BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

HỒ PHAN HIẾU

ĐÁNH GIÁ MỨC ĐỘ GIỐNG NHAU CỦA VĂN BẢN TIẾNG VIỆT

LUẬN ÁN TIẾN SĨ KỸ THUẬT

Đà Nẵng, 10/2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

---ꕥ❀ꕥ---

HỒ PHAN HIẾU

ĐÁNH GIÁ MỨC ĐỘ GIỐNG NHAU CỦA VĂN BẢN TIẾNG VIỆT

Chuyên ngành: KHOA HỌC MÁY TÍNH

Mã số : 62 48 01 01

LUẬN ÁN TIẾN SĨ KỸ THUẬT

Người hướng dẫn khoa học:

1. **PGS.TS. Võ Trung Hùng**
2. **TS. Nguyễn Thị Ngọc Anh**

Đà Nẵng, 10/2019

LỜI CAM ĐOAN

Tôi tên là Hồ Phan Hiếu. Tôi xin cam đoan đây là công trình nghiên cứu do tôi thực hiện. Các nội dung và kết quả nghiên cứu được trình bày trong Luận án là trung thực và mọi tham khảo đều được trích dẫn, chỉ rõ nguồn tham khảo theo đúng quy định.

Tác giả

NCS. Hồ Phan Hiếu

MỤC LỤC

LỜI CAM ĐOAN
MỤC LỤC ..... **i**
DANH MỤC CÁC TỪ VIẾT TẮT ..... **iv**
DANH MỤC BẢNG BIỂU ..... **v**
DANH MỤC HÌNH VẼ ..... **vi**
DANH MỤC THUẬT TOÁN ..... **viii**
LỜI MỞ ĐẦU **1**
1. Đặt vấn đề 1
2. Mục tiêu nghiên cứu 3
3. Đối tượng và phạm vi nghiên cứu 4
4. Phương pháp nghiên cứu 4
5. Nhiệm vụ nghiên cứu và kết quả đạt được 5
6. Bố cục của luận án 5
7. Đóng góp chính của luận án 6
Chương 1. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU **8**
1.1. Một số khái niệm sử dụng trong luận án 8
1.2. Một số đặc điểm của ngôn ngữ tiếng Việt 12
1.2.1. Khái quát 12
1.2.2. Một số khó khăn và nhập nhằng trong xử lý văn bản tiếng Việt 13
1.3. Mô hình biểu diễn văn bản 15
1.3.1. Giới thiệu 15
1.3.2. Mô hình biểu diễn văn bản 16
1.3.3. Nhận xét và đánh giá 25
1.4. Các phương pháp tính độ tương tự văn bản 27
1.4.1. Hướng tiếp cận 27
1.4.2. Bài toán so khớp chuỗi 28
1.5. So sánh văn bản và ứng dụng trong phát hiện sao chép 33
1.5.1. Giới thiệu 33
1.5.2. Các vấn đề liên quan về sao chép 34
1.5.3. Phát hiện sao chép tại PAN 38
1.6. Kết luận Chương 1 41
Chương 2. SO SÁNH VĂN BẢN DỰA TRÊN MÔ HÌNH VECTOR **42**
2.1. Giới thiệu 42

2.2. Tính độ tương tự văn bản trong mô hình vector 43
2.2.1. Biểu diễn văn bản theo mô hình vector 43
2.2.2. Phương pháp tính trọng số từ chỉ mục 45
2.2.3. Phương pháp tính độ tương tự 49
2.2.4. Nhận xét 51
2.3. Một số phương pháp so sánh văn bản dựa trên mô hình vector 52
2.3.1. Mô hình vector hóa văn bản 52
2.3.2. Phương pháp cải tiến sử dụng độ đo Cosine 57
2.4. Đánh giá các phương pháp dựa trên mô hình vector 64
2.4.1. Tạo bộ dữ liệu để đánh giá các thuật toán 64
2.4.2. Đánh giá các thuật toán dựa trên mô hình vector 65
2.4.3. Nhận xét 68
2.5. Kết luận Chương 2 68
Chương 3. PHÁT HIỆN SAO CHÉP VĂN BẢN DỰA TRÊN BIẾN ĐỔI WAVELET RỜI RẠC 70
3.1. Đặt vấn đề 70
3.1.1. Phát biểu bài toán 70
3.1.2. Đề xuất ý tưởng 72
3.2. Cơ sở lý thuyết về DWT và bộ lọc Haar 72
3.2.1. Cơ sở lý thuyết về DWT 72
3.2.2. Bộ lọc Haar 75
3.2.3. Chuỗi DNA 77
3.3. Đề xuất mô hình hệ thống phát hiện sao chép 77
3.3.1. Giới thiệu 77
3.3.2. Đề xuất mô hình hệ thống áp dụng cho phương pháp dựa trên DWT 78
3.4. Đề xuất quy trình chuyển đổi dữ liệu 81
3.5. Đề xuất phương pháp và giải thuật xử lý 81
3.5.1. Tiền xử lý dữ liệu 82
3.5.2. Quy trình số hóa 82
3.5.3. Giải thuật cho bộ lọc Haar 85
3.5.4. Tổ chức dữ liệu cho bộ DNA nguồn 88
3.6. Đề xuất thuật toán phát hiện sự giống nhau 90
3.6.1. Mã hóa dữ liệu và tính DNA của văn bản đánh giá 90
3.6.2. So sánh và đưa ra quyết định 90
3.6.3. Độ phức tạp của thuật toán phát hiện sự giống nhau 91
3.7. Kết quả thử nghiệm phương pháp dựa trên DWT 92

3.7.1. Dữ liệu thử nghiệm 92
3.7.2. Kết quả thử nghiệm 96
3.8. Đánh giá 100
3.9. Kết luận Chương 3 103
Chương 4. PHÁT TRIỂN HỆ THỐNG PHÁT HIỆN SAO CHÉP VĂN BẢN TIẾNG VIỆT 106
4.1. Mô tả hệ thống 106
4.1.1. Mục đích 106
4.1.2. Các đối tượng sử dụng 106
4.1.3. Mô hình tổng quát 107
4.2. Xây dựng kho dữ liệu văn bản tiếng Việt 108
4.2.1. Giới thiệu 108
4.2.2. Kiến trúc hệ thống kho dữ liệu 109
4.2.3. Giải pháp xây dựng kho dữ liệu 111
4.2.4. Đánh giá về kho dữ liệu 115
4.3. Triển khai hệ thống phát hiện sao chép văn bản 116
4.4. Đề xuất hướng phát triển để xử lý dữ liệu lớn 121
4.4.1. Giới thiệu 121
4.4.2. Đề xuất giải pháp xử lý 121
4.4.3. Đề xuất phương pháp biểu diễn DNA bằng Tensor 123
4.5. Kết luận Chương 4 124
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 126
1. Kết luận 126
2. Hướng phát triển 127
DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ 128
TÀI LIỆU THAM KHẢO 129

DANH MỤC CÁC TỪ VIẾT TẮT

CGs	Conceptual Graphs (*Đồ thị khái niệm*)
DM	Data Mart (*Kho dữ liệu cục bộ*)
DNA	DeoxyriboNucleic Acid (*Chuỗi DNA*)
DW	Data Warehouse (*Kho dữ liệu*)
DWT	Discrete Wavelet Transform (*Phép biến đổi Wavelet rời rạc*)
GA	Genetic Algorithms (*Giải thuật di truyền*)
IDF	Inverse Document Frequency (*Nghịch đảo tần số văn bản*)
LSI	Latent Semantic Indexing (*Chỉ mục ngữ nghĩa tiềm ẩn*)
NDD	Near Dupplicate Detection (*Phát hiện gần trùng lặp*)
NLP	Natural Language Processing (*Xử lý ngôn ngữ tự nhiên*)
PAN	Plagiarism Analysis, Authorship Identification, and Near-Duplicate detection (*Hội nghị quốc tế thường niên về đạo văn*)
SVD	Singular Value Decomposition (*Phân tích giá trị đơn*)
TF	Term Frequency (*Tần số từ khóa*)
CSDL	Cơ sở dữ liệu
ĐHĐN	Đại học Đà Nẵng

DANH MỤC BẢNG BIỂU

Bảng 1.1. Phương pháp và thuật toán đánh giá sự giống nhau của văn bản.....28
Bảng 1.2. So sánh và đánh giá một số thuật toán so khớp chuỗi.....32
Bảng 1.3. Một số phương pháp phát hiện sao chép văn bản.....35
Bảng 1.4. Kết quả các nhóm xếp thứ nhất trong nhiệm vụ EPD 40
Bảng 2.1. Các tài liệu mẫu để so với giá trị ước lượng.....64
Bảng 2.2. Tổng hợp kết quả của các phương pháp 66
Bảng 3.1. Tổng hợp so sánh các kho dữ liệu tại các cuộc thi PAN 92
Bảng 3.2. Các giá trị thiết lập cho quá trình thử nghiệm 97
Bảng 3.3. Kết quả thực nghiệm.....98
Bảng 4.1. Số tài liệu thử nghiệm được cập nhật vào kho dữ liệu 115

DANH MỤC HÌNH VẼ

Hình 1.1. Mối quan hệ giữa $\mathcal{S}$ và $\mathcal{D}$ 12
Hình 1.2. Quá trình mô hình hóa văn bản 16
Hình 1.3. Mô hình xử lý tổng quát để phát hiện sao chép [124].....39
Hình 2.1. Mô hình vector tạo thành ma trận trọng số Từ/Tài liệu 44
Hình 2.2. Ví dụ về góc tạo bởi hai vector d_1 và d_2.....44
Hình 2.3. Quá trình vector hóa theo đơn vị từ 53
Hình 2.4. Quá trình vector hóa theo đơn vị câu 62
Hình 2.5. Biểu đồ so sánh kết quả thuật toán với tập tài liệu 66
Hình 2.6. Biểu đồ so sánh văn bản theo đơn vị từ và câu 67
Hình 3.1. Mô tả cách xử lý để phát hiện sao chép văn bản.....71
Hình 3.2. Phân tích đa phân giải sử dụng DWT 73
Hình 3.3. Đường tín hiệu qua DWT [50].....75
Hình 3.4. Đường sóng Haar Wavelet.....76
Hình 3.5. Đề xuất mô hình hệ thống phát hiện sự giống nhau của văn bản 79
Hình 3.6. Quá trình xử lý để đánh giá văn bản cần kiểm tra 80
Hình 3.7. Mô hình tạo bộ dữ liệu thử nghiệm tiếng Việt.....94
Hình 3.8. Giá trị *prec* và *rec* đạt được qua các mức ngưỡng khác nhau.....98
Hình 3.9. Kết quả thực nghiệm với ngưỡng $\varepsilon = 10^{-11}$ 99
Hình 3.10. Giao diện kết quả một lần thực nghiệm 100
Hình 4.1. Quy trình phát hiện sao chép.....107
Hình 4.2. Kiến trúc hệ thống kho dữ liệu chi tiết.....110
Hình 4.3. Quy trình xây dựng kho dữ liệu 111
Hình 4.4. Quy trình xử lý, cập nhật tài liệu vào kho dữ liệu 115
Hình 4.5. Giao diện hệ thống thử nghiệm.....117
Hình 4.6. Mô hình phát hiện đánh dấu nội dung giống nhau.....118
Hình 4.7. Đánh dấu nội dung giống nhau trên tài liệu cần kiểm tra 120

Hình 4.8. Mô hình hệ thống tách và lưu trữ tài liệu theo MapReduce 123
Hình 4.9. Biểu diễn tài liệu theo mô hình Tensor [71] 124

DANH MỤC THUẬT TOÁN

Thuật toán 2.1. Tính độ tương tự có yếu tố trật tự các từ trong văn bản 61
Thuật toán 2.2. Tính độ tương tự dựa trên mô hình vector đơn vị câu.....63
Thuật toán 3.1. Quy trình mã hóa văn bản thành chuỗi tín hiệu số DNA.....81
Thuật toán 3.2. Xác định giá trị cho các chuỗi DNA.....88
Thuật toán 3.3. Lưu trữ bộ DNA nguồn.....89
Thuật toán 3.4. Phát hiện sự giống nhau.....91
Thuật toán 4.1. Đánh dấu và tô màu văn bản.....119

LỜI MỞ ĐẦU

1. Đặt vấn đề

Ngày nay, cùng với sự phát triển của Internet, hoạt động trao đổi, chia sẻ tài liệu diễn ra rất phổ biến, các tài liệu như bài báo, sách, luận văn tốt nghiệp, báo cáo, đồ án,… được số hóa và phổ biến trên mạng Internet ngày càng nhiều. Người sử dụng có thể tìm thấy những thông tin cần thiết một cách nhanh chóng và dễ dàng. Tuy nhiên, bên cạnh ưu điểm là cung cấp một nguồn tài liệu tham khảo phong phú thì tình trạng "sao chép" cũng đang trở thành một vấn nạn. Để góp phần giải quyết bài toán này, vấn đề đặt ra là làm thế nào để đánh giá được mức độ giống nhau của văn bản và chỉ ra được những nội dung sao chép trên văn bản.

Trên thế giới, các nghiên cứu và ứng dụng về xử lý ngôn ngữ tự nhiên (NLP) đã có một lịch sử phát triển lâu dài và đạt được những thành tựu nhất định. Trong những năm gần đây, NLP đã trở thành một lĩnh vực khoa học công nghệ mũi nhọn, ngày càng phát triển với nhiều ứng dụng phổ biến đã đem lại hiệu quả lớn cho xã hội như: tìm kiếm, dịch tự động, trích chọn thông tin, tóm tắt văn bản, khai phá văn bản, web ngữ nghĩa, trí tuệ nhân tạo,… và trong đó có bài toán so sánh, đánh giá độ tương tự của văn bản. Hiện có nhiều kết quả nghiên cứu về đánh giá độ tương tự trong văn bản tiếng Anh, trong đó có nhiều công trình nghiên cứu và ứng dụng hữu ích, đặc biệt là trong việc phát hiện "sao chép" hay phát hiện "đạo văn" [15, 39, 80, 90, 109]. Trong bối cảnh NLP hiện đang là một lĩnh vực ngày càng có nhiều nhà khoa học trên thế giới quan tâm thì ở Việt Nam, việc nghiên cứu xử lý ngôn ngữ tiếng Việt vẫn chưa đạt được kết quả khả quan, các công trình nghiên cứu còn hạn chế,… nên rất cần có sự đóng góp của các nhà khoa học, các nhóm nghiên cứu để góp phần đem lại hiệu quả trong lĩnh vực xử lý tiếng Việt. Vì vậy, vấn đề đánh giá sự tương tự của các đơn vị văn bản và trên cơ sở đó phát hiện ra nội dung sao chép vẫn còn nhiều thách thức cần được nghiên cứu. Đặc biệt, đối với việc xử lý tiếng Việt, do mang nhiều đặc trưng ngôn ngữ riêng nên xử lý tiếng Việt đòi hỏi các phương pháp, kỹ thuật khác so với các ngôn ngữ khác.

Hiện nay, đã có nhiều hệ thống phát hiện sao chép khá hiệu quả như: Turnitin, Copyscape, iThenticate, EVE Plagiarism Detection System, Writecheck, Grammarly,

Plagium, Dupli Checker, Plagiarism Checker, Plagiarismdetect,… Tuy nhiên, những hệ thống này chỉ cho phép phát hiện sự trùng lặp trong nguồn cơ sở dữ liệu gốc và chủ yếu dành cho các tài liệu tiếng Anh. Bên cạnh đó, việc mở rộng cơ sở dữ liệu mẫu theo yêu cầu người sử dụng trở nên khó khăn và tốn chi phí rất cao.

Mặc dù đã có nhiều nghiên cứu về hệ thống phát hiện sao chép văn bản, nhưng đến nay vẫn chưa có cơ sở chung nào để đánh giá tính hiệu quả của chúng. Trong khi đó, ở Việt Nam cũng mới bắt đầu có các nhóm nghiên cứu về lĩnh vực này [30, 85, 130] và có rất ít nhóm xây dựng hệ thống và ứng dụng vào thực tiễn hoặc thương mại hóa, nhất là các hệ thống ứng dụng cho văn bản tiếng Việt. Qua thực tế, để góp phần ngăn chặn "đạo văn", một số trường đại học đang sử dụng công cụ phát hiện sao chép "Turnitin" như: Trường Đại học FPT, Đại học RMIT, Đại học Kinh tế Quốc dân, Đại học Hoa Sen, Đại học Hàng hải Việt Nam,... công cụ này có thể truy quét khoảng 67 tỉ trang web, gần 1 tỉ bài làm của sinh viên và hơn 178 triệu bài viết từ các công trình nghiên cứu khoa học, ấn phẩm học thuật đang hoạt động trên mạng Internet để phát hiện các nội dung sao chép của văn bản[1]. Bên cạnh đó, cũng có nhiều trường đại học triển khai xây dựng phần mềm riêng hoặc thực hiện các đề tài liên quan đến phát hiện đạo văn như: Trường Đại học Cần Thơ, Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội, Trường Đại học Công nghệ TP.HCM, Trường Đại học Mở Hà Nội, Trường Đại học Duy Tân, Trường Đại học Khoa học Tự nhiên TP.HCM, Trường Đại học Kinh tế TP.HCM, Trường Đại học Tôn Đức Thắng, Học viện Kỹ thuật quân sự,... Tuy nhiên, việc phát hiện sao chép chủ yếu dừng ở mức ứng dụng trong các phạm vi riêng lẻ mà chưa có nhiều kết quả nghiên cứu khoa học để góp phần giải quyết tốt hơn vấn đề này.

Để phát triển các hệ thống phát hiện sao chép, chúng ta cần giải quyết các vấn đề chính như: 1) Xây dựng một kho dữ liệu đủ lớn, có độ bao phủ và độ liên quan cao với nội dung của các văn bản cần phát hiện nội dung sao chép; 2) Có phương pháp

[1] https://www.turnitin.com/about/content (truy cập tháng 10/2019)

biểu diễn văn bản phù hợp và hiệu quả cho quá trình so sánh; 3) Các giải thuật để tính độ tương tự giữa các đơn vị văn bản và chỉ ra các nội dung sao chép; 4) Xử lý cho khối lượng văn bản cực lớn.

Nhằm góp phần giải quyết các vấn đề trên, tôi đã chọn đề tài: **"Đánh giá mức độ giống nhau của văn bản tiếng Việt"** làm nội dung nghiên cứu cho luận án Tiến sĩ kỹ thuật của mình với mục tiêu cuối cùng là phát hiện các nội dung sao chép trên một văn bản hiệu quả nhất có thể.

Đây không phải là vấn đề mới, tuy nhiên trong nghiên cứu cần có những phát hiện, đề xuất làm cơ sở để phát triển, vì vậy ý tưởng nổi bật của luận án này là nghiên cứu, ứng dụng những thành tựu đã đạt được trong lĩnh vực tin sinh học, xử lý tín hiệu số vào lĩnh vực NLP. Điểm chung của các lĩnh vực này là khối lượng dữ liệu cần xử lý rất lớn và mục đích là chỉ ra được sự giống nhau hoặc khác biệt giữa các đơn vị dữ liệu cần xử lý. Cụ thể, luận án đề xuất một hướng tiếp cận mới trong xử lý văn bản bằng cách áp dụng phương pháp DWT và ứng dụng bộ lọc Haar để chuyển văn bản thành các chuỗi số DNA; tổ chức lưu trữ và đề xuất các giải thuật so sánh, tìm kiếm hiệu quả trong xử lý dữ liệu lớn để phát hiện và đánh giá được mức độ giống nhau trên các chuỗi DNA này. Đây là một hướng nghiên cứu mới, đầy tiềm năng để giải quyết bài toán về xử lý văn bản và dữ liệu lớn.

2. Mục tiêu nghiên cứu

Mục tiêu của luận án là tìm ra các giải pháp hiệu quả để biểu diễn, đánh giá mức độ giống nhau của các văn bản và áp dụng cho việc phát hiện sao chép. Các mục tiêu cụ thể của luận án gồm:

- Đề xuất được phương pháp hiệu quả trong biểu diễn văn bản để phục vụ tốt nhất cho quá trình phát hiện sao chép văn bản.
- Đề xuất các giải thuật nhằm cải thiện tốc độ và độ chính xác để phát hiện sao chép khi xử lý dữ liệu lớn.
- Xây dựng hệ thống phát hiện sao chép văn bản tiếng Việt và ứng dụng thử nghiệm tại ĐHĐN.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của luận án gồm:

- Các mô hình, phương pháp biểu diễn văn bản.
- Các phương pháp, thuật toán tính độ tương tự văn bản.
- Bài toán phát hiện nội dung sao chép trên văn bản.
- Các hệ thống phát hiện sao chép văn bản.

Giới hạn phạm vi nghiên cứu trong luận án này gồm:

- Tập trung vào phương pháp biểu diễn văn bản dựa trên mô hình vector. Nghiên cứu một số mô hình, phương pháp biểu diễn văn bản, chuyển dữ liệu văn bản thô thành kho dữ liệu dựa trên mô hình vector.

- Nghiên cứu đề xuất thuật toán tính độ tương tự văn bản. Luận án chỉ tập trung vào các phương pháp đo độ tương tự văn bản mà không xét đến yếu tố ngữ nghĩa của văn bản. Luận án chỉ tính toán độ tương tự văn bản dựa trên các phương pháp liên quan đến chuỗi.

- Đề xuất giải pháp tính độ tương tự văn bản tiếng Việt và triển khai thử nghiệm tại ĐHĐN.

4. Phương pháp nghiên cứu

- *Phương pháp tài liệu:* Nghiên cứu các tài liệu có liên quan đến các nội dung nghiên cứu như: Khai phá văn bản, biểu diễn và lưu trữ văn bản, một số đặc trưng cơ bản của tiếng Việt, hệ thống phát hiện sao chép văn bản, độ tương tự văn bản, so sánh văn bản để phát hiện sao chép tại PAN, phép biến đổi DWT, bộ lọc Haar, tìm kiếm nhị phân, xử lý dữ liệu lớn.

- *Phương pháp thực nghiệm:* Nghiên cứu đánh giá thực nghiệm các mô hình, phương pháp so khớp văn bản trong phát hiện sao chép. Xây dựng các chương trình so khớp văn bản. So sánh, đánh giá các phương pháp đề xuất với các phương pháp đã có qua kết quả thực nghiệm. Cuối cùng, phát triển hệ thống thực nghiệm tại ĐHĐN và đánh giá kết quả.

5. Nhiệm vụ nghiên cứu và kết quả đạt được

Để đạt được mục tiêu đề ra, nhiệm vụ nghiên cứu tập trung vào các vấn đề chính sau đây:

- Nghiên cứu, phân tích các phương pháp biểu diễn văn bản nói chung và mô hình vector nói riêng, từ đó đề xuất các thuật toán để so sánh, đánh giá và phát triển ứng dụng cụ thể.
- Khảo sát các nguồn dữ liệu cần thiết, tổng hợp nguồn tài liệu số, đề xuất giải pháp tổ chức lưu trữ, đánh chỉ mục, biểu diễn dữ liệu phù hợp.
- Nghiên cứu bài toán so sánh văn bản để phát hiện sao chép tại PAN, đề xuất giải pháp xử lý phát hiện sao chép văn bản hiệu quả.
- Khảo sát, nghiên cứu lý thuyết về DWT và bộ lọc Haar trong xử lý tín hiệu số, đề xuất giải pháp để chuyển văn bản thành chuỗi số thực (được gọi là DNA).
- Nghiên cứu đề xuất các giải thuật xử lý thông qua bộ lọc Haar, giải pháp tổ chức lưu trữ DNA phù hợp, đề xuất thuật toán phát hiện sự giống nhau.
- Nghiên cứu xây dựng bộ dữ liệu tiếng Việt thử nghiệm để phục vụ đánh giá.
- Triển khai thực nghiệm và đánh giá kết quả.

6. Bố cục của luận án

Trên cơ sở các nội dung nghiên cứu, để đạt mục tiêu đề ra và đảm bảo tính logic, ngoài phần *mở đầu* và phần *kết luận*, luận án được tổ chức thành các chương như sau:

Chương 1. Tổng quan tình hình nghiên cứu. Chương này trình bày cơ sở lý thuyết, kết quả nghiên cứu tổng quan về các vấn đề nghiên cứu trong luận án; phân tích, đánh giá các công trình nghiên cứu liên quan và chỉ ra một số hạn chế trong lĩnh vực nghiên cứu (phát hiện sao chép hay trùng lặp nội dung văn bản). Trên cơ sở các phân tích, đánh giá sẽ định hướng, đề xuất và xác định các nội dung nghiên cứu sẽ được triển khai.

Chương 2. So sánh văn bản dựa trên mô hình vector. Chương này giới thiệu tổng quan về bài toán so sánh văn bản, trình bày phương pháp tính trọng số các đặc trưng của văn bản biểu diễn trên mô hình vector; giới thiệu bài toán và thực nghiệm

một số phương pháp so sánh văn bản dựa trên mô hình vector để chứng minh mô hình biểu diễn văn bản dựa trên vector là thông dụng và có thể ứng dụng trong phát hiện sao chép văn bản tiếng Việt. Trên cơ sở phân tích, đánh giá cách tính trọng số, các phương pháp đo độ tương tự văn bản được dùng phổ biến, luận án đề xuất thuật toán thử nghiệm để đánh giá sự tương tự của văn bản tiếng Việt dựa trên mô hình vector.

Chương 3. Phát hiện sao chép văn bản dựa trên biến đổi Wavelet rời rạc. Hạn chế của mô hình vector là không gian lưu trữ với số chiều rất lớn, đối với kho dữ liệu lớn thì chi phí xử lý, tính toán cao. Chương này giới thiệu kết quả nghiên cứu, phân tích và đề xuất hướng tiếp cận mới để giải quyết bài toán so sánh văn bản dựa trên phép biến đổi Wavelet rời rạc và sử dụng bộ lọc Haar. Nội dung trình bày tập trung vào hướng tiếp cận và các vấn đề mới đối với xử lý văn bản, đó là biến đổi Wavelet rời rạc và bộ lọc Haar; phương pháp đề xuất để giải quyết bài toán. Thực nghiệm, so sánh và đánh giá kết quả đạt được để chứng minh hiệu quả của phương pháp đề xuất.

Chương 4. Phát triển hệ thống phát hiện sao chép văn bản tiếng Việt. Trình bày kết quả giải pháp xây dựng kho dữ liệu văn bản tiếng Việt và phát triển hệ thống phát hiện sao chép văn bản dựa trên các kết quả nghiên cứu đạt được về mô hình vector và phương pháp DWT. Chương này cũng trình bày kết quả triển khai thử nghiệm tại ĐHĐN và một số đánh giá. Cuối cùng là đề xuất hướng phát triển để xử lý dữ liệu lớn với giải pháp triển khai dựa trên mô hình MapReduce và phương pháp biểu diễn DNA bằng Tensor.

7. Đóng góp chính của luận án

Luận án đã góp phần giải quyết bài toán đánh giá mức độ giống nhau của văn bản ứng dụng trong phát hiện nội dung giống nhau của văn bản nhằm phát hiện sao chép.

Dưới đây là những đóng góp chính của luận án:

- *Đề xuất cải tiến mô hình vector để tính toán độ tương tự văn bản dựa trên đơn vị từ và câu.* Nghiên cứu và triển khai thực nghiệm phương pháp so sánh văn bản dựa trên mô hình vector với tập dữ liệu tiếng Việt và cho kết quả khả quan để chứng minh phương pháp dựa trên mô hình vector và sử dụng độ đo Cosine là phương pháp thông dụng có thể giải quyết được bài toán đánh giá mức độ giống nhau của văn bản.

- *Đề xuất được cách tiếp cận mới để đánh giá mức độ giống nhau của văn bản gồm phương pháp biểu diễn văn bản thành các chuỗi số thực DNA và ứng dụng phương pháp DWT và bộ lọc Haar*. Đây là cách tiếp cận hoàn toàn mới để giải quyết bài toán phát hiện sự giống nhau của văn bản.

- *Đề xuất được quy trình xử lý, xây dựng thuật toán phát hiện sự giống nhau giữa các văn bản bằng cách tính toán khoảng cách Euclid nhỏ nhất từ DNA cần đánh giá đến các DNA nguồn và so sánh với một mức ngưỡng thích hợp để đưa ra kết luận về sự giống nhau giữa văn bản được kiểm tra với văn bản nguồn trong kho dữ liệu.* Các kết quả thực nghiệm trên bộ dữ liệu chuẩn của PAN và bộ dữ liệu tiếng Việt thử nghiệm đã chứng minh thuật toán đề xuất đem lại hiệu quả cao trong phát hiện sự giống nhau của văn bản.

- *Đề xuất được các giải pháp, thuật toán để xử lý dữ liệu lớn hiệu quả*. Từ hướng tiếp cận và phương pháp đề xuất trên, trong luận án đã tính đến xử lý dữ liệu lớn với việc mã hóa dữ liệu văn bản sang dạng tín hiệu số thông qua các chuỗi DNA được sắp xếp theo thứ tự tăng dần cho phép tìm kiếm nhị phân. Đây là một trong những phương pháp tìm kiếm nhanh nhất khi làm việc với dữ liệu lớn. Hơn nữa, DWT cho độ phức tạp tính toán chỉ là hàm đa thức trong mỗi lần lấy mẫu con nên giải pháp đề xuất sẽ càng hiệu quả trong quá trình xử lý dữ liệu lớn.

- Xây dựng các bộ dữ liệu tiếng Việt để thực nghiệm, xây dựng hệ thống phát hiện sao chép văn bản và triển khai ứng dụng thử nghiệm tại ĐHĐN mang ý nghĩa thực tiễn cao.

Chương 1. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU

Chương này trình bày cơ sở lý thuyết, kết quả nghiên cứu tổng quan về các vấn đề nghiên cứu trong luận án; thực hiện việc phân tích, đánh giá các công trình nghiên cứu liên quan và chỉ ra một số hạn chế trong lĩnh vực nghiên cứu (phát hiện sao chép hay trùng lặp nội dung văn bản). Trên cơ sở các phân tích, đánh giá sẽ định hướng, đề xuất và xác định các nội dung nghiên cứu sẽ được triển khai.

1.1. Một số khái niệm sử dụng trong luận án

1) Văn bản

Văn bản (*Document/Text*) là đối tượng nghiên cứu của nhiều ngành khoa học, nhiều lĩnh vực khác nhau, tùy theo góc độ nghiên cứu và mục đích tiếp cận sẽ có nhiều định nghĩa khác nhau về văn bản.

Thông thường trong các hệ thống lưu trữ và truy xuất thông tin, văn bản là các bản ghi thông tin, là phương tiện để ghi nhận, lưu giữ và truyền đạt các thông tin từ chủ thể này sang chủ thể khác bằng một ký hiệu hay ngôn ngữ nhất định nào đó. Văn bản bao gồm các tài liệu, tư liệu, giấy tờ có giá trị pháp lý, mỗi văn bản gồm tập hợp các câu có tính trọn vẹn về nội dung, hoàn chỉnh về hình thức, có tính liên kết chặt chẽ và hướng tới một mục tiêu giao tiếp nhất định [18].

Trong tin học, văn bản là một chuỗi ký tự được mã hóa thành các định dạng có thể đọc được trên máy tính, hay còn gọi là văn bản điện tử. Trong luận án, văn bản được đề cập là đối tượng nghiên cứu chính, là văn bản điện tử ở dạng văn bản thuần (*Plain text*[2]), việc xử lý, tính toán sẽ loại bỏ các hình ảnh, công thức, bảng biểu... Hay nói cách khác, các phương pháp đề xuất trong luận án được xử lý trên đối tượng văn bản thuần, từ các văn bản điện tử được lưu trữ dưới dạng các tệp tin *.doc*, *.docx*, *.pdf*... hay tính toán, xử lý trên các nội dung văn bản được lưu trữ trong CSDL.

---

[2] https://en.wikipedia.org/wiki/Plain_text (truy cập tháng 10/2019)

2) Độ tương tự

Độ tương tự (*Similarity measures*) hay độ tương đồng là một khái niệm quan trọng và đã được sử dụng rộng rãi. Các định nghĩa trước đây về độ tương tự được gắn liền trong một ứng dụng cụ thể hoặc một dạng thể hiện của tri thức.

Dekang Lin [70] định nghĩa độ tương tự thông qua trực giác (*Intuitions*):

- *Trực giác 1*: Độ tương tự giữa A và B có liên quan tới sự tương tự (giống nhau) của chúng. Sự tương tự càng nhiều, độ tương tự càng lớn.
- *Trực giác 2*: Độ tương tự giữa A và B có liên quan tới những sự khác biệt giữa chúng. Càng nhiều sự khác biệt, độ tương tự càng thấp.
- *Trực giác 3*: Độ tương tự lớn nhất giữa A và B đạt được khi A và B giống hệt nhau hay gọi là đồng nhất.

Mỗi một độ đo độ tương tự phụ thuộc vào một ứng dụng cụ thể hoặc giả định trong một mô hình miền cụ thể. Độ tương tự được tính là giá trị nằm trong khoảng [0, 1]. Độ tương tự giữa hai đối tượng bằng 1 thì hai đối tượng đó hoàn toàn giống nhau và nếu độ tương tự bằng 0 thì hai đối tượng đó khác nhau hoàn toàn.

3) Độ tương tự văn bản

Độ tương tự văn bản (*Text similarity*) là mức độ giống nhau của văn bản. Khi so sánh hai đơn vị văn bản (toàn văn bản hoặc một phân đoạn của văn bản) với nhau thì đó là mức độ giống nhau giữa đơn vị văn bản này với đơn vị văn bản kia; so sánh giữa hai văn bản thì đó là mức độ giống nhau giữa văn bản này với văn bản kia; so sánh giữa một văn bản kiểm tra với tập văn bản khác thì đó là mức độ giống nhau giữa văn bản kiểm tra với các văn bản khác… Nói cách khác, tính độ tương tự văn bản là tìm ra tỷ lệ giống nhau của văn bản.

Meuschke and Gipp [80] trình bày việc tính độ tương tự văn bản theo hai hướng: Một là so sánh độ tương tự của các phân đoạn văn bản (*Text segment*), gọi là tương tự cục bộ và hai là so sánh độ tương tự của toàn bộ văn bản (*Full text*), gọi là tương tự toàn cục.

Để đo độ tương tự văn bản thường dùng là độ tương tự chuỗi (*String similarity*). Để đo độ tương tự các chuỗi, người ta sử dụng các độ đo khoảng cách để tính độ tương tự chuỗi [14] hoặc dùng các phương pháp đo độ tương tự ngữ nghĩa

(*Semantic similarity*) để hiểu được nghĩa của văn bản, thường liên quan đến cơ sở tri thức, các khái niệm, Ontology,... [39, 75].

Tóm lại, các văn bản tương tự nhau là những văn bản có tần số từ tương đối giống nhau, vì vậy có thể đo độ tương tự giữa các văn bản hoặc giữa một văn bản với các văn bản khác trong kho dữ liệu thường dựa vào bảng tần số từ. Trong khai phá văn bản có nhiều độ đo khác nhau để tính toán mức độ tương tự của các văn bản, trong đó độ đo thường được sử dụng nhất là độ đo Cosine.

Trong phạm vi đặt ra, luận án tập trung vào các phương pháp đo độ tương tự văn bản mà không xét đến yếu tố ngữ nghĩa của văn bản. Luận án tính toán độ tương tự văn bản dựa trên các phương pháp liên quan đến chuỗi.

4) So khớp văn bản

Khái niệm "So khớp" được hiểu theo nhiều cách khác nhau. Dù có những khác biệt, chúng đều có điểm chung là để so sánh hai hay nhiều đối tượng để đánh giá mức độ giống nhau giữa chúng. So khớp văn bản (*Text alignment*) là so sánh sự giống nhau giữa các phân đoạn văn bản để tìm ra độ tương tự của văn bản [101, 124].

5) Đạo văn

Đạo văn được xem là hành vi thiếu trung thực và vi phạm các chuẩn mực đạo đức. Đó là hành động chiếm hữu một cách sai trái, công bố văn bản dựa trên ý tưởng, nội dung hay cách diễn đạt của người khác và xem chúng như là những gì do mình tự tạo ra [7].

Theo Meuschke và Gipp [80], đạo văn là việc sử dụng các ý tưởng của người khác mà không đưa ra lời xác nhận và trích dẫn tài liệu tham khảo phù hợp; trình bày ý tưởng hay lời nói của người khác như là của riêng mình; đó là trộm cắp, ăn cắp ý tưởng hay lời nói từ những người khác.

Tóm lại, đạo văn là một phạm trù rộng lớn, mang hàm ý "ăn cắp", "chiếm đoạt", "không trung thực", "dối trá",... Trong luận án sẽ không đi sâu phân tích về vấn đề "đạo văn" mà chỉ tập trung vào việc đánh giá mức độ giống nhau của văn bản nhằm hỗ trợ để phát hiện sao chép, việc xem xét kết luận "đạo văn" còn phụ thuộc vào nhiều yếu tố khác.

6) Phát hiện sao chép

Phát hiện sao chép (*Copy detection*) là tìm ra các đoạn văn bản giống nhau của một văn bản cần kiểm tra so với các văn bản khác đã có [49]. Trong luận án sẽ tập trung nghiên cứu, đề xuất các phương pháp để phát hiện các văn bản sao chép mà không phân tích các trường hợp đó có đạo văn hay trích dẫn hợp lệ hay không...

7) Kho ngữ liệu

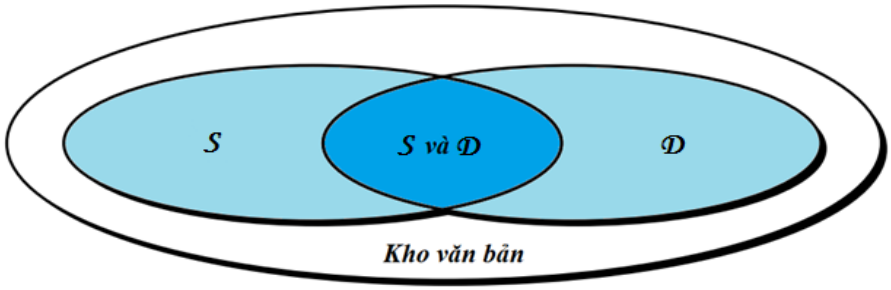
Kho ngữ liệu (*Corpus*) là kho dữ liệu hay kho sưu tập các tài liệu đã qua xử lý để phục vụ một mục đích nào đó... Ví dụ như: Kho ngữ liệu song ngữ Anh - Việt, kho ngữ liệu đơn ngữ tiếng Việt, kho ngữ liệu quốc gia Anh quốc (*British National Corpus* - BNC)... Một kho ngữ liệu là một tập hợp đủ lớn các văn bản có cấu trúc đã qua xử lý và thường được lưu trữ ở dạng điện tử. Các kho ngữ liệu là một tài nguyên quan trọng trong NLP. Dựa trên kho ngữ liệu, chúng ta có thể tính toán được tần suất, xác suất xuất hiện của các từ, tính toán ngữ nghĩa,... Tùy thuộc vào mục đích sử dụng, ta có thể xây dựng kho ngữ liệu thành các loại như: kho ngữ liệu thô (*Raw corpus*) là tập hợp các dữ liệu chỉ qua bước tiền xử lý; kho ngữ liệu được gán nhãn (*Tagged corpus*) là các dữ liệu trong kho ngữ liệu đã được xử lý như phân tích từ, phân tích cú pháp, gắn nhãn từ loại, thực thể có tên,... Để đảm bảo việc thống kê chính xác đòi hỏi các kho ngữ liệu phải có độ bao phủ, độ lớn và có tính đại diện cao [8].

Kho ngữ liệu thường gắn với yếu tố ngữ nghĩa, tuy nhiên trong luận án không tập trung nghiên cứu về vấn đề ngữ nghĩa nên chỉ đề cập đến "kho dữ liệu" hay bộ sưu tập tài liệu văn bản mà không liên quan đến ngữ nghĩa của văn bản.

8) Độ đo tính toán hiệu năng

Để kiểm tra kết quả so sánh văn bản, chúng ta thường sử dụng các phép đo tính toán hiệu năng như: Precision, Recall, F-score,...

Giả sử, ta gọi $\mathcal{S}$ và $\mathcal{D}$ lần lượt là các chuỗi văn bản nguồn bị sao chép và các chuỗi văn bản đánh giá. Mối quan hệ giữa các tập chuỗi văn bản nguồn bị sao chép và các chuỗi văn bản đánh giá được phát hiện là giống với các chuỗi trong văn bản nguồn được thể hiện như hình sau:



Hình 1.1. Mối quan hệ giữa $\mathcal{S}$ và $\mathcal{D}$

Như vậy, các chuỗi văn bản chung của văn bản nguồn và văn bản đánh giá được phát hiện là giống với các chuỗi trong văn bản nguồn sẽ là $\mathcal{S} \cap \mathcal{D}$ các độ đo cơ bản để đánh giá hiệu năng được tính theo các công thức sau [79]:

- Precision (*prec*): Là tỷ lệ các chuỗi văn bản được trả về thực sự có liên quan đến các chuỗi văn bản nguồn bị sao chép.

$$prec = \frac{|\mathcal{S} \cap \mathcal{D}|}{|\mathcal{D}|} \tag{1.1}$$

- Recall (*rec*): Là tỷ lệ các chuỗi văn bản có liên quan đến các chuỗi văn bản nguồn bị sao chép và các chuỗi văn bản đánh giá được phát hiện là giống với các chuỗi trong văn bản nguồn.

$$rec = \frac{|\mathcal{S} \cap \mathcal{D}|}{|\mathcal{S}|} \tag{1.2}$$

- Để cân đối giữa hai độ đo Precision và Recall, có một độ đo khác cũng thường được sử dụng là F-score (*Fscore*) được xây dựng dựa trên hai độ đo Precision và Recall.

$$Fscore = \frac{rec \times prec}{(rec + prec)/2} \tag{1.3}$$

1.2. Một số đặc điểm của ngôn ngữ tiếng Việt

1.2.1. Khái quát

Tiếng Việt là một trong những ngôn ngữ khó nhất thế giới. Tiếng Việt dựa trên bảng kí tự Latinh, cùng với các thanh điệu. Tiếng Việt là loại hình ngôn ngữ đơn lập, mỗi âm tiết còn được gọi là "tiếng" được phát âm tách rời nhau và được thể hiện bằng

một chữ viết, mỗi "từ" có thể được cấu thành bởi một hoặc nhiều tiếng. Từ là đơn vị nhỏ nhất có nghĩa và hoạt động tự do trong câu.

Chữ viết tiếng Việt có một đặc điểm đặc biệt đó là sự xuất hiện của các dấu thanh và dấu của các ký tự. Tiếng Việt có thanh điệu và trong văn bản, dấu của ký tự này làm phong phú và tăng thêm độ biểu cảm cũng như độ phức tạp của tiếng Việt.

Từ trong tiếng Việt được cấu thành bởi một hay nhiều tiếng tổ hợp lại. Từ ở dạng nguyên thể, hình thức và ý nghĩa độc lập với cú pháp [31]. Có các loại từ như: *Từ đơn* là những từ được cấu thành bằng một tiếng (gọi là *âm tiết*); *từ láy* là từ tạo nên từ hai tiếng trở lên, tiếng đứng ở vị trí đầu tiên là tiếng gốc (thường có nghĩa) và tiếng đứng sau láy lại âm hoặc vần của tiếng gốc; *từ ghép* là từ được cấu thành từ hai hay nhiều tiếng, mà những tiếng đó có quan hệ ngữ nghĩa với nhau; *từ đồng nghĩa* là những từ tương đồng với nhau về nghĩa nhưng khác nhau về mặt chữ viết; *từ đồng âm khác nghĩa* là hiện tượng giống nhau về ngữ âm nhưng khác nhau về ý nghĩa của từ.

Để hiểu một cách đơn giản thì tiếng là đơn vị cơ sở cấu tạo ngữ pháp tiếng Việt, còn từ là trung tâm của ngữ pháp tiếng Việt và mỗi câu mang một ý nghĩa thông tin hoàn chỉnh. Vì vậy, việc tách câu và tách từ trong xử lý văn bản là nhiệm vụ rất quan trọng để giải quyết các bài toán xử lý tiếng Việt.

Hiện nay có khá nhiều cách mã hóa các ký tự tiếng Việt khác nhau, dẫn tới có nhiều bảng mã khác nhau được sử dụng khi trình bày văn bản. Có thể liệt kê một số bảng mã thông dụng như: Unicode là bảng mã chuẩn hóa để biểu diễn hầu hết các văn bản trong các hệ thống chữ viết trên thế giới, mã UTF-8 được thiết kế để tương thích với chuẩn ASCII [129]. Và một số bảng mã tiếng Việt khác như: TCVN (ABC), VietWare-X, VNI for Windows,... Như vậy, nội dung văn bản tiếng Việt có thể được tạo ra bằng nhiều bộ gõ và bảng mã khác nhau nên việc xử lý, chuẩn hóa cũng phức tạp hơn so với tiếng Anh.

1.2.2. Một số khó khăn và nhập nhằng trong xử lý văn bản tiếng Việt

Khi nghiên cứu về văn bản tiếng Việt thường gặp rất nhiều khó khăn trong việc số hóa, biểu diễn phục vụ cho việc khai phá nguồn dữ liệu văn bản. Một số khó khăn và nhập nhằng trong xử lý văn bản tiếng Việt như:

- Nguồn dữ liệu: Cần thu thập nguồn dữ liệu có tính bao phủ, quy mô, phù hợp với yêu cầu đặt ra.

- Về ngữ pháp tiếng Việt: Để phân tích và kiểm tra tính chính xác của một văn bản tiếng Việt, chúng ta phải tiến hành phân tích từ vựng, phân tích cú pháp, phân tích ngữ nghĩa,...

- Đặc điểm chính tả tiếng Việt cũng là một yếu tố quan trọng làm ảnh hưởng đến việc xử lý dữ liệu văn bản như: Xử lý các tiếng đồng âm, các từ địa phương, cách gõ, vị trí dấu, ký tự ngắt câu,…

- Vấn đề đa nghĩa và nhập nhằng trong ngôn ngữ, nhập nhằng cú pháp sẽ gây khó khăn trong quá trình phân tích và xử lý.

- Ngữ nghĩa tiếng Việt rất phức tạp, mang tính nhập nhằng cao nên rất khó để áp dụng vào hệ thống so sánh văn bản mang yếu tố ngữ nghĩa. Bên cạnh đó, kho ngữ liệu văn bản tiếng Việt phục vụ cho việc nghiên cứu, ứng dụng trong lĩnh vực NLP để giải quyết vấn đề ngữ nghĩa còn rất nhiều hạn chế so với ngôn ngữ tiếng Anh và các ngôn ngữ khác.

- Quy tắc bỏ dấu: Có nhiều quy tắt bỏ dấu khác nhau trong tiếng Việt nên để loại bỏ nhập nhằng này cần chuẩn hóa về một chuẩn bỏ dấu nào đó. Tuy nhiên hiện nay, văn bản tiếng Việt được lưu trữ rất lớn và trong văn bản hình thức bỏ dấu cũng có nhiều kiểu nên sẽ rất phức tạp khi chuẩn hóa dấu.

Bên cạnh đó, các phương pháp, kỹ thuật, công cụ giải quyết bài toán tách đoạn, tách câu, tách từ cũng không thể đạt kết quả chính xác tuyệt đối, đặc biệt là đối với xử lý văn bản tiếng Việt có nhiều nhập nhằng so với tiếng Anh và các ngôn ngữ khác.

- Bài toán tách đoạn, tách câu thường không phân biệt ngôn ngữ vì hầu hết các ngôn ngữ khi phân tích để tách đoạn thường dựa trên ký tự xuống dòng; tách câu thường dựa trên các dấu câu như dấu chấm (.), dấu chấm than (!), dấu chấm hỏi (?) và dấu chấm phẩy (;),… Tuy nhiên, để có kết quả chính xác cao thì phụ thuộc rất lớn vào nguồn văn bản, nội dung văn bản.

- Với bài toán tách từ trong tiếng Việt, không sử dụng dấu cách (space) để làm ranh giới phân tách từ, nó chỉ có ý nghĩa phân tách các từ đơn (âm tiết) với nhau. Vì vậy, để xử lý văn bản tiếng Việt, công đoạn tách từ là một trong những bài toán cơ

bản, quan trọng và tạo tiền đề để giải quyết các bài toán về NLP khác như: Phân loại văn bản, tóm tắt văn bản, máy dịch tự động, hiểu văn bản, khai thác văn bản,…

Hiện nay, có nhiều công trình nghiên cứu để xử lý tách từ tiếng Việt và đạt được những kết quả chính xác cao như bộ công cụ tách câu, tách từ, gán nhãn từ loại JvnTextPro[3] của nhóm tác giả Nguyễn Cẩm Tú, công cụ tách từ vnTokenizer[4] của tác giả Lê Hồng Phương,… Tuy nhiên, mục tiêu của luận án không đặt ra vấn đề giải quyết về mặt ngữ nghĩa mà hướng đến việc đề xuất phương pháp mới để mang lại hiệu quả khi so sánh văn bản và tính đến xử lý dữ liệu lớn. Vì vậy, trong luận án không sử dụng các công cụ tách câu, tách từ tiếng Việt hiện có mà sử dụng phương pháp tách từ đơn để xử lý cho phù hợp với các đề xuất của luận án.

Tóm lại, với những khó khăn như đề cập ở trên, so với các ngôn ngữ khác thì xử lý văn bản tiếng Việt sẽ gặp những khó khăn và phức tạp hơn do đặc thù riêng của ngôn ngữ tiếng Việt. Vì vậy, khó có thể áp dụng một mô hình hoàn chỉnh từ các ngôn ngữ khác cho tiếng Việt. Trong hướng nghiên cứu của mình, luận án sẽ hạn chế việc xử lý những nhập nhằng về ngôn ngữ, ngữ nghĩa tiếng Việt,… mà chủ yếu tập trung vào nghiên cứu đề xuất các giải thuật xử lý, phương pháp so sánh và hiệu năng tính toán để xử lý văn bản tiếng Việt một cách hiệu quả.

