Nghiên cứu luật hiệu chỉnh kết quả dùng phương pháp MST phân tích cú pháp phụ thuộc tiếng Việt

Nguyễn Lê Minh
Japan Advanced Institute of Science and Technology

Hoàng Thị Điệp
Đại học Công Nghệ - ĐHQG Hà Nội

Trần Mạnh Kế
Đại học Công Nghệ - ĐHQG Hà Nội

Tóm tắt

Phân tích cú pháp có vai trò quan trọng trong lĩnh vực xử lý văn bản vì nó là bước trung gian của nhiều bài toán lớn như: tóm tắt văn bản, dịch máy, hỏi đáp tự động. Trong thời gian gần đây, phân tích cú pháp phụ thuộc thu hút được sự quan tâm của nhiều nhóm nghiên cứu xử lý ngôn ngữ tự nhiên trên thế giới bởi quan hệ phụ thuộc giữa hai từ vựng có thể có ích trong khử nhập nhằng và cú pháp này có khả năng mô hình hóa các ngôn ngữ có trật tự từ tự do. Trong báo cáo này, chúng tôi trình bày phương pháp Maximum Spanning Tree để phân tích cú pháp phụ thuộc câu tiếng Việt và sử dụng bộ hiệu chỉnh cây bằng luật để cải thiện đầu ra của MST. Cuối cùng chúng tôi đưa ra một số kết quả thực nghiệm trên tập ngữ liệu 450 câu tiếng Việt và đề xuất hướng phát triển phương pháp MST cho bài toán này.

1 Giới thiệu

1.1 Tình hình nghiên cứu tự động phân tích cú pháp phụ thuộc tiếng Việt

Phân tích cú pháp phụ thuộc[1] trong vài năm gần đây thu hút được sự quan tâm của cộng đồng nghiên cứu xử lý ngôn ngữ tự nhiên [8] vì cú pháp phụ thuộc là một dạng biểu diễn câu có nhiều ứng dụng cho các bài toán phức tạp như trích chọn thông tin hay tóm tắt văn bản. Tuy nhiên, các tiếp cận cho bài toán này đều dựa trên học máy và đòi hỏi kho ngữ liệu với nhiều thông tin về từ loại và quan hệ phụ thuộc nên hiện chưa có ai công bố nghiên cứu về phân tích cú pháp phụ thuộc tiếng Việt.

1.2 Cú pháp phụ thuộc

Cú pháp phụ thuộc là cấu trúc cú pháp chứa các mục từ vựng nối với nhau bởi các quan hệ nhị phân không đối xứng gọi là sự phụ thuộc [5]. Quan hệ phụ thuộc này có thể được đặt tên để làm rõ liên hệ giữa hai mục từ.

Hình 2 là minh họa cú pháp phụ thuộc của một câu tiếng Việt. Theo quy ước phổ biến trong các tài liệu về cú pháp phụ thuộc thì mục từ nằm ở gốc của mũi tên là từ chính – gọi là head, mục từ nằm ở đầu mũi tên là từ phụ - gọi là dependent.

Theo [7], ta cũng có thể định nghĩa một cách hình thức: cú pháp phụ thuộc của một câu cho trước là một đồ thị định hướng với gốc *root* là một nút giả, thường được chèn vào bên trái câu, các nút còn lại là các mục từ của câu. Đồ thị này có các tính chất sau:

1. Nó liên thông yếu (có xét hướng)
2. Mỗi mục từ có chính xác một cạnh đi vào (trừ *root* là không có cạnh đi vào)

[1] Thuật ngữ tiếng Anh là "denpendency parsing"

3. Không có chu trình
4. Nếu có n mục từ trong câu (kể cả *root*) thì đồ thị có chính xác (n-1) cạnh

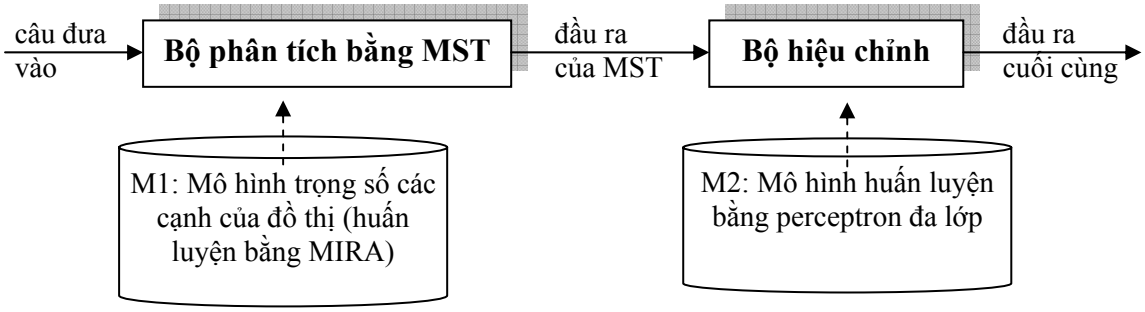
Nhờ cách mô hình hóa như trên, cú pháp phụ thuộc biểu diễn được những ngôn ngữ có trật tự từ tự do (xem thêm Phần 2.3), đây là điều mà cú pháp cấu trúc cụm[2] - vốn phù hợp với những ngôn ngữ có nhiều quy tắc chặt chẽ trong cấu thành câu - không làm được. Tuy vậy, không có nghĩa là phân tích ngôn ngữ có trật tự từ xác định thì chỉ dùng cấu trúc cụm hay phân tích ngôn ngữ có trật tự từ tự do thì chỉ dùng cấu trúc phụ thuộc [10].

1.3 Bài toán tự động phân tích cú pháp phụ thuộc

Phân tích cú pháp phụ thuộc là tìm cây phụ thuộc cho một câu. Mục tiêu của nghiên cứu này là tìm ra phương pháp sinh cây phụ thuộc chính xác nhất cho câu tiếng Việt đưa vào, nghĩa là làm cực đại số cung chính xác trong cây và số nhãn gán đúng cho các cung.

1.4 Tóm tắt về hướng tiếp cận trong báo cáo này

Hình 1 mô tả quá trình xác định cây phụ thuộc của một câu tiếng Việt của nghiên cứu này, nó gồm hai bước: 1- thiết lập đồ thị định hướng có trọng số bằng cách khai thác mô hình trọng số và đưa về bài toán tìm cây khung tối đại[3] trong đồ thị [7], 2- tự động phát hiện lỗi của cây đầu ra MST và lựa chọn các luật hiệu chỉnh cây phù hợp [9].



