

Embedding 报告

姓名：颜舒逸

学号：523031910677

源代码和LOG探究

1. 加载 `model/` 文件夹下的模型，初始化相关参数。

2. 调用 `basic()` 函数，测试单词之间的相似度。

- 用两个嵌入向量之间的余弦相似度来表示不同单词之间的相似性。
- 以下这些测试包含的语义已注释在相应的语句上。

```
// 如果单词 Apple 与 Banana 的相似度大于 Apple 与 Man, 则通过的测试数+1
if (sim_matrix["Apple"]["Banana"] > sim_matrix["Apple"]["Man"]) {
    passed_count++;
}

// 如果单词 Apple 与 Orange 的相似度大于 Apple 与 Chicken, 则通过的测试数+1
if (sim_matrix["Apple"]["Orange"] > sim_matrix["Apple"]["Chicken"]) {
    passed_count++;
}

// 如果单词 Banana 与 Orange 的相似度大于 Banana 与 Man, 则通过的测试数+1
if (sim_matrix["Banana"]["Orange"] > sim_matrix["Banana"]["Man"]) {
    passed_count++;
}
```

- 耗时 24.28ms

3. 调用 `large()` 函数，测试大文本。

- 对问题 `What did Ward do and what did he won in Michigan High School?` 进行语义搜索。
- 以下测试所包含的语义已注释在语句上。

```
// 如果搜索到的文本与预期不符，则报错；否则通过的测试数+1
if (max_sim_sentence != supposed_sentence) {
    std::cerr << "Failed to find the correct sentence" << std::endl;
    return passed_count;
}
passed_count++;
```

- 耗时 55680.01ms