MỘT SỐ CẢI TIẾN CHO GIẢI THUẬT EARLEY ĐỂ PHÂN TÍCH CÚ PHÁP TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN

LÊ VIẾT MẪN

1. MỞ ĐẦU

Giải thuật Earley [1, 2] là một trong số những giải thuật được sử dụng để phân tích cú pháp trong xử lý ngôn ngữ tự nhiên. Nó là một giải thuật tổng quát, có thể phân tích bất kỳ văn phạm phi ngữ cảnh nào. Nhưng giải thuật này vẫn còn nhiều hạn chế cần khắc phục.

Đầu tiên, Kilbury [3] đã nhận xét rằng giải thuật Earley là không hiệu quả trong xử lý ngôn ngữ tự nhiên. Vì nó phải duyệt qua quá nhiều luật sinh không cần thiết (trong bài này chúng tôi sẽ gọi là luật dư thừa) trong giai đoạn đoán nhận (predict). Đối với các văn phạm lớn, điều này sẽ làm giảm đáng kể tiến độ xử lý.

Mặt khác, giải thuật Earley trong xử lý ngôn ngữ tự nhiên còn gặp phải hiện tượng bùng nổ tổ hợp, bởi vì muốn phân tích một câu của ngôn ngữ tự nhiên thì bộ phân tích phải kiểm tra từ vài chuỗi đến hàng chục, hàng trăm chuỗi từ loại khác nhau. TS. Phan Thị Tươi đã nêu lên vấn đề trên trong [6] và đồng thời cũng nêu lên hướng giải quyết cho các giải thuật Earley và Chart. Nhưng cải tiến cho giải thuật Earley trong [6] chỉ hiệu quả trong trường hợp câu nhập vào là đúng. Còn nếu câu nhập vào là sai thì giải thuật không hiệu quả.

Với những điều như trên, trong bài này, chúng tôi sẽ trình bày giải thuật Earley cải tiến loại bỏ hoàn toàn việc phải duyệt qua các luật sinh dư thừa. Và đồng thời, chúng tôi sẽ bàn tới hướng giải quyết hiện tượng bùng nổ tổ hợp dựa trên cải tiến trong [6].

Bài báo sẽ được tổ chức như sau: Phần 2 – Chúng tôi sẽ trình bày giải thuật Earley. Phần này còn bao gồm những nhận xét và một ví dụ cho giải thuật Earley. Phần 3 – Chúng tôi sẽ nói đến những luật dư thừa mà giải thuật Earley phải duyệt qua và giải thuật Earley cải tiến. Đồng thời, chúng tôi đưa ra một đề nghị về dạng luật sinh để hỗ trợ tăng tốc độ tiến trình xử lý. Phần 4 – Chúng tôi sẽ bàn về hiện tượng bùng nổ tổ hợp, phương pháp giải quyết trong [6] và cuối cùng sẽ đề ra phương pháp giải quyết trường hợp câu nhập sai mà phương pháp trong [6] còn thiếu. Phần 5 – Chúng tôi sẽ thực hiện một giả mã cho giải thuật Earley cải tiến.

2. GIẢI THUẬT EARLEY

Cho G=(V, W, S, P) là một văn phạm phi ngữ cảnh, và $w=a_1 \ldots a_n \in W^*$. Khi đó, $A \rightarrow \alpha \bullet \beta$ là một luật có chấm khi $A \rightarrow \alpha\beta \in P$. Giải thuật Earley được biểu diễn thông qua việc xây dựng bảng chứa tập các luật có chấm. Người ta xây dựng bảng Earley với các cột I_i (i=0..n), cột I_0 nhận giá trị khởi tạo, n là độ dài của chuỗi từ

manleviet@yahoo.com.hk

loại nhập. Mỗi ô sẽ có các giá trị: ***giá trị gốc*** để biết luật đó phát sinh từ cột nào, và ***luật có chấm***.

Ví dụ: Giá trị gốc Luật có chấm

0 S→•CN VN

1 VN→ĐT• DT

2.1. Giải thuật

Giải thuật bao gồm ba bước:

(1) Đoán nhận (Predict): Tại cột I_i

Đối với các luật có ký tự không kết thúc ở bên phải dấu chấm, ta thêm các luật mới mà ký tự không kết thúc đó là vế trái của các luật. Giá trị gốc là i.

Tức là, với mỗi $[A\rightarrow\alpha\bullet B\beta, j]$ trong I_i ta thêm $[B\rightarrow\bullet\gamma, i]$ vào I_i nếu $B\rightarrow\gamma \in P$.