Hình 1 Sơ đồ minh họa quá trình phân tích phụ thuộc khảo sát

Mô hình M1 được sinh ra bằng phương pháp học máy MIRA[4] [11] học trên dữ liệu huấn luyện. Còn M2 được sinh bằng Perceptron đa lớp [11] học trên tập kết hợp đầu ra của MST và dữ liệu huấn luyện.

1.5 Sơ lược cấu trúc báo cáo

Trong các phần sau đây của báo cáo, chúng tôi trình bày một số đặc trưng của ngữ pháp tiếng Việt (tham khảo chủ yếu từ các tài liệu về ngôn ngữ) có thể liên quan tới quá trình tự động phân tích cú pháp phụ thuộc. Sau đó lần lượt trình bày cách xây dựng bộ phân tích cú pháp phụ thuộc MST và cách xây dựng bộ hiệu chỉnh cây phụ thuộc để cải thiện kết quả. Mô tả phương pháp đánh giá, thước đo và kết quả thử nghiệm ban đầu của các phương pháp này trên tiếng Việt sẽ được trình bày ở cuối báo cáo.

[2] Thuật ngữ tiếng Anh là "phrase structure syntax"
[3] Thuật ngữ tiếng Anh là "Maximum Spanning Tree" - viết tắt là MST
[4] MIRA là viết tắt của Margin Infused Relaxed Algorithm

2 Một số đặc trưng ngữ pháp tiếng Việt liên quan

Bảng 1 Tóm tắt các đặc trưng ngữ pháp tiếng Việt

Đặc trưng	Tính phân tích	Tính đơn hình	Trật tự từ	Điều kiện xạ ảnh	Từ loại của vị tố
Tiếng Việt	*có*	*có*	*SVO*	*đa số nhưng không phải toàn bộ*	*động từ, tính từ, danh từ, một số hư từ*

Mục này trình bày một số đặc trưng ngữ pháp của tiếng Việt, ở cả góc độ ngôn ngữ (gồm tính phân tích, tính đơn hình và trật tự từ [1]) và góc độ bài toán tự động phân tích phụ thuộc (gồm điều kiện xạ ảnh [5] và từ loại của vị tố [6]). Thực tế thì ngữ pháp tiếng Việt còn nhiều đặc trưng khác nhưng trong nghiên cứu này chúng tôi chỉ tổng hợp những đặc trưng có thể liên quan tới quá trình phân tích phụ thuộc.

2.1 Tính phân tích [2]

Ngôn ngữ phân tích[5] là ngôn ngữ có ngữ pháp và ngữ nghĩa được hình thành nhờ nhờ cách dùng các tiểu từ và trật tự từ hơn là nhờ vào các biến tố. Ngược với ngôn ngữ phân tích là ngôn ngữ tổng hợp[6]. Các ngôn ngữ như tiếng Hi Lạp, tiếng La-tinh, tiếng Đức, tiếng Ý, tiếng Nga, tiếng Ba Lan và tiếng Séc là ví dụ điển hình cho loại tổng hợp. Theo [2] thì tiếng Việt cùng một số ngôn ngữ trong khu vực Đông Nam Á (trừ tiếng Malay) và tiếng Trung Quốc là ngôn ngữ phân tích.

2.2 Tính đơn hình [2, 3]

Khái niệm ngôn ngữ đơn hình[7] không đồng nhất với khái niệm ngôn ngữ phân tích. Ngôn ngữ đơn hình là ngôn ngữ có phần lớn hình vị là hình vị tự do và có đủ tiêu chuẩn là một từ. Mức độ đơn được xác định theo tỉ lệ số lượng hình vị - trên - số lượng từ. Ngôn ngữ đơn hình phổ biến ở các nước Đông Nam Á, trong đó có Việt Nam, và Trung Hoa cổ.

2.3 Trật tự từ[8] [4]

Trong ngôn ngữ học, hệ thống phân loại theo trật tự từ nói tới nghiên cứu về cách mà ngôn ngữ sắp xếp tương đối các thành phần của một câu và về quan hệ giữa các cách sắp này.

Với hầu hết các ngôn ngữ có danh từ chiếm đa số thì ta có thể định nghĩa một trật tự từ cơ bản theo động từ nguyên thể (V) và các đối số của nó, chủ ngữ (S) và tân ngữ (O). Theo đó có 6 trật tự cơ bản: SVO, SOV, VSO, VOS, OSV, OVS. Ngữ pháp Việt Nam thuộc loại SVO.

Bên cạnh các trật tự đã đề cập, còn một lớp các ngôn ngữ đáng lưu ý được gọi là ngôn ngữ có trật tự từ tự do (free word order language) – ví dụ như tiếng La-tinh, Séc, Hung-ga-ri, Ba Lan, Nga - đòi hỏi các phương pháp nghiên cứu phức tạp hơn trong bài toán phân tích tự động cú pháp phụ thuộc.

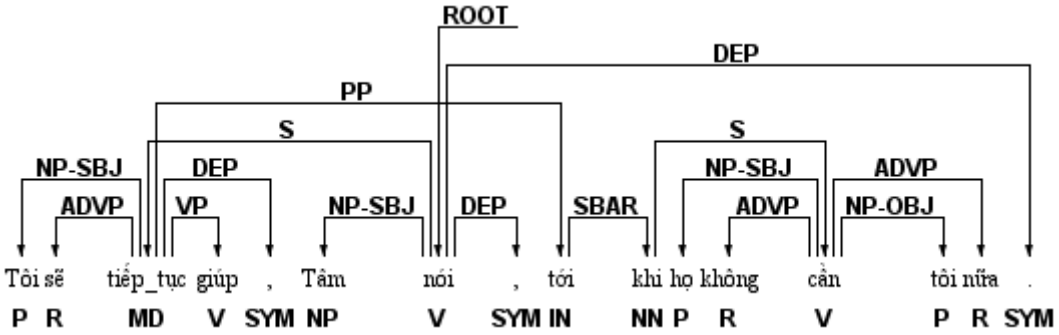
[5] Thuật ngữ tiếng Anh là "analytic language"
[6] Thuật ngữ tiếng Anh là "synthetic language"
[7] Thuật ngữ tiếng Anh là "isolating language"
[8] Thuật ngữ tiếng Anh là "word order"

2.4 Điều kiện xạ ảnh[9] [5]

Điều kiện xạ ảnh cho đồ thị phụ thuộc được phát biểu một cách hình thức trong bài giảng [5] như sau:

> Một đồ thị phụ thuộc được gọi là có tính xạ ảnh khi
> Nếu có $i \to j$ thì $i \to^* i'$ với i' bất kỳ thỏa mãn $i < i' < j$ hoặc $j < i' < i$.

