Tạp chí Tin học và Điều khiển học, T.30, S.1 (2014), 39–48

CÁCH TIẾP CẬN DỊCH MÁY THÔNG KÊ DỰA TRÊN CÚ PHÁP GIẢI BÀI TOÁN TỰ ĐỘNG KHÔI PHỤC DẤU CHO VĂN BẢN

NGUYỄN MINH HẢI, NGUYỄN MINH TUẤN

Học viện Công nghệ Bưu chính - Viễn thông; haihth2004; nmtuan@yahoo.com

Tóm tắt. Trong bài báo này việc tự động hóa khôi phục dấu cho văn bản được mô hình hóa như một bài toán dịch máy thông kê dựa trên cú pháp với đầu vào là các văn bản không dấu và đầu ra là các văn bản có dấu của cùng một ngôn ngữ. Kỹ thuật suy diễn văn phạm ABL trong [2] được mở rộng để xây dựng văn phạm phi ngữ cảnh đồng bộ xác suất từ ngữ liệu chỉ chứa các câu phẳng (plain text) có dấu. Việc khôi phục dấu cho văn bản chính là việc phân tích cú pháp cho các câu của văn bản bằng phiên bản xác suất của thuật toán phân tích cú pháp CKY trên văn phạm nhận được. Phương pháp được thử nghiệm trên tiếng Việt và cho kết quả tốt. Do tính độc lập ngôn ngữ cao nên hệ thống có thể áp dụng cho các ngôn ngữ khác.

Từ khóa. Khôi phục dấu tự động, dịch máy dựa trên cú pháp, suy diễn văn phạm, văn phạm phi ngữ cảnh đồng bộ, thuật toán phân tích cú pháp CKY.

Abstract. In this paper, the automatic diacritization of a language is modeled as a statistical syntax-based machine translation problem with the source undiacritized text and the target diacritized text of the same languaget. The grammatical inference technique ABL proposed in [2] is extended for learning a probabilistic synchronous context-free grammar from training corpus containing plain diacritized sentences only. The diacritization is to parse input sentences by the probabilistic CKY parsing algorithm for received grammar. This method is applied to Vietnamese with high quality result. As language independent building way, it can be applied to the other languages.

Key words. Automatic diacritization, syntax-based machine translation, grammatical inference, synchronous context-free grammar, CKY parsing algorithm.

1. GIỚI THIỆU

Trên thế giới có rất nhiều ngôn ngữ có sử dụng dấu trong hệ thống chính tả [9]. Đối với một số ngôn ngữ, do nhiều nguyên nhân (lịch sử, mã hóa, công cụ soạn thảo, hiệu quả công việc...) nên nhiều tài liệu thường được lưu trữ dưới dạng không dấu. Các văn bản không dấu không chỉ gây nên nhầm lẫn cho con người (phát âm, ngữ nghĩa, chức năng...) mà việc loại bỏ dấu như vậy sẽ làm mất mát nhiều thông tin về từ vựng, hình thái, ngữ âm... rất cần thiết trong nhiều lĩnh vực ứng dụng công nghệ ngôn ngữ. Bởi vậy, việc khôi phục dấu cho các văn bản không dấu sẽ mang lại nhiều giá trị trong việc xây dựng ngữ liệu ngôn ngữ nói riêng và trong công nghệ ngôn ngữ nói chung.

Đã có nhiều đề xuất các phương pháp tự động khôi phục dấu cho văn bản không dấu của các ngôn ngữ khác nhau có sử dụng dấu trong hệ thống chính tả [3–9]. Nhưng cho đến nay

các phương pháp đã được đề xuất đều có nhược điểm chung là chỉ sử dụng thông tin cục bộ mà bỏ qua các mối phụ thuộc mang tính toàn cục ràng buộc sự đồng xuất hiện của các từ cũng như từ loại của chúng trong câu ở những khoảng cách xa. Ví dụ, ta xét một đoạn văn bản không dấu trong tiếng Việt

"Cho me cho cho con canh cong cho". (1)

Trong câu đó có 4 âm tiết không dấu "cho" xuất hiện ở các vị trí khác nhau. Âm tiết không dấu này có thể ứng với nhiều âm tiết/từ có dấu của tiếng Việt. Ta liệt kê một số biến thể khác nhau của nó như: "cho" (động từ), "chó" danh từ chỉ động vật, động từ "chờ", danh từ "chợ"...

Quay trở lại với câu (1), ta có thể có các câu tiếng Việt có dấu tương ứng như sau

"Chó mẹ chờ chó con cạnh cổng chợ". (2)

Nếu biết câu (1) có cấu trúc ngữ pháp: ⟨subj⟩ chờ ⟨obj⟩ ⟨adv⟩ với ⟨subj⟩="Chó mẹ", ⟨obj⟩="chó con" và ⟨adv⟩="cạnh cổng chợ"

"Chó mẹ cho chó con canh cổng chợ" (3)

nếu biết câu (1) có cấu trúc ngữ pháp: ⟨subj⟩ cho ⟨obj⟩ ⟨verb phrase⟩ với ⟨subj⟩="Chó mẹ", ⟨obj⟩=⟨chó con⟩ và ⟨verb phrase⟩="canh cổng chợ".

