KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

**HỌC KỲ I, NĂM HỌC 2023 – 2024**

**TÌM HIỂU VÀ CÀI ĐẶT THUẬT TOÁN ĐÁNH GIÁ SỰ TƯƠNG ĐỒNG CỦA VĂN BẢN SỬ DỤNG ĐỘ ĐO JACCARD VÀ LEVENSHTEIN**

*Sinh viên thực hiện:*

Họ tên: **Nguyễn Anh Tuấn**

MSSV: **110121123**

Lớp: **DA21TTA**

*Giáo viên hướng dẫn:*

**Nguyễn Nhứt Lam**

***Trà Vinh, ngày 8 tháng 1 năm 2024***

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

**HỌC KỲ I, NĂM HỌC 2023 – 2024**

**TÌM HIỂU VÀ CÀI ĐẶT THUẬT TOÁN ĐÁNH GIÁ SỰ TƯƠNG ĐỒNG CỦA VĂN BẢN SỬ DỤNG ĐỘ ĐO JACCARD VÀ LEVENSHTEIN**

*Sinh viên thực hiện:*

Họ tên: **Nguyễn Anh Tuấn**

MSSV: **110121123**

Lớp: **DA21TTA**

*Giáo viên hướng dẫn:***Nguyễn Nhứt Lam**

***Trà Vinh, ngày 8 tháng 1 năm 2024***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời nói đầu tiên cho tôi xin gửi lời cảm ơn và biết ơn sâu sắc đến Thầy Nguyễn Nhứt Lam đã hướng dẫn và chỉ bảo tận tình cho tôi trong suốt quá trình nghiên cứu và quá trình thực hiện đồ án cơ sở này.

Tôi xin chân thành cảm ơn các thầy, cô đã tạo cho tôi những điều kiện thuận lợi để học tập và nghiên cứu tại trường Đại Học Trà Vinh.

Tôi cũng xin gửi lời cảm ơn đến các bạn sinh viên trong nhóm nghiên cứu đồ án trong nghành Công Nghệ Thông Tin đã giúp đỡ, ủng hộ và động viên tôi trong quá trình nghiên cứu và thực hiện đồ án.

Cuối cùng, tôi muốn gửi lời cảm ơn và biết ơn đến bố, mẹ, tất cả bạn bè và những người thân yêu của tôi.

*Xin chân thành cảm ơn!*

Sinh viên

Nguyễn Anh Tuấn

**MỤC LỤC**

ㅤ

[**MỞ ĐẦU** 1](#_Toc155516481)

[Lý do chọn đề tài 1](#_Toc155516482)

[Mục đích 1](#_Toc155516483)

[Đối tượng 2](#_Toc155516484)

[Phạm vi nghiên cứu 2](#_Toc155516485)

[**CHƯƠNG 1: TỔNG QUAN** 4](#_Toc155516486)

[**1.1. MÔ TẢ VẤN ĐỀ** 4](#_Toc155516487)

[1.1.1. Sự tương đồng trong văn bản 4](#_Toc155516488)

[1.1.2. Thuật toán đánh giá sự tương đồng 4](#_Toc155516489)

[1.1.3. Kết quả của sự tương đồng 4](#_Toc155516490)

[**1.2. MỤC TIÊU NGHIÊN CỨU** 5](#_Toc155516491)

[1.2.1. Tìm hiểu về của hai thuật toán **Jaccard** & **Levenshtein** 5](#_Toc155516492)

[1.2.2. Tìm hiểu về ngôn ngữ **Python** 6](#_Toc155516493)

[1.2.3. Tìm hiểu về những văn bản đầu vào 6](#_Toc155516494)

[1.2.4. Theo sát quá trình so khớp diễn ra như thế nào 6](#_Toc155516495)

[**1.3. HƯỚNG TIẾP CẬN** 6](#_Toc155516496)

[1.3.1. Sau khi tìm hiểu thuật toán và ngôn ngữ lập trình 6](#_Toc155516497)

[1.3.2. Tối ưu quá trình thực hiện 6](#_Toc155516498)

[1.3.3. Khắc phục những lỗi sai nếu có 6](#_Toc155516499)

[**1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU** 6](#_Toc155516500)

[1.4.1. Người dùng cá nhân 6](#_Toc155516501)

[1.4.2. Tổ chức 7](#_Toc155516502)

[1.4.3. Doanh nghiệp 7](#_Toc155516503)

[1.4.4. Nhà nghiên cứu xử lý ngôn ngữ tự nhiên NLP 7](#_Toc155516504)

[1.4.5. Phạm vi nghiên cứu 7](#_Toc155516505)

[**CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT** 9](#_Toc155516506)

[**2.1. LÝ THUYẾT** 9](#_Toc155516507)

[2.1.1. Giới thiệu 9](#_Toc155516508)

[2.1.2. Độ đo **Jaccard** 11](#_Toc155516509)

[2.1.3. Độ đo **Levenshtein** 13](#_Toc155516510)

[**2.2. LÝ LUẬN** 15](#_Toc155516511)

[2.2.1. Sự quan trọng của tương đồng văn bản 15](#_Toc155516512)

[2.2.2. Tính ứng dụng rộng rãi 16](#_Toc155516513)

[**2.3. GIẢ THUYẾT KHOA HỌC** 19](#_Toc155516514)

[2.3.1. Giả thuyết chính 19](#_Toc155516515)

[2.3.2. Giả thuyết phụ 20](#_Toc155516516)

[2.3.3. Dự đoán lợi ích về mọi mặt 21](#_Toc155516517)

[**2.4. PHƯƠNG PHÁP NGHIÊN CỨU** 21](#_Toc155516518)

[2.4.1. Xây dựng thuật toán 21](#_Toc155516519)

[2.4.2. Tối ưu hóa và đánh giá 23](#_Toc155516520)

[2.4.3. So sánh và phân tích 24](#_Toc155516521)

[2.4.4. Ứng dụng thực tế 25](#_Toc155516522)

[**CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU** 27](#_Toc155516523)

[**3.1. CHUẨN BỊ NỀN TẢNG NGHIÊN CỨU** 27](#_Toc155516524)

[3.1.1. Tìm hiểu cơ sở lý thuyết 27](#_Toc155516525)

[3.1.2. Xây dựng nền tảng lý luận 29](#_Toc155516526)

[**3.2. THIẾT KẾ THUẬT TOÁN** 29](#_Toc155516527)

[3.2.1. Xây dựng thuật toán 29](#_Toc155516528)

[3.2.2. Tối ưu hóa và điều chỉnh 30](#_Toc155516529)

[**3.3. HIỆN THỰC HÓA THUẬT TOÁN** 31](#_Toc155516530)

[3.3.1. Ngôn ngữ lập trình và môi trường 31](#_Toc155516531)

[3.3.2. Cài đặt thuật toán 32](#_Toc155516532)

[**3.4. ĐÁNH GIÁ VÀ THỬ NGHIỆM** 32](#_Toc155516533)

[3.4.1. Chuẩn bị dữ liệu kiểm thử 32](#_Toc155516534)

[3.4.2. Thực hiện thử nghiệm 32](#_Toc155516535)

[**3.5. SO SÁNH VÀ PHÂN TÍCH KẾT QUẢ** 33](#_Toc155516536)

[3.5.1. So sánh kết quả 33](#_Toc155516537)

[3.5.2. Phân tích hiệu suất 33](#_Toc155516538)

[**3.6. KẾT LUẬN VÀ TÍNH ỨNG DỤNG** 33](#_Toc155516539)

[3.6.1. Rút kinh nghiệm và kết luận 33](#_Toc155516540)

[3.6.2. Tính ứng dụng và khả năng mở rộng 34](#_Toc155516541)

[**CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU** 35](#_Toc155516542)

[**4.1. HIỆU NĂNG** 35](#_Toc155516543)

[4.1.1. Hiệu suất đạt được 35](#_Toc155516544)

[4.2.2. Độ chính xác 35](#_Toc155516545)

[**4.2. CÀI ĐẶT THUẬT TOÁN** 36](#_Toc155516546)

[4.2.1. Thuật toán **Jaccard** 36](#_Toc155516547)

[4.2.2. Thuật toán **Levenshtein** 37](#_Toc155516548)

[**4.3. TRẢI NGHIỆM NGƯỜI DÙNG** 39](#_Toc155516549)

[4.3.1. Giao diện sử dụng 39](#_Toc155516550)

[**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 43](#_Toc155516551)

[**5.1. KẾT QUẢ** 43](#_Toc155516552)

[**5.2. HƯỚNG PHÁT TRIỂN** 44](#_Toc155516553)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 45](#_Toc155516554)

[**PHỤ LỤC** 45](#_Toc155516555)

**DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

**Hình 1.** Ví dụ về độ đo Jaccard

**Hình 2.** Ví dụ về độ đo Levenshtein

**Hình 3.** Giải thích từ viết tắt NLP

**Hình 4.** Mô hình tính độ tương đồng văn bản

**Hình 5.** Quá trình xử lý chuyên sâu

**Hình 6.** Các bước tính độ tương đồng văn bản sử dụng độ đo Jaccard

**Hình 7.** Các bước tính độ tương đồng văn bản sử dụng độ đo Levenshtien

**Hình 8.** Thể hiện các mô hình trong lĩnh vực y tế

**Hình 9.** Văn bản pháp luật

**Hình 10.** Chuyển đổi số trong lĩnh vực giáo dục

**Hình 11.** Sự nhận xét góp phần phát triển

**TÓM TẮT NIÊN LUẬN/ĐỒ ÁN CƠ SỞ NGÀNH**

**Những nghiên cứu về hai thuật toán cần sử dụng:**

* Tìm hiểu và nghiên cứu về hai thuật toán cần sử dụng là **Jaccard** & **Levenshtein**.
* Tìm hiểu về tính năng của thuật và cách sử dụng độ đo của hai thuật toán.
* Phân tích và đánh giá sự tương đồng giữa các đoạn văn bản sử dụng hai độ đo chính: **Jaccard** & **Levenshtein**.
* Kiểm tra thật kĩ các vấn đề về đo đường của hai thuật toán sử dụng cho mục đích đo lường độ tương đồng văn bản.
* Xem nhiều bài viết trên Internet, tạp chí nói về tính chất của hai thuật toán để có thể biết được hai thuật toán hoạt động ra sao và đo lường như thế nào.
* Tìm hiểu và chọn lọc những cách phù hợp để biểu diễn thuộc tính của hai thuật toán.
* Xem thêm về các cách biểu diễn thuật trên nhiều nền tảng khác nhau để tìm ra cách phù hợp để trình bày thuật toán.

**Hướng tiếp cận vấn đề:**

* Sau khi đã tìm hiểu và nghiên cứu ở phần đầu tiên thì bước kế tiếp sẽ đi sâu về cách biểu diễn hai thuật toán bằng ngôn ngữ lập trình phù hợp.
* Tìm hiểu rõ về ngôn ngữ để biểu diễn hai thuật toán sao cho thể hiện được sự so khớp văn bản là điều cần thiết.
* Tiếp theo thì ta tiến vào trọng tâm của thuật toán đầu tiên chính là **Jaccard:** theo sự tìm hiểu và nghiên cứu cho thấy thuật toán **Jaccard** tính độ tương đồng văn bản bằng cách tính số phần tử chung của hai văn bản chia cho tổng số phần tử của hai văn bản sẽ được kết quả độ tương đồng văn bản bao nhiêu phần trăm.
* Và thuật toán thứ hai là **Levenshtein:** cũng theo tìm hiểu và nghiên cứu cho biết được thuật toán sẽ đánh giá sự tương đồng thông qua các thao tác chỉnh sửa như: “Thêm, Xóa, Sửa” các từ hoặc các kí tự trong văn bản để đưa ra kết quả độ tương đồng văn bản theo phần trăm.
* Bước cuối để tiếp cận chúng ta sẽ tiến hành vào cài đặt hai thuật toán để cho ra được kết quả so khớp với hai tính chất đã nêu ở trên theo mỗi thuật toán.

**Cách giải quyết vấn đề:**

* Phân tích và so sánh ưu điểm và nhược điểm của cả hai độ đo **Jaccard** & **Levenshtein**.
* Hiểu rõ cơ sở lý thuyết và công thức tính toán cho mỗi độ đo.
* Xây dựng thuật toán bằng ngôn ngữ lập trình đã chọn để tạo và thực thi chương trình tính độ tương đồng văn bản.
* Tối ưu hóa để tập trung vào vấn đề đánh giá sự tương đồng văn bản.
* Hiển thị kết quả đạt được lên một giao diện sao cho người dùng nhìn vào thì thấy được kết quả đánh giá một cách rõ ràng và đẹp mắt.

**Kết quả đạt được sau khi nghiên cứu và triển khai:**

* Cài đặt được chương trình tính độ tương đồng văn bản sử dụng hai độ đo **Jaccard** & **Levenshtein.**
* Thể hiện được sự giống và khác nhau trong quá trình so khớp văn bản để người dùng có thể chọn độ đo phù hợp và sử dụng cho mục đích của họ.
* Ngoài việc so khớp văn bản hai độ đo còn có thể thực hiện một số tính năng khác như: Tìm kiếm văn bản, phân loại văn bản, trích xuất thông tin, tóm tắt thông tin, kiểm tra độ tin cậy của văn bản,…

# MỞ ĐẦU

## Lý do chọn đề tài

* Đề tài nghiên cứu của tôi là “**Tìm hiểu và cài đặt thuật toán đánh giá sự tương đồng văn bản sửa dụng độ đo Jaccard** & **Levenshtein”,** tôi chọn đề tài nghiên cứu này bởi vì tôi thấy nó rất bổ ích trong việc kiểm tra bản quyền văn bản bị lạm dụng sao chép trái phép.
* Việc đánh giá này góp phần giúp ít cho mọi người có thêm tính sáng tạo khi soạn thảo văn bản tránh việc sao chép văn bản của người khác sẽ dẫn đến tình trạng lười suy nghĩ.
* Ngoài những ý kiến trên mục đích tôi chọn đề tài dành cho việc hiểu biết thêm về ngôn ngữ lập trình Python và biết được văn bản cũng có thể so sánh thông qua việc tính toán.

## Mục đích

* **Hướng tới việc so khớp văn bản:** Mục tiêu của nghiên cứu này là tìm hiểu phát triển sự hiểu biết trọng tâm về cách so sánh văn bản và đánh giá sự tương đồng thông qua hai độ đo chính: Jaccard và Levenshtein.
* **Thể hiện sự quan trọng của sự tương đồng văn bản:** Có những nguồn văn bản bị lạm dụng sao chép một cách tùy tiện khiến cho việc đánh bản quyền ngày càng phức tạp và không hiệu quả.
* **Ứng dụng trực tiếp vào NLP:** Hiểu rõ cách thức tính toán sự tương đồng giữa văn bản có thể hỗ trợ trong việc phát triển các ứng dụng NLP hiệu quả, đồng thời giảm chi phí xử lý bằng cách chọn lọc các đoạn văn bản có độ tương đồng cao.
* **Tiếp thu thêm kiến thức về việc chuyển đổi văn bản:** Chuyển đổi văn bản thành dạng số, là hình thức chuyển đổi giúp cho ngôn ngữ máy dể hiểu và dễ áp dụng tính toán.
* **Tiếp thu thêm kiến thức về biễu diễn văn bản:** Biễu diễn văn bản thành dạng tập hợp riêng biệt để ngôn ngữ máy xử lí gọn hơn nếu như không biểu diễn thành những tập hợp riêng lẻ.