(2) Duyệt (Scan): Tại cột I_i

Đối với các luật mà ký tụ kết thúc ở bên phải dấu chấm, luật này sẽ được chuyển sang cột I_{i+1} với dấu chấm được dịch ra sau ký tự kết thúc.

Tức là, với $[A\rightarrow\alpha\bullet a\beta, j]$ sẽ được đổi thành $[A\rightarrow\alpha a\bullet\beta, i]$ trong cột I_{i+1}.

(3) Hoàn thiện (Complete): Tại cột I_i

Khi có luật $[A\rightarrow\alpha\bullet, j]$ thì sao chép và đổi $[B\rightarrow\alpha\bullet A\beta, k]$ trong cột I_j thành $[B\rightarrow\alpha A\bullet\beta, k]$ trong cột I_i.

2.2. Nhận xét

a. Đây là dạng phân tích từ trên xuống bởi vì chúng ta bắt đầu với việc đoán nhận. Nếu chúng ta thay đổi thứ tự trên, chúng ta sẽ có kiểu phân tích từ dưới lên.

b. Thông thường, phân tích từ trên xuống có vấn đề với đệ qui trái, nhưng thuật toán Earley đã giải quyết bằng cách:

Mỗi luật giống nhau sẽ chỉ xuất hiện một lần trong mỗi cột. Có nghĩa là trong các bước thực hiện thuật toán, trước khi thêm một luật vào bảng thì phải kiểm tra xem nó có trùng với luật nào đã có trong cột cần thêm vào không. Nếu không thì thêm vào, còn có thì không thêm vào.

c. Chuỗi từ loại là sai cú pháp khi ta đã duyệt qua hết các luật trong I_i mà I_{i+1} rỗng và chưa thể kết thúc bảng hợp lệ.

d. Chuỗi từ loại là đúng cú pháp khi kết thúc chuỗi từ loại mà ta có luật khởi tạo tại cột cuối cùng.

Nói chung, chuỗi đúng khi tại điểm kết thúc chuỗi nhập, mà dấu chấm đã di chuyển ra sau ký tự bắt đầu S.

e. Với việc sử dụng giá trị đoán nhận trước ta có thể giúp tránh dư thừa.

Ví dụ, ta có luật VN → VN •BN tại vị trí kết thúc chuỗi nhập. Thông thường, ta sẽ đi đoán nhận BN, nhưng trong trường hợp này là không nên, ta chỉ nên làm như thế nếu còn từ trong chuỗi nhập.

manleviet@yahoo.com.hk

Mặt khác, giá trị đoán nhận trước cũng gây ra sự phức tạp, và tăng số lượng luật được lưu trữ.

f. Độ phức tạp thời gian của thuật toán là $O(n^3)$, với n là độ dài chuỗi nhập (bằng số lượng cột của bảng).

2.3. Ví dụ

Cho văn phạm G với các luật sinh sau:

S → CN VN	DT → mẹ
CN → DL DT	DT → con
CN → DT	DT → chân
VN → ĐT CN	DL → cái
VN → VN BN	ĐT → rửa
BN → GT CN	GT → cho

Chú thích: CN – Chủ ngữ
VN – Vị ngữ
DT – Danh từ
DL – Danh từ chỉ loại
ĐT – Động từ
BN – Bổ ngữ
GT – Giới từ

Và chuỗi nhập là: mẹ rửa cái chân cho con.

Với thuật toán Earley được trình bày như trên ta sẽ có bảng như sau:

0	1	2	3	4	5	6
0 ROOT .S	0 DT mẹ.	1 ĐT rửa.	2 DL cái.	3 DT chân.	4 GT cho.	5 DT con.
0 S .CN VN	0 CN DT.	1 VN ĐT .CN	2 CN DL .DT	2 CN DL DT.	4 BN GT .CN	5 CN DT.
0 CN .DL DT	0 S CN .VN	2 CN .DL DT	3 DT .mẹ	1 VN ĐT CN.	5 CN .DL DT	4 BN GT CN.
0 CN .DT	1 VN .ĐT CN	2 CN .DT	3 DT .con	0 S CN VN.	5 CN .DT	1 VN VN BN.
0 DL .cái	1 VN .VN BN	2 DL .cái	3 DT .chân	1 VN VN .BN	5 DL .cái	0 S CN VN.
0 DT .mẹ	1 ĐT .rửa	2 DT .mẹ		0 ROOT S.	5 DT .mẹ	0 ROOT S.
0 DT .con		2 DT .con		4 BN .GT CN	5 DT .con	
0 DT .chân		2 DT. Chân		4 GT .cho	5 DT .chân	

