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.2016.000104

XÂY DỰNG MÔ HÌNH PHÂN TÁN CHO PHÂN LỚP KHỐI LƯỢNG LỚN VĂN BẢN THEO CHỦ ĐỀ

Nguyễn Hồ Duy Trí, Nguyễn Trung Quân, Nguyễn Văn Tiến, Ngô Thanh Hùng
Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh
trinhd@uit.edu.vn, 12520683@gm.uit.edu.vn, tiennv@uit.edu.vn, hungnt@uit.edu.vn

TÓM TẮT— *Sự xuất hiện của các trang mạng xã hội đã làm cho số lượng người sử dụng và lượng thông tin trao đổi trên mạng internet trở nên rất lớn và không ngừng gia tăng. Phần lớn người sử dụng mạng xã hội, blog thường bày tỏ một cách chân thật các kiến thức, ý kiến, quan điểm, cảm xúc… của chính mình. Việc phân tích chủ đề từ những trao đổi, tài liệu trên mạng xã hội nhằm nắm bắt, quản lý và trích xuất thông tin là vô cùng quan trọng và có ý nghĩa lớn trong giáo dục, kinh tế, chính trị, xã hội, tâm lý học... Tuy nhiên để có được những thông tin hữu ích chúng ta phải giải quyết các vấn đề phức tạp ở cả hai giai đoạn: thu thập dữ liệu từ các trang mạng xã hội và phân tích thông tin từ nguồn dữ liệu lớn.*

Thông thường bài toán phân tích thông tin, cụ thể là phân lớp bài viết theo chủ đề, là bài toán xử lý, phân loại văn bản truyền thống nhưng khi áp dụng cho dữ liệu mạng xã hội thì gặp phải khó khăn về dung lượng dữ liệu cần xử lý, có thể lên đến hàng TeraByte, ZettaByte. Để có thể lưu trữ và xử lý lượng dữ liệu này cần sử dụng các công nghệ tính toán phân tán Cluster Computing, trong đó phổ biến nhất là mô hình MapReduce.

Từ khóa— *text classification, distributed model, classification by topic, big data, spark.*

I. GIỚI THIỆU

Phân lớp văn bản là một trong những bài toán cổ điển trong khai thác dữ liệu. Nội dung bài toán phân lớp chính là đi tìm chủ đề thích hợp (tên/nhãn lớp) trong tập hữu hạn các chủ đề đã được xác định trước. Tiêu chí lựa chọn chủ đề phù hợp cho các văn bản dựa trên độ tương đồng về ngữ nghĩa giữa chúng với các văn bản trong tập ngữ liệu huấn luyện. Việc tự động phân lớp văn bản vào một chủ đề giúp cho việc tổ chức sắp xếp, lưu trữ và truy vấn tài liệu dễ dàng hơn về sau. Bên cạnh đó, phân lớp văn bản còn được sử dụng để hỗ trợ trong quá trình tìm kiếm, chiết lọc thông tin. Ngoài ra, với sự bùng nổ của mạng xã hội, việc chia sẻ những thông điệp cũng chứa đựng vô vàn thông tin hữu ích. Giải quyết bài toán phân lớp chủ đề trên tập thông điệp khổng lồ này mang lại rất nhiều ý nghĩa như: tìm ra xu hướng, chủ đề chung của cộng đồng, phát hiện người dẫn dắt ý tưởng (key player), đánh giá mức độ hữu ích của văn bản, phát hiện đạo văn hay lựa chọn văn bản làm đại diện cho tập ngữ liệu. Xa hơn nữa, nếu biết được sự quan tâm của tác giả thông điệp đến chủ đề nào, ta có thể phân tích, nắm bắt được 'tâm lý' của người dùng, từ đó dễ dàng gợi ý những tài liệu, sản phẩm… tương đồng với chủ đề và phù hợp với thị hiếu, từ đó định hướng tốt hơn cho truyền thông và marketing hiện đại.

