Tạp chí Tin học và Điều khiển học, T.22, S.4(2006), 325–338

CẢI TIẾN GIẢI THUẬT CYK CHO BÀI TOÁN PHÂN TÍCH CÚ PHÁP TIẾNG VIỆT

ĐINH THỊ PHƯƠNG THU, HOÀNG VĨNH SƠN, HUỲNH QUYẾT THẮNG

Khoa Công nghệ thông tin - Trường Đại học Bách Khoa Hà Nội

Abstract. The CYK (Cocke-Younger-Kasami) is one of algorithms used for parsing in the natural language processing. It is a simple algorithm for context free grammar that must be in Chomsky Normal Form. Because of Vietnamse's complexity, applying the CYK directly to the Vietnamese parsing is less effectiveness. In this paper, we propose some modifications of the CYK algorithm, data structure and data storage applying on the vietnamese parsing to improve the accuracy and increase performance. The experimental results are compared with the modificative Early algorithm ([6]) to show the efficiency of these modifications.

Tóm tắt. Giải thuật CYK (Cocke-Younger-Kasami) là một trong số những giải thuật được sử dụng để phân tích cú pháp trong xử lý ngôn ngữ tự nhiên. Đây là một giải thuật tổng quát, sử dụng tương đối đơn giản cho các văn phạm phi ngữ cảnh có các luật sinh tuân theo chuẩn Chomsky. Nhưng khi áp dụng giải thuật CYK cho tiếng Việt là một ngôn ngữ có tính phức tạp thì có nhiều hạn chế phải khắc phục. Trong bài báo này chúng tôi trình bày một đề xuất mở rộng giải thuật CYK và một số cải tiến quá trình phân tích, lưu trữ dữ liệu trung gian áp dụng trong bài toán phân tích cú pháp tiếng Việt để nâng cao tốc độ và độ chính xác. Kết quả thử nghiệm đã được so sánh với giải thuật phân tích cú pháp Early cải tiến ([6]) và đã chứng minh được tính hiệu quả.

1. MỞ ĐẦU

Trên thế giới, bài toán phân tích cú pháp đã được nghiên cứu từ lâu và phát triển mạnh mẽ với nhiều mô hình phân tích cú pháp khác nhau. Ứng dụng của phân tích cú pháp có ý nghĩa thực tế rất lớn đối với những hệ thống phát hiện và sửa lỗi chính tả, sửa lỗi cú pháp, tóm tắt, phân lớp văn bản,... Tuy nhiên tại Việt Nam những kết quả nghiên cứu về phân tích cú pháp tiếng Việt còn rất ít và hạn chế.

Để phân tích cú pháp trong xử lý ngôn ngữ tự nhiên có hai giải thuật phân tích cú pháp điển hình là giải thuật Cocke-Younger-Kasami (CYK) và giải thuật Earley. Cả hai giải thuật này đều dựa trên phương pháp phân tích bảng, thời gian tính là $O(n^3)$ ([1–6]).

Giải thuật CYK là một giải thuật dựa trên chiến lược chia để trị (divide-and-conquer), nghĩa là tìm cách chia chuỗi phân tích đầu vào thành các xâu con rồi phân tích từng xâu con đó. Thông thường trong xử lý ngôn ngữ tự nhiên, chúng ta bắt đầu từ mỗi xâu con là các âm tiết cách nhau bởi ký tự trắng. CYK là một giải thuật tương đối đơn giản và dễ cài đặt. Quá trình thực hiện từ trên xuống xuất phát từ ký hiệu khởi đầu, thử các luật sinh theo kiểu vét cạn, đồng thời xây dựng dần câu từ trái qua phải. Nhược điểm của giải thuật CYK là văn phạm không được đệ quy trái, nếu không quá trình phân tích sẽ rơi vào vòng lặp vô hạn

([1–5]).

