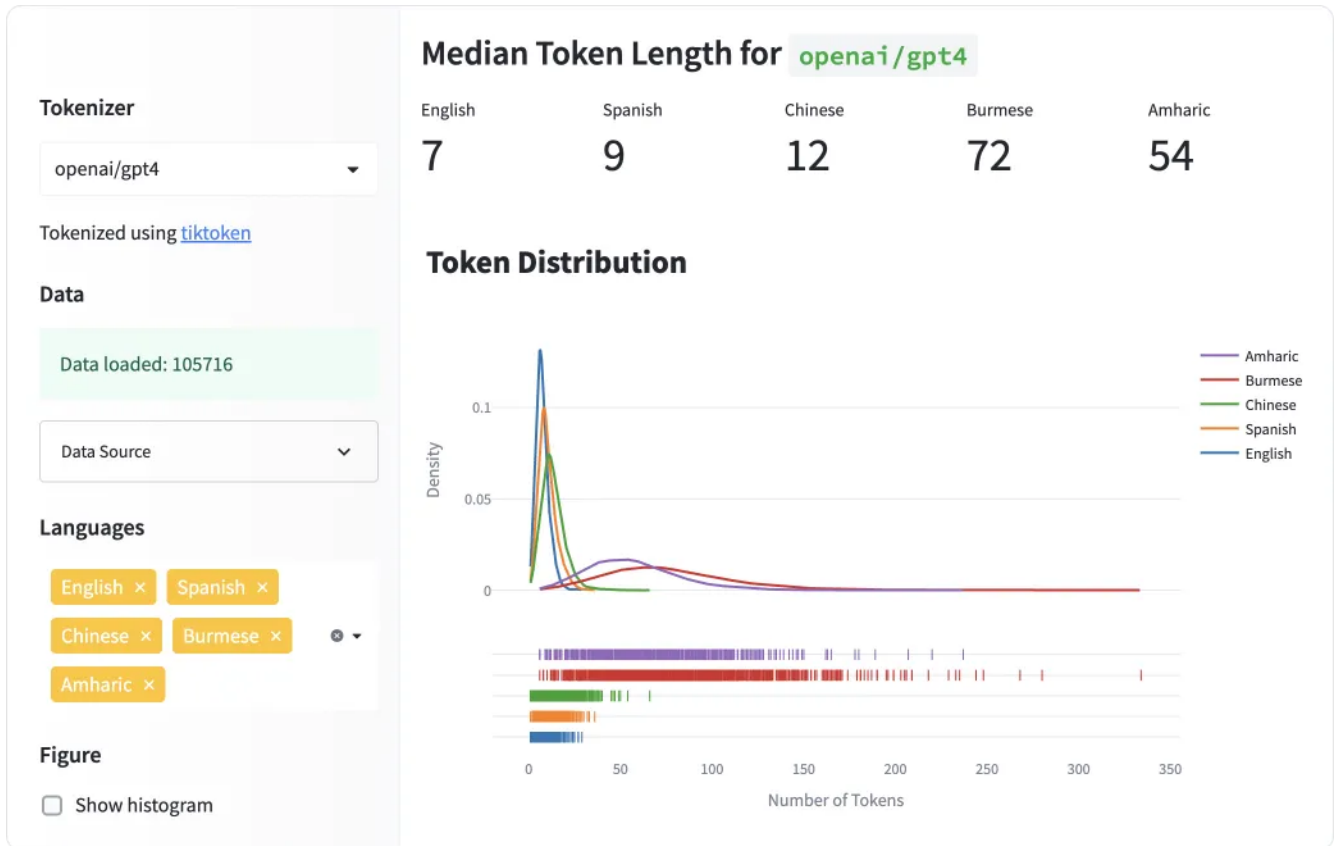
An example of the same message translated into five languages and the corresponding number of tokens required to tokenize that message (using OpenAI's tokenizer). The text comes from [Amazon's MASSIVE dataset](#).

In this article, I explore the tokenization process and how it varies across different languages:

- Analysis of token distributions in a parallel dataset of short messages that have been translated into 52 different languages
- Some languages, such as Armenian or Burmese, require **9 to 10 times more tokens than English** to tokenize comparable messages
- The impact of this language disparity
- **This phenomenon is not new to AI** — this is consistent with what we observe in Morse code and computer fonts

Try it yourself!

Try out the [exploratory dashboard I made](#), available on [HuggingFace spaces](#). Here, you can compare the token lengths for different languages and for different tokenizers (which was not explored in this article, but which I explore the reader to do on their own).



