VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

BÁO CÁO
KHAI PHÁ DỮ LIỆU

Tên đề tài: Tự động phân loại tin tức văn bản dựa trên các phương pháp học máy

Giảng viên : PGS.TS. Nguyễn Đức Dũng

Học viên : Phan Thanh Hoài

Lớp : ITT18A-01

Hà Nội, 2018

VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

BÁO CÁO
KHAI PHÁ DỮ LIỆU

Tên đề tài: Tự động phân loại tin tức văn bản dựa trên các phương pháp học máy

Giảng viên : PGS.TS. Nguyễn Đức Dũng

Học viên : Phan Thanh Hoài

Lớp : ITT18A-01

Hà Nội, 2018

MỤC LỤC

DANH MỤC HÌNH ẢNH

LỜI MỞ ĐẦU

Trong khuôn khổ môn học Khai phá dữ liệu, nằm nắm vững hơn các thuật toán, đồng thời thể hiện được việc nghiên cứu và học tập của mình, em xin được thực hiện đề tài "Tự động phân loại tin tức văn bản dựa trên các phương pháp học máy".

Đề tài của em xuất phát từ nhu cầu môn học và tìm hiểu tin tức sự kiện được đăng trên các báo điện tử.

Từ nhu cầu thực tế và thông qua các kiến thức đã được học, em đề xuất giải pháp sử dụng thu thập tự động bằng cách sử dụng jsoup phiên bản 1.8.3 để tìm kiếm và trích xuất thông tin. Dữ liệu được lấy từ báo điện tử Dân Trí (http://dantri.com.vn) với 5 chủ đề nổi bật được lựa chọn là Sức khoẻ, Pháp luật, Kinh doanh, Thể thao, và Xe.

Trong quá trình tìm hiểu em có sử dụng một số tài liệu tham khảo bao gồm tài liệu môn học và các tài liệu chuyên ngành.

Trong quá trình tìm hiểu, do thời gian và kiến thức còn hạn chế nên trong báo cáo còn có sai sót, em rất mong thầy giáo và các bạn đóng góp ý kiến để đề tài của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

I. GIỚI THIỆU

1.1. Tự động phân loại văn bản

Trong thời đại công nghệ hiện nay, trí tuệ nhân tạo (AI) đã có thể thay thế con người trong khá nhiều công việc, trong đó bằng khả năng tính toán rất nhanh, máy tính đã chiếm ưu thế lớn trong một số nhiệm vụ nhất định, có thể kể đến như khả năng chơi cờ vây (AlphaGo chiến thắng đại kiện tướng cờ vây Lee Sedol), hay khả năng nhận diện khuôn mặt với độ chính xác cực cao, hơn cả con người. Do sự bùng nổ dữ liệu lớn, việc phân tích dữ liệu văn bản của các bài báo, tin tức khó có thể thực hiện thủ công.

Hiểu nội dung văn bản đóng vai trò rất quan trọng trong việc tự động hoá phân loại tin tức, từ đó máy tính có thể hiểu và tự động gợi ý các bài báo liên quan cho người xem sau khi đọc một số bài báo nhất định (hệ khuyến nghị) [1], hay có thể phân loại để lọc các bài viết có nội dung hướng đến một nhóm đối tượng cụ thể, hoặc đơn giản là phân loại các bài báo cũ vào các danh mục phù hợp. Việc này cũng rất có ích để kiểm soát các nội dung được đăng tải trên không gian mạng, giúp chính phủ quản lý chặt chẽ hơn các thông tin sai lệch trên mạng internet, đặc biệt phù hợp khi luật An ninh mạng sắp được thực hiện tới Việt Nam từ năm 2019.

Mặc dù công nghệ này không mới, đã được áp dụng trong nhiều ngôn ngữ khác nhau (ví dụ tiếng Anh, tiếng Trung, tiếng Nhật), nhưng cho tiếng Việt còn khá hạn chế. Xây dựng thành công hệ thống này có thể giúp các trang báo điện tử tự động gợi ý các bài báo mà người đọc có thể thích, từ đó tăng lượt truy cập và phần nào góp phần truyền tải nhiều thông tin hơn tới cộng đồng.

1.2. Yêu cầu bài toán

Bài toán yêu cầu xây dựng một hệ thống có thể tự động phân loại các tin tức văn bản, trong đó, cần phải giải quyết các bài toán con sau:

- *Thu thập dữ liệu:* Đây là phần quan trọng đầu tiên trong mọi hệ thống cung cấp thông tin. Với các phương pháp học máy, nhìn chung, càng nhiều dữ

liệu huấn luyện, độ chính xác càng được cải thiện. Với số lượng rất lớn, dữ liệu về các bài báo phải được tự động thu thập, tiền xử lý và chuẩn hoá.

- *Xây dựng mô hình dự đoán:* Mỗi bài báo có thể được phân loại vào một chủ đề nhất định cho trước (ví dụ Thế thao, Sức khoẻ, hay Kinh doanh). Mục tiêu của bài toán này là xây dựng một mô hình để dự đoán chủ đề của một bài báo dựa trên dữ liệu đã được chuẩn hoá của văn bản đó.