1.3. Mô hình biểu diễn văn bản

1.3.1. Giới thiệu

Văn bản dưới dạng dữ liệu phi cấu trúc thường chiếm khoảng 80% [93], được lưu trữ ở các hình thức khác nhau của tài liệu như các báo cáo, bài báo, luận văn và trên các website là chủ yếu. Do vậy muốn xử lý chúng, trước hết phải biểu diễn chúng dưới dạng có cấu trúc theo mô hình biểu diễn văn bản phù hợp. Bên cạnh đó, việc xác định mối liên quan và thực hiện các phép biến đổi, ánh xạ văn bản cũng đòi hỏi

[3] http://jvntextpro.sourceforge.net/
[4] https://github.com/phuonglh/vn.vitk

những dạng này phải có khả năng thao tác bằng những phép toán cơ bản (cộng, nhân, đại số quan hệ,...) và các phép toán phức tạp khác.

Biểu diễn văn bản là một bước tiền xử lý rất quan trọng trong nhiều lĩnh vực như khai phá văn bản, truy vấn thông tin, NLP. Mô hình biểu diễn văn bản truyền thống như: Mô hình túi từ và không gian vector là các mô hình được sử dụng phổ biến nhất [5]. Mô hình không gian vector biểu diễn văn bản như một vector đặc trưng của các thuật ngữ (từ, cụm từ) xuất hiện trong toàn bộ tập văn bản. Trọng số các đặc trưng thường được tính qua độ đo TF, IDF, TF-IDF [111].

Trong xử lý văn bản có rất nhiều phương pháp có cách tính toán khác nhau, nhưng nhìn một cách tổng quan thì các phương pháp đó thường không tương tác trực tiếp trên tập dữ liệu thô ban đầu, mà thường thực hiện các bước chung như sau:

Hình 1.2. Quá trình mô hình hóa văn bản

- *Tiền xử lý*: Văn bản trước khi được mô hình hóa, cần phải được tiền xử lý (tách câu, tách từ, xử lý chữ viết hoa/chữ thường, loại bỏ từ dừng...). Quá trình tiền xử lý sẽ giúp nâng cao hiệu suất và giảm độ phức tạp của thuật toán.
- *Mô hình văn bản*: Là bước chọn mô hình biểu diễn văn bản phù hợp để đem lại hiệu quả trong tính toán, xử lý. Nói cách khác, một trong những nhiệm vụ đầu tiên trong việc xử lý văn bản là chọn được một mô hình biểu diễn văn bản thích hợp. Một văn bản ở dạng thô (dạng chuỗi) cần được chuyển sang một mô hình khác để tạo thuận lợi cho việc biểu diễn và tính toán. Tùy thuộc vào từng thuật toán xử lý khác nhau để có thể lựa chọn mô hình biểu diễn riêng.

1.3.2. Mô hình biểu diễn văn bản

Hiện nay, trên thế giới có nhiều nghiên cứu về các mô hình biểu diễn văn bản, chúng tương đối đa dạng và mỗi mô hình mang nét đặc trưng riêng.

Có thể chia thành hai hướng tiếp cận chính cho việc biểu diễn văn bản, đó là: Hướng thống kê và hướng ngữ nghĩa. Trong tiếp cận theo hướng thống kê, các văn bản được biểu diễn theo một số tiêu chí phục vụ đo lường dựa trên thống kê, trong

khi các phương pháp tiếp cận theo hướng ngữ nghĩa liên quan đến khái niệm và thực hiện việc phân tích cú pháp và ngữ nghĩa. Sau đây là các mô hình và phương pháp biểu diễn văn bản:

- ***Hướng tiếp cận thống kê*:** Một số mô hình nổi tiếng được nghiên cứu theo hướng tiếp cận thống kê thuần túy có thể kể đến là mô hình Boolean, mô hình không gian vector, mô hình xác suất và mô hình túi từ. Ý tưởng chính theo hướng tiếp cận này là biểu diễn nội dung của văn bản bằng danh sách các từ hay thuật ngữ (term hay chunk) xuất hiện trong văn bản.

- ***Hướng ngữ nghĩa*:** Thường dựa trên phân tích cấu trúc ngữ pháp, sử dụng từ điển ngữ nghĩa như: WordNet, Brown, Penn TreeBank, Wikipedia,... hay dựa trên các Ontology, mô hình đồ thị, mô hình khái niệm, phân tích từ loại, trật tự từ... Các phương pháp này thường xử lý chậm, tốn nhiều chi phí hơn hướng tiếp cận thống kê tuy nhiên về mặt ngữ nghĩa thì cho kết quả chính xác hơn.

Biểu diễn văn bản là bước xử lý không thể thiếu trong các bài toán về xử lý văn bản. Luận án đã khảo sát và trình bày những nội dung cơ bản cũng như những nhận xét, đánh giá về các mô hình biểu diễn văn bản, cụ thể như sau:

1.3.2.1. Mô hình Boolean

Giả sử có một tập D gồm *m* văn bản $D=\{d_1, d_2, ..., d_m\}$, với d_i là văn bản thứ *i*. Mỗi văn bản gồm *n* từ khóa $T=\{t_1, t_2, ..., t_n\}$. Gọi $W=\{w_{ij}\}$ là ma trận trọng số, trong đó w_{ij} là trọng số của từ khóa t_i trong văn bản d_j.

Mô hình Boolean là mô hình đơn giản nhất, trong đó trọng số các từ trong văn bản là 0 hoặc 1. Khi đó, mỗi văn bản sẽ được biểu diễn dưới dạng tập hợp $d_i = \{t_{ij}\}$, trong đó t_{ij} là từ t_i có trọng số w_{ij} trong văn bản d_j [112]. Trọng số được tính theo công thức đơn giản như sau:

$$w_{ij} = \begin{cases} 1 & t_i \in d_j \\ 0 & t_i \notin d_j \end{cases} \qquad (1.4)$$

- Ưu điểm:
  + Mô hình lý thuyết chặt chẽ, rõ ràng.

+ Đơn giản, dễ hiểu, dễ cài đặt và sử dụng.

+ Trả về kết quả chứa chính xác các từ khóa sau khi so sánh giống nhau.

- Nhược điểm:

+ Mô hình Boolean kiểm tra sự xuất hiện của một từ khóa biểu diễn trong một văn bản hoặc là có hoặc không, vì vậy việc so sánh theo Boolean hoặc là đúng hoặc sai. Do đó, hiệu quả truy vấn hay so sánh không cao.

+ Không thể xếp hạng kết quả trả về nên không xác định được mức độ giống nhau giữa các văn bản được so sánh.

+ Mô hình này chưa xét đến tần suất xuất hiện của các từ khóa trong văn bản và các mối quan hệ ngữ nghĩa như: hình thái của từ, thứ tự của các từ hay vị trí xuất hiện của từ trong tài liệu nên không thể so sánh sự tương tự mang yếu tố ngữ nghĩa của văn bản.

1.3.2.2. Mô hình không gian vector

Trong mô hình này, các văn bản được thể hiện trong một không gian có số chiều lớn, trong đó mỗi chiều của không gian tương ứng với một từ khoá (term) trong văn bản [114].

Theo cách biểu diễn của mô hình này, mỗi văn bản (tài liệu) d_i được biểu diễn dưới dạng vector đặc trưng cho văn bản là $\mathbf{d}_i = [t_1, t_2, ..., t_n]$, với *n* là số lượng từ khóa riêng biệt (từ vựng), tức là số chiều của vector $\mathbf{d}_i$; t_k là trọng số của từ thứ *k* với (*k* = 1…*n*). Gọi s_j là văn bản thứ *j* trong tập các văn bản trong tập dữ liệu nguồn, thì vector đặc trưng của văn bản $\mathbf{s}_j = [s_{1j}, s_{2j}, ..., s_{nj}]$, và s_{ij} là trọng số từ khóa thứ *i* của vector đặc trưng $\mathbf{s}_j$.

Như vậy, chúng ta cần quan tâm đến cách đánh trọng số cho ma trận. Theo mô hình Boolean, cách gán giá trị là 0 hay 1 là tùy vào sự xuất hiện của từ khóa trong tài liệu. Theo mô hình vector, ta có thể gán các giá trị tần suất xuất hiện của các từ khóa trong tài liệu. Việc xác định trọng số của từ khóa theo giá trị tần suất xuất hiện phụ thuộc vào các phương pháp khác nhau, thông thường dựa vào các kỹ thuật đánh trọng số phổ biến như TF, IDF, TF-IDF. Đây là mô hình được dùng phổ biến nhất trong xử lý văn bản [111].

Với mô hình không gian vector chuẩn như trên thì số chiều của các vector đặc trưng sẽ rất lớn (bằng với số từ khóa có trong tất cả các tài liệu) nên dẫn tới việc lưu trữ sẽ chiếm nhiều không gian và lãng phí tài nguyên. Trong một tài liệu, các từ khóa không xuất hiện trong tài liệu được gán chỉ số bằng 0 và như vậy vector biểu diễn tài liệu sẽ có vô số giá trị bằng 0.

Mô hình vector thưa sẽ khắc phục hạn chế trên. Tức là, một vector biểu diễn văn bản gồm các cặp *chỉ số - trọng số* của các từ khóa xuất hiện trong văn bản đó (trọng số trong trường hợp này luôn khác 0). Với phương pháp này, tuy ta tốn thêm tài nguyên lưu trữ chỉ số của từ khóa nhưng chỉ lưu trữ số lượng từ khóa ít hơn rất nhiều. Bản chất của mô hình này đơn giản là: Mỗi cặp *i* - $\mathbf{d}_i$ đại diện cho một vector chuẩn có trọng số của từ khóa thứ *i* là $\mathbf{d}_i$ và tất cả các trọng số thành phần còn lại là 0. Vector biểu diễn văn bản là tổng của các vector trên. Như vậy, mỗi vector đặc trưng cho văn bản chỉ còn lưu trữ các trọng số của các từ khóa có trong văn bản đó. Mô hình biểu diễn này hoàn toàn có thể chuyển đổi qua lại với mô hình không gian vector chuẩn một cách nhanh chóng và không làm ảnh hưởng đến việc tính toán độ tương tự giữa các văn bản.

Như vậy, thay vì biểu diễn vector văn bản dạng chuẩn, chúng ta còn có thể biểu diễn chúng thành tuyến tính các vector thành phần. Các vector này có thể là vector đơn vị, vector trực giao, vector trực chuẩn,... Tùy thuộc vào cách chọn các vector thành phần sẽ đem lại cho chúng ta các mô hình không gian vector tiên tiến hơn nhằm đem lại hiệu quả cho các bước xử lý văn bản phù hợp với bài toán đặt ra.

- Ưu điểm:

  + Việc đánh trọng số các từ khóa sẽ làm tăng hiệu suất tính toán.

  + Biểu diễn theo mô hình vector thích hợp để tính độ tương tự giữa các văn bản và dễ dàng xếp hạng các tài liệu trả về sau khi tính toán và tìm kiếm.

- Nhược điểm:

  + Mỗi từ khóa khác nhau được biểu diễn riêng nên số chiều của vector đặc trưng lớn.

  + Số chiều biểu diễn cho tập văn bản có thể rất lớn nên tốn không gian lưu trữ và làm giảm tốc độ tính toán.

1.3.2.3. Mô hình túi từ

Theo mô hình này, văn bản là tập hợp các từ được dùng trong văn bản nên được gọi là mô hình túi từ (bag of words). Mô hình này không phụ thuộc vào trật tự của từ, cấu trúc, ngữ pháp,... nhưng tính xác suất số lần mỗi từ xuất hiện trong văn bản nên đây cũng là dạng của mô hình xác suất. Trong mô hình túi từ, một văn bản sẽ được biểu diễn bởi một vector, với trọng số của mỗi thành phần của vector sẽ có giá trị là 0 nếu như từ đó không xuất hiện trong văn bản và ngược lại thì sẽ là số lần mà từ đó xuất hiện trong văn bản [25].

- Ưu điểm:

+ Văn bản được sắp xếp dựa vào xác suất liên quan đến tài liệu truy vấn.

+ Mô hình xác suất đạt chất lượng về hiệu năng cao hơn trong tìm kiếm so với các mô hình không áp dụng phương pháp xác suất.

- Nhược điểm:

+ Mô hình túi từ đối với những tập ngữ liệu lớn thì kích thước vốn từ sẽ lên đến hàng trăm nghìn và đôi khi lên đến cả triệu từ, gấp nhiều lần số lượng từ phân biệt của một ngôn ngữ.

+ Một trong những hạn chế lớn của mô hình này là giả định các từ độc lập với nhau, nghĩa là các mối tương quan ngữ nghĩa của các từ này không được xét đến và do đó không thể so sánh giữa những từ đồng nghĩa.

1.3.2.4. Mô hình Latent Semantic Indexing

Khác với các mô hình vector thường dựa trên các thuật ngữ hay từ khóa, *Latent Semantic Indexing* (LSI) là mô hình tạo chỉ mục ngữ nghĩa tiềm ẩn dựa trên khái niệm. Mô hình LSI khắc phục được hai hạn chế tồn tại trong mô hình không gian vector cũng như các mô hình Boolean và túi từ, đó là tính đồng nghĩa và tính đa nghĩa [25, 45]. Với tính đồng nghĩa, nhiều từ có thể được sử dụng để biểu diễn một khái niệm, vì vậy với phương pháp so khớp chuỗi thì không phát hiện được sự giống nhau của các văn bản có sử dụng từ đồng nghĩa. Tương tự, với tính đa nghĩa, một từ có thể có nhiều nghĩa, vì vậy phương pháp so khớp chuỗi có thể phát hiện những văn bản giống nhau nhưng nội dung lại khác nhau nên sẽ cho kết quả sai. Trong thực tế, trong kho dữ liệu văn bản thì các văn bản được viết bởi rất nhiều tác giả, với cách dùng từ

và phong cách viết rất khác nhau và đặc biệt là sự đa dạng, nhập nhằng của văn bản tiếng Việt nên xét về mặt ngữ nghĩa sẽ rất phức tạp.

Mô hình LSI dựa trên giả thiết là có quan hệ ngữ nghĩa tiềm ẩn trong việc sử dụng từ như có nhiều từ biểu diễn cho một khái niệm và một khái niệm có thể được biểu diễn bởi nhiều từ. Và mô hình này sử dụng phân tích giá trị đơn (*Singular Value Decomposition* - SVD) để giảm kích thước của ma trận trọng số từ/tài liệu (term/document) để từ đó phát hiện ra các quan hệ ngữ nghĩa tiềm ẩn của các từ trong toàn bộ văn bản.

- Ưu điểm:

+ Mô hình này áp dụng các mối quan hệ giữa các khái niệm nên đạt được hiệu quả về mặt ngữ nghĩa.

+ Kỹ thuật LSI giảm kích thước của ma trận, rút gọn được tập đặc trưng nên giảm số chiều không gian biểu diễn.

- Nhược điểm:

+ Việc giảm chiều ma trận sẽ khó cài đặt thuật toán hơn các mô hình biểu diễn vector.

+ Cách biểu diễn bằng khái niệm thay cho từ khóa cũng phức tạp hơn, tốn chi phí trong việc xây dựng các quan hệ khái niệm.

1.3.2.5. Dựa trên khái niệm mờ

Trong các mô hình vector, thông thường việc biểu diễn văn bản với các thành phần là các từ khóa. Theo mô hình biểu diễn dựa trên khái niệm mờ (fuzzy) thì các thành phần của văn bản là các khái niệm mờ. Trọng số của các khái niệm mờ phụ thuộc vào các từ khóa liên quan đến văn bản, số lượng từ khóa liên quan càng nhiều thì trọng số của khái niệm càng lớn, điều này giúp cho việc xác định ngữ nghĩa được rõ ràng hơn. Mô hình này có thể giải quyết vấn đề về từ đồng nghĩa. Với tập hợp các từ đồng nghĩa xuất hiện trong văn bản, chúng ta có thể tích hợp lại trong một khái niệm chung bằng cách sử dụng hàm tích hợp mờ. Việc xử lý văn bản sẽ được tính toán trên một khái niệm chung này thay vì phải tính toán trên các từ khóa. Với cách biểu diễn này sẽ làm giảm bớt số chiều của vector và phát hiện xử lý được các từ đồng nghĩa trong văn bản [13, 139].

- Ưu điểm:

+ Mô hình này có thể tính toán, xử lý tốt vấn đề từ đồng nghĩa trong văn bản.

+ Giảm được số chiều của vector biểu diễn và giảm sự phức tạp trong tính toán.

- Nhược điểm:

+ Việc xây dựng các khái niệm mờ, các hàm tích hợp làm tốn nhiều công sức và chi phí.

+ Rất khó triển khai trong thực tế và đặc biệt đối với xử lý tiếng Việt.

1.3.2.6. Mô hình đồ thị

Trong các mô hình vector thường bỏ qua các thông tin về cấu trúc, trật tự từ, vị trí xuất hiện của từ trong văn bản. Để giải quyết các hạn chế trên, mô hình đồ thị với việc tận dụng được các thông tin quan trọng về cấu trúc văn bản nên được đánh giá có nhiều tiềm năng để giải quyết các bài toán về xử lý văn bản.

Mô hình đồ thị khái niệm (*Conceptual Graphs* - CGs) được ra đời vào cuối những năm 1970 có thể dùng để biểu diễn văn bản bằng cách định nghĩa các khái niệm và các mối quan hệ [120]. Hiện nay, mô hình đồ thị đã không ngừng phát triển dựa trên ý tưởng của mô hình đồ thị khái niệm, được ứng dụng để giải quyết các bài toán về xử lý văn bản. Có rất nhiều loại đồ thị khác như: mô hình đồ thị hình sao, đồ thị khái niệm sử dụng từ điển ngữ nghĩa, đồ thị vô hướng, đồ thị có hướng...

Mô hình biểu diễn đồ thị thông dụng trong xử lý văn bản thường được biểu diễn thông qua các đỉnh và cạnh của đồ thị. Đỉnh thường là các thành phần thích hợp nhất trong văn bản và cạnh biểu diễn mối quan hệ hiệu quả nhất giữa các đỉnh. Ví dụ, đỉnh của đồ thị có thể biểu diễn các từ khóa, thuật ngữ, hay câu...; cạnh của đồ thị có thể dùng để thể hiện những mối quan hệ khác nhau giữa các đỉnh như: trật tự của các từ, tần số xuất hiện của các từ, vị trí của các từ...[5].

- Ưu điểm:

+ Mô hình đồ thị biểu diễn văn bản một cách trực quan và chính xác.

+ Mô hình đồ thị hiệu quả trong việc lưu trữ các thông tin cấu trúc như vị trí, thứ tự xuất hiện và sự gần nhau của từ.

- Nhược điểm:

+ Xây dựng mô hình này khá phức tạp, đòi hỏi phải phân tích ngữ nghĩa sâu, chuyên biệt và phụ thuộc vào lĩnh vực cụ thể.

+ Với sự phức tạp và nhập nhằng của tiếng Việt thì mô hình này khó áp dụng.

+ Khi dữ liệu văn bản lớn thì mô hình này tạo nên không gian lưu trữ lớn.

1.3.2.7. Dựa trên mô hình n-gram

N-gram là mô hình xác suất cho việc dự đoán phần tử tiếp theo trong một chuỗi. Phương pháp biểu diễn văn bản theo mô hình n-gram có đơn vị là ký tự, từ,… Thông thường sử dụng đơn vị từ làm gram, trong đó có thể gồm 1 từ (unigram), 2 từ (bigrams), 3 từ (trigrams),... Việc sử dụng n-gram trong biểu diễn là để làm giàu thêm tính ngữ nghĩa của văn bản, tăng độ chính xác trong biểu diễn và có thể làm giảm không gian lưu trữ các từ khóa [17, 129].

- Ưu điểm:

+ Biểu diễn theo mô hình này có thể làm tăng thêm tính ngữ nghĩa và độ chính xác.

+ Khi biểu diễn văn bản theo mô hình không gian vector kết hợp các từ khóa theo n-gram từ sẽ làm giảm không gian lưu trữ và đem lại hiệu quả trong xử lý văn bản.

- Nhược điểm:

+ Khi xây dựng theo mô hình này, các cụm từ có thể được biểu diễn xếp chồng lên nhau nên số lượng n-gram lớn.

1.3.2.8. Phương pháp chiếu ngẫu nhiên

Đối với nhiều ứng dụng vấn đề trọng tâm của việc biểu diễn văn bản là giúp định nghĩa được khoảng cách giữa các văn bản. Tuy nhiên nếu biểu diễn trong không gian gốc với thành phần là các từ chỉ mục thì số chiều quá lớn, gây khó khăn trong tính toán. Biểu diễn của văn bản trong không gian dữ liệu có số chiều lớn đó sẽ được

chiếu lên một không gian có số chiều ít hơn (tức là làm giảm số chiều), sao cho những khoảng cách gốc được duy trì một cách gần đúng nhất. Kết quả là những vector cơ sở trực giao trong không gian gốc được thay thế bởi những vector có xác suất trực giao gần đúng. Một ví dụ về văn bản được tổ chức thành các từ vựng dựa trên những thông tin về ngữ cảnh mà chúng có khuynh hướng xuất hiện trong đó. Số chiều của các biểu diễn ngữ cảnh được giảm nhờ thay thế mỗi chiều của không gian gốc bằng một chiều ngẫu nhiên trong một không gian có số chiều ít hơn [108].

- Ưu điểm:

+ Thuận lợi của phép chiếu ngẫu nhiên là sự tính toán nhanh do số chiều trong không gian vector giảm.

+ Phương pháp này có thể áp dụng được cho biểu diễn các vector đặc trưng của văn bản với số chiều lớn, hay phương pháp này phù hợp với dữ liệu văn bản lớn.

- Nhược điểm:

+ Làm giảm số chiều của không gian gốc không hề dễ dàng nên việc triển khai thuật toán sẽ rất khó khăn.

+ Chiếu ngẫu nhiên làm cho vector biểu diễn văn bản bị thay đổi nên độ chính xác sẽ không cao.

1.3.2.9. Mô hình phân tích cú pháp

Các mô hình phân tích cú pháp được áp dụng nhiều trong NLP. Đối với tiếng Việt, có nghiên cứu về hệ phân tích cú pháp tiếng Việt theo hướng phân tích cú pháp dựa trên luật và theo thống kê [129]. Một ví dụ về mô hình này là mỗi văn bản được phân tích cú pháp và cho kết quả là thông tin chi tiết về chủ đề của văn bản đó. Sau đó, tiến hành lập chỉ mục cho các chủ đề của từng văn bản bằng cách lập chỉ mục trên các từ khóa xuất hiện trong chủ đề. Như vậy, các văn bản sẽ được quản lý thông qua các chủ đề này nên khi tìm kiếm sẽ dò tìm nhanh và chính xác.

- Ưu điểm:

+ Đối với những ngôn ngữ đơn giản về mặt ngữ pháp thì việc biểu diễn theo mô hình phân tích cú pháp có thể đạt được độ chính xác cao.

+ Tìm kiếm theo phương pháp này đơn giản và khá hiệu quả.

- Nhược điểm:

+ Chất lượng của hệ thống theo phương pháp này hoàn toàn phụ thuộc vào chất lượng của hệ thống phân tích cú pháp và đoán nhận nội dung tài liệu.

+ Trong thực tế, việc xây dựng hệ thống phân tích cú pháp là rất phức tạp, phụ thuộc vào đặc điểm của từng ngôn ngữ vì vậy rất khó áp dụng đối với văn bản tiếng Việt.

1.3.2.10. Biểu diễn Tensor

"Tensor" là một thuật ngữ trong đại số đa tuyến tính. Nó là sự khái quát hóa các khái niệm "vector" và "ma trận" trong lĩnh vực đại số tuyến tính. Thông thường, một cấu trúc dữ liệu biểu diễn dạng vector được gọi là tensor bậc 1 và một cấu trúc dữ liệu dạng ma trận được gọi là một tensor bậc 2 hay một cấu trúc dữ liệu dạng khối lập phương được gọi là một tensor bậc 3,... Như vậy, các tensor bậc cao hơn là các cấu trúc dữ liệu trừu tượng được khái quát hóa như dạng vector và ma trận. Biểu diễn văn bản theo cấu trúc tensor để mô tả các tài liệu văn bản và sử dụng các kỹ thuật của đại số đa tuyến tính để có thể làm tăng hiệu suất việc truy xuất thông tin [71].

- Ưu điểm:

+ Mô hình này xử lý được trên nhiều chiều và áp dụng hiệu quả trong các dữ liệu chuỗi thời gian thực.

+ Mô hình này có khả năng áp dụng để giải quyết các bài toán xử lý dữ liệu lớn.

- Nhược điểm:

+ Rất phức tạp và khó cài đặt thuật toán.

1.3.3. Nhận xét và đánh giá

Qua các mô hình biểu diễn văn bản đã được trình bày ở trên, chúng ta nhận thấy mô hình không gian vector khắc phục nhược điểm của mô hình Boolean với việc sử dụng trọng số cho từ khóa khác với trọng số nhị phân. Mô hình dựa trên tập mờ cũng xử lý tốt vấn đề đồng nghĩa và đa nghĩa, tuy nhiên rất khó để đề xuất thuật toán. Với mô hình biểu diễn văn bản thành đồ thị, mô hình phân tích cú pháp và mô hình

túi từ thì có nhiều nhược điểm và không phù hợp để giải quyết bài toán phát hiện sao chép văn bản. Kỹ thuật LSI được cải tiến từ mô hình vector và có xét đến vấn đề ngữ nghĩa nên có hiệu quả hơn một số phương pháp khác, tuy nhiên việc áp dụng kỹ thuật LSI cho xử lý tiếng Việt vẫn còn những hạn chế và không có tính khả thi do sự nhập nhằng, phức tạp của tiếng Việt. Trong khi đó, mô hình biểu diễn theo Tensor sẽ là một hướng phát triển để biểu diễn văn bản theo hướng xử lý dữ liệu lớn và tăng dần theo thời gian. Phương pháp biểu diễn theo vector kết hợp với phương pháp tính tần suất xuất hiện của từ hoặc cụm từ (n-gram từ) hiện được dùng phổ biến cho các bài toán xử lý văn bản, tuy có những ưu điểm nhất định, nhưng phương pháp này cũng còn những hạn chế về số chiều vector đặc trưng lớn làm tốn không gian lưu trữ nên khó áp dụng cho việc tính toán và mở rộng với dữ liệu lớn.

Với việc khảo sát và nghiên cứu ban đầu về cách biểu diễn dữ liệu văn bản, để áp dụng giải quyết bài toán so khớp văn bản nhằm đánh giá được mức độ giống nhau của văn bản tiếng Việt là vấn đề khó, đặc biệt là về ngữ nghĩa thì khó có thể giải quyết hay đánh giá đầy đủ về mức độ giống nhau trong văn bản tiếng Việt. Vì vậy, hướng nghiên cứu của luận án đặt ra là tập trung chính vào việc so khớp chuỗi ký tự để đạt độ chính xác cao và có thể tính toán mở rộng trên nguồn dữ liệu lớn, về ngữ nghĩa thì chủ yếu nghiên cứu triển khai thử nghiệm trên các bộ dữ liệu đơn giản để đánh giá hiệu quả của phương pháp.

Tóm lại, biểu diễn văn bản là bước đầu tiên cần thiết của quá trình xử lý văn bản, không tồn tại một mô hình biểu diễn văn bản tối ưu mà chỉ tồn tại các mô hình phổ biến, mỗi mô hình biểu diễn đều có những ưu và nhược điểm khác nhau. Chính vì vậy, tùy vào bài toán cụ thể chúng ta có thể lựa chọn mô hình biểu diễn, phương pháp tiếp cận phù hợp để giải quyết bài toán. Qua khảo sát và nghiên cứu, có thể thấy hầu hết các bài toán xử lý văn bản đều liên quan đến cách biểu diễn theo mô hình vector và có những cải tiến để ứng dụng giải quyết bài toán một cách phù hợp nhất. Luận án tập trung khảo sát, phân tích kỹ lưỡng hơn cũng như triển khai một số phương pháp dựa trên mô hình vector được đề cập ở trong Chương 2 và đề xuất giải pháp, hướng tiếp cận mới để xử lý số hóa văn bản và ứng dụng phép biến đổi Wavelet rời rạc và bộ lọc Haar kết hợp với biểu diễn dạng vector để giải quyết bài toán so khớp văn bản, nội dung này được trình bày chi tiết ở Chương 3.

1.4. Các phương pháp tính độ tương tự văn bản

1.4.1. Hướng tiếp cận

Tính độ tương tự văn bản để đánh giá sự giống nhau giữa các văn bản được ứng dụng vào nhiều mục đích khác nhau như: Phân loại văn bản, phát hiện sao chép, truy vấn thông tin, dịch máy, tìm kiếm... Đây không còn là vấn đề mới, đã có nhiều nghiên cứu được thực hiện, với nhiều giải pháp khác nhau được đưa ra.

Qua khảo sát có thể chia các nghiên cứu trong lĩnh vực này thành ba hướng tiếp cận chính.

Trong ba hướng tiếp cận thì phương pháp dựa trên tập dữ liệu và dựa trên tri thức sẽ xác định sự giống nhau về mặt ngữ nghĩa của từ. Phương pháp dựa trên chuỗi xác định sự giống nhau về mặt hình thức (từ, câu) [39, 75].

- **Phương pháp dựa trên chuỗi (*String-Based*)**

Phương pháp này thao tác trên chuỗi, xác định sự giống nhau giữa các chuỗi dựa vào thành phần cấu tạo nên chuỗi. Nó được tách thành hai phương pháp là dựa trên ký tự (character-based) và dựa trên từ (term-based). Có 14 thuật toán phổ biến dựa theo hướng tiếp cận này, trong đó có 7 thuật toán dựa trên ký tự và còn lại là dựa trên từ.

- **Phương pháp dựa trên tập dữ liệu (*Corpus-Based*)**

Phương pháp này xác định sự giống nhau giữa các từ về mặt ngữ nghĩa, dựa vào thông tin thu được từ các tập dữ liệu văn bản lớn.

- **Phương pháp dựa trên tri thức (*Knowledge-Based*)**

Phương pháp dựa trên tri thức xác định sự giống nhau giữa các từ về mặt ngữ nghĩa, thường dựa vào từ điển ngữ nghĩa. Từ điển ngữ nghĩa là một bộ sưu tập các danh từ, động từ, tính từ và trạng từ,… có mối quan hệ với nhau tạo thành các kho tri thức có sẵn như WordNet, Wikipedia,...

Phương pháp này chia làm hai nhóm là:

+ Đánh giá sự giống nhau về ngữ nghĩa (*measures of semantic similarity*).

+ Đánh giá sự liên quan về ngữ nghĩa (*measures of semantic relatedness*).

Bảng 1.1. Phương pháp và thuật toán đánh giá sự giống nhau của văn bản

TT	Phương pháp	Thuật toán
I	**Dựa trên chuỗi**	
I.1	*Dựa trên ký tự*	Chuỗi con chung dài nhất (LSC), Damerau-Levenshtein, Jaro, Jaro-Winkler, Needleman-Wunsch, Smith-Waterman, n-gram
I.2	*Dựa trên từ*	Khoảng cách Manhattan, Cosine Similarity, hệ số Dice, khoảng cách Euclid, Jaccard Similarity, hệ số Matching và hệ số Overlap
II	**Dựa trên tập ngữ liệu**	Hyperspace Analogue to Language (HAL), Latent Semantic Analysis (LSA), Generalized Latent Semantic Analysis (GLSA), Explicit Semantic Analysis (ESA), The Cross-Language Explicit Semantic Analysis (CL-ESA), Pointwise Mutual Information - Information Retrieval (PMI-IR), Second-order CO-occurrence Pointwise Mutual Information (SCO-PMI), Normalized Google Distance (NGD), Extracting DIStributionally similar words using CO-occurrences (DISCO)
III	**Dựa trên tri thức**	
III.1	*Đánh giá sự giống nhau về ngữ nghĩa*	Resnik (res), Lin (lin), Jiang & Conrath (jcn), Leacock & Chodorow (lch), Wu & Palmer (wup) và Path Length (path)
III.2	*Đánh giá sự liên quan về ngữ nghĩa*	St.Onge (hso), Lesk (lesk) và cặp vector (vector)

Với phạm vi của luận án, chúng tôi xác định tập trung nghiên cứu để đánh giá mức độ giống nhau của văn bản dựa trên chuỗi với các thuật toán so khớp chuỗi và các độ đo tương tự.

1.4.2. Bài toán so khớp chuỗi

So khớp chuỗi là một kỹ thuật đóng vai trò nền tảng trong lĩnh vực xử lý văn bản. Với bài toán so khớp văn bản, có thể giải quyết thông qua việc so khớp chuỗi. Bài toán so khớp chuỗi có phép toán cơ bản gọi là đối sánh mẫu (pattern matching), đó là việc so sánh một hoặc nhiều chuỗi (gọi là mẫu) với văn bản để tìm vị trí và số lần xuất hiện của chuỗi đó trong văn bản. Một số thuật toán điển hình để giải quyết bài toán so khớp chuỗi như: Brute-Force, Naïve, Morris-Pratt, Knuth-Morris-Pratt

(KMP), Boyer-Moore, Rabin-Karp, Horspool,... [27, 118, 133]. Những thuật toán này tập trung vào vấn đề so sánh hai chuỗi ký tự bất kỳ và phát hiện sự giống nhau giữa chúng. Đây là dạng bài toán khó và phức tạp, luôn được các nhà nghiên cứu quan tâm để làm sao có được kết quả chính xác và độ phức tạp thuật toán thấp.

Với một số trường hợp trong so khớp văn bản, việc đo độ tương tự giữa hai đoạn văn bản là việc sử dụng so khớp từ đơn giản và tạo ra một điểm tương tự trên số đơn vị từ vựng xảy ra ở cả hai đoạn văn bản đầu vào. Việc thu thập, tách câu, tách từ, xử lý chữ hoa chữ thường, loại bỏ các từ dừng, gán nhãn từ loại, gán các trọng số... đều có thể được tích hợp để mang lại hiệu quả cho phương pháp so khớp văn bản.

Các thuật toán so khớp chuỗi được phân loại như sau [81]:

- *Thuật toán cổ điển* (Classical Algorithms): Là các thuật toán chủ yếu dựa vào sự so sánh giữa các ký tự. Các thuật toán điển hình như: Brute-Force, Naïve,...
- *Thuật toán ô-tô-mát hậu tố* (Suffix Automaton Algorithms): Là các thuật toán sử dụng cấu trúc dữ liệu ô-tô-mát hậu tố để nhận ra tất cả các hậu tố của mẫu. Các thuật toán điển hình như Knuth-Morris-Pratt, Boyer-Moore, Horspool,...
- *Thuật toán băm* (Hashing Algorithms): Là các thuật toán sử dụng kỹ thuật băm, điển hình như thuật toán Rabin-Karp.

Độ phức tạp của thuật toán so khớp chuỗi: Trên thực tế có nhiều loại ký tự khác nhau như chuỗi nhị phân (Binary), bảng chữ cái (Alphabet), số (Numeric), chuỗi DNA,... và mỗi loại ký tự có độ phức tạp khác nhau. Độ phức tạp tính toán tỷ lệ thuận với chiều dài của mẫu, chiều dài của văn bản và độ lớn của tập các ký tự.

Các thuật toán so khớp chuỗi thường được thực hiện theo 2 bước xử lý sau:

- *Bước tiền xử lý*: Thường là bước xử lý mẫu và khởi tạo cấu trúc dữ liệu.
- *Bước so khớp*: Thực hiện so khớp mẫu trong văn bản.

Phát biểu bài toán so khớp chuỗi như sau: Cho trước một chuỗi văn bản *s* có độ dài *n* và một mẫu *p* có độ dài *m*, hãy tìm sự xuất hiện của mẫu *p* trong chuỗi văn bản *s*. Để tìm sự xuất hiện của tất cả các mẫu trong văn bản, thông thường thực hiện bằng cách quét qua toàn bộ văn bản một cách tuần tự. Việc đối sánh mẫu diễn ra với nhiều lần thử trên các đoạn khác nhau của văn bản. Trong đó, cửa sổ là một chuỗi *m*

ký tự liên tiếp trên văn bản. Qua mỗi lần thử sẽ kiểm tra sự giống nhau giữa mẫu với cửa sổ hiện thời. Tùy theo kết quả kiểm tra cửa sổ sẽ được dịch sang phải hoặc trái trên văn bản cho lần thử kế tiếp.

Việc tích hợp các thuật toán so khớp chuỗi là một trong những khâu cơ bản trong việc tìm kiếm, so sánh và đánh giá được mức độ giống nhau của văn bản. Qua quá trình khảo sát, trong luận án trình bày nội dung nghiên cứu các thuật toán so khớp chuỗi để có thể ứng dụng trong bài toán tính độ tương tự văn bản. Như giới thiệu ở phần trên, có một số thuật toán kinh điển nhằm giải quyết bài toán so khớp chuỗi như: Brute-Force, Naïve, Morris-Pratt, KMP, Boyer-Moore, Rabin-Karp, Horspool. Dưới đây là những phân tích, mô tả, so sánh và nhận xét về các thuật toán này.

1.4.2.1. Thuật toán Brute-Force

Thuật toán Brute-Force [27] là một thuật toán theo kiểu vét cạn. Bằng cách dịch chuyển biến đếm j từ trái qua phải lần lượt từng ký tự của tập tin văn bản có chiều dài n. Sau đó lấy m ký tự liên tiếp trong s (bắt đầu từ vị trí j) tạo thành một chuỗi phụ r. So sánh r với p, nếu giống nhau thì trả về kết quả. Thực hiện lại quá trình trên cho đến khi $j>n-m+1$. Thuật toán này không có bước tiền xử lý.

Xuất phát từ ý tưởng vét cạn nên Brute Force có một số đặc điểm là ít tốn không gian bộ nhớ, không sử dụng thêm mảng phụ để lưu trữ và dễ triển khai, nhưng có thể rất chậm so với các thuật toán khác.

Thuật toán Brute-Force so khớp tất cả các vị trí xuất hiện của đoạn mẫu trong văn bản nên trong trường hợp xấu nhất có độ phức tạp rất lớn. Độ phức tạp thực thi của thuật toán này là $O(mn)$.

1.4.2.2. Thuật toán Naïve

Đây là phương pháp đơn giản nhất, thực hiện lần lượt xét từng vị trí i trong xâu ký tự gốc từ 1 đến $n-m+1$, so sánh bằng cách xét từng cặp ký tự một và đưa ra kết quả so khớp. Phương pháp này còn được gọi là cách tiếp cận ngây thơ [33].

Độ phức tạp tốt nhất của thuật toán này là $O(m+n)$, nhưng trong trường hợp xấu nhất thuật toán Naïve có độ phức tạp là $O(mn)$.

1.4.2.3. Thuật toán Knuth-Morris-Pratt

Thuật toán Knuth-Morris-Pratt (hay thuật toán KMP) [54] là một thuật toán so khớp chuỗi tuyến tính để tìm kiếm sự xuất hiện của một "từ" trong một "chuỗi văn bản" bằng cách tiếp tục quá trình tìm kiếm khi không phù hợp, bỏ qua quá trình kiểm tra lại các ký tự đã so sánh trước đó.

Thuật toán Knuth-Morris-Pratt dựa trên thuật toán Brute-Force với ý tưởng sử dụng lại những thông tin của lần thử trước cho lần kiểm tra sau. Trong thuật toán Brute-Force vì chỉ dịch cửa sổ đi một ký tự nên có đến m-1 ký tự của cửa sổ mới là những ký tự của cửa sổ vừa xét. Trong đó, có thể có rất nhiều ký tự đã được so sánh giống với mẫu và hiện giờ lại nằm trên cửa sổ mới nhưng được dịch đi về vị trí so sánh với mẫu. Việc xử lý những ký tự này có thể được tính toán trước rồi lưu lại kết quả. Vì vậy, nhờ đó lần thử sau có thể dịch đi được nhiều hơn một ký tự và giảm số ký tự phải so sánh lại.

Thuật toán Knuth-Morris-Pratt ở giai đoạn tìm kiếm có độ phức tạp là O(n), thuật toán tạo mảng để cho phép thuật toán so sánh mỗi ký tự của chuỗi s không quá một lần có độ phức tạp là O(m). Đây là một thuật toán rất hiệu quả, có chi phí về thời gian là O(m+n) với nhiều nhất là 2n-1 số lần so sánh ký tự trong quá trình tìm kiếm.

1.4.2.4. Thuật toán Boyer-Moore

Khác với thuật toán Knuth-Morris-Pratt, thuật toán Boyer-Moore kiểm tra các ký tự của mẫu từ phải sang trái và khi phát hiện sự khác nhau đầu tiên thì thuật toán sẽ tiến hành dịch cửa sổ đi. Trong thuật toán này, với ý tưởng là giả sử có chuỗi s và chuỗi p, cần tìm p trong s; bắt đầu kiểm tra các ký tự của p và s từ phải sang trái và khi phát hiện sự khác nhau đầu tiên, thuật toán sẽ tiến hành dịch p qua phải để thực hiện so sánh tiếp [6, 27].

Thuật toán Boyer-Moore có độ phức tạp trong trường hợp tốt nhất là O(n/m). Tuy nhiên, vì cách dịch của thuật toán này không phân tích triệt để các thông tin của những lần thử trước, những đoạn đã so sánh rồi vẫn có thể bị so sánh lại nên độ phức tạp thực thi của thuật toán Boyer-Moore trong trường hợp này là O(mn).

1.4.2.5. Thuật toán Rabin-Karp

Thuật toán Rabin-Karp trong bài toán đối sánh mẫu không khác nhiều so với bài toán tìm kiếm thông thường. Thuật toán Rabin-Karp sử dụng hàm băm để chuyển đổi mỗi chuỗi thành một số nguyên và việc so sánh hai chuỗi được đưa về phép toán so sánh các số nguyên [40, 60].

Với mẫu $p[1...m]$ đã cho, gọi p là biểu diễn số tương ứng của nó. Tương tự như thế, với văn bản $T[1...n]$, ký hiệu t, là biểu diễn số của chuỗi con $T[s+1...s+m]$ có độ dài m, với $s= 0,1..., n-m$. Hiển nhiên, $t_s=p$ nếu và chỉ nếu $T[s+1...s+m]=p[1...m]$.

Trong quá trình tiền xử lý của thuật toán Rabin-Karp có độ phức tạp là O(m), tuy nhiên trong quá trình so sánh thì trường hợp tốt nhất sẽ có độ phức tạp là O(n-m). Nhưng trong trường hợp xấu nhất thì việc so sánh phải thực hiện thêm lệnh kiểm tra $P[1..m]$ và $T[i+1,i+m]$, điều này làm tiêu tốn O(m) thời gian thực thi. Vì vậy, độ phức tạp trong trường hợp xấu nhất của thuật toán Rabin-Karp là O((n-m)m).

1.4.2.6. Thuật toán Horspool

Thuật toán Horspool tương tự như thuật toán Knuth-Morris-Pratt nhưng đảo ngược thứ tự so sánh và là một biến thể đơn giản hóa của thuật toán Boyer-Moore [27].

Khi thực hiện thuật toán này có thể nhảy qua nhiều ký tự nên độ phức tạp về thời gian thực thi trung bình là O(n). Tuy nhiên, trong trường hợp xấu nhất thuật toán có độ phức tạp trong thực thi là O(mn).

Bảng 1.2. So sánh và đánh giá một số thuật toán so khớp chuỗi

Thuật toán	**Độ phức tạp**		**Đánh giá**
	Tiền xử lý	**So khớp**	
Brute-Force		O(mn)	Theo kiểu vét cạn nên độ phức tạp lớn.
Naïve		O(mn)	Tương tự như Brute-Force, thuật toán Naïve lần lượt xét từng vị trí trong xâu ký tự gốc nên số bước thực hiện lớn.
Knuth-Morris-Pratt	O(m)	O(m+n)	Dùng ít phép toán so sánh hơn Brute-Force, thực hiện tìm kiếm tuần tự trong văn bản và không yêu cầu phải sao lưu nên ít tiêu tốn bộ nhớ đệm hơn.

Boyer-Moore	O(*m*)	O(*mn*)	Sử dụng hai hàm dịch chuyển, từ phải sang trái, thuật toán cho kết quả tìm kiếm nhanh trên bảng chữ cái lớn và trên chuỗi nhị phân.
Rabin-Karp	O(*m*)	O((*n*-*m*)*m*)	So sánh dựa trên giá trị băm, tính toán nhanh, không phụ thuộc vào chiều dài chuỗi cần tìm kiếm. Tiết kiệm bộ nhớ đệm vì không lưu trữ lại nhiều kết quả tìm kiếm trước đó.
Horspool	O(*m*)	O(*mn*)	Cải tiến của thuật toán Boyer-Moore là chỉ dịch ký tự bên phải nhất của cửa sổ trượt nên các bước thực hiện ít hơn.

1.4.2.7. Nhận xét

Nghiên cứu các thuật toán so khớp chuỗi điển hình trên là cần thiết để có thể ứng dụng vào bài toán so sánh văn bản.

Trong thực tế, tùy thuộc vào mô hình, dữ liệu, phương pháp đề xuất, chúng ta có thể phát triển các thuật toán so khớp chuỗi trên cho phù hợp với yêu cầu.

Việc tìm hiểu, đánh giá được độ phức tạp của các thuật toán sẽ giúp cho việc sánh với phương pháp đề xuất được tốt hơn. Nghiên cứu các thuật toán so khớp chuỗi để làm nền tảng cho việc tính toán độ tương tự văn bản và so sánh hiệu quả của phương pháp đề xuất dựa trên độ phức tạp tính toán.

1.5. So sánh văn bản và ứng dụng trong phát hiện sao chép

1.5.1. Giới thiệu

Bài toán so sánh văn bản thực chất là tính toán được mức độ giống nhau hay độ tương tự của văn bản. Với mục đích nghiên cứu là đánh giá mức độ giống nhau của văn bản để ứng dụng trong phát hiện sao chép, luận án tập trung nghiên cứu theo hướng giải quyết bài toán so sánh văn bản theo dạng so khớp chuỗi mà không đi sâu về mặt ngữ nghĩa cũng như không đề cập sâu về các hình thức sao chép như kiểu sao chép cấu trúc và ý tưởng, tự sao chép, trích dẫn không phù hợp,...

Bài toán phát hiện các văn bản giống nhau là một bài toán khó. Nếu như các văn bản giống nhau hoàn toàn như bị sao chép hoàn toàn, không có thay đổi nào thì dễ dàng phát hiện được. Tuy nhiên, hầu hết là kiểu phát hiện các văn bản gần trùng lặp; nên đây là một vấn đề khó hơn nhiều và các dạng trùng lặp là vô cùng đa dạng.

Một văn bản có thể được sao chép toàn bộ hay chỉ một phần, các phần văn bản sao chép có thể bị thay đổi như thêm, xoá hoặc bị xáo trộn vị trí và nằm ở những vị trí bất kỳ của văn bản mới. Văn bản mới sau khi sao chép có thể chỉ sai khác với văn bản cũ ở một vài phần nhỏ hoặc cũng có thể không giống nhau bao nhiêu. Ngoài ra, còn có kiểu sao chép ý tưởng… Chính vì sự đa dạng trong việc sao chép văn bản mà không thể có một giải thuật hay kỹ thuật nào đo được một cách chính xác sự giống nhau giữa các văn bản. Bài toán này tuy không phải là mới, nhưng ở Việt Nam vẫn chưa có những nghiên cứu và ứng dụng rõ ràng được công bố.

1.5.2. Các vấn đề liên quan về sao chép

1.5.2.1. Các hình thức sao chép

Theo Meuschke và Gipp [80] các hình thức đạo văn được phân loại như sau:

- *Sao chép nguyên xi*: Tức là sao phép hoàn toàn nội dung mà không có một sự thay đổi nào, dưới hình thức "Sao chép và dán" đây là hình thức phổ biến nhất của đạo văn.

- *Sao chép được ngụy trang*: Là hình thức sao chép “Trộn và dán”, đây là hình thức sao chép, ghép nối các đoạn văn bản và có một vài thay đổi nhỏ so với tài liệu nguồn bằng cách thay đổi trật tự từ, thay thế từ đồng nghĩa, thêm hoặc xóa bớt các từ, hay là dùng hình thức diễn giải lại, hay là sử dụng các kỹ thuật để ngụy trang.

- *Sao chép bằng cách dịch* từ ngôn ngữ này sang ngôn ngữ khác.

- *Sao chép cấu trúc và ý tưởng*: Đề cập một loại sử dụng cấu trúc, ý tưởng của người khác, diễn giải khái niệm rộng hơn mà không đưa ra trích dẫn nguồn phù hợp.

- *Tự sao chép*: Là tái sử dụng một phần hoặc hoàn toàn bài viết của riêng mình mà không ghi nguồn phù hợp.

1.5.2.2. Phương pháp phát hiện sao chép văn bản

Qua quá trình nghiên cứu, khảo sát và đánh giá, luận án tổng hợp các phương pháp, kỹ thuật phát hiện sao chép văn bản có thể được phân loại gồm: Các phương pháp dựa trên ký tự (*Character-based methods*), dựa trên tần suất (*Frequency-based methods*), dựa trên cấu trúc (*Structural-based methods*), dựa trên phân lớp và gom cụm (*Classification and Cluster-based methods*), dựa trên cú pháp (*Syntax-based*

methods), phát hiện gần trùng lặp (*Near Dupplicate Detection*), dựa trên ngữ nghĩa (*Semantic-based methods*), dựa trên trích dẫn (*Citation-based methods*), kế thừa văn bản (*Recognizing Textual Entailment*).