3. GIẢI QUYẾT VẤN ĐỀ LUẬT DƯ THỪA TRONG GIẢI THUẬT EARLEY

Với giải thuật đã nêu ở trên, ta có thể nhận thấy có rất nhiều luật dư thừa vẫn được lưu trữ trong bảng Earley. Như thế đồng nghĩa với việc phải duyệt qua quá nhiều luật dư thừa như Kilbury đã nhận xét [3]. Qua nghiên cứu, chúng tôi nhận thấy các luật dư thừa có dạng như sau:

- thứ nhất, luật chỉ có một kí tự kết thúc ở vế trái mà không khớp với giá trị đoán nhận
- thứ hai, luật không dẫn đến đệ qui trái.

manleviet@yahoo.com.hk

3.1. Dạng luật sinh

Để giải quyết triệt để vấn đề này, chúng tôi đã thực hiện lập luật sinh với dạng riêng. Tất cả các luật sinh đều thuộc vào một trong hai dạng sau:

- $A \rightarrow \alpha, \alpha \in V^*$
- $A \rightarrow a, a \in W$

Dạng luật trên vẫn phù hợp với văn phạm phi ngữ cảnh.

Với các luật sinh thuộc vào hai dạng trên thì các luật dư thừa sẽ là:

- Luật có vế phải chỉ là một ký tự kết thúc ($A \rightarrow a$)
- Luật có vế phải là các ký tự không kết thúc mà không dẫn đến đệ qui trái. ($A \rightarrow B\alpha$)

3.2. Giải thuật cải tiến

Với dạng luật như trên, giải thuật được cải tiến chỉ còn lại hai bước như sau:

(1) Đoán nhận:

Với mỗi $[A \rightarrow \alpha \bullet B\beta, j]$ trong I_i.

Lấy các luật trong từ điển luật sinh.

Duyệt qua các luật dạng $B \rightarrow a$, nếu khớp với giá trị đoán nhận thì đưa luật khớp cùng với các luật dạng $B \rightarrow B\alpha$ vào bảng Earley.

Ngược lại nếu không so khớp với giá trị đoán nhận thì đưa toàn bộ các luật dạng $B \rightarrow \alpha$ vào bảng Earley.

(2) Hoàn thiện:

Như giải thuật Earley cũ.

Giải thuật cải tiến ở trên chỉ còn hai bước, bước quét đã được bỏ đi là do dạng luật sinh và giai đoạn đoán nhận mới đã giải quyết luôn bước này. Giải thuật cải tiến đã giải quyết được vấn đề luật dư thừa, nó không còn phải duyệt qua các luật không cần thiết trong phân tích cú pháp nữa. Như thế, sẽ cải thiện tốc độ của tiến trình xử lý nhiều hơn. Ngoài ra, còn giảm không gian lưu trữ. Nhưng giải thuật cải tiến vẫn có độ phức tạp thời gian là $O(n^3)$. Vì giải thuật chỉ mới thay đổi nội dung bên trong cấu trúc của giải thuật Earley chứ chưa thay đổi được cấu trúc của giải thuật nên độ phức tạp thời gian vẫn là như cũ.

3.3. Ví dụ

Với ví dụ như trong phần 2.3, chúng ta sẽ có bảng như sau:

0	1	2	3	4	5	6
0 ROOT .S	0 DT mẹ.	1 ĐT rửa.	2 DL cái.	3 DT chân.	4 GT cho.	5 DT con.
0 S .CN VN	0 CN DT.	1 VN ĐT .CN	2 CN DL .DT	2 CN DL DT.	4 BN GT .CN	5 CN DT.
0 CN .DL DT	0 S CN .VN	2 CN .DL DT	3 DT .chân	1 VN ĐT CN.	5 CN .DL DT	4 BN GT CN.
0 CN .DT	1 VN .ĐT CN	2 CN .DT		0 S CN VN.	5 CN .DT	1 VN VN BN.
0 DT .mẹ	1 VN .VN BN	2 DL .cái		1 VN VN .BN	5 DT .con	0 S CN VN.
	1 ĐT .rửa			0 ROOT S.		0 ROOT S.
				4 BN .GT CN		
				4 GT .cho		

manleviet@yahoo.com.hk

4. GIẢI QUYẾT VẤN ĐỀ BÙNG NỔ TỔ HỢP

Hiện tượng bùng nổ tổ hợp xảy ra do một từ có thể thuộc vào nhiều từ loại khác nhau. Nhưng một đặc điểm dễ nhận thấy của các tổ hợp từ loại được sinh ra từ một câu là luôn có những đoạn con từ loại giống nhau trên các tổ hợp từ loại.