II. THU THẬP DỮ LIỆU

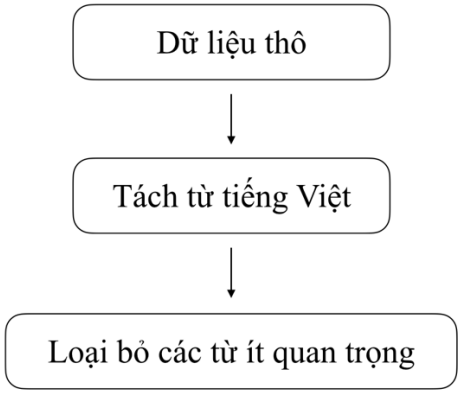
Trong báo cáo này, dữ liệu được thu thập tự động bằng cách sử dụng jsoup[1] phiên bản 1.8.3. Jsoup là một thư viện mã nguồn mở để trích chọn các thông tin từ nội dung HTML. Nó cung cấp các API rất thuận tiện cho việc tìm kiếm và trích xuất thông tin. Dữ liệu được lấy từ báo điện tử Dân Trí (http://dantri.com.vn) với 5 chủ đề nổi bật được lựa chọn là Sức khoẻ, Pháp luật, Kinh doanh, Thể thao, và Xe. Theo lý thuyết, số lượng bài báo có thể thu thập được là tất cả các bài được xuất bản công khai, nhưng do thời gian giới hạn, bước đầu 5300 bài báo đã được thu tập từ 5 danh mục trên.

Chủ đề	**Số lượng**
Sức khoẻ	1.038
Pháp luật	1.042
Kinh doanh	1.065
Thể thao	1.102
Xe	1.056
Tổng: 5 chủ đề	**Tổng:** 5.303 bài báo

Hình 1. Phân loại bài báo và số lượng đã được thu thập

[1] "jsoup Java HTML Parser, with best of DOM, CSS, and jquery." 2009. 21 Mar. 2016 http://jsoup.org/

III. TIỀN XỬ LÝ DỮ LIỆU



Hình 2. Các bước tiền xử lý dữ liệu

Dữ liệu thô sau khi thu thập từ Dân Trí sẽ được xử lý qua 2 bước tiền xử lý: *Tách từ tiếng Việt* và *Loại bỏ các từ không quan trọng* (stopwords).

3.1. Tách từ tiếng việt

Khác với tiếng Anh, mỗi từ trong tiếng Việt có thể gồm nhiều âm tiết khác nhau, điều này dẫn tới việc nhập nhằng trong việc xác định từ. Ví dụ câu "ông già đi nhanh quá" có thể được hiểu theo 2 cách khác nhau "ông_già đi nhanh quá" hoặc "ông già_đi nhanh quá". Hay câu "học sinh học sinh học" có thể được tách theo 2 kiểu khác nhau mà không vi phạm nguyên tắc cấu tạo từ: "học_sinh học sinh_học" và "học sinh_học sinh_học". Vì vậy, việc tách từ cho tiếng Việt rất quan trọng. Đầu tiên, tập dữ liệu sẽ được tách từ tiếng Việt, sử dụng công cụ mã nguồn mở JVnTextPro[2] [2]. Đây là một công cụ về xử lý ngôn ngữ tự nhiên cho tiếng Việt, được xây dựng dựa trên Conditional Random Fields (CRFs) [3] và Maximum Entropy (Maxent).

3.2. Loại bỏ stopwords

Sau khi sử dụng JVnTextPro, ta có một tập dữ liệu đã được tách từ, tuy nhiên số lượng các từ không quan trọng (stopwords) còn khá nhiều. Vì thế, các stopwords không mang nhiều ý nghĩa được loại bỏ, ví dụ như "nhưng, và,

[2] "JVnTextPro - A Java-based Vietnamese Text Processing Tool <http://jvntextpro.sourceforge.net/>

cũng,...". Danh sách các stopwords này được thu thập trên Internet. Sau khi chuẩn hoá dữ liệu, số lượng từ trong tập dữ liệu còn 37.395 từ (so với khoảng 81.773 từ khi chưa xử lý) Việc này giúp tăng độ chính xác của việc vector hoá sau và dự đoán này, vì nó chỉ chứa những từ khoá chính và quan trọng, giúp tập dữ liệu sạch và bớt nhiễu hơn.

đau lưng xảy bất_kỳ bất_kỳ độ tuổi hậu_quả chấn_thương song đa_phần lưng thói_quen âm_thầm hại lưng bốn bí_quyết lưng ủng_hộ chuyên giathay túi đeo vai ba_lô phân gánh trọng_lượng vai vai chăm vận động cơ_bắp lưng tư_thế cúi thứ eric robertson đại_học caliornia khác_biệt quan_trọng thỉnh_thoảng cúi chả sao tư_thế cúi tư_thế lâu tưởng_tượng thức dậy quay vâng quay sang trái cổ cảm_thấy cứng khó_chịu cố_gắng tư_thế cử_động khớp cơ thông_qua loạt chuyển_động tìm_hiểu bài_tập công_việc bàn_giấy động_tác gấp duỗi tay khác_biệt chỗ tư_thế ngủ nệm chuyên_gia nằm nghiêng ngủ hầu họng thông_thoáng cơ_chế_cơ_học đặt cột_sống tư_thế lý_tưởng lưng lưng nằm ngửa nằm tư_thế bào_thai nằm nệm sử dụng nệm mềm chắc thử sai tìm nệm thật tìm chiếc túi túi đeo vai sang chảnh mang thứ chuyển sang ba_lô quai đeo gánh trọng_lượng phân chúng vai vai mang túi lưng vai đau_đầu natalie lovit chuyên_gia vật_lý_trị_liệu thành_phố ne_york mang chiếc túi tăng_cường mất cân_bằng cơ_bắp vai cột_sống mất cân_bằng diễn chuỗi chẳng_hạn dáng chườm nóng chườm đá tự hỏi quy_tắc kiểu lưng hệ_quả chấn_thương chườm đá sưng lưng cứng cơ chườm ấm sức nóng thúc_đẩy lưu_thông máu khu_vực cứng khó_chịu nhức cảm_thấy dễ_dàng vận_động

