大模型碎碎念

B站:比飞鸟贵重的多_HKL

LARGE LANGUAGE MODEL

书生

&

Hugging Face

B站:比飞鸟贵重的多_HKL

从Transformers开始读起

https://github.com/InternLM/InternLM

tokenizer_config.json

```
{
  "add_bos_token": true,
  "add_eos_token": false,
  "auto_map": {
    "AutoTokenizer": [
      "tokenization_internlm2.InternLM2Tokenizer",
      "tokenization_internlm2_fast.InternLM2TokenizerFast"
    ]
  },
  "bos_token": "<s>",
  "clean_up_tokenization_spaces": false,
  "decode_with_prefix_space": false,
  "eos_token": "</s>",
  "model_max_length": 1000000000000000019884624838656,
  "pad_token": "</s>",
  "sp_model_kwargs": null,
  "tokenizer_class": "InternLM2Tokenizer",
  "unk_token": "<unk>"
}
```

Bos: beginning of sentence

Eos: end of sentence

对应的python文件

Tokenizer的配置文件，里面包含了InternLM模型对应Tokenizer的类名字以及一些基本信息

从Transformers开始读起

https://github.com/InternLM/InternLM

tokenization_internlm2_fast.py
里面包含两个类：

- InternLM2Converter
- InternLM2TokenizerFast

从这里也可以看出一个问题，为什么hugging face支持这么多开源的算法，核心就两个Tokenizer和model这两个都是模型作者按照HF的规范写的，这样发布的时候只要包含：权重；Tokenizer；Mode；配置就可以发布！

从Transformers开始读起

https://github.com/InternLM/InternLM

tokenize.json

```
{
  "version": "1.0",
  "truncation": null,
  "padding": null,
  "added_tokens": [],
  "normalizer": {
  "pre_tokenizer": null,
  "post_processor": {
  "decoder": {
  "model": {
}
```

截断

```
"normalizer": {
  "type": "Sequence",
  "normalizers": [
    {
      "type": "Replace",
      "pattern": {
        "String": " "
      },
      "content": "▁"
    }
  ]
},
```

```
"model": {
  "type": "BPE",
  "dropout": null,
  "unk_token": "<unk>",
  "continuing_subword_prefix": null,
  "end_of_word_suffix": null,
  "fuse_unk": true,
  "byte_fallback": true,
  "vocab": {
    "<unk>": 0,
    "<s>": 1,
    "</s>": 2,
    "<0x00>": 3,
    "<0x01>": 4,
    "<0x02>": 5,
    "<0x03>": 6,
    "<0x04>": 7,
    "<0x05>": 8,
    "<0x06>": 9,
    "<0x07>": 10,
    "<0x08>": 11,
    "<0x09>": 12,
    "<0x0A>": 13,
    "<0x0B>": 14,
    "<0x0C>": 15,
```

```
  "merges": [
    "<s >",
    "▁ ▁",
    "… …",
    "▁▁ ▁▁",
    "▁ ▁▁▁▁",
    "▁▁▁ ▁▁▁",
    "▁▁▁▁ ▁",
    "i n",
    "▁ t",
    "▁ ▁▁▁▁▁▁▁▁",
    "▁▁▁ ▁▁▁▁▁",
    "▁▁▁▁▁ ▁▁▁▁▁",
    "▁▁▁▁▁ ▁▁▁▁▁",
    "▁▁▁▁▁▁▁▁▁ ▁",
    "▁▁▁▁▁▁▁ ▁▁▁▁",
    "▁▁▁▁▁▁▁▁ ▁▁▁",
    "e r",
    "▁ ▁▁▁",
    "▁▁▁ ▁",
    "o n",
    "▁ a",
    "r e",
    "a t",
    "s t",
    "e n",
    "o r",
    "▁ th",
    "▁t h",
    "\n \n",
    "▁ c",
    "l e",
    "▁ s",
    "i t",
  "vocab": {
  "merges": [
}
```