Dựa vào đặc điểm này PTS Phan Thị Tươi trong [6] đã nêu một phương pháp để tăng tốc độ phân tích cú pháp như sau:

Nếu bộ phân tích thất bại khi đang kiểm tra một chuỗi, thì nó sẽ so trùng các chuỗi còn lại với đoạn vừa kiểm tra thành công và sẽ tiếp tục quy trình phân tích ở vị trí của một chuỗi khác có chuỗi con dài nhất trùng với đoạn đã phân tích. Quá trình này được lặp lại cho tới khi bộ phân tích duyệt qua hết một chuỗi nào đó. Lúc đó, câu nhập được xác nhận là đúng cú pháp. Ngược lại khi đi đến chuỗi cuối cùng mà vẫn không phân tích thành công thì bộ phân tích sẽ kết luận rằng câu nhập vào không đúng cú pháp.

Đây là một phương pháp hay, nó giúp ta tránh phải phân tích lại những gì đã phân tích rồi. Nhưng phương pháp trên lại chỉ hiệu quả trong trường hợp câu nhập vào là đúng, còn trong trường hợp câu nhập vào là sai thì không hiệu quả.

Qua nghiên cứu, chúng tôi còn nhận thấy còn có hiện tượng trùng một phần (chỉ một phần ngắn của đoạn đã kiểm tra thành công) bên cạnh hiện tượng trùng cả đoạn như đã nói ở trên. Điều này cho thấy ta có thể giảm thêm được một số bước phân tích nữa.

Thừa kế từ phương pháp trong [6] và bổ sung điều mới phát hiện, chúng tôi đưa ra phương pháp như sau:

Ta sắp xếp các chuỗi tổ hợp từ loại theo thứ tự. Như thế, chuỗi ngay sau chuỗi đang xét sẽ là chuỗi có khả năng trùng đoạn đã phân tích. Khi kiểm tra một chuỗi thất bại ta chỉ việc so khớp đoạn vừa kiểm tra thành công với chuỗi ngay sau nó và lấy số từ loại so khớp liên tục bắt đầu từ đầu chuỗi. Việc phân tích sẽ được thực hiện tiếp với chuỗi ngay sau tại vị trí đầu tiên không so khớp.

Bước đầu tiên của giải thuật trên là sắp xếp các chuỗi tổ hợp từ loại, chúng tôi thực hiện bước này là để có thể sử dụng phương pháp trong [6]. Khi đã được sắp xếp, rõ ràng chuỗi từ loại ngay sau chuỗi vừa kiểm tra có xác suất trùng đoạn vừa kiểm tra thành công là lớn nhất. Như thế, ta chỉ việc thực hiện tiếp tục phân tích với chuỗi từ loại tiếp ngay sau. Giải thuật cải tiến của chúng tôi đã thừa kế chọn vẹn phương pháp trong [6], đồng thời còn thay thế công đoạn tìm kiếm chuỗi trùng khớp bằng chỉ một bước đơn giản là tăng giá trị duyệt tổ hợp chuỗi từ loại lên một. Đây là một cải tiến đáng kể.

manleviet@yahoo.com.hk

5. CÀI ĐẶT

Với những cải tiến ở trên, chúng tôi đã cài đặt theo giả mã như sau:

Bảng được xây dựng thành một mảng hai chiều. Mỗi ô là một record với hai trường. Một trường số nguyên nhận giá trị gốc, một trường kiểu string nhận luật có chấm.

Giá trị i, j được khai báo toàn cục.

Vào: tổ hợp chuỗi từ loại, tập luật

Ra: chỉ số tổ hợp đúng, là -1 nếu tất cả đều sai

Phần chương trình:

```
Function KiemTraKetThucDung:Boolean;
Begin
    Thực hiện hoàn thiện và đoán nhận cho các luật trong cột cuối cùng
    Nếu gặp luật ROOT →•S thì KiemTraKetThucDung:=True;
    Ngược lại, KiemTraKetThucDung:=False;
End;

Function DoanNhan(KKT):Boolean;
Begin
    Lấy các luật sinh dạng KKT→a
    If a = giá trị đoán nhận then
        Đưa luật KKT→a so khớp vào bảng tại cột i
        Đưa luật KKT→a trên vào bảng tại cột i+1 với giá trị gốc là i
        Lấy các luật dạng KKT→KKTα đưa vào bảng tại cột i
        j:=j+1;
        DoanNhan:=True;
        Exit;
    End if
    Đưa các luật dạng KKT→α trong từ điển vào bảng.
    i:=i+1;
End;

Procedure HoanThien;
Begin
    Lấy vế trái của luật và giá trị gốc.
    Duyệt qua cột có chỉ số là giá trị gốc.
        Sao chép luật có dấu chấm ở ngay trước vế trái của luật đang xét mà không
        trùng luật và giá trị gốc với những luật đã có ở cột i
    i:=i+1;
```