Hình 3. Ví dụ về dữ liệu đầu vào sau khi đã được chuẩn hoá

IV. VECTOR HOÁ DỮ LIỆU

Để có thể sử dụng như đầu vào của các mô hình học máy, sau khi tiền xử lý, dữ liệu phải được chuẩn hoá thành dạng vector hoặc ma trận. Trong báo cáo này, tôi sử dụng mô hình chủ đề ẩn, để biểu diễn mỗi văn bản bằng một vector số thực n chiều, thể hiện xác suất văn bản đó được phân vào n chủ đề ẩn.

4.1. Mô hình chủ đề ẩn

Trong các bài toán về thu thập thông tin, học máy, khai phá dữ liệu, mô tả đặc tính của các tài liệu là một vấn đề rất quan trọng. Việc phân tích được nội dung của các tài liệu có thể làm tiền đề quan trọng cho việc tổ chức, phân loại, tìm kiếm chúng. Mô hình chủ đề ẩn [4] mô tả mỗi tài liệu chứa một xác suất phân bố các chủ đề, trong đó mỗi chủ đề lại là một phân bố của các từ. Tài liệu được biểu diễn dưới dạng phân bố xác suất có rất nhiều ưu điểm so với biểu diễn bằng vector các từ thông thường. Các mô hình chủ đề ẩn tiếp cận bằng cách xác định phân bố xác suất các chủ đề của tập tài liệu, cũng như phân bố xác suất các từ khoá của mỗi chủ đề đó. Sau đó với một tài liệu mới hoàn toàn, ta có thể dùng mô hình đã được xây dựng để suy diễn ra phân bố chủ đề của tài liệu mới đó.

Hai mô hình nổi tiếng về phân tích chủ đề ẩn là LDA (Latent Dirichlet Allocation) và pLSA (Probabilistic Latent Semantic Analysis). pLSA được phát triển dựa trên phương pháp LSA kết hợp với một mô hình xác suất, được biết đến như một kĩ thuật quan trọng để mô hình hoá dữ liệu dạng văn bản, nhưng nó vẫn tồn tại một số nhược điểm như chưa xây dựng được mô hình xác suất thực sự tốt ở mức tài liệu, do đó khi suy diễn phân bố cho một tài liệu hoàn toàn mới không có trong tập dữ liệu đào tạo chất lượng sẽ không được tốt. Mô hình LDA hoàn thiện hơn pLSA, khắc phục được các nhược điểm trên. Trong báo cáo này, tôi sử dụng mô hình LDA để phân tích các chủ đề ẩn của tài liệu.

4.2. Latent dirichlet allocation (LDA)

LDA là một mô hình sinh xác suất cho tập dữ liệu rời rạc, là phát triển của một vài mô hình như Naive Bayes, Hofman's aspect (pLSI) và được dùng để trích xuất các chủ đề ẩn của tài liệu, bằng cách coi một tài liệu là một tập phân bố xác suất của các chủ đề, và mỗi chủ đề là một phân bố xác suất của các từ mà không cần quan tâm đến thứ tự xuất hiện của chúng [4]. Dựa vào hướng tiếp cận này, LDA có thể trích xuất được đặc tính và chủ đề của các tài liệu, giải quyết được các vấn đề về trích chọn, khai phá thông tin trong tập dữ liệu ngày càng lớn như hiện nay. Mô hình được mô tả như sau:

Giả sử ta có một tập dữ liệu với tập từ vựng là V của M tài liệu: $D = \{d_1, d_2, \dots, d_M\}$. Tài liệu d gồm N_d từ w_i được lấy từ tập từ vựng: $T = \{t_1, t_2, \dots, t_V\}$. Trong quá trình ước lượng mô hình LDA, tài liệu đầu tiên $\overrightarrow{w_d} = \{w_{d,n}\}_{n=1}^{d}$ được tính bằng cách lấy một phân bố trên chủ đề $\overrightarrow{v_d}$ từ một phân bố Dirichlet $Diric(\vec{\alpha})$. Sau đó, mỗi chủ đề $z_{d,n}$ được tạo ra từ phân phối đa thức $Multi(\overrightarrow{v_d})$ và mỗi từ $w_{d,n}$ được tính bằng cách lấy mẫu từ phân phối đa thức $Multi(\overrightarrow{\varphi_{z_{d,n}}})$. Ta có giả mã như sau:

for mỗi tài liệu $d \in [1, M]$ **do**
    lấy mẫu tỉ lệ $\overrightarrow{v_d} \sim Diric(\vec{\alpha})$
    lấy mẫu độ dài tài liệu $N_d \sim Poiss(\xi)$
    for mỗi từ $n \in [1, N_d]$ **do**
        lấy mẫu chỉ số chủ đề $z_{d,n} \sim Multi(\overrightarrow{v_d})$
        lấy mẫu từ $w_{d,n} \sim Multi(\overrightarrow{\varphi_{z_{d,n}}})$
    end for
end for