Trên thế giới, đặc biệt là đối với tiếng Anh, đã có nhiều công trình nghiên cứu đạt được những kết quả khả quan. Tuy nhiên, những nghiên cứu và ứng dụng trên ngôn ngữ tiếng Việt còn nhiều hạn chế do gặp phải không ít khó khăn về ngữ pháp, sự nhập nhằng về ngữ nghĩa trong quá trình tách câu, tách từ. Có thể liệt kê ra một số nghiên cứu với những hướng tiếp cận khác nhau đối với bài toán phân lớp văn bản như sau: phân loại với máy học vector hỗ trợ (SVM) [1], cách tiếp cận sử dụng lý thuyết tập thô [2], cách tiếp cận thống kê hình vị [3], cách tiếp cận sử dụng phương pháp học không giám sát và đánh chỉ mục [4], cách tiếp cận theo luật kết hợp [5]. Những nghiên cứu trên đều đạt được những kết quả khá tốt, tuy nhiên khó mà so sánh chúng với nhau vì tập dữ liệu thực nghiệm trong mỗi phương pháp là khác biệt. Nhưng dù tiếp cận theo hướng nào đi nữa thì các phương pháp nêu trên đều đa phần sử dụng toàn văn nội dung của một văn bản để thực hiện phân lớp, điều này đồng nghĩa với việc các mô hình phân lớp luôn phải đối phó với một lượng lớn các đặc trưng. Trong bối cảnh thông tin được chia sẻ trên các mạng xã hội hiện nay với khối lượng dữ liệu khổng lồ, không ngừng gia tăng đáng kể hàng ngày thì việc phải thực hiện phân loại từng văn bản với nội dung cực lớn sẽ là một thách thức không nhỏ.

Trong giới hạn của bài báo, nhóm tác giả không thể khảo sát hết các hướng tiếp cận đã nêu mà chỉ chọn một phương pháp tiếp cận truyền thống theo phương pháp SVM để từ đó đề xuất mô hình nhằm giải quyết nhu cầu xử lý khối lượng dữ liệu lớn hiện nay.

II. MÔ HÌNH PHÂN LỚP THEO TIẾP CẬN TRUYỀN THỐNG BẰNG PHƯƠNG PHÁP SVM

Qua khảo sát những công trình [6][7][8][9], có thể khái quát mô hình phân lớp theo tiếp cận truyền thống bằng phương pháp SVM như Hình 1. Mô hình này gồm 3 bước cơ bản:

- Bước 1: Tiền xử lý dữ liệu

Tập văn bản ban đầu sẽ được xử lý tách câu, tách từ, loại bỏ các dấu câu và các stopword. Sau bước này, mỗi văn bản sẽ là tập hợp của các từ đã được sàng lọc trong văn bản đó.

- Bước 2: Vector hóa

Tập từ thu được từ bước tiền xử lý đang ở dạng không cấu trúc do đó để xử lý phân lớp bằng các phương pháp máy học cần vector hóa chúng. Mô hình túi từ được áp dụng, theo mô hình này, dữ liệu văn bản không có cấu trúc (độ dài khác nhau) được biểu diễn thành dạng véc tơ tần số xuất hiện của từ trong văn bản.

Từ tần số của từ, vector của từng văn bản sẽ được tính bằng công thức TF*IDF. Đây là công thức giúp đánh giá mức độ quan trọng của một từ đối với văn bản trong bối cảnh của tập ngữ liệu. TF (term frequency) là tần số xuất hiện của một từ trong một văn bản. IDF (inverse document frequency) là tần số nghịch của 1 từ trong tập ngữ liệu. Công thức như sau:

$$TF_{ij} = \frac{f_j}{n_i}$$

$$IDF_{ij} = \log\frac{N}{f(t_j)}$$

$$w_{ij} = TF_{ij} \times IDF_{ij}$$

Trong đó, f_j là số lần xuất hiện của từ t_j trong tài liệu d_i, n_i là tổng số từ trong d_i, N là tổng số tài liệu, $f(t_j)$ là số lượng các tài liệu chứa từ t_j.