Các câu (2) và (3) cho thấy việc thêm dấu cho các âm tiết 3 ("cho") và 6 ("canh") cũng như ngữ nghĩa của câu phụ thuộc vào cấu trúc cú pháp nào được áp vào câu (1). Đây cũng chính là vấn đề đặt ra để giải quyết trong bài này.

Phần tiếp theo của bài báo được cấu trúc như sau: Mục 2 trình bày một số khái niệm cơ bản về văn phạm phi ngữ cảnh xác suất, văn phạm phi ngữ cảnh đồng bộ xác suất cũng như mô hình cơ sở của một hệ dịch máy thống kê dựa trên cú pháp; Mục 3 trình bày bài toán tự động khôi phục dấu văn bản cho các ngôn ngữ có sử dụng dấu trong hệ thống chính tả và đề xuất mô hình hệ thống tự động khôi phục dấu tổng quát bằng cách tiếp cận dịch máy thống kê dựa trên cú pháp; Mục 4 trình bày vấn đề cài đặt và thử nghiệm hệ thống trên các văn bản tiêng Việt và Mục 5 là một vài kết luận.

2. MÔ HÌNH HỆ DỊCH MÁY THỐNG KÊ DỰA TRÊN CÚ PHÁP

Trong mục này sẽ đưa ra một số khái niệm cơ bản được sử dụng trong lý thuyết dịch máy thống kê dựa trên cú pháp và mô hình cơ sở của một bộ dịch.

2.1. Một số khái niệm cơ bản

Định nghĩa 1. (PCFG) Văn phạm phi ngữ cảnh xác suất là một bộ 4

$$G = (N, S, T, R)$$

trong đó N là tập các ký hiệu không kết thúc của văn phạm, $S \in N$ là ký hiệu khởi đầu, T là tập từ vựng (hay các ký hiệu kết thúc), R là tập các quy tắc sản xuất của văm phạm.

Mỗi quy tắc sản xuất của R có dạng

$$X \to \langle \alpha, \omega \rangle \tag{4}$$

trong đó $X \in N$ là một ký hiệu không kết thúc, $\alpha \in (N \cup T)^*$, ω là xác suất áp dụng của quy tắc và thỏa mãn với mỗi X

$$\sum_{\alpha: X \to \langle \alpha, \omega \rangle \in R} \omega = 1.$$

Vế phải của (4) được gọi là một dạng câu.

Định nghĩa 2. Giả sử r là một quy tắc sản xuất $X_r \to \langle \alpha_r, \omega_r \rangle$. Khi đó kết quả của việc áp dụng quy tắc r vào một dạng câu $\langle \alpha X_r \beta, \omega \rangle$ sẽ là dạng câu $\langle \alpha\alpha_r\beta, \omega\omega_r \rangle$ và ta viết

$$\langle \alpha X_r \beta, \omega \rangle \xrightarrow{r} \langle \alpha\alpha_r\beta, \omega\omega_r \rangle.$$

Một cách tổng quát nếu ta áp dụng liên tiếp như trên một số hữu hạn hoặc đếm được các quy tắc vào một dạng câu $\langle \alpha, \omega \rangle$ để được một dạng câu mới $\langle \alpha_1, \omega_1 \rangle$, ta sẽ viết

$$\langle \alpha, \omega \rangle \xrightarrow{*} \langle \alpha_1, \omega_1 \rangle.$$

Định nghĩa 3. Một dạng câu được dẫn xuất từ dạng câu ban đầu $\langle S, 1 \rangle$ và không chứa ký hiệu kết thúc được gọi là một câu được sinh ra bởi văn phạm. Ngôn ngữ của văn phạm phi ngữ cảnh xác suất G, ký hiệu $L(G)$, là tập tất cả các câu được sinh ra bởi văn phạm G

$$L(G) = \{\langle \xi, w \rangle \in T^* \times [0,1] | \langle S, 1 \rangle \xrightarrow{*} \langle \xi, w \rangle\}.$$

Định nghĩa 4. Bậc của văn phạm phi ngữ xác suất là số cực đại các ký kiệu không kết thúc có thể có trong α của các quy tắc sản suất $X \to \langle \alpha, \omega \rangle$.

Định nghĩa 5. (PSCFG) Văn phạm phi ngữ cảnh đồng bộ xác suất là bộ 5

$$G = (N, S, T_\sigma, T_\tau, R)$$

trong đó N là tập các ký hiệu không kết thúc của văn phạm, $S \in N$ là ký hiệu khởi đầu, T_σ và T_τ là các tập từ vựng (hay các ký hiệu kết thúc) của ngôn ngữ nguồn và ngôn ngữ đích tương ứng, R là tập các quy tắc sản xuất của văm phạm.