Khi cho trước các tham số Dirichlet, ta có thể xác định phân phối đồng thời của cả những biến đã biết và biến ẩn như sau:

$$p(\overrightarrow{w_d}, \overrightarrow{z_d}, \overrightarrow{v_d}|\vec{\alpha}, \Phi) = \prod_{n=1}^{N_d} p\left(w_{d,n}\middle|\overrightarrow{\varphi_{z_{d,n}}}\right) p\left(z_{d,n}\middle|\overrightarrow{v_d}\right) p(\overrightarrow{v_d}, \vec{\alpha})$$

Xác suất của tài liệu $\overrightarrow{w_d}$ được tính bởi tích phân trên $\overrightarrow{v_d}$ trên miền $\overrightarrow{z_d}$ như sau:

$$p(\overrightarrow{w_d}|\vec{\alpha}, \Phi) = \int p(\overrightarrow{v_d}|\vec{\alpha}). \prod_{n=1}^{N_d} p(w_{d,n}|\overrightarrow{v_d}, \Phi)\, d(\overrightarrow{v_d})$$

Cuối cùng, xác suất của cả tập dữ liệu $W = \{\overrightarrow{w_d}\}_{d=1}^{M}$ là tích của tất cả các xác suất của các tài liệu:

$$p(W|\vec{\alpha}, \Phi) = \prod_{d=1}^{M} p(\overrightarrow{w_d}|\vec{\alpha}, \Phi)$$

Trong đó:

- α, β : hyperparameters.
- $\overrightarrow{v_d}$: phân phối của chủ đề trong tài liệu thứ *d* (tham số cấp độ tài liệu).
- $z_{d,n}$: chỉ số chủ đề (từ thứ *n* của tài liệu *d*).
- $w_{d,n}$: từ thứ *n* của văn bản *d* chỉ bởi $z_{d,n}$ (biến độ cấp độ từ).
- $\overrightarrow{\varphi_k}$: phân phối của các từ được sinh ra từ chủ đề $z_{d,n}$. $\overrightarrow{\varphi_k}$ biểu diễn tham số cho $p(t|z = k)$, thành phần trộn của chủ đề *k*.
- M : số lượng tài liệu.
- N_d : số lượng các từ trong tài liệu thứ *d* (độ dài văn bản).
- K : số lượng các chủ đề ẩn.
- *Diric* và *Multi* lần lượt là các phân phối Dirichlet và phân phối đa thức.

Với một tập hợp các tài liệu, ta có thể tính toán được phân phối tài liệu - chủ đề, và chủ đề - từ khoá của mô hình LDA. Đây là quá trình huấn luyện mô hình. Vector phân bố xác suất có thể được sử dụng như một đặc trưng để biểu diễn một tài liệu. Cách biểu diễn này có kích thước nhỏ hơn rất nhiều so với tài liệu gốc và cũng rất tiện lợi để làm bước tiền xử lý cho các bài toán khác.

Trong việc phân loại tài liệu văn bản, ta có thể áp dụng LDA để huấn luyện mô hình dựa trên một tập dữ liệu mẫu và sau đó sử dụng mô hình này để suy diễn ra phân bố xác suất của một tài liệu mới. Sau khi có sự phân bố xác suất của các chủ đề ẩn đó, nó sẽ được sử dụng như 1 biểu diễn vector của dữ liệu gốc, được sử dụng như đầu vào cho các thuật toán phân loại sau này.

4.2.1. Ước lượng mô hình LDA với lấy mẫu Gibbs

Lấy mẫu Gibbs là một trường hợp đặc biệt của chuỗi Markov Monte Carlo (MCMC) [5]. Phương pháp này có thuật toán suy diễn xấp xỉ khá đơn giản. Giả sử $\vec{w}$ là vector của tất cả các từ trong tài liệu và $\vec{z}$ là vector các chủ đề trong tập dữ liệu W. Ta có thể ước lượng của Φ và Θ bằng cách sử dụng phân phối hậu nghiệm $p(\vec{z}|\vec{w})$. Mô hình xác suất hoàn chỉnh như sau:

$$w_i|z_{i,}\Phi^{z_i}\sim Multi(\Phi^{z_i})$$

$$\Phi\sim Diric(\beta)$$

$$z_i|\Theta^{d_i}\sim Multi(\Theta^{d_i})$$

$$\Phi^{d_i}\sim Diric(\alpha)$$

Áp dụng mô hình trên, phân bố chủ đề cho một từ bất kì có thể tính được dựa trên phân bố chủ đề của các từ khác. Phân phối chủ đề của một từ t được lấy mẫu từ phân phối đa thức sau:

$$p(z_i = k|\overrightarrow{z_{\neg\iota}}), \vec{w} = \frac{n_{k,\neg i}^{(t)} + \beta_t}{\left[\sum_{v=1}^{V} n_k^{(v)} + \beta_v\right] - 1} = \frac{n_{d,\neg i}^{(k)} + \alpha_k}{\left[\sum_{j=1}^{K} n_{\tilde{d}}^{(j)} + \alpha_j\right] - 1}$$

Trong đó t là số lần từ được phân phối cho chủ đề k.

$\sum_{v=1}^{V} n_k^{(v)} - 1$ là số từ được phân phối cho chủ đề k trừ phân phối hiện thời.

$n_{d,\neg i}^{(k)}$ là số từ trong tài liệu d được phân phối cho chủ đề k.