Kết quả của bước này là vector phân bố xác suất của tập từ biểu diễn chủ đề của từng văn bản.

- Bước 3: Phân lớp văn bản

Tập văn bản đầu vào sau khi trải qua các bước xử lý sẽ được đại diện bằng tập các vector. Chúng sẽ là đầu vào của giải thuật SVM truyền thống.

Hình 1. Mô hình phân lớp theo phương pháp tiếp cận truyền thống bằng phương pháp SVM

III. MÔ HÌNH PHÂN TÁN ĐỀ XUẤT

A. Mô hình phân tán trong Apache Spark

Để thao tác trên khối lượng dữ liệu lớn nhóm tác giả chọn thực hiện việc cải tiến mô hình trên nền tính toán phân tán. Một trong những mô hình xử lý dữ liệu lớn rất phổ biến là MapReduce, được hiện thực bằng hai framework mã nguồn mở nổi tiếng là Apache Hadoop và Apache Spark.

map(k1,v1) → list(k2,v2)

reduce(k2,list(v2)) → list(v3)

Hình 2. Quá trình map và reduce trong mô hình MapReduce

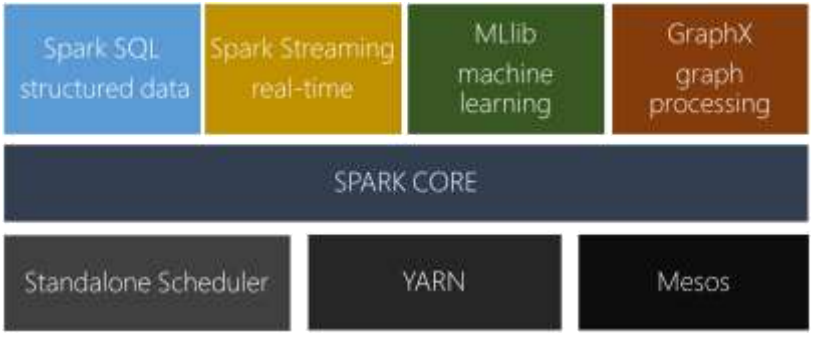
Với MapReduce, đây là một mô hình luồng dữ liệu, nó thích hợp và được ứng dụng với đa số các công cụ xử lý dữ liệu lớn hiện nay. Nhưng cũng có những ứng dụng không thích hợp khi áp dụng mô hình này, đó là những ứng dụng có dạng mô hình lặp. Trong mô hình này, quá trình xử lý cứ được lặp đi lặp lại. Lúc đó mô hình MapReduce sẽ bộc lộ những hạn chế sau [10]:

(1) Thứ nhất, có rất nhiều các giải thuật máy học thực hiện các hàm lặp đi lặp lại trên cùng một tập dữ liệu để tối ưu các tham số. Mỗi vòng lặp có thể được khai báo là một lần thực hiện quá trình MapReduce. Như vậy, mỗi lần thực thi sẽ là một lần truy vấn lại dữ liệu từ đĩa cứng, điều này làm cho cả quá trình bị chậm đi rất nhiều.

(2) Thứ hai, một quá trình MapReduce thường sử dụng một lượng lớn dữ liệu. Nếu quá trình này cứ lặp lại nhiều lần thì lý tưởng nhất là chúng ta nên tải những dữ liệu này lên bộ nhớ đệm trên các máy và truy vấn nhiều lần trên đó. Tuy nhiên Hadoop phải chịu độ trễ hàng chục giây với mỗi lần thực hiện quá trình MapReduce, bởi vì, mỗi thành phần công việc đọc dữ liệu từ đĩa và được thực hiện riêng biệt.

