Trường Đại học Quy Nhơn

Khoa Công Nghệ Thông Tin

BÁO CÁO HỌC PHẦN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Chủ đề: Plagiarism detection (Nhận diện đạo văn)

BÁO CÁO TIỂU LUẬN

Sinh Viên	: Tạ Xuân Kiên
Ngành & Khóa	: Công Nghệ Thông Tin K43
Lớp	: Công Nghệ Thông Tin K43C
Giảng Viên	: TS. Lê Quang Hùng

Mục lục

I. Giới thiệu chung:

Đạo văn là một trong những vấn nạn trong môi trường học thuật. Với sự phát triển nhanh chóng của Internet và các thiết bị Công nghệ thông tin (CNTT), việc đạo văn gần đây đã được thực hiện rất dễ dàng. Người vi phạm có nhiều phương tiện để tìm kiếm và ăn cắp nội dung hay ý tưởng của người khác bởi vì những nghiên cứu và ý tưởng gần như có sẵn rất nhiều trên mạng Internet. Hơn nữa, họ cũng tận dụng kỹ thuật của CNTT để dấu việc đạo văn của họ. Ở Việt Nam, đạo văn là một trong những mối quan tâm đặc biệt trong hầu hết các trường đại học. Mỗi trường đại học có chính sách riêng về đạo văn của mình để ngăn chặn sinh viên đạo luận văn, tài liệu học thuật. Tuy nhiên, đạo văn vẫn còn tồn tại và có chiều hướng gia tăng trong học đường ở Việt Nam

Do đó, với mong muốn có một Ứng dụng phát hiện đạo văn nhanh chóng, rõ ràng, cô đọng và hiệu quả, Ứng dụng phát hiện đạo văn đã được tạo ra để đáp ứng các yêu cầu đó.

II. Phát biểu bài toán:

Phát hiện đạo văn hoặc phát hiện sự giống nhau về nội dung là quá trình xác định các trường hợp đạo văn hoặc vi phạm bản quyền trong một tác phẩm hoặc tài liệu. Việc sử dụng rộng rãi máy tính và sự ra đời của Internet đã khiến việc đạo văn tác phẩm của người khác trở nên dễ dàng hơn.

Theo Meuschke và Gipp (Meuschke and Gipp, 2013), đạo văn là việc sử dụng các ý tưởng của người khác, mà không đưa ra lời xác nhận và tài liệu tham khảo phù hợp. Người phạm tội trình bày ý tưởng hay lời nói của người khác như là của riêng của họ. Meuschke và Gipp nói rằng một số nhà nghiên cứu mô tả đạo văn học văn học như trộm cắp, ăn cắp ý tưởng hay lời nói từ những người khác (Ercegovac and Richardson, 2004; Park, 2003).

Tình trạng đạo văn học trên thế giới đã được thảo luận trong (Gipp, 2014). Nó cho thấy rằng đạo văn xảy ra trên toàn thế giới và trở thành một vấn đề chưa được giải quyết. Một nghiên cứu được tiến hành trên 80.000 sinh viên trong ba năm ở Mỹ và Canada 2002-2005 (McCabe, 2005) cho thấy 38% sinh viên đại học và 25% sinh viên sau đại học đã sao chép hoặc diễn giải các câu văn mà không đưa ra nguồn gốc. Các nghiên cứu khác bên ngoài Mỹ và Canada cũng cho thấy tỷ lệ đạo văn rất cao trong môi trường học tập. Một số Ứng dụng phát hiện đạo văn đã được thực hiện và họ phát hiện 20% hoặc nhiều tài liệu có nội dung đáng ngờ (Barrett and Malcolm, 2006; Culwin, 2006). Dựa trên những số liệu này, Gipp và Bela kết luận rằng đạo văn trong môi trường học thuật là một vấn đề nghiêm trọng.

Ở Việt Nam, đạo văn học đã thực sự được quan tâm trong xã hội. Có rất nhiều cuộc thảo luận, hội thảo, hội nghị tập trung vào đạo văn trong học đường. Tuy nhiên, có rất ít nghiên cứu về đạo văn trong học thuật được xuất bản gần đây. Hầu như tất cả các trường hợp đạo văn được đưa tin trên các tờ báo như Thanh Niên, Tuổi Trẻ,... Những tờ báo này mô tả đạo văn xảy ra khá phổ biến trong cả hai chương trình đại học và sau đại học. Họ đề nghị các trường đại học Việt Nam phải chống đạo văn nghiêm ngặt, nghiêm túc hơn. Hơn nữa, ứng dụng CNTT để phát hiện đạo văn cũng được đề cập đến như một trong những cách thức hiệu quả để giảm đạo văn. Các trường đại học có thể xây dựng một số Ứng dụng phát hiện đạo

văn để giúp cả sinh viên và giảng viên kiểm tra đạo văn.