$\sum_{j=1}^{K} n_{\tilde{d}}^{(j)} - 1$ là tổng số từ trong tài liệu d không kể từ t

Thông thường, các biến $\vec{\alpha}, \vec{\beta}$ là đều, mọi α_k là giống nhau, tương tự với $\vec{\beta}$.

Sau quá trình lấy mẫu Gibbs, ma trận Φ và Θ được tính như sau:

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{v=1}^{V} n_k^{(v)} + \beta_v}$$

$$\vartheta_{d,k} = \frac{n_d^{(k)} + \alpha_k}{\sum_{j=1}^{K} n_d^{(j)} + \alpha_j}$$

4.2.2. Suy diễn chủ đề theo mô hình LDA

Sau khi có mô hình LDA đã được ước lượng, ta dùng nó để suy diễn phân bố chủ đề của các tài liệu mới. Giả sử một tài liệu mới $\tilde{d}$ được coi như một vector của các từ $\widetilde{\overrightarrow{w_d}}$. Ta cần suy diễn phân phối hậu nghiệm của cả các chủ đề $\tilde{\vec{z}}$ với vector các từ $\widetilde{\overrightarrow{w_d}}$ và mô hình LDA:

$$L(\Theta, \Phi): p(\tilde{\vec{z}}|\tilde{\vec{d}}, L) = p(\tilde{\vec{z}}, \tilde{\vec{d}}, \vec{d}, \vec{z})$$

Mẫu Gibbs được cập nhật như sau:

$$p\left(\vec{z}_i = k \middle| \widetilde{\overrightarrow{z_{\neg l}}}, \widetilde{\vec{w}}\right) = \frac{n_k^{(t)} + \tilde{n}_{k,\neg i}^{(t)} + \beta_t}{\left[\sum_{v=1}^{V} n_k^{(v)} + \tilde{n}_k^{(v)} + \beta_v\right] - 1} = \frac{n_{d,\neg i}^{(k)} + \alpha_k}{\left[\sum_{z=1}^{K} n_{\tilde{d}}^{(z)} + \alpha_z\right] - 1}$$

Trong đó $\tilde{n}_k^{(t)}$ là số thể hiện của *t* và chủ đề *k* trong tài liệu mới. Sau khi lấy mẫu chủ đề, phân bố chủ đề của $\tilde{d}$ là $\widetilde{v_{\tilde{d}}} = \{v_{\tilde{d},1}, v_{\tilde{d},2}, \ldots, v_{\tilde{d},k}\}$ trong đó:

$$v_{\tilde{d},k} = \frac{n_{\tilde{d}}^{(k)} + \alpha_k}{\sum_{z=1}^{K} n_{\tilde{d}}^{(z)} + \alpha_z}$$

4.2.3. GibbsLDA++

Trong báo cáo này, để áp dụng mô hình LDA, tôi sử dụng GibbsLDA++[3] [6], một công cụ mã nguồn mở được viết trên C++ và Java, áp dụng mô hình Latent Dirichlet Allocation với lấy mẫu Gibbs cho việc ước lượng tham số và suy diễn. GibbsLDA++ được dùng để trích xuất các chủ đề ẩn từ tập dữ liệu đã được thu thập và làm sạch. Công cụ này khá nhanh và được thiết kế để xử lý các chủ đề ẩn của các bộ dữ liệu lớn. Nó bao gồm 2 phần chính:

[3] "GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation" - Xuan-Hieu Phan, Cam-Tu Nguyen <http://gibbslda.sourceforge.net/>

- Ước lượng mô hình LDA: sinh ra một mô hình LDA từ tập dữ liệu đào tạo, cung cấp các tuỳ chọn để lưu lại model ở các bước cụ thể, và có thể tiếp tục ước lượng, suy diễn từ bước hiện tại.
- Suy diễn phân bố chủ đề cho một tài liệu mới: Sử dụng mô hình LDA đã được tạo ra ở trên để suy diễn sự phân bố chủ đề của một tài liệu mới.

GibbsLDA++ được biên dịch với các tham số sau:

- -est: Ước lượng LDA model.
- -alpha <double>: Giá trị alpha, hyper-parameter của LDA. Mặc định của alpha là 50 / K (K là số lượng chủ đề).
- -beta<double>: Giá trị beta, hyper-parameter của LDA. Mặc định của beta là 0.1
- -ntopics<int>: Số lượng chủ đề, phụ thuộc vào tập dữ liệu đầu vào, giá trị mặc định là 100.
- -niters<int>: Số vòng lặp của Gibbs sampling. Giá trị mặc định là 2000.
- -savestep<int>: Số bước để lưu mô hình LDA vào đĩa cứng. Giá trị mặc định là 200.
- -twords <int>: Số lượng từ khoá lưu lại ở mỗi chủ đề có thể thuộc vào từng chủ đề, giá trị mặc định là 20. Ví dụ với –twords = 20, GibbsLDA++ sẽ in ra 20 từ gần sát nhất với mỗi chủ đề khác nhau khi LDA model được lưu lại.
- -dfile <string>: Đường dẫn đến tập dữ liệu training.