Chính vì thế bài báo chọn cài đặt xử lý dữ liệu lớn trên framework Apache Spark [10]. Được cải tiến và khắc phục những khuyết điểm từ mô hình Hadoop MapReduce, Apache Spark sử dụng một đối tượng bộ nhớ đặc biệt gọi là RDD (Resilient Distributed Dataset), nó là một tập hợp chỉ đọc chứa các đối tượng dữ liệu được phân tán lưu trữ ở các

nút tính toán (các máy con trong mạng tính toán). Tập hợp này cũng có khả năng mở rộng một cách mềm dẻo, tự cân bằng và khả năng chịu lỗi, phục hồi khi có sự cố xảy ra giống như Hadoop. Khi thao tác RDD sẽ được Spark tải lên bộ nhớ đệm của những nút tính toán để sử dụng nhiều lần qua các quá trình tính toán song song MapReduce, chính vì thế tốc độ của Spark có thể nhanh hơn Hadoop đến gấp 10 lần.



Hình 3. Các thành phần của framework Apache Spark

B. Tổ chức tập văn bản và tiền xử lý dữ liệu

Tập văn bản đã thu thập và gán nhãn được sử dụng cho việc huấn luyện và phân lớp được lưu trữ dưới dạng thô (plain text). Những tập tin văn bản thô này sẽ được chứa trong các thư mục tương ứng với những chủ đề khác nhau.

Sau khi thu thập và tổ chức lưu trữ dữ liệu, bước kế tiếp của giai đoạn tiền xử lý là tách câu, tách từ. Không như tiếng Anh, khoảng trắng trong tiếng Việt không thể đóng vai trò dấu hiệu phân tách các từ. Từ trong tiếng Việt có thể là từ đơn hay từ ghép, thêm vào đó sự nhập nhằng về nghĩa làm cho bài toán phân tách các từ khó có thể đạt được sự chính xác tuyệt đối. Một trong những công cụ tách từ có độ chính xác cao (theo công bố của tác giả là trong khoảng từ 96% đến 98%) đó là thư viện vnTokenizer [11], bài báo sẽ sử dụng công cụ này vào quá trình tiền xử lý tập văn bản. Kết quả thu được sau bước tách từ sẽ là đầu vào của mô hình xử lý phân tán sẽ được trình bày ở Mục C.

Tập tin văn bản sau khi xử lý tách từ sẽ được lưu trữ phân tán ra các thành phần xử lý trong mạng tính toán.

C. Đề xuất mô hình phân tán nhằm xử lý lượng lớn dữ liệu

Để xây dựng phương pháp phân lớp khối lượng văn bản lớn theo chủ đề, bài báo áp dụng mô hình phân tán trong Apache Spark vào phương pháp phân lớp theo tiếp cận truyền thống bằng phương pháp SVM đã trình bày ở Phần II. Đầu vào của mô hình phân tán là tập từ của văn bản đã được cắt bằng thư viện VnTokenizer.

Mô hình phân tán bao gồm các bước sau:

Bước 1: Ở bước đầu tiên, danh sách các file đầu vào sẽ được chia ra từng phần ứng với các nút tính toán trong mạng phân tán. Ở mỗi nút, ta tiến hành xóa các stopword và thống kê tần số xuất hiện của các từ.