Vì các lý do đã nêu ở trên, việc giải quyết bài toán phát hiện đạo văn là cần thiết và có ích trong nhiều lĩnh vực, đặc biệt là trong lĩnh vực nghiên cứu và giáo dục. Nếu ta có thể phát hiện và ngăn chặn được hành vi sao chép nội dung của người khác một cách hiệu quả, chúng ta có thể đảm bảo rằng các kết quả nghiên cứu được công bố là chính xác và có tính khả thi, đồng thời đảm bảo tính toàn vẹn và uy tín của các tác giả và tổ chức nghiên cứu. Ngoài ra, việc giải quyết bài toán này cũng đóng góp vào việc phát triển và cải tiến các công nghệ xử lý ngôn ngữ tự nhiên.

Vấn đề: Cho một tập dữ liệu tài liệu đáng ngờ Dq và một tập hợp nguồn lớn D, tìm tất cả các phần đáng ngờ sq từ dq:dqϵDq mà tương tự với các phần sx từ dx:dxϵDx dựa trên phương pháp tương tự dựa trên ý nghĩa mờ như sẽ được mô tả trong phần 3.
Yêu cầu ngưỡng: Đầu tiên, trích xuất một tập hợp các đặc trưng cho mỗi dqϵDq và dϵD. Thứ hai, tìm một danh sách các tài liệu tiềm năng nhất Dx trong đó Dx$\subset$D dựa trên shingling và hệ số tương đồng Jaccard được biết đến trong IR. Thứ ba, thực hiện phân tích sâu hơn từng câu sử dụng phương pháp dựa trên ý nghĩa mờ. Cuối cùng, thực hiện các hoạt động xử lý sau cùng để kết hợp các tuyên bố tương tự thành đoạn văn hoặc đoạn.
Giới hạn: Thuật toán này không xử lý việc phát hiện đạo văn nội tại (tức là các biến thể về phong cách viết) và đạo văn giữa các ngôn ngữ khác nhau. Đó là, ngôn ngữ của cả tài liệu đáng ngờ và tài liệu ứng cử viên được coi là đồng nhất.

III. Các Phương pháp:

1. Đo độ đồng dạng văn bản (Text similarity measurement):

Phát hiện đạo văn thường dựa trên việc so sánh hai hoặc nhiều tài liệu văn bản. Để so sánh hai hoặc nhiều tài liệu và suy ra mức độ tương đồng giữa chúng, cần phải gán giá trị số, gọi là điểm tương đồng cho mỗi tài liệu. Điểm này có thể được dựa trên các chỉ số khác nhau. Có nhiều tham số và khía cạnh trong tài liệu có thể được sử dụng làm chỉ số. Trong bài viết này, chúng tôi không chú ý đến các chỉ số cụ thể được sử dụng cho phát hiện đạo văn trong mã nguồn, chẳng hạn như chỉ số Halstead . Các chỉ số chung được sử dụng phổ biến nhất được miêu tả trong phần này.

Lancaster và Culwin trong công trình của họ đã cố gắng phân loại các chỉ số được sử dụng cho phát hiện đạo văn. Họ đã đề xuất hai cách phân loại chỉ số. Phân loại đầu tiên dựa trên số lượng tài liệu tham gia vào quá trình tính toán chỉ số, và phân loại thứ hai dựa trên độ phức tạp tính toán của các phương pháp được sử dụng để tìm sự tương đồng. Trong phân loại đầu tiên, chỉ số có thể được phân loại thành chỉ số đơn hoặc chỉ số ghép đôi, và thành chỉ số văn cảnh hoặc đa chiều, tùy thuộc vào số lượng tài liệu được xử lý và tùy thuộc vào tập tài liệu tham gia xử lý. Chỉ số văn cảnh hoạt động trên toàn bộ tập tài liệu. Chỉ số đa chiều hoạt động trên một số tài liệu được lựa chọn. Trong phân loại thứ hai, chỉ số có thể được phân loại thành chỉ số bề ngoài và chỉ số cấu trúc. Chỉ số bề ngoài là đo lường sự tương đồng có thể đánh giá đơn giản bằng cách xem một hoặc nhiều tài liệu. Trong trường hợp này, không cần kiến thức về các đặc trưng ngôn ngữ tự nhiên. Chỉ số cấu trúc là đo lường sự tương đồng yêu cầu kiến thức về cấu trúc của một hoặc nhiều tài liệu.

