Subject: CS336, Lecture 1, Overview and Tokenization

Date: from July 17, 2025 to July 24, 2025

Contents

Introduction

Youtube：Stanford CS336 Language Modeling from Scratch, Lecture 1: Overview and Tokenization.

本节对语言模型进行了概述，介绍了本课程的理念以及为何需要从头开始构建语言模型. 除此之外，引入了语言模型的基本工具 —— tokenization，介绍了 Character-based、Byte-based、Word-base 三种有缺陷的方法，最后介绍了目前常用的 BPE 方法.

Outline

Outline

Core ideas: *"Understanding via Building."*

What can we learn?

1. **Mechanics.** How things work(transformer, parallelism on GPUs).
2. **Mindset.** Squeeze the most out of the hardware, take scale seriously(scaling law).
3. **Intuitions.** Which data and model decisions yield good accuracy.

注：本课程目的是从头开始构建语言模型(<1B)，而非在课上构建大语言模型，由于大语言模型具有特殊的性质和现象(例如 emergence)，因此这之间还存在一些差距.

- 推荐的 Tokenization 讲解视频：Let's build the GPT Tokenizer – Andrej Karpathy
- 在线 tokenizer 网站；Tiktokenizer

Tokenlization

What is Tokenization?

Tokenization 是将自然原始文本(用 Unicode strings 表示)转化为一组整数的过程，其中每一个整数都代表一个 token，所有可能出现 tokens 的数量被称为 **vocabulary size**. 我们在语言模型中做的事情就是：

$$\text{stings } A \xrightarrow{encode} \text{tokens } B \xrightarrow[reasoning]{process} \text{tokens } C \xrightarrow{decode} \text{strings } D$$

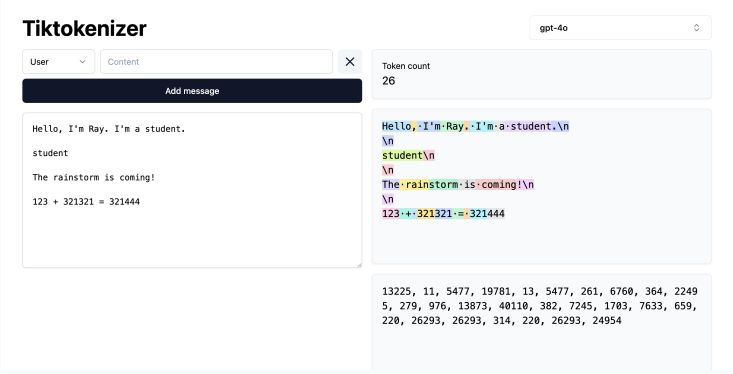

Figure 1: An example for tokenization

根据上面的例子可以观察到一些 tokenization 的现象：

1. 空格也是组成 token 的一部分(例如" student")
2. 同一个词处于句首和句中可能 token 表示不同
3. 一个单词或数字也可以被拆分成多个 token(例如"rainstrom", "123321")
4. 换行也计入 token(即"\n")

在 Tokenization 中一个重要的指标是 **Compression Ratio**(压缩率)，定义为：

$$\frac{\text{bytes number}}{\text{tokens number}} \tag{1}$$

即表示文本需要的字节数(bytes number) 与表示文本所需要的 tokens 数(tokens number).

Note 1. 一般而言使用 *UTF-8* 编码时一个字符*(*字母*/*汉字*)*对应一个或更多 *byte*[a]，因此使用 *bytes* 编码文本的表示空间甚至稍大于文本空间. *Tokenization* 的作用之一就是进行压缩，使用较小的空间表示文本空间，而 *Compression Ratio* 衡量了这一压缩程度.

[a]例如表情符号和中文需要更多 byte 表示，可以在 UTF-8 string length & byte counter s中测试

Several Tokenization methods

Character-based tokenization. 将一个字符(character)对应一个整数进行编码，例如使用 Unicode 编码时，"a" 对应 97. 但实际上很多 character 使用频率很低，然而它们都需要占据 vocabulary 中的一个位置，使得 vocabulary size 很大，故不是对预算的有效利用.

Byte-based tokenization. 直接使用 byte 构建 vocabulary，例如使用 utf-8 编码，字符 "a" 就对应 byte "a"，表情"地球"对应 byte "\xf0\x9f\x8c\x8d". 此时 vocabulary 即 $[0, 255]$[a] . 虽然解决了 Character-based 中部分编码稀疏的问题，但此时 Compression Ratio = 1，注意力机制的计算对于序列长度是二次关系，因此对于长序列计算效率很低.

Word-based tokenization. 对句子以单词为单位进行"切割"，是 NLP 任务中的经典方法. 例如 "I'll say supercalifragilisticexpiadocious." 拆分为 ["I", "'", "ll", "say", "supercalifragilisticexpiadocious"]. 但是这种方法问题在于完整的 vocabulary 会很大，且 vocabulary 一些单词出现的很少，对预算利用效率低.

[a]因为一个 byte 最多只有 256 个取值

Byte Pair Encoding(BPE)

BPE 是一种为数据压缩开发的经典算法(1)，由 Sennrich 等人引入自然语言处理任务中(2). 其基本想法是常见的 sequence 用单个 token 表示，不常见的 sequence 用多个 token 表示. BPE 的基本流程为：

$$\text{string} \xrightarrow{\text{encode}} \text{integral sequence} \to \text{count number pair frequency} \to \text{merge} \to \text{count} \ldots$$

例如对于句子"the cat in the hat"，BPE 的流程如下：

1. 将 string 转化为由 byte 构成的整数序列（即 Byte-based tokenization）：[116, 104, 101, 32, 99, 97, 116, 32, 105, 110, 32, 116, 104, 101, 32, 104, 97, 116]
2. 统计各 integral pair 出现的频次：{"(116, 104)": 2, "(104, 101)": 2, "(101, 32)": 2, "(32, 99)": 1, "(99, 97)": 1, "(97, 116)": 2, "(116, 32)": 1, "(32, 105)": 1, "(105, 110)": 1, "(110, 32)": 1, "(32, 116)": 1, "(32, 104)": 1, "(104, 97)": 1}
3. 将出现最多的组合用一个 byte 表示：(116, 194) $\to$ 256[a]
4. 统计替换后的各 integral pair 出现的频次 $\to$ 替换 $\to$ 循环直至终止条件

[a] 注意 256 已经超过了 [0, 255] 范围.

Last updated: July 24, 2025

References

[1] Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

[2] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.