Có thể phát biểu lại là: nếu từ tố *j* phụ thuộc vào từ tố *i* thì từ tố *i'* bất kỳ nằm giữa *i* và *j* phải phụ thuộc (có thể là gián tiếp) vào từ tố *i*.



Hình 2 Ví dụ câu tiếng Việt không thỏa mãn điều kiện xạ ảnh

Đa số các câu trong kho ngữ liệu của chúng tôi (Phần 5.1) thỏa mãn tính chất xạ ảnh mô tả ở trên, nhưng trong tiếng Việt vẫn tồn tại những câu ghép không có tính xạ ảnh như minh họa trong Hình 2. Rõ ràng là ta cần quan tâm tới những trường hợp này khi nghiên cứu giải thuật phân tích cú pháp phụ thuộc cho tiếng Việt.

2.5 Từ loại của vị tố trong câu tiếng Việt

Khái niệm từ khóa của câu (mục từ phụ thuộc vào nút giả *root*) trong phân tích phụ thuộc chính là khái niệm vị tố trong ngôn ngữ học. Trong tiếng Anh thì vị tố luôn là động từ, nhưng trong tiếng Việt, từ loại của vị tố rất đa dạng. Các ví dụ bên dưới được trích từ chương 1, phần 2.2. "Các kiểu câu cơ bản của tiếng Việt" trong cuốn "Ngữ pháp Việt Nam" [6]. Vị tố là các từ hay cụm từ in đậm.

Từ loại của vị tố	Ví dụ
động từ	*Giáp **đưa cho** Tị tờ báo.*
tính từ	*Trăng **sáng** quá.*
danh từ	*Em bé này **sáu tuổi**.*
hư từ "là"	*Anh này **là** thợ mộc.*
hư từ "bằng"	*Cái áo này **bằng** lụa.*

Từ loại của vị tố	Ví dụ
hư từ "tại", "do", "bởi"	*Việc này **tại** nó.* *Hàng này **do** họ làm.*
hư từ "để"	*Bàn ấy **để** uống nước.*
hư từ chỉ vị trí	*Ông tôi **ngoài** vườn.*
hư từ "như"	*Đỏ **như** hoa vông.*
hư từ "của"	*Xe này **của** Giáp.* *Hàng này **của** họ làm.*

3 Xây dựng bộ phân tích phụ thuộc theo tiếp cận MST

Ryan McDonald trong [7] đã đề xuất tiếp cận dựa trên đồ thị, cụ thể là đưa bài toán phân tích cú pháp phụ thuộc về bài toán tìm cây khung tối đại của một đồ thị định hướng có trọng

[9] Thuật ngữ tiếng Anh là "projectivity"

số (bài toán MST). Có hai phiên bản MST: bậc 1 và bậc 2. MST bậc 1 hoạt động đơn giản hơn và thực nghiệm trên kho ngữ liệu tiếng Việt cho thấy MST bậc 1 cho kết quả tốt hơn, do đó trong khuôn khổ nghiên cứu này chúng tôi dừng lại ở MST bậc 1.

3.1 Đưa về bài toán MST

Với mỗi câu $\boldsymbol{x}$, ta định nghĩa một đồ thị $G_{\boldsymbol{x}}$ với tập đỉnh $V_{\boldsymbol{x}}$ và tập cạnh $E_{\boldsymbol{x}}$ như sau:

$$V_x = \{ x_0 = root, x_1, ..., x_n\}$$

$$E_x = \{(i , j) : x_i \neq x_j, x_i \in V_x, x_j \in V_x - root\}$$

McDonald [7] đã chứng minh: tìm một cây phụ thuộc (xạ ảnh) có điểm số cao nhất tương đương với tìm cây khung (xạ ảnh) tối đại của đồ thị $G_{\boldsymbol{x}}$ có gốc tại nút giả *root* . Trong đó, điểm của một cây được phân tích thành tổng điểm tất cả các cạnh đơn lẻ trong cây, dạng phân tích này được kiểm chứng là đơn giản và hiệu quả. Đây chính là giải thích cho cách đặt tên MST bậc 1. Các đặc trưng trình bày trong Phần 3.2 và giải thuật trình bày trong Phần 3.3 cũng là các phiên bản gắn với MST bậc 1 này.

3.1.1 Tính điểm một cạnh

Điểm của cạnh *(i , j)* là tích vô hướng giữa vectơ biểu diễn đặc trưng của cạnh và một vectơ trọng số:

$$s(i , j) = \mathbf{w} \cdot \mathbf{f}(i , j)$$

$\mathbf{f}(i , j)$ là ký hiệu rút gọn cho $\mathbf{f}(\boldsymbol{x}, i , j)$ vì nó chứa cả những đặc trưng của câu $\boldsymbol{x}$.

Như vậy, điểm của cây phụ thuộc $\boldsymbol{y}$ cho câu $\boldsymbol{x}$ là

$$s(\boldsymbol{x} , \boldsymbol{y}) = \sum_{(i,j)\in \boldsymbol{y}} s(i , j) = \sum_{(i,j)\in \boldsymbol{y}} \mathbf{w} \cdot \mathbf{f}(i , j)$$

a)

Đặc trưng Uni-gram cơ bản
x_i-word, x_i-pos
x_i-word
x_i-pos
x_j-word, x_j-pos
x_j-word
x_j-pos

b)

Đặc trưng Bi-gram cơ bản
x_i-word, x_i-pos, x_j-word, x_j-pos
x_i-pos, x_j-word, x_i-pos
x_i-word, x_j-word, x_i-pos
x_i-word, x_i-pos, x_j-pos
x_i-word, x_i-pos, x_j-word
x_i-word, x_i-word
x_i-pos, x_j-pos

c)

Đặc trưng từ loại giữa hai mục từ
x_i-pos, b-pos, x_j-pos
Đặc trưng từ loại xung quanh hai mục từ
x_i-pos, x_i-pos+1, x_j-pos-1, x_i-pos
x_i-pos-1, x_i-pos, x_j-pos-1, x_j-pos
x_i-pos, x_i-pos+1, x_j-pos, x_j-pos+1
x_i-pos-1, x_i-pos, x_j-pos, x_j-pos+1

Hình 3 Các đặc trưng dùng trong MST bậc một[10]

[10] Trong hình này, ký hiệu word là mục từ, pos là từ loại, +1 là về bên phải, -1 là về bên trái.