Mỗi quy tắc sản xuất của R có dạng:

$$X \to \langle \alpha, \beta, \sim \omega \rangle \tag{5}$$

trong đó $X \in N$ là một ký hiệu không kết thúc, $\alpha \in (N \cup T_\sigma)^*$, $\beta \in (N \cup T_\tau)^*$, $\sim$ là một tương ứng 1-1 giữa các ký hiệu không kết thúc của α và β (quy ước sử dụng cùng một ký hiệu không kết thúc cho mỗi cặp ký hiệu không kết thúc tương ứng và như vậy $\sim$ sẽ là một tập các ký hiệu không kết thúc xuất hiện đồng thời trong cả α và β), ω là xác suất áp dụng của quy tắc và thỏa mãn với mỗi X

$$\sum_{\alpha,\beta:X\to\langle\alpha,\beta,\sim\omega\rangle\in R} \omega = 1.$$

Các biểu diễn như vế phải của (5) được gọi là một dạng câu.

Định nghĩa 6. Giả sử r là một quy tắc sản xuất $X_r \to \langle \alpha_r, \delta_r, \sim_r \omega_r \rangle$ và $X_r \in \sim$. Khi đó kết quả của việc áp dụng quy tắc r vào một dạng câu $\langle \alpha X_r \beta, \delta X_r \varphi, \sim, \omega \rangle$ sẽ là dạng câu $\langle \alpha\alpha_r\beta, \delta\delta_r\varphi, \sim \cup \sim_r -\{X\}, \omega\omega_r \rangle$ và ta viết

$$\langle \alpha X_r \beta, \delta X_r \varphi, \sim, \omega \rangle \xrightarrow{r} \langle \alpha\alpha_r\beta, \delta\delta_r\varphi, \sim \cup \sim_r -\{X\}, \omega\omega_r \rangle.$$

Một cách tổng quát nếu ta áp dụng liên tiếp như trên một số hữu hạn hoặc đếm được các quy tắc vào một dạng câu $\langle \alpha, \beta, \sim, \omega \rangle$ để được một dạng câu mới $\langle \alpha_1, \beta_1, \sim_1, \omega_1 \rangle$, ta sẽ viết

$$\langle \alpha, \beta, \sim, \omega \rangle \xrightarrow{*} \langle \alpha_1, \beta_1, \sim_1, \omega_1 \rangle.$$

Định nghĩa 7. Một dạng câu được dẫn xuất từ dạng câu ban đầu $\langle S, S, \{S\}, 1\rangle$ và không chứa ký hiệu không kết thúc (tương đương với $\sim= \phi$) được gọi là một câu sinh ra bởi văn phạm. Ngôn ngữ của văn phạm phi ngữ cảnh đồng bộ xác suất G, ký hiệu $L(G)$, là tập tất cả các câu được sinh ra bởi văn phạm G

$$L(G) = \{\langle \xi, \xi, \phi, w\rangle \in T_\sigma^* \times T_\tau^* \times \phi \times [0,1] : \langle S, S, \{S\}, 1\rangle \xrightarrow{*} \langle \xi, \xi, \phi, w\rangle\}.$$

Định nghĩa 8. Bậc của mỗi quy tắc sản suất là lực lượng của ánh xạ trong quy tắc $|\sim|$. Bậc của văn phạm phi ngữ cảnh đồng bộ xác suất là bậc cực đại của các quy tắc. Nói cách khác bậc của một ngôn ngữ PSCFG là

$$k = \max\{|\sim| : X \longrightarrow \langle \alpha, \beta, \sim, \omega\rangle\}$$

trong đó $|\sim|$ chỉ lực lượng của tập $\sim$.

Định nghĩa 9. Hai ngôn ngữ G và G' được gọi là tương đương khi và chỉ khi $L(G) = L(G')$.

2.2. Mô hình cơ sở hệ dịch máy thống kê dựa trên cú pháp

Giả sử ta đã có một văn phạm phi ngữ cảnh đồng bộ xác suất G với ngôn ngữ $L(G)$ như Định nghĩa trong 2.1. Khi đó, với mỗi xâu kết thúc f trên T_σ, bản dịch của nó theo văn phạm G là xâu các ký tự kết thúc e^* trên T_τ sao cho $\langle S, S, \{S\}, 1\rangle \xrightarrow{*} \langle f, e^*, \phi, w\rangle$ và w đạt cực đại. Nói cách khác, bài toán dịch máy được hình thức hóa như sau

$$e^* = \arg \max_{e:\langle S,S,\{S\},1\rangle \xrightarrow{*} \langle f,e^*,\phi,w\rangle} w. \tag{6}$$

Như vậy để có được một hệ thống dịch máy thống kê dựa trên cú pháp (hay để có thể giải bài toán (6)), chúng ta cần có một mô hình ngôn ngữ phi ngữ cảnh đồng bộ xác suất G và một bộ phân tích cú pháp P_G tương ứng với văn phạm cho phép tính xác suất sinh ra mỗi câu của ngôn ngữ $L(G)$.