105
92651
257765

```
"decoder": {
  "type": "Sequence",
  "decoders": [
    {
      "type": "Replace",
      "pattern": {
        "String": "▁"
      },
      "content": " "
    },
    {
      "type": "ByteFallback"
    },
    {
      "type": "Fuse"
    },
    {
      "type": "Strip",
      "content": " ",
      "start": 1,
      "stop": 0
    }
  ]
},
```

```
"post_processor": {
  "type": "TemplateProcessing",
  "single": [
    {
      "SpecialToken": {
        "id": "<s>",
        "type_id": 0
      }
    },
    {
      "Sequence": {
        "id": "A",
        "type_id": 0
      }
    }
  ],
  "pair": [
    {
      "SpecialToken": {
        "id": "<s>",
        "type_id": 0
      }
    },
    {
      "Sequence": {
        "id": "A",
        "type_id": 0
      }
    },
    {
      "SpecialToken": {
        "id": "<s>",
        "type_id": 1
      }
    },
    {
      "Sequence": {
        "id": "B",
        "type_id": 1
      }
    }
  ],
  "special_tokens": {
    "<s>": {
      "id": "<s>",
      "ids": [
        1
      ],
      "tokens": [
        "<s>"
      ]
    }
  },
```

tokenize.model里面似乎存放着一些神秘的内容

从Transformers开始读起

https://github.com/InternLM/InternLM

spical_tokens_map.json

```
{
  "bos_token": "<s>",
  "eos_token": "</s>",
  "pad_token": "</s>",
  "unk_token": "<unk>"
}
```

takenizer_config.json（展开）
+ spical_tokens_map（展开）
+ tokenize.json（路径）
+ tokenize.model（路径）

tokenization_internlm2_fast.py
InternLM2TokenizerFast 类中方

完成了tokenize的构造

从Transformers开始读起

https://github.com/InternLM/InternLM

HKLLLLLL

CausalLM 表示的是什么

ChatGLM

“CausalLM"通常指的是"causal language model”，中文可以翻译为“因果语言模型”。在这种上下文中，“因果”一词并不是指模型具有处理因果关系的能力，而是指这种模型是自回归的，也就是说，它在生成文本时是按顺序一个词一个词地生成的，每个词都依赖于它之前的词。

在自然语言处理中，自回归语言模型（例如GPT-3）是一种能够基于给定的文本上下文预测下一个词或标记的模型。这种模型在生成文本时，会考虑到之前生成的所有词，因此它的输出不仅取决于当前的输入，还取决于它之前生成的所有输出。

这种因果（或自回归）性质使得CausalLM非常适合于生成文本的任务，如自动写作、对话系统、文本补全等。与之相对的是非自回归模型，它们可以同时生成整个序列，而不需要按顺序生成每个词。非自回归模型通常更快，但可能不如自回归模型在生成连贯和自然的文本方面表现得好。

从Transformers开始读起

https://github.com/InternLM/InternLM

config.json

```
{
  "architectures": [
    "InternLM2ForCausalLM"
  ],
  "attn_implementation": "eager",
  "auto_map": {
    "AutoConfig": "configuration_internlm2.InternLM2Config",
    "AutoModelForCausalLM": "modeling_internlm2.InternLM2ForCausalLM",
    "AutoModel": "modeling_internlm2.InternLM2ForCausalLM"
  },
  "bias": false,
  "bos_token_id": 1,
  "eos_token_id": 2,
  "hidden_act": "silu",
  "hidden_size": 2048,
  "initializer_range": 0.02,
  "intermediate_size": 8192,
  "max_position_embeddings": 32768,
  "model_type": "internlm2",
  "num_attention_heads": 16,
  "num_hidden_layers": 24,
  "num_key_value_heads": 8,
  "pad_token_id": 2,
  "rms_norm_eps": 1e-05,
  "rope_scaling": null,
  "rope_theta": 1000000,
  "tie_word_embeddings": false,
  "torch_dtype": "bfloat16",
  "transformers_version": "4.33.0",
  "use_cache": true,
  "vocab_size": 92544,
  "pretraining_tp": 1
}
```

