GAME CARO

BẰNG NGÔN NGỮ LẬP TRÌNH PYTHON

Tên nhóm: **RESTUDY**

STT MSSV Họ và tên Email

1 18127158 Lê Thành Nam 18127158@student.hcmus.edu.vn

2 18127121 Nguyễn Đăng Khoa 18127121@student.hcmus.edu.vn

3 18127138 Nguyễn Duy Long 18127138@student.hcmus.edu.vn

4 18127107 Lý Đăng Huy 18127107@student.hcmus.edu.vn

5 18127134 Lê Huỳnh Long 18127134@student.hcmus.edu.vn

**DANH SÁCH CHỨC NĂNG**

1. **Tạo kho dữ liệu :** Nội dung trong tất cả các files sẽ được lưu dưới dạng dictionary với key là tên file, value là nội dung của file được lưu dưới dạng list.

Ví dụ: {'file' : ['hello', 'world']}

1. **Term Frequency: Tần suất xuất hiện của một từ.**

Lưu thông số dưới dạng Dictionary với key là tên file của file đang tính, value là độ TF của từ trong input.

TF = số lần xuất hiện của 1 từ trong 1 file / số từ của 1 file

TF *=* File\_Dicts[file\_name].count(word)/(len(File\_Dicts[file\_name]))

Ví dụ: { file: ["hello": 0.1, "world": 0,2]}

1. **Inverse Document Frequency: Tần số nghịch của một từ trong một kho dữ liệu**

IDF = log(tổng số file / 1 + số file chứa từ )

IDF = math.log(length/float((amount + 1)))

1. **TF \* IDF: Trích lọc ra những từ phổ biến bằng cách nhân TF và IDF với nhau để giảm hệ số.**
2. **ĐỘ TƯƠNG ĐỒNG:** Tổng của tích TF \* IDF của từng files với TF \* IDF của input .

Dùng thuật toán Cosine Similarty để tính độ tương đồng.

**Cosine (a, b) = tích vô hướng(a , b) / (tích độ dài (a) \* tích độ dài (b))**

* 1. **Tính tích độ dài:** Tích của căn bậc 2 của tổng bình phương tf\_idf của từng file với căn bậc 2 của tổng bình phương tf\_idf của input.

Tích độ dài = sqrt(TF\*IDF của file) \* math.sqrt(TF\*IDF của input)

* 1. **Tính tích vô hướng:** Tổng của tích TF\*IDF của từng file với TF\*IDF của input.

Tích vô hướng: sum ( TF\*IDF của input \* TF\*IDF của file)

Từ đó ra được độ tương đồng của từng file đối với input đã nhập

1. **Show\_content:** Để hiện thị kết quả tìm kiếm và thời gian tìm kiếm.

**CÔNG VIỆC CỦA TỪNG CÁ NHÂN**

**Lê Huỳnh Long, Nguyễn Duy Long:**

* Viết hàm tạo dữ liệu từ từng files.
* Viết hàm tính TF và tính IDF.
* Viết hàm nhập xuất dữ liệu.

**Lê Thành Nam, Lý Đăng Huy:**

* Thiết kế giao diện.
* Viết các hàm để scroll bằng cả chuột.
* VIết hàm hiện màn hình kết quả tìm kiếm, thời gian tìm kiếm.

**Nguyễn Đăng Khoa:**

* Lên ý tưởng, phân công công việc.
* Viết hàm tính độ tương đồng bằng thuật toán Cosine.
* Tổng hợp các phần của nhóm.
* Sửa lỗi, tối ưu thuật toán.

**Đánh giá chung:**

* + Tất cả thành viên trong nhóm đều hoàn thành tốt công việc được giao.
  + Tất cả thành viên đều đóng góp ý tưởng để cải thiện đến mức tốt nhất.
  + Tuy có nhiều khó khăn nhưng vẫn tìm được hướng giải quyết.
  + Có mâu thuẫn, bất đồng quan điểm nhưng đã thống nhất lại.
  + Sử dụng kiểu Dictionary thay cho kiểu List giúp cho phần mềm tìm kiếm nhanh hơn 4 đến 6 giây.
  + Kho dữ liệu lên đến 20 nghìn files.
  + Lần cuối thử nghiệm chương trình chạy 12 giây với key gồm 8 từ.

**BẢNG TỔNG KẾT CÁ NHÂN**

|  |  |
| --- | --- |
| **Tên** | **Mức độ hoàn thiện** |
| Nguyễn Đăng Khoa | 100% |
| Lê Thành Nam | 98% |
| Nguyễn Duy Long | 95% |
| Lê Huỳnh Long | 95% |
| Lý Đăng Huy | 95% |