3.2 Các đặc trưng được khảo sát

Kết quả thực nghiệm trình bày trong nghiên cứu này ứng với những vectơ đặc trưng **f** đơn giản (minh họa trong Hình 3), chưa bao hàm các đặc thù của tiếng Việt đề cập trong phần 2. Cụ thể là với một cung *(i,j)*, ta sẽ xét:

+ Nhóm a và b: xét từ loại và mục từ của cung *(i,j)* trong ngữ cảnh Uni-gram và Bi-gram.
+ Ngoài ra, nếu mục từ *i* hay *j* có nhiều hơn 5 ký tự thì xét thêm đặc trưng 5-gram phía trước mục từ đó.
+ Nhóm c: bổ sung cho bối cảnh cây phụ thuộc (nhóm a và b), ta xét các mục từ trong bối cảnh câu, cụ thể là thông qua từ loại của các mục từ nằm giữa mục từ *i* và mục từ *j*, cộng thêm từ loại của các mục từ nằm bên phải và bên trái mục từ i và mục từ *j*.

Tác giả của [7] đã thử thêm bớt nhiều lần và chứng minh được bằng thực nghiệm rằng bộ đặc trưng này là hiệu quả nhất cho phân tích phụ thuộc của tiếng Anh.

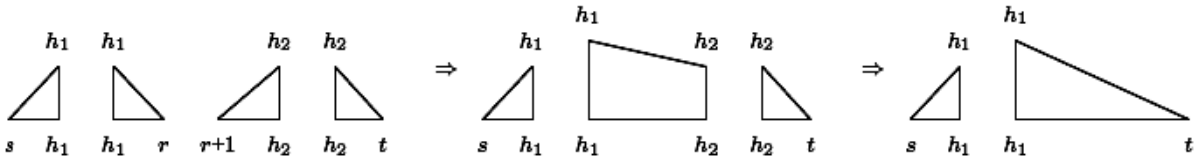
3.3 Các giải thuật tìm cây phụ thuộc

Giả sử đã thiết lập các trọng số cho đồ thị G_x (Phần 3.1).

3.3.1 Giải thuật Eisner cho trường hợp có xạ ảnh

a) Ý tưởng

Giải thuật Eisner là giải thuật phân tích biểu đồ quy hoạch động dưới-lên với độ phức tạp thời gian $O(n^3)$ nhờ một cải tiến trên giải thuật phân tích biểu đồ CYK độ phức tạp thời gian $O(n^5)$: phân tích các dependent trái của một mục từ độc lập với các dependent bên phải, và về sau sẽ kết hợp chúng.



Hình 4 Giải thuật phân tích Eisner bậc ba

Hình 4 minh họa giải thuật này. Ký hiệu r, s và t cho chỉ số bắt đầu và kết thúc của các mục biểu đồ, và h_1, h_2 cho chỉ số của head các mục biểu đồ. Ban đầu, tất cả các mục đều hoàn chỉnh, được thể hiện bằng các tam giác vuông. Giải thuật sau đó sẽ tạo ra các mục chưa hoàn chỉnh từ các mục từ nằm từ h_1 tới h_2 (với h_1 là head của h_2). Mục này đến cuối cũng sẽ được hoàn chỉnh. Cũng giống như quá trình phân tích CKY khác, những mục lớn hơn được tạo từ các cặp mục nhỏ hơn theo phương pháp dưới-lên.

b) Giả mã

Hình 5 là giả mã Ryan [7] viết cho giải thuật Eisner. Ký hiệu C[s][t][d][c] là bảng quy hoạch động lưu điểm số của cây con tốt nhất từ vị trí s đến vị trí t, $s \leq t$, với hướng d và giá trị hoàn chỉnh c. Biến $d \in \{\leftarrow, \rightarrow\}$ biểu thị hướng của cây con (nhóm các dependent trái hay phải). Nếu d=← thì t là head của cây con, nếu d=→ thì s là head của cây con. Biến $c \in \{0,1\}$

hàm ý một cây con là hoàn chỉnh (c=1, không thể thêm dependent) hay chưa hoàn chỉnh (c=0, cần được hoàn chỉnh).

Dòng được đánh dấu (*) có nghĩa là để tìm điểm số tốt nhất cho một cây con trái chưa hoàn chỉnh ta chỉ cần tìm chỉ số s≤r<t sẽ đem lại điểm số cao nhất có thể khi ghép hai cây con hoàn chỉnh.

Initialization: $C[s][s][d][c] = 0.0 \quad \forall s, d, c$
for $k : 1..n$
  for $s : 1..n$
    $t = s + k$
    if $t > n$ then break

    % First: create incomplete items
    $C[s][t][\leftarrow][0] = \max_{s \le r < t} (C[s][r][\rightarrow][1] + C[r+1][t][\leftarrow][1] + s(t,s))$ (*)
    $C[s][t][\rightarrow][0] = \max_{s \le r < t} (C[s][r][\rightarrow][1] + C[r+1][t][\leftarrow][1] + s(s,t))$

    % Second: create complete items
    $C[s][t][\leftarrow][1] = \max_{s \le r < t} (C[s][r][\leftarrow][1] + C[r][t][\leftarrow][0])$
    $C[s][t][\rightarrow][1] = \max_{s < r \le t} (C[s][r][\rightarrow][0] + C[r][t][\rightarrow][1])$

  end for
end for

Hình 5 Giả mã của giải thuật Eisner

Theo ràng buộc phải có một gốc duy nhất nằm bên trái câu, điểm số của cây tốt nhất cho cả câu là C[1][n][←][1].