Trong khi đó Early là một giải thuật có quá trình phân tích được thực hiện từ dưới lên, xuất phát từ xâu đầu vào, bằng cách thử áp dụng thu gọn các suy dẫn phải, tiến hành từ trái qua phải để đi đến ký hiệu khởi đầu. Giải thuật Earley có ưu điểm nổi bật là có thể sử dụng trong mọi ngôn ngữ phi ngữ cảnh mà không bị giới hạn bởi luật sinh dạng chuẩn Chomsky như CYK. Tuy nhiên Early lại có nhược điểm là phải duyệt qua quá nhiều luật sinh không cần thiết (luật dư thừa) trong giai đoạn đón nhận gây giảm tốc độ xử lý ([1, 3–6]).

Vì vậy cách tiếp cận bài toán phân tích cú pháp tiếng Việt trong nghiên cứu của chúng tôi dựa trên giải thuật CYK, nhưng có sự hiệu chỉnh và cải tiến để phù hợp với đặc điểm và luật cú pháp tiếng Việt nhằm giúp nâng cao độ chính xác và tốc độ xử lý.

Cấu trúc bàibáo: sau mục giới thiệu, Mục 2 trình bày về vấn đề lựa chọn văn phạm để mô hình hóa luật cú pháp tiếng Việt và xây dựng một tập luật cú pháp tiếng Việt trong văn phạm đó. Mục 3 sẽ trình bày về giải thuật CYK và đề xuất giải thuật mở rộng CYK để áp dụng cho phân tích cú pháp tiếng Việt. Các mở rộng nhằm giải quyết những vấn đề chính bao gồm tập luật cú pháp tiếng Việt không ở dạng chuẩn Chomsky và hiện tượng bùng nổ tổ hợp do đặc điểm tiếng Việt khi phân tích phải duyệt qua rất nhiều chuỗi từ loại khác nhau. Các kết quả thử nghiệm sẽ được trình bày trong Mục 4 của bài báo.

2. MÔ HÌNH HÓA VÀ XÂY DỰNG TẬP LUẬT CÚ PHÁP TIẾNG VIỆT

2.1. Mô hình hóa tiếng Việt

Mỗi ngôn ngữ trên thế giới đều có một hệ thống các quy tắc cú pháp riêng. Quy tắc cú pháp của ngôn ngữ tự nhiên rất đa dạng, phong phú và phức tạp. Tuy nhiên để giao tiếp giữa người và máy, hoặc giữa máy với máy ta cần một ngôn ngữ có các quy tắc cú pháp chặt chẽ chứ không thể áp dụng những đặc trưng của ngôn ngữ tự nhiên đơn thuần vào máy tính. Nói cách khác, ta cần một ngôn ngữ hình thức đủ mạnh để sử dụng biểu diễn thay cho ngôn ngữ tự nhiên và một văn phạm để biểu diễn ngôn ngữ đó.

Chomsky đã phân loại văn phạm thành bốn nhóm trên cơ sở định nghĩa văn phạm nói chung ([9]):

+ Nhóm Văn phạm ngữ cấu (Unrestricted Grammars): mọi quy tắc $r \in R$ đều có dạng $r = \alpha \to \beta$, trong đó $\alpha \in V^+, \beta \in V^*$.

+ Nhóm Văn phạm cảm ngữ cảnh CSG (Context-sensitive Grammars): mọi quy tắc $r \in R$ đều có dạng $r = \alpha \to \beta$, trong đó $\alpha \in V^+, \beta \in V^*$ và $|\alpha| \leqslant |\beta|$. Văn phạm này gọi là cảm ngữ cảnh vì α chỉ sinh ra β khi α phải nằm trong ngữ cảnh xác định nào đó.

+ Nhóm Văn phạm phi ngữ cảnh CFG (Context Free Grammars): mọi quy tắc $r \in R$ đều có dạng $r = A \to \theta, A \in N, \theta \in V^* = (T \cup N)^*$. Văn phạm này gọi là phi ngữ cảnh hay ngữ cảnh tự do vì biến A có thể sinh ra xâu θ mà không phụ thuộc vào bên cạnh nó có ngữ cảnh gì.