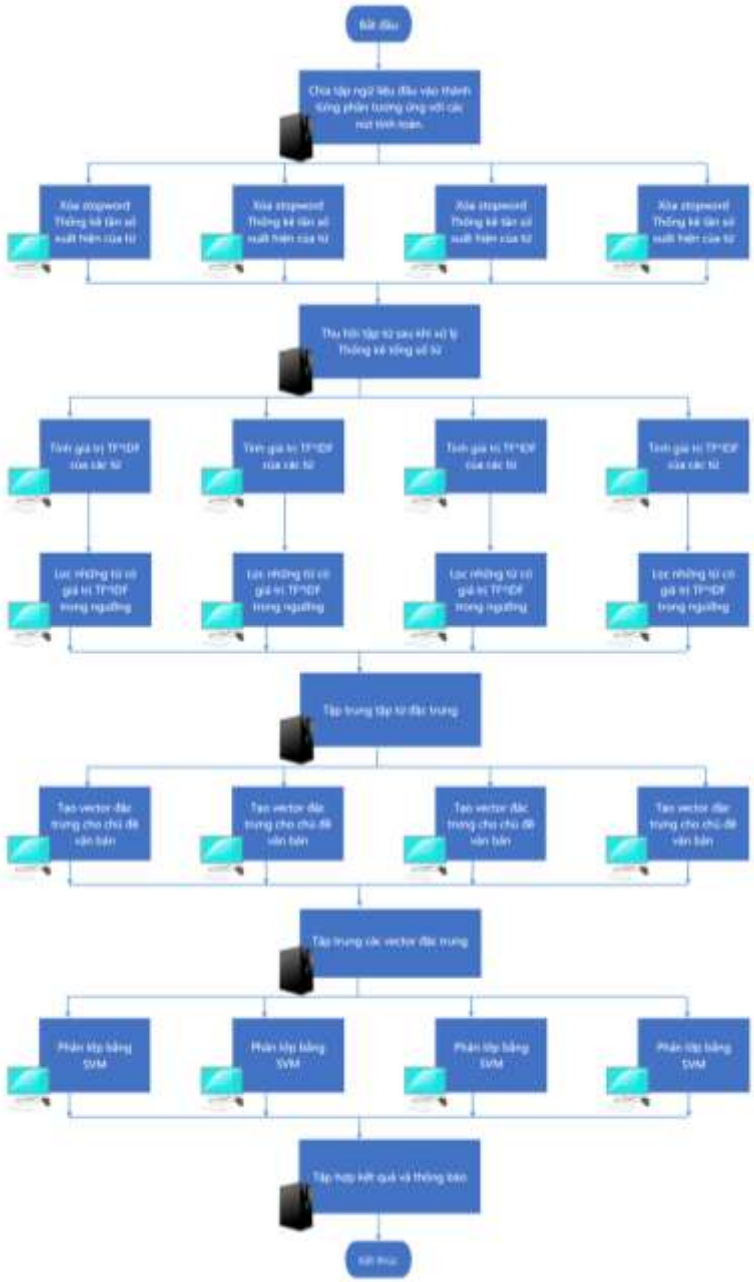
Bước 2: Tập từ qua xử lý ở bước 1 sẽ được máy tính trung tâm (driver) thu hồi lại, thống kê tổng số từ

Bước 3: Kết quả thống kê từ bước 2 sẽ được phân tán trở lại các nút tính toán để tính giá trị TF*IDF của các từ.

Bước 4: Sau khi tính TF*IDF, giải thuật sẽ lọc lấy những từ có giá trị trong ngưỡng cài đặt trước. Việc lọc này nhằm lựa ra những từ đủ tính chất đặc trưng cho chủ đề, loại bỏ những từ quá hiếm xuất hiện hoặc xuất hiện quá phổ biến. Giá trị của ngưỡng sẽ được tối ưu dần từ thực nghiệm.

Bước 5: Tiếp theo, tập từ đặc trưng sẽ được driver tập trung lại và phân tán ra các máy trạm để tạo các vector đặc trưng cho chủ đề của văn bản.

Bước 6: Tất cả vector được tập hợp tại driver và phân tán ra các máy trạm để phân lớp bằng SVM.



Hình 4. Mô hình phân tán nhằm xử lý lượng lớn dữ liệu.