Bảng 1.3. Một số phương pháp phát hiện sao chép văn bản

Phương pháp	Tóm tắt nội dung phương pháp
Dựa trên ký tự	Các kỹ thuật phát hiện sao chép đơn giản là dựa trên các phương pháp ký tự để so sánh các tài liệu bị nghi ngờ với các tài liệu gốc. Chuỗi ký tự giống nhau có thể được phát hiện một cách chính xác hoặc một phần bằng cách sử dụng các tiếp cận so khớp ký tự dựa trên các thuật toán so khớp chuỗi như Brute-Force, Boyer-Moore, Rabin-Karp,... Độ tương tự giữa văn bản được xác định bằng độ đo khoảng cách như Euclid, Manhattan, Levenshtein hoặc độ tương tự như Cosine, Jaccard, Dice,… Phương pháp dựa trên ký tự hay từ thường sử dụng n-gram, dấu vân tay và thống kê tần suất xuất hiện. - **Phương pháp n-gram:** Thường sử dụng các cụm n từ để tính độ tương tự. *Ví dụ:* bigrams từ [128], trigrams từ trong hệ thống Ferret [56, 73], 5-grams từ [52, 141], 8-grams từ [12], 16-grams ký tự trong hệ thống ENCOPLOT [41, 42]. - **Phương pháp dấu vân tay** được cho là hiệu quả hơn vì nó tạo ra một mô tả hàm băm, đó là "dấu vân tay" cho mỗi tài liệu và sau đó dấu vân tay của các tài liệu có thể được so sánh thay vì toàn bộ tài liệu. *Ví dụ:* Các hệ thống Diff, COPS, SCAM, KOALA, SSK, CHECK, MDR, PP Checker, SNITCH và Ferret sử dụng kỹ thuật dấu vân tay và truy xuất thông tin mờ để tìm các chuỗi khớp nhau và phát hiện giống nhau dựa trên tỷ lệ dấu vân tay [15], sử dụng tần số từ (3-grams hoặc 4-grams) tạo thành các dấu vân tay [44], hệ thống KOPI sử dụng chuỗi n-gram và dấu vân tay [95], 7-grams từ và dấu vân tay [119], kết hợp của dấu vân tay và VSM [124], dấu vân tay trên dữ liệu có cấu trúc [64]. - **Phương pháp thống kê tần suất xuất hiện** để tính giá trị trọng số tần suất xuất hiện của các từ khóa trong tài liệu. *Ví dụ:* Sử dụng phương pháp TF-IDF và so khớp n-gram từ [11, 84], sử dụng n-gram từ và tính toán độ tương tự bằng cách đếm tần số xuất hiện của những từ chung theo ngữ nghĩa [10], sử dụng TF-IDF dựa trên ngữ nghĩa thông qua danh mục thuật ngữ được lưu trong CSDL [138].

Phương pháp	Tóm tắt nội dung phương pháp
Dựa trên cấu trúc	Phương pháp này thường dựa vào các đặc điểm cấu trúc của văn bản như các tiêu đề, phần, đoạn văn và tài liệu tham khảo. *Ví dụ:* Biểu diễn các đặc trưng theo cấu trúc cây là một mô hình ánh xạ tự tổ chức nhiều tầng (ML-SOM) [103, 104], có những ý tưởng dựa trên hai lớp, lớp trên và lớp dưới như: Lớp trên biểu diễn việc phân cụm các tài liệu, còn lớp dưới sử dụng hệ số tương tự Cosine để đánh giá văn bản giống nhau [24]; hay hệ thống Multilevel Text Comparison sử dụng 4-grams từ cho lớp trên và lớp dưới sử dụng độ đo khoảng cách Levenshtein [140]. Các ví dụ khác dựa trên cấu trúc như: Hệ thống Plagiarism Pattern Checker sử dụng n-gram từ trong một câu và kết hợp với WordNet để kiểm tra các từ đồng nghĩa [51]; dựa trên TF-IDF với chỉ mục ngữ nghĩa tiềm ẩn (LSI) [107]; sử dụng cây phân tích cú pháp để tìm ra mối quan hệ cấu trúc giữa các văn bản [83].
Dựa trên phân lớp và gom cụm	Kỹ thuật gom cụm là một trong những kỹ thuật truy vấn thông tin có thể được dùng trong phát hiện sao chép [52, 141]. Nó được sử dụng để cải thiện hiệu quả với việc giảm thời gian tìm kiếm để làm giảm thời gian so sánh trong việc phát hiện sao chép. Đây là cách tiếp cận sử dụng những từ khóa để tìm các cụm từ giống nhau giữa các tài liệu [115, 140].
Dựa trên cú pháp	Kỹ thuật này xem xét việc sử dụng các thẻ từ loại (POS) mang tính cú pháp để biểu diễn cấu trúc của văn bản làm cơ sở so sánh và phân tích thêm. Kỹ thuật này xếp hạng thứ tự các tài liệu dựa trên các thẻ POS hay sử dụng dãy con chung dài nhất (LCS) để tính toán sự tương tự giữa các tài liệu và xếp hạng chúng theo thứ tự các tài liệu dựa trên mức độ liên quan [34, 35].
Phát hiện gần trùng lặp (NDD)	Phương pháp này phát hiện những nội dung văn bản hay một trang web gần trùng lặp nhằm tối ưu trong lưu trữ dữ liệu thường sử dụng các hàm băm để xử lý. *Ví dụ:* Dựa trên tiếp cận phân cụm để phát hiện gần trùng lặp [137], sử dụng dấu vân tay và mô hình vector nhằm xử lý văn bản trên web [77], tích hợp các thuật toán lọc [136], sử dụng mô hình vector và NLP trong tách từ, gốc từ, loại bỏ từ dừng và thay thế từ đồng nghĩa [110].
Dựa trên WordNet	Sử dụng WordNet để giải quyết vấn đề ngữ nghĩa của văn bản. *Ví dụ:* Xây dựng mạng từ WordNet để tính toán sự tương tự ngữ nghĩa giữa các từ, giữa các tài liệu [82], áp dụng từ đồng nghĩa, từ đồng âm và thay thế từ đồng âm sử dụng WordNet trong hệ thống ROUGE, tính

Phương pháp	Tóm tắt nội dung phương pháp
	độ tương tự dựa trên tần suất n-gram và LCS [23, 69], sử dụng từ điển đồng nghĩa tiếng Séc và một từ điển tiếng Anh tương ứng và đối chiếu với WordNet tiếng Anh [3, 19], sử dụng đo độ được kết hợp với các mô hình ngôn ngữ dựa trên WordNet để tính độ tương tự ngữ nghĩa [26]. Xác định sự tương tự ngữ nghĩa giữa hai từ bằng cách tính toán mức độ liên hệ giữa các từ này dựa trên thông tin từ một từ điển. Sau đó tính toán mức độ quan hệ giữa những từ đó bằng cách kiểm tra hệ thống phân cấp của từ trong một từ điển đồng nghĩa WordNet [131].
Dựa trên ngữ nghĩa	Bên cạnh phương pháp dựa trên WordNet, còn có các phương pháp khác như: Đếm số nút của đường ngắn nhất giữa hai khái niệm để xác định sự tương tự về ngữ nghĩa [58]. Hay sử dụng kỹ thuật yêu cầu gán nhãn ngữ nghĩa (*Semantic Role Labeling* - SRL) tập trung phân tích và so sánh văn bản dựa trên phân bổ ngữ nghĩa cho từng thành phần bên trong câu [91], lồng ghép phân tích ngữ nghĩa tiềm ẩn (LSA) với kỹ thuật tiền xử lý văn bản, loại bỏ các từ phổ biến trong các văn bản và dựa vào gốc từ [20, 36], sử dụng thuật toán Normalised Word Vector để tính toán mức độ tương tự dựa trên mô hình VSM với từ đồng nghĩa tổng quát [32].
Dựa trên trích dẫn	Kỹ thuật dựa trên trích dẫn là một trong những phương pháp mới trong việc phát hiện sao chép. Các phương pháp được sử dụng để xác định các tài liệu học thuật đã được đọc và sử dụng mà không đề cập đến các tài liệu đó. Phương pháp này thuộc về kỹ thuật phát hiện đạo văn ngữ nghĩa vì nó tập trung vào việc phát hiện dựa trên ngữ nghĩa có trong các trích dẫn được sử dụng trong một tài liệu học thuật [37, 38].
Phát hiện kế thừa văn bản (RTE)	Để nghiên cứu về diễn giải trong các văn bản, một lĩnh vực được quan tâm là nhiệm vụ phát hiện kế thừa văn bản (*Recognizing Textual Entailment* - RTE). Bài toán RTE tập trung vào suy luận văn bản và rất thường xuyên đề cập đến các biến thể ngữ nghĩa giữa các cặp biểu thức. Cho một văn bản T (nhiều câu) và một giả thuyết H (câu đơn), mục đích của RTE là để phát hiện xem T được suy ra từ H. Sự khác biệt lớn giữa diễn giải và kế thừa văn bản đó là diễn giải mang tính hai chiều, trong khi kế thừa văn bản chỉ suy luận rằng một văn bản T đòi hỏi một giả thuyết H [28]. Để giải quyết bài toán RTE thường sử dụng các kỹ thuật NLP như dùng VerbNet, gán nhãn ngữ nghĩa sử dụng PropBank và kết hợp với dịch máy thống kê [4].

Nhìn chung, các nghiên cứu tập trung vào so khớp từ sử dụng phương pháp n-gram, dấu vân tay, thống kê tần suất xuất hiện của từ. Mục đích nghiên cứu chính của đề tài là đánh giá mức độ giống nhau của văn bản để áp dụng trong phát hiện sao chép văn bản. Vì vậy, hướng nghiên cứu sẽ tập trung vào các phương pháp tiếp cận về n-gram, phương pháp dựa trên tần số hay thống kê, dấu vân tay, phương pháp cấu trúc và kết hợp với NLP để đi sâu vào so sánh sự giống nhau của văn bản dựa trên chuỗi ký tự. Dựa trên những kết quả nghiên cứu về phương pháp phát hiện sao chép hiện có, các phương pháp tiếp cận được phân loại như trên, để giải quyết bài toán, trong luận án có thể kết hợp các phương pháp sau: Trùng lặp n-gram đơn vị từ, phương pháp dựa trên tần số, phương pháp dấu vân tay và phương pháp cấu trúc.

1.5.3. Phát hiện sao chép tại PAN

1.5.3.1. Giới thiệu

Hội nghị quốc tế thường niên PAN [5] (**P**lagiarism Analysis, **A**uthorship Identification, and **N**ear-Duplicate detection) xuất hiện lần đầu tiên vào năm 2007 và thường xuyên tổ chức định kỳ hằng năm cho đến nay. Đây là hội nghị uy tín nhất thế giới hiện nay về lĩnh vực phát hiện sao chép. Hội nghị tập trung vào những nhóm chủ đề như: Phát hiện sao chép (*Plagiarism Detection*), bản quyền tác giả (*Author Identification*) và hồ sơ tác giả (*Author Profiling*) mà trong đó nhiệm vụ phát hiện sao chép là một hướng nghiên cứu được sự quan tâm của nhiều nhóm tác giả.

1.5.3.2. Nhiệm vụ phát hiện sao chép của PAN

Tại PAN, có hai nhiệm vụ chính để giải quyết bài toán phát hiện sao chép [80]:

- ***External Plagiarism Detection*** (tạm dịch là phát hiện sao chép bên ngoài): Phát hiện sao chép trong một tài liệu văn bản bằng cách tìm sự tương tự giữa nó với các tài liệu khác trong kho dữ liệu.

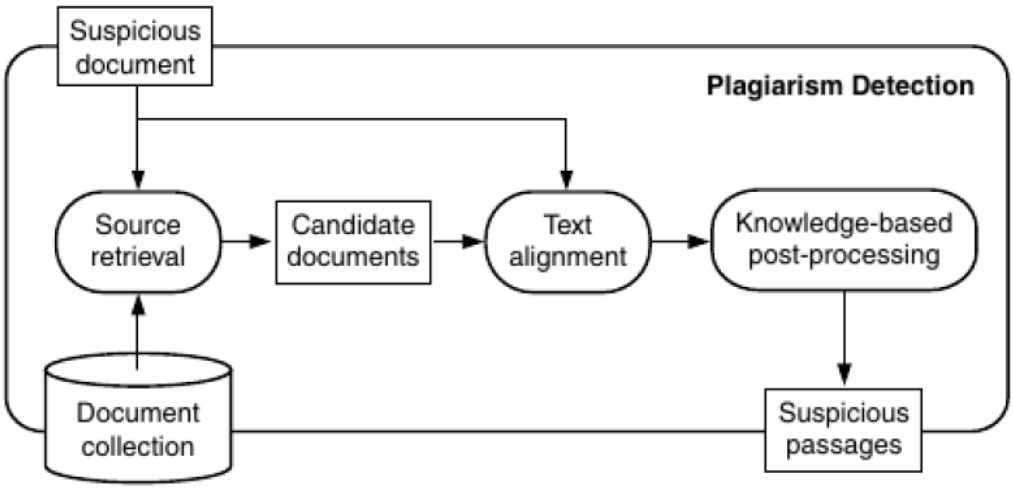
[5] https://pan.webis.de/ (truy cập tháng 10/2019)

- ***Intrinsic Plagiarism Detection*** (tạm dịch là phát hiện sao chép nội tại): Phát hiện ra sáo chép dựa vào sự không đồng nhất trong phong cách viết.

Hằng năm, ban tổ chức của PAN sẽ đưa ra yêu cầu và nhiệm vụ cần giải quyết, đồng thời cung cấp bộ dữ liệu để huấn luyện (*Training corpus*), các phương pháp dùng để đo hiệu suất và bộ dữ liệu kiểm tra (*Test corpus*) để các nhóm nghiên cứu tham gia và giải quyết bài toán cụ thể. Vì vậy, có rất nhiều nhóm nghiên cứu đến từ các phòng thí nghiệm trên thế giới tham gia.

1.5.3.3. Mô hình hệ thống phát hiện sao chép của PAN

Một mô hình tổng quan cho quá trình xử lý để phát hiện sao chép đã được đề xuất trong các giải pháp có hiệu quả cao trong cuộc thi của PAN [101, 124]. Với một tài liệu nghi ngờ (*Suspicious document*), quá trình tìm kiếm để phát hiện sao chép sẽ thực hiện tìm kiếm, kiểm tra trên một tập dữ liệu rất lớn (*Document collection*).



Hình 1.3. Mô hình xử lý tổng quát để phát hiện sao chép [124]

Quá trình này gồm ba bước chính:

- **Bước 1: Lọc ra *(Source retrieval)* các tài liệu tiềm năng bị sao chép *(Candidate documents)*:** Chọn một nhóm nhỏ các tài liệu ứng viên được xem là sao chép *(Suspicious document)* từ tập tài liệu lớn hay là kho dữ liệu *(Document collection)*. Các tài liệu ứng viên là các tài liệu được xác định có khả năng cao là nguồn của đạo văn liên quan đến tài liệu nghi ngờ.

- **Bước 2: So khớp văn bản *(Text alignment)*:** So sánh tài liệu nghi ngờ với từng tài liệu ứng viên và trích xuất các đoạn tương tự từ mỗi cặp tài liệu này.

- **Bước 3: Hậu xử lý *(Knowledge-based post-processing)*:** Xử lý, trình bày và so khớp từng đoạn đạo văn *(Suspicious passage)* trên một giao diện phù hợp nhằm giúp cho người sử dụng có thể xử lý các tác vụ về sau.

Trên đây là các bước chính, tuy nhiên để hệ thống phát hiện sao chép có thể sử dụng được trong thực tiễn thì phải có giải pháp thích hợp cho việc tạo lập và duy trì chỉ mục của tất cả tài liệu trong tập tài liệu nguồn cũng như có mô hình tính toán phù hợp để đáp ứng hiệu năng về độ chính xác và thời gian.

1.5.3.4. Đánh giá các nghiên cứu liên quan tại PAN

Chủ đề phát hiện sao chép (*Plagiarism Detection*) tại PAN được các nhà khoa học tập trung nghiên cứu và giải quyết liên tục trong nhiều năm qua các lần Hội nghị thường niên. Trong đó, nhiệm vụ phát hiện sao chép trong một tài liệu bằng cách tìm sự tương tự giữa nó với các tài liệu khác trong kho dữ liệu (*External Plagiarism Detection - EPD*) là đề tài chính được bàn luận, nghiên cứu và triển khai thông qua các cuộc thi từ năm 2009 đến 2011. Các năm về sau, PAN tập trung nghiên cứu và giải quyết bài toán theo hướng ngữ nghĩa, các hình thức đạo văn tinh vi khác.

Trước mỗi lần Hội nghị, PAN đưa ra bài toán cụ thể và các nhiệm vụ cần giải quyết, cung cấp bộ dữ liệu để huấn luyện và các hướng dẫn kèm theo để các nhóm nghiên cứu trên thế giới tham gia tiến hành nghiên cứu và đề xuất các giải pháp cũng như thuật toán cụ thể để giải quyết bài toán và thử nghiệm trên bộ dữ liệu huấn luyện và sử dụng các phương pháp đo hiệu suất của PAN để tính toán kết quả đạt được. Sau đó, các nhóm nghiên cứu gửi dự án để Ban Giám khảo của PAN chấm dựa trên bộ dữ liệu huấn luyện và bộ dữ liệu kiểm tra để đánh giá kết quả cuối cùng qua các độ đo hiệu suất và kết quả được công bố trong các Hội nghị của PAN. Dưới đây là kết quả các nhóm xếp thứ nhất của các lần thi trong nhiệm vụ EPD.

Bảng 1.4. Kết quả các nhóm xếp thứ nhất trong nhiệm vụ EPD

Cuộc thi	Nhóm tác giả	Kết quả các độ đo (%)		
		prec	***rec***	***Fscore***
PAN-1, 2009	Grozea, Gehl, and Popescu [100]	74.18	65.85	69.76
PAN-2, 2010	J. Kasprzak and M. Brandejs [102]	94.05	69.15	79.68
PAN-3, 2011	J. Grman and R. Ravas [122]	93.50	39.69	55.69

1.5.3.5. Xác định hướng nghiên cứu liên quan đến PAN

Lĩnh vực nghiên cứu về sao chép rất rộng lớn, với rất nhiều hình thức sao chép. Bên cạnh đó, mục tiêu chính của luận án là nghiên cứu, đề xuất phương pháp tính độ tương tự của văn bản, đánh giá mức độ giống nhau về nội dung văn bản. Vì vậy, qua khảo sát và nghiên cứu tổng quan, chúng tôi nhận thấy nhiệm vụ phát hiện sao chép trong một tài liệu văn bản bằng cách tìm sự tương tự giữa nó với các tài liệu khác trong kho dữ liệu (*External Plagiarism Detection*) phù hợp với hướng nghiên cứu mà luận án đặt ra và hướng đến.

Qua kết quả đạt được tại PAN, chúng ta thấy rằng việc phát hiện các văn bản giống nhau khó đạt kết quả tuyệt đối. Vì vậy, đây cũng chính là cơ sở để luận án nghiên cứu giải quyết bài toán theo hướng chủ đề này.

1.6. Kết luận Chương 1

Nội dung Chương 1 trình bày các vấn đề về khảo sát, phân tích và đánh giá các mô hình biểu diễn văn bản, các phương pháp tính độ tương tự văn bản. Những nghiên cứu tổng quan về phát hiện sao chép văn bản được khảo sát và đề cập chi tiết đến các nội dung liên quan như: Hệ thống phát hiện sao chép văn bản, các hướng tiếp cận và phương pháp phát hiện sao chép văn bản.

Trên cơ sở các phân tích, đánh giá và các định hướng nghiên cứu đề ra, chúng tôi nhận thấy đề tài "***Đánh giá mức độ giống nhau của văn bản tiếng Việt***" là một hướng nghiên cứu mang tính thời sự, có ý nghĩa khoa học và phù hợp với tình hình thực tế. Đề tài này bao hàm những nội dung rộng lớn, nên qua phân tích, đánh giá, chúng tôi xác định phạm vi giải quyết của đề tài tập trung chính vào nhiệm vụ đánh giá mức độ giống nhau của văn bản dựa trên các phương pháp so khớp văn bản nhằm phát hiện sự giống nhau của một văn bản so với kho văn bản lớn sẵn có. Tuy hiện nay có nhiều nghiên cứu và ứng dụng được triển khai sử dụng nhưng để giải quyết một cách triệt để, chính xác thì vẫn còn nhiều vấn đề cần tiếp tục nghiên cứu, nhất là giải quyết bài toán xử lý văn bản tiếng Việt.

Chương 2. SO SÁNH VĂN BẢN DỰA TRÊN MÔ HÌNH VECTOR

Chương này giới thiệu tổng quan về bài toán so sánh văn bản, trình bày phương pháp tính trọng số các đặc trưng của văn bản biểu diễn trên mô hình vector; giới thiệu bài toán và thực nghiệm một số phương pháp so sánh văn bản dựa trên mô hình vector để chứng minh mô hình biểu diễn văn bản dựa trên vector là thông dụng và có thể ứng dụng trong phát hiện sao chép văn bản tiếng Việt. Từ đó, trong luận án đề xuất phương pháp cải tiến bằng cách tính trọng số theo đơn vị từ và câu. Ngoài ra, nội dung Chương này trình bày lại việc tạo ra các bộ dữ liệu tiếng Việt thử nghiệm và xây dựng ứng dụng để triển khai và đánh giá các phương pháp dựa trên mô hình vector.

2.1. Giới thiệu

So sánh văn bản thực chất là bài toán tính độ tương tự giữa các văn bản. Đây là bài toán được sử dụng phổ biến trong lĩnh vực NLP. Một số phương pháp tính độ tương tự giữa các văn bản được phân loại như sau:

- Độ tương tự về mặt chuỗi (*String similarity*): Tính độ tương tự giữa hai văn bản thông qua việc so khớp về mặt ký tự hay từ, cụm từ của hai văn bản để tìm ra tỷ lệ giống nhau. Để tính độ tương tự về mặt chuỗi có thể sử dụng các thuật toán so khớp chuỗi như Brute-Force, Naïve, Morris-Pratt, KMP, Boyer-Moore, Rabin-Karp, Horspool, Levenshtein,… hay các phương pháp thống kê sử dụng các độ đo như Cosine, Jaccard, Euclide, Manhattan,…

- Độ tương tự về mặt ngữ nghĩa (*Semantic similarity*): Là một giá trị tin cậy để phản ánh mối quan hệ ngữ nghĩa của văn bản, nó sẽ làm cho việc so sánh văn bản trở nên thông minh và chính xác hơn. Tính độ tương tự về mặt ngữ nghĩa thường dựa trên cơ sở tri thức và các từ điển ngữ nghĩa như WordNet, Brown, Penn TreeBank, Wikipedia,... Trên thực tế, vấn đề ngữ nghĩa chỉ được hiểu đầy đủ trong một ngữ cảnh cụ thể nên khó có thể tính được giá trị có độ chính xác cao và thường phụ thuộc nhiều vào ngôn ngữ, quan hệ ngữ nghĩa giữa các từ.

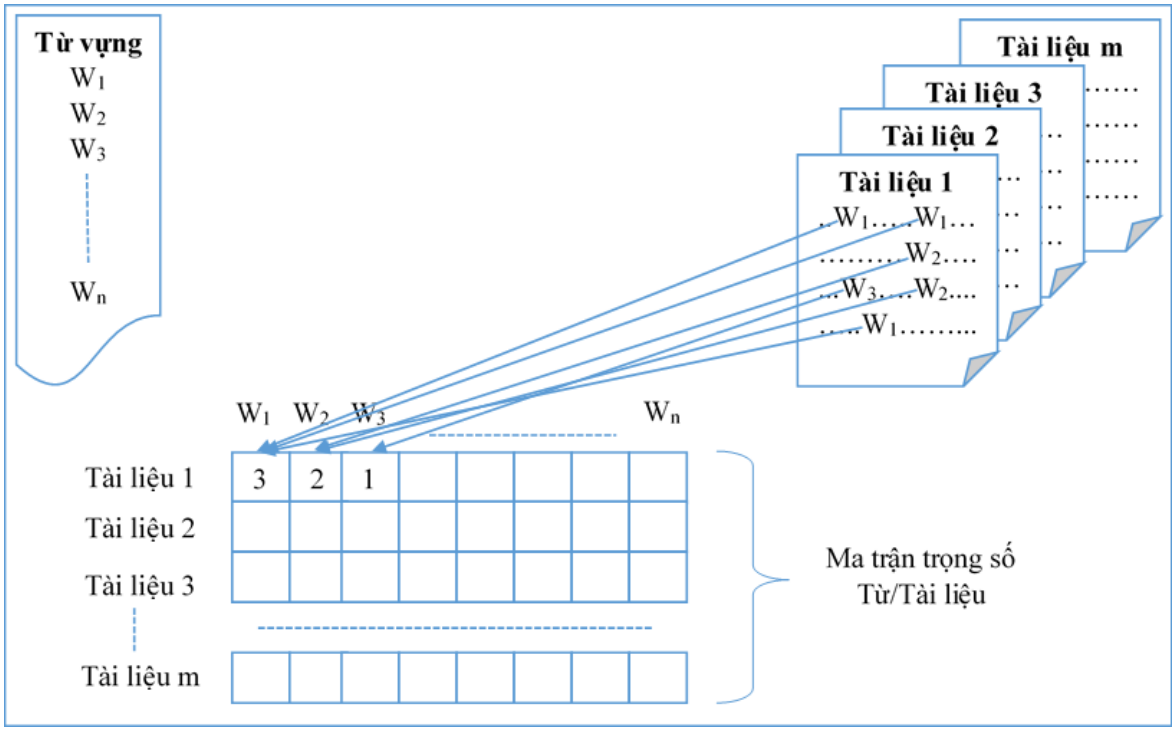
Vì vậy, trong phạm vi nghiên cứu của luận án này tập trung vào tính toán độ tương tự văn bản mà không xem xét đến yếu tố ngữ nghĩa. Nội dung nghiên cứu trong Chương này được đề cập liên quan đến phương pháp thống kê tần suất xuất hiện của từ hay cụm từ được biểu diễn dựa trên mô hình vector và sử dụng một số độ đo tương tự phổ biến để tính độ tương tự văn bản. Trình bày các kết quả nghiên cứu liên quan đến việc so sánh mức độ giống nhau của hai văn bản. Việc so sánh này phục vụ mục đích xác định mức độ tương tự của một văn bản này với một văn bản khác. Phương pháp đề xuất là chuyển các văn bản thành các vector, mỗi phần tử của vector là trọng số tương ứng với từ chỉ mục xuất hiện trong văn bản. Việc so sánh mức độ giống nhau của hai văn bản là tính độ tương tự bằng cách tính góc tạo bởi hai vector hay khoảng cách các điểm giữa hai vector, tùy thuộc vào việc áp dụng công thức tính độ đo. Cuối cùng, trong luận án đã trình bày việc tạo ra các bộ dữ liệu tiếng Việt và phát triển công cụ thử nghiệm phục vụ so sánh hai văn bản hoặc một văn bản với một tập các văn bản cho trước. Kết quả đạt được phản ánh được mức độ giống nhau của văn bản và cho thấy phương pháp biểu diễn văn bản dựa trên mô hình vector là thông dụng và phù hợp để tính độ tương tự văn bản.

2.2. Tính độ tương tự văn bản trong mô hình vector

2.2.1. Biểu diễn văn bản theo mô hình vector

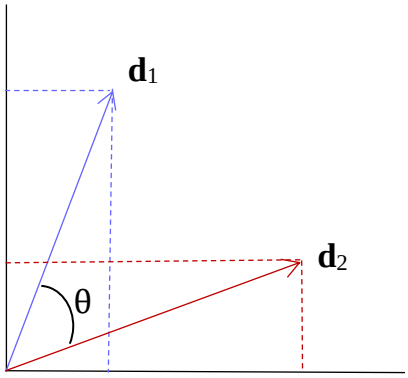
Sau khi tiền xử lý, một văn bản được mô tả bởi một tập các từ khóa (term) hay cụm từ (gọi là từ chỉ mục). Tập các từ chỉ mục xác định một không gian mà mỗi từ chỉ mục tượng trưng cho một chiều trong không gian đó. Các từ chỉ mục này cũng chính là các từ chứa nội dung chính của tập văn bản, mỗi từ chỉ mục này có thể được gán một trọng số để biểu diễn mức độ quan trọng của một từ (thường dùng phương pháp TF-IDF) [114]. Như vậy, tất cả các văn bản được biểu diễn theo mô hình vector sẽ tạo thành ma trận trọng số Từ/Tài liệu như Hình 2.1.

Mô hình vector là cách biểu diễn văn bản đơn giản và được dùng phổ biến trong NLP. Tuy nhiên, một trong những điểm yếu của nó là tạo thành không gian vector với số chiều rất lớn. Những nội dung liên quan sẽ được trình bày ở các phần tiếp theo.



Hình 2.1. Mô hình vector tạo thành ma trận trọng số Từ/Tài liệu

Ví dụ 2.1: Biểu diễn hai văn bản lần lượt là hai vector: $\mathbf{d}_1$ (w_{11},w_{21}....,w_{n1}) và $\mathbf{d}_2$ (w_{12},w_{22}...,w_{n2}); w_{ij} là trọng số từ khóa thứ *i* trong văn bản j. Độ tương tự của hai văn bản thường được định nghĩa là khoảng cách các điểm hoặc là góc giữa những vector trong không gian [47]. Hình dưới là ví dụ về góc tạo bởi hai vector $\mathbf{d}_1$ và $\mathbf{d}_2$.



Hình 2.2. Ví dụ về góc tạo bởi hai vector $\mathbf{d}_1$ và $\mathbf{d}_2$

Để tính giá trị góc tạo bởi hai vector thường sử dụng độ đo Cosine, giá trị tính được sẽ nằm trong khoảng [0, 1], giá trị càng lớn thể hiện độ tương tự (giống nhau) giữa hai vector càng cao. Ngoài ra, để tính độ tương tự của hai vector ta có thể sử dụng các độ đo khoảng cách giữa hai vector (như Euclid, Manhattan, Levenshtein).

2.2.2. Phương pháp tính trọng số từ chỉ mục

Biểu diễn văn bản dựa trên mô hình vector nhằm chuyển văn bản thành các vector với các thành phần là trọng số của từ chỉ mục. Phương pháp biểu diễn phổ biến nhất là mô hình túi từ (BoW) với trọng số là số lần xuất hiện của từ chỉ mục có trong văn bản và phương pháp tính trọng số TF-IDF [111].

Trọng số của một từ chỉ mục phản ánh mức độ quan trọng của từ đó trong văn bản, nó chỉ ra mối quan hệ giữa tần số và độ quan trọng của các từ chỉ mục trong tập văn bản đang xét. Sử dụng các phương pháp lựa chọn từ chỉ mục ta nhận được tập từ vựng T dùng để biểu diễn văn bản. Đặc trưng cho độ quan trọng của từ chỉ mục thuộc tập T trong một văn bản bất kỳ là một giá trị số được gán cho từ đó trong văn bản. Công việc tính trọng số của từ chỉ mục còn được gọi là đánh trọng số các từ chỉ mục trong văn bản [6]. Dưới đây là các ví dụ về phương pháp tính trọng số phổ biến được dùng trong mô hình vector [97, 113, 126].

Ví dụ 2.2: Giả sử có 3 tài liệu/văn bản có nội dung như sau:

- VB1: "*Thành phố Đà Nẵng là thành phố của biển, núi và con người*"
- VB2: "*Đà Nẵng là thành phố đáng sống nhất Việt Nam*"
- VB3: "*Đà Nẵng tôi yêu*"

Một cách đơn giản để biểu diễn văn bản là dựa trên mô hình túi từ, tạo ra ma trận cho các từ/văn bản với trọng số là số lần xuất hiện của từ trong mỗi văn bản. Với các văn bản trên, ta có được ma trận chi tiết như sau:

	Thành	phố	Đà	Nẵng	là	của	biển	núi	và	con	người	đáng	sống	nhất	Việt	Nam	tôi	yêu
VB1	2	2	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
VB2	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0
VB3	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Tuy nhiên, trong thực tế các văn bản có độ dài ngắn khác nhau, tức là tổng số từ trong mỗi văn bản là khác nhau. Trong các văn bản có độ dài lớn thì số lần xuất hiện của một từ có thể sẽ nhiều hơn so với các văn bản có độ dài nhỏ hơn. Do đó, chúng ta cần phải chuẩn hóa tần số xuất hiện của một từ trong một văn bản dựa trên kích thước của văn bản đó bằng cách chia số lần xuất hiện của một từ cho tổng số từ trong văn bản. Phương pháp này được gọi là TF (*Term Frequency*).

Các ký hiệu sử dụng trong nội dung này được mô tả như sau:

- *t* là từ chỉ mục trong văn bản.
- d là một văn bản trong tập các văn bản.
- D là tập các văn bản (như một corpus).
- *N* là số lượng văn bản có trong tập D.
- N_d là số từ có trong văn bản d.
- $N_{t,d}$ là số lần xuất hiện của từ *t* trong văn bản d.
- **Phương pháp dựa trên tần số từ (TF)** được tính theo công thức sau:

$$\text{tf}(t,d) = \frac{N_{t,d}}{N_d} \quad (2.1)$$

Các giá trị trọng số được tính dựa trên tần suất (hay số lần) xuất hiện của từ *t* có trong văn bản d chia cho tổng số từ có trong văn bản d.

Ví dụ 2.3: Trong VB1, từ "*Thành*" xuất hiện hai lần và tổng số từ trong VB1 là 13. Do đó, tần số xuất hiện được chuẩn hóa theo TF của từ "*Thành*" là 2/13 = 0.153846 (làm tròn thành 0.15); từ "*biển*" là 1/13 = 0.076923 (làm tròn thành 0.08). Tương tự, ta tính được giá trị TF cho các từ khác còn lại trên các văn bản khác như: VB2 có 10 từ, giá trị TF của từ "*Đà*" là 1/10 = 0.1; VB3 có 4 từ, giá trị TF của từ "*Đà*" là 1/4 = 0.25…

Từ đó, ta có ma trận trọng số với tần suất xuất hiện được chuẩn hóa của các từ trong cả 3 văn bản như sau:

	Thành	phố	Đà	Nẵng	là	của	biển	núi	và	con	người	đáng	sống	nhất	Việt	Nam	tôi	yêu
VB1	0.15	0.15	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0	0	0	0	0	0	0
VB2	0.10	0.10	0.10	0.10	0.10	0	0	0	0	0	0	0.10	0.10	0.10	0.10	0.10	0	0
VB3	0	0	0.25	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0.25	0.25

Trong phương pháp này, trọng số của từ tỷ lệ thuận với số lần xuất hiện của từ *t* trong văn bản d. Khi số lần xuất hiện từ khóa *t* trong văn bản d càng lớn thì điều đó có nghĩa là văn bản d càng phụ thuộc vào từ *t*, hay nói cách khác từ *t* mang nhiều thông tin trong văn bản d.

Nhưng suy luận trên không phải lúc nào cũng đúng. Một ví dụ điển hình là từ "*và*" xuất hiện nhiều trong hầu hết các văn bản, nhưng trên thực tế từ này lại không mang nhiều ý nghĩa như tần suất xuất hiện của nó. Hoặc có những từ không xuất hiện trong văn bản này nhưng lại xuất hiện trong văn bản khác. Một phương pháp khác ra

đời khắc phục được nhược điểm của phương pháp TF, đó là phương pháp IDF (*Inverse Document Frequency*).

- **Phương pháp dựa trên nghịch đảo tần số văn bản (IDF)**

Trong phương pháp này, giá trị trọng số được tính theo công thức sau:

$$\text{idf}(t, D) = \log \frac{N}{\left|\{d \in D : t \in d\}\right|} \quad (2.2)$$

Trọng số trong công thức này được tính dựa trên độ quan trọng của từ *t* trong văn bản d. Nếu *t* xuất hiện càng ít trong văn bản d thì trọng số của nó đối với văn bản d càng lớn. Đây là điểm quan trọng để phân biệt văn bản d với các văn bản khác và hàm lượng thông tin trong nó càng lớn. Công thức được tính dựa trên hàm logarit của tử số là tổng số văn bản trong tập D (là *N*) và số lượng văn bản có chứa từ *t* (là *n*).

Theo công thức trên, nếu một từ nào đó xuất hiện ở tất cả các văn bản trong tập văn bản thì giá trị IDF = log(1) = 0, nhưng thực ra từ đó vẫn có một trọng số nhất định, khác với các từ không có trong văn bản nào. Bên cạnh đó, để tránh việc tính giá trị TF-IDF thực hiện phép nhân với giá trị IDF = 0 sẽ cho kết quả bằng 0, điều này ảnh hưởng đến việc tính giá trị trọng số. Vì vậy, ta thường cộng thêm 1 vào giá trị này. Việc thay đổi cơ số logarit trong công thức trên không làm thay đổi giá trị của một từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. Nói cách khác, việc thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF. Như vậy, việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng nhau và công thức tính được quy đổi như sau [78, 88]:

$$\text{idf}(t, D) = 1 + \ln(\frac{N}{n}) \quad (2.3)$$

Ví dụ 2.4: Với các ví dụ trên, ta có *N* = 3. Khi *n* = 1 thì giá trị IDF = 1 + ln(3) = 1 + 1.098612 = 2.098612 (làm tròn thành 2.099); khi *n* = 2 thì giá trị IDF = 1 + ln(3/2) = 1 + 0.405465= 1.405465 (làm tròn thành 1.405); khi *n* = 3 thì giá trị IDF = 1 + ln(1) = 1 + 0 = 1.

Áp dụng công thức 2.3 tính trọng số IDF của các từ ta có bảng giá trị sau:

Từ	Thành	phố	Đà	Nẵng	là	của	biển	núi	và	con	người	đáng	sống	nhất	Việt	Nam	tôi	yêu
IDF	1.405	1.405	1	1	1.405	2.099	2.099	2.099	2.099	2.099	2.099	2.099	2.099	2.099	2.099	2.099	2.099	2.099

- **Phương pháp TF-IDF**

Phương pháp này dựa trên mức độ quan trọng của mỗi từ trong văn bản để thống kê, vì khi một từ xuất hiện trong càng ít văn bản (tức là giá trị IDF nhỏ) thì khả năng phân biệt các văn bản dựa trên từ đó càng cao. Phương pháp này thường được sử dụng nhiều nhất để tính trọng số các đặc trưng, giá trị của ma trận trọng số được tính như sau:

$$\text{TF - IDF}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D) \quad (2.4)$$

Theo phương pháp TF-IDF, trọng số từ là tích của tần suất từ (TF) và tần suất nghịch đảo của từ đó (IDF). Đây là phương pháp kết hợp được ưu điểm của cả hai phương pháp trên. Trọng số được tính bằng tần suất xuất hiện của từ *t* trong văn bản d và độ hiếm của từ *t* trong toàn bộ tập văn bản.

Ví dụ 2.5: Với các ví dụ trên, ta tính được giá trị TF-IDF của 3 văn bản với trọng số các từ như bảng sau:

	Thành	phố	Đà	Nẵng	là	của	biển	núi	và	con	người	đáng	sống	nhất	Việt	Nam	tôi	yêu
VB1	0.216	0.216	0.08	0.08	0.108	0.161	0.161	0.161	0.161	0.161	0.161	0	0	0	0	0	0	0
VB2	0.141	0.141	0.10	0.10	0.141	0	0	0	0	0	0	0.21	0.21	0.21	0.21	0.21	0	0
VB3	0	0	0.25	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0.525	0.525

Để dễ theo dõi, ta quy đổi các giá trị được làm tròn như sau:

- 0.15 * 1.405 = 0.216225 (làm tròn thành 0.216)
- 0.08 * 1.405 = 0.108112 (làm tròn thành 0.108)
- 0.08 * 2.099 = 0.161296 (làm tròn thành 0.161)
- 0.1 * 1.405 = 0.140547 (làm tròn thành 0.141)
- 0.1 * 2.099 = 0.209861 (làm tròn thành 0.21)
- 0.25 * 2.099 = 0.524653 (làm tròn thành 0.525)

Như vậy, việc áp dụng mô hình vector để biểu diễn văn bản thì mỗi văn bản được mô hình hóa thành một vector đặc trưng và không gian các đặc trưng của tất cả các văn bản đang xét sẽ bao gồm tất cả các từ. Giả sử cần mô hình hóa *N* văn bản, với *N* văn bản này tách ra được *K* từ khóa riêng biệt (thông thường thì số từ tách ra đã được loại bỏ các từ dừng) thì không gian đặc trưng sẽ có *K* chiều.

Ví dụ 2.6: Mô hình hóa 1.000 văn bản/tài liệu (N = 1.000). Tách được 9.500 từ từ tập các văn bản này. Sau khi loại bỏ các từ dừng thì còn lại 8.235 từ (K = 8.235). Mô tả cho việc mô hình hóa này bằng một ma trận 1.000 hàng (văn bản) và 8.235 cột (từ). Với mỗi ô giao nhau của hàng và cột, tính một giá trị gọi là trọng số của hàng và cột tương ứng theo phương pháp TF-IDF dựa trên các công thức 2.1, 2.3, 2.4.

2.2.3. Phương pháp tính độ tương tự

Độ đo Cosine được sử dụng phổ biến nhất, đây là phương pháp tương đối đơn giản và cho kết quả với độ chính xác cao. Để tính giá trị tương tự của hai văn bản bằng Cosine thường dùng cách tính tích vô hướng của vector dựa trên những phép chia tỷ lệ của hai vector tần suất đặc trưng cho hai văn bản. Xét trên phương diện hình học, giá trị tương tự Cosine là giá trị lượng giác Cosine của góc giữa hai vector văn bản trong không gian đa chiều. Góc phân biệt càng nhỏ thì hai văn bản càng gần nhau hơn trong không gian nên độ giống nhau càng cao [46, 116].

Sau khi biểu diễn văn bản thành tập vector chứa các giá trị trọng số trong không gian vector, mỗi từ trong không gian vector sẽ có một trục riêng (một chiều). Để đo độ tương tự giữa hai vector **a** và **b**, ta sử dụng công thức để tính góc giữa hai vector theo độ đo Cosine như sau:

$$\text{Sim}(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\times\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \qquad (2.5)$$

Với các ví dụ trên, sau khi đã đánh trọng số cho các từ trong văn bản để biểu diễn thành ma trận trọng số Từ/Văn bản. Tiếp theo là các ví dụ để tính độ tương tự dựa trên độ đo Cosine:

Ví dụ 2.7: Cho văn bản kiểm tra VB4: "*Đà Nẵng*"

Văn bản này cũng được biểu diễn như là một vector. Chúng ta tính giá trị TF-IDF cho VB4 như sau:

	TF	IDF	TF-IDF
Đà	0.5	1	0.5
Nẵng	0.5	1	0.5

Tính độ tương tự giữa VB4 so với các văn bản khác theo độ đo Cosine:

- So với VB1: $\text{Sim(VB4, VB1)} = \frac{0.5\times0.08+0.5\times0.08}{\sqrt{0.5^2+0.5^2}\times\sqrt{0.08^2+0.08^2}} = \frac{0.08}{0.08} = 1$

- So với VB2: $\text{Sim(VB4, VB2)} = \frac{0.5\times0.1+0.5\times0.1}{\sqrt{0.5^2+0.5^2}\times\sqrt{0.1^2+0.1^2}} = \frac{0.1}{0.1} = 1$

- So với VB3: $\text{Sim(VB4, VB3)} = \frac{0.5\times0.25+0.5\times0.25}{\sqrt{0.5^2+0.5^2}\times\sqrt{0.25^2+0.25^2}} = \frac{0.25}{0.25} = 1$

Kết quả tính độ tương tự giữa VB4 so với các văn bản khác như sau:

	VB1	VB2	VB3
VB4	1	1	1

Với kết quả trên, ta thấy rằng VB4 là "*Đà Nẵng*" có trong tất cả các văn bản nên độ tương tự giữa VB4 với các văn bản còn lại đều cho kết quả bằng 1, tức là có độ tương tự 100%.

Ví dụ 2.8: Tương tự ta tính độ tương tự giữa VB5: "*Đà thành*" với tập văn bản đã cho.

Văn bản này cũng được biểu diễn như là một vector. Chúng ta tính giá trị TF-IDF cho VB5 như sau:

	TF	IDF	TF-IDF
Đà	0.5	1	0.5
thành	0.5	1.406	0.703

Tính độ tương tự giữa VB5 so với các văn bản khác theo độ đo Cosine:

- So với VB1:

$$\text{Sim(VB5, VB1)} = \frac{0.5\times0.08+0.703\times0.216}{\sqrt{0.5^2+0.703^2}\times\sqrt{0.08^2+0.216^2}} = \frac{0.192}{0.199} = 0.965$$

- So với VB2:

$$\text{Sim(VB5, VB2)} = \frac{0.5\times0.1+0.703\times0.141}{\sqrt{0.5^2+0.703^2}\times\sqrt{0.1^2+0.141^2}} = \frac{0.149}{0.149} = 1$$

- So với VB3:

$$\text{Sim(VB5, VB3)} = \frac{0.5\times0.25+0.703\times0}{\sqrt{0.5^2+0.703^2}\times\sqrt{0.25^2+0^2}} = \frac{0.125}{0.216} = 0.579$$

Kết quả tính độ tương tự giữa VB5 so với các văn bản khác như sau:

	VB1	VB2	VB3
VB5	0.965	1	0.579

Với kết quả trên, ta thấy rằng VB5 "*Đà thành*" với mỗi từ đều có trong văn bản VB1 và VB2 nhưng kết quả cho thấy độ tương tự giữa VB5 với VB1 là 0.965, vì trong VB1 có chứa 2 từ "*thành*", "*phố*" nên độ quan trọng của từ "*thành*", "*phố*" trong văn bản đó giảm, tức trọng số IDF giảm. Vì vậy, mặc dù từ "*Đà*", "*thành*" đều có trong VB1 nhưng độ tương tự Cosine có giá trị gần bằng 1.

2.2.4. Nhận xét

Mô hình vector và các biến thể của nó là các mô hình biểu diễn văn bản được sử dụng khá phổ biến trong xử lý, khai phá văn bản, tìm kiếm và truy xuất thông tin, mối quan hệ giữa các văn bản được tính toán dựa trên các vector nên dễ dàng thực hiện. Kết hợp giữa mô hình vector và phương pháp đánh trọng số TF-IDF được ứng dụng nhiều trong các bài toán xử lý văn bản. Bài toán tính độ tương tự văn bản sử dụng phương pháp này được thực hiện tương đối đơn giản, việc tính toán độ tương tự của các văn bản thường dựa vào khoảng cách các điểm hoặc là góc giữa những vector trong không gian.

Tuy nhiên, vì mỗi văn bản được biểu diễn thành một vector *n* chiều, với số chiều thường là số từ khóa khác nhau trong tập văn bản, do đó không gian biểu diễn sẽ có số chiều rất lớn, nên việc lưu trữ và tính toán trên vector trở nên tốn kém và phức tạp. Ngoài ra, theo mô hình vector thì việc lưu trữ các từ khóa hay đánh chỉ mục là yếu tố rất quan trọng. Chỉ cần một thay đổi nhỏ trong bảng từ vựng có thể sẽ vector hóa lại toàn bộ các tài liệu. Vì vậy, việc xử lý dữ liệu lớn cần phải kết hợp nhiều phương pháp để biểu diễn văn bản được tối ưu.

Phần tiếp theo, luận án trình bày những nội dung nghiên cứu, cài đặt một số phương pháp so sánh văn bản dựa trên mô hình vector và đề xuất phương pháp để tăng thêm độ chính xác. Xây dựng ứng dụng và thực nghiệm trên bộ dữ liệu tiếng Việt quy mô nhỏ để đánh giá phương pháp.

2.3. Một số phương pháp so sánh văn bản dựa trên mô hình vector

Trong phần này, luận án trình bày cách biểu diễn văn bản theo mô hình vector, sau đó ứng dụng các độ đo như Cosine, Jaccard, Manhattan, Levenshtein tính khoảng cách giữa hai vector để biết được độ tương tự của hai văn bản. Bằng cách tương tự cũng tính được độ tương tự giữa một văn bản cần kiểm tra với tập văn bản đã có trong kho dữ liệu. Qua kết quả nghiên cứu và thực nghiệm, có thể thấy được mô hình vector là một phương pháp phù hợp để biểu diễn văn bản trong tính toán độ tương tự văn bản.