manleviet@yahoo.com.hk

```
End;

Procedure KhoiTao;
Begin
    Đưa luật ROOT→ •S vào ô (i=1,j=0)
End;

Function NSK(SK, WCSet: String):Integer;
Begin
    NSK = 0;
    tu = Lấy từ loại đầu tiên trong SK;
    SK = Xoá từ loại đầu tiên trong SK;
    While tu <> "" do
        Begin
            p = InStr(1, WCS, tu); //Xác định vị trí từ loại trong WCS
            If p = 1 Then
                NSK = NSK + 1;
                tu = Lấy từ loại đầu tiên trong SK;
                SK = Xoá từ loại đầu tiên trong SK;
                WCS = Xoá từ loại đầu tiên trong WCS
            Else
                Break;
            End;
        End;
End;

Function PhanTich:Integer; // chỉ số tổ hợp đúng, bằng -1 nếu tất cả sai
Begin
    Parse = -1
    SK = ""
    For các tổ hợp từ loại do
    Begin
        WCSet=chuỗi từ loại
        n = NSK(SK, WCSet)
        If n = 0 Then
            KhoiTao;
            i := 1;
            j := 0;
        Else
            SK = Left(SK, n * 3)
            WCSet = Mid(WCSet, n * 3 + 1)
            i:=1;
            j:=n+1;
```

manleviet@yahoo.com.hk

```
        End;
        Repeat
            Lấy giá trị đoán nhận trước.
            If giá trị đoán nhận trước là giá trị kết thúc chuỗi nhập then
                If KiemTraKetThucDung then
                    Ghi nhận chuỗi từ loại đúng
                    PhanTich:=j;
                    Exit;
                Else
                    Ghi nhận chuỗi từ loại là kết thúc sai
                    Break;
                End;
            Else
                Duyệt qua cột j
                    Lấy ký tự sau dấu chấm.
                    If ký tự sau dấu chấm là ký tự không kết thúc then
                        If DoanNhan then break;
                    Else { A→α•}
                        HoanThien;
                    End;
                Cho đến khi hết cột j;
                If hết cột mà không thể tiếp tục sang cột khác then
                    Ghi nhận chuỗi từ loại sai
                    Break;
                End;
            End;
        Until giá trị đoán nhận trước là giá trị kết thúc chuỗi nhập;
    End;
End;
```

manleviet@yahoo.com.hk

6. KẾT LUẬN

Qua quá trình nghiên cứu, chúng tôi đã đưa ra được giải thuật cải tiến. Giải thuật này đã giải quyết được hai hạn chế trong giải thuật Earley là luật dư thừa và bùng nổ tổ hợp. Những cải tiến này giúp hoàn thiện hơn giải thuật Earley để phân tích cú pháp trong xử lý ngôn ngữ tự nhiên.

TÀI LIỆU THAM KHẢO

[1]. Jay Earley, *An efficient context-free parsing algorithm*. *Commun*. *ACM* 13, 2 (Feb. 1970) 94-102.

[2]. Jay Earley, *An Efficient Context-Free Parsing Algorithm*. PhD Thesis, Carnegie-Mellon University. 1968.

[3]. J. Kilbury, *Earley-basierte Algorithmen für direktes Parsen mit ID/LP-Grammatiken*. KIT - Rep. 16, Institut für angewandte Informatik, TU Berlin, Berlin, June 1984.

[4]. Phan Thị Tươi, "Trợ giúp bắt lỗi chính tả tiếng Việt tự động bằng máy tính (giai đoạn 1)", đề tài cấp thành phố, Trường Đại học Bách Khoa TPHCM, 1998.

[5]. Phan Thị Tươi, "Trợ giúp bắt lỗi chính tả tiếng Việt tự động bằng máy tính (giai đoạn 2)", đề tài cấp thành phố, Trường Đại học Bách Khoa TPHCM, 2001.

[6]. Phan Thị Tươi, *Cải tiến một số giải thuật phân tích cú pháp trong xử lý ngôn ngữ tự nhiên*. Tạp chí Tin học và Điều khiển học, T.18, S.3 (2002), 279-284.

[7]. Phan Thị Tươi, *Trình biên dịch*. NXB GD, 1996.

manleviet@yahoo.com.hk