**Embedding文档**

# 一、创建index

基于不同的模型，不同的文档类型，需要把同一个模型（当前使用包括bge-large-zh-v1.5和m3e，优先m3e），同一种文档类型（例如论文）存储在同一个index中。目前的需求主要涉及2种模型，一种类型文档（期刊）。

## 1.1 bge-large-zh-v1.5

创建index，需要指定body对应的格式，基本格式如下：

request\_body\_mpnet ={

"settings": { "number\_of\_shards": 4, "number\_of\_replicas": 1 },

"mappings": { "properties": {

"text": {"type": "text"},

"vector": {"type": "dense\_vector", "dims": 1024, 'index': True, 'similarity': 'cosine'}

}

}

}

创建index名称是paperbge的index。

es\_client.indices.create(index = "paperbge" , body = request\_body\_mpnet)

## 1.2 m3e-base

创建index，需要指定body对应的格式，基本格式如下：

request\_body\_mpnet ={

"settings": { "number\_of\_shards": 4, "number\_of\_replicas": 1 },

"mappings": { "properties": {

"text": {"type": "text"},

"vector": {"type": "dense\_vector", "dims": 768, 'index': True, 'similarity': 'cosine'}, # dimensionality of m3e-base 是768

}

}

}

创建index名称是paperbge的index。

es\_client.indices.create(index = "paperm3e" , body = request\_body\_mpnet)

# 二、划分chunk大小

文章内容已经根据段落进行切分，为适应模型embedding对token限定的大小（一般是512个token），需要对段落进行进一步切分。

同一篇文章，需要有字段，标明**文本及表格**在整篇文章中的顺序（例如sequence）。例如文章开始段落后紧接一个表格，段落sequence是0，表格sequence是1。

## 2.1文本切分

对每篇文章的内容，按段落（**含摘要**），最大切分长度max\_len = 400（需要预留一些给header），chunk\_overlap = 50进行文本切分。样例如下。

max\_len = 400

text\_splitter = CharacterTextSplitter(

separator = "\n",

chunk\_size = max\_len,

chunk\_overlap = 50,

length\_function = len,

)

相同段落切分成n个chunk时，chunk需要增加一个字段“part”，从0开始递增：0，1，2，3…，用来表示chunk在原始段落中的顺序。

**注意：不同段落之间，不要合并后切分。而是按照逐个sequence切分。**

## 2.2 表格切分

表格从原始html，转化成文本形式。需要留意当表格是merge的，需要设置自动填充。

Python的pandas读取html时，可以实现自动填充。(遇到不能转化的，pass)

```Python

import pandas as pd

pd.read\_html(format\_html)

```

样例如下：

|  |  |
| --- | --- |
| 岩相 | 颗粒大小 |
| RT | 1 |
| 2 |
| 3 |

转化为文本后：

“””

|岩相|颗粒大小|\n

|RT|1|\n

|RT|2|\n

|RT|3|

“””

对每个表格，按表格，最大切分长度max\_len = 400（需要预留一些给header），chunk\_overlap = 50进行文本切分。**但要求同一行表格，在同一个切分内**。

相同表格切分成n个chunk时，chunk需要增加一个字段“part”，从0开始递增：0，1，2，3…，用来表示chunk在原始表格中的顺序。

## 2.3 未抓到html版本的论文切分(当前没有这部分)

对文章整体，按最大切分长度max\_len = 400（需要预留一些给header），chunk\_overlap = 50进行文本切分。

这时候的metadata包含为内容（其他meta详见3.1）：

“filename”：文章标题

“author”：文章作者

“doi”：文章doi

“keywords”：关键词

“publisher”：出版社

“journal”：期刊

“organization”：论文发布的机构

“sequence”：chunk中正文的位置，和part一致

“part”：chunk中正文的位置，和sequence一致

“header1”：空

“header2”：空

“header3”：空

“type”：等于0，代表未识别html的区分类型

# 三、逐个chunk存入es数据库

基于各个index，逐个chunk，存入\_source如下内容

## 3.1. metadata

metadata包含以下内容：

“filename”：文章标题

“author”：文章作者

“doi”：文章doi

“keywords”：关键词

“publisher”：出版社

“journal”：期刊

“organization”：论文发布的机构

“sequence”：段落或表格在文本中的顺序

“part”：chunk中段落或表格中的位置

“header1”：段落的标题title1。表格同理。摘要是空。

“header2”：段落的标题title2。表格同理。摘要是空。

“header3”：段落的标题title3。表格同理。摘要是空。

“type”：用1，2，3区分：正文，表格，摘要。

### 3.1.1 表格需求（以embeding优先，往后）

**沉积岩的重量**

*英文XXX*

单位：千克

|A | B|

|---|---|

|1|2|

上述表格说明XXX

需求：抓取表上的前3行，分别存在title1,title2,title3。或者直接抓取标题。

## 3.2. text

表格或者段落切分成chunk后的文本信息。

## 3.2. vector

逐个chunk的文本信息，embedding之后，存入vector中。

其中文本需要添加**前置文本信息**再进行embedding：

“以下信息来源于科学论文{ filename }。论文发表在{ publisher}({journal}) 上。”

其中表格需要添加**前置文本信息**再进行embedding：

“以下信息来源于科学论文{ filename }。论文发表在{ publisher}({journal}) 上。”