Trong hệ thống, các tham số được sử dụng là:

- alpha = 0.5
- beta = 0.1
- ntopics = 100
- niters = 2000
- savestep = 200

- twords = 20

và được huấn luyện với hơn 5.300 bài báo. Sau quá trình huấn luyện, 100 chủ đề ẩn được tạo ra, mô hình LDA được lưu lại tại bước cuối cùng. Lưu ý rằng, các chủ đề ẩn này chỉ được đánh số thứ tự mà không được gán nhãn trước. Ví dụ chủ đề thứ 1 có các từ khoá chính như "thuế, lệ_phí, tài_chính", nó có thể được hiểu là 1 chủ đề về kinh doanh hoặc kinh tế, nhưng chúng ta không cần phải gán nhãn cụ thể. Trong thực tế, số lượng chủ đề ẩn có thể được chọn rất lớn, và các từ khoá trong đó cũng khá đa dạng, vì vậy rất khó để gán nhãn chính xác.

Chủ đề 0	Chủ đề 1	Chủ đề 2
trường 0.032914281573799	thuế 0.10873316209486086	đất 0.05322655134772021
chương_trình 0.018664106465715046	tính 0.020989784514029136	đà_nng 0.04252036275086069
tổ_chức 0.017939521290727727	thu 0.018342777592596223	tp 0.023207238223192542
học_sinh 0.016369586744921866	phí 0.01736240465873218	đà 0.01974347132420858
tham_gia 0.015705383667850157	lệ_phí 0.01432324856375365	nng 0.013655638592661012
hội 0.014618505905369176	tài_chính 0.013931099390208033	ubnd 0.013550675959358468
học 0.013531628142888195	ô_tô 0.01246053998941197	quyết_định 0.013550675959358468
thi 0.011055962128348187	nộp 0.012362502696025566	tiền 0.013025862792845745
hoạt_động 0.010391759051276477	châu_âu 0.009715495774592653	đấu_giá 0.012815937526240656
xã_hội 0.009063352897133057	chịu 0.008833160134115016	nộp 0.012815937526240656
cộng_đồng 0.008761442407555007	quy_định 0.008539048253955803	văn_phòng 0.011241498026702492
chạy 0.008097239330483298	trở 0.007460638026705358	hoài 0.009667058527164329
nghề 0.007372654155495978	miễn 0.007362600733318954	hủy 0.00872239482744143
giáo_dục 0.007191507861749148	đóng 0.007166526146546145	sở 0.00851246956083634
giải 0.007191507861749148	hưởng 0.005794004039136488	chậm 0.008302544294231252
tặng 0.007010361568002318	bổ_sung 0.005794004039136488	la 0.00798765639432362
đào_tạo 0.006466922686761828	eu 0.0056959667457500834	hộ 0.007882693761021076
lớp 0.006346158490930608	ưu_đãi 0.005401854865590871	công_chứng 0.007462843227810898
quỹ 0.006225394295099388	đăng_ký 0.005303817572204467	vipico 0.006728104794693088
sinh_viên 0.006044248001352558	nghị_định 0.005009705692045254	dân 0.006518179528088

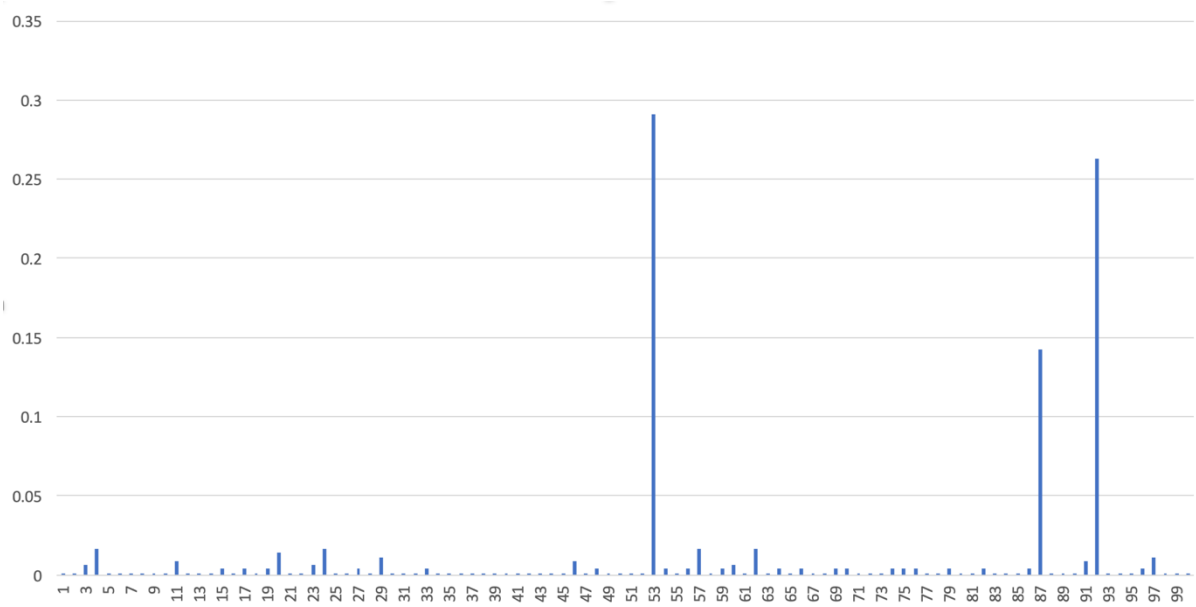
Hình 4. Một số chủ đề ẩn thu được sau khi sử dụng GibbsLDA++

Sau khi áp dụng LDA, một số chủ đề ẩn thu được được mô tả như hình trên. Mỗi chủ đề được biểu diễn bởi một phân bố các từ khoá. Từ mô hình LDA đã được sinh ra, phân bố chủ đề của một tài liệu mới có thể được suy diễn bằng GibbsLDA++:

```
lda -inf -dir<string> -model<string> [-niters<int>] [-twords<int>] -dfile<string>
```

Trong đó:

- *-dir*: đường dẫn tới thư mục chứa mô hình LDA.
- *-model*: tên mô hình LDA.
- *-niters*: số vòng lặp.
- *-twords*: số từ khoá lưu lại.
- *-dfile*: đường dẫn tới file chứa nội dung mới cần suy diễn.