3. MÔ HÌNH HỆ THỐNG KHÔI PHỤC DẤU VĂN BẢN BẰNG CÁCH TIẾP CẬN DỊCH MÁY THỐNG KÊ DỰA TRÊN CÚ PHÁP

Ví dụ đưa ra trong phần giới thiệu cho thấy, việc khôi phục dấu cho mỗi câu trong văn bản không dấu phụ thuộc nhiều vào cấu trúc cú pháp của câu. Nếu ta có thể xác định được cấu trúc cú pháp của câu phù hợp với cấu trúc cú pháp của câu nguyên thủy, thì việc khôi phục dấu cho nó trở nên hiệu quả và chính xác hơn.

Trước hết ta có nhận xét rằng nếu $x \in T^*$ là một câu có dấu trong ngôn ngữ và $\overline{x}$ là biến thể không dấu của x thì thứ tự tuyệt đối của mỗi âm tiết/từ trong x trùng với thứ tự tuyệt đối của biến thể không dấu của nó trong $\overline{x}$.

Thứ hai, ta cũng có thể khẳng định rằng việc hình thành văn bản không dấu hoàn toàn tuân theo những ràng buộc cú pháp ẩn trong tư duy của người soạn thảo – những ràng buộc được người đó dùng khi soạn văn bản có dấu để diễn đạt nội dung cần trình bày.

Từ những quan sát đó, ta đi đến đề xuất cách sử dụng các thông tin cú pháp chứa trong các văn bản có dấu để xác định cấu trúc cú pháp ẩn trong văn bản không dấu phục vụ việc giải quyết bài toán đặt ra.

3.1. Mô hình hệ thống

Như đã được phân tích trong mục 2.2, việc xây dựng mô hình dịch máy thống kê dựa trên cú pháp để giải bài toán tự động khôi phục dấu cho văn bản đồng nghĩa với việc xây dựng một văn phạm PSCFG cho ngôn ngữ được quan tâm và một bộ phân tích cú pháp tương ứng với nó. Sau đây chúng ta lần lượt đưa ra giải pháp cho từng vấn đề.

3.1.1. Mô hình văn phạm ngữ phi ngữ cảnh đồng bộ xác suất

Giả sử ta đã có một văn phạm phi ngữ cảnh xác suất $G = (N, S, T, R)$ sinh ra các câu của một ngôn ngữ có sử dụng dấu trong hệ thống chính tả được quan tâm $L(G)$.

Ta sẽ xây dựng một văn phạm PSCFG $\overline{G} = (N, S, \overline{T}, T, \overline{R})$ dựa trên văn phạm G như sau:

- Tập các ký hiệu không kết thúc của $\overline{G}$ cũng chính là tập các ký hiệu không kết thúc N của G.
- Ký hiệu khởi đầu của $\overline{G}$ cũng chính là ký hiệu khởi đầu S của G.
- $\overline{T}$ là tập nhận được từ tập từ vựng T của G bằng cách bỏ đi dấu của các âm tiết/từ trong T.
- Tập các quy tắc sản xuất $\overline{R}$ được hình thành như sau: Với mỗi quy tắc

$$X \longrightarrow \langle x_1 A_1 x_2 A_2 ... A_n x_{n+1}, \omega \rangle \in R$$

trong đó $x_i \in T,\ i = 1, n+1$, các x_i có thể là các xâu rỗng $\varepsilon,\ A_i \in N,\ i = 1, n$ ta đưa vào $\overline{R}$ quy tắc sản xuất sau

$$X \longrightarrow \langle \overline{x_1} A_1 x_2 A_2 ... A_n \overline{x_{n+1}}, A_1 x_2 A_2 ... A_n x_{n+1}, I, \omega \rangle \tag{7}$$

với $\overline{x_i} \in \overline{T}^*$ là biến thể không dấu tương ứng của x_i và $I = \{A_1, ..., A_n\}$ là ánh xạ đơn điệu theo thứ tự giữa các cặp ký hiệu không kết thúc. Do mọi quy tắc đều chứa ánh xạ đơn điệu trên các ký hiệu kết thúc nên ta có thể bỏ qua mà không sợ bị lầm lẫn và từ nay về sau ta viết gọn các quy tắc của $\overline{R}$ thành

$$X \longrightarrow \langle \overline{x_1} A_1 x_2 A_2 ... A_n \overline{x_{n+1}}, A_1 x_2 A_2 ... A_n x_{n+1}, \omega \rangle \tag{8}$$

Định lý 1. *$\overline{G}$ là một văn phạm phi ngữ cảnh đồng bộ xác suất. Hơn nữa thứ tự tuyệt đối của các âm tiết/từ và các biến thể không dấu của chúng trong mọi dạng câu (và câu của ngôn ngữ $L(\overline{G})$) được bảo tồn.*

Chứng minh: Với cách xây dựng tập các quy tắc sản xuất như (7), dễ thấy rằng trong mọi quy tắc $X \to \langle \alpha, \beta, \omega \rangle \in \overline{R}$, số các ký hiệu không kết thúc trong α và β là bằng nhau và chúng được tương ứng với nhau bằng một ánh xạ đơn điệu tăng theo vị trí tuyệt đối của chúng trong α và β. Do đó văn phạm $\overline{G}$ là một PSCFG.

