Tokenization

Tafseer Ahmed

Why?

Tokenization is at the heart of much weirdness of LLMs. Do not brush it off.

- Why can't LLM spell words? **Tokenization**.
- Why can't LLM do super simple string processing tasks like reversing a string? **Tokenization**.
- Why is LLM worse at non-English languages (e.g. Japanese)? **Tokenization**.
- Why is LLM bad at simple arithmetic? **Tokenization**.
- Why did GPT-2 have more than necessary trouble coding in Python? **Tokenization**.
- Why did my LLM abruptly halt when it sees the string "<|endoftext|>"? **Tokenization**.
- What is this weird warning I get about a "trailing whitespace"? **Tokenization**.
- Why the LLM break if I ask it about "SolidGoldMagikarp"? **Tokenization**.
- Why should I prefer to use YAML over JSON with LLMs? **Tokenization**.
- Why is LLM not actually end-to-end language modeling? **Tokenization**.
- What is the real root of suffering? **Tokenization**.

Subword Tokenization

Algorithm	Key Concept	Examples	Transformers Using It
BPE	Iteratively merges most frequent pairs.	["un", "happi", "ness"]	GPT-2, RoBERTa
WordPiece	Maximizes likelihood of subword sequences.	["play", "##ground"]	BERT, ALBERT, DistilBERT
Unigram Model	Probabilistic pruning of subwords.	["friend", "ship"]	T5, XLNet, ByT5
SentencePiece	Works on raw text, handles spaces/punctuation.	["_Hello", ",", "_world", "!"]	T5, mBART

BPE (Byte Pair Encoding)

Start with vocabulary of all individual characters:

= { A, B, C, D, ... a, b, c, d }

Repeat:

- Choose the two symbols that are most frequently adjacent in the training corpus, let's say "A" and "B"
- Add a new merged symbol "AB" to the vocabulary
- Replace every adjacent "A" "B" with "AB" in the corpus

Until *k* merges have been done

BPE (Byte Pair Encoding) - training

fred fed ted bread, and ted fed fred bread

vocabulary = { 'a', 'b', 'd', 'e', 'f', 'n', 'r', 't', ' ' }

BPE (Byte Pair Encoding)

fred fed ted bread, and ted fed fred bread

vocabulary = { 'a', 'b', 'd', 'e', 'f', 'n', 'r', 't', ' ', 'd ' }

1. Choose the two symbols that are most frequently adjacent in the training corpus
2. Add the new merged symbol to the vocabulary
3. Replace every adjacent 'd'+' ' with 'd ' in the corpus

d : 7
ed : 6
re : 4
 f : 3

BPE (Byte Pair Encoding)

fred fed ted bread, and ted fed fred bread

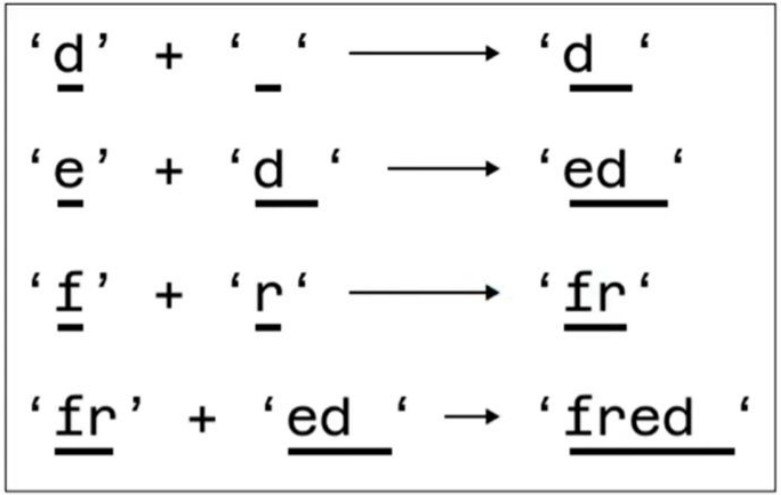
vocabulary = { 'a', 'b', 'd', 'e', 'f', 'n', 'r', 't', ' ', **'d '**, 'ed ' }

1. Choose the two symbols that are most frequently adjacent in the training corpus
2. Add the new merged symbol to the vocabulary
3. Replace every adjacent 'e'+'d ' with 'ed ' in the corpus

ed : 6
re: 4
d f: 4
fr: 3

BPE (Byte Pair Encoding)

4 merges:



vocabulary = { 'a', 'b', 'd', 'e', 'f', 'n', 'r', 't', ' ', 'd ', 'ed ', 'fr', 'fred '}

Online Tokenizer Demo

https://tiktokenizer.vercel.app/