Hình 5. Phân bố chủ đề của 1 bài báo

Hình trên là phân bố chủ đề của bài báo "Cơ thể gây ám ảnh của cô gái giảm 141kg trong hơn 1 năm" (https://dantri.com.vn/suc-khoe/co-the-gay-am-anh-cua-co-gai-giam-141kg-trong-hon-1-nam-20181109072920159.htm). Có thể thấy được, bài báo này chủ yếu rơi vào chủ đề 53, 87, và 92. Tuy không biết rõ nội dung các chủ đề này là gì, nhưng với các bài báo khác nhau, chúng được đánh số giống nhau, vì thế sẽ không ảnh hưởng đến việc phân loại sau này.

Trước khi xử lý, dữ liệu là một đoạn văn bản gồm nhiều từ tiếng Việt. Sau khi xử lý qua 3 bước trên, dữ liệu đã được chuẩn hoá, mỗi bài báo được biểu diễn bằng một vector 100 chiều.

V. THỰC NGHIỆM

Mã nguồn của chương trình có thể tải về tại: https://github.com/conmagan/Data-Mining. Mã nguồn được viết trên Java, gồm 3 phần chính: thu thập dữ liệu, tiền xử lý dữ liệu và vector hoá dữ liệu.

5.1. Thu thập dữ liệu

Để thực hiện việc thu thập dữ liệu, gọi hàm `Crawler.crawl("phap-luat")` trong đó "phap-luat" là chủ đề cần thu thập.

```
public class Crawler extends BaseCrawler {
    private static String getURL(String category) {
        return "http://dantri.com.vn/" + category;
    }

    static void crawl(String category) {
        mState = new State();
        int pageIndex = mState.getCurrentPage() + 1;
        while (true) {
            String URL = getURL(category) + "/trang-" + pageIndex +
".htm";
            Logs.infoLn("Crawling page: " + pageIndex + " " + URL);
            Document d = Utils.getDoc(URL);
            if (d == null) return;
            Elements e = d.getElementsByClass("mr1");
            if (e.toString().equals("")) break;
            for (int j = mState.getCurrentTopic() + 1; j < e.size();
j++) {
                int jj = j + 1;
                String                    link                    =
e.get(j).select("a").attr("abs:href");
                Logs.infoLn("Topic: " + jj + "/" + e.size() + " " +
link);
                Document topicDoc = Utils.getDoc(link);
                String                   title                    =
topicDoc.getElementsByClass("mgb15").text();
                Element                    e1                     =
topicDoc.getElementsByClass("sapo").first();
                String content1 = "";
                String content2 = "";
                if (e1 != null) {
                    e1.select("a").remove();
                    content1 = e1.text();
                }
```

```
                Element                    e2                    =
topicDoc.getElementById("divNewsContent");
                if (e2 != null && e2.select("div.news-tag") != null)
{
                    e2.select("div.news-tag").remove();
                    content2 = e2.text();
                }
                title = title.replace("/", " ");
                FileUtils.write("data/" + category + "/raw/" + title
+ ".txt", content1 + content2);
                FileUtils.write("data/" + category + "/normalized/"
+ title + ".txt",
                        Normalizer.normalizeContent(content1     +
content2));
            }
            mState.setCurrentPage(pageIndex);
            mState.setCurrentTopic(-1);
            pageIndex++;
        }
    }
}
```

Hình 6. Một đoạn mã nguồn để thu thập dữ liệu từ Dân Trí

5.2. Tiền xử lý dữ liệu

Để tiền tách từ và loại bỏ stopwords, chạy hàm: `Normalizer.normalizeContent(sentence)` trong đó sentence là đoạn văn bản cần được chuẩn hoá.

```
public class Normalizer {
    public static VnTextPro vnTextPro = null;
    public static Map<Character, Boolean> charMap = new HashMap<>();
    public static HashMap<String, Boolean> stopwords;

    static {
        initVnTextPro();
        initNormalized();
    }

    public static String segment(String content) {
        String                    segmentedStr                    =
vnTextPro.segmentText(content.trim().toLowerCase()).toString();
        String       result       =       segmentedStr.substring(1,
segmentedStr.length() - 1);
        return result;
    }
    public static String normalize(String content) {
        String normalizeStr = content.replaceAll("[0-9]", "");
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < normalizeStr.length(); i++) {
```

```
            char ch = normalizeStr.charAt(i);
            if (isValidChar(ch)) {
                sb.append(ch);
            }
        }
        StringBuilder result = new StringBuilder();
        String[] arr = sb.toString().trim().split(" ");
        for (int i = 0; i < arr.length; i++) {
            String str = arr[i];
            if (stopwords.get(str) == null && str.length() > 1) {
                result.append(str);
                result.append(" ");
            }
        }
        return result.toString().trim();
    }
    private static boolean isValidChar(char s) {
        return charMap.get(s) != null;
    }

    public static void initVnTextPro() {
        vnTextPro = new VnTextPro(true, true, true);
        vnTextPro.init();
    }

    public static void initNormalized() {
        Logs.infoLn("Initializing for text normalization...");
        stopwords = new HashMap<>();
        getStopwords(Configs.STOPWORDS_PATH);
        for (int i = 0; i < Configs.VALID_CHARS.length(); i++) {
            charMap.put(Configs.VALID_CHARS.charAt(i), true);
        }
    }
    public static void getStopwords(String path) {
        BufferedReader br;
        try {
            br = new BufferedReader(new FileReader(path));
            String w = br.readLine();
            while (w != null) {
                stopwords.put(w.trim(), true);
                w = br.readLine();
            }
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public static String normalizeContent(String s) {
        String segmented = segment(s);
        return normalize(segmented);
    }
}
```