2.3.1. Mô hình vector hóa văn bản

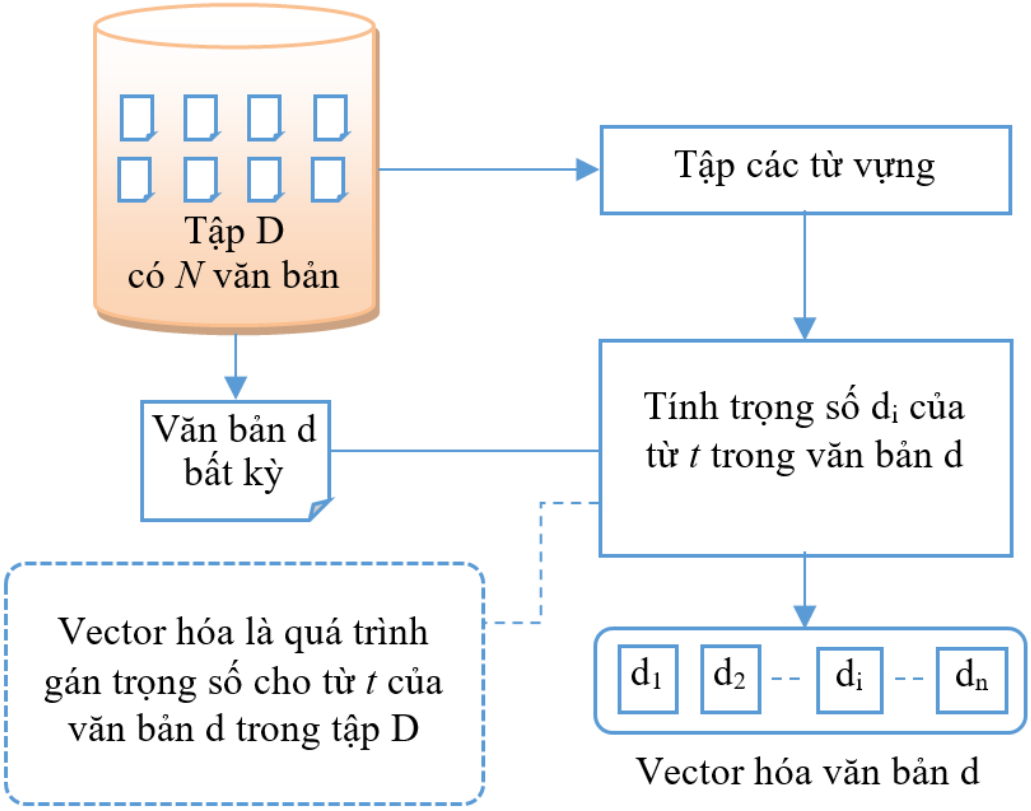
2.3.1.1. Vector hóa văn bản theo đơn vị từ

Trong phần này, luận án trình bày cách thức thực hiện mô hình hóa văn bản thành vector và tính toán độ tương tự giữa các vector và hiển thị kết quả là tỷ lệ giống nhau của hai văn bản hoặc của một văn bản với tập văn bản trong kho dữ liệu.

Sau bước tiền xử lý thực hiện tách từ đơn, loại bỏ từ dừng, văn bản sẽ gồm tập các từ khóa mà ý nghĩa của chúng tạo nên nội dung của văn bản và được biểu diễn dưới dạng vector như $\mathbf{d}_i = [t_1, t_2, ..., t_n]$, với n là số lượng từ khóa, t_k là trọng số của từ thứ k trong văn bản d với ($k = 1 \ldots n$), cách biểu diễn này gọi là vector hóa văn bản. Trong quá trình so sánh, bước xử lý vector hóa nhằm mục đích biểu diễn các văn bản dưới dạng vector để phục vụ cho việc so sánh sau này.

Việc tính độ tương tự giữa hai văn bản A và B được kí hiệu là Sim(A, B), được tính toán theo các công thức tính độ tương tự như Cosine, Jaccard... Nếu độ tương tự này là lớn (thường là vượt một ngưỡng nào đó) thì mức độ giống nhau giữa hai văn bản này là cao.

Trong phần này đề xuất và thực nghiệm mô hình vector hóa văn bản có thể thực hiện dựa trên đơn vị từ (mỗi phần tử vector là từ hay n-gram từ) theo quy trình như Hình 2.3.



Hình 2.3. Quá trình vector hóa theo đơn vị từ

Để tính giá trị đặc trưng cho văn bản, luận án thực hiện bằng phương pháp TF-IDF qua các bước như sau:

- **Bước 1** - Tính các giá trị: *N* là số lượng văn bản có trong tập D, *n* là số từ có trong văn bản d, $N_{t,d}$ là số lần xuất hiện của từ *t* trong văn bản d.
- **Bước 2** - Tính trọng số TF=tf(t,d) *theo công thức (2.1)*
- **Bước 3** - Tính trọng số IDF=idf(t,D) *theo công thức (2.3)*
- **Bước 4** - Tính trọng số d_i của từ *t*: TF-IDF= TF×IDF *theo công thức (2.4)*

Như vậy, ta đã biểu diễn văn bản thành mô hình vector với trọng số Từ/Văn bản. Giai đoạn tiếp theo là ứng dụng các độ đo tương tự để tính mức độ giống nhau giữa các văn bản.

2.3.1.2. Các độ đo tương tự

Sự tương tự giữa hai văn bản là sự giống nhau về nội dung giữa hai văn bản đó. Do đó, hai văn bản là bản sao hoặc gần giống nhau thì sẽ có nội dung giống nhau nhiều, hay "*độ tương tự*" giữa hai văn bản là cao. Độ tương tự nằm trong khoảng [0,

1], như vậy độ tương tự càng gần 1 thì khả năng các văn bản là bản sao hoặc gần giống nhau là cao và ngược lại. Do đó, để xét xem các văn bản có phải là bản sao hoặc gần giống nhau hay không ta phải tính độ tương tự giữa chúng [16].

Luận án thực hiện tính độ tương tự giữa hai văn bản qua các bước chính và ứng dụng các độ đo như Cosine, Jaccard, Manhattan, Levenshtein để tính khoảng cách giữa hai vector để biết được độ tương tự của hai văn bản. Các bước xử lý chính nhau sau:

- **Bước 1** - Tiền xử lý (Tách từ đơn, loại bỏ từ dừng, tạo danh sách từ vựng...).
- **Bước 2** - Xây dựng tập từ vựng chung T = {t_1, t_2..., t_n}
- **Bước 3** - Mô hình hóa văn bản thành vector: Dựa vào T ta tạo vector **a** và **b** với trọng số các từ của A và B lần lượt là a_i, b_j (bằng cách tính TF-IDF)
- **Bước 4** - Áp dụng công thức tính độ tương tự theo các độ đo.
- **Bước 5** - Hiển thị kết quả.

Giả sử, hai văn bản được biểu diễn dưới dạng vector là **a** và **b**. Với *n* là tổng số từ riêng biệt của hai văn bản. Ngoài công thức tính độ tương tự được sử dụng nhiều nhất là Cosine, còn một số công thức tính độ tương tự như Jaccard, Manhattan, Levenshtein [46, 62, 72].

a) Độ đo khoảng cách Manhattan

Khoảng cách Manhattan là một dạng khoảng cách giữa hai điểm trong không gian Euclid với hệ tọa độ Descartes. Đại lượng này được tính bằng tổng chiều dài của hình chiếu của đường thẳng nối hai điểm này trong hệ trục tọa độ Descartes.

Khi hai văn bản đươc biểu diễn thành hai vector đặc trưng **a** và **b** ta có công thức Manhattan [53, 57, 59] là:

$$\text{Manhattan}(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{n} \left| a_i - b_i \right| \tag{2.6}$$

Manhattan(**a**, **b**) nằm trong khoảng [0, 1], mức độ tương tự giữa hai vector này được tính toán bằng công thức sau:

$$\mathrm{Sim}_{\mathrm{M}}(\mathbf{a},\mathbf{b}) = 1 - \frac{\sum_{i=1}^{n} \left|a_i - b_i\right|}{n} \tag{2.7}$$

b) Độ đo sử dụng hệ số Jaccard

Ban đầu độ đo này dùng cho vector nhị phân với tử số là phép giao của tập các thành phần vector giống nhau của **a** và **b**, mẫu số là phép hợp của các thành phần vector khác nhau của **a** và **b**. Hệ số Jaccard được tính theo công thức sau [46, 87]:

$$\mathrm{Jaccard_bin}(\mathbf{a},\mathbf{b}) = \frac{\left|a \cap b\right|}{\left|a \cup b\right|} \tag{2.8}$$

Độ đo này được mở rộng cho các vector được liên kết với các trọng số được áp dụng để tính theo công thức sau [46, 87]:

$$\mathrm{Jaccard}(\mathbf{a},\mathbf{b}) = \frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n} a_i b_i} \tag{2.9}$$

c) Độ đo khoảng cách Levenshtein

Trong lĩnh vực lý thuyết thông tin, khoa học máy tính, khoảng cách Levenshtein dùng để đo độ tương tự giữa hai chuỗi, là một cách để đo khoảng cách hiệu chỉnh. Khoảng cách Levenshtein giữa chuỗi A và chuỗi B là số bước nhỏ nhất để biến đổi chuỗi A thành chuỗi B thông qua ba phép biến đổi là: Xóa một ký tự, thêm một ký tự và thay ký tự này thành ký tự khác. Để tính khoảng cách Levenshtein, trong trường hợp sử dụng từ hay cụm n-gram từ thì có hai khả năng so sánh chuỗi: So sánh trực tiếp sử dụng các phép biến đổi ký tự hoặc xem mỗi phần tử của từ hay n-gram từ là một đơn vị để so sánh [116, 125].

Với hai văn bản A và B, sau khi tiền xử lý được chuyển thành 2 chuỗi A và B biểu diễn dưới dạng vector lần lượt là **a** và **b**, độ dài hai chuỗi A và B cũng chính là độ dài của hai vector **a** và **b**.

Khoảng cách Levenshtein sử dụng thuật toán quy hoạch động, tính toán trên mảng 2 chiều (*n*+1)*(*m*+1) với *n* và *m* lần lượt là độ dài của hai vector **a** và **b**. Khoảng cách Levenshtein được tính có giá trị là *d*[*m*, *n*], với *s* là độ dài của chuỗi dài nhất. Độ đo tương tự của hai văn bản dựa trên khoảng cách Levenshtein được tính theo công thức sau [48, 57, 125]:

$$\mathrm{Sim}_{\mathrm{L}}\left(\mathbf{a},\mathbf{b}\right)=1-\frac{d[m,\,n]}{s} \tag{2.10}$$

d) Đánh giá

Trong luận án sử dụng các độ đo dựa vào thống kê tần suất xuất hiện của từ trong văn bản như các phương pháp trên. Trong đó, xác định độ tương tự văn bản bằng cách: 1) Tính góc của những vector sử dụng độ đo Cosine và hệ số Jaccard; 2) Dựa trên tính khoảng cách giữa các điểm bằng độ đo khoảng cách Manhattan và Levenshtein... Ưu điểm của các phương pháp này là tốc độ xử lý nhanh, tốn ít chi phí nhưng không đảm bảo độ chính xác về mặt ngữ nghĩa.

Đối với các văn bản tiếng Việt, phương thức ngữ pháp chủ yếu là dựa trên trật tự của từ trong câu văn bản nên nếu áp dụng mô hình biểu diễn văn bản dựa trên tần số xuất hiện của từ có thể không cho kết quả chính xác như mong đợi. Nguyên nhân là do vector biểu diễn văn bản không đảm bảo giữ được mối liên quan về mặt nội dung giữa các từ, vị trí các từ, cụm từ và vị trí của câu trong văn bản, hai vector có trật tự từ khác nhau vẫn có thể cho độ tương tự là hoàn toàn giống nhau.

Trong luận án đã ứng dụng, cài đặt các thuật toán trên để tính độ tương tự giữa hai văn bản. Từ kết quả nghiên cứu và qua thực nghiệm, với cách tính khoảng cách giữa các điểm có nhược điểm là không chính xác, vì khoảng cách sẽ lớn với các vector có chiều dài khác nhau nên sử dụng cách tính góc giữa những vector trong không gian cho kết quả chính xác hơn là cách tính khoảng cách giữa cách điểm. Vì vậy, trong luận án đã chọn độ đo **Cosine** là phương pháp thông dụng và phù hợp nhất để ứng dụng và cải tiến trong đề xuất thuật toán so sánh văn bản dựa trên từ hay câu và phát triển công cụ thử nghiệm. Phần sau sẽ giới thiệu và đề xuất phương pháp cải tiến để tính độ tương tự văn bản dựa trên mô hình vector sử dụng độ đo Cosine.

2.3.2. Phương pháp cải tiến sử dụng độ đo Cosine

2.3.2.1. Ảnh hưởng của trật tự các từ trong văn bản

Với phương pháp tính độ tương tự sử dụng độ đo Cosine như đã trình bày ở phần trước cho kết quả chính xác nhưng có một nhược điểm là không xem xét đến thứ tự các từ trong văn bản. Các văn bản cùng chứa một tập từ giống nhau nhưng các từ khác nhau về vị trí thì ý nghĩa cũng sẽ khác nhau. Điều này sẽ ảnh hưởng đến ý nghĩa của văn bản [66]. Dưới đây là một ví dụ để thấy việc thay đổi trật tự các từ trong văn bản sẽ ảnh hưởng đến độ tương tự của văn bản.

Ví dụ 2.9: Giả sử có hai văn bản a và b như sau:

- a: "*Đà Nẵng thành phố tôi yêu*" (là tựa đề của bài hát)
- b: "*Tôi yêu thành phố Đà Nẵng*" (thể hiện tình cảm của cá nhân)

Với hai văn bản trên, ta có tập từ chung T = {*Đà, Nẵng, thành, phố, tôi, yêu*}. Nếu biểu diễn theo mô hình vector với trọng số Từ/Văn bản và áp dụng công thức tính độ tương tự Cosine như đề cập ở phần trước thì kết quả Sim(a, b) sẽ bằng 1, tức hai văn bản này hoàn toàn giống nhau. Tuy nhiên, trên thực tế hai văn bản này mang ý nghĩa khác nhau. Vì vậy, ta cần phải tính đến thứ tự của các từ có trong văn bản để tăng thêm độ chính xác khi tính độ tương tự của văn bản.

Cách xác định về thứ tự của các từ trong mỗi văn bản như sau:

- Nếu từ trong văn bản có trong tập từ chung thì từ đó sẽ có cùng thứ tự với từ trong trong tập từ chung.
- Ngược lại, nếu từ trong văn bản không có trong tập từ chung thì thứ tự của từ đó sẽ bằng 0.

Gọi **r** là vector thứ tự các từ trong văn bản, với hai văn bản a và b ở trên, ta có hai vector thứ tự từ $\mathbf{r}_a$ và $\mathbf{r}_b$ tương ứng như sau:

- $\mathbf{r}_a$ = [1 2 3 4 5 6]
- $\mathbf{r}_b$ = [5 6 3 4 1 2]

Và với văn bản c: "*Đà thành*" thì hai văn bản a và c sẽ có hai vector thứ tự từ $\mathbf{r}_a$ và $\mathbf{r}_c$ tương ứng như sau:

- $\mathbf{r}_a$ = [1 2 3 4 5 6]
- $\mathbf{r}_c$ = [1 3 0 0 0 0]

Ta có công thức tính độ tương tự của thứ tự từ giữa hai vector [1, 65]:

$$\mathrm{Sim}_R(\mathbf{a},\mathbf{b}) = 1 - \frac{\left\|\mathbf{r}_a - \mathbf{r}_b\right\|}{\left\|\mathbf{r}_a + \mathbf{r}_b\right\|} = 1 - \frac{\sqrt{\sum_{i=1}^{m}\left(r_{ia} - r_{ib}\right)^2}}{\sqrt{\sum_{i=1}^{m}\left(r_{ia} + r_{ib}\right)^2}} \qquad (2.11)$$

Trong đó: $\mathbf{r}_a$ là vector thứ tự các từ trong văn bản a, $\mathbf{r}_b$ là vector thứ tự các từ trong văn bản b, *m* là số từ chung trong hai văn bản, r_{ia} là thứ tự của từ *i* trong văn bản a, r_{ib} là thứ tự của từ *i* trong văn bản b.

Như vậy, độ tương tự nội dung văn bản đại diện cho sự giống nhau về mặt từ vựng, còn độ tương tự về thứ tự các từ sẽ cung cấp thông tin về mối quan hệ giữa các từ. Những từ xuất hiện trong câu hoặc đứng trước hoặc đứng sau các từ khác đều đóng vai trò truyền đạt ý nghĩa của câu. Vì thế, độ đo giống nhau của các văn bản là sự kết hợp giữa độ đo tương tự về mặt nội dung và thứ tự của từ trong văn bản. Ta có công thức tính như sau:

$$S(\mathbf{a},\mathbf{b}) = \delta \mathrm{Sim}(\mathbf{a},\mathbf{b}) + (1-\delta)\mathrm{Sim}_R(\mathbf{a},\mathbf{b}) = \delta \frac{a \cdot b}{\|a\| \times \|b\|} + (1-\delta)\frac{\left\|\mathbf{r}_a - \mathbf{r}_b\right\|}{\left\|\mathbf{r}_a + \mathbf{r}_b\right\|} \qquad (2.12)$$

Với $0 \le \delta \le 1$, δ là giá trị có thể tinh chỉnh trong quá trình huấn luyện để có được kết quả với độ chính xác tốt nhất. Tùy thuộc vào bộ dữ liệu, khi thực nghiệm ta có thể tinh chỉnh qua nhiều mức khác nhau để lựa chọn giá trị δ phù hợp trên bộ dữ liệu huấn luyện đó. Trong luận án việc thực nghiệm phương pháp trên bộ dữ liệu tiếng Việt nhỏ với mong muốn cho kết quả trong mức độ chấp nhận được nên luận án lựa chọn tham số δ = 0.5 để xem như lấy giá trị trung bình cộng của hai độ đo trên, đó là độ đo tương tự tính theo Cosine - S(**a,b**) và độ đo thứ tự của từ Sim_R (**a,b**).

Ví dụ 2.10: Tính toán độ tương tự có tính đến trật tự của các từ của hai văn bản sau:

- a: "*Đà Nẵng là thành phố đáng sống nhất Việt Nam*"

- b: "*Đà thành*"

Ta có vector thứ tự các từ của hai văn bản là:

- $\mathbf{r}_a$ = [1 2 3 4 5 6 7 8 9 10]
- $\mathbf{r}_b$ = [1 4 0 0 0 0 0 0 0 0]

Tính trọng số TF của hai vector **a** và **b**:

a		*Đà*	*Nẵng*	*là*	*thành*	*phố*	*đáng*	*sống*	*nhất*	*Việt*	*Nam*	*tôi*	*yêu*
	TF	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

b	Từ	*Đà*	*thành*
	TF	0.5	0.5

Tính trọng số IDF:

	Đà	*Nẵng*	*là*	*thành*	*phố*	*đáng*	*sống*	*nhất*	*Việt*	*Nam*	*tôi*	*yêu*
IDF	1	1.693	1.693	1	1.693	1.693	1.693	1.693	1.693	1.693	1.693	1.693

Tính trọng số TF-IDF:

	Đà	*Nẵng*	*là*	*thành*	*phố*	*đáng*	*sống*	*nhất*	*Việt*	*Nam*	*tôi*	*yêu*
a	0.1	0.17	0.17	0.1	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
b	0.5	0	0	0.5	0	0	0	0	0	0	0	0

Tính độ tương tự giữa hai vector **a** và **b**:

$$\text{Sim}(\mathbf{a},\mathbf{b}) = \frac{0.5\times0.1+0.5\times0.1}{\sqrt{0.5^2+0.5^2}\times\sqrt{0.1^2+0.1^2}} = \frac{0.1}{0.1} = 1$$

Tính độ tương tự thứ tự từ giữa hai vector a và b:

$$\text{Sim}_R(\mathbf{a},\mathbf{b}) = 1 - \frac{\sqrt{(1-1)^2+(2-4)^2+(3-0)^2+...+(10-0)^2}}{\sqrt{(1+1)^2+(2+4)^2+(3+0)^2+...+(10+0)^2}} = \frac{19.60}{20.49} = 0.04$$

Với δ = 0.5, áp dụng công thức tính độ tương tự toàn văn bản ta có kết quả:

$$S(\mathbf{a},\mathbf{b}) = 0.5\times1+(1-0.5)\times0.04 = 0.52$$

Mở rộng với kho dữ liệu văn bản, khi xét đến vị trí hay thứ tự các từ trong tập từ chung đã được sắp xếp theo trật tự nhất định bằng cách đánh chỉ mục sắp xếp các văn bản theo thứ tự và các từ chung cũng được sắp xếp theo thứ tự của từ xuất hiện trong một văn bản và thứ tự của các văn bản để tạo thành tập từ chung T chứa các từ đơn riêng biệt theo một trật tự nhất định. Khi có văn bản cập nhật vào kho dữ liệu thì

văn bản được đánh chỉ mục tiếp theo và các từ cũng được bổ sung vào sau của tập từ chung. Với cách sắp xếp này, ta tạo được vector tập từ chung chứa các từ đã có thứ tự. Với văn bản kiểm tra cũng xét từng từ đơn theo thứ tự xuất hiện từ đầu đến cuối văn bản để xác định vector thứ tự từ của văn bản đó. Như vậy, với cách sắp xếp này thì sẽ duy trì được trật tự từ một cách thống nhất, tức là các vector đặc trưng văn bản đã có thứ tự nhất định nên không làm ảnh hưởng đến kết quả tính khoảng cách của hai vector.

Như vậy, nhằm tăng độ chính xác trong quá trình huấn luyện thực nghiệm, luận án đã bổ sung giai đoạn tính độ tương tự theo trật tự các từ trong văn bản để đảm bảo độ chính xác cao hơn. Quy trình để so sánh độ tương tự giữa hai văn bản có xét đến trật tự các từ được thực hiện qua các bước sau:

- **Bước 1:** Xác định danh sách tập từ chung cho hai văn bản. Tập từ chung này bao gồm tất cả những từ/cụm từ phân biệt có trong hai văn bản.

- **Bước 2:** Tính trọng số từ chỉ mục của các vector bằng phương pháp TF-IDF nhằm tăng độ phủ của từ/cụm từ. Kỹ thuật tính tần suất đặc trưng dựa trên tần số từ khóa và tần số văn bản nghịch đảo TF-IDF cho phép nhấn mạnh vai trò của từ/cụm từ trong nội dung của văn bản.

- **Bước 3:** Tìm vị trí của từ/cụm từ có trong tập từ chung để đưa ra vector thứ tự từ trong mỗi văn bản.

- **Bước 4:** Áp dụng các công thức tính độ tương tự Cosine, độ tương tự trật tự từ và độ tương tự tổng hợp giữa hai văn bản.

Phần sau đây, luận án đề xuất các thuật toán tính toán độ tương tự văn bản dựa trên mô hình vector theo đơn vị từ và câu, có xét đến yếu tố trật tự của từ để có thể tăng độ chính xác khi tính độ tương tự văn bản. So sánh hai phương pháp này dựa trên kết quả thực nghiệm trên bộ dữ liệu tiếng Việt từ các luận văn tốt nghiệp và có những nhận xét để làm tiền đề cho các nghiên cứu và đề xuất tiếp theo.

2.3.2.2. Tính độ tương tự có yếu tố trật tự các từ trong văn bản

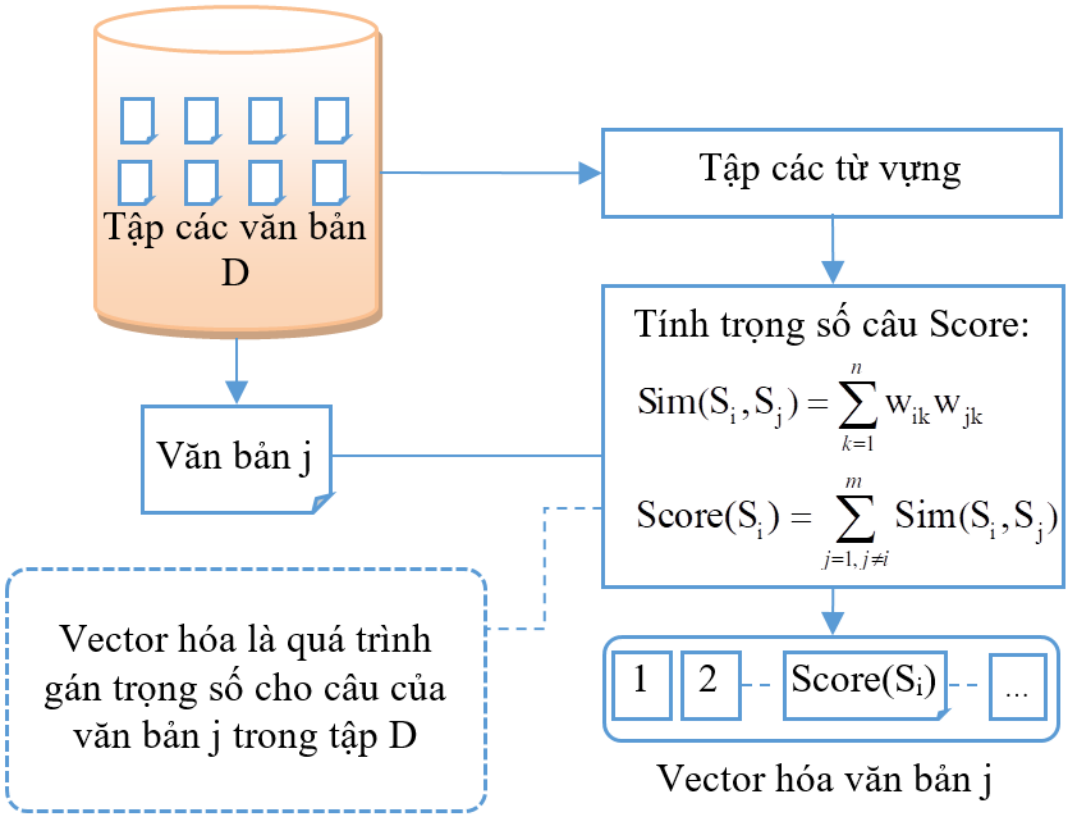
Phương pháp đề xuất với mục đích là tính mức độ giống nhau của văn bản kiểm tra với tập văn bản mẫu cho trước, trong đó có xét đến yếu tố trật tự các từ trong văn bản.

Thuật toán 2.1. Tính độ tương tự có yếu tố trật tự các từ trong văn bản

	Đầu vào: Tập văn bản cho trước và văn bản kiểm tra. **Đầu ra:** Độ tương tự giữa văn bản kiểm tra với các văn bản cho trước. **Xử lý:** Thực hiện qua các bước sau:
1	Bước 1: *Tiền xử lý* (Tách từ đơn, loại bỏ từ dừng, tạo danh sách từ vựng...)
2	Bước 2: *Tính trọng số các từ chỉ mục* w_{td}=TF-IDF *(theo công thức 2.4), cụ thể như sau:*
3	- Tính trọng số TF: $\mathrm{tf}(t,d)=\frac{N_{t,d}}{N_d}$ *// Từ chỉ mục t, văn bản d, $N_{t,d}$ là số lần xuất hiện của t trong d, N_d tổng số từ trong văn bản d*
4	- Tính trọng số IDF: $\mathrm{idf}(t,D)=1+\ln\left(\frac{N}{N_t}\right)$ *// D là tập văn bản, N: Số văn bản trong tập D, N_t: Số văn bản có từ t xuất hiện.*
5	- Tính trọng số TF-IDF của từ chỉ mục *t* trong văn bản d theo công thức: $w_{td}=\mathrm{tf}(t,d)\times \mathrm{idf}(t,D)$
6	Bước 3: *Xây dựng ma trận trọng số Từ/Văn bản biểu diễn tập văn bản D và vector văn bản kiểm tra* **q**
7	Bước 4: *Tính độ tương tự giữa vector văn bản kiểm tra* **q** *với văn bản* **k** *trong tập D bằng độ đo Cosine (theo công thức 2.5)* $\mathrm{Sim}(\mathbf{q},\mathbf{k})=\frac{\sum_{i=1}^{n} q_i k_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} k_i^2}}$
8	Bước 5: *Tạo vector thứ tự giữa 2 văn bản là* $\mathbf{r}_q$ *và* $\mathbf{r}_k$.
9	Bước 6: *Tính độ tương tự thứ tự từ giữa 2 vector* **q** *và* **k** *(theo công thức 2.11)* $\mathrm{Sim}_R(\mathbf{q},\mathbf{k})=1-\frac{\sqrt{\sum_{i=1}^{m}\left(r_{iq}-r_{ik}\right)^2}}{\sqrt{\sum_{i=1}^{m}\left(r_{iq}+r_{ik}\right)^2}}$
10	Bước 7: *Tính độ tương tự tổng hợp giữa 2 vector* **q** *và* **k** *(theo công thức 2.12)* $S(\mathbf{q},\mathbf{k})=\delta\mathrm{Sim}(\mathbf{q},\mathbf{k})+(1-\delta)\mathrm{Sim}_R(\mathbf{q},\mathbf{k})$
11	Lặp lại từ bước 4 đến bước 7 để tính độ tương tự giữa vector văn bản kiểm tra **q** với các văn bản còn lại trong tập văn bản D

2.3.2.3. Tính độ tương tự dựa trên mô hình vector đơn vị câu

Phương pháp được dùng phổ biến để tính độ tương tự văn bản dựa trên mô hình vector với các trọng số đặc trưng cho từ/cụm từ (n-gram từ) sử dụng độ đo Cosine để tính độ tương tự. Như đã trình bày ở phần trước, phương pháp này đơn giản và cho kết quả khả quan. Tuy nhiên khi tính toán phương pháp này sẽ duyệt qua tất cả các từ chỉ mục ảnh hưởng đến việc xử lý. Để số lần duyệt qua tập văn bản ít hơn, luận án đề xuất thuật toán tính độ tương tự của hai văn bản dựa trên đơn vị là câu. Dưới đây là mô hình quá trình vector hóa theo đơn vị là câu.



Hình 2.4. Quá trình vector hóa theo đơn vị câu

Trọng số của một câu được tính bằng tổng độ tương tự của câu đó với tất cả các câu khác được biểu diễn trong mô hình không gian vector văn bản. Mỗi trọng số của câu được tính theo công thức sau:

$$Sim(S_i, S_j) = \sum_{k=1}^{n} w_{ik} w_{jk} \quad (2.13)$$

$$Score(S_i) = \sum_{j=1, j\neq i}^{m} Sim(S_i, S_j) \quad (2.14)$$

Trong đó: $Sim(S_i, S_j)$ là độ tương tự giữa hai câu *i* và *j*; w_{ik} là trọng số của từ thứ *k* trong câu thứ *i*; *n* là số từ chung của hai câu *i* và *j*; *m* là số câu trong văn bản bất kỳ.

Thuật toán so sánh văn bản dựa trên mô hình vector đơn vị là câu được đề xuất như sau:

Thuật toán 2.2. Tính độ tương tự dựa trên mô hình vector đơn vị câu

	Đầu vào: Tập văn bản cho trước và văn bản kiểm tra. **Đầu ra:** Độ tương tự giữa văn bản kiểm tra với các văn bản cho trước. **Xử lý:** Thực hiện qua các bước sau:
1	Bước 1: *Tiền xử lý* (Tách câu, tách từ đơn, tạo danh sách từ vựng...)
2	Bước 2: *Xây dựng ma trận trọng số Từ/Văn bản bằng cách tính trọng số TF-IDF*
3	Bước 3: *Tính trọng số của các câu trong vector văn bản* **q**
4	- Tính độ tương tự giữa hai câu trong văn bản kiểm tra, với w_{ik} là trọng số của từ thứ *k* trong câu thứ *i* theo công thức: $Sim(S_q, S_j) = \sum_{k=1}^{n} w_{qk} w_{jk}$
5	- Tính trọng số của câu trong văn bản kiểm tra theo công thức: $Score(S_q) = \sum_{j=1, j\neq q}^{m} Sim(S_q, S_j)$
6	Bước 4: *Xây dựng vector trọng số câu cho văn bản kiểm tra* **q**
7	Bước 5: *Tính trọng số của câu trong tập văn bản D cho trước*
8	- Tính độ tương tự giữa hai câu trong một văn bản, với w_{ik} là trọng số của từ thứ *k* trong câu thứ *i* theo công thức: $Sim(S_i, S_j) = \sum_{k=1}^{n} w_{ik} w_{jk}$
9	- Tính trọng số của câu trong văn bản bất kỳ theo công thức: $Score(S_i) = \sum_{j=1, j\neq i}^{m} Sim(S_i, S_j)$
10	Bước 6: *Xây dựng ma trận trọng số của Câu/Văn bản cho toàn bộ tập văn bản*
11	Bước 7: *Tính độ tương tự giữa 2 văn bản theo Cosine (công thức 2.10)*

2.3.2.4. Nhận xét

Qua nghiên cứu tính toán độ tương tự văn bản dựa trên các độ đo Cosine, Jaccard, Manhattan, Levenshtein, luận án đã thực nghiệm, cài đặt các độ đo đó để tính độ tương tự văn bản được biểu diễn theo mô hình vector. Trên cơ sở đó, luận án đã đề xuất phương pháp so sánh văn bản dựa trên mô hình vector với đơn vị là từ và câu với trọng số đặc trưng được tính theo TF-IDF và sử dụng độ đo Cosine để tính độ tương tự, trong đó có tính đến độ tương tự theo trật tự từ nhằm tăng độ chính xác. Bên cạnh đó, chúng tôi tạo ra bộ dữ liệu để thực nghiệm các thuật toán dựa trên độ đo Cosine, Jaccard, Manhattan, Levenshtein và bộ dữ liệu gồm các luận văn tốt nghiệp để đánh giá phương pháp đề xuất. Nội dung thực nghiệm và đánh giá được trình bày ở phần sau.

2.4. Đánh giá các phương pháp dựa trên mô hình vector

2.4.1. Tạo bộ dữ liệu để đánh giá các thuật toán

2.4.1.1. Để đánh giá các phương pháp tính độ tương tự văn bản

Chúng tôi tạo bộ dữ liệu mẫu gồm các văn bản tiếng Việt bằng cách tạo hai tài liệu A và B với nội dung hoàn toàn khác nhau bằng cách tạo ra mỗi tài liệu có 1.000 từ riêng biệt. Sau đó, chọn ngẫu nhiên trong tài liệu A có 200 từ (chiếm 20% tổng số từ của tài liệu) để thay thế cho 200 từ được chọn ngẫu nhiên trong tài liệu B. Như vậy, chúng ta có thể ước lượng chính xác tỷ lệ giống nhau giữa chúng là 20%. Tương tự, chúng tôi đã tạo ra các tài liệu có tỷ lệ giống nhau là 40%, 60%, 80% và 100%.

Để kiểm tra tính chính xác của các thuật toán, khi so sánh một tài liệu bất kỳ với tài liệu đầu tiên (Doc_1.docx) thì kết quả tính toán của các thuật toán được so sánh với giá trị ước lượng. Dưới đây là bảng mô tả các trường hợp thử nghiệm.

Bảng 2.1. Các tài liệu mẫu để so với giá trị ước lượng

Trường hợp thử nghiệm	Tài liệu	Ước lượng tỷ lệ giống nhau (%)
TH1	Doc_2.docx	0
TH2	Doc_3.docx	20
TH3	Doc_4.docx	40
TH4	Doc_5.docx	60

TH5	Doc_6.docx	80
TH6	Doc_7.docx	100

2.4.1.2. Để thử nghiệm với phương pháp cải tiến sử dụng độ đo Cosine

Trong luận án đã tạo bộ dữ liệu thử nghiệm với 100 luận văn tốt nghiệp của sinh viên Khoa Công nghệ Thông tin, Trường Đại học Bách khoa, ĐHĐN. Những luận văn này sẽ được xử lý để giữ lại phần nội dung văn bản (text), những nội dung khác được loại bỏ (hình ảnh, bảng số liệu...).

2.4.2. Đánh giá các thuật toán dựa trên mô hình vector

Luận án đã xây dựng một ứng dụng với các chức năng như: Tiền xử lý văn bản, vector hóa, so khớp, hiển thị kết quả và vẽ biểu đồ. Kết quả thực nghiệm như sau:

2.4.2.1. Đánh giá các phương pháp tính độ tương tự văn bản

Kết quả thực nghiệm đã xử lý được dữ liệu để phục vụ cho quá trình so sánh văn bản, tỷ lệ so khớp có sự chênh lệch giữa các phương pháp và so với các trường hợp ước lượng không lớn.

Các thuật toán đều cho kết quả so sánh có giá trị là 100% khi hai văn bản giống nhau hoàn toàn và kết quả là 0% khi hai văn bản khác nhau hoàn toàn. Trong các trường hợp còn lại, kết quả của thuật toán so với giá trị ước lượng có độ chênh lệch tương đối, cụ thể là:

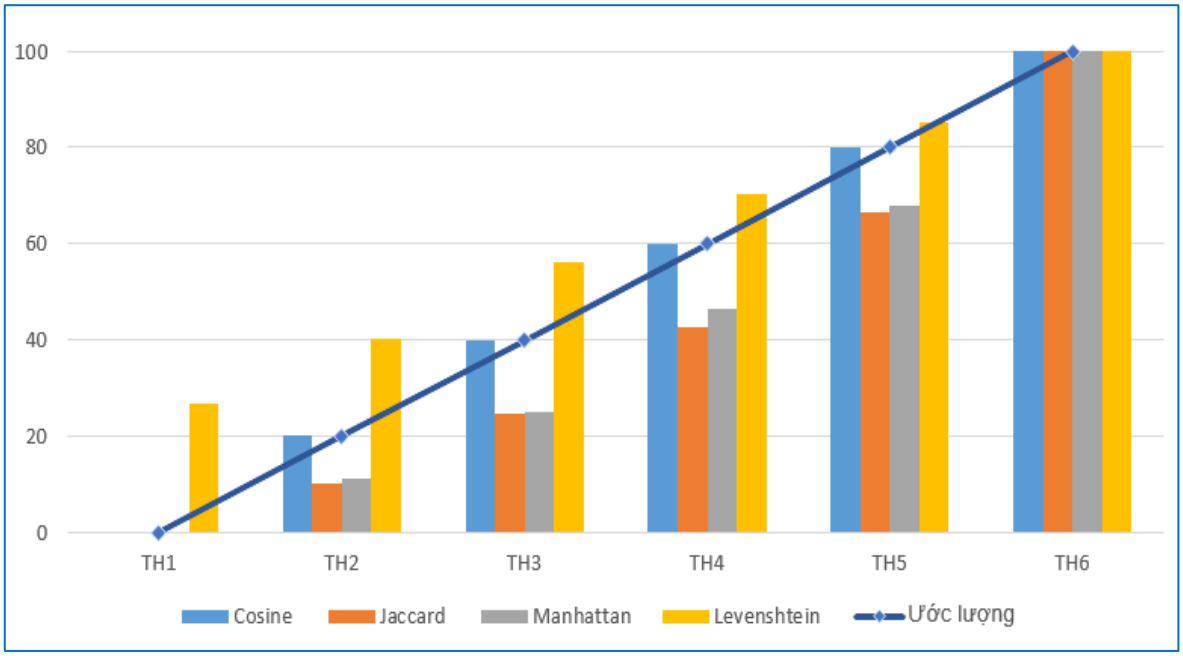
- *Dựa trên độ đo Cosine*: Thuật toán sử dụng phương pháp tần số từ, vì dữ liệu thử nghiệm được lựa chọn theo chuẩn đặt ra với tập từ vựng riêng biệt đã cho độ chính xác tuyệt đối so với giá trị ước lượng nên phương pháp này cho kết quả chính xác. Vì vậy, phương pháp tính độ tương tự văn bản dựa trên độ đo Cosine đem lại hiệu quả trong phát hiện sự giống nhau của văn bản khi so sánh theo từ vựng.
- *Dựa trên độ đo Jaccard và Manhattan*: Mặc dù hai phương pháp này có công thức tính độ tương tự khác nhau nhưng khi sử dụng tách từ đơn (n-gram bằng 1) thì cho ra hai kết quả khá tương đồng. Nếu thực nghiệm với cụm từ sử dụng bigram hoặc trigram (hoặc n lớn hơn) sẽ cho hai kết quả khác nhau dù chênh lệch không nhiều.

- *Dựa trên độ đo Levenshtein* có ưu điểm dùng để đo khoảng cách giữa hai chuỗi dựa vào ký tự, vì vậy trong trường hợp hai văn bản khác nhau hoàn toàn về từ thì cũng có thể giống nhau về ký tự và khoảng trắng, nên độ đo tương tự của hai văn bản khác nhau hoàn toàn cũng có thể lớn hơn 0%. Hơn nữa việc tính toán dựa vào độ chính xác của các ký tự nên phương pháp này sử dụng đo độ tương tự của văn bản sẽ không hiệu quả, tốn chi phí.

Dưới đây là bảng tổng hợp kết quả ghi được của các phương pháp so với các trường hợp thử nghiệm và kết quả ước lượng.

Bảng 2.2. Tổng hợp kết quả của các phương pháp

Trường hợp	Kết quả thử nghiệm tính theo độ đo (%)				Ước lượng (%)
	Cosine	Jaccard	Manhattan	Levenshtein	
TH1	0	0	0	26.86	0
TH2	20	10.21	11.11	40.27	20
TH3	40	24.51	25	56.12	40
TH4	60	42.78	46.45	70.26	60
TH5	80	66.61	67.75	85.14	80
TH6	100	100	100	100	100



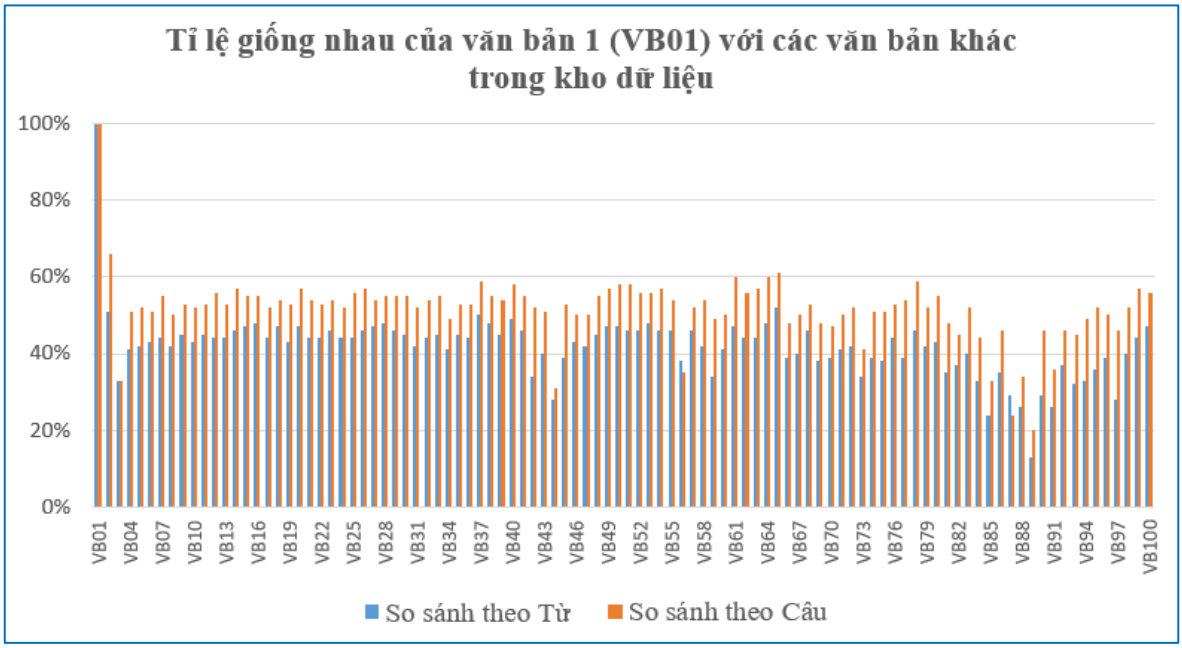
Hình 2.5. Biểu đồ so sánh kết quả thuật toán với tập tài liệu

Với kết quả trên, cho thấy luận án đã nghiên cứu cách biểu diễn văn bản theo mô hình vector, sử dụng các độ đo để tính độ tương tự của văn bản và xây dựng hệ thống thực nghiệm. Tuy nhiên, hệ thống so khớp mới thử nghiệm ở việc so sánh với bốn độ đo khác nhau trên các văn bản .doc hoặc .docx và kho dữ liệu nhỏ.

Thời gian và dung lượng tiêu tốn cho quá trình so khớp phụ thuộc vào độ dài của văn bản so khớp (tức số lượng từ vựng có trong văn bản). Với việc biểu diễn *n* văn bản và *m* từ chỉ mục sẽ tạo thành ma trận vector có số chiều *nxm*. Khi tính độ tương tự, duyệt qua tất cả các từ và các văn bản nên độ phức tạp về thời gian của thuật toán khi so sánh là O(*nm*).

2.4.2.2. Đối với phương pháp cải tiến sử dụng độ đo Cosine

Kết quả thực nghiệm trên bộ dữ liệu với 100 luận văn tốt nghiệp và cho kết quả so sánh nội dung của văn bản dựa trên đơn vị từ và câu. Dưới đây là kết quả so sánh một văn bản cần kiểm tra với các văn bản khác trong kho dữ liệu chứa 100 luận văn tốt nghiệp.



Hình 2.6. Biểu đồ so sánh văn bản theo đơn vị từ và câu

Kết quả so sánh có giá trị là 100% khi hai văn bản giống nhau hoàn toàn. Qua thực nghiệm, chúng ta nhận thấy đường so sánh theo đơn vị từ và câu khá tương tự nhau và kết quả tỷ lệ so khớp có sự chênh lệch giữa vector hóa văn bản dựa trên từ và câu là do phụ thuộc vào phương pháp và các hàm tính trọng số.

2.4.3. Nhận xét

Mặc dù, phương pháp dựa trên vector và các độ đo tương tự phổ biến như đã đề cập ở trên đã đem lại hiệu quả và giải quyết được mục tiêu bài toán là đánh giá độ tương tự các văn bản. Tuy nhiên, với phương pháp đề xuất này thì độ chính xác cũng chưa cao. Bên cạnh đó, phương pháp biểu diễn theo vector vẫn còn hạn chế về số chiều biểu diễn cho tập văn bản sẽ rất lớn nên tốn không gian lưu trữ, độ phức tạp của thuật toán khi so sánh tăng và làm giảm tốc độ tính toán.

Với những nội dung nghiên cứu đạt được ở Chương này là cơ sở để chúng tôi tiếp tục ứng dụng mô hình biểu diễn vector một cách phù hợp nhất trong phạm vi nghiên cứu của luận án. Nội dung chi tiết có liên quan được đề cập trong các chương sau.

2.5. Kết luận Chương 2

Qua các nghiên cứu và khảo sát, mô hình vector và các biến thể của mô hình này là các phương pháp được dùng phổ biến nhất để xử lý văn bản. Hầu hết các nghiên cứu sử dụng mô hình vector để biểu diễn văn bản với thành phần là các đặc trưng về từ, cụm từ (n-gram),... Trong nội dung Chương 2 đã trình bày cách tính trọng số đặc trưng của văn bản, phương pháp biểu diễn văn bản theo mô hình vector và ứng dụng các độ đo để tính độ tương tự giữa các văn bản bằng cách đo góc hoặc khoảng cách giữa các vector. Xây dựng các bộ dữ liệu tiếng Việt và ứng dụng để thử nghiệm nhằm đánh giá các kết quả đạt được, định hướng cải tiến và đề xuất phương pháp mới để có thể giải quyết bài toán đề ra đem lại hiệu quả tốt hơn. Kết quả nghiên cứu trong Chương 2 có ở các công trình công bố trên tạp chí trong nước và nước ngoài[6].

---

[6] (1) Hung Vo Trung, Ngoc Anh Nguyen, Hieu Ho Phan, Thi Dung Dang, Comparison of the Documents Based On Vector Model: A Case Study of Vietnamese Documents. American Journal of Engineering Research (AJER), Vol 6(7), 2017, (251-256).

(2) Hồ Phan Hiếu, Võ Trung Hùng, Nguyễn Thị Ngọc Anh, Một số phương pháp tính độ tương tự văn bản dựa trên mô hình vector. Tạp chí Khoa học và Công nghệ ĐHĐN, Số 11(120), 2017, (112-117).

Trên cơ sở khảo sát, phân tích, thực nghiệm và đánh giá, chúng ta có thể khẳng định phương pháp biểu diễn văn bản theo mô hình vector là thông dụng nhất. Trong Chương này đã trình bày nội dung liên quan về mô hình vector và đề xuất phương pháp tính độ tương tự giữa hai văn bản hoặc một văn bản cần kiểm tra với kho dữ liệu dựa trên mô hình vector với phương pháp đánh trọng số TF-IDF và ứng dụng các độ đo Cosine, Jaccard, Manhattan, Levenshtein để tính độ tương tự giữa các vector. Trên cơ sở đó, chúng tôi đã chọn độ đo Cosine để tính độ tương tự dựa trên đơn vị từ và câu thực nghiệm trên tập dữ liệu thực tế để chứng minh phương pháp đề xuất có thể tính toán độ tương tự của văn bản tiếng Việt.

Trong nhiệm vụ nghiên cứu, luận án không tập trung về mặt ngữ nghĩa mà chủ yếu nghiên cứu và đề xuất, cải tiến những phương pháp phổ biến và tìm ra phương pháp mới nhằm đem lại độ chính xác cũng như độ phức tạp tính toán thấp hơn so với các phương pháp khác. Luận án tiếp tục nghiên cứu và đề ra giải pháp mới để giải quyết bài toán này tối ưu hơn, đó là chuyển từ văn bản sang số để tận dụng những ưu điểm về số trong xử lý dữ liệu lớn, tìm kiếm nhanh và độ chính xác cao. Bên cạnh đó, trong phương pháp đề xuất mới, luận án ứng dụng những nghiên cứu trong phương pháp biểu diễn vector cho các chuỗi số thực làm đầu vào của bộ lọc Haar để tạo thành các DNA, lưu trữ và đánh dấu dữ liệu nhằm hỗ trợ tìm kiếm và truy xuất các tài liệu nguồn. Nội dung trọng tâm của nghiên cứu này sẽ được trình bày chi tiết trong Chương 3.

Chương 3. PHÁT HIỆN SAO CHÉP VĂN BẢN DỰA TRÊN BIẾN ĐỔI WAVELET RỜI RẠC

Chương này trình bày nội dung đề xuất theo hướng tiếp cận mới để giải quyết bài toán dựa trên phép biến đổi Wavelet rời rạc (*Discrete Wavelet Transform* - DWT) và sử dụng bộ lọc Haar. Nội dung chính sẽ trình bày vắn tắt về DWT, bộ lọc Haar và chuỗi số DNA; đề xuất ý tưởng ứng dụng bộ lọc Haar; đề xuất mô hình hệ thống phát hiện sao chép; đề xuất quy trình chuyển dữ liệu văn bản thô thành các chuỗi số thực (DNA); đề xuất các phương pháp và giải thuật xử lý; đánh giá độ phức tạp của thuật toán và so sánh với phương pháp biểu diễn bằng vector. Thực nghiệm và đánh giá kết quả đạt được để chứng minh hiệu quả của phương pháp đề xuất.