🎨 art fish intelligence 🌞 is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

Data

MASSIVE is a parallel dataset introduced by Amazon consisting of 1 million realistic, parallel short texts translated across 52 languages and 18 domains. I used the dev split of

the dataset, which consists of 2033 texts translated into each of the languages. The dataset is available on [HuggingFace](#) and is licensed under the [CC BY 4.0 license](#).

A focus on OpenAI's Tokenizers

While many other language model tokenizers exist, this article mainly focuses on [OpenAI's Byte Pair Encoding \(BPE\) tokenizer](#) (used by ChatGPT and GPT-4) for three main reasons:

- First, [Denys Linkov's article](#) compared several tokenizers and found that GPT-2's tokenizer had the highest token length disparity among different languages. This prompted me to concentrate on OpenAI models, including GPT-2 and its successors.
- Second, since we lack insight into ChatGPT's full training dataset, investigating OpenAI's black box models and tokenizers help to better understand their behaviors and outputs.
- Finally, the widespread adoption of ChatGPT in various applications (from language learning platforms like [Duolingo](#) to social media apps like [Snapchat](#)) highlights the importance of understanding tokenization nuances to ensure equitable language processing across diverse linguistic communities.

To calculate the number of tokens a text contains, I use the `cl100k_base` tokenizer available on [tiktoken](#), which is the BPE tokenizer used by OpenAI's ChatGPT models (``gpt-3.5-turbo`` and ``gpt-4``).

Thanks for reading 🎨 art fish intelligence 🌟.

This post is public so feel free to share it.

Results

Some languages consistently tokenize to longer lengths

The following distribution plot compares the distribution of token lengths for five languages. The curve for English is tall and narrow, meaning that English texts

consistently tokenize to a smaller number of tokens. On the other hand, the curve for languages such as Hindi and Burmese are short and wide, meaning that these languages tokenize texts into many more tokens.

Distribution of token lengths for all 2033 messages and 52 languages. Five of the languages have been bolded and colored; the rest are shown in gray.

English has the shortest median token length

For each language, I calculated the median token length for all of the texts in the dataset. The following chart compares a subset of the languages. English texts had the smallest median length of 7 tokens and Burmese texts had the largest median length of 72 tokens. Romance languages such as Spanish, French, and Portuguese tended to result in a similar number of tokens as English.

A subset of the 52 languages and their median token length.

As English had the shortest median token length, I calculated the ratio of the other languages' median token length to that of English. Languages such as Hindi and Bengali (over 800 million people speak either of these languages) resulted in a median token length of about 5 times that of English. The ratio is 9 times that of English for Armenian and over 10 times that of English for Burmese. In other words, **to express the same sentiment, some languages require up to 10 times more tokens.**

A subset of the 52 languages and the ratio of that language's median token length to that of English.

Thanks for reading 🎨 art fish intelligence 🌟.

This post is public so feel free to share it.

Discussion

Implications of tokenization language disparity

Overall, requiring more tokens (to tokenize the same message in a different language) means:

- You're limited by how much information you can put in the prompt (because the context window is fixed). As of March 2023, GPT-3 could take up to 4K tokens and GPT-4 could take up to 8K or 32K tokens in its input. ¹
- It costs more money
- It takes longer to run

OpenAI's models are increasingly being used in countries where English is not the dominant language. According to SimilarWeb.com, the United States only accounted for 10% of the traffic sent to ChatGPT in Jan-March 2023.

Top 5 countries sending the most traffic to chat.openai.com in Jan-March 2023. Sourced from [similarweb.com](https://www.similarweb.com) on May 2, 2023.

Additionally, ChatGPT was used **in Pakistan to grant bail in a juvenile kidnapping case** and **in Japan for administrative tasks**. As ChatGPT and similar models are becoming increasingly integrated into products and services worldwide, it is crucial to understand and address such inequalities.

Language Disparity in Natural Language Processing

This digital divide in natural language processing (NLP) is an active area of research. 70% of research papers published in a computational linguistics conference only evaluated English.² Multilingual models perform worse on several NLP tasks on low resource languages than on high resource languages such as English.³ According to **W3Techs** (World Wide Web Technology Surveys), English dominates more than half (55.6%) of the content on the Internet.⁴

Percentages of websites using various content languages (as of April 30, 2023). Data source:

https://w3techs.com/technologies/overview/content_language.

