Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.00094

ỨNG DỤNG GIẢI THUẬT SINGULAR VALUE DECOMPOSITION TRÊN NỀN HỆ THỐNG PHÂN TÁN VÀO BÀI TOÁN PHÁT HIỆN SAO CHÉP

Nguyễn Võ Thông Thái [1], Bùi Võ Quốc Bảo[2], Huỳnh Phụng Toàn[2], Trần Cao Đệ[2]

[1] Trung tâm Công nghệ phần mềm, Đại học Cần Thơ
[2] Khoa Công nghệ thông tin & Truyền thông, Đại học Cần Thơ
nvtthai@ctu.edu.vn, bvqbao@ctu.edu.vn, hptoan@ctu.edu.vn, tcde@ctu.edu.vn

TÓM TẮT— *Ngày nay, có rất nhiều tài liệu văn bản có thể truy xuất được dễ dàng dưới dạng tài liệu kỹ thuật số và vì vậy người ta có thể truy cập và sao chép dễ dàng. Vấn đề đạo văn nói chung và sao chép luận văn, đồ án nói riêng có thể nói là những mặt tiêu cực phổ biến hiện nay cần được phát hiện và ngăn chặn. Các phương phát phát hiện sao chép tài liệu mới nhất được công bố trong các Hội thảo PAN Workshop vừa qua thường dựa trên lập chỉ mục nghịch đảo cho cụm 4 từ (4-gram). Việc xác định một tập hợp con các tài liệu tiềm năng (có thể bị sao chép) dựa trên ngưỡng số 4-gram chung cho thấy một số hạn chế như tập tiềm năng có thể rất lớn và không thể xếp độ ưu tiên theo số lượng 4-gram chung nên có thể dẫn đến việc tìm kiếm sao chép lâu. Trong bài báo này, chúng tôi đề xuất một phương pháp phát hiện ra tập tiềm năng có sử dụng thuật toán tách giá trị đơn theo mô hình lập trình song song. Các cài đặt và thử nghiệm của chúng tôi đã cho thấy có thể áp dụng phương pháp để phát hiện ra các tập tiềm năng bị sao chép và sắp xếp (ranking) chúng, từ đó có thể hạn chế số lượng tập tin cần phân tích, so sánh để phát hiện ra các đoạn bị sao chép. Đồng thời phương pháp được đề xuất cũng có thể song song hóa để chạy trên một cụm máy tính, nhờ đó có thể áp dụng trên các tập dữ liệu có dung lượng lớn như là một thư viện điện tử thực thụ.*

Từ khóa— *Đạo văn, tách giá trị đơn, xử lý phân tán, tính toán song song.*

I. GIỚI THIỆU

Vấn đề (hay vấn nạn) sao chép tài liệu (đạo văn) ngày nay đang là một vấn đề nghiêm trọng trong môi trường giáo dục, Với sự phát triển mạnh mẽ của công nghệ thông tin và các kỹ thuật lưu trữ, tìm kiếm như Google, Bing, … thì việc sao chép sẽ được thực hiện một các dễ dàng hơn. Sự sao chép ngày càng phổ biến ở mọi cấp độ từ đồ án, tiểu luận, luận văn tốt nghiệp đại học cho đến luận văn tiến sĩ. Nhiều sao chép khác như giáo trình, bài giảng cũng còn khá phổ biến. Có rất nhiều bài viết trên các báo có uy tín công khai tình trạng sao chép bừa bãi luận văn[1, 2].

Ngày nay, đã có nhiều phần mềm hỗ trợ cho việc phát hiện đạo văn. Đa phần là các phần mềm thực hiện kiểm tra sao chép một tài liệu từ "kho tài liệu" trên internet, tức là kiểm tra với tài liệu nguồn từ internet. Các phần mềm này có ưu điểm là kiểm tra với một nguồn hết sức phong phú. Tuy vậy, ở nước ta không có nhiều phần mềm được biết rõ hỗ trợ kiểm tra trên một CSDL đóng của một tổ chức, ví dụ thư viện của một trường hay kho luận văn của một trường.

Trong bài báo này chúng tôi xây dựng một ứng dụng cho phép kiểm tra phát hiện sao chép trên một CSDL đóng của một tổ chức và bài báo này cũng tiếp cận theo các giải pháp được trình bày tại hội thảo PAN Workshop [18] và đề xuất thêm các cải tiến với sự tích hợp giải thuật tách giá trị đơn để có thể rút ngắn thời gian phát hiện sao chép bằng cách hạn chế số lượng tập tin tiềm năng và sự song song hóa.

II. PHÁT HIỆN SAO CHÉP THEO GIẢI PHÁP PAN

1. Phát hiện sao chép theo PAN Workshop

Theo Meuschke và Gipp [1], sao chép là việc sử dụng các suy nghĩ, ý tưởng, hoặc phát biểu của người khác [2, 3] và trình bày như là tác phẩm gốc của chính mình mà không chú thích, trích dẫn phù hợp.

Ta định nghĩa một trường hợp sao chép $s = \{s_{sc}, d_{sc}, s_{ng}, d_{ng}\}$, trong đó một đoạn văn bản s_{sc} trong tài liệu d_{sc} là sao chép từ một đoạn văn bản s_{ng} trong tài liệu d_{ng}. Với một tài liệu d_{sc} cho trước, nhiệm vụ của một hệ thống phát hiện sao chép là phát hiện ra s bằng cách chỉ ra một trường hợp sao chép $r = \{r_{sc}, d_{sc}, r_{ng}, d'_{ng}\}$, bao gồm một đoạn văn bản được cho là sao chép r_{sc} trong tài liệu d_{sc} và đoạn văn bản nguồn của nó r_{ng} trong d'_{ng} xấp xỉ nhất có thể với s. Chúng ta kết luận rằng r phát hiện được s nếu và chỉ nếu $s_{sc} \cap r_{sc} \neq \emptyset$, $s_{ng} \cap r_{ng} \neq \emptyset$ và $d_{ng} = d'_{ng}$.