热切的

对应的python文件

generation_config.json

```
{
  "_from_model_config": true,
  "bos_token_id": 1,
  "eos_token_id": 2,
  "pad_token_id": 2,
  "transformers_version": "4.33.0"
}
```

tokenizer

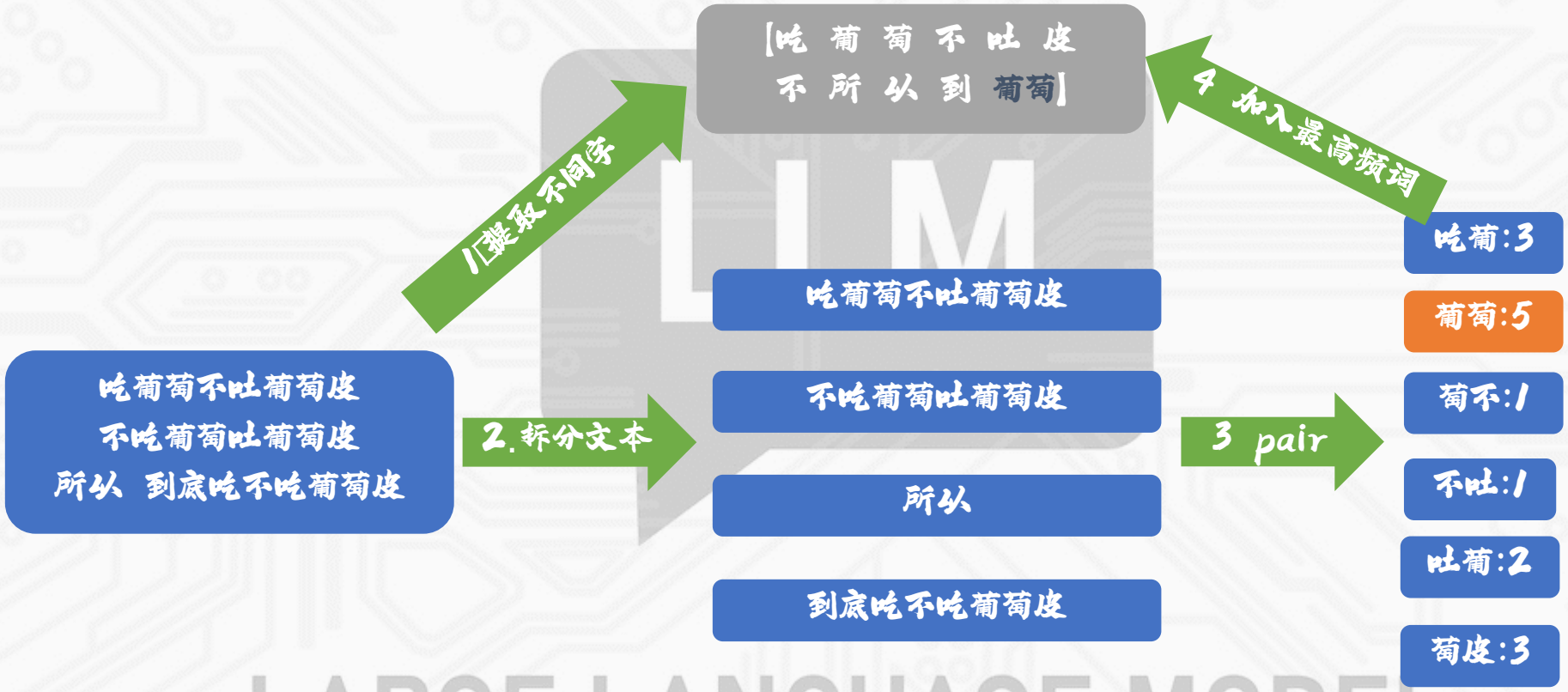
文本转数字 ？