Similarly, English makes up **over 46% of the Common Crawl corpus** (billions of webpages from the Internet **crawled for over a decade**), versions of which have been used to train many large languages such as Google's T5 and OpenAI's GPT-3 (and likely ChatGPT and GPT-4). Common Crawl makes up 60% of GPT-3 training data.⁵

Addressing the digital divide in NLP is crucial to ensure equitable language representation and performance in AI-driven technologies. Bridging this gap calls for a

concerted effort from researchers, developers, and linguists to prioritize and invest in the development of low-resource languages, fostering a more inclusive and diverse linguistic landscape in the realm of natural language processing.

Historical example: Representing Chinese Typography using Morse Code

Such a disparity of technological costs for different languages is not new to AI or even to computing.

Over a hundred years ago, telegraphy, a revolutionary technology of its time (“the internet of its era”), faced language inequities similar to those we see in today’s large language models. Despite its promises of open exchange and collaboration, telegraphy exhibited discrepancies in speed and cost across languages. For instance, encoding and transmitting a message in Chinese (compared to an equivalent message in English) was

- 2 times as expensive
- Took 15-20 times longer

Sound familiar?

Telegraphy was “designed first and foremost for Western alphabetic languages, English above all.”⁶ Morse code assigned different lengths and costs to dots and dashes, resulting in a cost-efficient system for English. However, the Chinese language, which relies on ideograms, faced challenges in telegraphy. A Frenchman named Viguier devised a mapping system for Chinese characters to Morse code.

Essentially, each Chinese ideogram was mapped to a four-digit code, which had to then be translated into Morse code. This took a long time looking up the codes in the codebook (which lacked meaningful correlations) and was more costly to transmit (as each character was represented by four digits, and a single digit was more expensive to transmit than a single letter). This practice put the Chinese language at a disadvantage compared to other languages in terms of telegraphic speed and cost.

Manuscript on left from Zhang Deyim *Dianxin xinfā* 电报新法, 1873. Danish National Archives. <http://www5.kb.dk/permalink/2006/manus/350/eng/32/>. Red circle drawn in by author.

Another example: Inequity in representing fonts

Initially, I tried to visualize all 52 languages in a single word cloud. I ended up with something like this, where a majority of the languages were not rendered properly.

Word cloud visualizing “hey” in 52 languages. Many of the languages (including Arabic, Hindi, and Korean) cannot be rendered using a single font (depicted is the default WordCloud font DroidSansMono). Size corresponds to the number of tokens required to represent “hey” in that language.

This led me down a rabbit hole of trying to find a font that could render all of the language scripts. I went on Google Fonts to find this perfect font and found that one did not exist. Below is a screenshot showing how these 52 languages would render in 3 different fonts from Google Fonts.

To generate the word cloud at the beginning of this article, I (ehm) manually downloaded the 17 font files necessary to render all of the language scripts and displayed words one at a time. While I got the desired effect, it was a lot more work than it would have been if, for example, all of my languages used the same script (such as the Latin alphabet).



Conclusion

In this article, I explored the language disparity in language models by looking at how they process text through tokenization.

- Using a dataset of parallel texts translated into 52 languages, I showed that some languages require up to 10 times more tokens to express the same message in English
- I shared a [dashboard where you can explore different languages and tokenizers](#)
- I discussed the impacts of this disparity on certain languages in terms of performance, monetary cost, and time

- I showed how this pattern of linguistic technological disparity is not new, comparing the phenomenon to the historical case of Chinese Morse code and telegraphy

Language disparities in NLP tokenization reveal a pressing issue in AI: equity and inclusivity. As models like ChatGPT are predominantly trained on English, non-Indo-European and non-Latin script languages face barriers due to prohibitive tokenization costs. Addressing these disparities is essential to ensure a more inclusive and accessible future for artificial intelligence, ultimately benefiting diverse linguistic communities worldwide.

 art fish intelligence  is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

<input type="text" value="Type your email..."/>	<input type="button" value="Subscribe"/>
---	--

APPENDIX

Byte-Pair Encoding Tokenization

In the realm of natural language processing, tokenizers play a crucial role in enabling language models to process and understand text. Different models use different methods for tokenizing a sentence, such as splitting it into words, into characters, or into parts of words (also known as subwords; e.g. splitting "constantly" into "constant" and "ly").

One common tokenization is called **Byte-Pair Encoding** (BPE). This is the encoding used by OpenAI for their ChatGPT models. BPE is meant to decompose rare words into

meaningful subwords while keeping frequently used words intact. A comprehensive explanation of the BPE algorithm can be found on the [HuggingFace Transformers course](#).