3.3.2 Giải thuật Chu-Liu-Edmonds cho trường hợp không xạ ảnh

Chu-Liu-Edmonds(G, s)
  Graph $G = (V, E)$
  Edge weight function $s : E \rightarrow \mathbb{R}$
1. Let $M = \{(x^*, x) : x \in V, x^* = \arg\max_{x'} s(x', x)\}$
2. Let $G_M = (V, M)$
3. If G_M has no cycles, then it is an MST: return G_M
4. Otherwise, find a cycle C in G_M
5. Let $< G_C, c, ma >=$ contract(G, C, s)
6. Let $\boldsymbol{y} =$ Chu-Liu-Edmonds(G_C, s)
7. Find vertex $x \in C$
   such that $(x', c) \in \boldsymbol{y}$ and $ma(x', c) = x$
8. Find edge $(x'', x) \in C$
9. Find all edges $(c, x''') \in \boldsymbol{y}$
10. $\boldsymbol{y} = \boldsymbol{y} \cup \{(ma(c, x'''), x''')\}_{\forall (c,x''') \in \boldsymbol{y}}$
    $\cup\, C \cup \{(x', x)\} - \{(x'', x)\}$
11. Remove all vertices and edges in $\boldsymbol{y}$ containing c
12. return $\boldsymbol{y}$

contract$(G = (V, E), C, s)$
1. Let G_C be the subgraph of G excluding nodes in C
2. Add a node c to G_C representing cycle C
3. For $x \in V - C : \exists_{x' \in C}(x', x) \in E$
   Add edge (c, x) to G_C with
   $ma(c, x) = \arg\max_{x' \in C} s(x', x)$
   $x' = ma(c, x)$
   $s(c, x) = s(x', x)$
4. For $x \in V - C : \exists_{x' \in C}(x, x') \in E$
   Add edge (x, c) to G_C with
   $ma(x, c) = \arg\max_{x' \in C} [s(x, x') - s(a(x'), x')]$
   $x' = ma(x, c)$
   $s(x, c) = [s(x, x') - s(a(x'), x') + s(C)]$
   where $a(v)$ is the predecessor of v in C
   and $s(C) = \sum_{v \in C} s(a(v), v)$
5. return $< G_C, c, ma >$

Hình 6 Giải thuật Chu-Liu-Edmonds tìm cây khung tối đại của đồ thị định hướng

Hình 6 là phác thảo của Georgiadis cho giải thuật Chu-Liu-Edmonds. Có thể phát biểu bằng lời là: với mỗi đỉnh trong đồ thị, giải thuật chọn (bằng cách tham ăn) cạnh đi vào có trọng số cao nhất. Nếu tạo thành một cây thì đó chính là cây khung tối đại. Nếu không thì nó

phải là một chu trình. Thủ tục trong hình là để phát hiện một chu trình và rút gọn nó thành một đỉnh đơn và tính lại các trọng số cạnh đi vào và ra chu trình.

Tác giả cũng chứng minh: cây khung tối đại trên đồ thị đã rút gọn là tương đương với một cây khung tối đại trên đồ thị gốc. Vì vậy giải thuật có thể gọi đệ quy tới chính nó trên đồ thị mới. Ở dạng đơn giản nhất, giải thuật này chạy với thời gian $O(n^3)$. MST sử dụng phiên bản cải tiến của tác giả Tarjan có độ phức tạp thời gian $O(n^2)$ với đồ thị trù mật [7].

3.4 Vấn đề gán tên quan hệ phụ thuộc

3.4.1 Phương án kết hợp gán tên quan hệ phụ thuộc và tìm cây phụ thuộc

Đây là phương án dùng trong MST. Ta chỉnh sửa hàm chấm điểm cung (i,j). Việc này quy về chỉnh sửa trên vectơ đặc trưng **f** để nó chứa thông tin về tên *t* của quan hệ phụ thuộc.

$$s(i, j, t) = \mathbf{w} \cdot \mathbf{f}(i, j, t)$$

$$s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{(i,j,t)\in \boldsymbol{y}} \mathbf{w} \cdot \mathbf{f}(i, j, t)$$

Tác giả đã chứng minh được: khi đã xác định **w**, tên t thỏa mãn điều kiện

$t = \underset{t'}{\operatorname{argmax}}\ \mathbf{w} \cdot \mathbf{f}(i, j, t')$ cũng chính là tên của cung *(i,j)* trong cây khung tối đại.

Vì vậy chỉ cần xây dựng một bảng *bt(i,j)* để lưu tên tốt nhất cho từng cung và trong quá trình phân tích thì dùng *s(i,j,bt(i,j))*.

Phương án này tuy tận dụng được tri thức chung để suy luận ra cả cây phụ thuộc và tên các quan hệ nhưng về cơ bản lại bị giới hạn bởi phạm vi phân tích địa phương, cụ thể là chỉ xem xét đặc trưng của các cạnh đơn lẻ trên cây. Ngoài ra, với độ phức tạp $O(n^3 + |T|n^2)$ trong trường hợp có xạ ảnh và $O(|T|n^2)$ trong trường hợp không xạ ảnh thì phương án này không tối ưu khi số lượng T tên các quan hệ phụ thuộc rất lớn.

3.4.2 Phương án gán tên quan hệ phụ thuộc sau khi tìm ra cây phụ thuộc

Ở bài toán này, ta đi tìm tên cho từng cung khi đã có cây ***y*** trên câu ***x***. Một mô hình hiệu quả mà tác giả trong [7] đã thử nghiệm là gán tên cho một chuỗi cung, ứng với một chuỗi dependent của mục từ i:

Gọi $x_{j1}, \ldots, x_{jM}$ là các dependent của x_i; tương ứng là các tên quan hệ phụ thuộc $t_{(i,j1)}, \ldots, t_{(i,jM)}$. Chuỗi tên tốt nhất ứng với mục từ x_i là

$$(t_{(i,j1)}, \ldots, t_{(i,jM)}) = \boldsymbol{t}_{(i,)} = \underset{\bar{t}}{\operatorname{argmax}}\, s(\bar{t}\ , i, \boldsymbol{y}, \boldsymbol{x})$$

Vận dụng phân tích Markov bậc 1 cho hàm chấm điểm

$$\boldsymbol{t}(i,) = \underset{\bar{t}}{\operatorname{argmax}} \sum_{m=2}^{M} s(t_{(i,jm)}, t_{(i,jm-1)}, i, \boldsymbol{y}, \boldsymbol{x})$$

Sau đó dùng giải thuật Viterbi để tìm ra chuỗi tốt nhất.