Đối với khẳng đinh thứ 2, ta chứng minh bằng quy nạp theo số lượng các quy tắc dùng để sinh ra dạng câu hoặc câu. Xuất phát từ dạng câu $\langle S, S, 1 \rangle$ và không áp dụng quy tắc nào, ta có $\langle S, S, 1 \rangle \xrightarrow{\phi} \langle S, S, 1 \rangle$ và khẳng định là đúng.

Giả sử khẳng định đã đúng với tất cả các dãy suy diễn với độ dài m và dạng câu nhận được có biểu diễn

$$\overline{\alpha} X \overline{\beta}, \alpha X \beta, \omega$$

tức là các âm tiết/từ và các biến thể không dấu của chúng trong các cặp $(\overline{\alpha}, \alpha)$, $(\overline{\beta}, \beta)$ có thứ tự tuyệt đối trong dạng câu là trùng nhau.

Nếu áp dụng quy tắc sản suất

$$X \longrightarrow \langle \overline{x_1} A_1 ... A_n \overline{x_{n+1}}, x_1 A_1 x_2 A_2 ... A_n x_{n+1}, \omega' \rangle$$

vào dạng câu trên, ta nhận được dạng câu mới

$$\langle \overline{\alpha x_1} A_1 ... A_n \overline{x_{n+1} \beta}, \alpha x_1 A_1 x_2 A_2 ... A_n x_{n+1} \beta, \omega, \omega' \rangle.$$

Trong dạng câu mới này, thứ tự tuyệt đối của âm tiết/từ cũng như các biến thể không dấu trong cặp $(\overline{\alpha}, \alpha)$ không thay đổi, thứ tự tuyệt đối của âm tiết/từ trong cặp $(\overline{\beta}, \beta)$ được tịnh tiến một khoảng bằng độ dài $l(\overline{x_1} A_1 ... A_n \overline{x_{n+1}}) = l(x_1 A_1 x_2 A_2 ... A_n x_{n+1})$, thứ tự tuyết đối của các âm tiết/từ trong các cặp $(\overline{x_i}, x_i)$ là thứ tự tuyết đối của chúng trong quy tắc sản xuất được tịnh tiến một khoảng bằng $l(\overline{\alpha}) + l(\overline{x_1} A_1 ... \overline{x_{i-1}} A_i)$ với quy định $l(\overline{x_0} A_0) = 0$. Từ đó ta suy ra tính đúng đắn của khẳng định.

Định nghĩa 10. Văn phạm phi ngữ cảnh xác suất trái của văn phạm phi ngữ cảnh đồng bộ xác suất $\overline{G} = (N, S, \overline{T}, T, \overline{R})$, ký hiệu $\overline{G_t}$ là văn phạm

$$\overline{G_t} = (N, S, \overline{T}, \overline{R_t})$$

R_t là tập các quy tắc sản xuất được xây dựng như sau

$$X \longrightarrow \langle \alpha, \beta, \sim, \omega \rangle \in \overline{R} \text{ khi và chỉ khi } X \longrightarrow \langle \alpha, \omega \rangle \in \overline{R_t}. \tag{9}$$

Nếu ta gán cùng một nhãn cho các quy tắc sản xuất liên quan đến nhau trong (7), (8), (9) và r là một chuỗi có thứ tự các nhãn, ta có định lý sau.

Định lý 2. *Cho xâu $x \in T^*$ và $\overline{x} \in \overline{T}^*$ là biến thể không dấu của x. Khi đó nếu $\langle S, 1 \rangle \xrightarrow[G]{r} \langle x, \omega \rangle$ thì*

$$\langle S, S, 1 \rangle \xrightarrow[\overline{G}]{r} \langle \overline{x}, x, \omega \rangle;\ \langle S, 1 \rangle \xrightarrow[\overline{G_t}]{r} \langle \overline{x}, \omega \rangle.$$

Hay nói cách khác $\overline{x}$ có cùng cấu trúc cú pháp với x. (các ký hiệu $G, \overline{G}$ và $\overline{G_t}$ bên dưới dấu dẫn xuất dùng để chỉ chuỗi dẫn xuất được thực hiện bằng các quy tắc sản xuất của văn phạm tương ứng).

Chứng minh: dễ ràng suy ra trực tiếp từ mối liên quan của các quy tắc sản xuất được xây dựng cho các văn phạm $G, \overline{G}$ và $\overline{G_t}$.