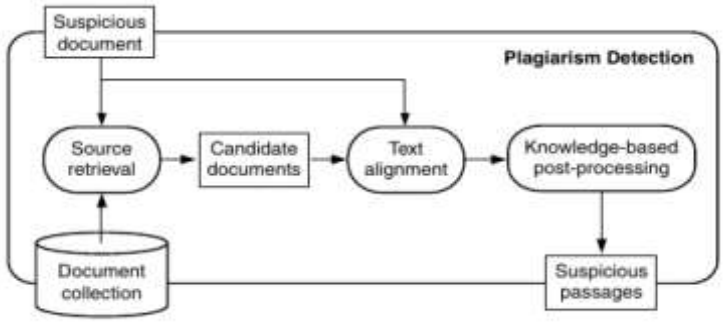
Meuschke and Gipp [1] đã phân loại hệ thống phát hiện đạo văn theo hai hướng sau : một là so sánh độ tương tự của các đoạn văn bản, hai là so sánh độ tương tự của cả văn bản. Trong bài báo này chúng tôi sẽ tập trung vào phương pháp thứ nhất để kiểm tra việc sao chép. Với phương pháp này ta phát hiện ra s bằng cách tìm kiếm tài liệu d'_{ng} từ một tập tài liệu D nào đó (ví dụ kho luận văn số hóa của một trường) và trích xuất ra r_{ng} và r_{sc} từ hai tài liệu d'_{ng} và d_{sc} dựa trên việc so sánh chi tiết giữa hai tài liệu.

[1] http://dantri.com.vn/giao-duc-khuyen-hoc/loan-sao-chep-trong-truong-dh-436609.html
[2] http://vietnamnet.vn/vn/giao-duc/143621/tien-si-dao-van-bi-thu-hoi-bang.html

2. Giải pháp bài toán sao chép theo PAN

Hình 1 chỉ ra các bước xử lý chính trong các hệ thống phát hiện sao chép [4, 5]. Một cách tổng quát, một hệ thống phát hiện sao chép thông thường sẽ được cài đặt trên ba bước xử lý cơ bản.

- Thứ nhất, từ một tài liệu kiểm tra d và kho dữ liệu các tài liệu nguồn D, hệ thống sẽ tìm ra một tập tài liệu tiềm năng $D_d \subset D$ được xác định sao cho D_d là nhỏ nhất có thể nhưng chứa nhiều nhất các tài liệu nguồn chính xác mà d sao chép.
- Thứ hai, phân tích so sánh giữa d với mỗi t ⋃ D_d, để tìm các đoạn bị "sao chép".
- Thứ ba, các cặp đoạn văn bản đã phát hiện được lọc lại dựa trên các quy tắc nào đó và có thể được biểu diễn trực quan cho người dùng. Ví dụ cho bước xử lý này gồm có loại bỏ các phát hiện quá ngắn, chồng chéo hoặc gộp các phát hiện liền kề thành một phát hiện duy nhất,...



Hình 1. Các bước chính trong quá trình phát hiện sao chép

Qua việc tìm hiểu các giải pháp được đề xuất tại PAN Workshop, chúng tôi thấy rằng giải pháp được đề xuất tại PAN Workshop năm 2010 [6] có thể làm mô hình tham khảo cơ sở cho nghiên cứu. Các bước chính của giải pháp

- Tiền xử lý văn bản:
  - Các tập tin văn bản được tách từ đơn, loại bỏ stopword.
  - Các tài liệu nguồn được phân tích và lưu trữ dưới dạng chỉ mục đảo ngược. Cách khá phổ biến trong hội thảo PAN là tách thành cụm 4-gram và lập chỉ mục trên các 4-gram.
- Tìm kiếm các tài liệu nguồn tiềm năng:
  - Vì số lượng tập tài liệu nguồn thường là rất lớn nên trước hết phải có một giải thuật nào đó "lọc" để giới hạn việc so sánh phát hiện sao chép chỉ trên một tập nhỏ các tài liệu tiềm năng. Cách thức lọc trong các giải pháp đưa ra trong PAN là "có ít nhất 20 4-gram chung". Các tập tin trong tài liệu nguồn có từ 20 4-gram chung với tài liệu kiểm tra được coi là "tiềm năng" và được giữ lại để thực hiện việc phân tích so sánh kỹ hơn.
  - Số lượng tài liệu nguồn tiềm năng cho mỗi tài liệu kiểm tra có thể giới hạn (ví dụ 100 tài liệu chẳng hạn) bằng cách sắp xếp giảm dần theo số lượng từ 4-gram chung và chọn từ cao xuống thấp. Con số 20 4-gram chung là một con số mang tính thực nghiệm.
- So sánh chi tiết các cặp tài liệu: Đối với mỗi tài liệu kiểm tra, sau khi tìm được một tập tài liệu nguồn tiềm năng, tiến hành so sánh chi tiết giữa các cặp tài liệu để xác định các đoạn văn bản giống nhau.
- Tinh lọc kết quả: Các đoạn văn bản hợp lệ được xem như các đoạn văn bản sao chép. Bước cuối cùng bao gồm việc loại bỏ các phát hiện chồng chép nhau sau đó biểu diễn cho người dùng.

III. MÔ HÌNH PHÁT HIỆN SAO CHÉP TÍCH HỢP GIẢI THUẬT SVD

1. Mô hình đề xuất

Mặc dù giải pháp [6] của PAN Workshop được đánh giá cao nhưng vẫn còn tồn tại một số vấn đề cần được giải quyết. Theo giải pháp của PAN đã được trình bày phần trên, nhược điểm của giải pháp này xảy ra tại giai đoạn tìm ra tập tài liệu tiềm năng.

- Thứ nhất con số 20 4-gram chung do [6] đề xuất hay tổng quát hơn là n 4-gram chung đó chỉ là dựa vào kinh nghiệm, có thể không có hiệu quả trên nhiều trường hợp, nhất là trong các thư viện đóng với chủ đề gần nhau, chẳng hạn như kho luận văn ngành công nghệ thông tin.
- Sau khi tìm ra tập tài liệu có số n 4-gram chung. Nếu tập này lớn thì làm sao ưu tiên xét các tập tiềm năng nhất? Nếu sắp xếp (ranking) giảm dần theo số 4-gram chung và ấn định một số lượng giới hạn tập tiềm năng thì cũng không có cơ sở, ví dụ sắp xếp giảm dần theo con số 4-gram chung rồi lấy 100 tài liệu đầu tiên làm tập tiềm năng. Rõ ràng không phải cứ nhiều 4-gram chung hơn thì có khả năng bị sao chép cao hơn.

Xuất phát từ những cơ sở trên trong bài báo này chúng tôi để ra giải pháp mới cải tiến cho giai đoạn tìm tập tài liệu tiềm năng này.