3.1. Đặt vấn đề

Như đã đề cập, nhiệm vụ External Plagiarism Detection của PAN nhằm giải quyết bài toán phát hiện sao chép trong một tài liệu văn bản bằng cách tìm sự tương tự giữa nó với các tài liệu khác trong kho dữ liệu. Với mục tiêu đề ra, luận án tập trung nghiên cứu và đề xuất phương pháp so sánh văn bản dựa trên DWT và bộ lọc Haar để tính độ tương tự của một văn bản cần kiểm tra so với các văn bản nguồn có trong kho dữ liệu. Luận án sử dụng bộ dữ liệu huấn luyện của PAN để thực nghiệm và so sánh với các kết quả đạt được. Phần sau là phát biểu bài toán phát hiện sao chép văn bản trong PAN và đề xuất ý tưởng để giải quyết bài toán này.

3.1.1. Phát biểu bài toán

Theo Meuschke and Gipp [80, 122], chúng ta có thể mô hình hóa bài toán phát hiện sao chép tài liệu văn bản như sau:

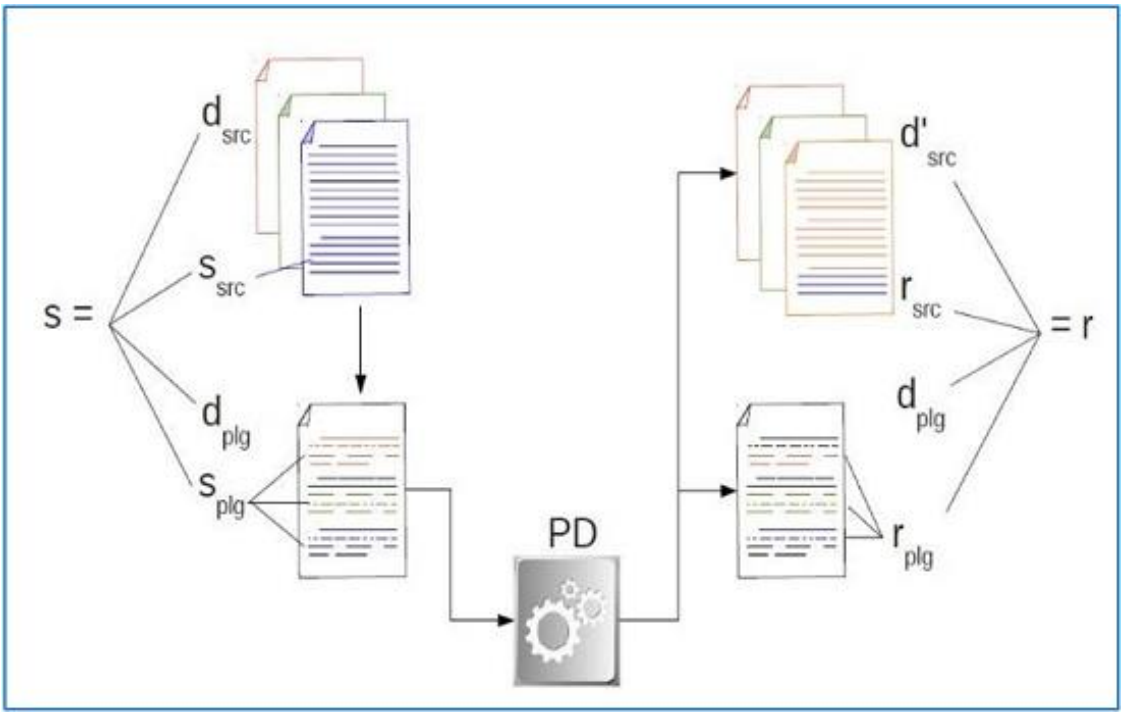
Ta định nghĩa một trường hợp sao chép $s = \{s_{plg}, d_{plg}, s_{src}, d_{src}\}$, trong đó một đoạn văn bản s_{plg} trong tài liệu d_{plg} là *sao chép* từ một đoạn văn bản s_{src} trong tài liệu d_{src} *(tài liệu nguồn).*

Với một tài liệu d_{plg} cho trước *(tài liệu cần kiểm tra),* nhiệm vụ của hệ thống phát hiện sao chép là phát hiện ra s bằng cách chỉ ra một trường hợp sao chép $r = \{r_{plg}, d_{plg}, r_{src}, d'_{src}\}$, bao gồm một đoạn văn được cho là sao chép r_{plg} trong tài liệu d_{plg}

và đoạn văn bản nguồn của nó là r_{src} trong d'_{src}, xấp xỉ nhất *(tương tự)* có thể với s. Chúng ta kết luận rằng *r phát hiện được s* nếu và chỉ nếu $s_{plg} \cap r_{plg} \neq \emptyset$, $s_{src} \cap r_{src} \neq \emptyset$ và $d_{src} = d'_{src}$. Thông thường thì đoạn văn bản r_{plg} phải *đủ dài* để tránh các trường hợp trùng lặp ngẫu nhiên, điều này có thể được thiết lập bằng một *ngưỡng t* nào đó.

Trong hệ thống phát hiện sao chép, để so sánh độ tương tự giữa các đoạn văn bản *(chunk)*, chúng ta phát hiện ra s bằng cách tìm kiếm tài liệu d'_{src} từ một tập tài liệu D rất lớn (như kho dữ liệu) và trích xuất ra r_{src} và r_{plg} từ hai tài liệu d'_{src} và d_{plg} dựa trên việc so sánh chi tiết giữa hai tài liệu.

Điểm chung của các phương pháp phát hiện sao chép là tính toán độ tương tự của văn bản T với các văn bản trong D. Việc đo độ tương tự của hai văn bản thường dựa trên việc đo độ tương tự giữa các đoạn văn bản hay thành phần đơn vị *(chunk)* trong T với các chunk của các văn bản trong D. Có nhiều giải pháp phát hiện sao chép khác nhau xác định thành phần đơn vị khác nhau. Các thành phần đơn vị này có thể là từ (*word*), cụm n từ (*phrase*), câu (*sentence*) hoặc cả đoạn (*paragraph*).



Hình 3.1. Mô tả cách xử lý để phát hiện sao chép văn bản

3.1.2. Đề xuất ý tưởng

Ngày nay, với sự phát triển vượt bậc của khoa học công nghệ thì nghiên cứu liên ngành đang là xu thế phát triển tất yếu và giữ vai trò quan trọng để giải quyết các vấn đề của khoa học và thực tiễn. Với việc tính đến xử lý dữ liệu lớn trong phát hiện sự giống nhau dựa trên nội dung văn bản thuộc lĩnh vực NLP, chúng tôi nhận thấy trong lĩnh vực xử lý tín hiệu số, tin y sinh thường xử lý với dữ liệu rất lớn, tốc độ nhanh và đã có nhiều kết quả đem lại hiệu quả trong tính toán. Chính vì vậy, trong luận án đề xuất ý tưởng để chuyển văn bản thành chuỗi tín hiệu số và xử lý, tính toán, so khớp trên dữ liệu này. Nội dung của Chương này tập trung trình bày đề xuất một phương pháp tiếp cận mới để so khớp văn bản dựa trên DWT để phát hiện sự giống nhau của văn bản.

Đánh giá mức độ giống nhau là tìm sự tương tự, độ tương tự thường nằm trong khoảng giữa [0, 1], nếu độ tương tự càng gần 1 thì khả năng gần giống nhau là cao và ngược lại [106]. Luận án đề xuất ý tưởng tìm sự tương tự bằng cách loại bỏ sự khác nhau nhỏ giữa hai văn bản (hay gọi là sự bất thường) và loại bỏ những bất thường này để giữ lại được những nội dung giống nhau, từ đó tìm ra được độ tương tự giữa hai văn bản.

Trong xử lý tín hiệu số, việc tìm kiếm và lọc ra những tín hiệu bất thường thường dùng các phép biến đổi tín hiệu và thông qua các bộ lọc. Để áp dụng trong việc đánh giá mức độ giống nhau của văn bản, những thách thức lớn đặt ra đó là: 1) Nghiên cứu để tìm cách chuyển đổi văn bản thành tín hiệu số và đảm bảo đầy đủ nội dung thông tin của văn bản; 2) Nghiên cứu sử dụng các phương pháp xử lý tín hiệu số phù hợp để tính toán; 3) Nghiên cứu áp dụng các độ đo để tính toán, lọc ra các tín hiệu bất thường để phát hiện các tín hiệu giống nhau; 4) Liên kết và truy xuất lại nội dung để đánh giá được mức độ giống nhau của các văn bản.

3.2. Cơ sở lý thuyết về DWT và bộ lọc Haar

3.2.1. Cơ sở lý thuyết về DWT

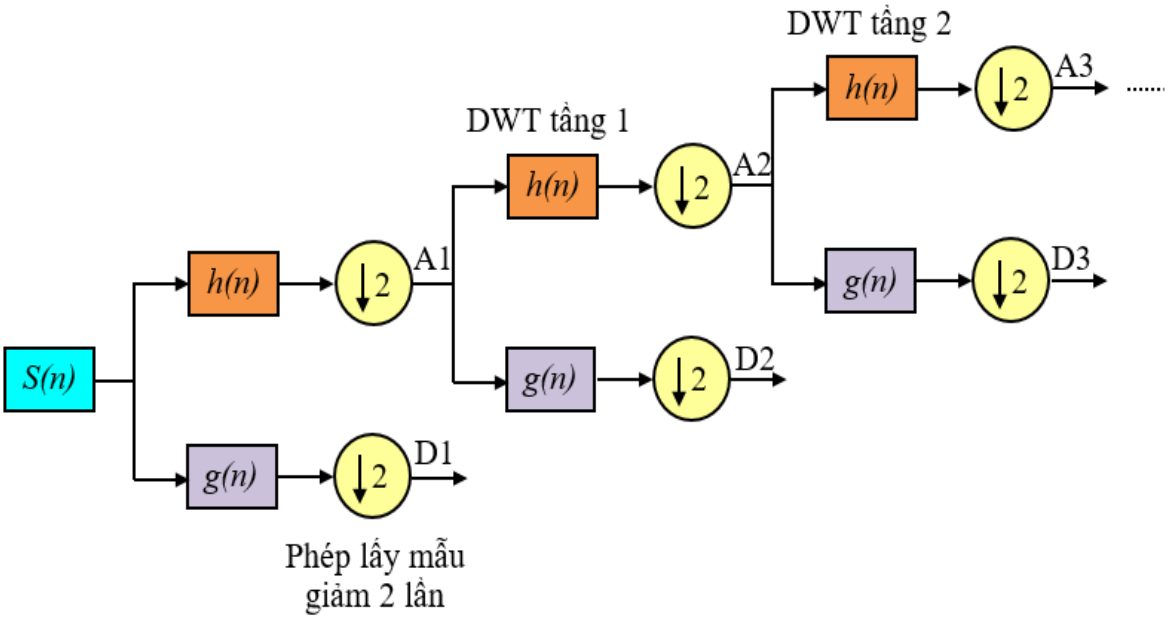
Phương pháp biến đổi Wavelet rời rạc (DWT) cho phép mã hóa đơn giản và nhanh nên được sử dụng rất hiệu quả trong xử lý tín hiệu số. DWT cũng được các nhà khoa học quan tâm và ứng dụng trong lĩnh vực khai phá dữ liệu [22, 63], truy

xuất thông tin [94], độ tương tự trong phân cụm, phân lớp văn bản [2, 135]… DWT cho tín hiệu một chiều được mô tả sơ lược như sau: Tín hiệu được chia thành hai phần, đó là phần tần số cao và phần tần số thấp; thành phần tần số thấp lại được chia tiếp thành hai phần có tần số cao và thấp; các bước như vậy gọi là lấy mẫu giảm xuống. Bên cạnh đó, độ phức tạp của mã hóa là tuyến tính và hỗ trợ nhiều mức phân giải. Phân tích đa phân giải sử dụng các kỹ thuật lọc tín hiệu số trong quá trình phân tích để lọc nhiễu và xác định tín hiệu bất thường [76, 127]. Phân tích đa phân giải có khả năng như hai bộ lọc tín hiệu, trong đó mỗi một tín hiệu được phân tích thành hai thành phần gồm: Thành phần xấp xỉ A tương ứng với thành phần tần số thấp và thành phần chi tiết D tương ứng với thành phần tần số cao, thông qua bộ lọc thông thấp sử dụng hàm tỷ lệ $\Phi(x)$ và bộ lọc thông cao sử dụng hàm Wavelet $\psi(x)$ như Hình 3.2. Mối quan hệ giữa hàm tỷ lệ và hàm Wavelet được cho bởi [76, 117, 127]:

$$\phi(x) = \sum_{k=0}^{N-1} C_k . \phi(2x - k) \tag{3.1}$$

$$\psi(x) = \sum_{k=0}^{N-1} (-1)^k C_k . \phi(2x + k - N + 1) \tag{3.2}$$

trong đó, N là số mẫu trong tín hiệu, C_k là chỉ số vô hướng nhằm xác định các hệ số tỷ lệ.



Hình 3.2. Phân tích đa phân giải sử dụng DWT

Các phép lọc được tiến hành với nhiều tầng khác nhau và để giảm khối lượng tính toán, khi qua mỗi bộ lọc, tín hiệu được lấy mẫu giảm xuống hai lần. Ứng với mỗi tầng, tín hiệu có độ phân giải khác nhau nên DWT được gọi là phân tích đa phân giải. Tại mỗi tầng, biểu thức của phép lọc được cho bởi công thức sau:

$$y_{high}(n) = \sum_{k=0}^{N-1} s(k).g(2n-k) \tag{3.3}$$

$$y_{\text{low}}(n) = \sum_{k=0}^{N-1} s(k).h(2n-k) \tag{3.4}$$

trong đó, *s(n)* là tín hiệu; *h(n)* là đáp ứng xung của các bộ lọc thông thấp tương ứng với hàm tỷ lệ *Φ(n)*; *g(n)* là đáp ứng xung của các bộ lọc thông cao tương ứng với hàm Wavelet *ψ(n)*.

Hai bộ lọc thông thấp và thông cao liên hệ nhau theo hệ thức:

$$h(N-1-n) = (-1)^n g(n) \tag{3.5}$$

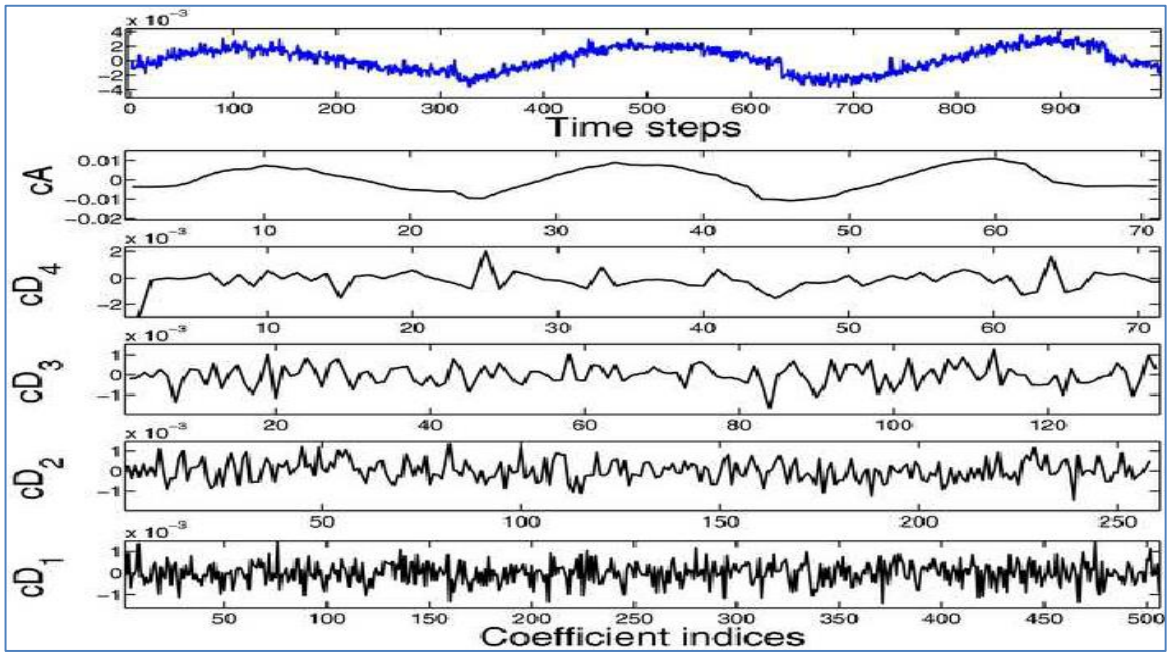
Như đề cập ở trên, chúng ta thấy DWT phân tách tín hiệu thành các thành phần xấp xỉ và chi tiết (hay hệ số xấp xỉ và hệ số chi tiết). Sau khi phân tách tín hiệu thì tiến hành tái tạo lại tín hiệu ban đầu sao cho không làm mất thông tin bằng cách kết hợp các thành phần xấp xỉ và chi tiết lại với nhau thông qua biến đổi Wavelet rời rạc ngược (*Inverse Discrete Wavelet Transform* - IDWT).

Tín hiệu *s(n)* có thể được tái tạo lại theo các bước ngược lại được cho bởi công thức:

$$s(n) = \sum_{k=0}^{N-1} y_{high}(k).g(2n-k) + y_{\text{low}}(k).h(2n-k) \tag{3.6}$$

trong đó, $y_{high}(k)$ và $y_{low}(k)$ lần lượt là tín hiệu đầu ra sau khi đi qua các bộ lọc thông cao và bộ lọc thông thấp.

Dưới đây là một ví dụ minh họa về đường sóng tín hiệu DWT:



Hình 3.3. Đường tín hiệu qua DWT [50]

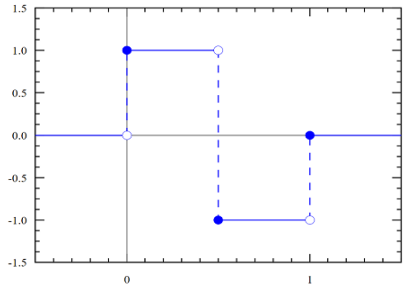
Tín hiệu trên cùng là tín hiệu gốc theo thời gian (có dạng sóng màu xanh). Khi tín hiệu gốc qua DWT sẽ chia thành hai phần: Tín hiệu qua bộ lọc thông thấp tương ứng với hàm tỷ lệ $\Phi(x)$ sẽ tạo thành đường xu hướng (*trendline*) của tín hiệu (hình cA) và tín hiệu khi qua bộ lọc thông cao sử dụng hàm Wavelet $\psi(x)$ sẽ cho ra đường sóng dao động của tín hiệu gốc ở các cấp độ phân giải khác nhau (hình cD_1, cD_2, cD_3, cD_4). Như vậy, bộ lọc thông thấp có nhiệm vụ lọc các tín hiệu nhiễu hay tín hiệu bất thường để giữ lại đường tín hiệu tổng thể của tín hiệu gốc. Vì vậy, luận án lựa chọn phương pháp DWT để tạo ra các tín hiệu chuỗi số thực sau khi loại bỏ các tín hiệu bất thường.

3.2.2. Bộ lọc Haar

Bộ lọc Haar được nhà toán học người Hungary Alfred Haar giới thiệu và đưa vào ứng dụng từ năm 1910 [121]. Haar là dạng sóng không liên tục, giống như một hàm bước nhảy. Trong biến đổi Wavelet rời rạc, đường Haar Wavelet hay được gọi là bộ lọc Haar được sử dụng phổ biến trong khai phá dữ liệu chuỗi thời gian và lập chỉ mục. Đường Haar Wavelet là một trong những ví dụ đầu tiên của sự chuyển đổi Wavelet dạng sóng nhỏ và trực giao [99]. Đường Haar Wavelet có dạng sóng vuông, có hai hàm (hàm tỷ lệ và hàm Wavelet) được định nghĩa theo các công thức bởi [21]:

$$\psi_i^j = \psi(2^j x - i) \qquad i = 0, ..., 2^j - 1 \tag{3.7}$$

$$\text{với} \qquad \psi(t) = \begin{cases} 1 & \text{với } 0 \le t < \frac{1}{2} \\ -1 & \text{với } \frac{1}{2} \le t < 1 \\ 0 & \text{với các trường hợp khác} \end{cases} \tag{3.8}$$



Hình 3.4. Đường sóng Haar Wavelet

Do ưu điểm về độ phức tạp tính toán thấp, bộ lọc Haar đã được sử dụng chủ yếu để nhận dạng mẫu, xử lý ảnh, xử lý tín hiệu số... [76, 127]. Tốc độ vận hành của bộ lọc Haar hoạt động nhanh nhất trong tất cả các sóng Wavelet vì hệ số của hàm Haar nhận được là 1 hoặc -1. Bộ lọc Haar có thể phân rã tín hiệu thành các thành phần khác nhau của miền tần số. Sự biến đổi Wavelet rời rạc một chiều (1-D DWT) phân tách một dãy tín hiệu đầu vào thành hai thành phần (đó là thành phần trung bình và thành phần chi tiết) bằng phép tính với bộ lọc thông thấp và bộ lọc thông cao [68].

Với dữ liệu đầu vào là một chuỗi các số, biến đổi Haar thực hiện ghép cặp hai số liên tiếp nhau trong chuỗi số này. Hiệu của các cặp số được lưu trữ lại, trong khi tổng của chúng tạo thành một chuỗi mới được sử dụng như là dữ liệu đầu vào cho lần biến đổi tiếp theo. Quá trình này được lặp lại và cuối cùng cho kết quả là một chuỗi số gồm giá trị hiệu của các cặp số qua các lần biến đổi và một giá trị tổng. Biến đổi Haar là một dạng nén đơn giản bao gồm các xử lý như tính trung bình, tính hiệu, lưu trữ các hệ số chi tiết, loại bỏ bớt dữ liệu và tái cấu trúc lại dữ liệu sao cho dữ liệu sau khi xử lý tương tự như dữ liệu ban đầu [105].

Vì vậy, việc nghiên cứu sử dụng bộ lọc Haar để biến đổi tín hiệu chuỗi thời gian thực thành các DNA để tính toán, xử lý và lọc tín hiệu là hướng nghiên cứu khả thi và có khả năng mang lại hiệu quả cao.

3.2.3. Chuỗi DNA

DNA (*DeoxyriboNucleic Acid*) là thuật ngữ trong sinh học, nó là phân tử mang thông tin di truyền mã hóa cho hoạt động sinh trưởng và phát triển của các dạng sống bao gồm cả virus. DNA trong lĩnh vực tin sinh học bao gồm các kỹ thuật lưu trữ, khai phá dữ liệu, tìm kiếm và thao tác với dữ liệu sinh học. Các kỹ thuật này mang đến những ứng dụng rộng rãi trong ngành khoa học máy tính, đặc biệt là các thuật toán tìm kiếm chuỗi, học máy và lý thuyết cơ sở dữ liệu. Trong đó, thuật toán tìm kiếm chuỗi hay so khớp để tìm sự xuất hiện của một dãy các ký tự trong một dãy các ký tự lớn hơn được ứng dụng nhiều và giải quyết tốt các bài toán xử lý dữ liệu lớn trong tin sinh học [9].

Chuỗi DNA đại diện cho sự riêng biệt, vì thế trong nghiên cứu của mình, luận án mô tả các chuỗi số thực mang thông tin đặc trưng có thể phân biệt được mức độ khác nhau giữa chúng được gọi là các chuỗi DNA. Luận án đề xuất các thuật toán để chuyển văn bản thành các chuỗi DNA riêng biệt nhưng vẫn giữ được đặc trưng của ký tự văn bản, tổ chức lưu trữ và đề xuất các giải thuật tìm kiếm chuỗi hiệu quả trong xử lý dữ liệu lớn để xử lý phát hiện giống nhau.

3.3. Đề xuất mô hình hệ thống phát hiện sao chép

3.3.1. Giới thiệu

Qua các khảo sát cho thấy phương pháp DWT được nghiên cứu, ứng dụng và đem lại hiệu quả trong xử lý tín hiệu số như mã hóa tiếng nói, xử lý ảnh, ứng dụng trong lọc nhiễu, nhận dạng và trong các lĩnh vực khác như công nghiệp, điện tử, viễn thông, y học... [89, 132]. Ở Việt Nam, các nghiên cứu về DWT ứng dụng chủ yếu trong lĩnh vực xử lý ảnh, một số nghiên cứu liên quan trong xử lý tín hiệu như phân tích các dị thường trọng lực, xử lý nhiễu tín hiệu điện tim, phát hiện nhiễu loạn điện áp, thủy vân số... [29, 98].

Đặc biệt, trong lĩnh vực xử lý ảnh, DWT và bộ lọc Haar được ứng dụng khá phổ biến trong nhận dạng hình ảnh, nâng cao và đánh giá chất lượng của ảnh số, nén ảnh để giảm dung lượng của hình ảnh nhằm tăng hiệu suất lưu trữ và chuyển tải hình ảnh, các kỹ thuật phát hiện và trích xuất văn bản từ video và hình ảnh sử dụng DWT...

[43, 55, 61, 92]. Tuy nhiên, qua khảo sát và nghiên cứu, chúng tôi hầu như chưa thấy công trình nghiên cứu về DWT được ứng dụng trong lĩnh vực xử lý văn bản.

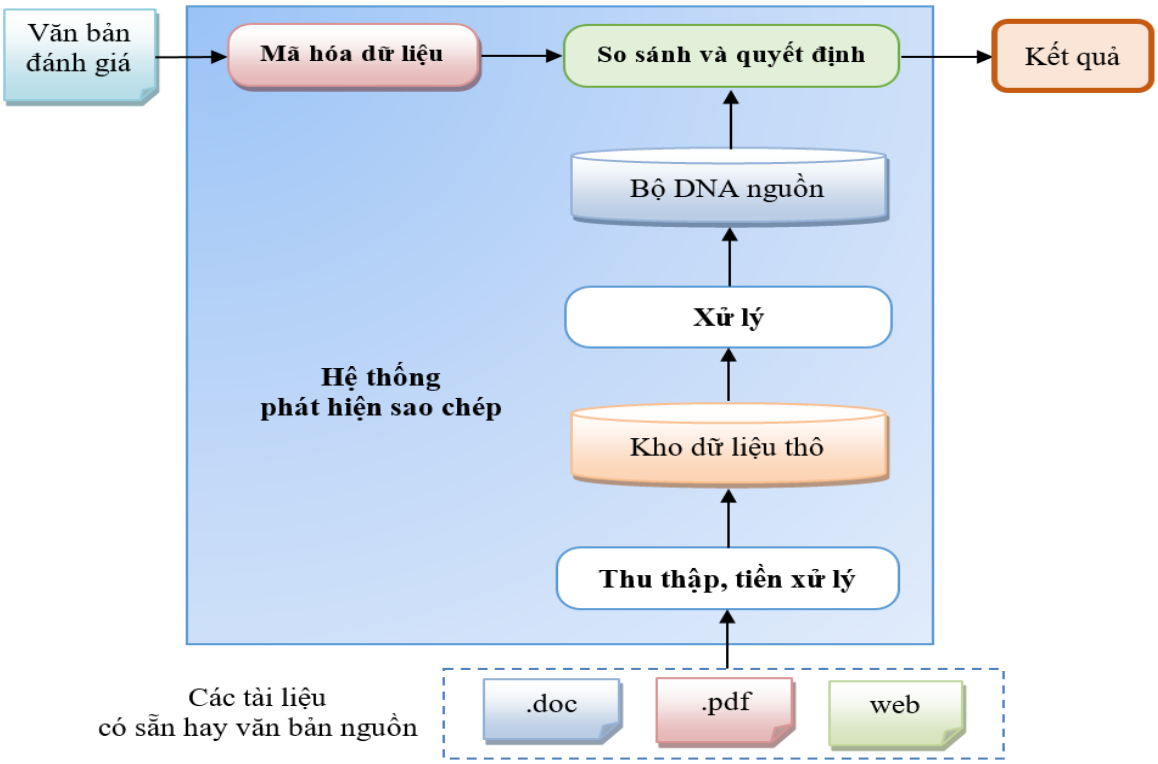
Qua nghiên cứu về DWT và bộ lọc Haar, trong luận án đề xuất phương pháp nhằm chuyển đổi nội dung văn bản thành dạng chuỗi thời gian thực (thông qua bộ số hóa) và sử dụng bộ lọc Haar trong DWT để loại bỏ các mẫu bất thường, dữ liệu văn bản được chuyển đổi và biểu diễn thành những dãy số thực dưới dạng vector $\mathbf{x}=[x_1\ x_2\ ...\ x_N]$. Trong luận án sẽ phân tích cơ sở lý thuyết và phát triển bộ lọc Haar nhằm tạo ra các DNA để ứng dụng trong hệ thống xử lý phát hiện giống nhau.

3.3.2. Đề xuất mô hình hệ thống áp dụng cho phương pháp dựa trên DWT

Như ý tưởng của bài toán đã nêu, qua nghiên cứu về DWT và các nghiên cứu liên quan để phát hiện, đánh giá mức độ giống nhau của văn bản, trong luận án đề xuất một phương pháp hoàn toàn mới để phát hiện sự giống nhau của văn bản dựa trên DWT. Phương pháp thực hiện qua các bước chính như: 1) Chuyển các tài liệu gốc sẵn có trong kho dữ liệu thành một tập các chuỗi tín hiệu số được gọi là các DNA nguồn thông qua DWT; 2) Văn bản bất kỳ cần kiểm tra cũng được tạo ra thành DNA dựa trên DWT và tính toán khoảng cách Euclid nhỏ nhất từ DNA này đến các DNA nguồn; 3) Cuối cùng là so sánh với một mức ngưỡng thích hợp để đưa ra sự giống nhau giữa văn bản được kiểm tra với văn bản nguồn trong kho dữ liệu.

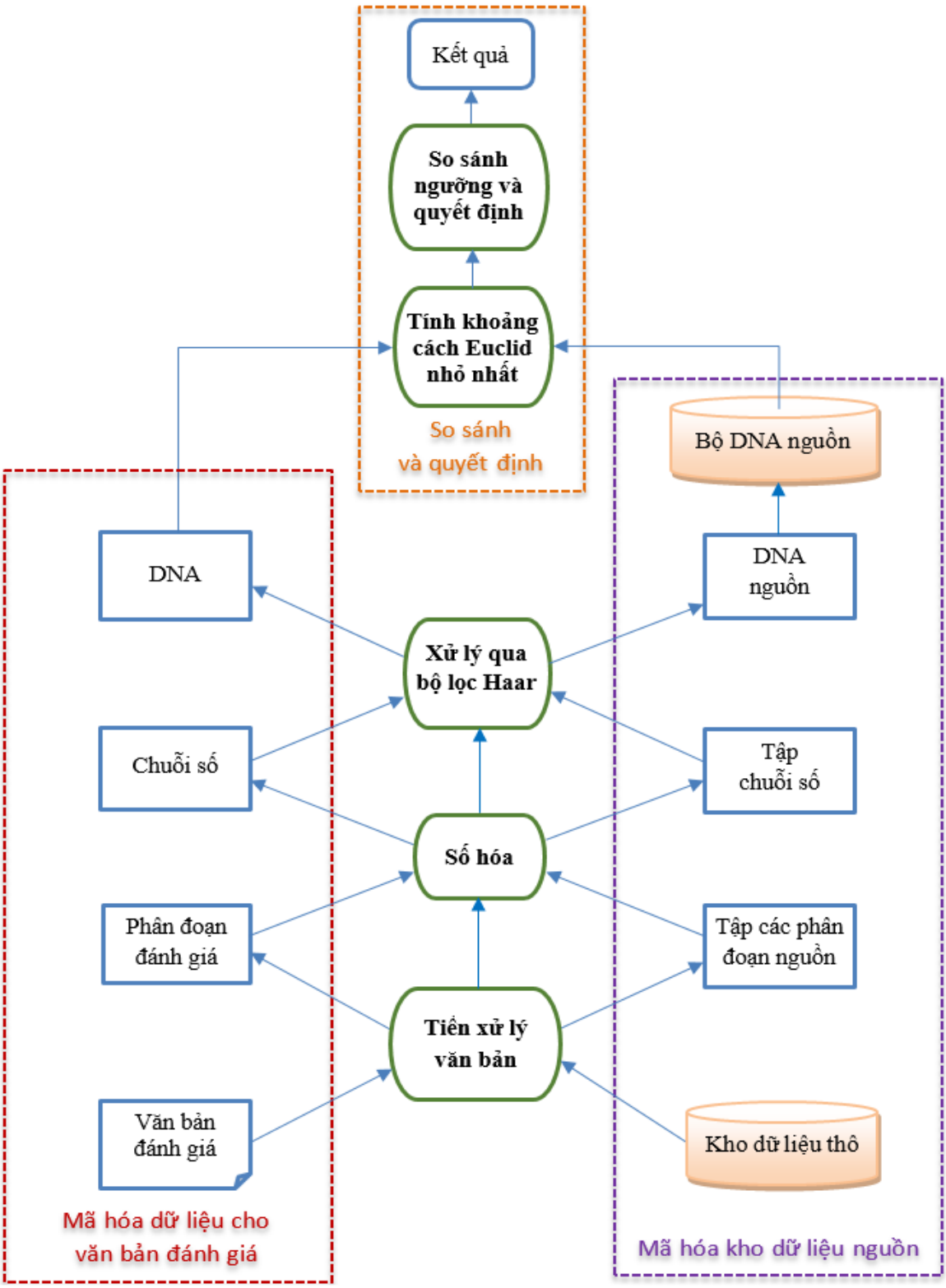
Trong phần này, luận án tập trung nghiên cứu đề xuất mô hình hệ thống phát hiện sự giống nhau của văn bản; đề xuất quy trình mã hóa văn bản, chuyển đổi văn bản thành tập các chuỗi tín hiệu số (DNA) thông qua DWT và bộ lọc Haar; đề xuất các phương pháp và giải thuật xử lý; xây dựng bộ dữ liệu tiếng Việt để thử nghiệm; đánh giá kết quả trên bộ dữ liệu của PAN và bộ dữ liệu tiếng Việt.

Với mục tiêu cuối cùng là để nghiên cứu, xây dựng một hệ thống phát hiện sự giống nhau của văn bản, luận án đã nghiên cứu, khảo sát các tài liệu tổng quan về hệ thống phát hiện sao chép văn bản, cũng như những nghiên cứu về DWT, bộ lọc Haar. Trên cơ sở đó, trong luận án đề xuất mô hình tổng quan của hệ thống như Hình 3.5 dưới đây.



Hình 3.5. Đề xuất mô hình hệ thống phát hiện sự giống nhau của văn bản

Luận án tập trung vào việc thiết kế các khối cho hệ thống phát hiện sự giống nhau của văn bản. Trước tiên, các tài liệu sẵn có được thu thập lại, đồng thời quá trình tiền xử lý sẽ loại bỏ các dấu câu, ký tự đặc biệt và lưu trữ dưới dạng dữ liệu thô. Để thuận tiện cho quá trình xử lý chính, trong giai đoạn tiền xử lý, văn bản thu thập sẽ được phân đoạn và lấy mẫu sao cho các mẫu có độ dài bằng nhau. Sau đó, các phân đoạn này được lưu trữ như là dữ liệu thô nhằm mục đích trích xuất các đoạn văn bản giống nhau (nếu có) tại đầu ra kết quả đánh giá. Trong giai đoạn xử lý chính, các văn bản sẽ được số hóa và cho qua bộ lọc Haar để thu được dữ liệu cho bộ DNA nguồn. Trong khi đó, văn bản đánh giá được cho qua bộ mã hóa để xử lý. Bộ mã hóa này thực chất giống như quá trình tính toán DNA cho văn bản nguồn. Tuy nhiên, văn bản đánh giá thô được tạo thành sau quá trình tiền xử lý sẽ được phân đoạn. Sau đó, từng phân đoạn trong văn bản đánh giá được mã hóa thành một DNA nhằm mục đích phát hiện sự giống nhau (nếu có) của phân đoạn đó với một phân đoạn khác thuộc bộ dữ liệu nguồn. Vì vậy, văn bản đánh giá thô có thể xem như các chuỗi tín hiệu cần xử lý. Hình 3.6 mô tả chi tiết quá trình xử lý để đánh giá văn bản kiểm tra so với tập văn bản nguồn (kho dữ liệu).



Hình 3.6. Quá trình xử lý để đánh giá văn bản cần kiểm tra

3.4. Đề xuất quy trình chuyển đổi dữ liệu

Để so sánh mức độ giống nhau giữa hai chuỗi, trong luận án đề xuất phương pháp nhằm chuyển đổi nội dung văn bản thành dạng chuỗi thời gian thực và sử dụng bộ lọc Haar trong phép biến đổi Wavelet rời rạc để loại bỏ các xung tín hiệu bất thường, dữ liệu văn bản được chuyển đổi và biểu diễn thành những dãy số thực ($T = t_1, t_2..., t_n$). Để xác định mức độ giống nhau giữa hai chuỗi số $X = x_1, x_2..., x_n$ và $Y = y_1, y_2..., y_n$ cần tính độ tương tự Sim (X, Y) của hai chuỗi thời gian này.

Luận án đề xuất quy trình mã hóa văn bản thành chuỗi tín hiệu số như sau:

Thuật toán 3.1. Quy trình mã hóa văn bản thành chuỗi tín hiệu số DNA

	Đầu vào: Văn bản
	Đầu ra: Chuỗi số DNA
	Xử lý: Mã hóa văn bản thành chuỗi tín hiệu số
1	- Tiền xử lý (tách đoạn, tách câu, loại bỏ các dấu câu, ký tự đặc biệt, đánh chỉ mục và lưu trữ dưới dạng dữ liệu thô...).
2	- Số hóa nhằm chuyển dữ liệu thô thành dạng chuỗi số.
3	- Xử lý qua bộ lọc Haar để mã hóa thành các DNA.

Để thực hiện quy trình chuyển đổi dữ liệu sẽ có những vấn đề cần giải quyết như: Thực hiện tiền xử lý văn bản qua nhiều công đoạn; làm sao để việc số hóa phải đảm bảo được các đặc trưng của văn bản và thực hiện tính toán nhanh; xử lý qua bộ lọc Haar để tạo thành các DNA với các giá trị riêng biệt để thuận lợi cho việc sắp xếp và tìm kiếm. Để thực hiện các nội dung này, luận án đã đề xuất các phương pháp và giải thuật xử lý chi tiết ở phần tiếp theo.

3.5. Đề xuất phương pháp và giải thuật xử lý

Trong phần này sẽ phân tích chi tiết nhiệm vụ của từng khối trong quá trình xử lý. Cụ thể là, giai đoạn tiền xử lý sẽ loại bỏ các ký tự đặc biệt và chia văn bản thành các phân đoạn. Tiếp đến, khâu số hóa ở giai đoạn xử lý chính sẽ chuyển đổi các từ trong từng phân đoạn thành một số thực đặc trưng trước khi đưa đến bộ lọc Haar để lấy mẫu cho việc tính toán các DNA nhằm phục vụ cho việc so sánh mức độ giống nhau của văn bản kiểm tra và các văn bản trong kho dữ liệu.

3.5.1. Tiền xử lý dữ liệu

Các văn bản nguồn sau khi được thu thập sẽ được đưa đến bộ tiền xử lý dữ liệu. Tại đây, hệ thống sẽ xem xét một câu như là đơn vị nhỏ nhất có thể bị sao chép. Theo đó, các phân đoạn sẽ được xác định dựa trên các dấu kết thúc câu như dấu (. ! ?). Sau đó, các ký tự không phải chữ số và chữ cái alphabet như (# () [] …) sẽ bị loại bỏ khỏi các phân đoạn. Nói cách khác, một phân đoạn sẽ đại diện cho một câu trong văn bản sau khi được loại bỏ các ký tự đặc biệt. Các phân đoạn này sẽ được lưu trữ như văn bản thô cho mục đích truy xuất kết quả. Các thao tác loại bỏ các ký tự đặc biệt, tách đoạn được thực hiện dễ dàng bằng các hàm hỗ trợ trong thư viện String của ngôn ngữ C#.

3.5.2. Quy trình số hóa

Để xây dựng bộ DNA cho văn bản nguồn, trong luận án thực hiện việc số hóa các từ trong mỗi phân đoạn. Tuy nhiên, có thể nhận thấy rằng mỗi phân đoạn thu thập được từ các văn bản nguồn là một câu. Theo đó, số lượng từ trong mỗi phân đoạn sẽ khác nhau, điều này dẫn đến độ dài chuỗi tín hiệu số thực cho mỗi phân đoạn là không giống nhau. Trong khi đó, việc tính toán các DNA sử dụng bộ lọc Haar đòi hỏi độ dài các chuỗi tín hiệu phải bằng nhau. Vì vậy, trong luận án sử dụng một cửa sổ để lấy mẫu cho mỗi phân đoạn. Cửa sổ lấy mẫu này sẽ dịch trên mỗi phân đoạn để trích ra các mẫu theo một tỷ lệ xếp chồng lên mẫu trước đó được cho sẵn. Để giảm thiểu số lần số hóa các từ trong phân đoạn, việc số hóa sẽ được thực hiện một lần ngay từ đầu trên toàn phân đoạn để tạo ra một chuỗi số tín hiệu thực trước khi lấy mẫu dữ liệu. Phần sau là nội dung chi tiết của việc số hóa và chuẩn hóa dữ liệu, lấy mẫu cho từng phân đoạn, tổ chức dữ liệu cho bộ DNA nguồn.

3.5.2.1. Số hóa và chuẩn hóa dữ liệu

Ở bước này sẽ số hóa mỗi từ trong phân đoạn thành một số thực mang đặc trưng của từ đó và phân biệt với số thực đại diện cho từ khác. Để làm được điều này, trước hết sử dụng bảng mã Unicode để chuyển đổi các ký tự trong mỗi từ thành một số nguyên ở hệ thập phân, sau đó ghép chuỗi các số nguyên này theo đúng trật tự tương ứng của chúng trong một từ để tạo thành một số nguyên đại diện.

Bảng mã Unicode ánh xạ một số duy nhất đến một ký tự (ký tự này có thể là chữ cái tiếng Anh như "A", "B", "C"; hoặc tiếng Việt "ư", "ô", "ỗ"; hoặc các thứ tiếng khác trên thế giới; hoặc là các ký tự đặc biệt như "!", "@", "#"; hoặc các dấu chấm câu như ".", ",", ";"...). Mỗi một số duy nhất đó được gọi là một điểm mã (*Code point*), mỗi điểm mã được biểu diễn dưới dạng U+số_hexa. Tiếng Việt sử dụng các ký tự trong khối Basic Latin để biểu diễn các ký tự không dấu như "A", "B", "C"… chứa dãy mã từ U+0020 đến U+007F là khối chứa các mã của tất cả chữ cái và mã điều khiển của bảng mã ASCII. Cụ thể, mỗi điểm mã đại diện cho một ký tự hay ký hiệu và được sắp xếp theo thứ tự tương ứng như trong bảng mã ASCII; còn các ký tự có dấu thì nằm trong các khối Latin-1 Supplement, Latin Extended-B, Latin Extended Additional như các ký tự "ư", "ô", "ỗ"... Với tiếng Việt trong bảng mã Unicode, các ký tự nằm ở các khối chứa số nguyên đại diện tối đa là 4 chữ số. Thông tin về các khối và điểm mã theo bảng mã Unicode[7].

Ví dụ 3.1: Biểu diễn ký tự "**H**" theo bảng mã Unicode là U+0048, tương đương số thập phân là 72. Hoặc biểu diễn từ "**Chuỗi**" sẽ là U+0043 U+0068 U+1ED7 U+0075 U+0069, tương đương với dãy số thập phân là 0067 0104 0117 7895 0105.

Tuy nhiên, số lượng chữ số cho mỗi ký tự trong bảng mã là khác nhau, nên sẽ xảy ra trường hợp một chuỗi số đã ghép có thể được tạo thành bởi nhiều từ khác nhau. Do đó, thực hiện chuẩn hóa độ dài chuỗi số mã hóa cho mỗi ký tự bằng cách xác định một số lượng chữ số tối đa cho mỗi ký tự dựa vào bảng mã Unicode, số lượng chữ số tối đa này gọi là *m*. Ví dụ, nếu một ký tự có giá trị trong bảng mã Unicode gồm *m'* chữ số, với *m'*<*m*, giá trị đó sẽ được thêm *m''* số 0 vào trước sao cho *m'*+*m''* = *m*. Điểm nổi bật của cách chuẩn hóa này là vị trí của giá trị số hóa cho mỗi ký tự sau khi ghép chuỗi sẽ được duy trì tương ứng với vị trí của nó trong một từ. Giả sử một từ thứ *i* nào đó trong phân đoạn có L_i ký tự, trong đó ký tự thứ *l* (*l* = 1, 2, ..., L_i) có giá

[7] https://unicode-table.com/en/ (truy cập tháng 10/2019)

trị theo bảng mã Unicode là $s_{i,l}$. Số nguyên đại diện của từ đó trong một phân đoạn (ký hiệu là s_i) sẽ được tính theo công thức sau:

$$s_i = \sum_{l=1}^{L_i} s_{i,l} \times 10^{m \times \left(L_i - l\right)} \tag{3.9}$$

Tuy nhiên, mỗi từ trong phân đoạn có độ dài khác nhau nên các số nguyên đại diện cho một từ sẽ có độ lớn rất khác nhau. Để việc so sánh được chính xác, chúng tôi chuẩn hóa giá trị cho một từ theo hàm logaric cơ số 10. Giả sử độ dài lớn nhất của chuỗi số mà một từ có khả năng tạo thành là $M = m*L_{max}$, với L_{max} là độ dài từ tối đa có thể. Cuối cùng, giá trị số thực đại diện cho một từ được sử dụng như tín hiệu được đưa đến bộ lọc Haar được xác định bởi công thức sau:

$$x_i^w = M - m \times L_i + \log_{10}\left(s_i\right) \tag{3.10}$$

Ví dụ 3.2: Quá trình số hóa và chuẩn hóa dữ liệu với từ "**Chuỗi**" được minh họa như sau:

Với từ "Chuỗi", ta có các giá trị L_i=5, m=4, với độ dài tối đa của 1 từ L_{max}=10, M=$m*L_{max}$=40

C	h	u	ỗ	i

Chuẩn hóa giá trị cho mỗi ký tự theo mã Unicode

0067	0104	0117	7895	0105

Số nguyên đại diện của từ: $s_i = \sum_{l=1}^{L_i} s_{i,l} \times 10^{m \times \left(L_i - l\right)}$

0067	0104	0117	7895	0105	0000	0000	0000	0000	0000

Đặc điểm của số nguyên đại diện cho từ là giữ đúng vị trí các ký tự và chèn thêm các số 0 để đảm bảo độ dài các chuỗi số biểu diễn cho các từ bằng nhau. Việc chuẩn hóa này nhằm mục đích làm cho giá trị chuỗi số biểu diễn cho các từ không chênh lệch nhiều nếu chúng là các từ gần giống nhau nhưng có số lượng ký tự khác nhau, ví dụ như hai từ "hàn" và "hàng". Giá trị chuỗi số sau đó được chuẩn hóa theo hàm logaric cơ số 10 thành một số thực nhỏ để giảm thiểu không gian lưu trữ và độ phức tạp tính toán trước khi lấy mẫu đưa vào bộ lọc Haar.

Giá trị số thực đại diện cho từ "Chuỗi" được xác định bởi công thức 3.10 là x = 40 - 5 * 4 + $\log_{10}$(670104011778950105) = 20 + 17.826142218 = 37.826142218

3.5.2.2. Lấy mẫu cho từng phân đoạn

Giả sử một phân đoạn có *W* từ và từ thứ *i* (*i* = 1, 2, ..., *W)* được đặc trưng bởi giá trị x_i^w như công thức 3.10. Để thực hiện việc tạo DNA cho phân đoạn, luận án thực hiện việc lấy mẫu cho phân đoạn đó. Cụ thể là, một cửa sổ lấy mẫu độ dài là $N = 2^K$ tín hiệu được sử dụng để trích ra *N* giá trị số thực liên tiếp trong *W* giá trị của phân đoạn bằng cách dịch từ trái qua phải. Do đó, một phân đoạn có thể tạo ra nhiều mẫu và các mẫu đều có độ dài bằng nhau là *N*. Độ dịch của cửa sổ có thể là một hoặc nhiều giá trị. Khoảng dịch càng xa thì độ phức tạp tính toán sẽ được giảm nhưng đồng thời độ chính xác cũng giảm theo. Tóm lại, đầu vào của bộ lọc Haar chính là một mẫu trong phân đoạn được cho bởi vector sau:

$$\mathbf{x} = \mathbf{x}^{(0)} = \left[x_1\ x_2 \ldots x_N \right] = \left[x_i^w\ x_{i+1}^w \ldots x_{i+N-1}^w \right] \quad (3.11)$$

Các mẫu này sau đó sẽ được đưa đến bộ lọc Haar để tạo thành tập các DNA của một phân đoạn.

Ví dụ 3.3: Cho một dãy số thực, giả sử một cửa sổ lấy mẫu độ dài là *N* = 4, ví dụ mỗi lần dịch 2 vị trí, ta có được hình minh họa như sau:

25.34 15.20 50.15 46.35 75.80 21.06 68.24 72.26

25.34 15.20 50.15 46.35 75.80 21.06 68.24 72.26

25.34 15.20 50.15 46.35 75.80 21.06 68.24 72.26

3.5.3. Giải thuật cho bộ lọc Haar

Có thể nhận thấy rằng bộ lọc Haar dùng cho việc tính toán các DNA đóng vai trò rất quan trọng cho việc so sánh và phát hiện giống nhau. Việc xử lý qua bộ lọc Haar được sử dụng nhiều lần cho cả văn bản nguồn và văn bản cần đánh giá. Vì vậy, trong phần này luận án trình bày những lý thuyết liên quan và phát triển giải thuật tạo ra các chuỗi DNA cho bộ lọc Haar, sau đó các thuật toán chính sẽ dùng lặp lại giải thuật này như một module chuẩn.