Từ đó bài toán khôi phuc dấu cho ngôn ngữ $L(G)$ có thể được phát biểu lại như sau:

Giả sử $\overline{x} \in \overline{T}^*$ là một biến thể không dấu của một câu chưa biết $x \in T^*$ thuộc ngôn ngữ $L(G)$. Khi đó bản gốc có dấu với độ tin cậy lớn nhất của $\overline{x}$ có thể coi là nghiệm cua bài toán:

$$x^* = \arg \max_{x: \langle S, S, 1 \rangle \xrightarrow[\overline{G}]{*} \langle \overline{x}, x, w \rangle} w. \tag{10}$$

3.1.2. Bộ phân tích cú pháp

Để làm bộ phân tích cú pháp cho các văn phạm PSCFG, người ta sử dụng biến thể mở rộng của thuật toán CKY – thuật toán phân tích cú pháp bottom-up sử dụng phương pháp quy hoạch động với độ phức tập tính toán của các thuật toán là hàm mũ theo bậc của văn phạm cũng như lực lượng của tập các quy tắc sản xuất [1]. Để có thể tăng hiệu quả cho thuật

toán, người ta thường tìm cách biến đổi văn phạm ban đầu thành một văn phạm tương đương ở dạng chuẩn Chomsky – dạng chuẩn mà các xâu trong vế phải của các quy tắc sản xuất chứa tối đa 2 ký hiệu không kết thúc (dạng nhị phân). Tuy nhiên, không phải mọi văn phạm PSCFG đều có thể đưa được về dạng chuẩn Chomsky.

Định lý 3. *Văn phạm $\overline{G}$ với các quy tắc sản xuất có dạng (8) có thể đưa được về dạng chuẩn Chomsky tương đương bằng một thủ tục có thời gian tuyến tính.*

Chứng minh: Quá trình nhị phân hóa là quá trình tách các quy tắc sản xuất thành một tập các quy tắc sản xuất tương đương, trong đó mỗi quy tắc nhận được có không quá 2 ký hiệu không kết thúc.

Giả sử ta có quy tắc:

$$X \longrightarrow \langle \overline{x_1} A_1 ... A_m \overline{x_{m+1}}, x_1 A_1 x_2 A_2 ... A_m x_{m+1}, \omega \rangle$$

ta sẽ biến đổi quy tắc đó thành 2 quy tắc:

$$X \longrightarrow \langle Y \overline{x_3} A_3 ... A_m \overline{x_{m+1}}, Y x_3 A_3 ... A_m x_{m+1}, 1 \rangle \tag{11}$$

$$Y \longrightarrow \langle \overline{x_1} A_1 \overline{x_2} A_2, x_1 A_1 x_2 A_2, \omega \rangle \tag{12}$$

với Y là một ký hiệu không kết thúc bổ xung.

Áp dụng phương pháp trên một cách đệ quy vào mọi quy tắc sản xuất của $\overline{G}$ để nhận được một tập các quy tắc sản xuất mới ở dạng nhị phân. Thay tập quy tắc sản xuất trong $\overline{G}$ bằng tập các quy tắc mới để tạo ra văn phạm mới $\overline{\overline{G}}$. Do cách xây $\overline{G}$ nên quá trình trên là thực hiện được và sau nhiều nhất $k-1$ bước, ta có thể biến đổi mỗi quy tắc về các quy tắc dạng nhị phân với k là bậc của văn phạm. Và như vậy thời gian biến đổi $\overline{G}$ về dạng chuẩn Chomsky $\overline{\overline{G}}$ là tuyến tính với số lượng quy tắc ban đầu của văn phạm.

Để chứng minh $L(\overline{\overline{G}}) = L(\overline{G})$ ta sẽ quy nạp theo bậc k của văn phạm $\overline{G}$.

Với $k = 2$, ta có $\overline{G} \equiv \overline{\overline{G}}$. Do đó khẳng định là đúng.

Giả sử khẳng định đã đúng với mọi văn phạm Γ có bậc $k \langle m$ với các quy tắc sản xuất dạng (8). Ta sẽ chứng minh khẳng định cũng đúng với các văn phạm có bậc m.

Giả sử $\overline{G}$ là văn phạm bậc m có các quy tắc dạng (8). Ta thực hiện xây dựng một văn phạm Γ bậc $m-1$ cũng có các quy tắc dạng (8) như sau:

- Đưa tất cả các quy tắc có bậc nhỏ hơn m từ $\overline{G}$ sang văn phạm Γ.

- Với mỗi quy tắc có bậc m trong $\overline{G}$

$$X \longrightarrow \langle \overline{x_1} A_1 ... A_m \overline{x_{m+1}}, x_1 A_1 x_2 A_2 ... A_m x_{m+1}, \omega \rangle$$

ta đưa vào văn phạm mới 2 quy tắc:

$$X \longrightarrow \langle Y \overline{x_3} A_3 ... A_m \overline{x_{m+1}}, Y x_3 A_3 ... A_m x_{m+1}, 1 \rangle$$

$$Y \longrightarrow \langle \overline{x_1} A_1 \overline{x_2} A_2, x_1 A_1 x_2 A_2, \omega \rangle$$

với Y là ký hiệu kết thúc mới trong văn phạm Γ. Như vậy Γ có bậc $m-1$.