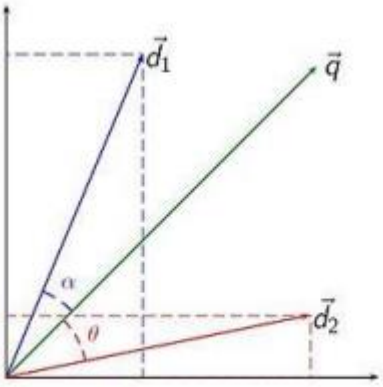
Cách phân loại khác là dựa trên nguyên tắc chính được tích hợp trong chỉ số, tức là phân tích nội dung của các tài liệu dựa trên phương pháp ngữ nghĩa hoặc

thống kê. Trong phương pháp thống kê, không cần phải hiểu ý nghĩa của tài liệu. Một phương pháp thống kê phổ biến là xây dựng các vectơ tài liệu dựa trên các giá trị mô tả tài liệu, chẳng hạn như tần số của các từ, chỉ số nén [16], cặp từ Lancaster và các chỉ số khác. Các chỉ số thống kê có thể độc lập với ngôn ngữ hoặc phụ thuộc vào ngôn ngữ . Phương pháp thống kê thuần túy là phương pháp N-gram, trong đó văn bản được mô tả với các chuỗi N ký tự liên tiếp . Dựa trên các đo lường thống kê, mỗi tài liệu có thể được mô tả bằng vân tay gọi là "fingerprint", trong đó các N-gram được băm và sau đó chọn một số để làm vân tay . Có thể có các đo lường cũng chứa xác suất. Những đo lường này là đo lường lý thuyết thông tin , BM25 và đo lường mô hình ngôn ngữ .

Trong nhiều trường hợp, điểm tương đồng giữa hai tài liệu được tính toán như khoảng cách Euclidean giữa các vectơ tài liệu. Sự tương đồng của các tài liệu giống nhau là không . Tương đồng cũng có thể được tính toán như tích vô hướng của các vectơ tài liệu chia cho độ dài của chúng. Điều này tương đương với cosin của góc giữa hai vectơ tài liệu được nhìn từ điểm gốc . Trong nhiều trường hợp, các vectơ tài liệu được tạo thành từ tần số và trọng số của từ, được tính tự động cho mỗi tài liệu. Tần số của từ được tính toán dựa trên hàm tỷ lệ . Công thức cosin cũng có thể có biến thể (xem, Công thức 1), trong đó cũng tính đến trọng số của từ :

$$S_{cos}(A,B) = \frac{\sum_{i=1}^{n}[\alpha_i^2 \times F_i(A) \times F_i(B)]}{\sqrt{\sum_{i=1}^{n}[\alpha_i^2 \times F_i^2(A)] \times \sum_{i=1}^{n}[\alpha_i^2 \times F_i^2(B)]}} \quad (1)$$

trong đó αi là vector trọng số từ; Fi(A), Fi(B) - tần suất của từ thứ i trong tài liệu A và B, tương ứng. Hàm cosin, hàm tỷ lệ, cũng như tích vô hướng, đo lường Jaccard, đo lường Dice, đo lường trùng lắp [21] là các đo lường tương đồng đối xứng . Các đo lường tương đồng đối xứng hoặc không đối xứng là một phân loại khác. Các đo lường tương đồng không đối xứng là vector tần số nặng và mô hình tỷ lệ bao gồm, được dẫn xuất từ hàm cosin và hàm tỷ lệ bằng cách kết hợp khái niệm tương đồng không đối xứng với vector tần số nặng . Các đo lường tương đồng không đối xứng có thể được sử dụng để tìm kiếm các tập con. Thông thường trong các công cụ khác nhau, các phương pháp thống kê được triển khai do tính đơn giản của chúng.