## Đối tượng

* **Hướng đến tất cả mọi người:** Chương trình nghiên cứu của tôi không hạn chế đối tượng người dùng, mà hướng đến tất cả mọi người ai cũng có thể sử dụng, một phần nào đó giúp cho chương trình nghiên cứu này phát triển rộng hơn và phổ biến hơn nữa, nhiều người sử dụng thì có thêm nhiều góp ý góp phần cải thiện chương trình nghiên cứu này ngày cần tốt hơn.
* **Các nhà nghiên cứu xử lý ngôn ngữ tự nhiên:** Các nhà nghiên cứu chuyên trong lĩnh vực xử lý ngôn ngữ tự nhiên sử dụng các thuật toán này sử dụng trong những chương trình của họ để triển khai.
* **Người dùng áp dụng xử lý ngôn ngữ tự nhiên vào thực tế:** Người sử dụng muốn thông qua việc xử lý ngôn ngữ tự nhiên để sử dụng trong một số mục đích riêng của họ như: tìm kiếm thông tin, phân loại văn bản, rút gọn văn bản, chia nhỏ văn bản thành những đoạn văn bản nhỏ,….
* **Các nhà an ninh, bảo mật thông tin:** Trong lĩnh vực an ninh này, việc so khớp văn bản để kiểm tra xem văn bản có đạo nhái hay không, không rõ nguồn gốc, sao chép ý tưởng,... Sau những việc kiểm tra này giúp cho các nhà bảo mật thông tin bảo vệ được những tập văn bản một cách an toàn nhất có thể và đem đến nguồn văn bản an toàn cho người dùng.
* **Doanh nghiệp:** Giúp cho doanh nghiệp kiểm tra tài liệu của doanh nghiệp có bị sao chép hay không, kiểm tra các báo cáo của nhân viên và những hợp đồng quan trọng.
* **Thương mại:** Áp dụng việc xử lý ngôn ngữ để cải thiện chất lượng của hệ thống và tích hợp vào sản phẩm và dịch vụ của họ để thuận lời hơn trong việc trao đổi sản phẩm thương mại.

## Phạm vi nghiên cứu

* **Độ đo Jaccard** & **Levenshtein:** Phạm vi nghiên cứu tập trung vào hai độ đo chính: **Jaccard** & **Levenshtein**. Thông qua việc tập trung vào cả hai độ đo này, nghiên cứu muốn đảm bảo sự đa dạng và toàn diện trong việc đánh giá sự tương đồng văn bản.
* **Ứng dụng trên một loạt các tập dữ liệu thô:** Phạm vi cũng mở rộng đến việc áp dụng thuật toán và đánh giá kết quả trên một loạt các tập dữ liệu kiểm thử để đảm bảo tính ứng dụng rộng rãi của phương pháp nghiên cứu.

# CHƯƠNG 1: TỔNG QUAN

## 1.1. MÔ TẢ VẤN ĐỀ

### 1.1.1. Sự tương đồng trong văn bản

Không riêng về những quá trình nghiên cứu lớn của các nhà khoa học hoặc một dự án lớn của một tổ chức hay cá nhân nào đó mà sự tương đồng văn bản cũng đóng vai trò quan trọng trong việc so khớp xem văn bản có bị sao chép hay không. Và để hiểu rõ sự tương đồng này thì chúng ta cần có một số phương thức để kiểm tra và cho ra kết quả.

### 1.1.2. Thuật toán đánh giá sự tương đồng

Trong xã hội hiện nay đã có rất nhiều công trình nghiên cứu về các loại phương thức để thực hiện việc kiểm tra sự tương đồng văn bản, và trong nghiên cứu này của tôi sẽ trình bày, thực hiện và đưa vào sử dụng hai phương thức để tính và thể hiện kết quả của sự tương đồng văn bản mà bất kỳ ai cũng có thể sử dụng cho nhiều mục đích khác nhau.

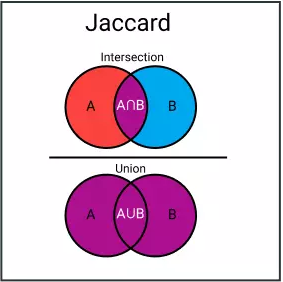
### 1.1.3. Kết quả của sự tương đồng

Công cụ tính toán mà tôi nghiên cứu trong quá trình này sẽ thể hiện kết quả như: tỉ lệ giống nhau của hai tập văn bản cần so khớp là bao nhiêu phần trăm, tổng số từ giống nhau của hai tập văn bản, tổng số từ khác nhau và cuối cùng là tính năng cho phép người dùng kiểm tra số lần xuất hiện của một từ nào đó có tần suất xuất hiện trong hai tạp văn bản là bao nhiêu lần.

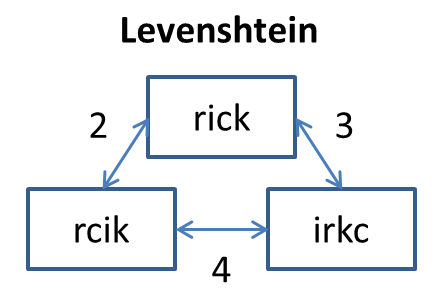
## 1.2. MỤC TIÊU NGHIÊN CỨU

### 1.2.1. Tìm hiểu về độ đo của hai thuật toán Jaccard & Levenshtein

Tìm hiểu xem hai thuật toán có những ưu và nhược điểm gì và cách thức hoạt động của hai thuật toán.



**Hình 1.** Ví dụ về độ đo Jaccard



**Hình 2.** Ví dụ về độ đo Levenshtein

### 1.2.2. Tìm hiểu về ngôn ngữ Python

Nghiên cứu những thư viện và cách để viết chương trình thể hiện được tính năng của hai thuật toán cần sử dụng. Những thư viện cần dùng và những lệnh cần thiết để viết chương trình.

### 1.2.3. Tìm hiểu về những văn bản đầu vào

Tìm hiểu xem có loại văn bản nào không thể đưa vào để chương trình tính toán hay không và nếu loại văn bản nào cũng đưa vào được thì có qua xử lý để mô hình ngôn ngữ máy có thể hiểu được hay không.

### 1.2.4. Theo sát quá trình so khớp diễn ra như thế nào

Phân tích xem quá trình thực hiên của thuật toán có phức tạp hay không có thể tính ra kết quả đạt yêu cầu hay không.

## 1.3. HƯỚNG TIẾP CẬN

### 1.3.1. Sau khi tìm hiểu thuật toán và ngôn ngữ lập trình

Tiến hành viết chương trình hoạt động hai thuật toán để thực hiện quá trình so sánh hai tập văn bản.

### 1.3.2. Tối ưu quá trình thực hiện

Diễn đạt quá trình một cách rõ ràng và mạch lạc cho người dùng dễ quan sát kết quả mà họ mong muốn.

### 1.3.3. Khắc phục những lỗi sai nếu có

Kiểm tra lại chương trình xem những công thức hay đoạn lệnh nào có bị lỗi dẫn đến kết quả sai.

## 1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

### 1.4.1. Người dùng cá nhân

Những người dùng cá nhân không phải một tổ chức hay doanh nghiệp nào đó thì họ cũng có nhu cầu cần sử dụng để so khớp dữ liệu của cá nhân họ.

**1.4.2. Tổ chức**

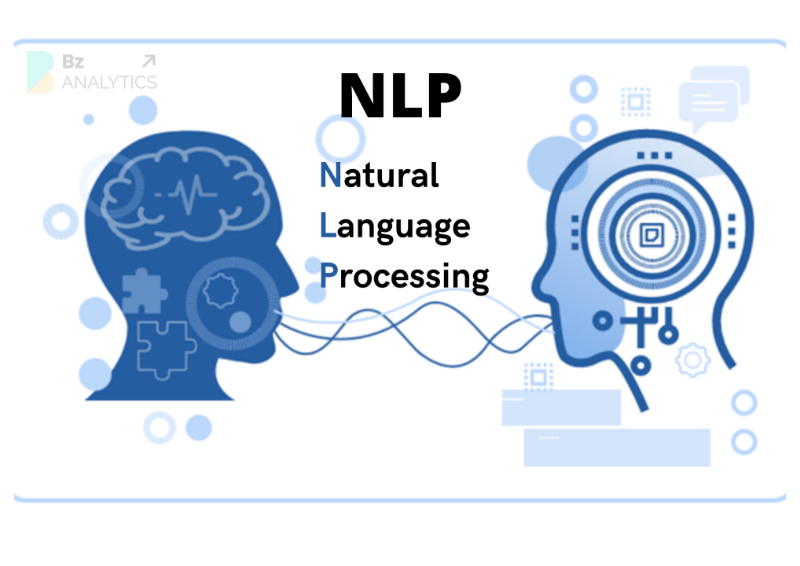
Bao gồm một tập hợp người sử dụng cho nhiều mục đích không chỉ riêng tính độ tương đồng văn bản, như phân tích một tập dữ liệu lớn cho một tổ chức có thể triển khai cho dự án của họ.

**1.4.3. Doanh nghiệp**

Giải quyết vấn đề văn bản cho doanh nghiệp lớn như những báo cáo về doanh thu, lợi nhuận, sản phẩm của họ, để tìm kiếm và phân tích những điểm quan trọng trong những tài liệu của doanh nghiệp, hiện nay văn bản là thứ quan trọng và phổ biến nhất để trình bày mọi việc có thể tiếp cận đối tác.

**1.4.4. Nhà nghiên cứu xử lý ngôn ngữ tự nhiên NLP**

Nghiên cứu này mang lại mang lại một nguồn thông tin quan trọng và giá trị. Trọng tâm trong nghiên cứu là tập trung vào đánh giá độ tương đồng của một hoặc nhiều tập văn bản. Họ sẽ nắm lấy được những cơ hội về những phương pháp mới trong việc đo lường sự tương đồng giữa các tập văn bản như đã đề cập ở trên.



**Hình 3.** Giải thích từ viết tắt **NLP**

**1.4.5. Phạm vi nghiên cứu**

* **Ứng dụng cụ thể:** áp dụng vào những lĩnh vực ứng dụng thực tế như: tài chính, kinh tế, giáo dục, y tế,… để giải quyết những vấn đề cần giải quyết trong từng lĩnh vực đó.
* **Tính thực tế:** nghiên cứu triển khai trực tiếp trong môi trường thực tế, đảm bảo chương trình nghiên cứu mang lại giá trị có thể sử dụng trong chính phần kết quả mang lại.
* **Tối ưu hóa hiệu suất của thuật toán nghiên cứu:** giảm bớt những bước phức tạp, chỉ tiến vào trọng tâm của thuật toàn tính độ tương đồng để tránh bị sai lệch kết quả khi chạy chương trình.
* **Giải quyết vấn đề:** nếu xảy ra lỗi chương trình sẽ xác định cụ thể lỗi trong lĩnh vực mà người dùng đang gặp phải và trực tiếp giải quyết ngay vấn đề đó.
* **Tương tác với người dùng:** trao đổi với họ về trải nghiệm khi sử dụng chương trình này, tiếp thu những những xét từ người dùng để cải tiến chương trình một cách tốt nhất.
* **Nói chung:** Phạm vi của chương trình sẽ hoạt động tại một ngưỡng nhất định và dựa vào sự nhận xét, góp ý từ phía người dùng. Điều này chính là điểm mạnh để chương trình có thể phát triển và phạm vi hoạt động ngày càng được mở rộng.

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

## 2.1. LÝ THUYẾT

### 2.1.1. Giới thiệu

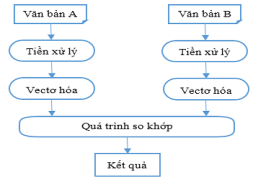
Phương pháp biểu diễn văn bản: Ngày nay, chúng ta thường thấy rất nhiều văn bản được đăng tải lên Internet ngày càng nhiều thông qua các nền tảng như MXH, bài báo,… và không thể thiếu những bài báo hoặc bài viết sao chép nội dung của người khác để đăng tải lên thành bài của riêng mình. Tình trạng này xuất hiện ngày càng nhiều khiến cho người đọc có thể hiểu nhầm nội dung chính của bài báo thật sự là gì.

Trên thực tế hiện nay đã có nhiều những công trình nghiên cứu về tính độ tương đồng văn bản, những công trình nghiên cứu đó đã tạo ra cho việc so sánh văn bản trở nên dễ hơn và được sử dụng trong một số lĩnh vực như: phát hiện sao chép văn bản, dịch thuật, tìm kiếm văn bản,…

Tính đến thời điểm hiện tại đã có một số thuật toán giải quyết vấn đề so khớp văn bản như: **Jaccard, Levenshtein, Cosine** & **Matthanan**. Những thuật toán này bước đầu tiên sẽ chuyển đổi văn bản thành những tập hợp đã qua quá trình xử lý để ngôn ngữ máy máy có thể hiểu được và đưa vào quá trình tính toán.

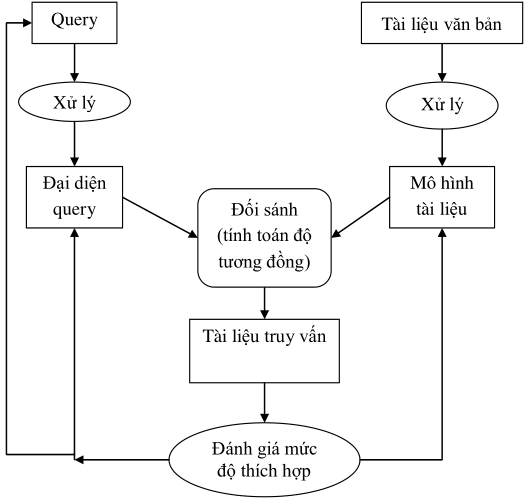
Cách biểu diễn văn bản: Trong việc xử lý văn bản để tính toán thì có rất nhiều cách nhưng nhìn chung thì những cách làm thông thường sẽ không tác động trực tiếp đến văn bản ban đầu mà phải trãi qua quá trình tiền xử lý và mô hình hóa văn bản. Quá trình tiền xử lý tùy theo độ phức tạp của văn bản như: có quá nhiều kí tự đặc biệt, chữ viết hoa, từ dừng (stopword), dấu ngắt câu,… Nói chung thì một tập dữ liệu văn bản ban đầu cần phải qua quá trình xử lý chặt chẽ để chương trình có thể tính toán một cách dễ dàng hơn và biễu diễn văn bản một cách rõ ràng hơn. Và sẽ tùy thuộc vào từng thuật toán mà sẽ có nhiều cách tính toán và biễu diễn văn bản cho phù hợp với thuật toán đó.