Giả sử $r = (r_1, ..., r_t)$ là một dãy các quy tắc sản xuất trong $\overline{G}$ để sinh ra một câu thuộc $L(\overline{G})$. Khi đó có 2 trường hợp xảy ra:

- Tất cả các quy tắc trong r đều có bậc nhỏ hơn m. Trong trương hợp này dãy các quy tắc sản xuất tương ứng trong Γ cũng sinh ra câu như vậy trong $L(\Gamma)$.

- Trong dãy r có quy tắc sản xuất nào đó có bậc m. Trong trường hợp này ta sẽ lập một dãy các quy tắc sản xuất trong Γ bằng cách tại mỗi vị trí của r chứa quy tắc sản xuất bậc m ta thay quy tắc đó bằng 2 quy tắc tương ứng theo (11) và (12). Rõ ràng dãy quy tắc sản xuất nhận được chỉ chứa các quy tắc của Γ đồng thời dãy quy tắc đó sinh ra câu đang xét trong $L(\Gamma)$.. Từ đây ta có $L(\overline{\Gamma}) \subseteq L(\Gamma)$.

Ngược lại giả sử r là một dãy các quy tắc sản xuất trong (Γ) sinh ra một câu trong $L(\Gamma)$. Cũng có 2 trường hợp xảy ra:

- Trong các quy tắc của r không có quy tắc nào chứa các ký hiệu không kết thúc mới so với $\overline{G}$. Khi đó dãy quy tắc tương ứng trong $\overline{G}$ cũng sinh ra câu đang xét trong $\overline{G}$.

- Trong các quy tắc của r có quy tắc chứa ký hiệu không kết thúc mới Y nào đó. Do các tạo quy tắc sản xuất của (Γ), ký hiệu không kết thúc này xuất hiện chỉ trong 1 cặp quy tắc dạng (11) và (12). Vì dãy quy tắc sản xuất sinh ta câu của ngôn ngữ nên cặp đó phải đồng thời nằm trong dãy r. Thay cặp quy tắc này bằng quy tắc sinh ra chúng từ $\overline{G}$ để nhận được dãy quy tắc mới. Dễ ràng chỉ ra dãy quy tắc mới cũng sinh ra câu đang xét trong $\overline{G}$. Nói cách khác $L(\Gamma) \subseteq L(\overline{\Gamma})$.

Từ đó ta có $L(\Gamma) = L(\overline{\Gamma})$. Định lý được chứng minh. ■

Định lý 3 cho thấy rằng ta có thể dùng biến thể CKY cho PSCFG làm bộ phân tích cú pháp trong mô hình của chúng ta sau khi đã nhị phân hóa $\overline{G}$ để nhận được văn phạm $\overline{\overline{G}}$. Khi đó độ phức tạp của thuật toán CKY là $O(n^3)$ với n là độ dài của xâu đầu vào $\overline{x}$.

3.2. Phương pháp huấn luyện mô hình

Trong Mục 3.1, đã giả sử có một văn phạm phi ngữ cảnh xác suất $G = (N, S, T, R)$ sinh ra các câu của một ngôn ngữ $L(G)$.

Để hoàn thiện mô hình, ta sử dụng phương pháp học không giám sát dựa trên gióng hàng (Alignment-based learning - ABL) của Menno M. van Zaanen [2] để xây dựng văn phạm phi ngữ cảnh xác suất G từ thông tin đầu vào là tập các câu phẳng (plain sentences) của ngôn ngữ có sử dụng dấu trong hệ thống chính tả mà ta quan tâm. Do khuôn khổ hạn chế của bài báo, nên không trình bày chi tiết về phương pháp ABL. Độc giả quan tâm có thể tham khảo [2].

Dựa trên văn phạm phi ngữ cảnh xác suất nhận được G, chúng ta tiến hành thủ tục xây dựng văn phạm phi ngữ cảnh đồng bộ xác suất bậc 2 $\overline{\overline{G}}$ như đề xuất Mục 3.1.

4. CÀI ĐẶT VÀ THỬ NGHIỆM

4.1. Cài đặt

Để cài đặt hệ thống, ta mở rộng gói phần mềm nguồn mở ABL4J của tác giả [2] bằng việc bổ xung thủ tục sinh văn phạm phi ngữ cảnh đồng bộ xác suất như trong mục 3.1.1 và thủ tục biến đổi về dạng chuẩn Chomsky như trong mục 3.1.2. Modul này sinh ra văn phạm phi ngữ cảnh đồng bộ xác suất bậc 2 $\overline{\overline{G}}$ cho mô hình từ tập ngữ liệu đầu vào gồm các câu có dấu của ngôn ngữ quan tâm. Modul hoạt động độc lập và có thể áp dụng cho các ngôn ngữ khác nhau để có được các văn phạm PSCFG ở dạng chuẩn Chomsky tương ứng.