Tín hiệu vào bộ lọc là các chuỗi số rời rạc gồm $N = 2^K$ số thực. Một phép biến đổi Haar rời rạc được thực hiện qua K bước lặp và tại lần lặp thứ k (hay mức thứ k), với $k = 1, 2, ..., K$; tín hiệu đầu ra của phép biến đổi được mô tả như sau:

$$\mathbf{x}^{(k)} = \left[\mathbf{x}_{\text{low}}^{(k)}\ \mathbf{x}_{\text{high}}^{(k)}\ \mathbf{x}_{\text{c}}^{(k-1)}\right] \tag{3.12}$$

trong đó, các hệ số xấp xỉ $\mathbf{x}_{\text{low}}^{(k)}$ và hệ số chi tiết $\mathbf{x}_{\text{high}}^{(k)}$ được cho bởi công thức lấy mẫu con như sau:

$$\mathbf{x}_{\text{low}}^{(k)} = \left(\mathbf{x}_{\text{a}}^{(k-1)} * \mathbf{f}_{\text{L}}\right) \downarrow 2, \tag{3.13}$$

$$\mathbf{x}_{\text{high}}^{(k)} = \left(\mathbf{x}_{\text{a}}^{(k-1)} * \mathbf{f}_{\text{H}}\right) \downarrow 2, \tag{3.14}$$

với $\mathbf{f}_{\text{L}} = [1\ 1]$ và $\mathbf{f}_{\text{H}} = [-1\ 1]$ lần lượt là đáp ứng bộ lọc thông thấp và thông cao; $\mathbf{x}_{\text{a}}^{(k-1)}$ và $\mathbf{x}_{\text{c}}^{(k-1)}$ tương ứng là các hệ số xấp xỉ của bước thứ *(k-1)* và tổng hợp các hệ số chi tiết của chuỗi tín hiệu thu được tại các bước trước đó.

Tại điểm khởi tạo, $\mathbf{x}_{\text{a}}^{(0)}$ và $\mathbf{x}_{\text{c}}^{(0)}$ được cho như sau:

$$\mathbf{x}_{\text{a}}^{(0)} = \mathbf{x}^{(0)}, \tag{3.15}$$

$$\mathbf{x}_{\text{c}}^{(0)} = [], \tag{3.16}$$

trong đó, $\mathbf{x}^{(0)}$ là chuỗi tín hiệu ban đầu (vector x sau khi được số hóa) và [] là vector rỗng. Các giá trị $\mathbf{x}_{\text{a}}^{(\text{k})} \in \mathbb{R}^{1 \times N_{\text{a}}(k)}$, với $N_{\text{a}}(k) = 2^{K-k}$, $\mathbf{x}_{\text{c}}^{(k)} \in \mathbb{R}^{1 \times N_{\text{c}}(k)}$ và $N_{\text{c}}(k) = \sum_{i=1}^{k} 2^{K-i}$, $k = 1,2...,K$ sẽ được cập nhật theo công thức sau:

$$\mathbf{x}_{\text{a}}^{(k)} = \mathbf{x}_{\text{low}}^{(k)}, \tag{3.17}$$

$$\mathbf{x}_{\text{c}}^{(k)} = \left[\mathbf{x}_{\text{high}}^{(k)}\ \mathbf{x}_{\text{c}}^{(k-1)}\right]. \tag{3.18}$$

Có thể dễ dàng chứng minh $N_{\text{a}}(k) + N_{\text{c}}(k) = 2^{K-k} + \sum_{i=1}^{k} 2^{K-i} = 2^K = N$, $k = 1,2...,K$. Qua đó, tín hiệu sau K bước lặp vẫn có độ dài N như ban đầu. Thực chất, các phân đoạn văn bản khác nhau sau khi được số hóa và qua bộ lọc Haar sẽ cho ra các

chuỗi số mang thông tin đặc trưng có thể phân biệt được mức độ khác nhau giữa chúng. Do đó, các chuỗi tín hiệu sau bộ lọc được gọi là các DNA.

Ví dụ 3.4: Tính toán các giá trị để tạo ra DNA, giả sử có chuỗi **x** có chiều dài là 8 số (N = 8) với 8 giá trị $\mathbf{x}_i$ từ 1 đến 8 được minh họa như sau:

x = 25.34 15.20 50.15 46.35 75.80 21.06 68.24 72.26

Ta sử dụng bộ lọc Haar để tách chuỗi **x** thành hai phần (xấp xỉ và chi tiết) bằng cách tính trung bình tổng theo cặp của các giá trị trong **x** trong khi vẫn duy trì sự khác biệt các cặp giữa các giá trị số thực trong chuỗi bằng cách tính trung bình hiệu. Thực hiện lấy mẫu (downsampling) lần thứ nhất, các giá trị xấp xỉ (tính theo trung bình tổng) được in đậm và các giá trị chi tiết (tính theo trung bình hiệu) được in nghiêng. Ta tính được dãy số cho lần lấy mẫu đầu tiên (L1) như sau:

L1: **20.27 48.25 48.43 71.25** *-5.07 -1.90 -27.37 2.01*

Giá trị đầu tiên được in đậm được tính bằng cách cộng hai số liên tiếp đầu tiên từ chuỗi thời gian ban đầu **x**, sau đó chia tổng cho 2, nghĩa là tính theo công thức ($\mathbf{x}_2$ + $\mathbf{x}_1$) / 2. Số thứ hai in đậm được tính bằng cách cộng hai số liên tiếp tiếp theo, sau đó chia tổng cho 2, tức là tính theo công thức ($\mathbf{x}_4$ + $\mathbf{x}_3$) / 2. Tương tự ta tính được 2 giá trị số đậm tiếp theo công thức tổng quát là ($\mathbf{x}_{n+1}$ + $\mathbf{x}_n$) / 2. Giá trị chi tiết (các số in nghiêng) được tính bằng cách trừ số thứ 2 cho số đầu tiên, sau đó chia cho 2 và các giá trị tiếp theo cũng được tính tương tự. Ta có công thức tổng quát để tính các giá trị in nghiêng là ($\mathbf{x}_{n+1}$ - $\mathbf{x}_n$) / 2.

Với $N = 8 = 2^3$ ta sẽ thực hiện 3 lần lấy mẫu để có được giá trị cuối cùng với 2 thành phần xấp xỉ và chi tiết. Với ví dụ này ta có các giá trị qua các lần lấy mẫu như sau:

x = 25.34 15.20 50.15 46.35 75.80 21.06 68.24 72.26

L1: **20.27 48.25 48.43 71.25** *-5.07 -1.90 -27.37 2.01*

L2: **34.26 59.84** *13.99 11.41*

L3: **47.05** *12.79*

Giá trị xấp xỉ cuối cùng (**47.05**) và các giá trị chi tiết qua các lần lấy mẫu (*12.79, 13.99, 11.41, -5.07, -1.90, -27.37, 2.01*) tạo thành chuỗi DNA cho dãy số

thực **x**, được sử dụng để so sánh với các DNA khác. Giá trị xấp xỉ cuối cùng (**47.05**) được lấy làm giá trị khóa để sắp xếp và tìm kiếm trên các DNA.

Không mất tính tổng quát, với bất kỳ bảng mã chuẩn tham chiếu cho trước, phương pháp số hóa đề xuất sẽ luôn chuyển được một từ sang một số thực tương ứng duy nhất. Theo đó, cho trước một tập các từ và các số thực tương ứng của chúng, nếu sắp xếp các số thực này theo thứ tự tăng dần thì các từ tương ứng này cũng sẽ được sắp xếp theo thứ tự từ điển. Do đó, có thể khẳng định rằng phương pháp số hóa đề xuất bảo đảm trật tự cho các từ.

Luận án phát triển một thuật toán để xác định giá trị cho các chuỗi DNA theo phân tích ở trên như sau:

Thuật toán 3.2. Xác định giá trị cho các chuỗi DNA

	Đầu vào: Tập chuỗi số thực.
	Đầu ra: Chuỗi số DNA cần tính sau *K* lần lấy mẫu.
	Khởi tạo: Các vector xấp xỉ và hệ số được cho bởi công thức (3.15) và (3.16)
1	**For** *k*:= 1→ *K*
2	- Tính chuỗi số thứ *k* theo công thức (3.12), (3.13) và (3.14)
3	- Cập nhật giá trị cho vector xấp xỉ và hệ số theo công thức (3.17) và (3.18)
4	**Endfor**

Phương pháp này là một đề xuất mới và rất quan trọng để chuyển dữ liệu số thành các DNA có giá trị đầu tiên làm khóa để sắp xếp theo thứ tự tăng dần, tổ chức dữ liệu thuận lợi cho tìm kiếm nhị phân được trình bày ở các phần sau.

3.5.4. Tổ chức dữ liệu cho bộ DNA nguồn

Sau khi thực hiện các bước ở quy trình số hóa, chúng ta sẽ có được một bộ DNA cho tập các văn bản thu thập được. Bước tiếp theo là sắp xếp bộ DNA theo giá trị đầu tiên của DNA tăng dần. Mục đích của việc sắp xếp là để hệ thống có thể thực hiện việc tìm kiếm nhị phân gần đúng để xác định DNA giống với DNA của một mẫu thuộc phân đoạn nào đó trong văn bản đánh giá. Qua đó có thể cải thiện được độ phức tạp của thuật toán đánh giá văn bản. Sở dĩ có thể dùng giá trị đầu tiên của DNA làm khóa sắp xếp vì đó chính là giá trị xấp xỉ hay tổng của các giá trị thành phần sau *K*

bước lặp. Vì vậy, tại vị trí này nếu giá trị của hai mẫu DNA (một mẫu thuộc văn bản nguồn và một mẫu văn bản đánh giá) giống nhau, hai mẫu văn bản tương ứng với hai DNA này sẽ giống nhau.

Cụ thể hơn, với phương pháp đề xuất dựa trên xử lý tín hiệu dùng DWT, bộ lọc Haar cho phép chuyển các vector mã hóa (vector số thực) đại diện cho đoạn văn bản thành các vector thành phần xấp xỉ (Approximation) và vector thành phần chi tiết (Detail). Trong đó, các vector thành phần này vẫn bảo toàn được trật tự các từ và tính đặc trưng của vector số thực đại diện cho văn bản. Theo lý thuyết về bộ lọc Haar, thành phần xấp xỉ có giá trị gần giống với vector mã hóa đại diện văn bản, nên nó được sử dụng làm khóa để sắp xếp các DNA (tức là DNA được tạo ra bằng cách ghép thành phần xấp xỉ và thành phần chi tiết lại), thuận tiện cho việc tìm kiếm nhị phân trên bộ dữ liệu DNA sau này. Giá trị khóa gần nhau sẽ tương ứng với các DNA gần giống nhau và suy ra các phân đoạn văn bản gần giống nhau. DNA thực ra cũng là một vector số thực vì nó được ghép từ vector thành phần xấp xỉ và vector thành phần chi tiết. Ngoài ra, vì tính chất đặc trưng của hai thành phần này sau phép biến đổi Haar nên một DNA sẽ đại diện cho trật tự từ và tính đặc trưng của một phân đoạn văn bản. Vì lý do đó, phương pháp đề xuất sử dụng khoảng cách Euclid để tính độ tương tự giữa hai DNA đại diện cho hai phân đoạn văn bản.

Thuật toán 3.3. Lưu trữ bộ DNA nguồn

	Đầu vào: Tập các văn bản nguồn thu thập được.
	Đầu ra: Kho dữ liệu đã được cập nhật và sắp xếp.
	Khởi tạo: Độ dài cho một DNA (*N*).
1	Tiền xử lý, phân đoạn và lưu trữ văn bản nguồn.
2	**For** mỗi phân đoạn, cần thực hiện:
3	- Số hóa phân đoạn.
4	- Lấy mẫu và xác định nhóm DNA của phân đoạn theo thuật toán 3.2.
5	**For** mỗi DNA trong nhóm, cần thực hiện:
6	- Sử dụng giá trị đầu tiên của chuỗi DNA (thành phần xấp xỉ) để tìm kiếm nhị phân trên kho dữ liệu nhằm xác định vị trí cần lưu trữ của các DNA sao cho các giá trị đầu tiên của các chuỗi DNA trong kho dữ liệu được sắp xếp tăng dần.

7	- Chèn các DNA vào kho dữ liệu theo đúng vị trí đã tìm.
8	**Endfor** // *Kết thúc vòng lặp for dòng 5*
9	**Endfor** // *Kết thúc vòng lặp for dòng 2*

Bên cạnh đó, luận án đã tạo cơ sở dữ liệu để lưu trữ các DNA, đánh chỉ mục các văn bản, đoạn, câu,… giúp cho việc tìm kiếm, truy xuất, liên kết dữ liệu được nhanh chóng và chính xác.

3.6. Đề xuất thuật toán phát hiện sự giống nhau

3.6.1. Mã hóa dữ liệu và tính DNA của văn bản đánh giá

Sau khi tiền xử lý văn bản đánh giá, chúng ta có thể dễ dàng thực hiện quy trình mã hóa dữ liệu văn bản đánh giá như đã trình bày ở phần trên. Đối với tập văn bản nguồn, các DNA của chúng được lưu trữ như cơ sơ dữ liệu, giống như thư viện sẵn có để đối chiếu sự giống nhau. Trong khi đó, văn bản đánh giá sau khi được phân đoạn sẽ được đưa vào mã hóa tuần tự như tín hiệu chuỗi thời gian thực. Theo đó, mỗi phân đoạn sẽ được lấy mẫu thành một nhóm các DNA và nhóm các DNA này sẽ được chuyển đến bộ so sánh. Sau đó, nhóm các DNA của phân đoạn sau cũng theo một quy trình như vậy cho đến khi hết văn bản đánh giá.

3.6.2. So sánh và đưa ra quyết định

Ở khối cuối cùng của hệ thống sẽ so sánh từng nhóm DNA của các phân đoạn với các DNA của tập dữ liệu nguồn được lưu trữ sẵn. Đối với mỗi mẫu DNA trong nhóm DNA đưa vào khâu so sánh, chúng tôi sẽ tìm kiếm nhị phân trong kho dữ liệu để xác định DNA nguồn nào có giá trị đầu tiên giống với DNA đang xét nhất. Tiếp theo, khoảng cách Euclid giữa hai DNA được tính rất đơn giản theo công thức sau:

$$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2 \tag{3.19}$$

trong đó, $\mathbf{x} \in \mathbb{R}^{1\times N}$ và $\mathbf{y} \in \mathbb{R}^{1\times N}$ lần lượt là vector DNA nguồn và vector DNA đang xét. Các vector **x** và **y** được tạo thành nhờ vào kỹ thuật xử lý tín hiệu với phép biến đổi DWT, vì vậy bảo toàn được trật tự và tính đặc trưng của văn bản. Qua đó, khoảng cách Euclid giữa hai vector sẽ tỷ lệ nghịch với độ tương tự giữa hai văn bản. Khoảng cách Euclid này sẽ được so sánh với một mức ngưỡng ε. Nếu $d(\mathbf{x}, \mathbf{y}) < \varepsilon$, hai DNA

được xem là giống nhau và vị trí tương ứng với DNA đang xét được đánh dấu lại để hệ thống đưa ra quyết định sau khi tổng hợp tất cả các mẫu DNA của phân đoạn. Thuật toán phát hiện giống nhau giữa văn bản cần đánh giá so với các văn bản trong kho dữ liệu (DNA nguồn) được mô tả trong Thuật toán 3.4.

Thuật toán 3.4. Phát hiện sự giống nhau

	Đầu vào: Văn bản cần đánh giá.
	Đầu ra: Đưa ra các phân đoạn chứa các từ giống với các phân đoạn nguồn (nếu có).
	Khởi tạo: Độ dài chuỗi DNA (N) và mức ngưỡng (ε) so sánh độ giống nhau.
1	Tiền xử lý, phân đoạn và lưu trữ dữ liệu để xuất kết quả.
2	**For** mỗi phân đoạn, cần thực hiện:
3	- Số hóa phân đoạn.
4	- Lấy mẫu và xác định nhóm DNA của phân đoạn theo Thuật toán 3.2.
5	**For** mỗi DNA **y** nào đó trong nhóm, cần thực hiện:
6	- Tìm kiếm nhị phân trên kho DNA nguồn để tìm ra một DNA **x** sao cho giá trị đầu của chuỗi DNA **y** đang xét gần nhất với giá trị đầu của DNA **x**.
7	- Tính khoảng cách Euclid $d(\mathbf{x}, \mathbf{y})$ theo công thức (3.19).
8	**If** $d(\mathbf{x}, \mathbf{y}) < \varepsilon$ **then**
9	- Đánh dấu DNA **y** đang xét.
10	**Endif**
11	**Endfor** // *Kết thúc vòng lặp for dòng 5*
12	- Tổng hợp các DNA **y** đã đánh dấu (nếu có) để thu được chuỗi các từ giống nhau của phân đoạn đang xét so với phân đoạn nguồn.
13	**Endfor** // *Kết thúc vòng lặp for dòng 2*

Trong quá trình huấn luyện và thử nghiệm, chúng tôi chọn các mức ngưỡng khác nhau để đánh giá nhằm đưa ra các phân đoạn DNA giống nhau theo các tỷ lệ ở các mức độ để biết được văn bản giống nhau nhiều hay ít.

3.6.3. Độ phức tạp của thuật toán phát hiện sự giống nhau

Với ưu điểm của phép biến đổi DWT và bộ lọc Haar, phương pháp đề xuất khi so khớp các đoạn tín hiệu là chỉ thực hiện các phép dịch qua trái, qua phải trên các

bit hay thực hiện các phép cộng (+), trừ (-) cho mỗi lần lấy mẫu con (downsampling) nên hệ thống sẽ xử lý rất nhanh.

Với một chuỗi độ dài n thì cần $\log_2(n)$ lần lấy mẫu và sau mỗi lần lấy mẫu thì độ dài chuỗi chỉ còn một nữa (1/2) so với lần trước. Vì vậy, độ phức tạp tính toán đối với việc mã hóa một chuỗi thành một DNA không quá $O(\log_2(n))$. Đối với một phân đoạn có M từ, độ dịch chuyển cửa sổ để tính các DNA cho phân đoạn là s từ, vậy độ phức tạp để mã hóa một phân đoạn thành các DNA là $O((M-n+1)/s)\log_2(n))$. Khi so sánh hai văn bản với nhau, thuật toán so sánh từng DNA của văn bản cần so sánh với bộ DNA của văn bản nguồn. Việc tìm kiếm DNA trên văn bản nguồn bằng thuật toán tìm kiếm nhị phân gần đúng trên giá trị khóa của chuỗi DNA (giá trị xấp xỉ) có độ phức tạp $O(M\log_2((M-n+1)/s))$. Độ phức tạp này nhỏ hơn nhiều so với thuật toán tìm chuỗi con chung và hầu hết các thuật toán so khớp chuỗi cũng như thuật toán dựa trên mô hình vector có độ phức tạp là O(n^2) hay O(mn) đã được đề cập ở phần trước.

3.7. Kết quả thử nghiệm phương pháp dựa trên DWT

3.7.1. Dữ liệu thử nghiệm

3.7.1.1. Bộ dữ liệu của PAN

Một cách tiếp cận chung của việc đánh giá hệ thống phát hiện sao chép là đánh giá dựa trên kho dữ liệu (Document collection). Hằng năm, PAN cung cấp các bộ dữ liệu để phục vụ cho việc nghiên cứu, thực nghiệm và đánh giá kết quả của các nhóm nghiên cứu. Dưới đây là bảng tổng hợp các kho dữ liệu tại các cuộc thi PAN.

Bảng 3.1. Tổng hợp so sánh các kho dữ liệu tại các cuộc thi PAN

	PAN-1 2009	PAN-2 2010	PAN-3 2011	PAN-4 2012	PAN-5, 6, 7 2013-2015	PAN-8, 9 2016-2017
Số lượng tập tin	41.233	27.073	26.939	NA	NA	NA
Số trường hợp đạo văn	94.202	68.558	61.065	3.033	~6.000	NA
Kho dữ liệu thử nghiệm	Cục bộ	Cục bộ	Cục bộ	Clue Web09	Clue Web09 DUC01	Twitter

Ghi chú: NA (No Available information) - Thông tin không xác định

Ngoài ra, còn có các kho dữ liệu phục vụ các nghiên cứu về NLP như TREC, DUC, Mediline,... Tuy nhiên, với bài toán đặt ra, luận án nghiên cứu về kho dữ liệu của PAN phục vụ cho nhiệm vụ phát hiện sao chép văn bản. Trong đó, tập trung nghiên cứu giải quyết bài toán phát hiện sao chép dựa trên kho dữ liệu cục bộ như các hội nghị PAN năm 2009 đến 2011.

Trong cuộc thi PAN năm 2009 giới thiệu một kho dữ liệu nhân tạo để đánh giá hệ thống sao chép văn bản gồm 41.223 tài liệu với 94.202 trường hợp sao chép [100, 123]. Kho dữ liệu này chuyên phục vụ để đánh giá các phương pháp phát hiện sự giống nhau của văn bản được rất nhiều nhóm nghiên cứu và các phòng thí nghiệm trên thế giới sử dụng.

Bộ dữ liệu này chủ yếu để đánh giá các hệ thống phát hiện sự giống nhau của văn bản bằng phương pháp so khớp chuỗi với các trường hợp nhân tạo, các đoạn giống nhau được chú thích trong tập tin .XML tương ứng. Phần lớn các hệ thống tham gia cuộc thi sử dụng kỹ thuật so khớp Brute-Force, n-gram để phát hiện sự giống nhau [123]. Kết quả đạt được của các hệ thống qua đánh giá của cuộc thi lần này cao nhất theo độ đo *prec* (precision) là ***74.73%*** và *rec* (recall) là ***69.67%*** [100]. Dưới đây là cấu trúc của tập tin .XML tương ứng của tài liệu đáng ngờ cần kiểm tra. Thông qua tập tin này, chúng ta biết được những đoạn giống nhau trong tài liệu đáng ngờ so với các tài liệu nguồn.

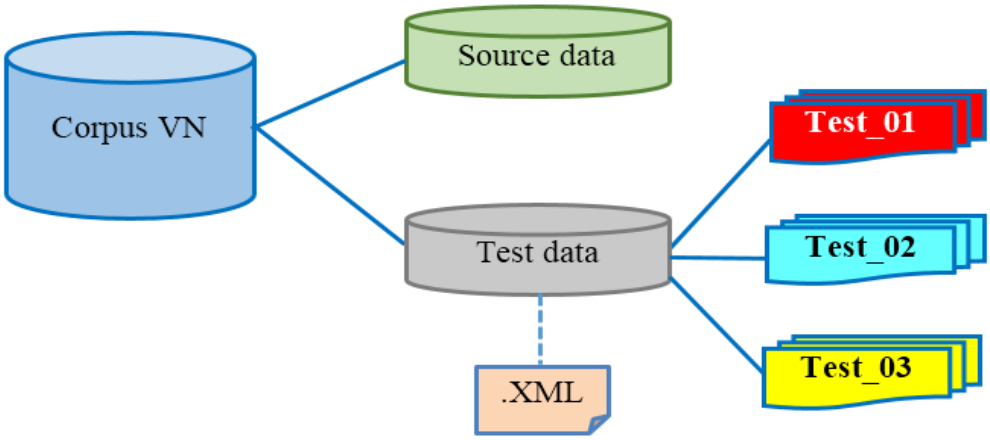
```
<document reference="suspicious-documentXYZ.txt">
   <feature name="detected-plagiarism"
               this_offset="..."     this_length="..."
               source_reference="source-documentABC.txt"
               source_offset="..."   source_length="..."
   />
   ...
</document>
```

Trong phần thực nghiệm, để đánh giá thuật toán đề xuất, luận án đã thực hiện thử nghiệm và đánh giá trên bộ dữ liệu chuẩn của PAN 2009 và đạt được hiệu quả rất cao. Bên cạnh đó, trong luận án cũng xây dựng bộ dữ liệu tiếng Việt tương tự như PAN để thử nghiệm thuật toán đề xuất.

3.7.1.2. Bộ dữ liệu văn bản tiếng Việt

Tiếng Việt mang nhiều đặc trưng ngôn ngữ riêng nên phức tạp hơn so với tiếng Anh và các ngôn ngữ khác như: Phân biệt các từ không xác định bằng dấu cách (như trong tiếng Anh phân biệt các từ bằng dấu cách), vì một từ có thể bao gồm nhiều âm tiết và chỉ mỗi âm tiết mới phân biệt bằng dấu cách. Điều này dẫn đến nhập nhằng trong phân đoạn từ tiếng Việt; ngữ pháp tiếng Việt phức tạp cùng với hiện tượng đồng âm khác nghĩa, từ đồng nghĩa hay thay đổi trật tự câu và các phép tu từ… dẫn đến nhập nhằng trong xác định ý nghĩa của câu, từ.

Hiện nay, ở Việt Nam chưa có một kho dữ liệu chuẩn để phục vụ cho các nghiên cứu về xử lý văn bản tiếng Việt. Vì vậy, bên cạnh việc nghiên cứu xây dựng hệ thống kho dữ liệu ứng dụng thực tế phục vụ phát hiện sự giống nhau của văn bản tiếng Việt, chúng tôi cũng đã tạo bộ dữ liệu để đánh giá hiệu quả thuật toán so sánh văn bản đề xuất. Bộ dữ liệu chia thành 2 tập dữ liệu gồm: Dữ liệu nguồn (Source data) chứa các văn bản gốc và dữ liệu đánh giá (Test data - hay gọi là các tài liệu đang ngờ/ tài liệu cần kiểm tra). Dưới đây là mô hình tạo bộ dữ liệu thử nghiệm.



Hình 3.7. Mô hình tạo bộ dữ liệu thử nghiệm tiếng Việt

Cách tạo bộ dữ liệu thử nghiệm:

- Dữ liệu gốc gồm 10 file (gồm các luận văn, bài báo...), chứa trong thư mục Source data.
- Thực hiện Tiền xử lý: Loại bỏ các thành phần không cần thiết, loại bỏ các từ dừng, tách đoạn, tách câu, tách từ… tạo thành văn bản sạch.

- Thực hiện tạo ra bộ dữ liệu thử nghiệm cho 3 trường hợp sau:

+ **TH1 - Tập kiểm tra 1 (Test_01)**: Dạng sao chép nguyên văn (trên 90%): gồm 10 tài liệu, mỗi tài liệu chỉ thay đổi thứ tự các đoạn và câu trong tài liệu.

+ **TH2 - Tập kiểm tra 2 (Test_02)**: Dạng sao chép với sửa đổi ít, tức là mức độ sao chép nhiều (từ 70% đến 90%): gồm 10 tài liệu, mỗi tài liệu thực hiện: 1) thay đổi thứ tự một số đoạn và câu văn bản trong tài liệu; 2) xóa một số đoạn và câu trong các tài liệu; 3) thêm vào một số đoạn và câu văn bản ở tài liệu khác vào vị trí tương ứng.

+ **TH3 - Tập kiểm tra 3 (Test_03)**: Dạng sao chép sửa đổi nhiều, tức là mức độ sao chép ít (từ 50% đến 70%): gồm 10 tài liệu, mỗi tài liệu: 1) Xóa từ 30% đến 50% các đoạn và câu văn bản trong các tài liệu; 2) Thêm vào các đoạn và câu văn bản ở tài liệu khác tại vị trí tương ứng; 3) Đổi vị trí một số câu trong văn bản.