Deeper Dive into Token Distribution for Languages

I augmented Amazon's MASSIVE dataset by using information about each of the 52 languages using the infobox section of that language's Wikipedia page, obtaining information such as writing script (e.g. Latin, Arabic alphabet) and main geographic region the language is predominant in (if relevant). I additionally use metadata from [The World Atlas of Language Structures](#) to obtain information such as [language family](#) (e.g. Indo-European, Sino-Tibetan).⁷

Note that the following analyses in this article uphold the assumptions made by Wikipedia, The World Atlas of Language Structures, and by the Amazon MASSIVE dataset. Since I am not a linguistics expert, I had to assume that whatever on Wikipedia and the World Atlas were canonically accepted as correct with regards to dominant geographic region or language family.

Also, there are debates about what constitutes a language versus a dialect. For example, while languages such as Chinese and Arabic have different forms that people may not understand, they are still called single languages. On the other hand, Hindi and Urdu are very similar and are sometimes grouped together as one language called Hindustani. Because of these challenges, we need to be careful when deciding what counts as a language or a dialect.

Breakdown by language. I chose the [12 most spoken languages](#) (a combination of both first-language and second-language speakers).

Breakdown by language family. Indo-European (e.g. Swedish, French), Austronesian languages (e.g. Indonesian, Tagalog), and Uralic languages (e.g. Hungarian, Finnish) resulted in shorter tokens. Dravidian languages (e.g. Tamil, Kannada) tended to have longer tokens.

Breakdown by main geographic region. Not all languages were specific to a single geographic region (such as Arabic, English, and Spanish, which are spread across many regions) — these languages were removed from this section. Languages spoken mostly in Europe tend to be shorter in token length, while languages spoken mostly in the Middle East, Central Asia, and the Horn of Africa tended to be longer in token length.

Breakdown by writing script. Other than the Latin, Arabic, and Cyrillic alphabets, all other languages use their own unique script. While the latter combines many very different unique scripts (such as Korean, Hebrew, and Georgian scripts), these unique scripts definitely tokenize to longer values. Compared to Latin-based scripts, which tokenize to shorter values.

English almost always ranks #1

For each text in the dataset, I ranked all languages based on number of tokens — the language with the least tokens was ranked #1 and the one with the most tokens was ranked #52. Then, I plotted the distribution of each language's *ranking*. Essentially, this should show how each language's token length compares with the other languages in this dataset. In the below figure, I labeled a few of the languages (the other languages show up as gray lines in the background).

While there were a few cases where some languages' tokens were fewer than that of English (such as a few examples in Indonesian or Norwegian), English almost always ranked number one. Does this come as a surprise to anyone? What surprised me most

was that there was no clear #2 or #3. English language texts consistently produce the shortest tokens, and the ranking fluctuates a bit more for other languages.

Quantifying token distributions differences using Earth Mover's Distance

To quantify how different the token length distribution between two languages were, I calculated the **earth mover's distance** (also known as the **Wasserstein distance**) between two distributions. Essentially, this metric calculates the minimum amount of “work” required to transform one distribution into another. Larger values mean the distributions are farther apart (more different) while smaller values mean the distributions are quite similar.

Here is a small subset of languages. Note that the distance says nothing about the length of the tokens, just how similar the distribution of token lengths are for two languages. For example, Arabic and Russian have similar distributions even though the languages themselves are not similar in a linguistic sense.

-
- 1 OpenAI. "Models". *OpenAI API*. Archived from the original on March 17, 2023. Retrieved March 18, 2023.
 - 2 Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2022. *Square One Bias in NLP: Towards a Multi-Dimensional Exploration of the Research Manifold*. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354, Dublin, Ireland. Association for Computational Linguistics.
 - 3 Shijie Wu and Mark Dredze. 2020. *Are All Languages Created Equal in Multilingual BERT?*. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

- 4 Usage statistics of content languages for websites". Archived from the original on 30 April 2023.
- 5 Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
- 6 Jin Tsu. *Kingdom of Characters: The Language Revolution That Made China Modern*. New York: Riverhead Books, 2022 (p. 124).
- 7 Dryer, Matthew S. & Haspelmath, Martin (eds.) 2013. *WALS Online (v2020.3)* [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.7385533>. Available online at <https://wals.info>, Accessed on 2023-04-30.



6 Comments



Write a comment...



john eckstein May 7  Liked by **Yennie Jun**

Great project. Would love to see a linguists take on it

 LIKE (1)  REPLY ...



Eta 8 hr ago

Great article. But I am curious how the number of tokens for the amharic language became 69? I count only 5 words!

 LIKE  REPLY ...

1 reply by **Yennie Jun**

4 more comments...

© 2023 Yennie · [Privacy](#) · [Terms](#) · [Collection notice](#)
[Substack](#) is the home for great writing