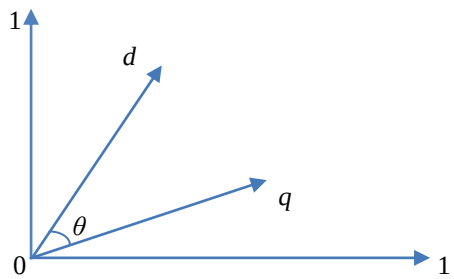
- Đề xuất sử dụng việc phân tích Singular Value Decomposition (SVD) [7] kết hợp với không gian vector để áp dụng cho giai đoạn tìm tập tài liệu tiềm năng.
    - Sử dụng mô hình không gian vector. Mỗi tài liệu trong tập tài liệu nguồn sẽ được mô hình hóa thành một vector đặc trưng. Và mỗi tài liệu kiểm tra sau khi qua bước tiền xử lý tách từ loại bỏ các stop-word lập thành ma trận từ - tài liệu. Ma trận này có số chiều khá lớn do đó sẽ áp dụng giải thuật SVD được áp dụng để làm giảm số chiều, loại bỏ những giá trị nhiễu, giữ lại những giá trị đặc trưng nhất và làm tăng hiệu quả.
    - Tiếp theo các tài liệu sẽ được đo độ tương đồng theo độ đo cosin và đó là cơ sở để trích lọc ra tập tài liệu tiềm năng. Nói cách khác các tài liệu trong thư viện sẽ được tính 1 độ tương đồng (độ đo cosin) với tài liệu kiểm tra và dựa theo độ tương đồng đó sẽ sắp xếp (ranking) cũng như ấn định ngưỡng xem xét theo độ tương đồng chứ không theo số lượng tập tin. Điều này sẽ tự nhiên hơn, nếu tập tin kiểm tra bị sao chép từ nhiều tập thì sẽ có nhiều tập tiềm năng, nếu không bị sao chép gì cả thì số lượng tập tiềm năng nhỏ hoặc có thể là 0.
    - Các bước tiếp theo để phân tích các tài liệu tiềm năng và phát hiện sao chép vẫn như giải pháp của PAN ở trên
- Vấn đề phát sinh việc tính toán SVD có thể lâu do ma trận từ - tài liệu cho 4-gram là lớn để khắc phục điểm này chúng tôi đề xuất sử dụng việc phân tích SVD trên nền tính toán song song. Do vậy, đề xuất cũng sẽ bao gồm xây dựng giải pháp song song, mỗi máy (hay cụm máy) sẽ đảm nhận một công việc riêng biệt, tăng hiệu suất tối đa xử lý.

Do đó trong bài báo này chúng tôi đề xuất việc dựa trên mô hình tổng thể của PAN để xây dựng ứng dụng và đề xuất dùng SVD trên nền tính toán song song phân tán, một mặt tận dụng các ưu điểm của PAN đưa ra mặt khác sẽ cải tiến mô hình nhằm cải thiện hiệu năng xử lý của hệ thống, hỗ trợ cho việc dò tìm phát hiện sao chép được thực hiện một cách nhanh nhất.

2. Mô hình không gian vector (Vector Space Model)

Mô hình không gian vector được đề xuất năm 1975 bởi Salton và cộng sự. Mô hình không gian vector sẽ làm nhiệm vụ đưa tất cả các văn bản trong tập văn bản được mô tả bởi một tập các từ khoá hay còn gọi là các từ chỉ mục (*index terms*) sau khi đã loại bỏ các từ ít có ý nghĩa (*stop word).*

Mỗi văn bản d được biểu diễn bằng một vector một chiều của các từ chỉ mục $\vec{d}$ = (t1, t2,…, tn) với ti là từ chỉ mục thứ i ($1 \le i \le n$) trong văn bản d. Tương tự tài liệu truy vấn cũng được biểu diễn bằng một vector $\vec{q}$ = (q_1, q_2,…, q_n). Lúc đó độ đo tương tự của văn bản *d* và tài liệu truy vấn *q* chính là độ đo cosin của chúng.



Hình 2. Góc giữa vector truy vấn và vector văn bản

3. Singular Value Decomposition (SVD)

Giải thuật SVD được Golub và Kahan giới thiệu năm 1965 [7], đó là một công cụ phân rã ma trận hiệu quả được sử dụng để giảm hạng (hay số chiều) của ma trận. Kỹ thuật này được áp dụng vào nhiều bài toán xử lý văn bản khác nhau như tóm tắt văn bản, phát hiện sao chép, lập chỉ mục và truy vấn. SVD cho phép phân tích một ma trận phức tạp thành ba ma trận thành phần. Mục đích nhằm đưa việc giải quyết bài toán liên quan đến ma trận lớn, phức tạp về những bài toán nhỏ hơn.

$$A = USV^T \quad (1)$$

Trong đó

- *U* là ma trận trực giao cấp *m* x *r* (*m số từ chỉ mục*) các vector dòng của U là các vector từ chỉ mục.
- *S* là ma trận đường chéo cấp r x r có các giá trị suy biến (*singular value*) $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r$, với *r = rank(A).*
- *V* là ma trận trực giao cấp *r* x *n* (n *số văn bản trong tập văn bản*) - các vector cột của V là các vector văn bản.
- Hạng của ma trận A là các số dương trên đường chéo của ma trận *S*. Giả sử hạng của ma trận A là **r** hay **rank(A) = r** thì số Frobenius của *A* là $\|A\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i}$.

Ta có thể sử dụng SVD để xấp xỉ ma trận A với n giá trị đơn:

$$A \approx A_k = U_k S_k V_k^T \quad (2)$$

Ma trận xấp xỉ $A_k = U_k S_k V_k^T$ có hạng là k với k << r, trong đó:

- U_k, V_k là ma trận trực giao
- S_k là ma trận chéo cấp $k \times k$
- r là hạng của A
- k là số chiều được chọn trong mô hình giảm lược ($k \leq r$).

Giảm lược số chiều, lựa chọn k là tới hạn. Đúng như ý tưởng, chúng ta muốn một giá trị k đủ lớn để phù hợp mọi đặc tính cấu trúc thực của dữ liệu, nhưng đủ nhỏ để lọc ra các chi tiết không phù hợp hay các chi tiết không quan trọng.

Việc tính toán phân rã ma trận với SVD đòi hỏi thời gian tính toán cao, vì vậy để rút ngắn thời gian tính toán có thể dùng giải pháp tính toán song song. Trong cài đặt cụ thể, có thể dùng một khung phát triển tính toán song song như JPPF.