Với mục đích đánh dấu các đoạn văn bản sao chép trong bộ dữ liệu thử nghiệm để có thể kiểm tra độ chính xác của các thuật toán so khớp sau này, trong luận án đã trình bày nghiên cứu để tạo ra tập tin .XML tương ứng cho mỗi tài liệu kiểm tra. Định nghĩa kiểu tài liệu (DTD) của tập tin .XML như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
   <document reference="tl_kiemtra_xxxxx.doc">
       <feature this_offset="vt_1" this_length="dodai_1"
       source_reference="tl_nguon_yyyyy.doc"
       source_offset="vt_2" source_length="dodai_2"
       obfuscation="capdo"
   />
   ...
</document>
```

Giải thích các thông số:

- **tl_kiemtra_xxxxx.doc**: Tài liệu kiểm tra thứ xxxxx (số thứ tự)
- **tl_nguon_yyyyy.doc**: Tài liệu nguồn thứ yyyyy (số thứ tự)
- **vt_i**: Vị trí bắt đầu của đoạn văn bản
- **dodai_n**: Độ dài đoạn văn bản

- **capdo**: Đoạn văn bản giống nhau với 3 cấp độ thay đổi hay xáo trộn khác nhau (**low, medium, high**). Mỗi cấp độ sao chép, thực hiện tô màu văn bản tương ứng là: low – vàng, medium – xanh, high – đỏ.

Với dữ liệu chuẩn của PAN và bộ dữ liệu tiếng Việt này, trong luận án đã phát triển một công cụ với thuật toán đề xuất ở trên để thực nghiệm và kết quả rất chính xác và nhanh. Phần sau sẽ trình bày chi tiết về kết quả thực nghiệm để chứng minh phương pháp so sánh văn bản dựa trên DWT đem lại hiệu quả cao và có thể ứng dụng để giải quyết bài toán phát hiện sao chép văn bản.

3.7.2. Kết quả thử nghiệm

Qua các nội dung trình bày và các bước xử lý chính như trên, luận án đã chuyển đổi được văn bản thành dạng tín hiệu số là các chuỗi số thực DNA, đảm bảo được tính duy nhất và toàn vẹn của thông tin. Với phương pháp đề xuất này, trong luận án đã xây dựng các module xử lý, trong đó có mã hóa các văn bản nguồn thành bộ DNA nguồn và tổ chức lưu trữ xếp hàng theo các lớp thông qua các giá trị khóa nên tốc độ tính toán của hệ thống nhanh và độ chính xác rất cao.

Để kiểm tra kết quả của giải thuật đề xuất, luận án sử dụng các phép đo trong PAN [100] để tính các giá trị *prec* và *rec*. Một cách cụ thể là gọi tập các chuỗi ký tự bị sao chép và tập chuỗi ký tự được phát hiện lần lượt như sau:

$$\mathcal{S} = \{S\} \tag{3.20}$$

$$\mathcal{D} = \{D\} \tag{3.21}$$

trong đó, $\mathcal{S}$ và $\mathcal{D}$ lần lượt là các chuỗi văn bản nguồn bị sao chép và các chuỗi văn bản đánh giá được phát hiện là giống với các chuỗi trong văn bản nguồn; với việc tính $\mathcal{S}$ và $\mathcal{D}$ nhằm đưa ra tỷ lệ giống nhau của văn bản đánh giá so với các văn bản nguồn bị sao chép. Các giá trị *prec* và *rec* được xác định bởi các công thức theo [100], đó là:

$$prec = \frac{1}{|\mathcal{D}|} \sum_{D \in \mathcal{D}} \frac{\left| D \cap \left(\bigcup_{S \in \mathcal{S}} S \right) \right|}{|D|} \tag{3.22}$$

$$rec = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \frac{\left| S \cap \left(\bigcup_{D \in \mathcal{D}} D \right) \right|}{|S|} \quad (3.23)$$

trong đó, $|\mathcal{S}|$ và $|\mathcal{D}|$ lần lượt là số phần tử trong tập hợp $\mathcal{S}$ và $\mathcal{D}$, $|\mathcal{S}|$ và $|\mathcal{D}|$ lần lượt là độ dài của chuỗi $S \in \mathcal{S}$ và $D \in \mathcal{D}$.

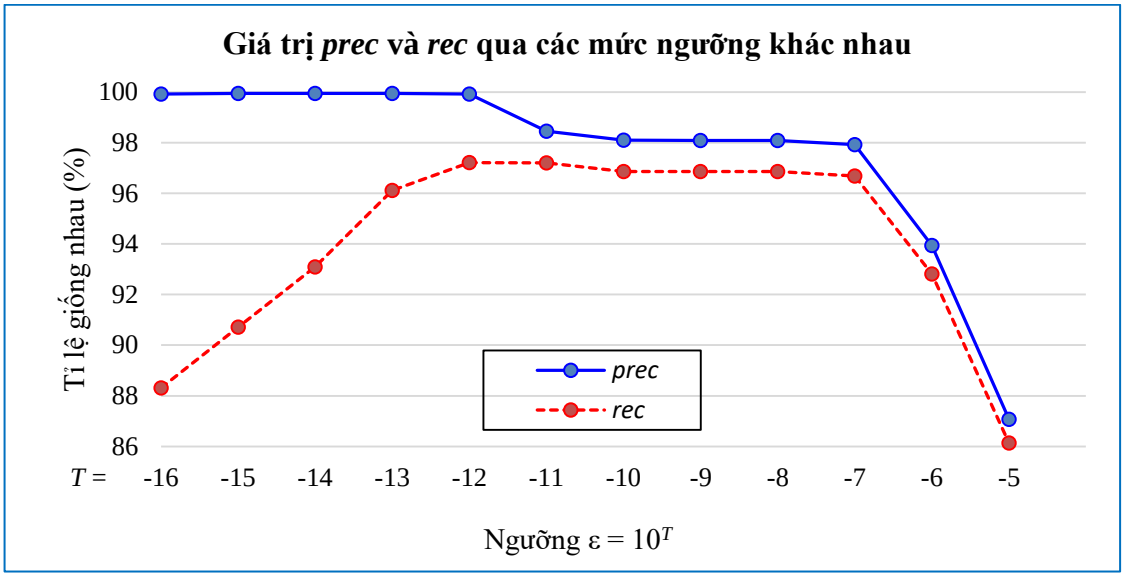
Luận án sử dụng công thức (3.19) để tính khoảng cách Euclid giữa hai phân đoạn **x** và **y**, và thiết lập giá trị ngưỡng ε trong quá trình huấn luyện để chỉ ra rằng với giá trị *d*(**x**, **y**) thu được thì **x** có giống **y** hay không? Do đó, *d*(**x**, **y**) tỷ lệ nghịch với mức độ giống nhau của hai văn bản, có nghĩa là *d*(**x**, **y**) càng nhỏ thì mức độ giống nhau càng lớn và ngược lại. Bên cạnh đó, giá trị ngưỡng ε không làm ảnh hưởng đến $|\mathcal{S}|$ và $|\mathcal{D}|$, thực tế là $|\mathcal{S}|$ và $|\mathcal{D}|$ phụ thuộc lớn vào bộ dữ liệu, trong khi ngưỡng ε được chọn ra trong quá trình huấn luyện sao cho giá trị *prec* và *rec* đạt giá trị tốt nhất.

Luận án thực hiện thử nghiệm trên bộ dữ liệu huấn luyện của PAN năm 2009[8], gồm 7.214 tài liệu nguồn và 7.214 tài liệu kiểm tra, với dung lượng tài liệu hơn 2.6 GB, thực hiện mỗi lần đánh giá 100 văn bản nghi ngờ hoàn toàn khác với văn bản sử dụng để tìm ngưỡng ε phù hợp thông qua kết quả đạt được của *prec* và *rec*. Trong luận án thiết lập các giá trị như Bảng 3.2 và kết quả đạt được như Hình 3.8.

Bảng 3.2. Các giá trị thiết lập cho quá trình thử nghiệm

Thông số	Giá trị thiết lập
Số lượng chữ số tối đa mã hóa một ký tự, *m*	5
Số ký tự tối đa trong một từ, L_{max}	40
Độ dài mẫu DNA, *N*	8
Số bước lặp để xử lý với bộ lọc Haar, *K*	3
Giá trị ngưỡng cho khoảng cách Euclid, ε	$10^{-16} \sim 10^{-5}$

[8] https://pan.webis.de/sepln09/pan09-web/plagiarism-detection.html (truy cập tháng 10/2019)



Hình 3.8. Giá trị *prec* và *rec* đạt được qua các mức ngưỡng khác nhau

Hình trên cho thấy sự thay đổi của độ đo *prec* và *rec* khi các mức ngưỡng thay đổi. Chúng ta nhận thấy, giá trị *prec* và *rec* đạt tỷ lệ khá thấp (87% và 86%) khi mức ngưỡng là 10^{-5}. Điều này cho thấy, khi chọn mức ngưỡng tăng lên thì nhiều câu nghi ngờ sẽ bị bỏ qua. Mặc dù, chỉ một số từ trong các câu này giống với một số từ ở các câu trong dữ liệu nguồn. Nhưng so với độ dài của câu, các từ giống nhau chiếm một tỷ lệ thấp. Do đó, khi mức ngưỡng tăng thì cả hai giá trị *prec* và *rec* tăng mạnh, trong khi sự giống nhau giữa chúng rất thấp, điều này dẫn tới giá trị *prec* và *rec* giảm xuống. Mặt khác, khi mức ngưỡng giảm xuống 10^{-12} các câu gần giống với các câu nguồn nên *prec* tiếp tục tăng trong khi *rec* giảm xuống đáng kể. Trên thực tế, tồn tại nhiều câu có độ tương tự cao, nhưng chúng không được phát hiện vì ngưỡng rất thấp đòi hỏi các câu đáng ngờ phải giống hoàn toàn hoặc gần giống với các câu nguồn.

Qua kết quả thực nghiệm như trên, khi chọn giá trị ngưỡng $\varepsilon = 10^{-11}$ cho kết quả *prec* và *rec* đạt tỷ lệ tốt và ổn định. Dưới đây là kết quả đạt được qua 10 lần thực nghiệm, mỗi lần đánh giá 100 văn bản nghi ngờ hoàn toàn khác nhau cho kết quả đạt được như Bảng 3.3, Hình 3.9 và Hình 3.10.

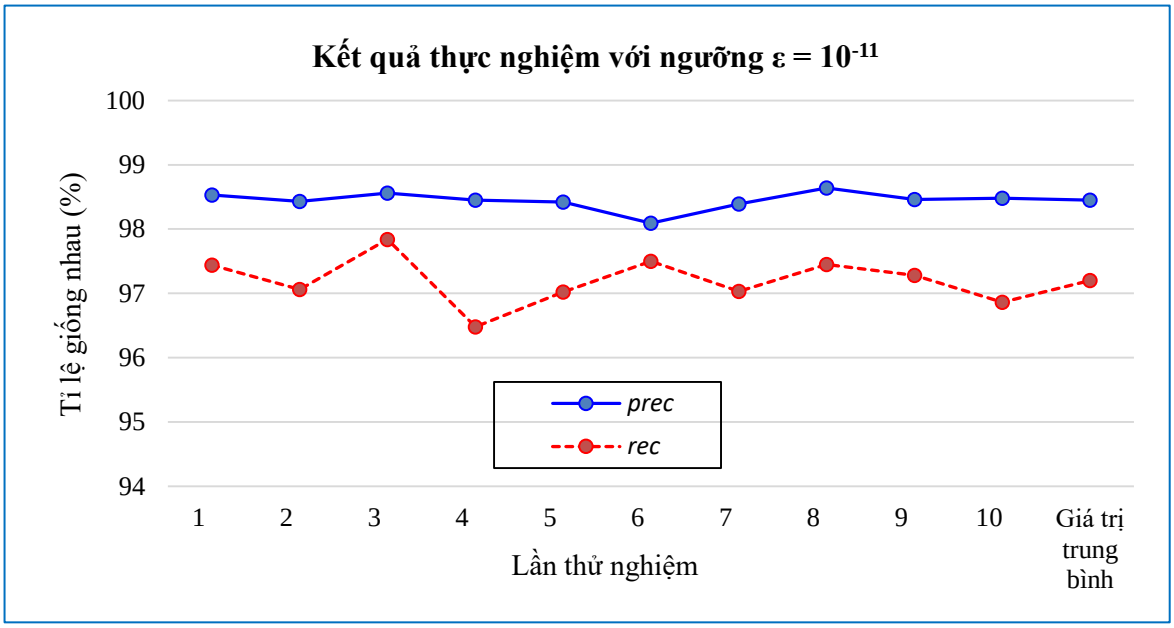
Bảng 3.3. Kết quả thực nghiệm

Lần thử nghiệm	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{D}\rvert$	*prec* (%)	*rec* (%)
1	6057	5990	98.53	97.44
2	5376	5301	98.43	97.06
3	9358	9290	98.56	97.84

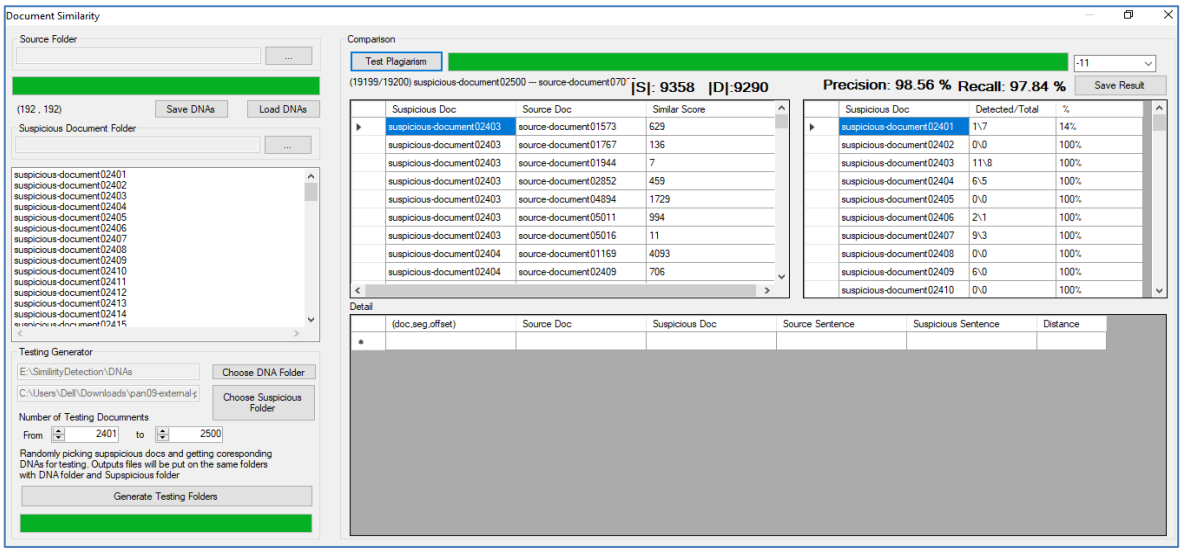
4	6567	6436	98.45	96.48
5	8132	8017	98.42	97.02
6	6989	6947	98.09	97.50
7	6530	6440	98.39	97.03
8	5647	5579	98.64	97.45
9	6570	6491	98.46	97.28
10	6091	5991	98.48	96.86
Giá trị trung bình	***6731.7***	***6648.2***	***98.45***	***97.20***

Với kết quả đạt được như trên, chúng ta thấy rằng với số lượng phần tử trong hai tập $\mathcal{S}$ và $\mathcal{D}$ khác nhau (từ khoảng 5300 đến 9300 phần tử), thuật toán đề xuất cho kết quả *prec* và *rec* rất cao và ổn định (trên 97%-98%). Các kết quả này cao hơn hẳn các kết quả của các nhóm tác giả khác được công bố tại PAN 2009, 2010, 2011.

Việc thực nghiệm trên bộ dữ liệu chuẩn của PAN được rất nhiều nhóm nghiên cứu và thực nghiệm trên thế giới sử dụng để đánh giá các phương pháp phát hiện sao chép cũng như sử dụng các độ đo dùng để đánh giá trong các cuộc thi của PAN cho thấy kết quả đạt được hoàn toàn tin cậy để đánh giá các thuật toán, hướng tiếp cận mới cũng như thuật toán đề xuất.



Hình 3.9. Kết quả thực nghiệm với ngưỡng ε = 10^{-11}



Hình 3.10. Giao diện kết quả một lần thực nghiệm

Do phương pháp đề xuất thực hiện mã hóa trên từng ký tự theo mã Unicode nên các ký tự khác nhau sẽ cho chuỗi số DNA riêng biệt, vì vậy phương pháp này có thể áp dụng cho các ngôn ngữ chữ viết Latin khác. Trong luận án đã thực nghiệm trên bộ dữ liệu tiếng Việt tự tạo đạt tỷ lệ chính xác rất cao và do nguồn dữ liệu tiếng Việt ít nên việc tìm kiếm rất nhanh và kết quả đạt được rất chính xác với nhiều mức ngưỡng ε khác nhau từ 10^{-7} đến 10^{-12}.

3.8. Đánh giá

Hiện nay, có nhiều kết quả nghiên cứu về phương pháp phát hiện sự giống nhau của văn bản, hầu hết các phương pháp truyền thống đều dựa trên cách biểu diễn văn bản theo mô hình vector [74, 134]. Trong phần nội dung Chương 2, luận án đã nghiên cứu và thực nghiệm một số phương pháp dựa trên mô hình vector để tính độ tương tự văn bản. Mặc dù đã đạt được những kết quả nhất định về độ chính xác và thực nghiệm trên tập dữ liệu tiếng Việt quy mô nhỏ. Tuy nhiên, với phương pháp biểu diễn văn bản theo mô hình vector đã tạo ra không gian lưu trữ rất lớn, khả năng mở rộng để triển khai vào thực tế gặp nhiều khó khăn. Việc đo độ tương tự dựa trên vector cho kết quả khả quan nhưng độ chính xác chưa cao và độ phức tạp của các thuật toán để xử lý dữ liệu theo vector lớn, tốc độ xử lý chậm khi thực hiện trên tập dữ liệu lớn.

Trong khi đó, phương pháp phát hiện sự giống nhau của văn bản dựa trên DWT và bộ lọc Haar được đề xuất trong luận án cho kết quả chính xác và nhanh hơn nhiều. Với phương pháp đề xuất này cho thấy việc nghiên cứu để áp dụng đặc tính

hữu dụng của xử lý tín hiệu đã đem lại hiệu quả cao về độ chính xác và độ phức tạp của thuật toán. Bên cạnh đó, với tính năng xử lý, tính toán dựa trên tín hiệu số thông qua các bit và khả năng ưu việt của tìm kiếm nhị phân trên các DNA được sắp xếp tăng dần nên thuật toán đề xuất giải quyết được các vấn đề về dữ liệu lớn hiệu quả hơn rất nhiều.

Với những nghiên cứu về phương pháp dựa trên mô hình vector ở Chương 2, luận án đã ứng dụng trong việc tổ chức dữ liệu DNA và kết hợp trong việc đánh chỉ mục cho tài liệu để lưu trữ thông tin, vị trí của văn bản phục vụ cho việc truy xuất tài liệu trong kho dữ liệu sau khi tìm kiếm và so sánh, đánh giá mức độ giống nhau của các chuỗi số DNA.

Nhằm giải quyết bài toán đánh giá mức độ giống nhau của văn bản trong phát hiện sao chép, trong nội dung Chương này, luận án đã khảo sát, nghiên cứu, phân tích và đề xuất những nội dung và đạt được những kết quả chính như sau:

1) Qua khảo sát và nghiên cứu, trong luận án đã đề xuất mô hình tổng quan của hệ thống phát hiện sự giống nhau của văn bản cho nghiên cứu của mình.

2) Luận án đã nghiên cứu, đề xuất một cách tiếp cận mới về xử lý văn bản, đã đề xuất được phương pháp, thuật toán để so khớp văn bản dựa trên phép biến đổi DWT và bộ lọc Haar đạt được hiệu suất và độ phức tạp rất tốt, cụ thể như sau:

- Chuyển từ văn bản sang tín hiệu số nên thuật toán so khớp được thực hiện dễ dàng hơn nhiều đối với khi so sánh chuỗi.
- DWT ứng dụng trong so sánh chuỗi là phương pháp hoàn toàn mới, nó cho phép các chuỗi ký tự có thể được thực hiện trình tự như xử lý tín hiệu thời gian thực.
- Dựa trên đặc tính của xử lý tín hiệu, việc liên kết các chuỗi cũng rất dễ dàng nên việc thống kê, đưa ra tỷ lệ giống nhau rất thuận tiện.
- Dùng DWT cho chuỗi ký tự sẽ cho ra một chuỗi DNA duy nhất. Do đó, tùy mức độ giống nhau giữa hai chuỗi, khoảng cách giữa chúng sẽ có giá trị lớn hay nhỏ. Qua đó, quá trình huấn luyện sẽ dễ dàng tùy chỉnh giá trị ngưỡng cho khoảng cách để thu được độ chính xác cao.
- Dễ dàng xây dựng cơ sở dữ liệu nguồn với rất ít bộ nhớ vì đã được mã hoá hoàn toàn sang định dạng số.

- Với đặc điểm của chuỗi DNA là phần tử đầu tiên có mã số riêng biệt nên việc lưu trữ các DNA nguồn đã được sắp xếp theo thứ tự tăng dần nên việc áp dụng phương pháp tìm kiếm nhị phân sẽ rất thuận lợi, nhanh và xử lý dữ liệu lớn rất tốt.

- DWT chỉ thực hiện các phép toán cộng (+), trừ (-) cho mỗi lần lấy mẫu con (downsampling) nên hệ thống sẽ xử lý rất nhanh.

- Độ phức tạp tính toán không quá O($n\log_2(n)$), nhỏ hơn nhiều so với thuật toán tìm chuỗi con chung có độ phức tạp O(n^2).

Với ứng dụng phát hiện sao chép, việc so sánh, đánh giá hiệu quả của phương pháp đề xuất so với các phương pháp khác qua thực nghiệm là một công việc rất phức tạp. Thông thường so sánh dựa vào tính năng, hiệu suất và thời gian thực thi. So sánh dựa trên tính năng mang tính chất định tính, thường mô tả các thuộc tính của ứng dụng, môi trường triển khai ứng dụng, thuật toán để so sánh,… So sánh dựa trên hiệu suất mang tính định lượng, thông thường dựa trên kết quả chạy thử nghiệm. Thời gian thực thi cũng là một vấn đề quan trọng đối với ứng dụng phát hiện sao chép, thời gian chạy phụ thuộc vào độ lớn của tập tin kiểm tra và kho dữ liệu, số lượng càng lớn chắc chắn thời gian kiểm tra càng lâu. Trong luận án đã thực nghiệm phương pháp đề xuất dựa trên bộ dữ liệu huấn luyện của PAN và các bộ dữ liệu tiếng Việt tự tạo để đánh giá kết quả đạt được và so sánh với các phương pháp đem lại kết quả cao tại PAN. Bên cạnh đó, luận án có những nhận xét đánh giá hiệu quả của phương pháp đề xuất ở Chương 3 dựa trên DWT với phương pháp dựa trên vector. Luận án cũng đã trình bày những ưu điểm của phương pháp đề xuất có khả năng xử lý dữ liệu lớn được hiệu quả và độ phức tạp của thuật toán được đánh giá tốt hơn các thuật toán xử lý chuỗi.

Một số điểm ưu việt của phương pháp đề xuất dựa trên DWT và định hướng phát triển của luận án:

- Trong luận án này, tác giả tập trung vào việc đề xuất và phát triển một phương pháp tổng quát dùng DWT. Trong đó, khoảng cách Euclid được sử dụng như thước đo độ tương tự nhằm đơn giản hóa việc mô tả phương pháp đề xuất.

- Bằng cách hiệu chỉnh đơn giản độ dài DNA, phương pháp đề xuất hoàn toàn có thể sử dụng kết hợp với độ đo Cosine (có hoặc không tính trật tự từ) thông qua công thức dùng vector chuẩn (vector norm).

- Khi xem xét sự sao chép văn bản, phương pháp đề xuất cho tỷ lệ phát hiện rất tốt với bộ dữ liệu chuẩn của PAN.
- Khi tính trật tự từ, phương pháp đề xuất có thể tận dụng trật tự sẵn có trong bảng mã Unicode để dẫn ra độ tương tự mà không cần truy xuất vị trí của từ trong tập văn bản nguồn.
- Có thể thấy rằng Unicode là một bảng mã đã được sắp xếp sẵn bao gồm cả tiếng Việt có dấu được chuẩn hóa quốc tế. Do đó, độ lớn của số thực đặc trưng cho một từ (theo cách số hóa đã đề xuất trong luận án) cũng đại diện cho vị trí của nó trong tập dữ liệu. Nhờ đó, việc tính độ tương tự cho văn bản tiếng Việt bao gồm cả dấu cũng được thực hiện một cách dễ dàng.
- Thông qua thuật toán số hóa dẫn ra các vector DNA trong trường số thực, các phương pháp lưu trữ và các độ đo trong NLP hiện đại sẽ dễ dàng được áp dụng cho xử lý tiếng Việt.

3.9. Kết luận Chương 3

Trong nội dung của Chương 3 đã đề xuất mô hình hệ thống, quy trình xử lý và trình bày một cách tiếp cận hoàn toàn mới trong phát hiện sự giống nhau của văn bản, đó là dựa trên phương pháp DWT và bộ lọc Haar. Đóng góp lớn trong Chương này cũng như của luận án là đề xuất mô hình hệ thống phát hiện sao chép, đề xuất thuật toán để chuyển văn bản thành các chuỗi số thực DNA, đề xuất phương pháp và giải thuật xử lý, đề xuất thuật toán phát hiện sự giống nhau. Các kết quả thực nghiệm trên bộ dữ liệu chuẩn của PAN và bộ dữ liệu tiếng Việt thử nghiệm đã chứng minh thuật toán đề xuất đem lại hiệu quả cao trong phát hiện sự giống nhau của văn bản. Kết quả

nghiên cứu trong Chương 3 có các công trình công bố trên tạp chí trong nước và quốc tế, các hội nghị khoa học quốc gia và quốc tế[9].

Từ hướng tiếp cận và phương pháp đề xuất này, luận án cũng đã tính đến xử lý dữ liệu lớn với việc mã hoá dữ liệu văn bản sang dạng tín hiệu số, sắp xếp dữ liệu theo thứ tự tăng dần cho phép tìm kiếm nhị phân, vì đây là một trong những phương pháp tìm kiếm nhanh nhất khi làm việc với dữ liệu lớn. Hơn nữa, DWT cho độ phức tạp tính toán chỉ là hàm tuyến tính trong mỗi lần lấy mẫu con nên giải pháp đề xuất sẽ càng hiệu quả trong quá trình xử lý dữ liệu lớn.

Kết quả đạt được đã chứng minh sự cải tiến của phương pháp DWT so với các phương pháp biểu diễn văn bản dựa trên mô hình vector để đem lại hiệu quả cao trong ứng dụng phát hiện văn bản giống nhau. Bên cạnh đó, luận án cũng đã kế thừa, áp dụng những nghiên cứu về biểu diễn văn bản theo mô hình vector để biểu diễn, đánh chỉ mục và sắp xếp các chuỗi DNA phục vụ cho việc tính toán độ tương tự và đánh dấu các đoạn giống nhau.

---

[9] (1) Hồ Phan Hiếu, Nguyễn Thị Ngọc Anh, Nguyễn Văn Hiếu, Đặng Thiên Bình, Võ Trung Hùng, *Một cách tiếp cận mới để phát hiện sự giống nhau của văn bản dựa trên phép biến đổi wavelet rời rạc*. Kỷ yếu Hội nghị Khoa học Công nghệ Quốc gia lần thứ X (Fair'10), lĩnh vực Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin, 2017, (479-487). *DOI: 10.15625/vap.2017.00057*

(2) Nguyen Thi Ngoc Anh, Ho Phan Hieu, Tran Anh Kiet, and Vo Trung Hung, *Similarity Detection for Higher-Order Structure of DNA Sequences*. Journal of Science and Technology: Issue on Information and Communications Technology, Vol. 3, No.2, 2017, (28-34). *DOI: 10.31130/jst.2017.51*

(3) Phan Hieu Ho, Ngoc Anh Thi Nguyen, Trung Hung Vo, *DNA Sequences Representation Derived from Discrete Wavelet Transformation for Text Similarity Recognition*. In Springer SCI Book, Modern Approaches for Intelligent Information and Database Systems, 2018, (75-85). *DOI: 10.1007/978-3-319-76081-0_7*

(4) Hồ Phan Hiếu, Nguyễn Thị Ngọc Anh, Võ Trung Hùng, *Phương pháp mã hóa văn bản thành chuỗi số DNA để đánh giá mức độ giống nhau của văn bản*. Hội thảo Khoa học Quốc gia Công nghệ thông tin và ứng dụng trong các lĩnh vực - lần thứ 7 (CITA2018).

(5) Phan Hieu Ho, Trung Hung Vo, Ngoc Anh Thi Nguyen, Ha Huy Cuong Nguyen, *A Narrative Method for Evaluating Documents Similarity based on Unique Strings*. International Journal of Recent Technology and Engineering (IJRTE), Vol 8(2S11), 2019, (473-479). DOI: 10.35940/ijrte.B1073.0982S1119

Trong luận án đã phát triển một công cụ dựa trên phương pháp DWT với các thuật toán chuyển văn bản thành các chuỗi số thực DNA, các giải thuật xử lý và thuật toán phát hiện sự giống nhau để đánh giá các kết quả thực nghiệm trên bộ dữ liệu chuẩn của PAN và bộ dữ liệu tiếng Việt thử nghiệm và kết quả đã chứng minh thuật toán đề xuất đem lại hiệu quả cao trong phát hiện sự giống nhau của văn bản.

Hướng phát triển trong thời gian đến, chúng tôi sẽ tiếp tục nghiên cứu để tối ưu hơn thuật toán đề xuất và thử nghiệm trên nhiều bộ dữ liệu khác. Đặc biệt là việc áp dụng phương pháp đề xuất vào hệ thống phát hiện sự giống nhau của văn bản tiếng Việt và triển khai ứng dụng vào thực tế tại ĐHĐN để góp phần hạn chế tình trạng sao chép và để nâng cao chất lượng nghiên cứu và đào tạo.

Chương 4. PHÁT TRIỂN HỆ THỐNG PHÁT HIỆN SAO CHÉP VĂN BẢN TIẾNG VIỆT

Để giải quyết bài toán phát hiện sao chép, trước tiên phải có một tập hợp các văn bản nguồn hay kho dữ liệu đủ lớn, có độ bao phủ cao. Trong Chương này trình bày giải pháp xây dựng kho dữ liệu văn bản tiếng Việt, ứng dụng các kết quả nghiên cứu đạt được để phát triển hệ thống phát hiện sao chép văn bản. Hệ thống được triển khai thử nghiệm vào thực tế tại ĐHĐN. Cuối cùng là đề xuất hướng phát triển để xử lý dữ liệu lớn với giải pháp triển khai dựa trên mô hình MapReduce và phương pháp biểu diễn DNA bằng Tensor.

4.1. Mô tả hệ thống

4.1.1. Mục đích

Mục đích của hệ thống là phát hiện những nội dung sao chép (giống nhau) trên một tài liệu so với những tài liệu đã có trong kho dữ liệu. Kho dữ liệu hiện tại giới hạn trong những luận văn, đồ án tốt nghiệp, bài báo và báo cáo khoa học tại ĐHĐN.

Ứng dụng các kết quả nghiên đạt được ở Chương 2 và Chương 3 để xây dựng hệ thống phát hiện sao chép văn bản có các thành phần chính như sau:

- Kho dữ liệu chứa các luận văn, đồ án tốt nghiệp, bài báo và báo cáo khoa học tại ĐHĐN.
- Ứng dụng tiền xử lý các tài liệu để phục vụ phát hiện sao chép.
- Ứng dụng phát hiện và đánh dấu những nội dung được sao chép từ các tài liệu đã có trong kho dữ liệu.

4.1.2. Các đối tượng sử dụng

Đối với hệ thống đề xuất, có các loại đối tượng sử dụng như sau:

- *Người quản trị:* Là chuyên viên quản trị hệ thống, có nhiệm vụ quản trị toàn bộ hệ thống, phân quyền người sử dụng, cập nhật thông tin, phát hiện các vấn đề phát sinh hay lỗi để khắc phục.

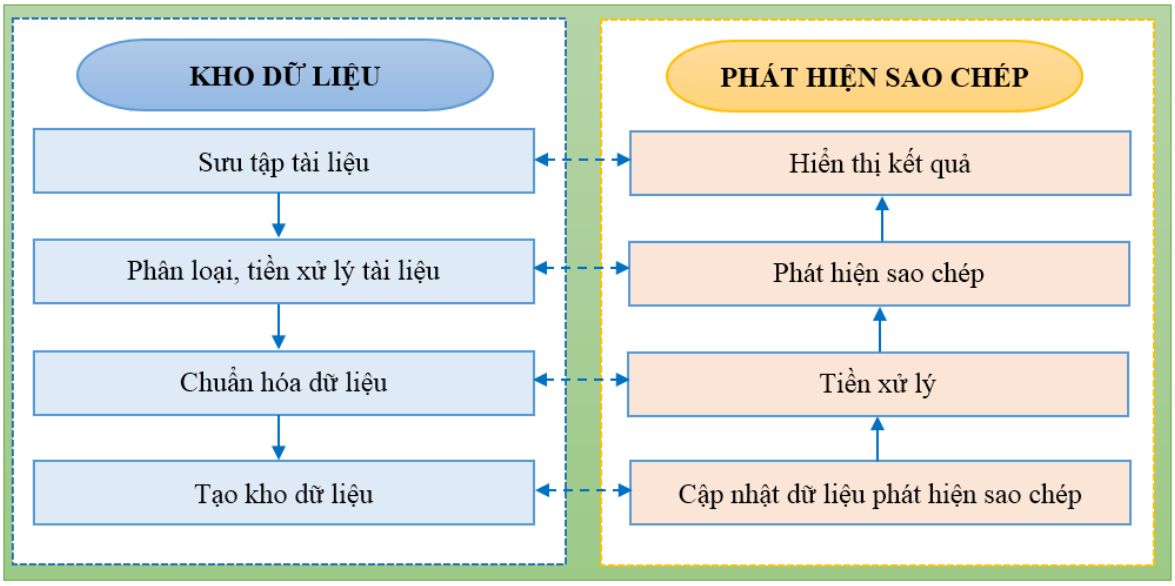
- *Người vận hành:* Là những người có quyền truy cập hệ thống để xem, sửa, xóa thông tin. Tùy theo nhiệm vụ, người quản trị có thể phân quyền để người vận hành có quyền truy cập và cập nhật những thông tin khác nhau trên hệ thống.

- *Người sử dụng đã đăng ký:* Là những người sử dụng được cấp phép để truy cập vào hệ thống nhằm xem thông tin, kiểm tra mức độ sao chép của tài liệu...

- *Người sử dụng chưa đăng ký:* Là những người sử dụng không cần đăng ký tài khoản truy cập vào hệ thống. Các đối tượng này chỉ được xem thông tin.

4.1.3. Mô hình tổng quát

Nhằm mục đích xây dựng kho dữ liệu và phát hiện sao chép văn bản, luận án đề xuất xây dựng một hệ thống với quy trình thực hiện như sau:



Hình 4.1. Quy trình phát hiện sao chép

- *Sưu tập tài liệu:* Nhân viên vận hành có trách nhiệm triển khai các hoạt động nhằm sưu tập các tài liệu để bổ sung vào kho dữ liệu của hệ thống. Việc sưu tập tài liệu có thể thực hiện tự động (xây dựng mô-đun crawler để tự động tìm kiếm các tài liệu trên Internet), bán tự động (xây dựng mô-đun để sinh viên, học viên cao học, nghiên cứu sinh hoặc các nhà khoa học cập nhật tài liệu theo quy định) hoặc thủ công (nhân viên vận hành tự tìm kiếm, sưu tập tài liệu từ các nguồn khác nhau và tự cập nhật vào hệ thống).

- *Phân loại, tiền xử lý:* Sẽ thực hiện việc phân loại tài liệu vào các lĩnh vực, thể loại và tiền xử lý dữ liệu. Hệ thống sẽ bóc tách, loại bỏ những thành phần không quan trọng (lời nói đầu, lời cảm ơn, lời cam đoan, hình ảnh...). Đồng thời, sẽ loại bỏ những đơn vị không cần thiết cho quá trình phát hiện sao chép như những cụm từ thường dùng, câu ngắn...

- *Chuẩn hóa dữ liệu:* Sau khi tiền xử lý, tài liệu sẽ được chuẩn hóa theo quy định trước khi cập nhật vào kho dữ liệu.

- *Tạo kho dữ liệu:* Tài liệu sau khi chuẩn hóa sẽ được đưa vào lưu trữ trong kho để phục vụ cho quá trình phát hiện sao chép sau này.

- *Cập nhật dữ liệu phát hiện sao chép:* Người sử dụng nhập thông tin, tài liệu cần kiểm tra để phát hiện sao chép.

- *Tiền xử lý:* Tiền xử lý tài liệu mà người sử dụng đưa vào theo tiêu chuẩn của hệ thống trước khi tiến hành phát hiện sao chép.

- *Phát hiện sao chép:* Tiến hành đối sánh tài liệu cần kiểm tra để phát hiện sao chép với các tài liệu đã có trong kho dữ liệu.

- *Hiển trị kết quả:* Hiển thị kết quả sau khi phát hiện sao chép. Việc sao chép được đánh giá ở 3 cấp độ: Toàn bộ tài liệu (dựa trên so sánh độ tương tự giữa nội dung của tài liệu đánh giá với các tài liệu đã có), mức độ đoạn (so sánh độ tương tự giữa các đoạn) và mức độ câu (so sánh các câu). Tất cả các mức độ đều có chỉ rõ mức độ, nguồn gốc sao chép và đánh dấu, tô màu các đoạn văn bản sao chép.

4.2. Xây dựng kho dữ liệu văn bản tiếng Việt

4.2.1. Giới thiệu

Hiện nay, kho dữ liệu trên thế giới rất đa dạng và lớn. Với bài toán phát hiện sao chép, làm thế nào để có kho dữ liệu tối ưu, dữ liệu trong kho phải được xử lý để giảm thời gian và độ phức tạp khi so sánh văn bản, xác định và lọc bỏ những dữ liệu thừa, không quan trọng giúp cho hệ thống phát hiện sao chép xác định độ tương tự giữa tài liệu cần đánh giá và tài liệu nguồn được hiệu quả hơn... là những vấn đề cần được nghiên cứu và đề xuất.

Với tình hình thực tế tại ĐHĐN, luận án xác định dữ liệu của kho gồm các đồ án tốt nghiệp của sinh viên, luận văn thạc sĩ của học viên cao học và các báo cáo, bài báo khoa học của cán bộ, giảng viên… được tổng hợp từ nhiều nguồn khác nhau.

Tài liệu trước khi đưa vào kho cần được mô tả, xác định các thông tin liên quan như: tên đề tài, tác giả, chuyên ngành, loại tài liệu, thời gian... Kho tài liệu có khả năng lưu trữ, hiển thị danh sách tài liệu có trong kho và được xử lý phục vụ cho việc phát hiện sao chép.

Dữ liệu của kho được khai thác bởi hai đối tượng:

- *Người sử dụng:* Có thể cập nhật, xem danh sách tài liệu có trong kho theo nhiều tiêu chí khác nhau thông qua giao diện web.
- *Hệ thống phát hiện sao chép:* Mục đích chính của việc xây dựng kho dữ liệu là phục vụ phát hiện sao chép văn bản, có thể sử dụng dữ liệu trong kho để phân tích so sánh nhằm đưa ra kết quả đánh giá mức độ giống nhau của một tài liệu cần kiểm tra so với dữ liệu trong kho.

Như vậy, kho dữ liệu phải có chức năng nạp (upload) tài liệu vào kho; xem danh sách tài liệu trong kho; phát triển hệ thống phát hiện sao chép có thể sử dụng dữ liệu của kho. Bên cạnh đó, luận án cũng tính toán để có thể mở rộng dữ liệu, lưu trữ nguồn dữ liệu lớn.

4.2.2. Kiến trúc hệ thống kho dữ liệu

4.2.2.1. Kiến trúc hệ thống

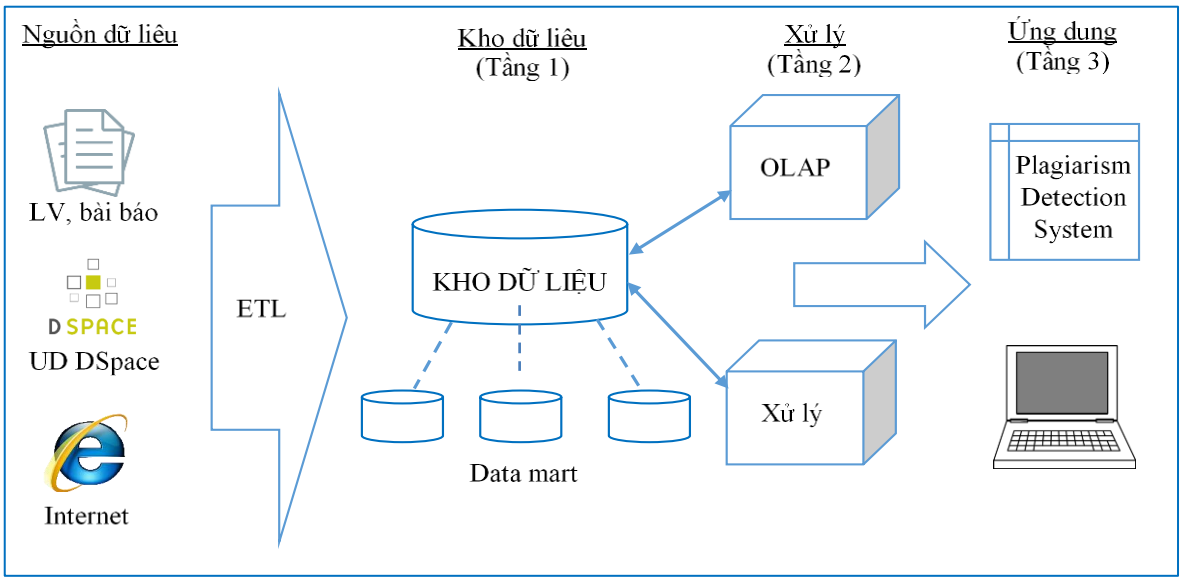
Kiến trúc hệ thống kho dữ liệu đề xuất được thực hiện theo mô hình kiến trúc 3 tầng như Hình 4.2, quy trình thực hiện qua các bước sau:

- **Bước 1:** (Extract - Transform - Load, **ETL**). 1) **Trích xuất** (Extract) dữ liệu từ hệ thống nguồn tài liệu khoa học của ĐHĐN (như đồ án tốt nghiệp của sinh viên, luận văn thạc sĩ, luận án tiến sĩ, bài báo khoa học…) với định dạng tập tin là .doc, .docx, .pdf, nguồn dữ liệu trên các DSpace của ĐHĐN và có thể trên Internet thu thập từ các thư viện, kho luận văn; 2) **Chuẩn hoá** (Transform) dữ liệu đầu vào phù hợp để lưu trữ trong kho như: chuyển đổi font chữ, chuẩn hóa về chữ thường, chuẩn hóa dấu, loại bỏ từ dừng, loại bỏ khoảng trắng, chuyển đổi cấu trúc, thể thức văn bản của tài

liệu về dạng chuẩn...; 3) Tách đoạn, tách câu, mã hóa, đánh chỉ mục, xử lý, **nạp dữ liệu** (Load) vào kho dữ liệu.

- **Bước 2:** Xây dựng CSDL phù hợp để lưu trữ dữ liệu tạo thành kho dữ liệu và phân ra thành những kho dữ liệu theo chủ đề (Data mart) gồm thể loại, lĩnh vực, năm…

- **Bước 3:** Xây dựng hệ thống phát hiện sao chép để khai thác dữ liệu từ kho dữ liệu này, và có thể ứng dụng hệ thống OLAP, kỹ thuật khai phá dữ liệu (Data mining)… để khai thác nguồn dữ liệu này trong các nghiên cứu về xử lý văn bản, tăng tính chính xác và hiệu quả trong tìm kiếm thông tin...



Hình 4.2. Kiến trúc hệ thống kho dữ liệu chi tiết

4.2.2.2. Phương pháp xử lý tiếng Việt

Tiếng Việt mang nhiều đặc trưng ngôn ngữ riêng nên phức tạp hơn so với tiếng Anh và các ngôn ngữ khác như: Phân biệt các từ không xác định bằng dấu cách (như trong tiếng Anh phân biệt các từ bằng dấu cách), vì một từ có thể bao gồm nhiều âm tiết và chỉ mỗi âm tiết mới phân biệt bằng dấu cách. Điều này dẫn đến nhập nhằng trong phân đoạn từ tiếng Việt; ngữ pháp tiếng Việt phức tạp cùng với hiện tượng đồng âm khác nghĩa, từ đồng nghĩa hay thay đổi trật tự câu và các phép tu từ… dẫn đến nhập nhằng trong xác định ý nghĩa của câu, từ.

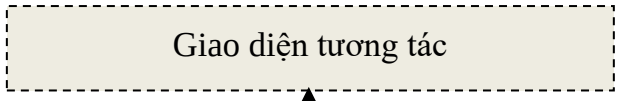
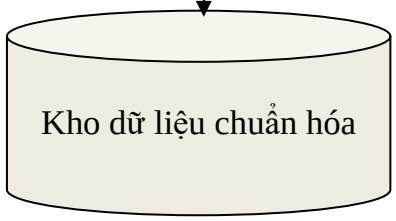
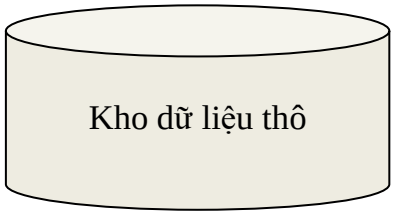
Hiện nay, ở Việt Nam chưa có một kho dữ liệu chuẩn để phục vụ cho các nghiên cứu về xử lý văn bản tiếng Việt. Vì vậy, với nghiên cứu của mình, chúng tôi

thiết kế, xây dựng kho dữ liệu trong đó có ứng dụng NLP cho tiếng Việt như chuẩn hóa văn bản thành chữ thường, loại bỏ từ dừng, loại bỏ các thông tin không cần thiết, loại bỏ hình ảnh, tách đoạn, tách câu, tách từ lưu vào từ điển và đánh chỉ mục lưu trữ vào CSDL,...

4.2.3. Giải pháp xây dựng kho dữ liệu

Sau khi khảo sát và phân tích các yêu cầu, luận án đề xuất các giải pháp xây dựng kho dữ liệu như sau:

4.2.3.1. Quy trình xây dựng kho dữ liệu

Các bước	Tiến trình
4	
Khai thác kho dữ liệu	
3	
Xử lý, cập nhật dữ liệu vào kho, biểu diễn dữ liệu, mã hóa thành các DNA nguồn, lưu trữ dưới dạng vector...	
2	
Tiền xử lý dữ liệu (chuẩn hóa dữ liệu, font chữ, loại bỏ các thông tin không cần thiết…)	
1	
Thu thập dữ liệu từ nguồn luận văn, bài báo, DSpace, web…	

Hình 4.3. Quy trình xây dựng kho dữ liệu

- **Bước 1**: Thu thập nguồn dữ liệu

+ Tài liệu gồm các đồ án tốt nghiệp của sinh viên, luận văn thạc sĩ, luận án tiến sĩ, bài báo khoa học… của cán bộ, giảng viên ĐHĐN với định dạng có thể xử lý đưa vào kho là tập tin .doc, .docx...; nguồn dữ liệu trên hệ thống DSpace.

+ Mở rộng cập nhật từ nguồn Internet, thu thập từ các thư viện, kho luận văn.

- **Bước 2**: Tiền xử lý dữ liệu

+ Xử lý dữ liệu đầu vào phù hợp để lưu trữ trong kho như: Chuyển đổi font chữ, chuẩn hóa về chữ thường, chuẩn hóa dấu, loại bỏ từ dừng, loại bỏ khoảng trắng, chuyển đổi cấu trúc, thể thức văn bản của tài liệu về dạng chuẩn...

- **Bước 3**: Chuẩn hóa dữ liệu

+ Tách câu, mã hóa, đánh chỉ mục, xử lý, cập nhật và lưu trữ dữ liệu vào CSDL.

+ Tách từ lưu vào từ điển để tính mức độ từ giống nhau.

+ Mã hóa dữ liệu thành các DNA nguồn và lưu trữ dưới dạng vector.

- **Bước 4**: Khai thác dữ liệu

+ Thiết kế giao diện để có thể khai thác các thông tin dữ liệu trong kho.

+ Phục vụ hệ thống phát hiện sao chép.

4.2.3.2. Xử lý tài liệu phục vụ phát hiện sao chép

Mục đích chính của xây dựng kho dữ liệu là để phục vụ hệ thống phát hiện sao chép, hệ thống sau này sẽ băm tài liệu thành những đơn vị so sánh (chunk) để xác định sự chồng chéo và phát hiện khả năng có sao chép. Có nhiều phương pháp để chọn đơn vị so sánh như so sánh câu, đoạn hoặc toàn bộ văn bản.

Bước đầu tiên để đánh giá phương pháp đề xuất cho so sánh văn bản, trong luận án đã lựa chọn tổ chức lưu trữ dữ liệu theo đơn vị câu và từ để phục vụ cho hệ thống phát hiện sao chép sử dụng làm đơn vị so sánh.

Để so sánh được nội dung văn bản, dữ liệu cần được lưu trữ ở dạng văn bản text và qua quá trình xử lý như sau:

- **Bước 1** (Tiền xử lý): Làm sạch dữ liệu nhằm tối ưu hơn giảm thiểu thời gian trong quá trình phát hiện sao chép bằng cách lọc bỏ những phần trong tài liệu như: mục lục, hình ảnh, công thức toán, câu ngắn... thực hiện chuyển đổi font chữ, chuẩn hóa về chữ thường, chuẩn hóa dấu, loại bỏ từ dừng, loại bỏ khoảng trắng, chuyển đổi cấu trúc, thể thức văn bản của tài liệu về dạng chuẩn.

- **Bước 2** (Tách đoạn): Một văn bản được xem như tập các đoạn. Khi tách đoạn chủ yếu dựa vào định dạng của văn bản để phát hiện biên giới của văn bản. Việc tách đoạn được thực hiện bằng cách nhận biết ký tự xuống dòng. Trong quá trình tách đoạn, chúng tôi loại bỏ những đoạn có độ dài nhỏ như: tựa đề, tiêu đề... nhằm tăng độ chính xác và giảm thời gian xử lý cho hệ thống phát hiện sao chép.

- **Bước 3** (Tách câu): Xác định ranh giới câu qua dấu câu như dấu chấm (.), dấu chấm than (!), dấu chấm hỏi (?) và dấu chấm phẩy (;)… thực hiện tách câu.

- **Bước 4** (Tách từ): Ngoài ra, chúng tôi cũng thực hiện tách văn bản thành danh sách các từ, đánh dấu và lưu trữ trong CSDL như bộ từ điển để biểu diễn tài liệu/từ trong mô hình vector phục vụ xử lý, tính toán độ tương tự.

- **Bước 5** (Đánh chỉ mục): Để lần vết trong quá trình tìm kiếm, xác định độ tương tự về sau thì việc đánh chỉ mục (Indexing) là bước rất quan trọng, các tài liệu, các câu trong từng tài liệu được mã hóa và đánh chỉ mục nhằm tăng tốc độ tìm kiếm, truy xuất.

Các thao tác trên được thực hiện bằng các hàm hỗ trợ trong thư viện String của ngôn ngữ C# và xây dựng một số thuật toán xử lý khác để tổ chức lưu trữ, tìm kiếm, truy xuất dữ liệu kết hợp với xử lý đa luồng, song song, đồng bộ, tránh xung đột để giúp tăng tốc độ và hiệu năng của ứng dụng.

4.2.3.3. Cập nhật dữ liệu

Cập nhật dữ liệu vào kho có tính chất quan trọng, việc cập nhập dữ liệu giúp kho dữ liệu có nguồn dữ liệu phong phú với nhiều chuyên ngành, thể loại và lĩnh vực khác nhau. Giúp kho dữ liệu có độ bao phủ lớn phục vụ cho phát hiện sao chép.

Cập nhập kho dữ liệu thông qua giao diện của ứng dụng gồm ba bước sau:

- **Bước 1**: Người sử dụng chọn đường dẫn đến tập tài liệu và thực hiện upload.
- **Bước 2**: Hệ thống sẽ xử lý và sao lưu tập tin gốc vào kho.

- **Bước 3**: Xử lý chuyển đổi dữ liệu lưu trữ vào kho.

4.2.3.4. Tổng hợp nguồn tài liệu

a) Giới thiệu

Bên cạnh việc khảo sát, nghiên cứu, thiết kế, xây dựng kho dữ liệu chứa các tài liệu là các luận văn, đồ án, bài báo khoa học... mang ý nghĩa thực tiễn của ĐHĐN, luận án đã nghiên cứu, đề ra giải pháp để tổng hợp nguồn tài liệu lớn của ĐHĐN, đề xuất này mang ý nghĩa thực tiễn cao và giúp cho kho dữ liệu có độ bao phủ lớn. Và nguồn tài liệu tổng hợp sẽ được cập nhật vào kho dữ liệu để phục vụ cho hệ thống phát hiện sao chép văn bản.

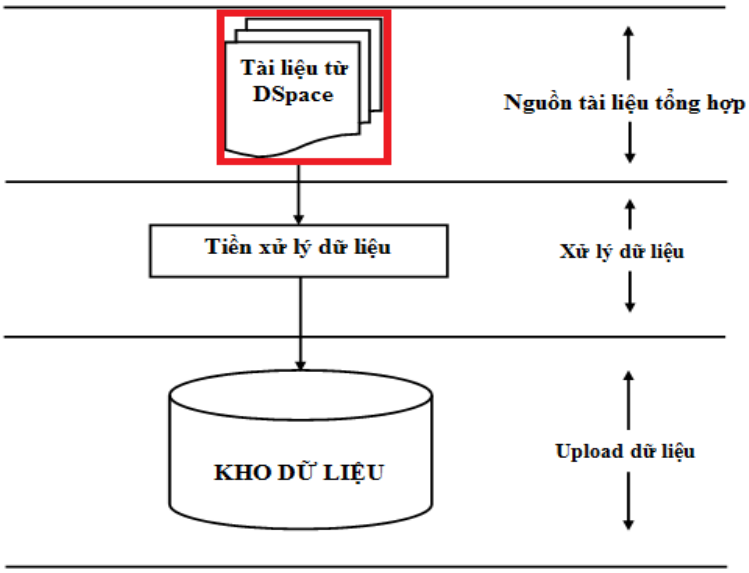
b) Đề xuất giải pháp triển khai

Từ việc khảo sát thực tế, với mục đích quản lý, lưu trữ và công bố các tài liệu khoa học như: Bài báo, luận văn, luận án, đồ án tốt nghiệp, đồ án môn học... của nhiều cơ sở giáo dục thành viên và đơn vị thuộc ĐHĐN, nên việc xây dựng một ứng dụng dùng chung cho toàn ĐHĐN sẽ gặp khó khăn. Qua thực tế và nghiên cứu, chúng tôi chọn cách xây dựng một ứng dụng để có thể triển khai được cho nhiều đơn vị và các ứng dụng đó được tổng hợp tại một địa chỉ sẽ là giải pháp khả thi hơn.

Qua nghiên cứu, chúng tôi cải tiến bộ phần mềm nguồn mở DSpace đang được rất nhiều trường đại học, thư viện trên thế giới và ở Việt Nam sử dụng với nhiều ưu điểm cho việc lưu trữ, quản lý nguồn tài liệu số để triển khai cho các đơn vị trong ĐHĐN.

Và để tập trung các DSpace từ các đơn vị, trong luận án đã xây dựng trang tài nguyên số để tổng hợp tất cả các hệ thống DSpace của ĐHĐN như một cổng thông tin tổng hợp nguồn tài liệu khoa học. Giải pháp này giúp dễ dàng tìm kiếm, truy cập vào các DSpace thông qua một địa chỉ duy nhất (http://tainguyenso.udn.vn).

Với kết quả triển khai các hệ thống DSpace, luận án đã đề xuất được giải pháp để tổng hợp được nguồn tài liệu lớn của ĐHĐN phục vụ cho việc học tập, nghiên cứu. Bên cạnh đó, chúng tôi lấy nguồn dữ liệu này để cập nhật vào hệ thống kho dữ liệu phục vụ phát triển hệ thống phát hiện sao chép văn bản.



Hình 4.4. Quy trình xử lý, cập nhật tài liệu vào kho dữ liệu

4.2.4. Đánh giá về kho dữ liệu

Luận án đã đề xuất giải pháp và xây dựng hệ thống kho dữ liệu giải quyết vấn đề thực tế của ĐHĐN và có độ bao phủ cao. Kết quả thử nghiệm, bước đầu chúng tôi đã cập nhật vào kho dữ liệu gần 2.000 tài liệu thuộc 6 lĩnh vực theo quy định của Bộ Khoa học và Công nghệ và phân theo các thể loại để phục vụ mục đích thử nghiệm cho hệ thống phát hiện sao chép. Dữ liệu này, chúng tôi sẽ tiếp tục cập nhật từ các nguồn dữ liệu của ĐHĐN để phục vụ cho việc kiểm tra sau này.

Bảng 4.1. Số tài liệu thử nghiệm được cập nhật vào kho dữ liệu

STT	Loại tài liệu	Số lượng (tập tin)	Định dạng
1	Đồ án tốt nghiệp sinh viên	1100	.doc/.docx
2	Luận án tiến sĩ	50	.doc/.docx
3	Luận văn thạc sĩ	400	.doc/.docx
4	Báo cáo khoa học	50	.doc/.docx
5	Bài báo khoa học	400	.doc/.docx

Đề xuất giải pháp tổng hợp nguồn tài liệu lớn của ĐHĐN phục vụ hệ thống phát hiện sao chép với số tài liệu được cập nhật từ các đơn vị thành viên rất nhiều. Bên cạnh đó, ĐHĐN hiện đang triển khai hệ thống thư viện trực tuyến có tích hợp

các hệ thống DSpace của ĐHĐN và liên kết với rất nhiều nguồn dữ liệu khác để tạo nên một lượng lớn các tài liệu được lưu trữ[10]. Thời gian tới, các trường đại học trong ĐHĐN cập nhật các tài liệu lên hệ thống DSpace và hệ thống thư viện trực tuyến sẽ tạo thành nguồn dữ liệu phong phú, dồi dào để có thể ứng dụng, khai thác cho nhiều mục đích phục vụ nghiên cứu và học tập.

Đây là bước chuẩn bị quan trọng cho việc đề xuất thuật toán giải quyết bài toán phát hiện sao chép. Việc lựa chọn cách biểu diễn văn bản và xây dựng kho dữ liệu có ý nghĩa thực tiễn cao nhằm áp dụng giải pháp đề xuất ứng dụng vào thực tiễn. Tuy nhiên, qua kết quả thực nghiệm trên mô hình vector và phương pháp lựa chọn để đo độ tương tự bằng Cosine mặc dù cơ bản đã giải quyết được vấn đề đặt ra, nhưng vẫn chưa mang lại kết quả khả quan và chưa thể ứng dụng để xử lý dữ liệu lớn. Việc biến đổi dữ liệu thô thành dữ liệu được biểu diễn theo mô hình vector có những hạn chế như không gian lưu trữ với số chiều rất lớn, đối với kho dữ liệu lớn thì chi phí xử lý, tính toán cao. Chính vì vậy, luận án đã nghiên cứu và đề xuất giải pháp để vừa có thể vận dụng các kết quả nghiên cứu tiền đề này và cải tiến, có hướng tiếp cận mới để đạt được kết quả tốt và hiệu quả hơn nhằm giải quyết vấn đề đặt ra trong đánh giá mức độ giống nhau của văn bản tiếng Việt.

4.3. Triển khai hệ thống phát hiện sao chép văn bản

Với những nghiên cứu đạt được, chúng tôi tiến hành phát triển hệ thống phát hiện sao chép văn bản thử nghiệm được đặt tại địa chỉ: http://bigdata.udn.vn.

Hệ thống được xây dựng trên nền web để dễ dàng tương tác với người dùng và sử dụng các dịch vụ web để giao tiếp giữa các máy chủ và ứng dụng web. Trong thử nghiệm, chúng tôi cài đặt, triển khai trên hệ thống máy chủ của ĐHĐN để thiết lập các kết nối, các điểm để có thể lưu trữ dữ liệu, xử lý đa luồng cho các tác vụ như tạo các DNA nguồn, tải dữ liệu, lập chỉ mục, tìm kiếm, so khớp, đánh dấu, truy xuất

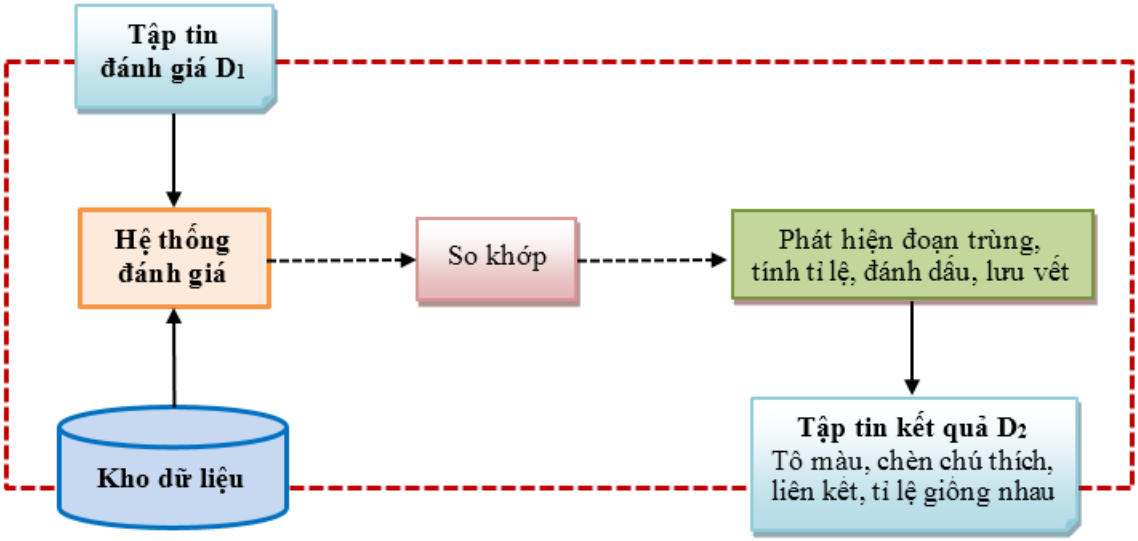
[10] http://libs.udn.vn (truy cập tháng 10/2019)

tài liệu nguồn,… Việc triển khai trên nền hệ thống máy chủ này làm tăng khả năng truy xuất, linh hoạt và dễ dàng thay đổi, cập nhật sau này.

Với yêu cầu đặt ra, hệ thống gồm hai phần chính đó là kho dữ liệu và phát hiện sao chép văn bản. Kho dữ liệu với các chức năng upload tài liệu vào kho, xem các tài liệu đã upload và thống kê các tài liệu đã được xử lý. Hệ thống phát hiện sao chép với chức năng cho phép upload tài liệu cần kiểm tra và theo dõi thống kê các tài liệu kiểm tra phát hiện sao chép. Khi kiểm tra, ta có thể chọn lĩnh vực để khoanh vùng tìm kiếm được nhanh hơn và chọn cấp độ so sánh với mức độ trung bình tương ứng với so sánh theo đoạn văn bản và mức độ chi tiết sẽ so sánh kỹ hơn theo từng câu. Dưới đây là hình giao diện của hệ thống phát hiện sao chép văn bản được thử nghiệm tại ĐHĐN.

Hình 4.5. Giao diện hệ thống thử nghiệm

Bên cạnh hiển thị kết quả so khớp trên giao diện web, luận án đã đề xuất thuật toán đánh dấu nội dung tài liệu sao chép trực tiếp trên tập tin tài liệu cần đánh giá. Ví dụ, để tiến hành so khớp giữa tập tin đầu vào Word (.doc, .docx) và các tài liệu có trong kho dữ liệu của hệ thống, việc phát hiện các câu, đoạn văn bản giống nhau sẽ được thực hiện như mô hình như sau:



Hình 4.6. Mô hình phát hiện đánh dấu nội dung giống nhau

Các bước thực hiện để phát hiện và đánh dấu nội dung văn bản giống nhau:

- **Bước 1:** Khoanh vùng tìm kiếm. Để tiến hành so khớp giữa tập tin đầu vào (.doc hoặc .docx) và kho dữ liệu (các DNA nguồn) của hệ thống, tùy theo lĩnh vực, chuyên ngành của văn bản đầu vào, người dùng có thể chọn lĩnh vực tương ứng để khoanh vùng, giới hạn số tài liệu có trong kho dữ liệu khi so khớp giúp cho việc tìm kiếm nhanh hơn.

- **Bước 2:** Thực hiện so khớp các đoạn DNA của văn bản cần kiểm tra (tập tin đầu vào) với các DNA nguồn được lưu trữ, đánh chỉ mục trong cơ sở dữ liệu đã được sắp xếp.

- **Bước 3:** Tính tỷ lệ giống nhau và lưu vết để truy xuất đánh dấu. Sau khi kiểm tra so khớp các đoạn DNA của văn bản cần kiểm tra với các DNA nguồn được lưu trữ trong hệ thống, công cụ sẽ lưu lại vết các đoạn giống nhau để thực hiện truy xuất trong các tài liệu nguồn.

- **Bước 4:** Từ vết lưu lại của các câu, đoạn bị đánh giá là giống nhau từ nguồn các tài liệu khác có trong hệ thống, công cụ sẽ đánh dấu ghi chú thích câu, đoạn đó giống nhau tài liệu nào, của tác giả nào, năm và tỷ lệ giống nhau của tài liệu trên.

- **Bước 5:** Đánh dấu và tô màu. Tùy vào mức độ giống nhau của câu, đoạn mà hệ thống sẽ thực hiện thay đổi định dạng và tô màu các đoạn văn bản đó với các mức như sau:

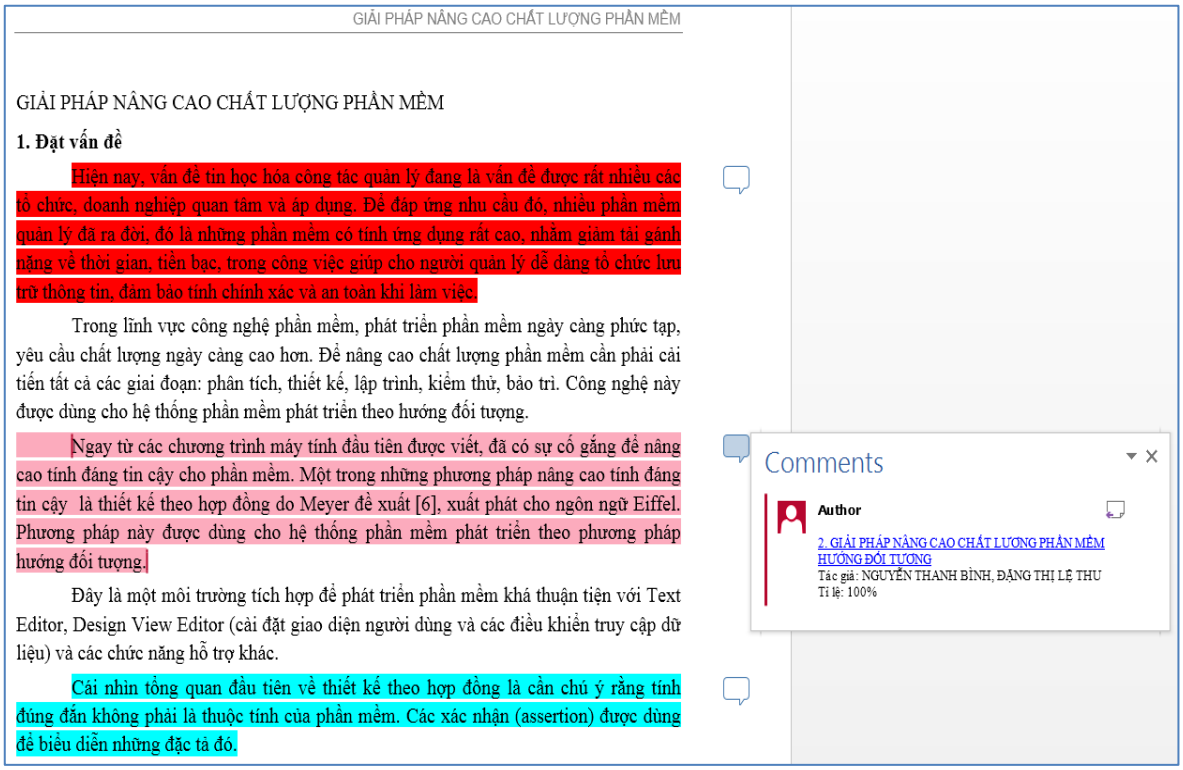
+ Sao chép từ 90% đến 100%: màu đỏ;
+ Sao chép từ 70% đến 89% : màu xanh;
+ Sao chép từ 50% đến 69% : màu vàng.

Giải thuật đánh dấu và tô màu văn bản được thực hiện như thuật toán sau:

Thuật toán 4.1. Đánh dấu và tô màu văn bản

	Đầu vào: Văn bản (tập tin .doc hay .docx).
	Đầu ra: Văn bản có đánh dấu, tô màu các câu nghi ngờ sao chép và tham chiếu đến tài liệu nguồn bị sao chép.
	Xử lý:
1	n = CountSent(D_1) // *Số lượng câu của tập tin cần kiểm tra D_1*
2	**For** i:= 1 → n
3	m = length(W) // *Số lượng câu trong kho dữ liệu*
4	Extract S_i // *Tách câu thứ i trong D_1*
5	Encode S_i // *Mã hóa câu thứ i trong D_1 thành DNA*
6	**For** j:= 1 → m
7	$S_j = DNA_j$ // *DNA của câu thứ j trong W*
8	**If** So_khop(S_i, S_j) trùng nhau (90%-100%) **then**
9	Chèn note, tô màu đỏ.
10	**Endif**
11	**If** So_khop(S_i, S_j) trùng nhau (70%-89%) **then**
12	Chèn note, tô màu xanh.
13	**Endif**
14	**If** So_khop(S_i, S_j) trùng nhau (50%-69%) **then**
15	Chèn note, tô màu vàng.
16	**Endif**
17	**Endfor** // *Kết thúc vòng lặp for dòng 6*
18	**Endfor** // *Kết thúc vòng lặp for dòng 2*

Một ví dụ về đoạn văn bản giống nhau sẽ được đánh dấu, tô màu và chú thích như hình sau:



Hình 4.7. Đánh dấu nội dung giống nhau trên tài liệu cần kiểm tra

Kết quả đạt được qua thực nghiệm: Luận án đã xây dựng hệ thống kho dữ liệu và phát triển hệ thống phát hiện sao chép văn bản tiếng Việt đã đưa vào sử dụng thử nghiệm đạt được những kết quả cụ thể sau:

- Về cơ bản, hệ thống đã đáp ứng được các yêu cầu đặt ra và kết quả phát hiện các đoạn văn bản giống nhau có độ chính xác cao.