Hàm chấm điểm gắn với một vectơ đặc trưng gồm: đặc trưng cạnh đang xét, đặc trưng của các cạnh khác cùng nút cha, đặc trưng ngữ cảnh câu.

3.5 Pha học mô hình trọng số bằng phương pháp MIRA

3.5.1 Lý do chọn MIRA

Các giải thuật đề cập phía trên đều phải dựa vào vectơ trọng số **w**. Vectơ này được học từ dữ liệu huấn luyện bằng phương pháp học máy MIRA. Các đặc tính của MIRA khiến nó phù hợp với bài toán phân tích cú pháp phụ thuộc và tiếng Việt là:

1) Nó là phương pháp học máy phân biệt[11].
2) Khác với các phương pháp tốt nhất hiện nay (như CRFs[12], M^3Ns[13]) đều học theo lô, MIRA học online. Đặc tính 1 và 2 giúp tạo ra các mô hình hoạt động tốt trong điều kiện thiếu dữ liệu tiếng Việt.
3) Phân lớp được chia thành nhiều bài toán con, trong số đó có bài toán học có cấu trúc bằng phân lớp tuyến tính. Phân tích phụ thuộc là bài toán học có cấu trúc, MIRA nằm trong số ít các phương pháp học máy giải quyết hiệu quả bài toán này.
4) Khi đã có mô hình, bước suy luận của MIRA dựa trên giải thuật Hildreth giải bài toán quy hoạch bậc hai. Nó không cần tới các giải thuật forward-backward, inside-outside phức tạp như CRFs hay các tính toán về phân phối và tối ưu phức tạp của CRFs và M^3Ns [7].

3.5.2 Cách tiếp cận của MIRA

MIRA là online SVMs[14] nhờ dùng phép xấp xỉ.

SVMs cho bài toán học có cấu trúc	**MIRA** (mỗi lần cập nhật **w** ta chọn vectơ trọng số mới gần với vectơ cũ nhất)
tìm $\min\|\|\mathbf{w}\|\|$ với những $s(\boldsymbol{x},\boldsymbol{y}) - s(\boldsymbol{x},\boldsymbol{y'}) \geq L(\boldsymbol{y},\boldsymbol{y'})$ cho $\forall\ (\boldsymbol{x},\boldsymbol{y}) \in T, \boldsymbol{y'} \in parses(\boldsymbol{x})$	$\mathbf{w}^{(i+1)} = \text{argmin}_{\mathbf{w}^*}\|\|\mathbf{w}^* - \mathbf{w}^{(i)}\|\|$ với những $s(\boldsymbol{x}_t,\boldsymbol{y}_t) - s(\boldsymbol{x}_t,\boldsymbol{y'}) \geq L(\boldsymbol{y}_t,\boldsymbol{y'})$ ứng với $\mathbf{w}^*$ cho $\forall\ \boldsymbol{y'} \in parses(\boldsymbol{x}_t)$

Hình 7 So sánh MIRA và SVMs

Trong đó $L(\boldsymbol{y},\boldsymbol{y'})$ là hàm xác định độ sai sót của $\mathbf{y'}$ so với **y**, tính bằng số mục từ trên $\mathbf{y'}$ có cung đi vào khác **y**; $parses(\boldsymbol{x})$ là không gian tất cả các cây phụ thuộc có thể ứng với câu $\boldsymbol{x}$.

3.5.3 Dùng k-best MIRA xấp xỉ MIRA để tránh số nhãn tăng theo hàm mũ

Chỉ áp dụng ràng buộc về lề cho *k* cây phụ thuộc $\boldsymbol{y'}$ có $s(\boldsymbol{x},\boldsymbol{y'})$ cao nhất.

[11] Thuật ngữ tiếng Anh là "discriminative learning"
[12] CRFs là viết tắt của "Conditional Random Fields"
[13] M3Ns là viết tắt của "Maximum Margin Markov Networks"
[14] SVMs là viết tắt của "Support Vector Machines"

$\mathbf{w}^{(i+1)} = \text{argmin}_{\mathbf{w}^*} \|\mathbf{w}^* - \mathbf{w}^{(i)}\|$

với những $s(\boldsymbol{x}_t,\boldsymbol{y}_t) - s(\boldsymbol{x}_t,\boldsymbol{y}') \geq L(\boldsymbol{y}_t,\boldsymbol{y}')$ ứng với $\mathbf{w}^*$

cho những $\boldsymbol{y}' \in best_k(\boldsymbol{x}_t, \mathbf{w}^{(i)})$

Hình 8 k-best MIRA

Hình 8 là k-best MIRA tổng quát, trong MST tác giả chỉ sử dụng k=1.

4 Hiệu chỉnh kết quả của MST

Để nâng cao độ chính xác bộ phân tích cú pháp phụ thuộc, chúng tôi thực hiện các luật hiệu chỉnh cây trên đầu ra của MST. Giải pháp sử dụng ở đây là tiếp cận Giuseppe Attardi đề xuất trong [9]: xem các luật hiệu chỉnh này như các nhãn phân loại, nhờ vậy đưa bài toán hiệu chỉnh đầu ra của một bộ phân tích cú pháp phụ thuộc về bài toán phân lớp.

4.1 Đưa về bài toán phân lớp

4.1.1 Phép hiệu chỉnh nguyên tử

Tác giả đưa ra một tập phép hiệu chỉnh nguyên tử nhất định trên cây (minh họa trong Bảng 2) , quy về hiệu chỉnh head của một mục từ x_i (vì trong cây phụ thuộc, mỗi mục từ chỉ có 1 head).