IV. CÀI ĐẶT THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

A. Mô tả dữ liệu, môi trường cài đặt

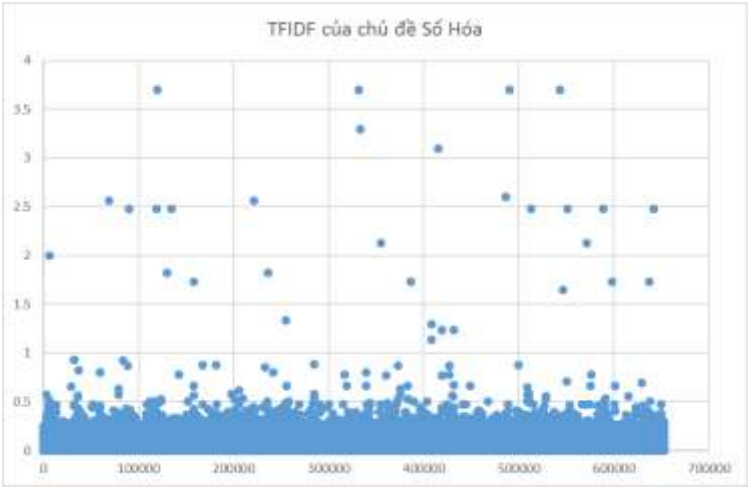
Dữ liệu văn bản được thu thập từ trang báo điện tử VnExpress, tổng số văn bản được sử dụng để thử nghiệm là 20000 bao gồm các thành phần chủ đề được mô tả trong Bảng 1 và biểu đồ Hình 5.

Bảng 1. Thống kê số tài liệu theo chủ đề

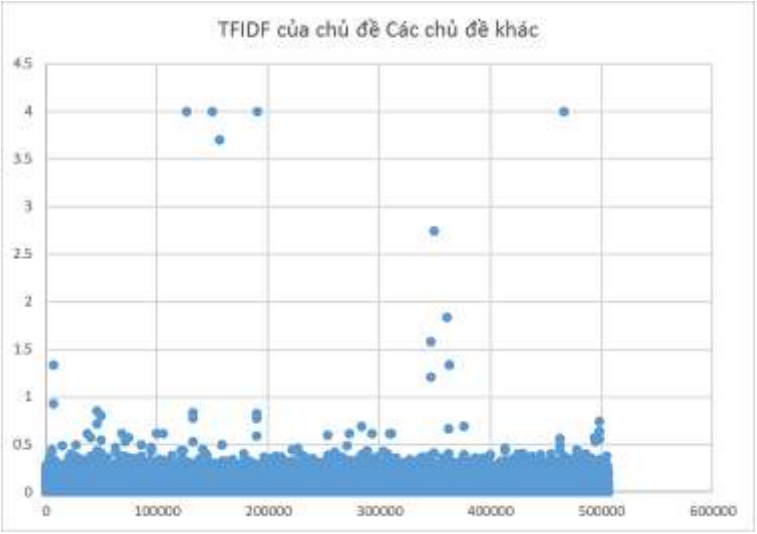
Chủ đề	Số tài liệu
Số hóa	*10000*
Cộng đồng	*1500*
Kinh doanh	*1500*
Giải trí	*1500*
Du lịch	*1500*
Thời sự	*1500*
Thể thao	*1500*
Pháp luật	*1000*
Tổng cộng	***20000***

Hình 5. Trực quan thành phần dữ liệu thử nghiệm theo chủ đề.

Tập ngữ liệu này được gán 2 chủ đề (nhãn) lớn là: Số hóa (10000 tài liệu) và Các chủ đề khác (10000 tài liệu). Trong quá trình thực nghiệm, tập ngữ liệu được chia thành 2 phần: 70% huấn luyện, 30% thử nghiệm. Sau khi tách câu, tách từ và loại bỏ stopword, nhóm tác giả thống kê chỉ số TF*IDF của các từ khóa trong tập ngữ liệu và thu được kết quả được tổng quát trong biểu đồ sau:



Hình 6. Biểu đồ phân bố giá trị TF*IDF của các từ khóa trong chủ đề Số hóa



Hình 7. Biểu đồ phân bố giá trị TF*IDF của các từ khóa trong chủ đề Các chủ đề khác