**Mô hình tổng quát:**



**Hình 4.** Mô hình tính độ tương đồng văn bản

Theo như lý thuyết và theo sự nghiên cứu của chương trình này thì việc chuyển đổi văn bản thành dạng vectơ sẽ có thuật toán cần thiết và có thuật toán sẽ không cần như thuật toán **Cosine & Matthanan** thì sẽ cần vì thuật toán này phải tính góc nên cần chuyển về dạng vectơ còn theo nhơ hai thuật toán trong nghiên cứu của tôi hầu như không cần phải biễu diễn thành dạng vectơ vì chúng sử dụng các từ và kí tự trong văn bản để tính toán và cho ra kết quả.



**Hình 5.** Quá trình xử lý chuyên sâu

### 2.1.2. Độ đo Jaccard

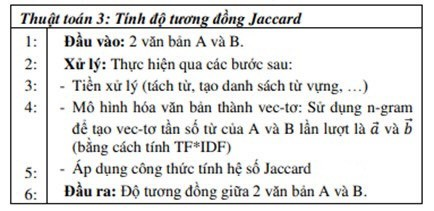
**2.1.2.1. Mô tả:** Độ đo **Jaccard** là một phương pháp đo lường độ tương đồng giữa hai tập hợp. Phương pháp này được sử dụng rộng rãi trong lĩnh vực khám phá dữ liệu, xử lý ngôn ngữ tự nhiên, và nhiều ứng dụng khác.

Độ đo **Jaccard** đo lường mức độ giống nhau giữa hai tập hợp bằng cách tính tỷ lệ giữa số phần tử chung của chúng và tổng số phần tử của hai tập hợp.

Và nói một cách đơn giản nhất **Jaccard** được tính bằng cách lấy số lượng phần tử chung giữa hai tập hợp và chia cho tổng số phần tử trong cả hai tập hợp.

Sử dụng thuật toán **Jaccard** này khá phù hợp trong việc đánh giá độ tương đồng văn bản vì nó rất dễ sử dụng chỉ cần có dữ liệu đầu vào và sau đó sử dụng thuật toán của Jaccard để tính ra kết quả tương đồng vì thuật toán Jaccard lấy phần giao nhau giữa 2 chuỗi để tính ra tỉ lệ giống nhau giữa 2 văn bản.

Quá trình sử dụng độ đo **Jaccard** để đánh giá độ tương đồng:



**Hình 6.** Các bước thực hiện tính độ tương đồng văn bản sử dụng độ đo Jaccard

**2.1.2.2. Công thức:**

∣*A*∪*B*∣ : phần giao của 2 tập hợp  
∣*A*∩*B*∣: phần hợp của 2 tập hợp

Cũng có rất nhiều công thức để tính nhưng công thức này tối ưu nhất và dễ áp dụng nhất.

**2.2.1.3. Ưu điểm và nhược điểm:**

* **Ưu điểm:**- Công thức tính toán đơn giản và dễ hiểu.

- Chấp nhận những tập dữ liệu lớn và không đòi hỏi nhiều tài nguyên để tính toán.

- Hiệu suất tốt khi so sánh các tập hợp lớn, vì chỉ cần tính số lượng phần tử chung và tổng số phần tử.

- Không quan trọng về thứ tự hoặc vị trí của các phần tử trong tập hợp, chỉ quan tâm đến sự xuất hiện hay không xuất hiện của chúng.

- Thích hợp cho nhiều ứng dụng như phân loại văn bản, tìm kiếm thông tin tương tự, và phân tích cụm dữ liệu.

* **Nhược điểm:**

**-** Chưa thể hiện được độ giống nhau chính xác giữa hai tập hợp, chỉ chú ý đến sự xuất hiện hoặc không xuất hiện của các phần tử.

**-** Không phản ánh mức độ giống nhau giữa các tập hợp. Hai tập hợp có thể có cùng kích thước nhưng khác biệt đáng kể về nội dung.

**-** Không xem xét trọng số của các phần tử, điều này có nghĩa là mỗi phần tử đều được xem xét như nhau mà không phụ thuộc vào tầm quan trọng của nó.

**-** Không quan trọng đến thứ tự của các phần tử, có thể dẫn đến việc bỏ qua thông tin quan trọng về cấu trúc hoặc sự liên kết giữa các phần tử.

**-** Đôi khi không phù hợp cho việc so sánh các tập hợp nhỏ, vì kết quả có thể bị thể hiện không chính xác. Nhưng người dùng sẽ không kiểm tra những tập văn bản nhỏ vì những tập văn bản này người dùng có thể tự kiểm tra được.

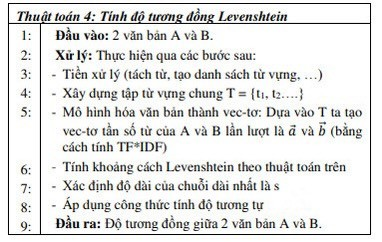
### 2.1.3. Độ đo Levenshtein

**2.1.3.1. Mô tả:** Độ đo **Levenshtein**, còn được gọi là “khoảng cách **Levenshtein**” hoặc chỉnh sửa khoảng cách, là một phương pháp đo độ tương đồng giữa hai chuỗi.

Phương pháp này đo lường số lần cần thực hiện các thao tác chỉnh sửa để chuyển đổi một chuỗi thành chuỗi khác.

Các thao tác chỉnh sửa bao gồm chèn một ký tự, xóa một ký tự hoặc thay thế một ký tự.

Quá trình sử dụng độ đo **Levenshtein** đánh giá độ tương đồng:



**Hình 7.** Các bước tính độ tương đồng văn bản sử dụng độ đo Levenshtien

**2.1.3.2. Công thức:**



d[m, n]: khoảng cách **Levenshtein** được tính bởi việc thực hiện các thao tác thêm, sửa, xóa kí tự.

S: độ dài của chuỗi dài nhất.

Theo như công thức trên thì sẽ lấy khoảng cách **Levenshtein** để áp dụng vào công thức nhưng trong sự tìm hiểu của tôi thì tỉ lệ của những công thức chỉ để đánh giá gần đúng và chỉ hiển thị xem hai tập văn bản giống và khác nhau bao nhiêu phần trăm cho nên cũng sẽ có rất nhiều biến đổi trong quá trình tính toán và thông qua nhiều nhà nghiên cứu khác nhau mà cho ra kết quả khác nhau.

Theo chương trình tính **Levenshtein** của tôi sẽ lấy tổng số kí tự được chỉnh sửa so với tổng kí tự của toàn bộ văn bản.

**2.1.3.3. Ưu điểm và nhược điểm:**

* **Ưu điểm  
  -** Cho phép đo lường sự giống nhau giữa hai chuỗi một cách cụ thể thông qua số lần thao tác chỉnh sửa.

**-** Duy trì thông tin về thứ tự và vị trí của các phần tử trong chuỗi, điều này làm cho nó phù hợp cho các bài toán yêu cầu sự chính xác về vị trí.

**-** Thường được sử dụng trong các ứng dụng kiểm tra chính tả vì có thể chỉ ra cụ thể các thao tác chỉnh sửa cần thiết.

**-** Có thể điều chỉnh để tính toán theo đơn vị thao tác chỉnh sửa hoặc theo khoảng cách trong khi giữ lại tính chính xác.

* **Nhược điểm  
  -** Độ đo **Levenshtein** yêu cầu tính toán phức tạp, đặc biệt là đối với các chuỗi dài.

**-** Không cung cấp thông tin về mức độ giống nhau tỷ lệ, chỉ cho biết số lần thao tác cần thiết.

**-** Rất phức tạp đối với các tập văn bản dài qua các thao tác chỉnh sửa.

**-** Không tính được trực tiếp mà phải trải qua đếm số kí tự trong quá trình chỉnh sửa mới có kết quả cần tính.

## 2.2. LÝ LUẬN

### 2.2.1. Sự quan trọng của tương đồng văn bản

Chương trình nghiên cứu này tập trung vào tầm quan trọng của sự tương đồng văn bản vì đây chính là yếu tố trọng điểm và then chốt trong việc phát triển các ứng dụng một cách hiệu quả trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Nắm rõ tầm quan trọng về mức độ tương đồng giữa các văn bản giúp chúng ta xác định mối liên kết ngữ nghĩa, phân loại chủ đề và cung cấp thông tin chính xác dựa trên nhu cầu của người sử dụng. Việc này có ảnh hưởng đến một số lĩnh vực liên quan đến những công cụ mà chương trình giải quyết được như: công cụ tìm kiếm, phân loại và xây dựng ứng dụng hỗ trợ, giải quyết vấn đề.

* **Xác định mối liên kết ngữ nghĩa:** Xác định mỗi liên kết ngữ nghĩa tức là có đi qua quá trình tiền xử lý nhưng sẽ chia nhỏ tập văn bản thành những đoạn văn nhỏ và có thể là những từ riêng lẻ nhưng không làm mất đi ý nghĩa của một tập văn bản gốc. Điều này rất hữu ích trong việc góp phần xây dựng nên nên các mô hình máy dịch thuật, mô hình ngôn ngữ thuần máy và trong các ứng dụng phân tích văn bản.
* **Phân loại chủ đề:** Các mô hình mà chúng ta đã nghiên cứu sẽ có trách nhiệm xem xét từng loại văn bản đầu vào và kiểm tra xem chúng thuộc loại chủ đề nào, phân loại một cách có trình tự. Để làm

được điều này thì trước đó chương trình sẽ nhờ vào quá trình tính độ tương đồng của văn bản để phân loại.

* **Công cụ tìm kiếm hiệu quả:** Trong việc tìm kiếm, sự so khớp mà chương trình đã thực hiện đóng vai trò khá quan trọng trong việc xác

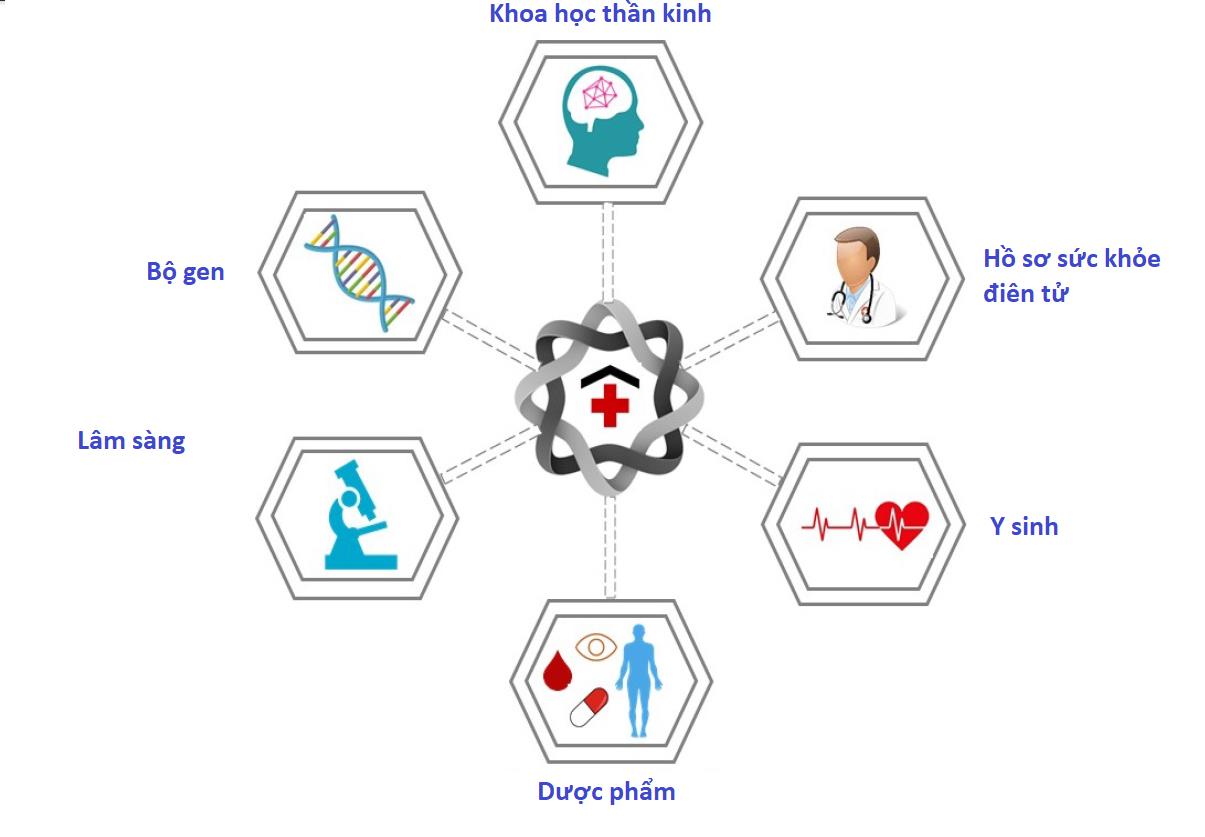
định một hoặc nhiều tập văn bản có liên quan đến việc truy vấn của người dùng hay không. Vì các thuật toán này đánh giá sự tương đồng giữa các đoạn văn bản giúp cải thiện độ chính xác và tăng hiệu suất của công cụ tìm kiếm.

* **Phân loại tự động:** Trong các nhiệm vụ phân loại tự động, sự tương đồng giữa văn bản là yếu tố quyết định. Chương trình sự tự động tra cứu và phân loại tự động không cần đến sự can thiệp của con người, điều này thúc đẩy quá trình tự động hóa một cách nhanh chóng và cụ thể.
* **Hỗ trợ quyết định:** Trong lĩnh vực hỗ trợ quyết định, sự tương đồng giữa các văn bản có thể giúp trong việc đánh giá và so sánh các sự lựa chọn. Hệ thống hỗ trợ người dùng có thể sử dụng thông tin về sự tương đồng để đưa ra các đề xuất hay truy vấn thông tin.

### 2.2.2. Tính ứng dụng rộng rãi

Việc áp dụng độ đo **Jaccard** & **Levenshtein** không chỉ giới hạn trong các bài toán cụ thể mà còn mở ra một loạt các ứng dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên.

* **Tìm kiếm thông tin:** Độ đo **Jaccard** sẽ góp phần cải thiện chất lượng kết quả tìm kiếm một cách nhanh chóng, hiệu quả và chính xác bằng việc xác định độ tương đồng giữa các đoạn văn bản đầu và và những tập tài liệu có sẵn từ phía người dùng cung cấp.
* **Phân loại văn bản:** Sử dụng đồng thời cả **Jaccard** & **Levenshtein** giúp tăng cường và hiệu quả khả năng phân loại vă văn bản dựa trên cấu trúc và nội dụng được thể hiện trong đoạn văn bản.
* **Xác định đối tượng giả mạo:** Việc so sánh kết quả từ cả hai độ đo giúp xác định các văn bản giả mạo hoặc có sự sao chép từ nguồn dữ liệu khác. Điều này quan trọng trong việc xác định tính trung thực và chất lượng thông tin sẽ ảnh hưởng đến chất lượng của nội dung trong tập văn bản nguồn.
* **Phát triển ứng dụng đánh giá:** Tạo ra ứng dụng để người dùng có thể sử dụng nhanh chóng hơn chỉ việc đưa vào hai văn bản cần so khớp thì ứng dùng sẽ cho ra kết quả trong khoảng thời gian ngắn nhất có thể và tùy vào tốc độ hiển thị kết quả nhưng tính chất của kết quả phải đảm bảo độ chính xác cao cung cấp thông tin chính xác giúp người dùng có trải nghiệm tốt nhất khi sử dụng chương trình này.
* **Trong lĩnh vực y tế:** Chương trình nghiên cứu áp dùng vào thực tế và không ngoài một lĩnh vực nào mà không thể ứng dụng, trong lĩnh vực y tế, chương trình sẽ hỗ trợ để so sánh những báo cáo lâm sàng, đề xuất phương pháp điều trị thích hợp cho bệnh nhân, theo dõi tiến triển của các tình trạng bệnh.