Bảng 2 Các phép hiệu chỉnh nguyên tử trên cây

Ký hiệu	Phép hiệu chỉnh nguyên tử
r	đặt head ở root
u	đặt head lên nút cha của head
-n	đặt head sang mục từ thứ n bên trái
+n	đặt head sang mục từ thứ n bên phải
[	đặt head bằng head của thành phần liền trước
]	đặt head bằng head của thành phần liền sau
>	đặt head bằng mục từ đầu tiên trong thành phần liền trước
<	đặt head bằng mục từ đầu tiên trong thành phần liền sau
d--	dịch head xuống con trái nhất của nó
d++	dịch head xuống con phải nhất của nó
d-1	dịch head xuống con trái đầu tiên của nó
d+1	dịch head xuống con phải đầu tiên của nó
dP	dịch head xuống mục từ có từ loại P

4.1.2 Luật hiệu chỉnh

Ta thường phải áp dụng nhiều phép hiệu chỉnh nguyên tử trên một mục từ để được kết quả mong muốn, vì vậy tác giả đưa ra khái niệm luật hiệu chỉnh. Luật hiệu chỉnh là một chuỗi không quá 4 phép hiệu chỉnh nguyên tử.

4.1.3 Phát biểu hình thức bài toán

Gọi $\boldsymbol{y}=(\boldsymbol{x},E)$ là cây phụ thuộc cho câu $\boldsymbol{x}$. Một luật hiệu chỉnh là một ánh xạ $r: E \rightarrow E$ biến cung $e = (i,t,j)$ thành cung $e'=(i,t,s)$. Cây sau khi hiệu chỉnh là $r(\boldsymbol{y})=(\boldsymbol{x},E')$ trong đó $E'=\{r(e):$

$e \in E\}$. Bằng cách xem mỗi luật hiệu chỉnh là một nhãn, ta đưa bài toán hiệu chỉnh cây về tìm một chuỗi nhãn cho các mục từ trong câu $\boldsymbol{x}$. Mỗi mục từ một nhãn.

Có hai lựa chọn cho việc dùng E': 1-áp dụng đồng thời tất cả các luật, 2-áp dụng từng luật riêng lẻ tạo ra cây trung gian, rồi lại tiếp tục tìm luật hiệu chỉnh trên cây trung gian này. Do cách 2 có thể tạo những dạng trung gian không phải là cây nên nghiên cứu chỉ dừng lại ở cách 1.

4.2 Học mô hình bằng Perceptron đa lớp

4.2.1 Ý tưởng

Dữ liệu huấn luyện là các cặp vectơ đặc trưng $\mathbf{f_i}$ và luật r_i ($\mathbf{f_i}$ có thể sinh từ $\boldsymbol{y_i}$; r_i có thể sinh từ cặp cây gồm cây gán nhãn bằng tay ($\boldsymbol{y}_{\mathrm{Mi}}$) và cây đầu ra của MST ($\boldsymbol{y_i}$)). Cần chú ý là mỗi cặp ($\mathbf{f}$, r) ứng với một mục từ trên cây.

Mục đích của chúng ta là học hàm C: F→R. R là không gian các luật hiệu chỉnh, với r_1 ký hiệu cho phép giữ nguyên cây.

Dùng Perceptron đa lớp để thực hiện nhiệm vụ này. Mỗi luật r sẽ có một vectơ trọng số $\mathbf{w}_\mathrm{r}$ tương ứng. Bài toán quy về học các vectơ trọng số $\mathbf{w}_\mathrm{r}$

$$C(\mathbf{f}) = \underset{\mathrm{r}}{\operatorname{argmax}} < \mathbf{f}, \mathbf{w_r} >$$

$\mathbf{w}_\mathrm{r}$ học được trong quá trình huấn luyện được chuẩn hóa bằng cách lấy trung bình

$\mathbf{w}_\mathrm{r} = \dfrac{1}{T}\sum_{t=1}^{T} \mathbf{w}_r^t$ với T là số mẫu huấn luyện.

4.2.2 Các đặc trưng khảo sát

Các đặc trưng dùng để huấn luyện ra mô hình hiệu chỉnh là mục từ, từ loại và tên quan hệ phụ thuộc của các đối tượng: nút hiện tại, cha, ông, cụ, con, mục từ trước, mục từ sau của nút hiện tại. Ngoài ra những cặp đặc trưng xuất hiện hơn 10 lần trong dữ liệu huấn luyện cũng được xét đến.

4.3 Hiệu chỉnh cây khi đã có mô hình huấn luyện

Khi đưa vào $\boldsymbol{y}_q$ nào đó vào bộ hiệu chỉnh, ban đầu các vectơ $\mathbf{f}$ ứng với các mục từ trong $\boldsymbol{y}_q$ sẽ được sinh ra. Công việc còn lại là kết hợp $\mathbf{f}$ và các $\mathbf{w}_r$ để hiệu chỉnh cây ứng với từng mục từ. Độ phức tạp của bộ hiệu chỉnh này là O(n).

5 Thực nghiệm trên tiếng Việt

5.1 Dữ liệu thực nghiệm

Kho ngữ liệu dùng cho thực nghiệm gồm 450 câu tiếng Việt trích ngẫu nhiên từ các bài báo ở nhiều chuyên mục khác nhau của báo điện tử Vietnamnet.

Dữ liệu được tiền xử lý (sửa lỗi chính tả), gán nhãn bằng tay các thông tin về từ loại và quan hệ phụ thuộc và định dạng theo chuẩn của Hội thảo quốc tế CoNLL-X 2006 [8].

Thông tin về bộ nhãn từ loại và bộ nhãn quan hệ phụ thuộc được mô tả chi tiết hơn trong tài liệu đi kèm kho ngữ liệu.