Phiên bản xác suất của bộ phân tích cú pháp CKY nhận $\overline{\overline{G}}$ làm tham số. Thông qua bộ phân tích cú pháp này, một văn bản không dấu đầu vào của ngôn ngữ quan tâm sẽ được tự

động khôi phục dấu.

4.2. Thu thập dữ liệu và thử nghiệm

Để đánh giá hiệu quả của phương pháp, ta lựa chọn khôi phục dấu cho tiếng Việt, một ngôn ngữ sử dụng tập dấu chính tả khá phong phú. Ngữ liệu được thu thập gồm 450 bài báo thuộc các chủ đề khác nhau trên các trang báo điện tử như Dân trí, Vietnamnet, Vnexpress...

Ngữ liệu này được tiền xử lý như biến chữ cái viết hoa thành chữ cái viết thường, tách câu theo dấu chấm câu, loại bỏ các câu trùng nhau... để nhận được một tập ngữ liệu mới chứa 14558 câu tiếng Việt và đưa vào để huấn luyện mô hình.

Với mô hình nhận được, ta tiến hành việc thử nghiệm như sau: Thu thập một số văn bản tiếng Việt mới và dùng tool của Unikey để loại bỏ dấu. Sau đó đưa các văn bản không dấu vào hệ thống để thực hiện việc khôi phục dấu theo từng câu (tự động tách câu theo dấu chấm câu). Văn bản khôi phục dấu được đối sánh với văn bản gốc để đánh giá mức độ chính xác của phương pháp bằng tỷ lệ của số âm tiết/từ giống nhau trong hai văn bản chia cho số lượng âm tiết/từ có trong văn bản. Kết quả cho thấy độ chính xác của phương pháp giao động quanh tỷ lệ 98%.

5. KẾT LUẬN

Trong bài này, việc khôi phục dấu cho văn bản được hình thức hóa toán học một cách chặt chẽ bằng một mô hình dich máy thông kê dưa trên cú pháp.

Kết quả thử nghiệm cho thấy việc sử dụng thông tin cú pháp trong bài toán tự động khôi phục dấu cho văn bản giúp nâng cao đáng kể độ chính xác của kết quả.

Tuy nhiên trong quá trình thử nghiệm cũng xuất hiện một số trường hợp xâu đầu vào không thể nhận biết được bởi văn phạm do hệ thống sinh ra từ tập ngữ liệu huấn luyện vì trong xâu xuất hiện các âm tiết/từ mới (OOV). Trước mắt trong những trường hợp này chúng tôi chọn giải pháp chép lại toàn bộ xâu vào văn bản kết quả. Trong tương lai có thể khắc phục hiện tượng đó bằng sự kết hợp với phương pháp khôi phục dấu ở mức ký tự.

Hệ thống có tính độc lập ngôn ngữ cao nên hoàn toàn có thể áp dụng cho các ngôn ngữ có sử dụng dấu khác.

TÀI LIỆU THAM KHẢO

[1] Philipp Koehn, "Statistical Machine Translation," University of Edinburgh, 2007

[2] Menno M. van Zaanen, "Boostrapping Structure into Language: Alignment-Based Learning," Ph.D. thesis, University of Leeds, 2001.

[3] Kiem-Hieu Nguyen et al., Diacritics restoration in vietnamese: letter based vs. syllable based model. PRICAI'10 *Proceedings of the 11th Pacific Rim International Conference on Trends in Artificial Intelligence*, Springer-Verlag Berlin, Heidenberg, 2010 (631–636).

[4] Guy De Pauw et al., *Automatic Diacritic Restoration for Resource-Scarce Languages,* In V. Matousek and P. Mautner (Eds.): TSD 2007, LNAI 4629, Springer Verlag Berlin, Heidenberg, 2007 (170–179).

[5] J. A. Mahar, G. Q. Memon, and H. Shaikh, Sindhi diacritics restoration by letter level learning approach, *Sindh Univ. Res. Jour. (Sci. Ser.)* **43** (2) (2011) 119–126.

[6] John Cocks and Te Taka Keegan, A word-based approach for diacritic restoration in Maori, *Proceedings of Australasian Language Technology Association Workshop*, Canberra, Australia, 2011 (126–130).

[7] Tuan Anh Luu et al., A pointwise approach for Vietnamese diacritics restoration, *International Conference on Asian Language Processing (IALP),* Hanoi, Vietnam, 2012.

[8] Tim Schlippe et al., Diacritization as a machine translation problem and as a sequence labeling problem, 2007 www.amtaweb.org/papers/3.05_Schlippe.pdf

[9] Rada F. Mihalcea, Diacritics Restoration: Learning from Letters versus Learning from Words. *CICLing, volume 2276 of Lecture Notes in Computer Science*, Springer, 2002 (339–348).

Ngày nhận bài 30 - 3 - 2013
Nhận lại sau sửa ngày 28 - 02 - 2014