**Hình 8.** Thể hiện các mô hình trong lĩnh vực y tế

* **Trong lĩnh vực tư pháp:** Còn trong lĩnh vực tư pháp chương trình hỗ trợ trong việc tìm kiếm thông tin chính xác liên quan đến các vụ

án và kiểm tra pháp lý, quản lý dữ liệu pháp lý (các quy tắc, văn bản, quyết định tòa án và tư vấn pháp lý), hệ thống nhận dạng văn bản

giúp chuyển đổi tài liệu văn bản sang tài liệu điện tử, kiểm tra pháp lý, cung cấp hệ thống thông tin pháp lý trực tuyến (tài liệu pháp luật, biểu mẫu và thông báo tư pháp) trong lĩnh vực pháp luật.



**Hình 9.** Văn bản pháp luật

* **Trong lĩnh vực giáo dục:** Đo lường tương đồng văn bản có thể được sử dụng để phát hiện việc sao chép trong các bài luận, nghiên cứu học thuật, so sánh toàn bộ văn bản, đánh giá độ giống nhau giữa các bài luận hoặc giáo trình.
* **Phân đoạn và so sánh:** Chia tài liệu thành những đoạn nhỏ, so khớp sự giống nhau giữa các tài liệu, xác định cụ thể những phần bị sao chép trong tài liệu của lĩnh vực.
* **Sử dụng ngôn ngữ máy:** Giúp biểu diễn văn bản bằng ngôn ngữ máy để hiểu được ngữ cảnh và mối quan hệ ngữ nghĩa giữa các từ hoặc cụm từ trong văn bản cần so khớp. Ngôn ngữ máy rất đa dạng còn rất nhiều cách biểu diễn văn bản tùy lùy độ phức tạp của từng loại văn bản mà chọn cách biểu diễn cho phù hợp.
* **Hệ thống phát hiện sao chép:** Tạo ra hệ thống tự động duyệt những tập văn bản mang tính sao chép từ ít đến nhiều trong các bài luận. Hệ thống này thường kết hợp với chương trình tính độ tương đồng văn bản để tăng cường độ chính xác cao hơn.
* **Duyệt bài kiểm tra & thi của học sinh, sinh viên:** Tra cứu xem học sinh, sinh viên có sao chép bài của nhau trong các lần kiểm tra và các

lần thi quan trọng giúp học sinh, sinh viên có tinh thần tự giác cao trong quá trình làm thực hiện bài kiểm tra hoặc thi của họ.



**Hình 10.** Chuyển đổi số trong lĩnh vực giáo dục

## 2.3. GIẢ THUYẾT KHOA HỌC

### 2.3.1. Giả thuyết chính

* Giả thuyết chính của nghiên cứu là hiệu suất cuối cùng của thuật toán đánh giá sự tương đồng văn bản, khi sử dụng đồng thời cả hai thuật toán **Jaccard** & **Levenshtein**, sẽ có tính khả thi hơn về kết quả khi sử dụng đồng thời cả hai đối với việc chỉ sử dụng một thuật toán độc lập. Chương trình giả định rằng sự kết hợp của hai thuật toán này sẽ mang lại cái nhìn tổng quan phong phú, chính xác hơn về mức độ tương đồng văn bản.

Điều này có thể tạo ra các ứng dụng vượt trội về việc tìm kiếm thông tin, phân loại văn bản, và xác định đối tượng giả mạo.

* Lựa chọn sự kết hợp của hai độ đo **Jaccard và Levenshtein** như một phương tiện để đánh giá tương đồng văn bản với hy vọng rằng sự đa

dạng trong cách đo lường này sẽ cung cấp thông tin đầy đủ và đồng thời cải thiện độ chính xác. Mỗi thuật toán có từng đặc điểm riêng, và việc kết hợp chúng có thể ứng dụng được trong những khía cạnh đặc biệt của

từng độ đo để tạo ra một phương pháp đánh giá toàn diện, tổng quan quá trình kết quả mang đến sự hài lòng cho người sử dụng.

* Ứng dụng của giả thuyết này có thể rất quan trọng trong nhiều ngữ cảnh, từ cải thiện hiệu suất của các hệ thống tìm kiếm thông tin đến việc hiểu rõ hơn về đối tượng giả mạo trong các ứng dụng an ninh. Qua đó, giả thuyết chính này mở ra những triển vọng tích cực về việc nâng cao khả

năng xử lý và hiểu ngôn ngữ tự nhiên, có ảnh hưởng đến nhiều lĩnh vực ứng dụng khác nhau.

### 2.3.2. Giả thuyết phụ

* Giả thuyết phụ của nghiên cứu là tính toán đồng thời độ đo **Jaccard** & **Levenshtein** không chỉ cung cấp một cái nhìn toàn diện về sự tương đồng mà còn giảm chi phí tính toán so với việc tính toán từng độ đo riêng lẻ. Chúng ta giả định rằng việc thực hiện tính toán đồng thời sẽ giảm độ phức tạp tính toán và tăng cường hiệu suất của thuật toán. Điều này có thể làm cho quá trình đánh giá sự tương đồng văn bản trở nên hiệu quả hơn, đặc biệt là trên các tập dữ liệu lớn và trong các ứng dụng đòi hỏi thời gian thực hiện nhanh chóng.
* Việc tính toán đồng thời **Jaccard** & **Levenshtein** được kỳ vọng sẽ giảm chi phí tính toán bằng cách sử dụng cùng một lượt đi qua dữ liệu văn bản. Điều này có thể giảm thiểu những lần tính toán không cần thiết và

tận dụng được các phép toán chung. Việc giảm độ phức tạp tính toán có thể đặc biệt quan trọng khi áp dụng thuật toán cho các tập dữ liệu lớn

hoặc trong các ứng dụng yêu cầu xử lý nhanh chóng để đáp ứng yêu cầu thời gian thực.

* Nếu giả thuyết phụ này được chứng minh đúng, nó có thể mang lại lợi ích lớn cho cả lĩnh vực xử lý ngôn ngữ tự nhiên và các ứng dụng liên quan. Điều này có thể tạo ra một phương pháp đánh giá tương đồng văn bản hiệu quả hơn và có khả năng xử lý tốt trên các tập dữ liệu đa dạng và trong môi trường yêu cầu tính toán cao.

### 2.3.3. Dự đoán lợi ích về mọi mặt

Giả thuyết chính và giả thuyết phụ được xây dựng dựa trên giả định rằng sự kết hợp của độ đo **Jaccard** & **Levenshtein** sẽ không chỉ cung cấp thông tin tốt hơn về mức độ tương đồng văn bản mà còn mang lại lợi ích về chi phí tính toán. Sự toàn diện của giả thuyết này có thể được ước lượng bằng cách tiến hành thử nghiệm trên nhiều tập dữ liệu và so sánh kết quả với việc sử dụng chỉ một độ đo. Nếu hiệu suất và chi phí tính toán đều cải thiện, điều này có thể chứng minh tính khả thi và lợi ích của phương pháp đề xuất.

* **Thử nghiệm trên nhiều tập dữ liệu:** Giả thuyết này sẽ được kiểm tra và ước lượng thông qua việc thực hiện thử nghiệm trên nhiều tập dữ liệu có tính đa dạng. Việc này nhằm đảm bảo tính chung của kết quả và khả năng áp dụng rộng rãi của phương pháp trong các ngữ cảnh thực tế.
* **Hiệu suất tính toán:** Hiệu suất của phương pháp sẽ được đo lường bằng cách so sánh độ tương đồng được đánh giá bởi thuật toán với việc sử dụng chỉ một độ đo. Nếu sự kết hợp của **Jaccard** & **Levenshtein** mang lại kết quả tốt hơn, đặc biệt là trong việc xác định mức độ tương đồng văn bản, thì giả thuyết chính sẽ được chứng minh.
* **Tính khả thi:** Nếu kết quả thử nghiệm cho thấy cả hiệu suất và chi phí tính toán đều được cải thiện so với việc sử dụng chỉ một độ đo, điều này có thể chứng minh tính khả thi và lợi ích của phương pháp đề xuất. Sự toàn diện của giả thuyết sẽ được đánh giá dựa trên kết quả từ các thử nghiệm và ứng dụng thực tế.

## 2.4. PHƯƠNG PHÁP NGHIÊN CỨU

### 2.4.1. Xây dựng thuật toán

Trước hết, nghiên cứu sẽ tập trung vào việc phát triển một thuật toán toàn diện cho việc đánh giá sự tương đồng văn bản, tích hợp cả độ đo **Jaccard** & **Levenshtein**. Thuật toán này sẽ được xây dựng với mục tiêu đảm bảo sự chính xác, linh hoạt, và hiệu quả trong việc đo lường mức độ giống nhau giữa các đoạn văn bản.

* **Mục tiêu:** Nghiên cứu sẽ tập trung vào phát triển một thuật toán toàn diện để đánh giá sự tương đồng văn bản, tích hợp cả độ đo **Jaccard** & **Levenshtein**. Mục tiêu chính là xây dựng một thuật toán chính xác, linh hoạt và hiệu quả trong việc đo lường mức độ giống nhau giữa các đoạn văn bản.
* **Chuẩn bị dữ liệu:**
* Thu thập và chuẩn bị các tập dữ liệu văn bản đa dạng để kiểm tra và đánh giá hiệu suất của thuật toán.
* Tập văn bản mẫu và những tập văn bản cần so khớp.
* Lấy một số dữ liệu mẫu chuyên kiểm tra độ tương đồng để xem chương trình hoạt động ra sao.
* **Thiết kế thuật toán:**
* Xây dựng một giải thuật kết hợp độ đo **Jaccard** & **Levenshtein** để đo lường sự tương đồng văn bản.
* Tối ưu hóa thuật toán để giảm độ phức tạp tính toán và tăng tốc quá trình đánh giá tương đồng.
* **Thử nghiệm và đánh giá:**
* Thực hiện thử nghiệm trên các tập dữ liệu đã chuẩn bị để đánh giá hiệu suất của thuật toán.
* So sánh kết quả đo lường tương đồng của thuật toán với việc sử dụng chỉ một độ đo để kiểm tra tính hiệu quả và chính xác.
* **Tối ưu hóa và điều chỉnh:**
* Dựa trên kết quả thử nghiệm, tối ưu hóa thuật toán để cải thiện hiệu suất và chi phí tính toán.
* Điều chỉnh các tham số để đảm bảo linh hoạt và độ chính xác của thuật toán.
* **Đánh giá và tính ứng dụng:**
* Áp dụng thuật toán vào các ứng dụng cụ thể như tìm kiếm thông tin, phân loại văn bản, và xác định đối tượng giả mạo.
* Đánh giá khả năng ứng dụng và hiệu quả thực tế của thuật toán trong các ngữ cảnh thực tế.
* Đối với tính ứng dụng thì hầu như lĩnh vực nào cũng có thể áp dụng vì văn bản là công cụ phổ biến nhất để lưu trữ tài liệu của tất cả các lĩnh vực.

### 2.4.2. Tối ưu hóa và đánh giá

**2.4.2.1. Tối ưu hóa thuật toán:** Sau khi xây dựng, thuật toán sẽ được tối ưu hóa để đảm bảo tính thực tế và hiệu suất cao. Các kỹ thuật tối ưu hóa sẽ được áp dụng để giảm độ phức tạp tính toán và tăng tốc quá trình đánh giá sự tương đồng văn bản.

* **Điều chỉnh tham số:**
* Điều chỉnh các tham số của thuật toán để tối ưu hóa hiệu suất trên các loại dữ liệu khác nhau.
* Tối ưu hóa ngưỡng quyết định để cân bằng giữa độ chính xác và tốc độ thực hiện.
* **Tối ưu hóa thuật toán động:** Sử dụng kỹ thuật tối ưu hóa động để điều chỉnh cấu trúc thuật toán dựa trên đặc điểm của dữ liệu cụ thể.

**2.4.2.2.** **Đánh giá hiệu suất:** Thuật toán sẽ được đánh giá trên nhiều loại dữ liệu kiểm thử, bao gồm cả các tình huống tương đồng và khác biệt

đáng chú ý. Điều này giúp kiểm chứng giả thuyết khoa học và đánh giá tính ổn định và linh hoạt của thuật toán trong nhiều ngữ cảnh sử dụng.

* **Dữ liệu kiểm thử:**
* **Tương đồng:** Bao gồm các cặp văn bản có mức độ tương đồng cao về nội dung hoặc chủ đề. Đánh giá khả năng của thuật toán trong việc nhận diện sự tương đồng chính xác trong các ngữ cảnh giống nhau.
* **Khác biệt:** Các cặp văn bản với sự khác biệt lớn về nội dung hoặc chủ đề. Kiểm tra tính ổn định và linh hoạt của thuật toán khi đối mặt với sự đa dạng trong dữ liệu.
* **Phương thức đánh giá:**
* **Độ chính xác:** So sánh kết quả đo lường tương đồng của thuật toán với kết quả sử dụng chỉ một độ đo để kiểm tra tính chính xác. Đánh giá khả năng phân biệt giữa các mức độ tương đồng khác nhau.
* **Thời gian thực hiện:** Đo lường thời gian thực hiện của thuật toán trên các tập dữ liệu thử nghiệm để đánh giá hiệu suất tính toán. So sánh với các phương pháp khác để đảm bảo tính hiệu quả.
* **Tính linh hoạt và ổn định:** Kiểm tra khả năng của thuật toán đối mặt với biến động và đa dạng trong dữ liệu. Đảm bảo rằng thuật toán không mất đi hiệu suất khi áp dụng vào nhiều ngữ cảnh sử dụng.
* Đánh giá trên cả tình huống tương đồng và khác biệt sẽ giúp kiểm chứng giả thuyết khoa học và đảm bảo rằng thuật toán là linh hoạt, ổn định và có khả năng hoạt động hiệu quả trong nhiều ngữ cảnh sử dụng khác nhau. Kết quả đánh giá sẽ cung cấp thông tin quan trọng để điều chỉnh và tối ưu hóa thuật toán, đồng thời xác nhận tính chính xác và đáng tin cậy của nghiên cứu.