Sau khi loại bỏ các stopword thì chủ đề số hóa bao gồm 652528 từ khóa, chủ đề Các chủ đề khác bao gồm 506130 từ khóa. Từ công thức TF*IDF đã nêu ở Phần II, ta có thể thấy trong một văn bản một từ càng xuất hiện nhiều lần thì giá trị TF càng lớn, ngược lại, nếu từ đó càng hiếm thì giá trị TF lại càng thấp. Đối với toàn bộ tập ngữ liệu, từ khóa càng xuất hiện nhiều thì giá trị IDF càng thấp, còn khi nó hiếm khi xuất hiện thì giá trị IDF lại càng cao. Để bộ phân lớp hoạt động tốt, ta phải lọc được những từ khóa đặc trưng cho lớp, không quá hiếm và cũng không được quá phổ biến. Qua thực nghiệm, nhóm tác giả chọn giá trị TF*IDF trong khoảng từ 0,007 đến 0,4.

Môi trường cài đặt hệ thống phân tán là 10 máy ảo có cấu hình bình thường với vi xử lý 8 nhân và bộ nhớ RAM 8GB. Các máy được kết nối với nhau, trong đó có một máy vừa là máy con với vai trò xử lý, vừa là máy chủ với vai trò quản lý cấp phát tài nguyên, dữ liệu; thu thập, tổng hợp kết quả, xử lý những tính toán cục bộ. Các máy chạy hệ điều hành Ubuntu 16.04, được cài đặt Apache Spark 1.6.2.

B. Kết quả và đánh giá

Thời gian thực hiện mô hình phân lớp trên hệ thống xử lý phân tán gồm 10 đơn vị tính toán là 8.6 tiếng, kết quả thu về được trình bày trong Bảng 2.

Bảng 2. Bảng kết quả

Precision	Recall	F-measure
88,14%	91,99%	90,02%

Qua quá trình thử nghiệm và kết quả như trên, có thể đưa ra một vài nhận định như sau:

(1) Kết quả thu được hết sức khả quan trong việc phân lớp văn bản theo chủ đề.

(2) Hệ thống với 10 đơn vị tính toán là quá nhỏ, chỉ có thể dùng để thực nghiệm khả năng xử lý được khối lượng dữ liệu lớn và tăng trưởng theo thời gian, điều mà các hệ thống đơn xử lý khó đảm bảo.

(3) Khối lượng dữ liệu trong thực nghiệm lớn hơn nhiều so với công trình [1] 4162 tài liệu, công trình [5] 5000 tài liệu, công trình [7] 7842 tài liệu, công trình [9] 2000 tài liệu.

(4) Với khối lượng dữ liệu tương đối lớn, bao quát các trường hợp và phương pháp lọc bỏ stopword, lọc bỏ những từ khóa không đủ sức đặc trưng cho chủ đề, làm cho kết quả chính xác hơn.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Việc chia sẻ công việc ra các nút tính toán bằng phương pháp phân tán, song song hóa thông qua mô hình MapReduce giúp cho việc giảm tải bộ nhớ khi phải tính toán với những khối lượng dữ liệu lớn, đồng thời giúp cho tốc độ đọc/ghi dữ liệu thu hẹp khoảng cách với tốc độ tính toán của vi xử lý góp phần tăng tốc toàn bộ quá trình thực hiện giải thuật. Quan trọng nhất, với việc xử lý khối lượng dữ liệu lớn, chúng ta sẽ có cái nhìn đa chiều, toàn diện hơn về bài toán, từ đó kết quả được cải thiện hơn so với những tiếp cận truyền thống.

Trong tương lai, nhóm tác giả sẽ tiến hành khảo sát với nhiều cách tiếp cận để cố gắng đạt được những kết quả cao hơn.

VI. LỜI CẢM ƠN

Nghiên cứu này là sản phẩm của đề tài "Nghiên cứu các kỹ thuật xử lý dữ liệu lớn, áp dụng cho việc xác định những cá nhân có tầm ảnh hưởng trong mạng xã hội" mã số D2015-07, thuộc Trường Đại học Công nghệ Thông tin – ĐHQG-HCM.