- Về kho dữ liệu, đến thời điểm hiện tại, trong kho dữ liệu đánh giá đã có gần 2.000 tài liệu bao gồm các luận án tiến sĩ, luận văn thạc sĩ, luận văn tốt nghiệp sinh viên, báo cáo khoa học, bài báo khoa học... thuộc nhiều lĩnh vực khác nhau. Trong thời gian đến, chúng tôi sẽ tiếp tục cập nhật thêm các tài liệu mới vào kho để độ bao phủ cao hơn.

- Về giải thuật, trong quá trình nghiên cứu, luận án đã triển khai thử nghiệm và đánh giá nhiều giải thuật khác nhau. Vấn đề đặt ra là khi số lượng tài liệu tăng lên (hàng triệu tài liệu) và số lượng người dùng nhiều (không giới hạn số lượng) thì thời gian xử lý và trả về kết quả cho người dùng là vô cùng quan trọng.

Như vậy, với những kết quả nghiên cứu, giải pháp đề xuất và ứng dụng thực tế đã đạt được trong nội dung nghiên cứu đã đáp ứng được những yêu cầu quan trọng cho việc đánh giá mức độ giống nhau của văn bản tiếng Việt ứng dụng cho hệ thống phát hiện sự giống nhau của văn bản. Tuy nhiên, để tăng độ chính xác và khả năng xử lý dữ liệu lớn, chúng tôi sẽ tiếp tục nghiên cứu và triển khai thêm nhiều giải pháp để nâng cao hiệu quả của ứng dụng.

4.4. Đề xuất hướng phát triển để xử lý dữ liệu lớn

4.4.1. Giới thiệu

Trong hệ thống phát hiện sao chép, để thực hiện nhiệm vụ so khớp, đối sánh văn bản sẽ dựa trên các tài liệu nguồn nằm trong một bộ sưu tập lớn các tài liệu (tức là kho dữ liệu). Và dữ liệu sẽ liên tục được cập nhật, không ngừng tăng lên và có thể vượt ra khỏi khả năng xử lý của các hệ thống truyền thống với việc tính toán tuần tự trên các máy tính riêng lẻ thì khó đáp ứng được yêu cầu. Điều này đòi hỏi phải tính tới một hệ thống có khả năng xử lý trên lượng dữ liệu lớn (Big data) và thực thi trong một khoảng thời gian chấp nhận được.

Để xây dựng một hệ thống có thể ứng dụng vào thực tế, chúng tôi cần phải sưu tập và xử lý một khối lượng dữ liệu rất lớn (gồm các báo cáo luận văn, đồ án, bài báo, đề tài khoa học...) từ nhiều nguồn khác nhau và ứng dụng các kỹ thuật Big data để xử lý và áp dụng các giải pháp so sánh văn bản phù hợp.

Vì vậy, ngoài các thuật toán xử lý dữ liệu lớn một cách hiệu quả đã được trình bày chi tiết ở Chương 3, phần tiếp theo luận án đề xuất hướng phát triển để xử lý nguồn dữ liệu lớn này theo hai giải pháp: Một là giải pháp xử lý dựa trên hệ thống máy tính, xử lý phân tán, tính toán song song...; hai là giải pháp biểu diễu dữ liệu đa chiều theo mô hình Tensor.

4.4.2. Đề xuất giải pháp xử lý

Từ những nghiên cứu về kỹ thuật xử lý Big data theo mô hình xử lý phân tán, chúng tôi hướng đến giải pháp xử lý nguồn dữ liệu lớn và phân tán dựa trên ứng dụng MapReduce.

- **Ý tưởng bài toán**: Ứng dụng mô hình MapReduce để tổ chức, lưu trữ các tài liệu với số lượng lớn và nằm phân tán nhiều nơi vào trong kho dữ liệu bằng cách xây dựng hệ thống để thu thập, tách các tài liệu và lưu trữ vào kho phục vụ phát hiện sao chép.
- **Phân tích yêu cầu**:

+ Hệ thống tách tài liệu làm cơ sở dữ liệu bao gồm một máy master (máy chủ) và nhiều máy slave (máy trạm).

+ Hệ thống tập tin phân tán HDFS sẽ lưu trữ các văn bản cần tách.

+ Với hệ thống này cho phép ta có thể chọn nhiều kiểu tập tin khác nhau và đặt phân tán để phân tích (có thể lấy từ các nguồn dữ liệu DSpace đã xây dựng).

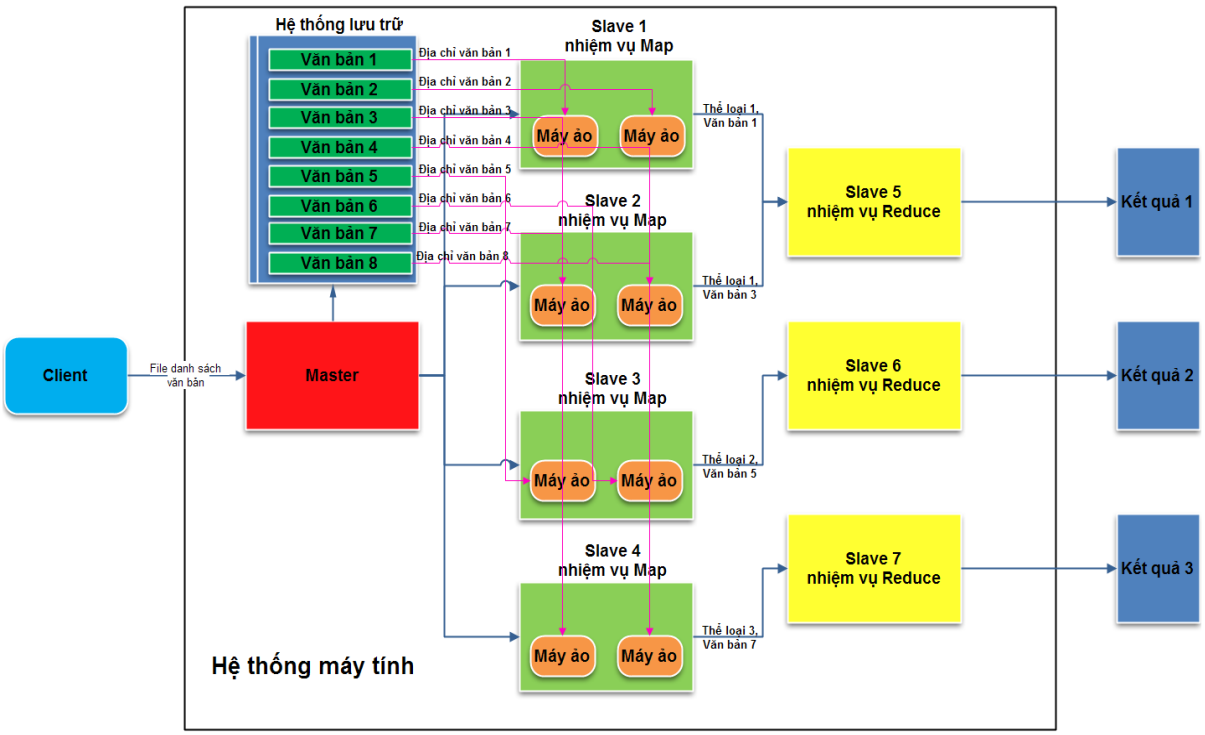
+ Máy client sẽ gửi danh sách tập tin văn bản và yêu cầu thực hiện tách đoạn, câu, từ để làm cơ sở dữ liệu.

+ Máy master sẽ xem xét những máy slave nào sẵn sàng và phân phối, gửi địa chỉ văn bản đến để làm nhiệm vụ Map.

+ Các máy slave làm nhiệm vụ Map sẽ tải văn bản từ HDFS và tiến hành tách đoạn, câu, từ. Kết quả của quá trình Map sẽ được gửi đến các máy slave để làm nhiệm vụ Reduce.

+ Kết quả phân tích cũng chính là kết quả của quá trình Reduce. Như vậy, ở các máy slave vừa làm nhiệm vụ Map, đồng thời vừa làm nhiệm vụ Reduce.

+ Ở các máy slave, chúng tôi cài đặt thêm một số máy ảo. Tùy vào cấu hình máy chủ, việc cài đặt nhiều máy ảo sẽ giúp giảm thời gian phân tích, tăng hiệu quả xử lý. Mục đích của việc cài đặt máy ảo là tạo ra môi trường an toàn không ảnh hưởng đến hệ thống khác. Các máy ảo này được lập trình để có thể chạy tự động mà không cần sự can thiệp của con người.



Hình 4.8. Mô hình hệ thống tách và lưu trữ tài liệu theo MapReduce

- **Nhiệm vụ của các máy Map:**

+ Các máy slave làm nhiệm vụ Map sẽ nhận đầu vào là một cặp <key, value> với key là tên văn bản, value là địa chỉ của văn bản. Dựa vào địa chỉ này, các máy cục bộ sẽ tải các văn bản về thực hiện.

+ Các máy slave này sẽ thực hiện việc tách đoạn, câu, từ và xử lý, đánh chỉ mục, lưu trữ vào CSDL.

- **Nhiệm vụ của các máy Reduce:** Sắp xếp lại theo thể loại và lưu vào CSDL. Với *n* máy làm nhiệm vụ Reduce thì tương ứng sẽ có *n* kết quả và *n* lần đưa vào trong CSDL. Nhờ xử lý phân tán mà tốc độ và hiệu năng của hệ thống sẽ tăng lên nhiều.

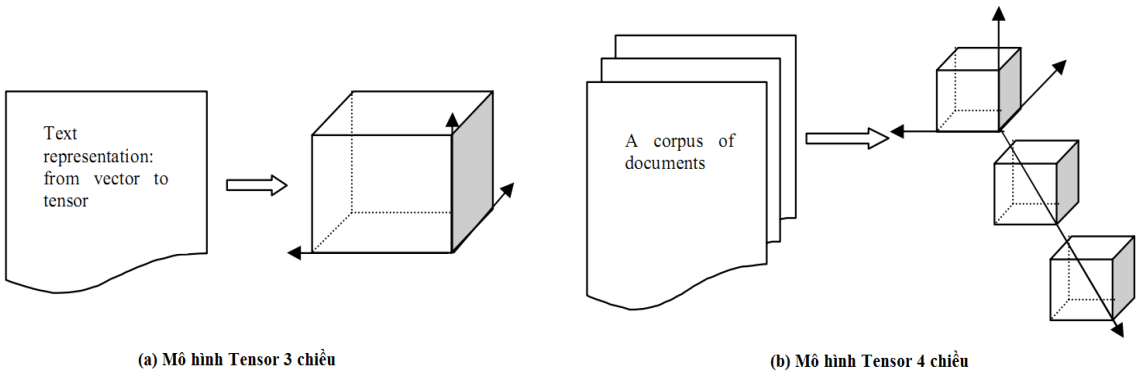
4.4.3. Đề xuất phương pháp biểu diễn DNA bằng Tensor

Biểu diễn Tensor được phát triển dựa trên mô hình vector, ma trận, nhưng mô hình Tensor là cách biểu diễn dữ liệu đa chiều. Với những tiến bộ trong khả năng thu thập và lưu trữ dữ liệu lớn đa chiều, gọi là biểu diễn dữ liệu bậc cao, đã được tạo ra trên các ứng dụng tin sinh học trong thời gian gần đây, đặc biệt là trong việc nhận dạng chuỗi DNA. Mô hình Tensor biểu diễn dữ liệu đa chiều có khả năng phát hiện

sự giống nhau trên nhiều chiều để tìm độ tương tự DNA với độ chính xác và độ tin cậy cao [67, 86].

Bên cạnh đó, biểu diễn Tensor đã có nhiều nghiên cứu và kết quả tối ưu trong việc trích xuất đặc trưng và phân lớp dữ liệu lớn, đặc biệt là giải quyết các bài toán về gen [96].

Qua nghiên cứu, chúng tôi có ý tưởng để hướng đến giải quyết bài toán so sánh văn bản theo phương pháp đã đề xuất để biểu diễn các DNA nhiều chiều dựa trên các đặc tính của văn bản về chủ đề, lĩnh vực, thể loại, thời gian... đề xuất giải pháp học máy để phân loại và biểu diễn văn bản qua các chuỗi DNA. Biểu diễn văn bản dựa trên mô hình Tensor là một ý tưởng mới mẻ nhằm giải quyết các vấn đề về dữ liệu lớn nhằm tận dụng những nghiên cứu tương tự như các bài toán về gen trong lĩnh vực tin sinh học và y sinh. Chúng tôi nhận thấy đây là hướng nghiên cứu đầy triển vọng để giải quyết bài toán theo giải pháp đề xuất của mình và góp phần đóng góp cho các nghiên cứu về xử lý văn bản theo hướng tiếp cận mới và đem hiệu quả cao hơn. Dưới đây là mô hình biểu diễn văn bản theo Tensor.



Hình 4.9. Biểu diễn tài liệu theo mô hình Tensor [71]

4.5. Kết luận Chương 4

Trong nội dung của Chương này đã trình bày phương pháp xây dựng kho dữ liệu mang ý nghĩa thực tiễn. Dữ liệu tuy được thu thập và cập nhật với số lượng chưa nhiều, nhưng nguồn dữ liệu có thể mở rộng và cập nhật sau này một cách dễ dàng. Bên cạnh nguồn dữ liệu thu thập được, luận án đã phát triển hệ thống phát hiện sao chép văn bản với các phương pháp đề xuất liên quan về DWT và kết hợp với phương pháp biểu diễn văn bản bằng vector cho việc lưu trữ và đánh dấu dữ liệu nhằm hỗ trợ

tìm kiếm và truy xuất các tài liệu nguồn. Hệ thống đang được triển khai thử nghiệm tại ĐHĐN. Kết quả thử nghiệm trong Chương 4 có các phần nội dung liên quan đến các công trình công bố trên tạp chí trong nước và quốc tế, hội nghị khoa học quốc gia và quốc tế[11].

[11] (1) Hồ Phan Hiếu, Trần Thanh Liêm, *Giải pháp hệ thống hóa tên miền và nguồn tài liệu khoa học của Đại học Đà Nẵng*. Tạp chí Khoa học và Công nghệ ĐHĐN, Số 12(97), 2015, (20-24).

(2) Phan Hieu Ho, Trung Hung Vo, Ngoc Anh Thi Nguyen, *Data Warehouse Designing for Vietnamese Textual Document-based Plagiarism Detection System*. Hội nghị quốc tế IEEE International Conference on System Science and Engineering (ICSSE 2017), 2017, (254-258). DOI: 10.1109/ICSSE.2017.8030873

(3) Hồ Phan Hiếu, Nguyễn Thị Ngọc Anh, Võ Trung Hùng, *Phương pháp mã hóa văn bản thành chuỗi số DNA để đánh giá mức độ giống nhau của văn bản*. Hội thảo Khoa học Quốc gia Công nghệ thông tin và ứng dụng trong các lĩnh vực - lần thứ 7 (CITA2018), (223-229).

(4) Phan Hieu Ho, Trung Hung Vo, Ngoc Anh Thi Nguyen, Ha Huy Cuong Nguyen, *A Narrative Method for Evaluating Documents Similarity based on Unique Strings*. International Journal of Recent Technology and Engineering (IJRTE), Vol 8(2S11), 2019, (473-479). DOI: 10.35940/ijrte.B1073.0982S1119

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Luận án đã nghiên cứu khá toàn diện về cách đo độ tương tự của văn bản và ứng dụng nó vào phát hiện sao chép. Những đóng góp mới của luận án và kết quả thực hiện có thể tóm tắt như sau:

- Đã khảo sát, phân tích, đề xuất những nội dung liên quan đến so khớp văn bản dựa trên mô hình vector, thực nghiệm trên tập dữ liệu thử nghiệm tiếng Việt và cho kết quả khả quan để chứng minh phương pháp dựa trên mô hình vector và sử dụng độ đo Cosine là phương pháp thông dụng có thể giải quyết được bài toán tính độ tương tự văn bản.

- Đề xuất quy trình số hóa văn bản bằng cách chuyển văn bản thành các chuỗi số thực DNA dựa trên phương pháp DWT và bộ lọc Haar. Đây là một cách tiếp cận hoàn toàn mới để giải quyết bài toán phát hiện sự giống nhau của văn bản.

- Đề xuất quy trình xử lý, xây dựng thuật toán phát hiện sự giống nhau giữa các văn bản bằng cách tính toán khoảng cách Euclid nhỏ nhất từ DNA cần đánh giá đến các DNA nguồn và so sánh với một mức ngưỡng thích hợp để đưa ra sự giống nhau giữa văn bản được kiểm tra với văn bản nguồn trong kho dữ liệu. Các kết quả thực nghiệm trên bộ dữ liệu chuẩn của PAN và bộ dữ liệu tiếng Việt thử nghiệm đã chứng minh thuật toán được đề xuất trong luận án đem lại hiệu quả cao trong phát hiện sự giống nhau của văn bản.

- Đã hướng đến xử lý dữ liệu lớn một cách hiệu quả với việc mã hoá dữ liệu văn bản sang chuỗi DNA, tổ chức lưu trữ theo dạng vector và sắp xếp theo thứ tự tăng dần các DNA để có thể thực hiện việc tìm kiếm nhị phân, đây là một trong những phương pháp tìm kiếm nhanh nhất khi làm việc với dữ liệu lớn. Hơn nữa, DWT cho độ phức tạp tính toán chỉ là hàm tuyến tính trong mỗi lần lấy mẫu con nên giải pháp đề xuất sẽ càng hiệu quả trong quá trình xử lý dữ liệu lớn.

- Thực nghiệm xây dựng kho dữ liệu và hệ thống phát hiện sao chép văn bản và triển khai ứng dụng thử nghiệm tại ĐHĐN.

Tóm lại, điểm nổi bật trong luận án là đề xuất nhiều phương pháp để giải quyết yêu cầu bài toán như: 1) Tìm cách để chuyển văn bản thành số nguyên đại diện đảm bảo đặc trưng của văn bản, giữ đúng vị trí ký tự sử dụng mã Unicode; 2) Chuyển số nguyên thành số thực nhỏ để dễ dàng lưu trữ và tính toán bằng cách đề xuất công thức tính toán sử dụng hàm logarit cơ số 10; 3) Lấy mẫu cho các phân đoạn để làm đầu vào cho bộ lọc Haar, đề xuất thuật toán để lấy mẫu, dịch cửa sổ trượt, tính toán số bước thực hiện tối ưu; 4) Đề xuất thuật toán sắp xếp các DNA theo thứ tự tăng dần để phục vụ tìm kiếm nhị phân gần đúng, xử lý dữ liệu lớn; 5) Tìm ra phương pháp tổ chức dữ liệu hiệu quả để dễ thực hiện việc so khớp và truy xuất kết quả; 6) Đề xuất thuật toán để so sánh và đánh giá mức độ giống nhau, chỉ ra các đoạn giống và tô màu theo các mức độ trùng. Như vậy, với mỗi đề xuất trên đều thể hiện được tính mới và có đóng góp vào ý tưởng, phương pháp chung để giải quyết bài toán phát hiện sao chép văn bản hiệu quả.

Mặc dù đã đạt được những kết quả khả quan nhưng luận án vẫn còn một số hạn chế như:

- Phương pháp dựa trên DWT và bộ lọc Haar tập trung vào độ chính xác và xử lý dữ liệu lớn nên chưa thể đánh giá về mặt ngữ nghĩa. Ngoài ra, phương pháp đề xuất này dựa trên đặc tính sắp xếp dữ liệu theo chuỗi thời gian thực, do đó trong trường hợp thay đổi thứ tự các từ trong tài liệu đáng ngờ thì hiệu quả sẽ thấp.

- Luận án chưa giải quyết một số vấn đề liên quan trong sao chép như: ngữ nghĩa (*liên quan đến cấu trúc của câu - từ, từ loại của từ, từ đồng nghĩa, phân tích cú pháp, gán nhãn từ loại, thứ tự từ trong câu, nhận dạng thực thể có tên, khái niệm,...*), dịch từ ngôn ngữ này sang ngôn ngữ khác, trích dẫn, bản quyền tác giả, tự sao chép,...

2. Hướng phát triển

- Tiếp tục nghiên cứu các phương pháp xử lý, tìm kiếm, so khớp DNA đạt hiệu quả cao hơn.

- Tổ chức dữ liệu DNA theo mô hình Tensor là một hướng đi đầy triển vọng cần tiếp tục nghiên cứu và thử nghiệm.

- Phát triển và triển khai hệ thống hoàn chỉnh ứng dụng vào thực tiễn để góp phần nâng cao chất lượng đào tạo và nghiên cứu khoa học.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ

1. Hồ Phan Hiếu, Trần Thanh Liêm, *Giải pháp hệ thống hóa tên miền và nguồn tài liệu khoa học của Đại học Đà Nẵng*. Tạp chí Khoa học và Công nghệ ĐHĐN, Số 12(97), 2015, (20-24).

2. Hung Vo Trung, Ngoc Anh Nguyen, Hieu Ho Phan, Thi Dung Dang, *Comparison of the Documents Based On Vector Model: A Case Study of Vietnamese Documents*. American Journal of Engineering Research (AJER), Vol 6(7), 2017, (251-256).

3. Hồ Phan Hiếu, Võ Trung Hùng, Nguyễn Thị Ngọc Anh, *Một số phương pháp tính độ tương tự văn bản dựa trên mô hình vector*. Tạp chí Khoa học và Công nghệ ĐHĐN, Số 11(120), 2017, (112-117).

4. Hồ Phan Hiếu, Nguyễn Thị Ngọc Anh, Nguyễn Văn Hiếu, Đặng Thiên Bình, Võ Trung Hùng, *Một cách tiếp cận mới để phát hiện sự giống nhau của văn bản dựa trên phép biến đổi wavelet rời rạc*. Kỷ yếu Hội nghị Khoa học Công nghệ Quốc gia lần thứ X (Fair'10), lĩnh vực Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin, 2017, (479-487). *DOI: 10.15625/vap.2017.00057*

5. Phan Hieu Ho, Trung Hung Vo, Ngoc Anh Thi Nguyen, *Data Warehouse Designing for Vietnamese Textual Document-based Plagiarism Detection System*. Hội nghị quốc tế IEEE International Conference on System Science and Engineering (ICSSE), 2017, (254-258). *DOI: 10.1109/ICSSE.2017.8030873* ***(Indexed in Scopus)***

6. Nguyen Thi Ngoc Anh, Ho Phan Hieu, Tran Anh Kiet, and Vo Trung Hung, *Similarity Detection for Higher-Order Structure of DNA Sequences*. Journal of Science and Technology: Issue on Information and Communications Technology, Vol. 3, No.2, 2017, (28-34). *DOI: 10.31130/jst.2017.51*

7. Phan Hieu Ho, Ngoc Anh Thi Nguyen, Trung Hung Vo, *DNA Sequences Representation Derived from Discrete Wavelet Transformation for Text Similarity Recognition*. In Springer SCI Book, Modern Approaches for Intelligent Information and Database Systems, 2018, (75-85). *DOI: 10.1007/978-3-319-76081-0_7* ***(Indexed in Scopus)***

8. Hồ Phan Hiếu, Nguyễn Thị Ngọc Anh, Võ Trung Hùng, *Phương pháp mã hóa văn bản thành chuỗi số DNA để đánh giá mức độ giống nhau của văn bản*. Hội thảo Khoa học Quốc gia CNTT và ứng dụng trong các lĩnh vực (CITA2018), (223-229).

9. Phan Hieu Ho, Trung Hung Vo, Ngoc Anh Thi Nguyen, Ha Huy Cuong Nguyen, *A Narrative Method for Evaluating Documents Similarity based on Unique Strings*. International Journal of Recent Technology and Engineering (IJRTE), Vol 8, 2019, (473-479). *DOI: 10.35940/ijrte.B1073.0982S1119* ***(Indexed in Scopus)***

TÀI LIỆU THAM KHẢO

[1] Achananuparp, P., Hu, X., and Shen, X., "The Evaluation of Sentence Similarity Measures", in *International Conference on data warehousing and knowledge discovery*, Springer, 2008, pp. 305-316.

[2] Aggarwal, C. C., "Similarity and Distances", in *Data Mining*, Ed: Springer, Cham, 2015, pp. 63-91.

[3] Alzahrani, S. and Salim, N., "Fuzzy semantic-based string similarity for extrinsic plagiarism detection", in *CLEF 2010 LABs and Workshops, Notebook Papers*, Braschler and Harman, 2010, pp. 1-8.

[4] Androutsopoulos, I. and Malakasiotis, P., "A survey of paraphrasing and textual entailment methods", *Journal of Artificial Intelligence Research*, pp. 135-187, 2010.

[5] Anh, N. H. T., Chi, N. T. K., and Phi, N. H., "Mô hình biểu diễn văn bản thành đồ thị", *Tạp chí Phát triển Khoa học và Công nghệ*, vol. 12, pp. 5-14, 2009.

[6] Apostolico, A. and Giancarlo, R., "The Boyer–Moore–Galil string searching strategies revisited", *SIAM Journal on Computing*, vol. 15, pp. 98-105, 1986.

[7] Armstrong 2nd, J., "Plagiarism: what is it, whom does it offend, and how does one deal with it?", *AJR. American journal of roentgenology*, vol. 161, pp. 479-484, 1993.

[8] Atkins, S., Clear, J., and Ostler, N., "Corpus design criteria", *Literary and linguistic computing*, vol. 7, pp. 1-16, 1992.

[9] Baldi, P. and Brunak, S., *Bioinformatics: the machine learning approach*, MIT press, 2001.

[10] Bao, J.-P., Shen, J.-Y., Liu, X.-D., Liu, H.-Y., and Zhang, X.-D., "Finding plagiarism based on common semantic sequence model", in *Advances in Web-Age Information Management*, Ed: Springer, 2004, pp. 640-645.

[11] Barrón-Cedeño, A. and Rosso, P., "On automatic plagiarism detection based on n-grams comparison", in *Advances in Information Retrieval*, Ed: Springer, 2009, pp. 696-700.

[12] Basile, C., Benedetto, D., Caglioti, E., Cristadoro, G., and Esposti, M., "A plagiarism detection procedure in three steps: Selection, matches and “squares”", in *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, San Sebastian, Spain, 2009, pp. 19-23.

[13] Begum, S., Ahmed, M. U., Funk, P., Xiong, N., and Von Schéele, B., "A case-based decision support system for individual stress diagnosis using fuzzy similarity matching", *Computational Intelligence*, vol. 25, pp. 180-195, 2009.

[14] Bilenko, M. and Mooney, R. J., "Adaptive duplicate detection using learnable string similarity measures", in *Proceedings of the ninth ACM SIGKDD*

international conference on Knowledge discovery and data mining, ACM, 2003, pp. 39-48.

[15] Bin-Habtoor, A. and Zaher, M., "A Survey on Plagiarism Detection Systems", *International Journal of Computer Theory and Engineering*, vol. 4, pp. 185-188, 2012.

[16] Broder, A. Z., "On the resemblance and containment of documents", in *Proceedings of the Compression and Complexity of Sequences 1997*, IEEE, 1997, pp. 21-29.

[17] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C., "Class-based n-gram models of natural language", *Computational linguistics*, vol. 18, pp. 467-479, 1992.

[18] Buckland, M. K., "What is a "document"?", *Journal of the American society for information science*, vol. 48, pp. 804-809, 1997.

[19] Ceska, Z., "Automatic plagiarism detection based on latent semantic analysis", University of West Bohemia, 2009.

[20] Ceska, Z. and Fox, C., "The influence of text pre-processing on plagiarism detection", in *Proceedings of the International Conference RANLP'09*, Borovets, Bulgaria, 2009, pp. 55-59.

[21] Chan, K.-P. and Fu, A. W.-C., "Efficient time series matching by wavelets", in *Proceedings of the 15th International Conference on Data Engineering*, IEEE, 1999, pp. 126-133.

[22] Chaovalit, P., Gangopadhyay, A., Karabatis, G., and Chen, Z., *Discrete Wavelet Transform-Based Time Series Analysis and Mining* vol. 43, 2011.

[23] Chen, C.-Y., Yeh, J.-Y., and Ke, H.-R., "Plagiarism detection using ROUGE and WordNet", *Journal of Computing*, vol. 2, pp. 34-44, 2010.

[24] Chow, T. W. and Rahman, M., "Multilayer SOM with tree-structured data for efficient document retrieval and plagiarism detection", *IEEE Transactions on Neural Networks*, vol. 20, pp. 1385-1402, 2009.

[25] Christopher, D. M., Raghavan, P., and Schutze, H. S., *Introduction to Information Retrieval*, vol. 544, 2009.

[26] Corley, C. and Mihalcea, R., "Measuring the semantic similarity of texts", in *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, Association for Computational Linguistics, 2005, pp. 13-18.

[27] Crochemore, M. and Hancart, C., "Pattern matching in strings", in *Algorithms and Theory of Computation Handbook*, Mikhail, J. A., Ed., Ed: CRC Press, 1998, pp. 11.1-11.28.

[28] Dagan, I., Glickman, O., and Magnini, B., "The PASCAL recognising textual entailment challenge", in *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, Ed: Springer, 2006, pp. 177-190.

[29] Đầu, D. H. and Liệt, Đ. V., "Dùng biến đổi wavelet rời rạc để phân tích các dị thường trọng lực", *Tạp chí Khoa học Trường Đại học Cần Thơ*, vol. 4, pp. 222-229, 2005.

[30] De, T. C., Lam, L. V., Bao, B. V. Q., Hung, N. G., and Tri, T. C., "Developing Plagiarism Detection System for Vietnamese University", in *12th Vietnam – Japan International Joint Symposium, Can Tho*, 2014.

[31] Điền, Đ., *Giáo trình Xử lý Ngôn ngữ tự nhiên*, Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh, 2006.

[32] Dreher, H., "Automatic conceptual analysis for plagiarism detection", *Issues in Informing Science and Information Technology*, vol. 4, pp. 601-628, 2007.

[33] El Kourdi, M., Bensaid, A., and Rachidi, T. E., "Automatic Arabic document categorization based on the Naïve Bayes algorithm", in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Association for Computational Linguistics, 2004, pp. 51-58.

[34] Elhadi, M. and Al-Tobi, A., "Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures", in *Proceedings of the Fourth International Conference on Computer Sciences and Convergence Information Technology (ICCIT'09)*, IEEE, 2009, pp. 679-684.

[35] Elhadi, M. and Al-Tobi, A., "Use of text syntactical structures in detection of document duplicates", in *Proceedings of the Third International Conference on Digital Information Management (ICDIM'08)*, IEEE, 2008, pp. 520-525.

[36] Fullam, K. and Park, J., "Improvements for scalable and accurate plagiarism detection in digital documents", in *Proceedings of the 8th International Conference on Parallel and Distributed Systems*, Kyongju, Korea, 2001, pp. 8-23.

[37] Gipp, B. and Beel, J., "Citation based plagiarism detection: a new approach to identify plagiarized work language independently", in *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, ACM, 2010, pp. 273-274.

[38] Gipp, B. and Meuschke, N., "Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence", in *Proceedings of the 11th ACM symposium on Document engineering*, ACM, 2011, pp. 249-258.

[39] Gomaa, W. H. and Fahmy, A. A., "A survey of text similarity approaches", *International Journal of Computer Applications*, vol. 68, pp. 13-18, 2013.

[40] Gonnet, G. H. and Baeza-Yates, R. A., "An analysis of the Karp-Rabin string matching algorithm", *Information Processing Letters*, vol. 34, pp. 271-274, 1990.

[41] Grozea, C., Gehl, C., and Popescu, M., "ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection", in *Proceedings of the*

SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse, San Sebastian, Spain, 2009, pp. 10-18.

[42] Grozea, C. and Popescu, M., "Who's the thief? automatic detection of the direction of plagiarism", in *Computational Linguistics and Intelligent Text Processing*, Ed: Springer, 2010, pp. 700-710.

[43] Gupta, D. and Choubey, S., "Discrete wavelet transform for image processing", *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, pp. 598-602, 2015.

[44] Hoad, T. C. and Zobel, J., "Methods for identifying versioned and plagiarized documents", *Journal of the American Society for Information Science and Technology*, vol. 54, pp. 203-215, 2003.

[45] Hofmann, T., "Probabilistic latent semantic indexing", in *ACM SIGIR Forum*, ACM, 2017, pp. 211-218.

[46] Huang, A., "Similarity measures for text document clustering", in *Proceedings of the sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008)*, Christchurch, New Zealand, 2008, pp. 49-56.

[47] Hùng, V. T., *Một số phương pháp và mô hình áp dụng trong Xử lý ngôn ngữ tự nhiên*, Nhà xuất bản Thông tin và Truyền thông, 2017.

[48] Hyyrö, H., "A bit-vector algorithm for computing Levenshtein and Damerau edit distances", *Nord. J. Comput.*, vol. 10, pp. 29-39, 2003.

[49] Jadalla, A. and Elnagar, A., "A plagiarism detection system for Arabic text-based documents", in *Pacific-Asia Workshop on Intelligence and Security Informatics*, Springer, 2012, pp. 145-153.

[50] Jilani, T. and Najamuddin, M., *A Review of Adaptive Bayesian Modeling for Time Series Forecasting* vol. 4, 2014.

[51] Kang, N., Gelbukh, A., and Han, S., "PPChecker: Plagiarism pattern checker in document copy detection", in *Proceedings of the International Conference on Text, Speech and Dialogue*, Springer, 2006, pp. 661-667.

[52] Kasprzak, J. and Brandejs, M., "Improving the reliability of the plagiarism detection system", presented at the CLEF 2010 LABs and Workshops, Notebook Papers, 2010.

[53] Khatibsyarbini, M., et al, "A hybrid weight-based and string distances using particle swarm optimization for prioritizing test cases", *Journal of Theoretical and Applied Information Technology*, vol. 95, pp. 2723-2732, 2017.

[54] Knuth, D. E., Morris, J., James H, and Pratt, V. R., "Fast pattern matching in strings", *SIAM Journal on Computing*, vol. 6, pp. 323-350, 1977.

[55] Lai, C.-C. and Tsai, C.-C., "Digital image watermarking using discrete wavelet transform and singular value decomposition", *IEEE Transactions on instrumentation and measurement*, vol. 59, pp. 3060-3063, 2010.

[56] Lane, P. C., Lyon, C., and Malcolm, J. A., "Demonstration of the Ferret plagiarism detector", in *Proceedings of the 2nd International Plagiarism Conference*, 2006.

[57] Lavoie, T. and Merlo, E., "An accurate estimation of the levenshtein distance using metric trees and manhattan distance", in *Proceedings of the 6th International Workshop on Software Clones*, IEEE Press, 2012, pp. 1-7.

[58] Leacock, C., Miller, G. A., and Chodorow, M., "Using corpus statistics and WordNet relations for sense identification", *Computational Linguistics*, vol. 24, pp. 147-165, 1998.

[59] Ledru, Y., Petrenko, A., Boroday, S., and Mandran, N., "Prioritizing test cases with string distances", *Automated Software Engineering*, vol. 19, pp. 65-95, 2012.

[60] Leonardo, B. and Hansun, S., "Text Documents Plagiarism Detection using Rabin-Karp and Jaro-Winkler Distance Algorithms", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, pp. 462-471, 2017.

[61] Lewis, A. S. and Knowles, G., "Image compression using the 2-D wavelet transform", *IEEE Transactions on image Processing*, vol. 1, pp. 244-250, 1992.

[62] Lewis, J., Ossowski, S., Hicks, J., Errami, M., and Garner, H. R., "Text similarity: an alternative way to search MEDLINE", *Bioinformatics*, vol. 22, pp. 2298-2304, 2006.

[63] Li, T., Li, Q., Zhu, S., and Ogihara, M., *A Survey on Wavelet Applications in Data Mining* vol. 4, 2002.

[64] Li, X., Dong, X. L., Lyons, K. B., Meng, W., and Srivastava, D., "Scaling up copy detection", in *Proceeding of the 31st International Conference on Data Engineering (ICDE'15)*, IEEE, 2015, pp. 89-100.

[65] Li, Y., Bandar, Z., McLean, D., and O'shea, J., "A Method for Measuring Sentence Similarity and iIts Application to Conversational Agents", in *FLAIRS Conference*, 2004, pp. 820-825.

[66] Li, Y., McLean, D., Bandar, Z. A., and Crockett, K., "Sentence similarity based on semantic nets and corpus statistics", *IEEE Transactions on Knowledge & Data Engineering*, pp. 1138-1150, 2006.

[67] Li, Y. and Ngom, A., "Classification of clinical gene-sample-time microarray expression data via tensor decomposition methods", in *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, Springer, 2010, pp. 275-286.

[68] Liang, C.-W. and Chen, P.-Y., "DWT based text localization", *International Journal of Applied Science and Engineering*, vol. 2, pp. 105-116, 2004.

[69] Lin, C.-Y., "Rouge: A package for automatic evaluation of summaries", in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Barcelona, Spain, 2004, pp. 74-81.

[70] Lin, D., "An information-theoretic definition of similarity", in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, Madison, Wisconsin, USA, 1998, pp. 296-304.

[71] Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., *et al.*, "Text representation: From vector to tensor", in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, IEEE, 2005, pp. 725-728.

[72] Ljubesic, N., Boras, D., Bakaric, N., and Njavro, J., "Comparing measures of semantic similarity", in *Proceedings of the 30th International Conference on Information Technology Interfaces (ITI2008)*, IEEE, 2008, pp. 675-682.

[73] Lyon, C., Barrett, R., and Malcolm, J., "Plagiarism is easy, but also easy to detect", pp. 57-65, 2006.

[74] M K, V. and K, K., "A Survey on Similarity Measures in Text Mining", *Machine Learning and Applications: An International Journal (MLAIJ)* vol. 3, pp. 19-28, 2016.

[75] Majumder, G., Pakray, P., Gelbukh, A., and Pinto, D., "Semantic textual similarity methods, tools, and applications: A survey", *Computación y Sistemas*, vol. 20, pp. 647-665, 2016.

[76] Mallat, S. G., "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, pp. 674-693, 1989.

[77] Manku, G. S., Jain, A., and Das Sarma, A., "Detecting near-duplicates for web crawling", in *Proceedings of the 16th International Conference on World Wide Web*, ACM, 2007, pp. 141-150.

[78] Matsuo, Y. and Ishizuka, M., "Keyword extraction from a single document using word co-occurrence statistical information", *International Journal on Artificial Intelligence Tools*, vol. 13, pp. 157-169, 2004.

[79] Melamed, I. D., Green, R., and Turian, J. P., "Precision and recall of machine translation", in *Companion Volume of the Proceedings of HLT-NAACL 2003-Short Papers*, San Francisco, CA, USA, Morgan Kaufmann, 2003, pp. 61-63.

[80] Meuschke, N. and Gipp, B., "State-of-the-art in detecting academic plagiarism", *International Journal for Educational Integrity*, vol. 9, pp. 50-71, 2013.

[81] Michailidis, P. D. and Margaritis, K. G., "On-line string matching algorithms: Survey and experimental results", *International Journal of Computer Mathematics*, vol. 76, pp. 411-434, 2001.

[82] Miller, G. and Fellbaum, C., *Wordnet: An electronic lexical database*, MIT Press Cambridge, 1998.

[83] Mozgovoy, M., Kakkonen, T., and Sutinen, E., "Using natural language parsers in plagiarism detection", in *Workshop on Speech and Language Technology in Education (SLaTE'07)*, Farmington, PA, USA, 2007, pp. 77-79.

[84] Nahnsen, T., Uzuner, O., and Katz, B., "Lexical chains and sliding locality windows in content-based text similarity detection", in *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, AIM, 2005, pp. 150-154.

[85] Nguyen, L. T., Toan, N. X., and Dien, D., "Vietnamese plagiarism detection method", in *Proceedings of the Seventh Symposium on Information and Communication Technology*, ACM, 2016, pp. 44-51.

[86] Nguyen, N. A. T., Yang, H.-J., and Kim, S., "HOKF: High Order Kalman Filter for Epilepsy Forecasting Modeling", *Biosystems*, vol. 158, pp. 57-67, 2017.

[87] Niwattanakul, S., Singthongchai, J., Naenudorn, E., and Wanapu, S., "Using of Jaccard coefficient for keywords similarity", in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2013, pp. 380-384.

[88] Okada, I., Saito, M., Oida, Y., Yamato, H., Hiekata, K., and Miura, S., "Development of the Method for the Appropriate Selection of the Successor by Applying Metadata to the Standardization Reports and Members", in *Joint International Semantic Technology Conference*, Springer, 2012, pp. 255-266.

[89] Olkkonen, J. T., *Discrete Wavelet Transforms-Theory and Application*, InTechOpen, 2011.

[90] Osman, A. H., Salim, N., and Abuobieda, A., "Survey of text plagiarism detection", *Computer Engineering and Applications Journal (ComEngApp)*, vol. 1, pp. 37-45, 2012.

[91] Osman, A. H., Salim, N., Binwahlan, M. S., Alteeb, R., and Abuobieda, A., "An improved plagiarism detection scheme based on semantic role labeling", *Applied Soft Computing*, vol. 12, pp. 1493-1502, 2012.

[92] Paliwal, S., Singh, R. S., and Mandoria, H., "A Survey on various text detection and extraction techniques from videos and images", *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)* vol. 6, pp. 1-10, 2016.

[93] Park, B. K. and Song, I. Y., "Toward total business intelligence incorporating structured and unstructured data", in *Proceedings of the 2nd International Workshop on Business intelligencE and the WEB*, ACM, 2011, pp. 12-19.

[94] Park, L. A., Ramamohanarao, K., and Palaniswami, M., "A novel document retrieval method using the discrete wavelet transform", *ACM Transactions on Information Systems (TOIS)*, vol. 23, pp. 267-298, 2005.

[95] Pataki, M., "Distributed similarity and plagiarism search", in *Proceedings of the Automation and Applied Computer Science Workshop*, Budapest, Hungary, 2006, pp. 121-130.

[96] Phan, A. H. and Cichocki, A., "Tensor decompositions for feature extraction and classification of high dimensional datasets", *Nonlinear theory and its applications, IEICE*, vol. 1, pp. 37-68, 2010.

[97] Philip, S., Shola, P., and Ovye, A., "Application of content-based approach in research paper recommendation system for a digital library", *International Journal of Advanced Computer Science and Applications*, vol. 5, 2014.

[98] Phương, N. H. and Sơn, V. M., "Tách ảnh dùng biến đổi Wavelet và phân tích thành phần độc lập", *Tạp chí Phát triển Khoa học và Công nghệ*, vol. 11, pp. 5-16, 2008.

[99] Popivanov, I., "Efficient similarity queries over time series data using wavelets", in *Proceedings of the 18th International Conference on Data Engineering*, San Jose, Calif, USA, 2002, pp. 273-282.

[100] Potthast, M., et al, "Overview of the 1st International Competition on Plagiarism Detection", *In Stein, B., et al (Ed), PAN'09*, pp. 1-9, 2009.

[101] Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., *et al.*, "Overview of the 5th International Competition on Plagiarism Detection", in *CLEF (Working Notes)*, 2013.

[102] Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., and Rosso, P., "Overview of the 2nd International Competition on Plagiarism Detection", *Proceedings of PAN at CLEF*, 2010.

[103] Rahman, M. and Chow, T. W., "Content-based hierarchical document organization using multi-layer hybrid network and tree-structured features", *Expert Systems with Applications*, vol. 37, pp. 2874-2881, 2010.

[104] Rahman, M., Yang, W. P., Chow, T. W., and Wu, S., "A flexible multi-layer self-organizing map for generic processing of tree-structured data", *Pattern Recognition*, vol. 40, pp. 1406-1424, 2007.

[105] Raviraj, P. and Sanavullah, M., "The modified 2D-Haar Wavelet Transformation in image compression", *Middle-East Journal of Scientific Research*, vol. 2, pp. 73-78, 2007.

[106] Reddy, G. S., Rajinikanth, T., and Rao, A. A., "Clustering and Classification of Text Documents Using Improved Similarity Measure", *International Journal of Computer Science and Information Security*, vol. 14, p. 39, 2016.

[107] Řehůřek, R., "Semantic-based plagiarism detection", Masaryk University, 2008.

[108] Ritter, H. and Kohonen, T., "Self-organizing semantic maps", *Biological cybernetics*, vol. 61, pp. 241-254, 1989.

[109] Rubini, P. and Leela, M. S., "A survey on plagiarism detection in text mining", *International Journal of Research in Computer Applications and Robotics*, vol. 1, pp. 117-119, 2013.

[110] Runeson, P., Alexandersson, M., and Nyholm, O., "Detection of duplicate defect reports using natural language processing", in *Proceedings of the 29th*

International Conference on Software Engineering (ICSE'07), IEEE, 2007, pp. 499-510.

[111] Salton, G., *Automatic text processing: The transformation, analysis, and retrieval of information by computer*, 1989.

[112] Salton, G., "Developments in automatic text retrieval", *Science*, vol. 253, pp. 974-980, 1991.

[113] Salton, G. and Buckley, C., "Term-weighting approaches in automatic text retrieval", *Information Processing & Management*, vol. 24, pp. 513-523, 1988.

[114] Salton, G., Wong, A., and Yang, C. S., "A vector space model for automatic indexing", *Communications of the ACM*, vol. 18, pp. 613-620, 1975.

[115] Si, A., Leong, H. V., and Lau, R. W., "Check: a document plagiarism detection system", in *Proceedings of the 1997 ACM Symposium on Applied Computing*, ACM, 1997, pp. 70-77.

[116] Sidorov, G., Gelbukh, A., Gómez-Adorno, H., and Pinto, D., "Soft similarity and soft cosine measure: Similarity of features in vector space model", *Computación y Sistemas*, vol. 18, pp. 491-504, 2014.

[117] Singh Choudhry, M., Kapoor, R., Abhishek, Gupta, A., and Bharat, B., *A survey on different discrete wavelet transforms and thresholding techniques for EEG denoising*, 2016.

[118] Singla, N. and Garg, D., "String matching algorithms and their applicability in various applications", *International Journal of Soft Computing and Engineering*, vol. 1, pp. 218-222, 2012.

[119] Sorokina, D., Gehrke, J., Warner, S., and Ginsparg, P., "Plagiarism detection in arXiv", in *Proceedings of the 6th International Conference on Data Mining (ICDM'06)*, IEEE, 2006, pp. 1070-1075.

[120] Sowa, J. F., "Conceptual graphs for a data base interface", *IBM Journal of Research and Development*, vol. 20, pp. 336-357, 1976.

[121] Stanković, R. S. and Falkowski, B. J., "The Haar wavelet transform: its status and achievements", *Computers & Electrical Engineering*, vol. 29, pp. 25-44, 2003.

[122] Stein, B., Barrón Cedeño, L. A., Eiselt, A., Potthast, M., and Rosso, P., "Overview of the 3rd International Competition on Plagiarism Detection", in *CEUR Workshop Proceedings*, CEUR Workshop Proceedings, 2011.

[123] Stein, B., Rosso, P., Stamatatos, E., Koppel, M., and Agirre, E., "3rd PAN workshop on uncovering plagiarism, authorship and social software misuse", in *25th Annual Conference of the Spanish Society for Natural Language Processing (SEPLN)*, San Sebastian, Spain, 2009, pp. 1-77.

[124] Stein, B., zu Eissen, S. M., and Potthast, M., "Strategies for retrieving plagiarized documents", in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2007, pp. 825-826.

[125] Su, Z., Ahn, B.-R., Eom, K.-Y., Kang, M.-K., Kim, J.-P., and Kim, M.-K., "Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm", in *Proceedings of the 3rd International Conference on Innovative Computing Information and Control (ICICIC'08)*, IEEE, 2008, pp. 569-569.

[126] Tang, X. L., Wang, X. R., and Wang, M., "Text Summarization Using Hybrid Parallel Genetic Algorithm", in *Advanced Materials Research*, Trans Tech Publ, 2011, pp. 1073-1076.

[127] Taufin M Jeeralbhavi, D. J. D. P., Shivananda V. Seeri, "Text Extraction and Localization From Captured Images", *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, vol. 4, pp. 119 -121, 2016.

[128] Tesar, R., Strnad, V., Jezek, K., and Poesio, M., "Extending the single words-based document model: a comparison of bigrams and 2-itemsets", in *Proceedings of the 2006 ACM Symposium on Document Engineering*, ACM, 2006, pp. 138-146.

[129] Thụy, H. Q., *Giáo trình Khai phá Dữ liệu Web*, Giáo dục Việt Nam, 2009.

[130] Toi, N. X., Hung, N. V., and Son, P. B., "A unified plagiarism detection framework", *VNU Journal of Science: Mathematics-Physics*, vol. 27, pp. 55-62, 2011.

[131] Torres, S. and Gelbukh, A., "Comparing similarity measures for original WSD lesk algorithm", *Research in Computing Science*, vol. 43, pp. 155-166, 2009.

[132] Vidakovic, B., *Statistical modeling by wavelets*, John Wiley & Sons, 2009.

[133] Wahlstrom, S., "Evaluation of String Searching Algorithms", in *IDT Mini-conference on Interesting Results in Computer Science and Engineering*, 2004.

[134] Weber-Wulff, D., Möller, C., Touras, J., and Zincke, E., "Plagiarism detection software test 2013", 2013.

[135] Xexéo, G., de Souza, J., Castro, P. F., and Pinheiro, W. A., "Using wavelets to classify documents", in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, 2008, pp. 272-278.

[136] Xiao, C., Wang, W., Lin, X., Yu, J. X., and Wang, G., "Efficient similarity joins for near-duplicate detection", *Proceedings of the ACM International Conference on Management of Data*, pp. 1033-1044, 2011.

[137] Yang, H. and Callan, J., "Near-duplicate detection by instance-level constrained clustering", in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2006, pp. 421-428.

[138] Zhanga, J., Suna, X., and Wangc, J., "Semantic Keyword-based Text Copy Detection Method", *Advanced Science and Technology Letters*, vol. 49, pp. 253-261, 2014.

[139] Zimmermann, H. J., *Fuzzy set theory and its applications*, Kluwer Academic Publishers, Boston/Dordrecht/London, 2001.

[140] Zini, M., Fabbri, M., Moneglia, M., and Panunzi, A., "Plagiarism detection through multilevel text comparison", in *Proceedings of the Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'06)*, IEEE, 2006, pp. 181-185.

[141] Zou, D., Long, W.-J., and Ling, Z., "A cluster-based plagiarism detection method", in *Notebook Papers of CLEF 2010 LABs and Workshops*, 2010.