

Vocab Size	Time	# Tokens	Token Segments
265	27.64s	16	A·l·e·x·a·n·d·e·r· ·G·a·r·c·i·a
4,096	30.55s	7	A·lex·and·er· G·arc·ia
32,768	32.00s	4	Alex·ander· Garc·ia

The training used for this is 8 FineWeb data shards. Vocabulary's size has an inverse relation to amount of tokens for my name. The higher the Vocab the lower the tokens. This happens because of the amount of merges allowed within BPE. With a lower vocab size BPE can not make many merges resulting in individual letter tokenization. The larger vocab size means more merging of common pairs to form new tokens. This is illustrated in the 4,096 vocab size since common words or endings like "lex" and "er" are split into tokens. With even more merges and given this is English text, "Alex" becomes its own token. The tradeoffs are compute time versus amount of tokens. In LLMS which consume tokens and have a context window the lower amount of tokens would give more room for overall text but at the cost of compute time and resources. When using OpenAI's tokenizer for GPT-4 "Alexander" and " Garcia" (space included as a token in my last name) are the only two tokens. This is even more efficient in terms of quantity and they can achieve this by having substantially larger corpus and also larger vocabulary. Since this is also trained on more than just text including English names it can then keep "Garcia" as one token.

Source on how the BPE works - <https://youtu.be/zduSFxRajkE?si=0MV6tj5A4eV69Zok> "Lets build the GPT tokenizer" by Andrej Karpathy