1	Tôi	tôi	P	P	_	3	NP-SBJ	_	_
2	sẽ	sẽ	R	R	_	3	ADVP	_	_
3	tiếp_tục	tiếp_tục	MD	MD	_	7	S	_	_
4	giúp	giúp	V	V	_	3	VP	_	_
5	,	,	SYM	SYM	_	3	DEP	_	_
6	Tâm	tâm	NP	NP	_	7	NP-SBJ	_	_
7	nói	nói	V	V	_	0	ROOT	_	_
8	,	,	SYM	SYM	_	7	DEP	_	_
9	tới	tới	IN	IN	_	3	PP	_	_
10	khi	khi	NN	NN	_	9	SBAR	_	_
11	họ	họ	P	P	_	13	NP-SBJ	_	_
12	không	không	R	R	_	13	ADVP	_	_
13	cần	cần	V	V	_	10	S	_	_
14	tôi	tôi	P	P	_	13	NP-OBJ	_	_
15	nữa	nữa	R	R	_	13	ADVP	_	_
16	.	.	SYM	SYM	_	7	DEP	_	_

Hình 9 Ví dụ dữ liệu theo chuẩn CONLL-X 2006 của câu trong Hình 1

5.2 Phương pháp đánh giá và thước đo

5.2.1 Phương pháp đánh giá

Do dữ liệu đòi hỏi quá trình xử lý bằng tay công phu nên chúng tôi chưa xây dựng được nhiều. Để kết quả đánh giá là chính xác nhất với 450 câu xây dựng được, chúng tôi đề xuất vận dụng linh hoạt phương pháp đánh giá chéo[15].

a) Phương pháp đánh giá MST

Chia dữ liệu thành 10 phần để đánh giá chéo.

b) Phương pháp đánh giá MST sau khi hiệu chỉnh

Chia dữ liệu thành 10 phần, ký hiệu là T1,..,T10. Để kiểm thử hiệu chỉnh trên T1, ta thực hiện quay vòng MST trên 9 phần còn lại (huấn luyện MST trên 8 phần và kiểm thử MST trên 1 phần) rồi gộp kết quả kiểm thử lại làm dữ liệu huấn luyện bộ hiệu chỉnh.

Làm tương tự với 9 phần còn lại và chia trung bình để được độ chính xác.

5.2.2 Thước đo

Chúng tôi dùng hai thước đo điển hình cho bài toán phân tích phụ thuộc là: UAS (viết tắt của Unlabeled Attachment Score) là độ chính xác khi chưa tính đến tên quan hệ phụ thuộc; và LAS (viết tắt của Labeled Attachment Score) là độ chính xác khi đã xét cả tên quan hệ phụ thuộc.

5.3 Kết quả thực nghiệm

Bảng 3 So sánh kết quả MST khi trước và sau hiệu chỉnh

Phương pháp	**UAS**	**LAS**
MST bậc 1	67.70%	63.11%
MST bậc 1 + hiệu chỉnh	66.49%	61.76%

[15] Thuật ngữ tiếng Anh là "cross validation"

Như vậy sau khi hiệu chỉnh độ chính xác của bộ phân tích lại giảm đi. Điều này là do các đặc trưng dùng cho bước hiệu chỉnh khác với các đặc trưng dùng trong MST và do dữ liệu huấn luyện quá ít (kho ngữ liệu chỉ chứa khoảng 2200 từ tố phân biệt trong khi từ điển tiếng Việt có khoảng 11 nghìn từ). Tuy là hệ thống thử nghiệm đầu tiên trên tiếng Việt và có hạn chế về kho ngữ liệu, độ chính xác trong khoảng này khá gần với LAS từ 70.98% đến 80.29% và UAS từ 75.53% đến 84.80% của MST trên một số ngôn ngữ liệt kê trong [12] cho thấy MST là một hướng khả thi giải quyết bài toán phân tích cú pháp phụ thuộc tiếng Việt.

6 Kết luận

Là một trong những công trình đầu tiên nghiên cứu về phân tích tự động cú pháp phụ thuộc cho câu tiếng Việt, bài báo đã trình bày chi tiết về bài toán. Về mặt ngôn ngữ, chúng tôi đã tổng hợp những đặc thù của ngữ pháp tiếng Việt có thể mô hình hóa để đưa thêm vào các vectơ đặc trưng nhằm nâng cao độ chính xác bộ phân tích. Bài báo cũng đề xuất một mô hình phân tích phụ thuộc cho tiếng Việt dựa trên kết hợp hai mô hình cho kết quả khả quan trên tiếng Anh: mô hình MST và mô hình hiệu chỉnh cây phụ thuộc. Kết quả thử nghiệm ban đầu trên kho ngữ liệu tiếng Việt chúng tôi đã xây dựng theo chuẩn CONLL-X 2006 cho thấy: độ chính xác sau khi hiệu chỉnh giảm khoảng 2%. Trong tương lai, có thể dùng chính các đặc trưng của MST cho phần hiệu chỉnh để hệ thống nhất quán hơn. Ta cũng có thể thay thế Perceptron đa lớp trong phần hiệu chỉnh bằng MIRA để tận dụng điểm mạnh phương pháp học máy phân biệt này cho học có cấu trúc và khả năng tương thích của nó với hạn chế tài nguyên ngôn ngữ - là một vấn đề lớn trong xử lý tiếng Việt hiện nay.

Tài liệu tham khảo

[1] Wikipedia (truy cập ngày 24/5/2008). *Vietnamese syntax*. http://en.wikipedia.org/wiki/Vietnamese_syntax

[2] Wikipedia (truy cập ngày 24/5/2008). *Analytic language*. http://en.wikipedia.org/wiki/Analytic_language

[3] Wikipedia (truy cập ngày 24/5/2008). *Isolating language*. http://en.wikipedia.org/wiki/Isolating_language

[4] Wikipedia (truy cập ngày 24/5/2008). *Word order*. http://en.wikipedia.org/wiki/Word_order

[5] Ryan McDonald, Joakim Nivre (2007). *Introduction to Data-Driven Dependency Parsing*. Introductory Course, ESSLLI 2007.

[6] Diệp Quang Ban (2005). *Ngữ pháp tiếng Việt*. NXB Giáo Dục.

[7] Ryan McDonald (2006). *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. University of Pennsylvania.

[8] CoNLL-X Shared Task: *Multi-lingual Dependency Parsing*. http://nextens.uvt.nl/~conll/

[9] G. Attardi, M. Ciaramita (2007). *Tree revision learning for dependency parsing*. Proceedings of HLT-NAACL 2007, Rochester.

[10] EAGLES (1996). *Recommendations for the Syntactic Annotation of Corpora*. http://www.ilc.cnr.it/EAGLES96/segsasg1/segsasg1.html

[11] K. Crammer and Y. Singer (2003*). Ultraconservative Online Algorithms for Multiclass Problems*. Journal of Machine Learning Research 3: pp.951-991.

[12] J. Nivre, J. Hall, S. Kubler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret (2007). *The CoNLL 2007 Shared Task on Dependency Parsing*. Conference on Empirical Methods in Natural Language Processing and Natural Language Learning.