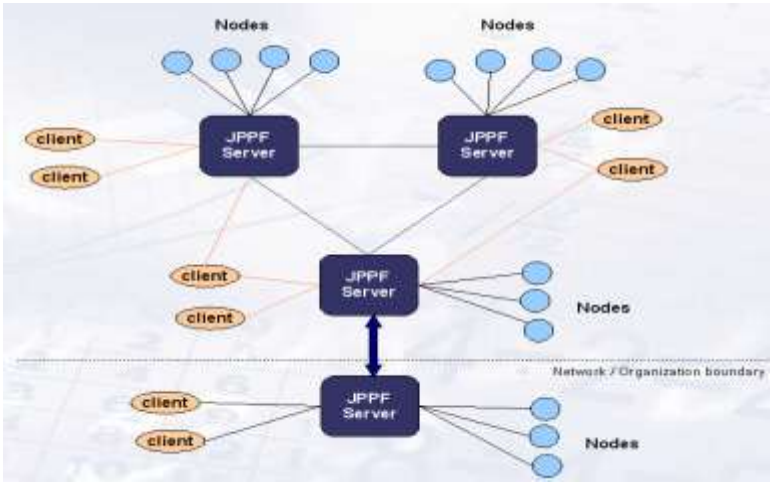
4. Kiến trúc phân tán với JPPF

JPPF [8] là một framework nguồn mở viết bằng Java cho phép phát triển các ứng dụng phân tán có nguồn dữ liệu lớn. Là một framework cho phép phát triển các ứng dụng phân tán, cung cấp hệ thống các lớp để tạo kết nối, thực hiện phân phối Task và chức năng quản lý. Các tính năng chính của JPPF:

- Dễ dàng mở rộng lưới và thực thi.
- Mô hình lập trình đơn giản, trừu tượng hoá độ phức tạp của xử lý phân tán và song song.
- Hỗ trợ công cụ trực quan để quản lý và theo dõi tác vụ.
- Tính chịu lỗi và tự sửa lỗi cao.

JPPF cung cấp những lớp giúp việc tạo kết nối giữa ứng dụng và Server Task bao gồm các tính năng sau:

- Quản lý một hoặc nhiều kết nối tới Server Task.
- Gửi Task tới Server Task và nhận kết quả từ Server Task.
- Xử lý những thông báo về việc thực thi Task.



Hình 3. Kiến trúc JPPF

Trong kiến trúc này, mỗi máy chủ JPPF Server đóng vai trò trung tâm, giao tiếp với các nút Nodes theo kiến trúc master/worker, tại đó mỗi máy master sẽ phân phối công việc cho các nút worker. Các ứng dụng Client được kết nối với một hoặc nhiều máy master, tăng khả năng chịu lỗi và cân bằng tải. Mô hình xử lý công việc của JPPF sẽ có hai đơn vị sau :

- Tác vụ (task): là đơn vị nhỏ nhất được thực hiện trong lưới JPPF.
- Công việc là một nhóm các task được gửi đi cùng lúc.

Với sự hỗ trợ của cụm máy JPPF, hệ thống có thể xử lý các yêu cầu một cách song song:

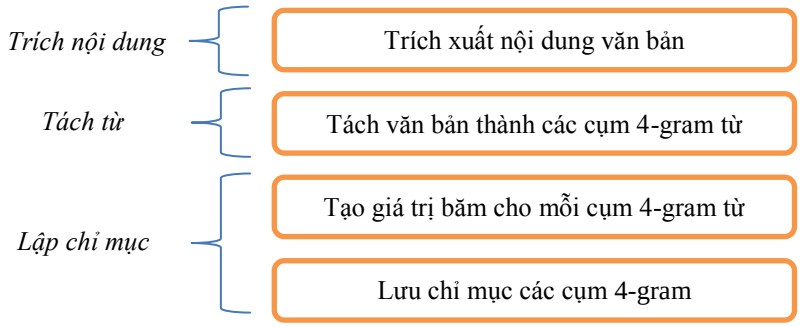
- Đối với các yêu cầu lập chỉ mục, việc lập chỉ mục 1 tài liệu và lưu vào 1 index nào đó được xem là một JPPF job độc lập được xử lý trên cụm máy JPPF. Tại một thời điểm có thể có nhiều job được thực hiện đồng thời (song song).
- Đối với các yêu cầu phát hiện sao chép, việc kiểm tra sao chép 1 tài liệu với 1 index được xem như một JPPF job được xử lý độc lập trên cụm máy JPPF. Có thể có nhiều job được thực hiện đồng thời tại một thời điểm (song song).

IV. ỨNG DỤNG SVD VÀO BÀI TOÁN PHÁT HIỆN SAO CHÉP

Áp dụng kỹ thuật tách giá trị đơn việc đầu tiên sẽ mô hình hóa tập văn bản nguồn và văn bản kiểm tra bằng mô hình không gian vector. Việc xác định mức độ sao chép được tính qua độ tương tự giữa văn bản kiểm tra và tập văn bản nguồn. Sơ đồ xử lý của hệ thống như sau :

1. Tiền xử lý văn bản, lập chỉ mục.

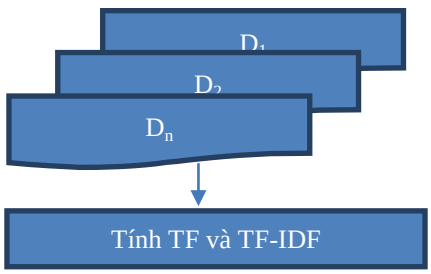
Mục đích của giai đoạn này là nhằm tăng độ chính xác của tập dữ liệu, từ đó sẽ làm tăng độ chính xác của hệ thống. Tách từ: Các tập tin văn bản được tách từ. Vị trí của ký tự bắt đầu và ký tự kết thúc của mỗi từ cũng được lưu trữ.



Hình 4. Các bước tiền xử lý văn bản, lập chỉ mục

2. Mô hình hóa văn bản

Áp dụng mô hình không gian vector để biểu diễn văn bản. Với mô hình này, mỗi văn bản được mô hình hóa thành một vector đặc trưng. Không gian các đặc trưng (tất cả đặc trưng) của tất cả các văn bản đang xét về nguyên tắc nó bao gồm tất cả các từ trong một ngôn ngữ, ở đây là tất cả các từ trong tiếng Việt.



Hình 5. Quá trình tính trọng số lập ma trận văn bản

Theo phương pháp TF-IDF trọng số từ là tích của tần suất từ (TF) và tần suất tài liệu nghịch đảo của từ đó (IDF). Trọng số TF-IDF kết hợp thêm giá trị tần suất tài liệu nghịch đảo IDF vào trọng số TF. Khi một từ xuất hiện trong càng ít tài liệu (tương ứng với giá trị IDF nhỏ) thì khả năng phân biệt các tài liệu dựa trên các từ đó càng cao. Giả sử ta phải mô hình hóa N tài liệu, mà từ N tài liệu này tách được K từ riêng biệt (không gian đặc trưng có K chiều).