Hình 1. *Ví dụ về góc tạo bởi hai vec-tơ $\vec{d}_1$, $\vec{d}_2$ với $\vec{q}$*

2. Phân loại văn bản:

Phương pháp phân loại văn bản là một trong những phương pháp phát hiện gian lận bản sao phổ biến nhất. Để phân loại một tài liệu như gốc hoặc bản sao, phương pháp này sẽ so sánh văn bản đó với các tài liệu khác để xác định sự tương đồng. Sau đó, nó sẽ đưa ra quyết định về xem văn bản đó có phải là bản sao hay không.

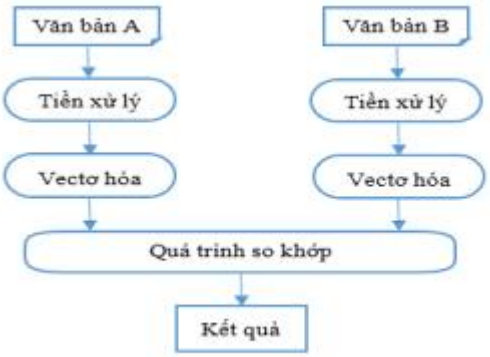
Các phương pháp phân loại văn bản thường sử dụng các thuật toán học máy để

học từ các tập dữ liệu huấn luyện, sau đó áp dụng các kỹ thuật phân loại để đưa ra kết quả. Các thuật toán phổ biến bao gồm Naive Bayes, Support Vector Machines (SVM), Decision Trees và Neural Networks.
Để phát hiện gian lận bản sao, phương pháp này sẽ so sánh văn bản của tài liệu được kiểm tra với các tài liệu khác để tìm kiếm những phần giống nhau. Điều này có thể được thực hiện bằng cách sử dụng các phương pháp như so sánh đối chiếu chuỗi, so sánh độ dài và phân tích ngôn ngữ tự nhiên.

Phương pháp phân loại văn bản là một trong những phương pháp phổ biến nhất trong việc phát hiện gian lận bản sao. Theo bài báo "Plagiarism Detection Techniques: A Review" của tác giả Arvind Dhiman, phương pháp này sử dụng các thuật toán học máy để học từ các tập dữ liệu huấn luyện, sau đó áp dụng các kỹ thuật phân loại để đưa ra kết quả. Các thuật toán phổ biến bao gồm Naive Bayes, Support Vector Machines (SVM), Decision Trees và Neural Networks.
Các kỹ thuật phân loại văn bản thường sử dụng các đặc trưng của văn bản, như tần suất xuất hiện của các từ hoặc các ký tự đặc biệt, để đưa ra quyết định. Ngoài ra, các kỹ thuật phân loại văn bản còn có thể sử dụng các phương pháp khác như phân tích cú pháp hoặc phân tích ngữ nghĩa để cải thiện độ chính xác.
Tuy nhiên, phương pháp phân loại văn bản cũng có những hạn chế. Theo bài báo "Text Classification: A Comprehensive Survey" của tác giả Fabrizio Sebastiani, các kỹ thuật phân loại văn bản có thể dẫn đến các lỗi phân loại do việc sử dụng các đặc trưng văn bản cụ thể quá nhiều hoặc quá ít. Ngoài ra, các kỹ thuật phân loại văn bản có thể bị ảnh hưởng bởi độ dài của văn bản hoặc các ký tự đặc biệt, dẫn đến việc cho ra các kết quả không chính xác.

IV. Ứng dụng nhận dạng đạo văn:

1. Phương pháp sử dụng:
   Phương pháp đo độ đồng dạng văn bản (Text similarity measurement):



Hình 2. *Mô hình so sánh hai văn bản*

2. Cách bước thực hiện:

a. Tiền xử lý dữ liệu: Loại bỏ các ký tự đặc biệt, các từ ngữ không cần thiết (stopwords), các dấu câu, chuyển tất cả thành chữ thường, ... để tạo ra một bản sao gọn hơn của văn bản.

```
def get_text_preprocessed(strings):
    strings = strings.lower()
    strings =  re.sub(r'[^\w\s]', '', strings).strip()
    return strings
```