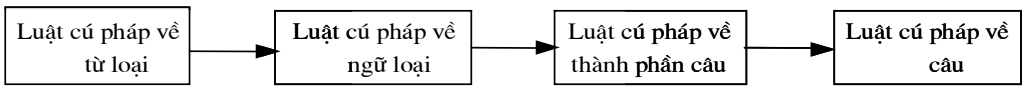
+ Nhóm Văn phạm chính quy (regular languge): gồm các quy tắc có dạng $A \to aB, A \to a, A \to Ba$ với $A, B \in N, a \in T$.

Về mặt cấu trúc cú pháp, các luật sinh của các văn phạm phi ngữ cảnh và văn phạm chính quy tương đối đơn giản và đã có nhiều ứng dụng trong thiết kế các ngôn ngữ lập trình, trong nghiên cứu chương trình dịch,... Trong ngôn ngữ tự nhiên nói chung, cụ thể là trong tiếng

Việt ([7, 8]), phần lớn các luật cú pháp đều ở dạng phi ngữ cảnh, nghĩa là vế trái của luật là một xâu ký hiệu không kết thúc, còn vế phải có thể là ký hiệu bất kỳ thuộc tập ký hiệu của nó, ví dụ: ⟨câu⟩ → ⟨chủ ngữ⟩ + ⟨vị ngữ⟩, ⟨chủ ngữ⟩ → ⟨danh ngữ⟩ hoặc ⟨vị ngữ⟩ → ⟨động từ⟩... Với các luật này thì văn phạm chính quy không đủ mạnh để biểu diễn ngôn ngữ tự nhiên do vế phải của văn phạm chính quy luôn có ký hiệu kết thúc. Bên cạnh đó, việc nhận biết một văn phạm cảm ngữ cảnh, văn phạm ngữ cấu phức tạp hơn rất nhiều so với văn phạm phi ngữ cảnh ([9]). Vì vậy, để mô hình hóa tập luật cú pháp tiếng Việt, chúng ta sẽ sử dụng văn phạm phi ngữ cảnh CFG để xây dựng.

2.2. Xây dựng tập luật cú pháp tiếng Việt cho văn phạm phi ngữ cảnh

Muốn mô hình hóa cú pháp tiếng Việt thì trước tiên ta phải xây dựng được tập luật cú pháp cho văn phạm biểu diễn ngôn ngữ tiếng Việt. Dựa trên quan điểm về ngữ pháp tiếng Việt trong [7, 8] chúng tôi đã thực hiện xây dựng các luật cú pháp văn phạm phi ngữ cảnh cho từ loại, ngữ loại, các thành phần câu và câu trong tiếng Việt (Hình 1).



Hình 1. Xây dựng tập luật cú pháp tiếng Việt cho văn phạm phi ngữ cảnh

2.2.1. Xây dựng các luật cú pháp về từ loại

Theo [7], từ loại tiếng Việt được chia thành các loại: danh từ, động từ, tính từ, phụ từ, kết từ, đại từ, trợ từ và cảm từ. Như vậy, chúng ta có thể coi mỗi từ loại này là một ký hiệu thuộc văn phạm.

Danh từ lại được chia thành các loại nhỏ hơn là danh từ đơn thể, danh từ tổng thể, danh từ chỉ sự vật trừu tượng, danh từ loại thể, danh từ số lượng, danh từ chỉ đơn vị, danh từ chỉ vị trí, danh từ riêng. Vì vậy, cũng lại có thể xem các loại danh từ trên cũng là ký hiệu của văn phạm. Từ đó ta sẽ xây dựng được các luật với danh từ như sau:

1. ⟨Danh từ⟩ → ⟨Danh từ đơn thể⟩	5. ⟨Danh từ⟩ → ⟨Danh từ đơn vị⟩
2. ⟨Danh từ⟩ → ⟨Danh từ tổng thể⟩	6. ⟨Danh từ⟩ → ⟨Danh từ chỉ vị trí⟩
3. ⟨Danh từ⟩ → ⟨Danh từ loại thể⟩	7. ⟨Danh từ⟩ → ⟨Danh từ riêng⟩
4. ⟨Danh từ⟩ → ⟨Danh từ số lượng⟩	