TÀI LIỆU THAM KHẢO

1. Nguyễn Linh Giang, Nguyễn Mạnh Hiển, Phân loại văn bản tiếng Việt với bộ phân loại vectơ hỗ trợ SVM. Tạp chí CNTT&TT, Tháng 6 năm 2006.
2. Nguyễn Ngọc Bình, "Dùng lý thuyết tập thô và các kỹ thuật khác để phân loại, phân cụm văn bản tiếng Việt", Kỷ yếu hội thảo ICT.rda'04. Hà nội 2004.
3. Nguyễn Linh Giang, Nguyễn Duy Hải, "Mô hình thống kê hình vị tiếng Việt và ứng dụng", Chuyên san "Các công trình nghiên cứu, triển khai Công nghệ Thông tin và Viễn thông, Tạp chí Bưu chính Viễn thông, số 1, tháng 7-1999, trang 61-67. 1999
4. Huỳnh Quyết Thắng, Đinh Thị Thu Phương, "Tiếp cận phương pháp học không giám sát trong học có giám sát với bài toán phân lớp văn bản tiếng Việt và đề xuất cải tiến công thức tính độ liên quan giữa hai văn bản trong mô hình vectơ", Kỷ yếu Hội thảo ICT.rda'04, trang 251-261, Hà Nội 2005.
5. Đỗ Phúc, "Nghiên cứu ứng dụng tập phổ biến và luật kết hợp vào bài toán phân loại văn bản tiếng Việt có xem xét ngữ nghĩa", Tạp chí phát triển KH&CN, tập 9, số 2, pp. 23-32, năm 2006
6. T. T. Huỳnh và S. T. Trần, "Hệ thống nhận dạng và phân loại văn bản", Đại học Công nghệ thông tin, Hồ Chí Minh, 2007.
7. Trần Đệ Cao; Phạm Khang Nguyên, "Phân loại văn bản với máy học vector hỗ trợ và cây quyết định", Tạp chí Khoa học, 21a, trang 52-63, 2012.

8. T. T. T. Trần, C. T. Vũ và N. Tạ, "Xây dựng hệ thống phân loại tài liệu Tiếng Việt", Khoa Công nghệ Thông tin, Trường ĐH Lạc Hồng, Biên Hòa, 11/2012.

9. Đ. Q. Trương, "Phân loại văn bản dựa trên rút trích tự động tóm tắt của văn bản", trong Kỷ yếu Hội nghị quốc gia lần thứ VIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin, Hà Nội, 2015.

10. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker and I. Stoica (2010), "Spark: Cluster Computing with Working Sets," in Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, Boston, MA.

11. Lê Hồng Phương, Nguyễn Thị Minh Huyền, Azim Roussanaly, Hồ Tường Vinh, "A Hybrid Approach to Word Segmentation of Vietnamese Texts", Language and Automata Theory and Applications: Second International Conference, LATA 2008, Tarragona, Spain, March 13-19, 2008. Revised Papers, Springer-Verlag, Berlin, Heidelberg, 2008 [doi>10.1007/978-3-540-88282-4_23].

BUILDING DISTRIBUTED MODEL FOR CLASSIFICATION MASSIVE TEXT DATA BY TOPIC

Nguyen Ho Duy Tri, Nguyen Trung Quan, Le Van Duyet, Ngo Thanh Hung

ABSTRACT*— The appearance of the social networking sites has attracted users and generated massive amounts of information every day. Social network users predominantly express their true emotions, sentiments, opinions and knowledge. It is important and necessary to classify social network's posts, conversations into topics for better information retrieval. Such rich information can be a useful resource for the economy, education, and psychology. To address this problem, we experiment with two stages: collect data from social network sites and examine large data. When applying text classification for such an extensive data from social networks, which may be terabytes, it can be difficult to store and analyze it. In this paper, we overcome that difficulty with a parallel computing technique based on MapReduce.*