b. Tính toán độ tương đồng: Sử dụng một số phương pháp tính toán độ tương đồng như Cosine similarity, Jaccard similarity, Euclidean distance, và Levenshtein distance để đo độ tương đồng giữa hai văn bản.

```
def get_cosine_similarity(sentence1, sentence2):
    corpus = set(sentence1).union(sentence2)
    numerator = sum(sentence1.get(k, 0) * sentence2.get(k, 0) for k in corpus)
    magnitude_sentence1 = math.sqrt(sum(sentence1.get(k, 0)**2 for k in corpus))
    magnitude_sentence2 = math.sqrt(sum(sentence2.get(k, 0)**2 for k in corpus))
    return numerator / (magnitude_sentence1 * magnitude_sentence2)
```

c. Xác định ngưỡng: Ngưỡng thường được sử dụng để quyết định liệu hai văn bản có được coi là "giống nhau" hay "không giống nhau" dựa trên mức độ tương tự được tính toán. Nếu mức độ tương tự vượt qua ngưỡng được xác định, thì văn bản được coi là "giống nhau", nếu không, thì được coi là "không giống nhau". Ngưỡng thường được thiết lập bằng cách kiểm tra các tập huấn luyện và thực nghiệm.

```
def get_threshold(file_1, file_2, num_buckets=1000000):
    strings1, strings2  = [], []
    with open(file_1, 'r', encoding="utf8") as f:
        strings1 = f.readlines()
    strings1 = ''.join(strings1)

    with open(file_2, 'r', encoding="utf8") as f:
        strings2 = f.readlines()
    strings2 = ''.join(strings2)

    strings1 = strings1.lower().split()
    strings2 = strings2.lower().split()

    sentence1 = {}
    sentence2 = {}
    for word in set(strings1 + strings2):
        sentence1[word] = strings1.count(word)
        sentence2[word] = strings2.count(word)

    similarities = []
    for i in range(num_buckets):
        threshold = (i + 1) / num_buckets
        similarity = get_cosine_similarity(sentence1, sentence2)
        similarities.append(similarity)
        if similarity < threshold:
            break

    return threshold
```

d. Xử lý kết quả: Tùy vào phương pháp đo độ đồng dạng văn bản được sử dụng, kết quả sẽ được đưa ra dưới dạng một giá trị số hoặc một ma trận độ tương đồng. Sau đó, ta có thể so sánh kết quả đó với một ngưỡng độ đồng dạng cố định để quyết định liệu hai văn bản có giống nhau hay không.

```python
def check_for_plagiarism(file_1: str, file_2: str, th:int) -> str:
    strings1, strings2  = [], []
    with open(file_1, 'r', encoding="utf8") as f:
        strings1 = f.readlines()
    strings1 = ''.join(strings1)

    with open(file_2, 'r', encoding="utf8") as f:
        strings2 = f.readlines()
    strings2 = ''.join(strings2)
    strings1 = get_text_preprocessed(strings1).split()
    strings2 = get_text_preprocessed(strings2).split()
    sentence1 = Counter(strings1)
    sentence2 = Counter(strings2)
    similarity = (get_cosine_similarity(sentence1,sentence2))
    return classify_plagiarism(similarity,th), similarity
```

3. Thử nghiệm:

   Cấu hình thử nghiệm:

   - CPU: i7 10750h
   - GPU: gtx 1650ti max-q
   - RAM : 16 GB
   - OS: Windown 10 pro

   Input : 2 đoạn văn bản :

   Text_1 : The issue of plagiarism has been strongly concerned in the past 10 years. This attention is paid so as to show respect to the brainpower of authors whose articles are referenced through books, newspapers, the Internet. These works, articles, scientific articles inherit a lot of data from previous articles. Authors may accidentally or intentionally completely copy a certain sentence belonged to the previous authors. Therefore, the command of justifying plagiarism is rising from scientific reports of students' works, master's and doctoral essays. To solve this problem, we might trace every sentence in an article with references or apply anti-plagiarism software which is considered to be more advanced. To clarify the use of anti-plagiarism software, in this article we will study anti-plagiarism programs that are being used online today so that we can find out how they work and compare the features of each. The article will aim at specific comparisons in many aspects such as: features, operation, interface, usefulness of the program. We will know the advantages and disadvantage end of the articles so as to have accurate suggestions for the need of testing plagiarism from writings

