Ứng dụng PageRank vào bài toán trích lược văn bản tiếng Việt

Bùi Minh Nhật
email: minhnhat@linuxmail.org

1 Bài toán trích lược văn bản

Cùng với sự phát triển của internet, lượng dữ liệu sinh ra ngày càng nhiều. Khối lượng thông tin chủ yếu và chiếm trên 80% là các thông tin văn bản. Thật khó để con người có thể phân tích, khai thác khối ngữ liệu đồ sộ này một cách thủ công. Do vậy, chúng ta cần các phương pháp rút gọn, tóm tắt văn bản để tiết kiệm thời gian và công sức. Cho đến nay, có 3 hướng tóm tắt chính:

Trích lược: sử dụng các câu có sẵn trong văn bản để tạo nên bản tóm tắt.

Tóm lược: tổng hợp các thông tin trích ra từ văn bản gốc, bản tóm tắt có thể chứa nhiều từ không xuất hiện trong văn bản gốc.

Nén câu: tỉnh lược bớt từ/cụm từ trong câu nhưng vẫn giữ đúng cấu trúc ngữ pháp.

Hướng trích lược văn bản phổ biến hơn cả vì nó đơn giản hơn hai phương pháp kia. Phần lớn các bài toán trích lược sử dụng thuật toán gom cụm, những câu gần giống nhau được gom về một cụm, sau đó sử dụng một câu trong cụm để đại diện cho cả cụm. Bài viết này tiếp cận trích lược theo cách xếp hạng câu, điểm quan trọng của một câu phụ thuộc vào điểm của các câu nó liên kết tới.

2 PageRank

Xin được trích lại và dịch một phần từ *2.1.1* của [1]:

> *Chúng tôi giả sử một trang web A có các trang $T_1...T_n$ trỏ đến nó. Tham số d gọi là hệ số giảm ảnh hưởng (damping factor) có thể được đặt trong khoảng $(0, 1)$. Chúng tôi thường chọn $d = 0.85$. $C(A)$ được định nghĩa là số lượng link mà A trỏ đến. PageRank của A được cho như sau:*
>
> $$PR(A) = (1 - d) + d\left(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)}\right)$$
>
> *Lưu ý rằng các PageRank tạo thành một dạng phân phối xác xuất khắp các trang web, do đó tổng của tất cả PageRank bằng một.*

Thoạt nhìn, có vẻ không thể tính được $PR(A)$, vì cần PR của tất cả các trang trỏ đến nó, và các trang này cũng tính theo cách tương tự. Tuy nhiên, $(1-d) = 0.15$ nếu không có trang web nào trỏ đến A, chúng ta có thể lợi dụng điều này để tính, bằng cách lặp lại một số lần nào đó để PR hội tụ.

3 Xếp hạng câu

Mỗi câu trong văn bản được xem là một trang web, biểu diễn bằng một tập hợp các từ có trong câu đó. Nếu câu S_i chứa một từ W_k trong câu S_j thì câu S_i trỏ đến S_j và S_j cũng đồng thời trỏ đến S_i. Đây là điểm khác biệt giữa trang web và câu. Độ tương đồng của hai câu S_i và S_j được xác định bởi:

$$Similarity(S_i, S_j) = \frac{|S_i \cap S_j|}{\log_2(|S_i|) + \log_2(|S_j|)}$$

Với $\log_2$ làm hệ số chuẩn hóa, tránh việc câu dài có quá nhiều ưu thế trong xếp hạng, vì nó chứa nhiều từ không đồng nghĩa nó chứa nhiều thông tin quan trọng.

Tạo đồ thị

Gọi các câu là đỉnh của đồ thị, n là số câu có trong văn bản, $G \in \mathbb{R}^{n \times n}$ là ma trận tương đồng của các câu.

Hai câu S_i và S_j được gọi là có liên kết nếu $|S_i \cap S_j| > 0$. Các đỉnh liên kết với S_i được định nghĩa là:

$$S_i.conn\_nodes = \{S_j : |S_i \cap S_j| > 0\} \quad \forall\, 0 < j \leq n, j \neq i$$

Trọng số của cạnh nối hai đỉnh S_i và S_j là:

$$G_{i,j} = Similarity(S_i, S_j) \quad \forall\, 0 < i, j \leq n$$

Trọng số của đỉnh S_i là:

$$S_i.weight = \left(\sum_{k=1}^{n} G_{i,k}\right) - G_{i,i}$$

Tính vế trái của ba phương trình trên với tất cả các đỉnh và cạnh ta được đồ thị hoàn chỉnh để cài đặt thuật toán. Lưu ý rằng khi cài đặt, tập hợp được hiểu là mảng một chiều.

Thuật toán xếp hạng

1. $d \leftarrow 0.85$
2. $W \leftarrow [1..n]$ *// PR cuối cùng*
3. for $i \leftarrow 1$ to n:
   . $W[i] \leftarrow S[i]$

4. for $time \leftarrow 1$ to 20: *// lặp để hội tụ*
   . $PR \leftarrow [1..n]$ *// $PR[i]$ là $PR(S_i)$*
   . for $i \leftarrow 1$ to n:
   . . $PR[i] \leftarrow (1 - d)$

   . for $i \leftarrow 1$ to n:
   . . for $j \leftarrow 1$ to n:
   . . . if $S[j] \in S[i].conn\_nodes$:
   . . . . $PR[i] \leftarrow PR[i] + d \cdot W[j]$ / $length(S[j].conn\_nodes)$

   . if $time < 20$:
   . . for $i \leftarrow 1$ to n:
   . . . $W[i] \leftarrow PR[i]$
5. return W

Chọn các câu có PR cao nhất, sau đó sắp xếp thứ tự của chúng như trong văn bản gốc, ta được bản trích lược.

Tài liệu

[1] Sergey Brin & Lawrance Page, *The anatomy of a large-scale hypertextual Web search engine.* Computer Networks and ISDN Systems 30, pp. 107-117, 1998.