文本高效优雅转数字 ✓

B站:比飞鸟贵重的多_HKL

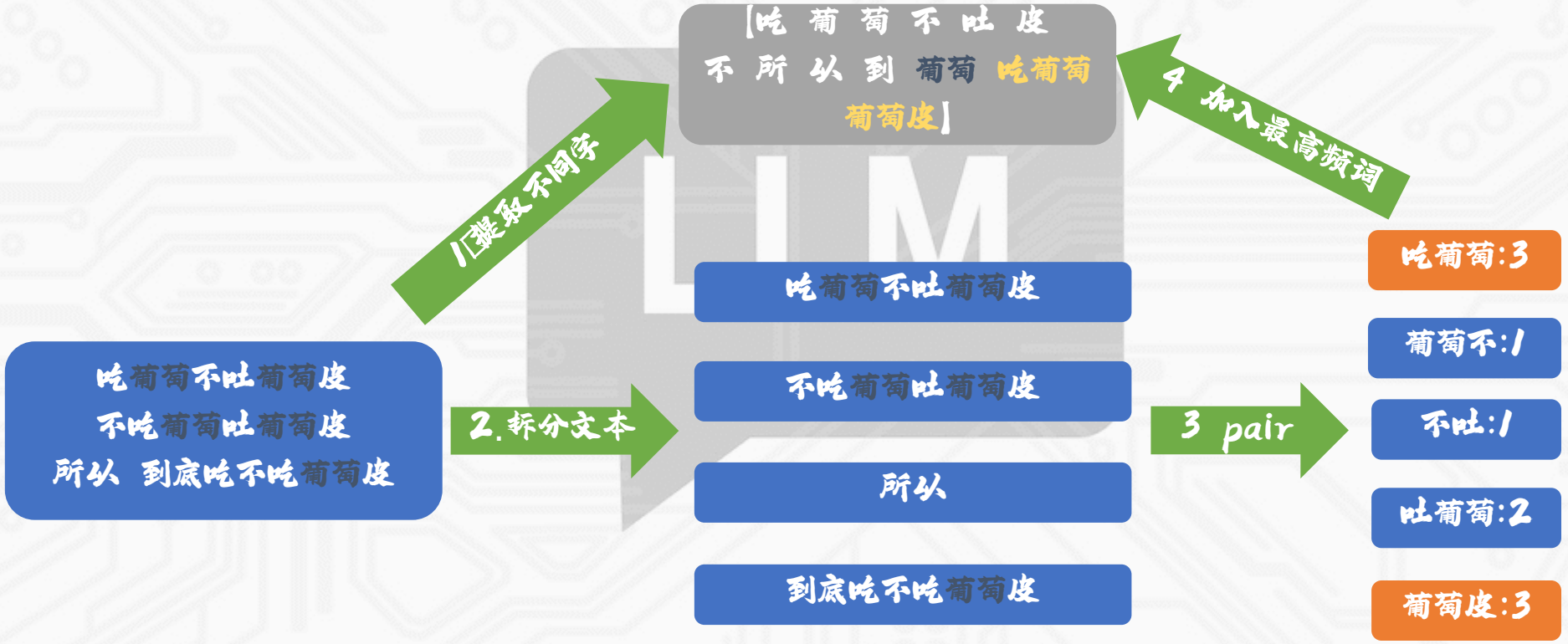
BPE(Byte Pair Encoding)分词算法

大语言模型LLM基础之Tokenizer完全介绍_哔哩哔哩_bilibili



BPE(Byte Pair Encoding)分词算法

大语言模型LLM基础之Tokenizer完全介绍_哔哩哔哩_bilibili



代码实现参考：https://github.com/OctopusMind/BBPE