Hình 7. Một đoạn mã nguồn thực hiện tách từ và loại bỏ stopwords

5.3. Vector hoá dữ liệu

Hàm `vectorize()` thực thi việc vector hoá cho tất cả các bài báo trong tập dữ liệu đã thu thập, sử dụng mô hình LDA đã được huấn luyện.

```
public class Vectorization {
    public static TopicLabeler topicLabeler = null;

    static {
        initTopicLabeler();
    }
    public static void initTopicLabeler() {
        topicLabeler = new TopicLabeler(Configs.LDAMODEL_DIRECTORY,
Configs.LDAMODEL_NAME);
        topicLabeler.init();
    }
    public static String infTopicsDistribution(String content, int
iterations) {
        ArrayList<String> documents = new ArrayList<>();
        documents.add(content);
        List<List<Double>>                  ft                  =
topicLabeler.doInferenceAndGetTopicDistributionsOfDocuments(docume
nts, iterations);
        for (List<Double> lft : ft) {
            StringBuilder sb = new StringBuilder();
            for (int k = 0; k < lft.size(); k++) {
                sb.append(lft.get(k)).append(",");
            }
            return sb.toString().trim();
        }
        return null;
    }
    public static void vectorize() {
        Logs.infoLn("Generating training data ...");
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < Configs.categories.length; i++) {
            String c = Configs.categories[i];
            String dir = "data/" + c + "/normalized";
            int finalI = i;
            FileUtils.listFiles(dir, s -> {
                if (!s.startsWith(".")) {
                    String content = FileUtils.readToString(dir +
"/" + s);
                    String topics = infTopicsDistribution(content,
Configs.LDA_ITERS);
                    sb.append(topics).append(finalI).append("\n");
                }
            });
        }
        FileUtils.write(Configs.TRAIN, sb.toString());
    }
}
```

Hình 8. Một đoạn mã nguồn thực hiện vector hoá dữ liệu sử dụng LDA

V. TỔNG KẾT

6.1. Kết quả đạt được

Trong báo cáo này, tôi đã thực hiện một phương pháp thu thập dữ liệu, tiền xử lý và vector hoá dữ liệu cho bài toán tự động phân loại tin tức văn bản cho tiếng Việt. Một số kết quả đạt được như sau:

- Các bài báo được thu thập tự động theo 5 chủ đề Sức khoẻ (1038), Kinh doanh (1065), Pháp luật (1042), Thể thao (1102), và Xe (1056) trên trang báo điện tử Dân Trí (http://dantri.com.vn). Tổng số lượng thu thập được: 5303 bài.
- Sau đó, dữ liệu được xử lý qua 3 bước:
    1. Tách từ tiếng Việt.
    2. Loại bỏ các từ dừng (stopwords) không mang nhiều ý nghĩa.
    3. Sử dụng mô hình chủ đề ẩn để huấn luyện một mô hình nhằm ước lượng phân bố xác suất của mỗi bài báo theo các chủ đề ẩn. Từ đó biểu diễn mỗi bài báo bằng 1 vector số thực 100 chiều, thể hiện phân bố chủ đề của văn bản đó trên 100 chủ đề ẩn. Dữ liệu này sau đó sẽ được sử dụng như đầu vào cho bài toán phân loại văn bản.

6.2. Định hướng tiếp theo

Trên thực tế, số lượng bài báo nên được thu thập nhiều hơn nữa, tuy nhiên thời gian xử lý và huấn luyện mô hình học máy sẽ lâu hơn. Trong tương lai, tôi sẽ tiến hành thu thập nhiều bài báo từ nhiều chủ đề khác nhau và các trang báo điện tử khác để làm dữ liệu đa dạng hơn.

Tiếp đó, dựa trên dữ liệu đã được thu thập và chuẩn hoá, một hệ thống phân loại văn bản tiếng Việt tự động sẽ được đề xuất và thực hiện, dựa trên các phương pháp học máy tiên tiến.

TÀI LIỆU THAM KHẢO

1. Maks Ovsjanikov, Ye Chen: Topic Modeling for Personalized Recommendation of Volatile Items.
2. Xuan-Hieu Phan, Cam-Tu Nguyen: JvnTextPro – A Java-based Vietnamese Text Processing Tool - http://jvntextpro.sourceforge.net.
3. Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001)
4. D. Blei, A. Ng, and M. Jordan: Latent Dirichlet Allocation, Journal of Machine Learning Research (2003).
5. C. Andrieu, N.D. Freitas, A. Doucet, and M. Jordan: An introduction to MCMC for machine learning, Machine Learning (2003)
6. Xuan-Hieu Phan, Cam-Tu Nguyen, "GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation"- http://gibbslda.sourceforge.net.