   Text_2 : The issue of plagiarism has been strongly concerned in the past 10 years. This attention is paid so as to show respect to the brainpower of authors whose articles are referenced through books, newspapers, the Internet. These works, articles, scientific articles inherit a lot of data from previous articles. Authors may accidentally or intentionally completely copy a certain sentence belonged to the previous authors. Therefore, the command of justifying plagiarism is rising from scientific reports of students' works, master's and doctoral essays. To solve this problem, we might trace every sentence in an

article with references or apply anti-plagiarism software which is considered to be more advanced. To clarify the use of anti-plagiarism software, in this article we will study anti-plagiarism programs that are being used online today so that we can find out how they work and compare the features of each. The article will aim at specific comparisons in many aspects such as: features, operation, interface, usefulness of the program. We will know the advantages and disadvantage end

4. Kết quả:

Ứng dụng hiển thị lên ngưỡng đạo văn của 2 văn bản và 2 văn đạo văn(tương đồng hay không)

Threshold: 0.98715

1(Có sự tương đồng)

```
PS D:\AI\xulinntunhien\Plagiarism-Detection-Using-Cosin-Similarity> python id_detector.py 1.txt 2.txt
Threshold 0.98715
1
```

V. Kết luận:

Trong thời đại công nghệ thông tin, việc sao chép bài viết đã trở thành một vấn đề nghiêm trọng. Trong bài báo này, các phương pháp để giảm thiểu vấn đề sao chép bài viết được thảo luận. Các phương pháp ngăn chặn sao chép bài viết dựa trên sự thay đổi thái độ của xã hội đối với vấn đề này không thể phủ nhận là biện pháp quan trọng nhất để đối phó với việc sao chép bài viết, nhưng việc triển khai những phương pháp này là một thách thức đối với toàn xã hội. Các cơ sở giáo dục cần tập trung vào các phương pháp phát hiện sao chép bài viết. Phân tích các phương pháp phát hiện sao chép bài viết phổ biến cho thấy thường sử dụng các thước đo thống kê khác nhau do tính đơn giản và dễ dàng triển khai trong các công cụ. Phân tích các công cụ phát hiện sao chép bài viết đã biết cho thấy rằng, mặc dù các công cụ này cung cấp dịch vụ tuyệt vời trong việc phát hiện văn bản trùng khớp giữa các tài liệu, thậm chí các phần mềm phát hiện sao chép bài viết tiên tiến cũng không thể phát hiện sao chép bài viết tốt hơn con người. Chúng có một số điểm hạn chế, vì vậy kiểm tra thủ công và nhận định của con người vẫn cần thiết. Não bộ con người là công cụ phát hiện sao chép đa năng, có thể phân tích tài liệu bằng các phương pháp thống kê và ngữ nghĩa, có thể hoạt động với thông tin văn bản và không phải văn bản. Hiện tại, các khả năng này chưa có sẵn cho các công cụ phát hiện sao chép bài viết. Theo [19] "... ít nhất là hiện tại - không có gì có thể hoàn toàn thay thế được sự chú ý của con người". Tuy nhiên, các công cụ phát hiện sao chép bài viết dựa trên máy tính có thể giúp đỡ đáng kể trong việc tìm các tài liệu bị sao chép.

VI. Tham khảo:

- Dhiman, A. (2018). Plagiarism Detection Techniques: A Review. International Journal of Computer Applications, 181(23), 28-31.
- Culwin, Fintan; Lancaster, Thomas (2001). "Plagiarism, prevention, deterrence and detection". CiteSeerX 10.1.1.107.178. Archived from the original on 18 April 2021. Retrieved 11 November 2022 – via The Higher Education Academy.
- Jump up to:a b Bretag, T., & Mahmud, S. (2009). A model for determining student

plagiarism: Electronic detection and academic judgement. Journal of University Teaching & Learning Practice, 6(1). Retrieved from http://ro.uow.edu.au/jutlp/vol6/iss1/6

- Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.
- Janis Grundspenkis, Vita Šakele(2007). Computer-based plagiarism detection methods and tools: An overview
- Salha Alzahrani, Naomie Salim (2010). Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection.
- Trần Cao Đệ, Lê Văn Lâm, Bùi Võ Quốc Bảo, Nguyễn Gia Hưng và Trần Cao Trị(2014). PHÁT TRIỂN ỨNG DỤNG PHÁT HIỆN ĐẠO VĂN CHO TRƯỜNG ĐẠI HỌC VIỆT NAM