### 2.4.3. So sánh và phân tích

Kế tiếp, nghiên cứu sẽ so sánh kết quả của việc sử dụng cả hai độ đo **Jaccard** & **Levenshtein** với việc sử dụng mỗi độ đo riêng lẻ. Bằng cách này, chúng ta có thể đánh giá ưu nhược điểm của việc kết hợp cả hai độ đo so với việc sử dụng chúng độc lập. Việc này sẽ giúp xác định tại sao và khi nào sự kết hợp có thể mang lại lợi ích cao hơn.

* **So sánh kết quả độ đo Jaccard:** Nghiên cứu sẽ xem xét kết quả khi sử dụng độ đo **Jaccard** độc lập trên tập dữ liệu kiểm thử. Phân tích các trường hợp thành công và thất bại của độ đo **Jaccard** trong việc đánh giá sự tương đồng giữa các đối tượng.
* **So sánh kết quả độ đo Levenshtein:** Tương tự như độ đo Jaccard, nghiên cứu sẽ đánh giá kết quả của độ đo **Levenshtein** khi được sử dụng độc lập trên tập dữ liệu kiểm thử. Phân tích những trường hợp mà độ đo **Levenshtein** đạt hiệu suất tốt và những trường hợp nó không phù hợp.
* **So sánh kết quả của cả hai độ đo:** Dữ liệu kiểm thử sẽ được đánh giá bằng cách sử dụng cả độ đo **Jaccard** & **Levenshtein** để tạo ra kết quả kết hợp. Phân tích những trường hợp mà sự kết hợp này mang lại lợi ích, đặc biệt là khi một trong hai độ đo không đủ để đánh giá một cách chính xác.
* **Ưu và nhược điểm của hai độ đo:** Đánh giá các trường hợp mà sự kết hợp giúp cải thiện độ chính xác của đánh giá tương đồng so với

việc sử dụng mỗi độ đo riêng lẻ. Phân tích các trường hợp khi sự kết hợp không mang lại lợi ích và có thể gây ra nhiễu loạn.

* **Cách sử dụng tốt nhất:** Dựa trên kết quả phân tích, đưa ra nhận định về những tình huống cụ thể khi sự kết hợp của cả hai độ đo sẽ mang lại lợi ích cao hơn.

### 2.4.4. Ứng dụng thực tế

Cuối cùng, thuật toán sẽ được áp dụng trong các kịch bản thực tế như phân loại văn bản và tìm kiếm thông tin. Việc này nhằm kiểm tra tính ứng dụng rộng rãi của thuật toán trong các ứng dụng thực tế, từ công nghiệp đến giáo dục, thương mại, luật pháp và y tế. Qua đó, chúng ta có thể đánh giá hiệu suất và độ linh hoạt của thuật toán trong nhiều ngữ cảnh sử dụng khác nhau.

* **Ứng dụng trong Giáo dục:**
* **Hỗ trợ nghiên cứu:** Giúp sinh viên và nhà nghiên cứu nhanh chóng truy cập các nguồn thông tin quan trọng và có liên quan đến đề tài nghiên cứu của họ.
* **Đánh giá bài luận:** Có thể được sử dụng để đánh giá tự động và phân loại bài luận, giúp giảng viên tiết kiệm thời gian và tăng tính công bằng.
* **Ứng dụng trong Y tế:**
* **Phân loại bệnh án:** Hỗ trợ trong việc phân loại và trích xuất thông tin từ bệnh án, giúp bác sĩ và nhân viên y tế tìm kiếm thông tin quan trọng nhanh chóng.
* **Dự đoán bệnh án:** Áp dụng thuật toán để dự đoán các chẩn đoán dựa trên các biểu hiện lâm sàng và kết quả xét nghiệm.
* **Ứng dụng trong Thương mại điện tử:**
* **Phân loại sản phẩm:** Tự động phân loại sản phẩm dựa trên mô tả, hình ảnh và đánh giá, giúp người mua dễ dàng tìm kiếm và so sánh.
* **Tư vấn sản phẩm:** Xây dựng hệ thống tư vấn dựa trên lịch sử mua sắm và ưa thích cá nhân của người dùng.
* **Ứng dụng trong công nghiệp:**
* **Phân loại:** phân loại quá trình xử lý và kiểm kê chất lượng của từng thành phần.
* **Tự động hóa:** hệ thống tự động hóa trong sản xuất để tăng hiệu suất cung – cầu và quản lý chất lượng được tạo ra.

# CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

## 3.1. CHUẨN BỊ NỀN TẢNG NGHIÊN CỨU

### 3.1.1. Tìm hiểu cơ sở lý thuyết

**3.1.1.1. Nghiên cứu và hiểu rõ về độ đo Jaccard và Levenshtein.**

* Hiểu rõ cách thức hoạt động của hai thuật toán và áp dụng vào quá trình tính độ tương đồng văn bản.
* **Kết hợp cả hai thuật toán:** Có thể kết hợp cả hai độ đo để có cái nhìn toàn diện về tương đồng văn bản. Độ đo **Jaccard** có thể đo lường tương đồng từ góc độ từ vựng, trong khi độ đo **Levenshtein** đo lường mức độ chỉnh sửa cần thiết.

**3.1.1.2. Tính ứng dụng của hai độ đo**

* **Độ đo Jaccard:**
* **Ứng dụng trong tìm kiếm thông tin: Jaccard** giúp xác định độ tương đồng giữa các tập từ khóa và văn bản, cải thiện chất lượng kết quả tìm kiếm.
* **Jaccard** có thể được sử dụng để xác định độ tương đồng giữa các tập từ khóa và văn bản trong quá trình tìm kiếm thông tin. Nó giúp cải thiện chất lượng kết quả tìm kiếm bằng cách xác định những văn bản có liên quan nhất đến các từ khóa đã nhập.
* **Phân loại văn bản:** Sử dụng **Jaccard** để đo lường sự tương đồng cấu trúc giữa các văn bản kết hợp hỗ trợ quá trình phân loại. Độ đo **Jaccard** có thể đo lường sự tương đồng cấu trúc giữa các văn bản, hỗ trợ quá trình phân loại. Nó còn giúp xác định xem hai văn bản có cấu trúc tương đồng nhau đến mức độ nào.
* **Xác định đối tượng giả mạo: Jaccard** có thể giúp phát hiện sự giống nhau giữa văn bản và ngăn chặn việc sao chép không đúng.
* **Ứng dụng trong băn bản:** Tính toán tương đồng dựa trên từng từ trong văn bản, không quan trọng thứ tự của từ.
  + Nói chung độ đo **Jaccard** là một công cụ mạnh mẽ trong việc đo lường văn bản giữa các tập hợp và chuỗi, ứng dụng của nó đặc biệt quan trọng trong các trường hợp tìm kiếm thông tin, phân loại văn bản và ngăn chặn việc lạm dụng sao chép.
* **Độ đo Levenshtein:**
* **Ứng dụng trong tìm kiếm thông tin: Levenshtein** thích hợp để đo lường sự giống nhau giữa các chuỗi, và có thể được sử dụng để phân loại văn bản dựa trên sự giống nhau cấu trúc. Điều này có thể áp dụng trong các hệ thống phân loại và đánh giá sự tương đồng giữa các mẫu văn bản. Sử dụng được trên nhiều tập văn bản khác nhau dựa trên sự giống nhau về cấu trúc. Việc này rất hữu ích trong các hệ thống phân loại văn bản, nơi cấu trúc và sự giống nhau về cú pháp từ ngữ và đặc biệt là không chú trọng về vị trí của từ.
* **Kiểm tra lạm dụng sao chép: Levenshtein** có thể được sử dụng để kiểm tra sự sao chép bằng cách đo lường khoảng cách giữa các văn bản. Khi khoảng cách này vượt quá một ngưỡng nhất định, có thể nói là có sự sao chép hoặc sửa đổi đáng kể giữa chúng. Hỗ trợ trong việc lạm dụng sao chép, nơi sự giống nhau giữa các văn bản được đánh giá dựa trên số thao tác chỉnh sửa cần thiết.
  + Tóm lại: đọ đo **Levenshtein** được ứng dụng rộng rãi trong các tình huống đòi hỏi đo lường sự giống và khác nhau giữa chuỗi, ứng dụng của nó có thể mở rộng đến nhiều lĩnh vực là điều hiển nhiên, từ xử lý ngôn ngữ tự nhiên đến kiểm tra sự tương đồng giữa các tài liệu. Trong các kịch bản tìm kiếm thông tin và kiểm tra lạm dụng sao chép giống như độ đo **Jaccard**, nó cũng có thể là một công cụ hữu ích để đo lường sự giống nhau và sửa đổi giữa các đoạn văn bản.

### 3.1.2. Xây dựng nền tảng lý luận

**3.1.2.1.** Đặt ra lý luận về tầm quan trọng của sự tương đồng văn bản và giả thiết khoa học về hiệu suất khi sử dụng cả hai độ đo.

* **Tầm quan trọng của sự tương đồng văn bản:** Sự tương đồng văn bản đóng vai trò quan trọng trong nhiều ứng dụng của xử lý ngôn ngữ tự nhiên và khoa học máy tính. Việc đo lường mức độ giống nhau giữa các văn bản không chỉ hỗ trợ quá trình tìm kiếm thông tin mà còn là chìa khóa cho các ứng dụng như phân loại văn bản, xác định đối tượng giả mạo, và tạo ra các công cụ thông minh có khả năng hiểu sâu sắc về ngôn ngữ tự nhiên. Hiểu biết sâu sắc về sự tương đồng giữa văn bản mang lại

ưu thế đối với nhiều ứng dụng và giúp cải thiện chất lượng và hiệu suất của các hệ thống liên quan đến xử lý ngôn ngữ.

* **Giả thuyết khoa học về hiệu suất khi sử dụng hai độ đo:** Giả thiết khoa học của nghiên cứu là sự kết hợp của cả độ đo **Jaccard** & **Levenshtein** trong quá trình đánh giá sự tương đồng văn bản sẽ mang lại hiệu suất vượt trội so với việc sử dụng mỗi độ đo một cách độc lập. Những độ đo này, khi kết hợp, có khả năng cung cấp cái nhìn đa chiều và chi tiết về mức độ giống nhau giữa các đoạn văn bản. Giả thiết này được đặt ra dựa trên ý niệm rằng sự kết hợp của các độ đo sẽ tăng cường

thông tin và độ chính xác của quá trình đánh giá, dẫn đến kết quả chính xác và hữu ích hơn trong các ứng dụng thực tế.

## 3.2. THIẾT KẾ THUẬT TOÁN

### 3.2.1. Xây dựng thuật toán

**3.2.1.1. Thuật toán được xây dựng sẽ bao gồm các bước cơ bản sau:**

* **Chuẩn bị dữ liệu:** Tiền xử lý dữ liệu để chuẩn hóa và tạo ra biểu diễn dễ xử lý cho mỗi đoạn văn bản.
* Tiền xử lý dữ liệu để chuẩn hóa và tạo ra biểu diễn dễ xử lý cho mỗi đoạn văn bản.
* Loại bỏ các ký tự không cần thiết, dấu câu, và chuyển đổi văn bản thành chữ thường.
* Tạo tập từ khóa hoặc chuỗi kí tự dựa trên văn bản sau quá trình tách từ.
* **Tính toán độ đo Jaccard:** Áp dụng công thức **Jaccard** để đo lường độ tương đồng giữa các tập từ khóa tạo ra từ mỗi văn bản.
* **Tính toán độ đo Levenshtein:** Sử dụng thuật toán **Levenshtein** để đo lường khoảng cách giữa các chuỗi kí tự của mỗi văn bản.
* **Kết hợp kết quả:** Tổng hợp kết quả từ cả hai độ đo để đưa ra một chỉ số toàn diện về sự tương đồng.
* Tổng hợp kết quả từ cả hai độ đo để đưa ra một chỉ số toàn diện về sự tương đồng.
* Thể hiện được những phần cần thiết như số từ giống nhau, khác nhau, kiểm tra tần suất xuất hiện của từ.
* Tổng hợp kết quả để đưa ra một chỉ số kết quả cuối cùng.
* Việc lựa chọn trọng số và quyết định cách kết hợp kết quả giữa độ đo **Jaccard** & **Levenshtein** cũng có thể được điều chỉnh để đáp ứng yêu cầu cụ thể của bài toán. Trong quá trình triển khai thuật toán, có thể có các tham số điều chỉnh và cải tiến tùy thuộc vào yêu cầu cụ thể của ứng dụng hoặc môi trường sử dụng.

### 3.2.2. Tối ưu hóa và điều chỉnh

**3.2.2.1.** Tối ưu hóa thuật toán để giảm độ phức tạp tính toán và đảm bảo hiệu suất cao.

* **Tối ưu hóa thuật toán Jaccard:**
* Sử dụng dữ liệu hiệu quả từ những nguồn dữ liệu ban đầu, phân đoạn từng văn bản có trình tự xử lý một cách hài hòa ngăn chặn quá trình lạm dụng tính toán.
* Loại bỏ những từ khóa và những từ dùng (stopwrod) trước khi tính toán. Vì những từ này không mang ý nghĩa gì đến toàn bộ tập văn bản đầu vào mà chỉ tăng thêm độ phức tạp cho quá trình xử lý.
* Phân đoạn văn bản để có một trình tự hợp lý giúp cho quá trình trở nên nhanh chóng và rút ngắn thời gian xử lý hơn.
* **Tối ưu hóa thuật toán Levenshtein:** Sử dụng các phương pháp như lập trình động để giảm thời gian tính toán của độ đo **Levenshtein**. Chia nhỏ chuỗi thành các phần nhỏ để giảm độ phức tạp của quá trình tính toán **Levenshtein**, đặc biệt là trong trường hợp chuỗi rất dài.

## 3.3. HIỆN THỰC HÓA THUẬT TOÁN

### 3.3.1. Ngôn ngữ lập trình và môi trường

**3.3.1.1.** **Python (vì “Python” là ngôn ngữ đang rất phổ hiện này dùng để viết các chương trình hoặc các phần mềm để xử lí các nhu cầu của người sử dụng).**

* Python được chọn làm ngôn ngữ lập trình chính để hiện thực hóa thuật toán. Điều này là do **Python** không chỉ dễ đọc và hiểu, mà còn có sẵn nhiều thư viện hỗ trợ mạnh mẽ cho xử lý ngôn ngữ tự nhiên và tính toán khoa học. Sự phổ biến của **Python** trong cộng đồng khoa học máy tính và xử lý ngôn ngữ tự nhiên cũng giúp đơn giản hóa quá trình triển khai và tương tác với các công cụ và thư viện khác nhau.
* **Hỗ trợ thư viện mạnh mẽ:** Các thư viện như chính jaccard\_score, levenshtein\_distance và Sklearn cung cấp các công cụ và hàm hỗ trợ mạnh mẽ cho tính toán khoa học và xử lý ngôn ngữ tự nhiên trong nghiên cứu này của tôi.
* **Tương tác và kiểm thử:** **Python** cho phép tương tác linh hoạt, giúp nhanh chóng kiểm tra và kiểm thử các phần của thuật toán.
* **Đa dạng:** Sự phổ biến của **Python** đồng nghĩa với sự hỗ trợ từ cộng đồng lập trình rộng lớn, cung cấp nguồn lực và kiến thức đa dạng.
* **Mã nguồn mở và cộng đồng dữ liệu:** **Python** thường được sử dụng trong các dự án nguồn mở và có sẵn cho cộng đồng khoa học dữ liệu.