Kí hiệu tài liệu thứ i là D_i (với $1 \le i \le N$). Tài liệu này được mô hình hóa thành vector sau:

$$D_i = (w_{i1}, w_{i2}, \dots, w_{iK})$$

với trọng số của từ j ($1 \le j \le K$) trong tài liệu i xác định theo công thức:

$$w_{ij} = TF_{ij} \times IDF_j = TF_{ij} \times \log(\frac{N}{DF_j}) \quad (3)$$

Trong đó:

- N: tổng số tài liệu trong tập ngữ liệu.
- K: tổng số từ tách từ N tài liệu, trong thực hành số từ tách có thể được loại bỏ các từ tầm thường.
- TF_{ij}: số lần xuất hiện của từ thứ j trong tài liệu thứ i.
- DF_j: tổng số tài liệu có chứa từ thứ j trong số N tài liệu.

Mô hình hóa 1900 tài liệu (N = 1900). Tách được 18000 từ từ tập tài liệu này. Sau khi loại bỏ các từ stopword còn lại 13274 từ (K = 13274). Ta có 1900 tài liệu được mô hình hoá thành 1900 vector với mỗi vector có số chiều là 13274.

Ta mô tả cho việc mô hình hóa này bằng một ma trận 13274 dòng (từ) và 1900 cột (tài liệu). Với mỗi ô giao nhau của hàng và cột, tính một giá trị gọi là trọng số của hàng và cột tương ứng theo công thức (3) ở trên.

3. Tìm kiếm các tài liệu tiềm năng

Do ma trận từ - tài liệu có số chiều lớn và có nhiều đặc trưng dư thừa gây nhiễu do đó sẽ áp dụng giải thuật SVD để làm giảm số chiều của ma trận này rút gọn về số chiều nhỏ hơn rất nhiều bằng việc chọn k giá trị đơn đầu tiên của ma trận giúp loại bỏ các đặc trưng gây nhiễu và tăng cường hiệu quả.

Tuy nhiên, các thuật toán cổ điển cho việc phân tích SVD là các giải thuật tuần tự cho một máy đơn, không thể thực hiện song song trong một cụm máy tính. Điều này dẫn tới khả năng áp dụng SVD vào các bài toán dữ liệu lớn rất là hạn chế. Với lý do này nên có một cách tiếp cận khác để song song hóa giải thuật tách giá trị đơn.

Khởi đầu, mỗi văn bản được mô hình hóa thành một véctơ cột trong không gian xác định bởi A_{mxn}. Sau khi giảm chiều ma trận A_{mxn} về A_k thì các tất cả các véctơ đang xét cũng được chiếu lên không gian A_k để có số chiều k theo công thức:

$$Proj\ (x) = x^T U_k S_k^{-1} \tag{4}$$

Trong đó :

- Proj(x): véctơ cột thứ x của ma trận A_{kxn}.
- x^T : chuyển vị của véctơ cột thứ x của ma trận ban đầu A_{mxn}.
- U : ma trận rút trích của U theo không gian k chiều.
- S_k^{-1} : ma trận rút trích nghịch đảo của S theo không gian k chiều.

Với q là một tài liệu kiểm tra bất kỳ ta sẽ lập Vector X_q Tuy nhiên, khi áp dụng việc phân tích SVD chúng ta chỉ quan tâm *k* khái niệm quan trọng chứ không xét hết tất cả *t* thuật ngữ. Do đó để tìm ra *p* tập tài liệu tiềm năng (d_1, d_2, …, d_p) phù hợp với q dựa trên tính toán độ tương đồng giữa tài liệu kiểm tra và các tài liệu nguồn theo độ đo cosin trong "không gian văn bản" (*document space*) – chính là so sánh các vector cột trong ma trận V_k^T .

- Chuyển vector tài liệu kiểm tra *q* như một cột mới vào V_k^T . Ta chiếu *q* vào không gian văn bản *k* chiều.
- Từ công thức ma trận $A_k = U_k S_k V_k^T$ ta suy ra $S_k^{-1} U_k^T A_k = V_k^T$ vì ($U_k U_k^T = I_k$) vậy ta có $V_k = A_k^T U_k S_k^{-1}$.
- Áp dụng tương tự cho vector tài liệu *q*: $q_k = q^T U_k S_k^{-1}$.
- Cuối cùng ta tính độ đo *cosin* của vector q_k với các vector văn bản trong ma trận V_k^T :

$$\cos\theta_j = \frac{q_k (V_k^T)_j}{\|q_k\|_2 \left\|(V_k^T)_j\right\|_2} \tag{5}$$

với $\left(V_k^T\right)_j$ là vector cột thứ *j* của ma trận V_k^T .

Như vậy sau quá trình áp dụng SVD để rút trích ma trận đặc trưng ta thu được một ma trận đặc trưng có số chiều nhỏ hơn nhiều so với ma trận ban đầu. Ngoài việc rút gọn số chiều của ma trận nó còn loại bỏ các thuộc tính nhiễu, chỉ giữ lại các thuộc tính nổi trội của văn bản.

4. Đo độ tương đồng của văn bản kiểm tra và văn bản nguồn

Vấn đề xác định độ tương đồmg giữa hai văn bản hoàn toàn dựa trên việc xác định độ tương đồng của các đoạn văn bản nằm trong văn bản. Tính độ tương đồng của hai đoạn văn bản bằng cosin góc của hai vector biểu diễn hai văn bản.

Giải thuật tính độ tương đồng của hai văn bản được xây dựng như sau:

- Giả sử văn bản kiểm tra có m đoạn là $T=\{T_1, \ldots, T_m\}$; Văn bản nguồn có n đoạn là $S=\{S_1, \ldots, S_n\}$. Ta có thể tính độ tương đồng cho từng cặp (T_i, S_j), i = 1,… m ; j = 1,… n.
- α là giá trị xác định ngưỡng "độ tương đồng", α một hằng số nào đó, 50% chẳng hạn, nếu độ tương đồng của hai đoạn lớn hơn hoặc bằng α thì kết luận hai đoạn văn bản giống nhau.
- Sau đó sẽ được xếp hạng theo thứ tự tính tương đồng giảm dần của văn bản kiểm tra với tập văn bản nguồn sau khi áp dụng SVD.

Trong mô hình không gian vector, giả sử rằng tồn tại cố định tập các thuật ngữ chỉ mục để đại diện tập tài liệu tiềm năng và tài liệu kiểm tra. Tài liệu tiềm năng D_i và tài liệu kiểm tra Q_j được biểu diễn như hai vector:

- $D_i = [T_{i1}, T_{i2}, \ldots, T_{ik}, \ldots, T_{iN}]$
- $Q_j = [Q_{j1}, Q_{j2}, \ldots, Q_{jk}, \ldots, Q_{jN}]$

trong đó, T_{ik} là trọng số của thuật ngữ k trong tài liệu i, Q_{jk} là trọng số của thuật ngữ k trong tài liệu kiểm tra j, và N là tổng số thuật ngữ sử dụng trong các tài liệu và tài liệu kiểm tra.