### 3.3.2. Cài đặt thuật toán

**3.3.2.1.** **Triển khai thuật toán sử dụng ngôn ngữ Python:**

* **Ngôn ngữ lập trình:** Sử dụng **Python** làm ngôn ngữ lập trình chính cho việc triển khai thuật toán.
* **Thư viện và công cụ:** Sử dụng các thư viện như NumPy, pandas, và scikit-learn để hỗ trợ việc xử lý dữ liệu và tính toán. Các thư viện này cung cấp các hàm và công cụ mạnh mẽ cho xử lý ma trận, chuỗi, và triển khai các thuật toán máy học.
* **Môi trường:** Sử dụng môi trường phát triển như Jupyter Notebook để kết hợp mã nguồn, kết quả tính toán, và trình bày mô tả trong cùng một văn bản, tạo ra một bản báo cáo tương tác và dễ hiểu.

## 3.4. ĐÁNH GIÁ VÀ THỬ NGHIỆM

### 3.4.1. Chuẩn bị dữ liệu kiểm thử

* Bộ dữ liệu đa dạng và phong phú.
* Bộ dữ liệu nên chứa các đoạn văn bản với mức độ tương đồng khác nhau.
* Bao gồm cả các trường hợp đoạn văn bản hoàn toàn khác biệt và những trường hợp có sự tương đồng cao.

### 3.4.2. Thực hiện thử nghiệm

* Chạy chương trình trên từng đoạn văn bản trong tập dữ liệu kiểm thử.
* Theo dõi các chỉ số đánh giá như độ chính xác, độ đồng đều, và thời gian thực hiện.
* Ghi nhận kết quả và các vấn đề xuất phát sinh trong quá trình thử nghiệm.
* Đánh giá khả năng của thuật toán đưa ra kết quả chính xác so với thực tế.
* Đánh giá khả năng của thuật toán trong việc đồng đều xác định sự tương đồng và khác biệt giữa các đoạn văn bản.
* Đánh giá hiệu suất thời gian của thuật toán, đặc biệt là đối với các tập dữ liệu lớn.

## 3.5. SO SÁNH VÀ PHÂN TÍCH KẾT QUẢ

### 3.5.1. So sánh kết quả

* Tiến hành so sánh kết quả khi sử dụng cả độ đo **Jaccard** & **Levenshtein** với trường hợp sử dụng từng độ đo độc lập.
* Đánh giá độ chính xác của thuật toán trong việc so sánh sự tương đồng giữa các đoạn văn bản.
* Phản ánh sự linh hoạt và độ đa dạng trong cách thuật toán đánh giá.

### 3.5.2. Phân tích hiệu suất

* **Độ đo Jaccard:**
* Hiệu suất tốt trong việc đo lường sự tương đồng giữa các tập từ khóa và văn bản.
* Phù hợp cho các nhiệm vụ phân loại và tìm kiếm thông tin.
* Không đo lường cấu trúc hay thứ tự của từng phần tử trong tập.
* **Độ đo Levenshtein:**
* Phù hợp cho việc đo lường sự giống nhau giữa các chuỗi và kiểm tra sự sao chép.
* Tích hợp đánh giá về cấu trúc và thứ tự của từng ký tự.
* Độ phức tạp tính toán cao đối với các chuỗi dài.

## 3.6. KẾT LUẬN VÀ TÍNH ỨNG DỤNG

### 3.6.1. Rút kinh nghiệm và kết luận

**3.6.1.1.** Trải qua quá trình nghiên cứu, việc tổng hợp kết quả và rút ra những kinh nghiệm quan trọng là bước quan trọng nhằm đánh giá sâu sắc về hiệu suất

của thuật toán. Điều này không chỉ giúp chúng ta đánh giá khả năng đáp ứng đa dạng của thuật toán trước các thử nghiệm, mà còn cung cấp cái nhìn tổng quan về khả năng áp dụng trong các bối cảnh thực tế. Đánh giá khả năng đáp ứng đa dạng của thuật toán trước các thử nghiệm.

### 3.6.2. Tính ứng dụng và khả năng mở rộng

**3.6.2.1. Ứng dụng thực tế:**

* **Tìm kiếm thông tin:** Cải thiện hiệu suất của hệ thống thông tin, đặc biệt là trong việc tìm kiếm và sắp xếp kết quả theo mức độ tương đồng.
* **Phân loại văn bản:** Hỗ trợ tự động hóa quá trình đánh giá và phân loại các văn bản dựa trên nội dung và sự tương đồng giữa chúng.
* **Đánh giá sự tương đồng:** Đóng vai trò quan trọng trong việc đánh giá mức độ liên quan giữa các văn bản khác nhau.

**3.6.2.2. Khả năng mở rộng:**

* Thuật toán có thể được mở rộng để áp dụng trong nhiều lĩnh vực khác nhau, tùy thuộc vào yêu cầu cụ thể của từng ứng dụng.
* Các tham số và trọng số có thể được điều chỉnh để tối ưu hóa hiệu suất trong ngữ cảnh sử dụng cụ thể.

# CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

## 4.1. HIỆU NĂNG

### 4.1.1. Hiệu suất đạt được

* **Tìm hiểu và cài đặt thành công:**
* Qua nhiều nguồn tài liệu khác nhau, đã thực hiện thành công quá trình tìm hiểu và cài đặt hai thuật toán.
* Nắm bắt được cơ sở lý thuyết, cách thức hoạt động và ứng dụng thực tế của độ đo **Jaccard** & **Levenshtein**.
* **Xử lí văn bản dạng số:**
* Đã đối mặt với và giải quyết nhiều vấn đề liên quan đến việc xử lý văn bản sang dạng số.
* Áp dụng thành công các bước tiền xử lý để chuẩn bị dữ liệu cho việc đánh giá sự tương đồng bằng hai độ đo.

### 4.2.2. Độ chính xác

* **Tùy chọn văn bản và độ thích hợp:** Hiệu suất của thuật toán có thể tùy thuộc vào lựa chọn văn bản mà người dùng sử dụng để kiểm tra. Văn bản phức tạp quá mức có thể dẫn đến kết quả không chính xác. Trong trường hợp này, sai không đồng nghĩa với sai hoàn toàn, mà chỉ là sự tương đối hoặc gần đúng.
* Lựa chọn tập từ khóa tối ưu và đơn giản nhất.
* Loại bỏ những từ khóa không quan trọng, không mang lại ý nghĩa.
* Phân thành từng đoạn văn bản.
* Số lần sai sót trong quá trình tính toán.
* Kiểm tra độ dài của tập văn bản nguồn.
* Theo dõi thời gian thực hiện.
* Quản lý bộ nhớ.
* Loại bỏ thông tin dư thừa không cần thiết.
* **Sự chính xác và tương đối:** Thuật toán có thể đưa ra kết quả tương đối, và sai sót có thể xuất hiện do cách diễn đạt của thuật toán. Điều này không phải là nguyên nhân chính dẫn đến kết quả hiển thị sai, mà là do việc chuyển đổi văn bản thành dạng số học để ngôn ngữ máy có thể hiểu và tính toán bị cài đặt sai bởi nhà nghiên cứu chương trình.
* Tương đối đới với các trường hợp còn một số từ không được kiểm tra bởi hai độ đo.
* Người sử dụng cần hiểu rõ về tính tương đồng là như thế nào thì mới hiểu được kết mà mà chương trình mang lại.
* Vấn đề sai sót số học xảy ra khi chương trình cài đặt sai không đúng mục đích.
* So sánh với một số phương pháp khác để mang lại một kết quả thực nghiệm nhất.
* **Chuyển đổi văn bảng sang dạng số học:** Quá trình chuyển đổi văn bản thành dạng số học có thể mang theo sự không chính xác. Tuy nhiên, điều này không làm giảm giá trị của công cụ đo, vì người dùng vẫn có thể biết được độ tương đồng của văn bản cần so sánh.
* Quá trình n-gram chuyển đổi văn bản thành những từ liên tiếp với độ dài n. Các n-gram để chuyển đổi góp phần tăng tính chính xác cho quá trình tính toán của chương trình.

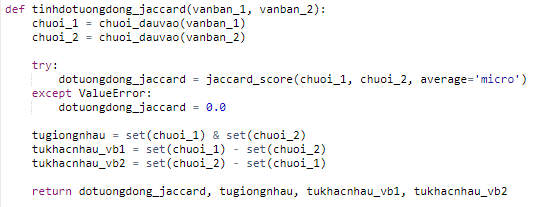
## 4.2. CÀI ĐẶT THUẬT TOÁN

### 4.2.1. Thuật toán Jaccard

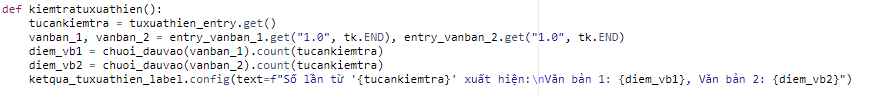
Bước đầu tiên là xử lý văn bản bằng quá trình tiền xử lý để chuyển đổi tất cả về chữ thường và chia chúng thành những từ riêng lẻ để thuận tiện cho việc tính toán.



Sau khi qua bước tiền xử lý đầu tiên thì ta gán công thức tính độ tương đồng **Jaccard** vào để tính ( vì trong công thức đã tính được số phần giao và hợp giữa hai văn bản nên chỉ cần ta gán công thức ngay lập tức công thức sẽ hoạt động. Song, liệt kê ra những từ giống và khác nhau giữa hai văn bản.

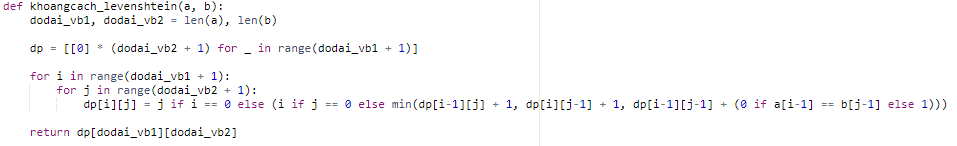


Cuối cùng thì hàm kiểm tra tần số xuất hiện của từ sẽ làm nhiệm vụ cuối cùng là hiển thị ra số lần xuất hiện của từ nào đó mà người dùng muốn kiểm tra trong hai văn bản. Hàm sẽ gọi hai văn bản và dùng phương thức “count” để điếm.

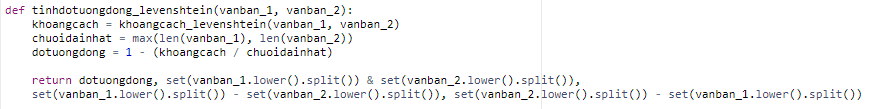


### 4.2.2. Thuật toán Levenshtein

Đầu tiên sẽ tạo hai chuỗi để lưu trữ giá trị của “vanban\_1” & “vanban\_2”, trong hàm này sẽ tính được “khoảng cách **Levenshtein**” bằng việc tính số kí tự mà các thao tác (Thêm, Xóa, Sửa) đã thực hiện để hai văn bản được tương đồng.

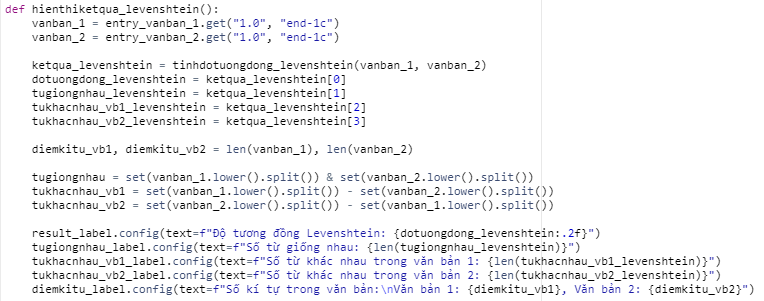


Bước tiếp theo sẽ áp dụng công thức tính độ tương đồng **Levenshtein** để tính tỉ lệ đã có và đã tính bước trước đó (khoảng cách **Levenshtein**).



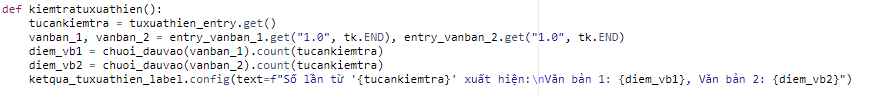
Bước kế, Gọi hàm “tinhdotuongdong\_levenshtein”: Hàm này thực hiện tính toán độ tương đồng **Levenshtein** giữa hai văn bản và trả về các thông tin liên quan như độ tương đồng, các từ giống nhau, và các từ khác nhau giữa hai văn bản.

Sau đó trả về các giá trị vừa tính được ở trên và hiển thị kết quả lên màn hình.



Cuối cùng cũng giống như độ đo **Jaccard** hàm kiểm tra tần số xuất hiện của từ sẽ làm nhiệm vụ cuối cùng là hiển thị ra số lần xuất hiện của từ nào đó mà

người dùng muốn kiểm tra trong hai văn bản. Hàm sẽ gọi hai văn bản và dùng phương thức “count” để điếm.

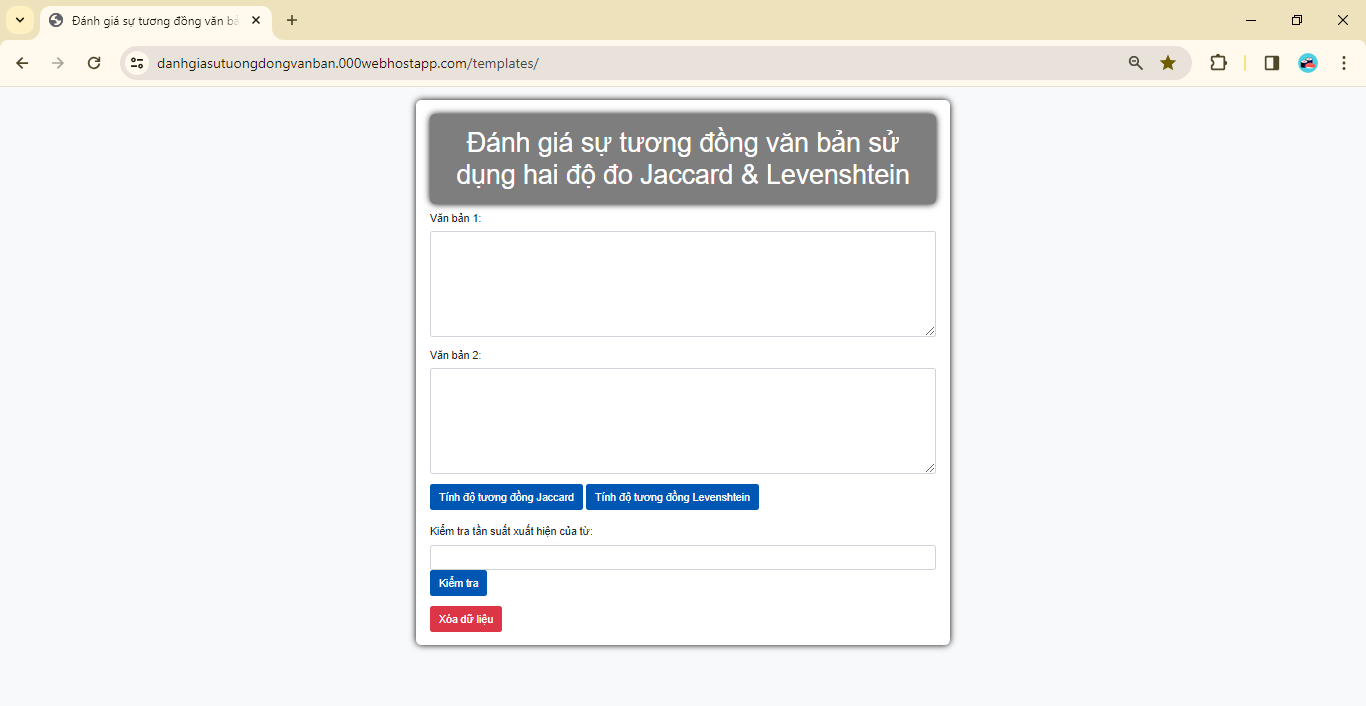


## 4.3. TRẢI NGHIỆM NGƯỜI DÙNG

### 4.3.1. Giao diện sử dụng:

Đây là website kết quả giao diện cho người sử dụng có thể trãi nghiệm:

<https://danhgiasutuongdongvanban.000webhostapp.com/templates/>

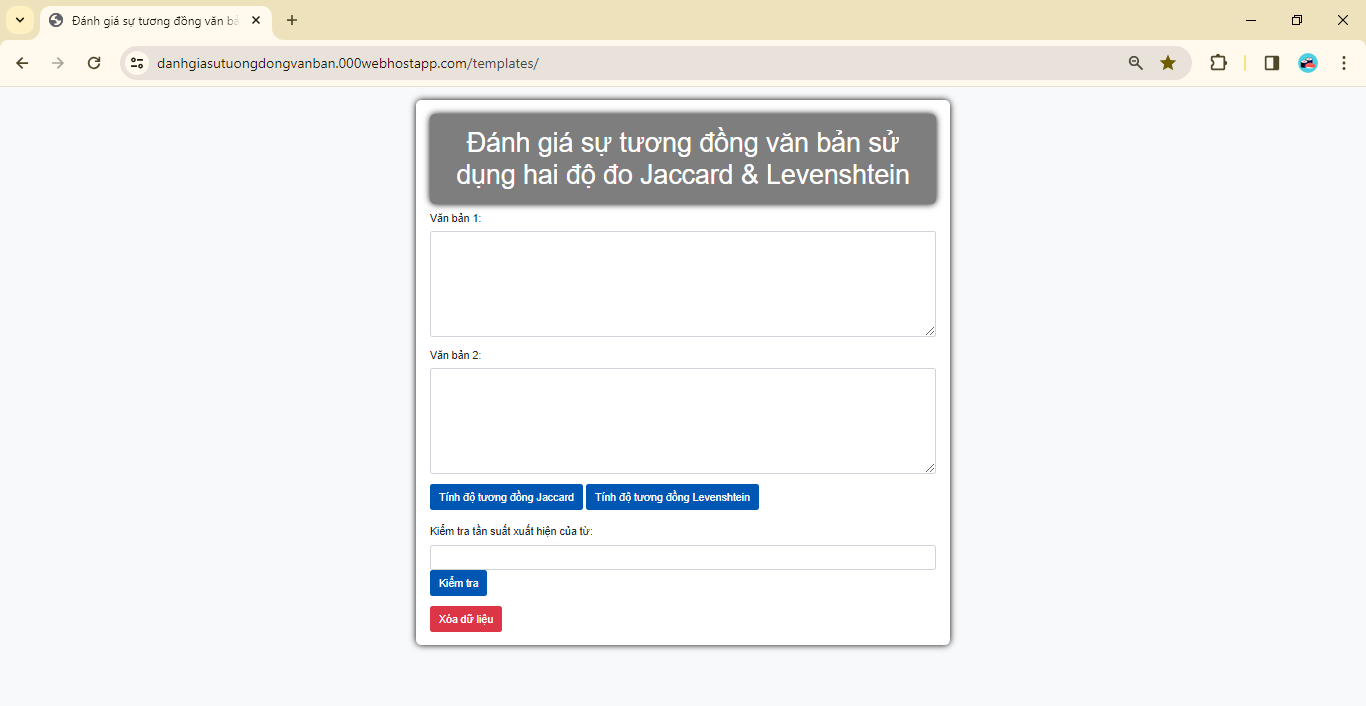


* Gồm có:

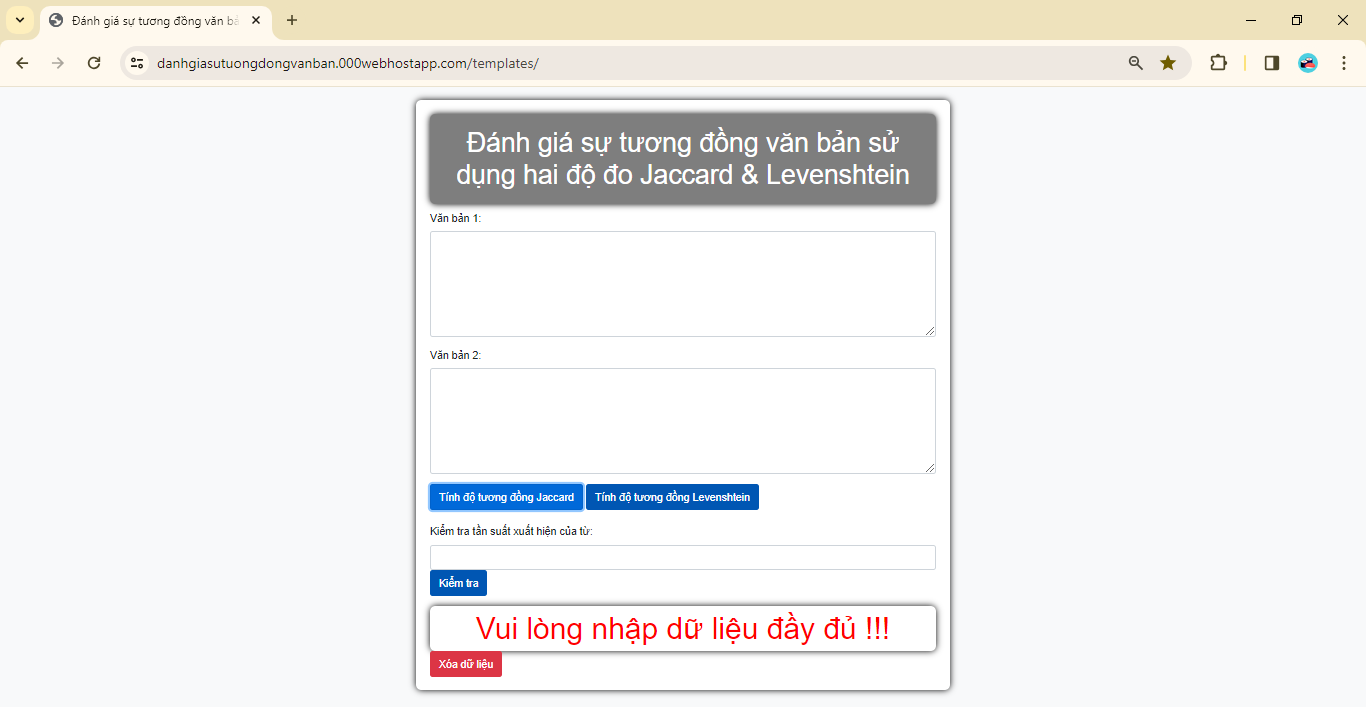
- Hai chế độ “Tính độ tương đồng **Jaccard**” & “Tính độ tương đồng **Levenshtein**”.

- Chế độ kiểm tra tần suất xuất hiện của từ bất kỳ có trong hai đoạn văn bản đầu vào.

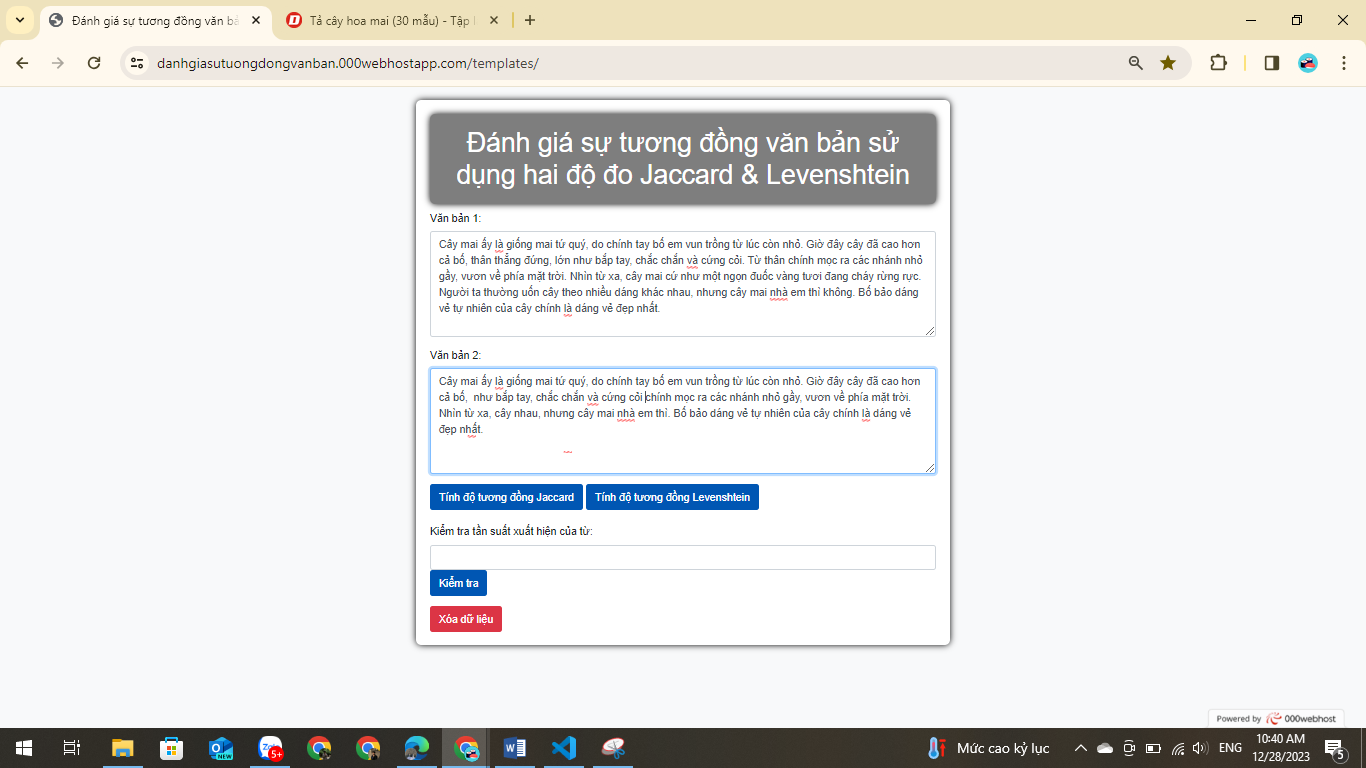
* Tối ưu hóa những phần phức tạp chỉ hiển thị kết quả tỉ lệ tương đồng **Jaccard** & **Levenshtein** và số từ giống và khác nhau của hai vă bản đầu vào, sau đó cho phép người dùng kiểm tra tần suất xuất hiện của từ nào đó trong hai đoạn văn bản đầu vào mà người dùng muốn kiểm tra.



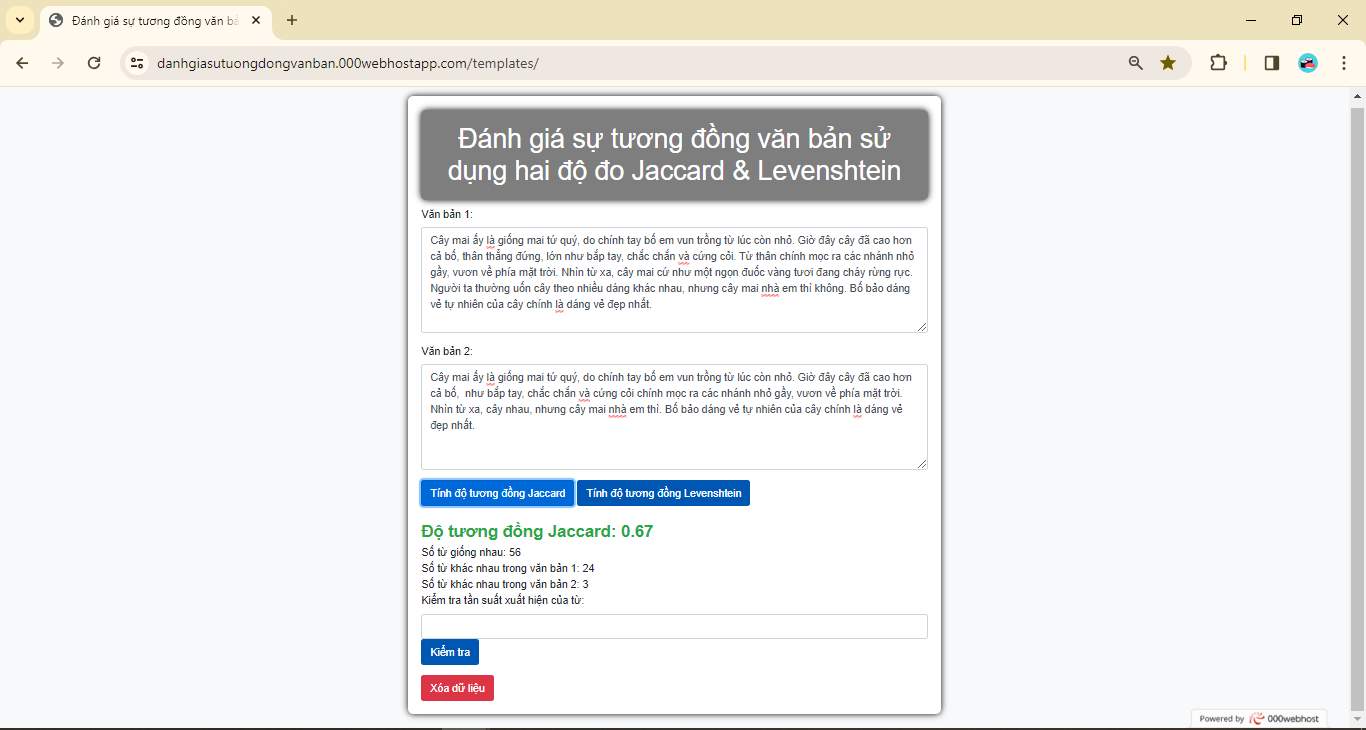
* Form sẽ báo lỗi khi người dụng chưa nhập dữ liệu vào văn bản để thực hiện kiểm tra đánh giá.