Các trọng số thuật ngữ T_{ik} và Q_{jk} có thể là chỉ số TF-IDF. Việc truy tìm trong mô hình không gian vector được thực hiện dựa trên cơ sở tính tương đồng giữa tài liệu kiểm tra và các tài liệu tiềm năng. Độ tương đồng giữa tập tài liệu D_i và tài liệu kiểm tra Q_j được tính như sau:

$$S\left(D_i, Q_j\right) = \sum_{k=1}^{N} T_{ik}, Q_{jk} \tag{6}$$

Để bù vào độ chênh lệnh giữa kích thước của tập tài liệu tiềm năng và kích thước của tài liệu kiểm tra, tính tương đồng trên có thể chuẩn hóa với θ là góc của hai vector (gọi là khoảng cách *cosin*) và được biểu diễn như sau:

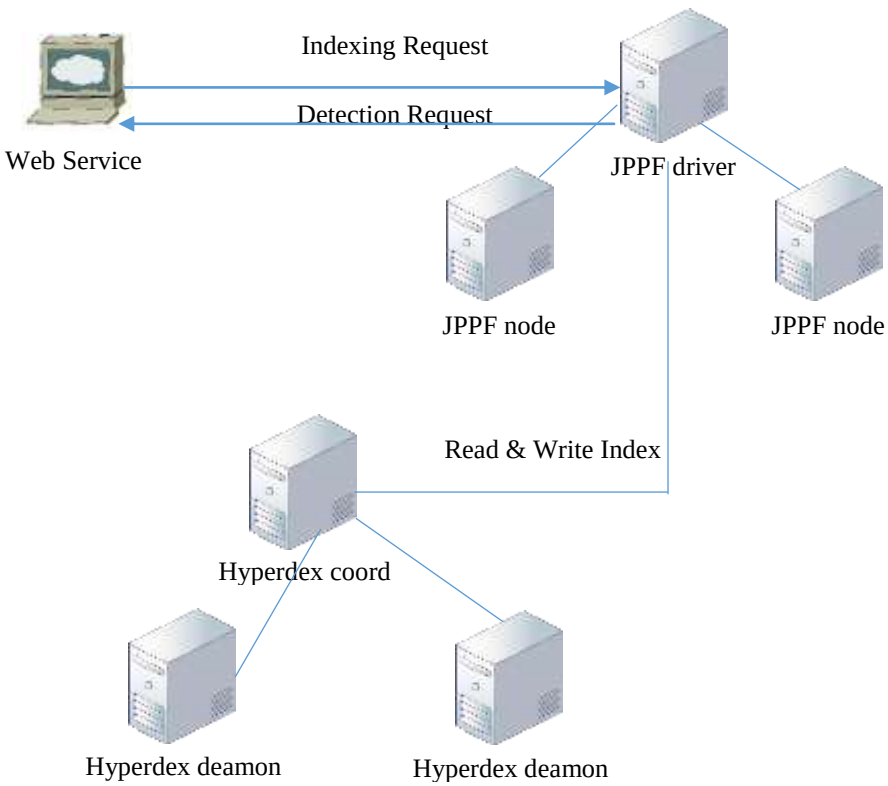
$$S\left(D_i, Q_j\right) = \cos\theta = \frac{D_i \times Q_j}{|D_i| \times |Q_j|} = \frac{\sum_{k=1}^{N} T_{ik} . Q_{jk}}{\sqrt{\sum_{k=1}^{N} T_{ik}^2} . \sqrt{\sum_{k=1}^{N} Q_{jk}^2}} \tag{7}$$

5. Tinh lọc kết quả

Các phát hiện chồng chéo nhau hoặc các phát hiện có tỉ lệ tương đồng giữ các đoạn văn bản sao chép và đoạn văn bản nguồn (hoặc ngược lại) nhỏ hơn ngưỡng α được loại bỏ ra khỏi kết quả.

V. THỰC NGHIỆM

1. Kiến trúc hệ thống thực nghiệm



Hình 6. Kiến trúc hệ thống phát hiện sao chép

Có 3 thành phần chính trong hệ thống:

- *Máy chủ giao diện web* (Web Server): đây là Server chạy các ứng dụng web giao diện tương tác giữa người dùng và hệ thống. Server này có nhiệm vụ thực thi 2 công việc chính:
  - Lập chỉ mục cho các tài liệu nguồn.
  - Thực hiện việc kiểm tra sao chép.

  Thông qua Web Server người quản trị sẽ có nhiệm vụ cập nhật chỉ mục cho các tài liệu của hệ thống, trong khi người dùng bình thường có thể kiểm tra việc sao chép của một tài liệu kiểm tra bất kỳ.
- *Máy chủ ứng dụng* (Application Servers): là các Server được cài đặt JPPF [8]. Hệ thống hỗ trợ xử lý song song các yêu cầu từ các Web Server các tác vụ như lập chỉ mục tài liệu hoặc kiểm tra sao chép. Bên cạnh đó JPPF còn cung cấp việc mở rộng hệ thống một cách dễ dàng (bằng việc kết nối thêm máy tính vào cụm máy JPPF đã có).
- *Kho lưu trữ chỉ mục* (Index store): ở đây hệ thống sử dụng phần mềm HyperDex. Đây là kho lưu trữ chỉ mục dạng phân tán có khả năng lưu trữ lượng lớn chỉ mục tài liệu và cũng như JPPF nó có thể mở rộng dễ dàng.

Một cách tổng quát, khi có một yêu cầu "*lập chỉ mục tài liệu*" hay "*kiểm tra sao chép*" từ các người dùng gởi đến hệ thống thông qua giao diện Web, Web Server sẽ gửi các yêu cầu dạng RESTful đến các máy chủ ứng dụng để xử lý. Các tác vụ "*lập chỉ mục*" hay "*kiểm tra sao chép*" đều được tiến hành song song tại các Application Server này dựa

trên kiến trúc JPPF nhằm giảm thời gian xử lý. Ngoài ra, để đọc hay ghi lại các chỉ mục Application Server sẽ tiến hành truy xuất đến Index Store để thực hiện các thao tác trên.

Ưu điểm của kiến trúc cho hệ thống trên:

- Kiến trúc hệ thống nhắm đến tính song song của các tác vụ, mỗi cụm máy Server sẽ đảm nhiệm một công việc riêng biệt, do đó các cụm máy này sẽ được sử dụng với hiệu suất tối đa.
- Dễ dàng nâng cấp, mở rộng các cụm máy Server khi có nhu cầu lưu trữ hay xử lý tăng cao, mà hoàn toàn không làm ảnh hưởng đến các thành phần còn lại của hệ thống.
    - Có thể thêm các máy tính riêng lẻ vào cụm Server Index Store để tăng khả năng lưu trữ.
    - Hoặc thêm vào cụm Application Server để tăng khả năng xử lý tính toán.

Ngoài ra trong kiến trúc hệ thống ở trên còn có hai thành phần cung cấp dữ liệu nguồn (thư viện) là các máy chủ CSDL sẵn có đang hoạt động, chứa các dữ liệu về luận văn và bài báo khoa học của thư viện điện tử của một trường nào đó ở bài báo này là sử dụng thư viện điện tử của Trường Đại học Cần Thơ

2. Dữ liệu thực nghiệm

a) Tập dữ liệu nguồn

Tài liệu nguồn cho hệ thống phát hiện sao chép dựa vào hai nguồn: các luận văn sau đại học và các bài báo khoa học của thư viện điện tử của Trường Đại học Cần Thơ. Các tập tin tài liệu này có nhiều định dạng khác nhau như (.pdf, .doc, .docx,…), với số lượng hơn 4000 tài liệu nhưng trong bài báo này chỉ trích lọc ra 1900 tài liệu để làm dữ liệu nguồn. Các văn bản kiểm tra sẽ được so sánh với tập nguồn này.

Từ tập dữ liệu mẫu qua các bước xử lý theo PAN (xử lý tất cả các văn bản, tách từ, loại stopword, tính trọng số TF-IDF). Sau quá trình xử lý thu được ma trận có kích thước 13274 x 1900.

b) Tập dữ liệu kiểm tra

Từ 1900 tập tin văn bản nguồn, 3 tập kiểm tra được thiết kế bằng cách lấy một số tập tin ngẫu nhiên rồi thay đổi cho phù hợp với nhu cầu:

- Tập kiểm tra gồm các sao chép nguyên văn (chép và dán).
- Tập kiểm tra gồm các sao chép có sửa đổi với mức độ ít.
- Tập kiểm tra gồm các sao chép có sửa đổi với mức độ cao.

Bảng 1. Xây dựng tập kiểm tra

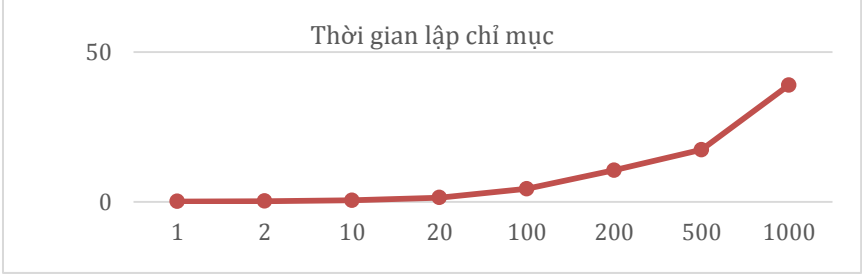
		Mức độ sửa đổi	**# Tài liệu kiểm tra**
1	Chép và dán	10%	30
2	Sao chép với sửa đổi ít	40%	30
3	Sao chép với sửa đổi nhiều	50%	30

- Tập kiểm tra 1 (Test 1) sao chép nguyên văn: gồm 30 tài liệu, đối với mỗi tài liệu sẽ thay đổi thứ tự các đoạn văn bản trong trong tài liệu, ghép một số câu văn ngắn thành một đoạn văn bản dài. Như vậy tập test1 gồm 30 tài liệu được coi là sao chép nguyên văn từ tập nguồn nằm trong 1900 tài liệu.
- Tập kiểm tra 2 (Test 2) sao chép với sửa đổi ít: gồm 30 tài liệu, đối với mỗi tài liệu sẽ thay đổi thứ tự các đoạn văn bản trong trong tài liệu, xóa một số đoạn trong các tài liệu, thêm vào một số đoạn khác tương ứng. Như vậy tập test 2 gồm 30 tài liệu được coi là sao chép với sửa đổi ít từ tập nguồn nằm trong 1900 tài liệu.
- Tập kiểm tra 3 (Test 3) sao chép với sửa đổi ít: gồm 30 tài liệu, đối với mỗi tài liệu sẽ xóa 1/2 đoạn trong các tài liệu, thêm vào một số 1/2 đoạn khác tương ứng. Như vậy tập test 3 gồm 30 tài liệu được coi là sao chép với sửa đổi nhiều từ tập nguồn nằm trong1900 tài liệu.

3. Kết quả thực nghiệm

Chúng tôi đánh giá giải thuật phát hiện được cài đặt dựa trên các độ đo đã được sử dụng rộng rãi là *Precision, Recall.*

a) Thời gian lập chỉ mục

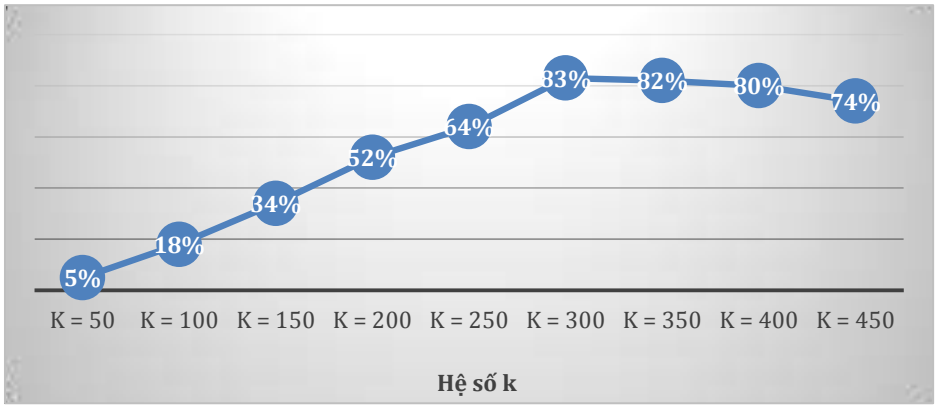


Hình 7. Biểu đồ thời gian lập chỉ mục

Hình 7 cho thấy thời gian thực hiện tác vụ lập chỉ mục tài liệu. Khi số lượng tập tin tăng lên, thời gian thực lập chỉ mục tài liệu tăng theo nhưng không đáng kể.

b) Lựa chọn tham số k

Đồ thị sau đây biểu diễn trên tập dữ liệu 1900 văn bản và 13274 từ. Ứng với mỗi k thử nghiệm sẽ cho kết quả với Precision tương ứng.