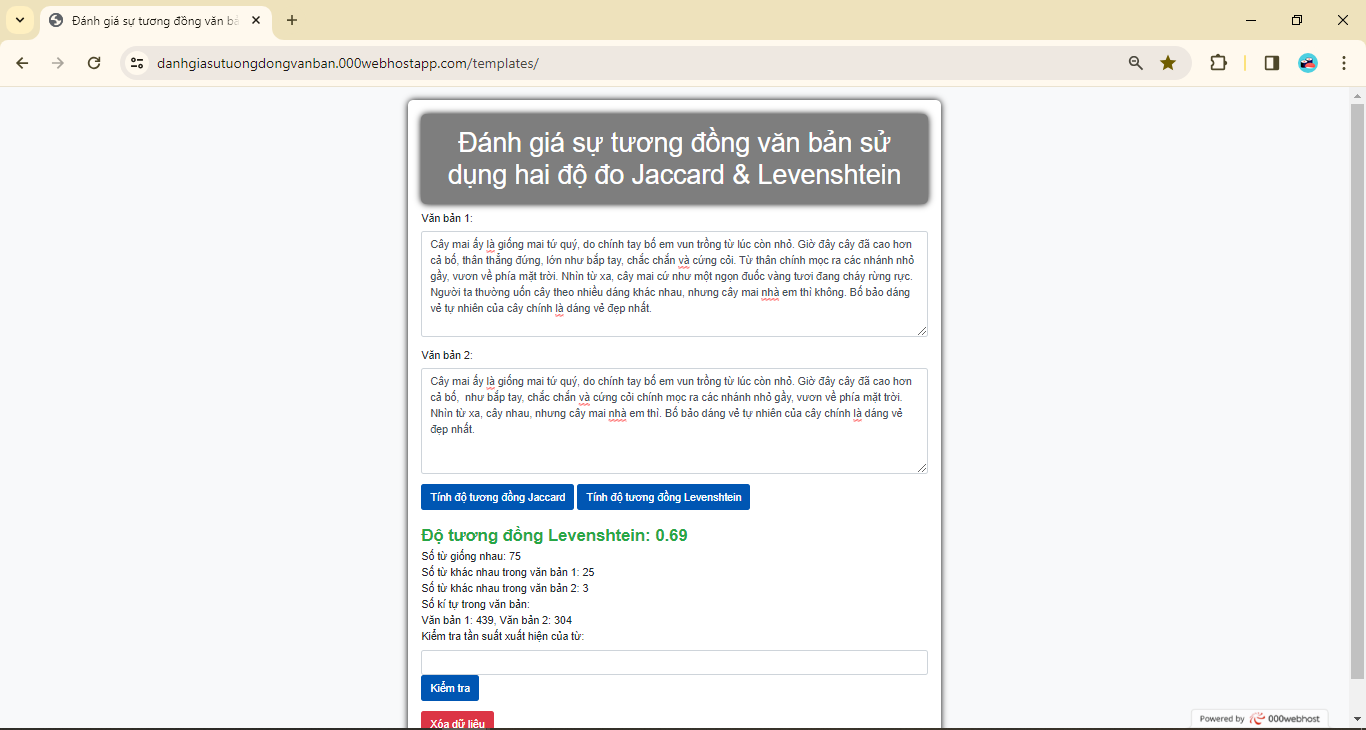
* Người dùng đưa dữ liệu để kiểm tra bằng cách chọn hai đoạn văn bản cần so sánh hoặc một văn bản mẫu và một văn bản cần so sánh với mẫu.



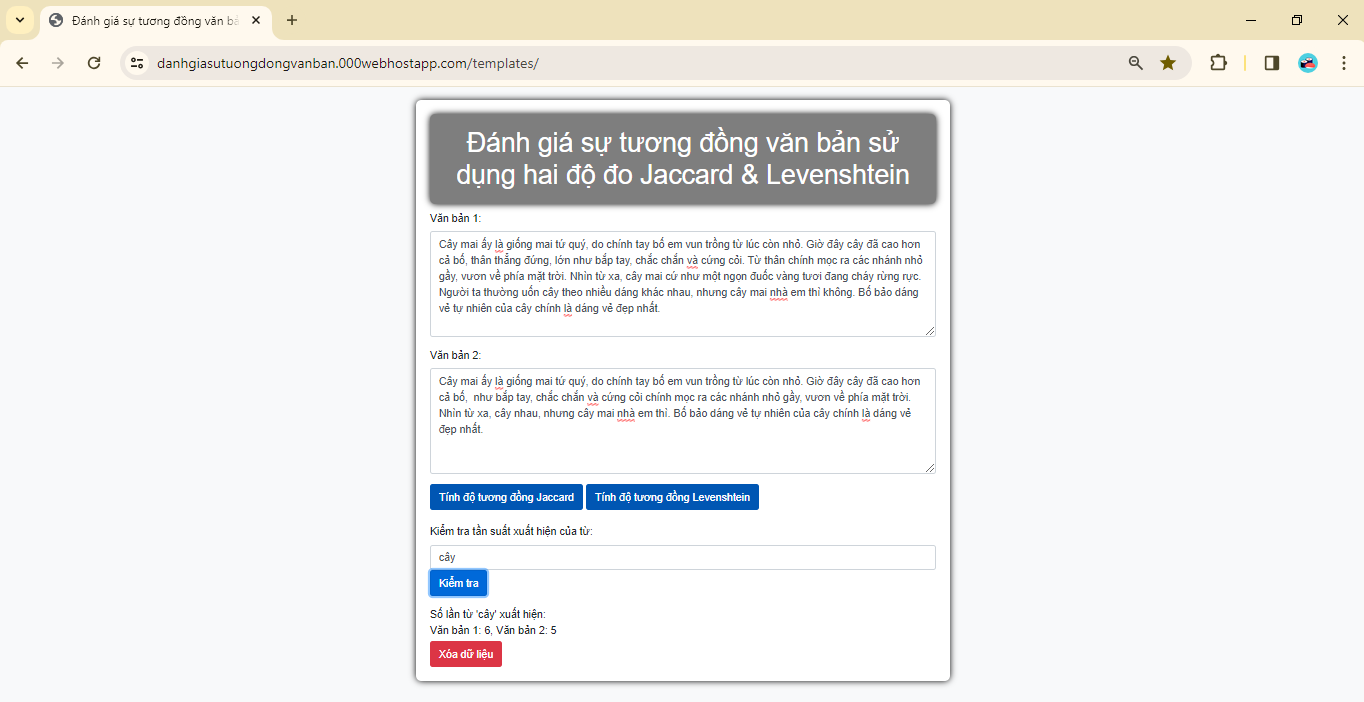
* Và sẽ có hai nút lệnh để người dùng chọn độ đo để tính tỉ lệ tương đồng của hai đoạn văn bản trên.
* Nút lệnh “Tính độ tương đồng **Jaccard**” sẽ tính độ tương đồng dựa trên thuật toán của **Jaccard**. Thuật toán này sẽ tính tỉ lệ tương đồng bằng cách lấy các phần tử chung của hai tập hợp sau đó chia cho tổng số phần tử của hai tập hợp.



* Nút lệnh “Tính độ tương đồng **Levenshtein**” sẽ tính độ tương đồng dựa trên trên thuật toán **Levenshtein**. Thuật toán này sẽ tính tỉ lệ tương đồng bằng cách tính khoảng cách **Levenshtein** (bao gồm các thao tác ‘Thêm, Xóa, Sửa để thay đổi kí tự của văn bản được so sánh với văn bản cần so sánh) sau đó áp dụng công thức **Levenshtein** cụ thể “1 – (khoảng cách **Levenshtein** đã tính trước đó / tổng số kí tự của 2 văn bản đầu vào).



* Và sau phần kết quả là ô nhập liệu để kiểm tra tần suất xuất hiện từ nào đó trong văn bản mà người dùng muốn kiểm tra.



# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. KẾT LUẬN

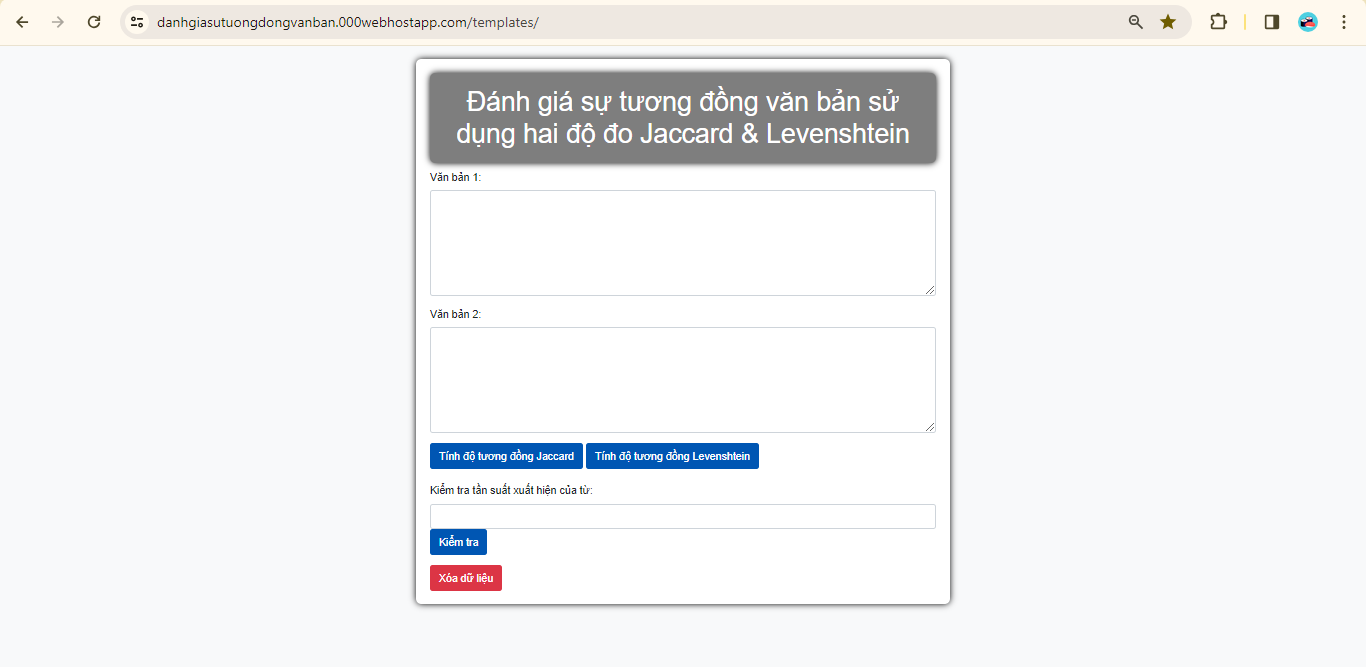
**5.1.1.** Tôi đã cài đặt thành công hai độ đo **Jaccard** & **Levenshtein** để áp dụng tính độ tương đồng văn bản.

**5.1.2.** Động cơ của dự án là xác định và triển khai thuật toán đánh giá tương đồng văn bản sử dụng độ đo **Jaccard** & **Levenshtein**. Kết quả cho thấy hiệu suất ổn định và khả năng áp dụng rộng rãi của thuật toán trong nhiều ngữ cảnh.

**5.1.3.** Đóng góp mới bao gồm việc xây dựng một hệ thống linh hoạt và hiệu quả cho việc so sánh văn bản, kết hợp cả hai độ đo **Jaccard** & **Levenshtein**. Điều này mở ra những cơ hội mới trong việc giải quyết các vấn đề liên quan đến tìm kiếm thông tin, phân loại văn bản và gợi ý nội dung.

**5.1.4.** Những đề xuất mới không chỉ tập trung vào cải thiện hiệu suất của thuật toán mà còn vào việc mở rộng chức năng và tích hợp với các công nghệ mới. Đề xuất tiếp tục nghiên cứu về cách thức đánh giá tương đồng văn bản trong bối cảnh ngôn ngữ đa dạng và dữ liệu lớn.

**5.1.5.** Upload Form tính độ tương đồng lên Website để cho phép người dùng có thể trải nghiệm kết quả nghiên cứu của tôi.



## 5.2. HƯỚNG PHÁT TRIỂN

**5.2.1**. Tôi muốn tìm hiểu thêm một số thuật toán tính độ tương đồng văn bản khác để xem quá trình tính có những điểm nào khác so với hai thuật toán trong nghiên cứu của tôi.

**5.2.2. Tăng hiệu suất:** Nghiên cứu cách tối ưu hóa thuật toán để làm cho nó chạy nhanh hơn trên dữ liệu lớn. Thử nghiệm và đánh giá các phương pháp mới để tăng hiệu suất độ đo **Jaccard** & **Levenshtein**.

**5.2.3. Triển khai thêm nhiều chức năng:** Mở rộng chức năng của thuật toán để hỗ trợ xử lý ngôn ngữ tự nhiên, bao gồm việc xử lý từ ngữ và sự hiểu biết sâu sắc về ngữ cảnh.

**5.2.4. Kết hợp với các độ đo khác:** Nghiên cứu cách kết hợp độ đo **Jaccard** **&** **Levenshtein** với các độ đo khác để tạo ra một phương pháp đánh giá tương đồng đa chiều, phong phú hơn.

**5.2.5. Đánh giá trên dữ liệu đa dạng:** Mở rộng thử nghiệm trên các tập dữ liệu đa dạng để đảm bảo tính toàn vẹn và độ chính xác của thuật toán

**5.2.6. Ứng dụng thực tế:** Áp dụng thuật toán vào các ứng dụng thực tế như hệ thống gợi ý, tìm kiếm thông tin và phân loại văn bản.

**5.2.7. Chia sẻ mã nguồn và cách vận hành:** Phát triển mã nguồn mở và cung cấp hướng dẫn chi tiết để cộng đồng có thể sử dụng và đóng góp vào dự án.



**Hình 11.** Sự nhận xét góp phần phát triển

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Hoàng Minh Hiền, “Độ tương đồng ngữ nghĩa giữa hai câu và ứng dụng trong bài toán tóm tắt văn bản”, Khóa luận tốt nghiệp đại học hệ chính quy, 2008, pp. 12 – 42.
2. Hồ Phan Hiếu, Võ Trung Hùng, Nguyễn Thị Ngọc Anh, “Một số phương pháp tính độ tương đồng văn bản dựa trên Vec-tơ”, 10/2017, pp. 112 – 117.
3. Meuschke, N. and B. Gipp, “State-of-the-art in detecting academic plagiarism”, International Journal for Educational Integrity, 9(1), 2013, pp. 50-71.
4. Khatibsyarbini, M., et al., “A hybrid weight-based and string distances using particle swarm optimization for prioritizing test cases”, Journal of Theoretical & Applied Information Technology, Vol. 95, Issue 12, 2017, pp. 2723-2732.

# PHỤ LỤC