Hình 8. Đồ thị độ đo Precision với tham số k

c) Thực nghiệm trên các tập kiểm tra

Với mỗi văn bản, trong tập Test1, 2, 3 thực hiện kiểm tra với hệ thống để xác định mức độ sao chép. Ứng với mỗi tập hệ thống sẽ đặt một ngưỡng α (ngưỡng độ tương đồng giữa tài liệu kiểm tra và các tài liệu) những tài liệu trả về nào lớn hơn ngưỡng này mới được lấy để thực hiện kiểm tra sao chép và kết quả như bảng sau:

Bảng 2. Kết quả Precision trung bình với ngưỡng α

		Precision		
STT	**Test**	*α =30*	*α = 45*	*α =60*
1	Test 1_1	24%	89%	100%
2	Test 1_3	22%	86%	100%
….	…..	…..	…..	……
29	Test 3_27	70%	62%	96%
30	Test 3_30	88%	85%	96%
Precision Trung bình		**64%**	**79%**	**90%**

Ngưỡng α thích hợp cho phương pháp này là 60% hệ thống cho kết quả Precision là 90% trong trường hợp văn bản có sao chép hoặc có mức độ tương đối giống với tài liệu nguồn.

d) So sánh và đánh giá giữa phương pháp mới và phương pháp PAN

- Kết quả thực nghiệm trên tập dữ liệu được thể hiện trong Bảng 1.

Bảng 3. Kết quả đo chỉ số đánh giá phương pháp mới

STT	**Tập dữ liệu**	**Precision**	**Recall**
1	Chép và dán (Test 1)	90.8%	90.9%
2	Sao chép với sửa đổi ít (Test 2)	88.1%	86.5%
3	Sao chép với sửa đổi nhiều (Test 3)	90.4%	84.1%
Tổng		**89.9%**	**86.9%**

- So sánh giữa phương pháp mới và phương pháp của PAN

Bảng 4. So sánh chỉ số đánh giá của hai mô hình

STT	**Mô hình giải thuật**	**Precision**	**Recall**
1	PAN	91%	89%
2	Cải tiến với SVD	90%	87%

  - Kết quả của mô hình mới sử dụng giải thuật tách giá trị đơn và độ đo cosin vào bài toán tuy có kết quả không bằng so với mô hình giải pháp PAN nhưng vẫn xấp xỉ với giải pháp PAN do có thể chấp nhận được.

- Độ chính xác xấp xỉ gần bằng với mô hình gốc của PAN tuy nhiên đã đề xuất được cách xác đinh tập tài liệu tiềm năng bị sao chép và sắp xếp (ranking) chúng, từ đó có thể hạn chế số lượng tập tin cần phân tích, so sánh để phát hiện ra các đoạn bị sao chép.

VI. KẾT LUẬN

Với việc cài đặt thành công giải thuật tách giá trị đơn trên mô hình tính toán song song, có thể tận dụng được sức mạnh của tính toán song song vào việc tách giá trị đơn cho một ma trận lớn và kích thước của ma trận có thể được mở rộng khi gia tăng các node trong mô hình, đồng thời rút ngắn thời gian thực hiện giải thuật tách giá trị đơn so với cách cài đặt truyền thống là cài đặt trên một máy tính duy nhất.

Qua thực nghiệm trên dữ liệu các luận văn sau đại học của Trường ĐHCT có thể thấy rằng việc áp dụng giải thuật tách giá trị đơn (SVD) vào hệ thống phát hiện sao chép đã cho phép xác định độ tương đồng của hai văn bản từ đó có thể làm cơ sở cho việc sắp xếp và lựa chọn số tập tiềm năng theo độ tương đồng. Điều này khắc phục được nhược điểm của việc xác định tập tiềm năng bằng số 4-gram chung đang dùng hiện nay trong các giải pháp của PAN. Về mặt tính toán kiến trúc song song với khung JPPF cho thời gian tính toán khả thi. Kiến trúc này còn dễ dàng mở rộng về sau nhằm tăng khả năng lưu trữ, tính toán và rút ngắn thời gian xử lý.

TÀI LIỆU THAM KHẢO

[1] Meuschke, N. and B. Gipp, "State of the Art in Detecting Academic Plagiarism", International Journal for Educational Integrity, 9(1): p. 50-71, 2013.

[2] Ercegovac, Z. and J.V. Richardson, "Academic Dishonesty, Plagiarism Included, in the Digital Age: A Literature Review". College & Research Libraries, 65(4): p. 301-318, 2004.

[3] Park, C., "In Other (People's) Words: Plagiarism by university students--literature and lessons". Assessment & Evaluation in Higher Education, 28(5): p. 471-488, 2003.

[4] Weber-Wulff, D, "Test cases for plagiarism detection software. in Proceedings of the 4th International Plagiarism Conference", 2010

[5] Kasprzak, J. and M. Brandejs, "Improving the reliability of the plagiarism detection system", Lab Report for PAN at CLEF, p. 359-366, 2010.

[6] Martin Potthast, Alberto Barrón-Cedeño, Andreas Eiselt, Benno Stein, and Paolo Rosso, "Overview of the 2nd International Competition on Plagiarism Detection". In Martin Braschler and Donna Harman, editors, Notebook Papers of CLEF 10 Labs and Workshops. ISBN 978-88-904810-0-0, 2010

[7] E Garcia,"SVD and LSI tutorial", MIIslita.com, 2006.

[8] JPPF Homepage (http://www.jppf.org).

PLAGIARISM DETECTION SYSTEM USING SINGULAR VALUE DECOMPOSITION ON DISTRIBUTED SYSTEMS

Nguyen Vo Thong Thai, Bui Vo Quoc Bao, Huynh Phung Toan, Tran Cao De

ABSTRACT— *Nowadays, most of documents are produced in digital format, which helps us be able to easily access and copy. Therefore, document copy detection is a very important tool for protecting the author's copyright. It helps verify and detect copyright violation. Singular Value Decomposition (SVD) is technique applied in latent sematic analysis to reduce the dimension thank to the rank cut. Although there are a plenty of researches approve the effectiveness of SVD, it requests more processing time and internal memory if matrix computed is extremely large. In this paper, we describe SVD based on parallel programming, built to solve big data problems on distributes systems to apply plagiarism detection. By this approach, the reduction of dimension is resolved thank to the rank cut and matrix approximation after applying SVD, the processing time is reduced thank for the parallel computing of a computer cluster.*