Tương tự như vậy, động từ cũng được chia thành động từ ngoại động, động từ nội động, động từ cảm nghĩ, động từ biến hóa, động từ phương hướng, động từ tồn tại, động từ ý chí, động từ tiếp thụ, động từ so sánh, động từ "là". Tính từ được chia thành hai loại: tính từ hàm chất và tính từ hàm lượng. Quá trình xây dựng tiếp tục như vậy với các từ loại còn lại.

2.2.2. Xây dựng các luật cú pháp về ngữ loại

Theo [7, 8] ta có các loại ngữ loại: danh ngữ, động ngữ và tính ngữ. Xét về cấu tạo, các ngữ đều có cấu tạo ba phần là phụ tố trước, chính tố và phụ tố sau. Tuy nhiên, mỗi ngữ tùy thuộc vào chính tố thuộc từ loại nào thì sẽ có các phụ tố tương ứng.

Ngữ có hai lớp cấu tạo: cấu tạo *ngữ* và cấu tạo *yếu tố của ngữ*. Cấu tạo yếu tố của ngữ nhiều khi rất phức tạp. Do đó, khi xây dựng các luật về ngữ loại cần phải tiến hành như sau: từ chính tố xác định các phụ tố có thể có. Tiến hành xây dựng luật về ngữ loại với chính tố

đó, sau đó xây dựng luật về các phụ tố - các yếu tố cấu tạo ngữ. Ví dụ cho ngữ danh từ, cấu tạo chung của ngữ danh từ như sau ([7]):

Phụ tố tổng thể/ Phụ tố số lượng/ Phụ tố loại thể đơn vị/ Chính tố/ Phụ tố hạn định/ Phụ tố chỉ định.

Trong 7 loại danh từ thì chỉ có 4 loại có khả năng làm chính tố của ngữ. Tập ký hiệu của văn phạm xây dựng sẽ có thêm 4 ký hiệu, đó là: ⟨danh ngữ danh từ trừu tượng⟩, ⟨danh ngữ danh từ vị trí⟩, ⟨danh ngữ danh từ đơn thể⟩, ⟨danh ngữ danh từ tổng thể⟩, và theo đó là 4 luật được xây dựng:

1. ⟨Danh ngữ⟩ → ⟨Danh ngữ danh từ trừu tượng⟩
2. ⟨Danh ngữ⟩ → ⟨Danh ngữ danh từ vị trí⟩
3. ⟨Danh ngữ⟩ → ⟨Danh ngữ danh từ đơn thể⟩
4. ⟨Danh ngữ⟩ → ⟨Danh ngữ danh từ tổng thể⟩

Với danh từ tổng thể, phụ tố trước chỉ có thể là phụ tố tổng thể, nên có thêm các luật:

5. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨phụ tố tổng thể⟩⟨danh từ tổng thể ⟩
6. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨phụ tố tổng thể⟩⟨danh từ tổng thể⟩⟨phụ tố hạn định⟩ ⟨phụ tố chỉ định⟩
7. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨phụ tố tổng thể⟩⟨danh từ tổng thể⟩⟨phụ tố hạn định⟩
8. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨phụ tố tổng thể⟩⟨danh từ tổng thể⟩⟨phụ tố chỉ định⟩
9. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨danh từ tổng thể⟩⟨phụ tố hạn định⟩⟨phụ tố chỉ định⟩
10. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨danh từ tổng thể⟩⟨phụ tố hạn định⟩
11. ⟨Danh ngữ danh từ tổng thể⟩ → ⟨danh từ tổng thể⟩⟨phụ tố chỉ định⟩

Vì phụ tố tổng thể do danh từ số lượng tổng thể đảm nhiệm, ta có thêm luật:

12. ⟨phụ tố tổng thể⟩ → ⟨danh từ số lượng tổng thể⟩

Vì phụ tố hạn định có thể do tính từ, động từ, danh từ hay một ngữ đảm nhiệm nên có thêm sáu luật:

13. ⟨phụ tố hạn định⟩ → ⟨Tính từ⟩
14. ⟨phụ tố hạn định⟩ → ⟨Động từ⟩
15. ⟨phụ tố hạn định⟩ → ⟨Danh từ⟩
16. ⟨phụ tố hạn định⟩ → ⟨Ngữ động từ⟩
17. ⟨phụ tố hạn định⟩ → ⟨Ngữ tính từ⟩
18. ⟨phụ tố hạn định⟩ → ⟨Ngữ danh từ⟩

Vì phụ tố chỉ định do các loại đại từ không gian - thời gian đảm nhiệm nên ta có thêm luật:

19. ⟨phụ tố chỉ định⟩ → ⟨đại từ không gian - thời gian⟩

Như vậy ta đã xây dựng xong tập luật cho ngữ danh từ tổng thể. Các ngữ danh từ khác, các ngữ động từ, tính từ cũng được xây dựng tương tự như vậy.

2.2.3. Xây dựng các luật cú pháp về thành phần câu

Các thành phần câu trong tập luật xây dựng bao gồm: chủ ngữ, vị ngữ, trạng ngữ, bổ ngữ, định ngữ. Chủ ngữ có thể là danh từ, danh ngữ, động từ, động ngữ, tính từ, tính ngữ, đại từ ([7]) do vậy ta xây dựng văn phạm có thêm một ký hiệu là ⟨Chủ ngữ⟩ và tập luật có thêm các luật sau:

1. ⟨Chủ ngữ⟩ → ⟨Tính từ⟩	5. ⟨Chủ ngữ⟩ → ⟨Ngữ tính từ⟩
2. ⟨Chủ ngữ⟩ → ⟨Động từ⟩	6. ⟨Chủ ngữ⟩ → ⟨Ngữ danh từ⟩
3. ⟨Chủ ngữ⟩ → ⟨Danh từ⟩	7. ⟨Chủ ngữ⟩ → ⟨Đại từ⟩
4. ⟨Chủ ngữ⟩ → ⟨Ngữ động từ⟩	

2.2.4. Xây dựng các luật cú pháp về câu

Câu bao gồm câu đơn và câu ghép. Câu đơn đơn giản có một kết cấu C-V, trong khi mẫu câu ghép có từ hai kết cấu C-V trở lên và chúng ta đã biết trước liên từ ([7]). Như vậy văn phạm xây dựng có thêm các ký hiệu ⟨Câu⟩, ⟨Câu ghép⟩ và các luật cú pháp như:

1. ⟨Câu⟩ → ⟨Chủ ngữ⟩⟨Vị ngữ⟩
2. ⟨Câu ghép⟩ → ⟨Nếu⟩⟨Câu⟩⟨Thì⟩⟨Câu⟩
3. ⟨Câu ghép⟩ → ⟨Tuy⟩⟨Câu⟩⟨Nhưng⟩⟨Câu⟩
4. ⟨Câu ghép⟩ → ⟨Mặc dù⟩⟨Câu⟩⟨Nhưng⟩⟨Câu⟩

3. CẢI TIẾN GIẢI THUẬT CYK ÁP DỤNG CHO PHÂN TÍCH CÚ PHÁP TIẾNG VIỆT

3.1. Giải thuật CYK

Khi áp dụng giải thuật CYK cho phân tích cú pháp ta phải thực hiện hai bước: bước tạo bảng phân tích và bước phân tích trái từ bảng phân tích đã tạo ra ([1–5, 9]).

Giải thuật CYK

Đầu vào:

+ Văn phạm $G = (N, T, S, P)$ dạng chuẩn Chomsky, không chứa sản xuất rỗng.

+ Xâu vào $w = a_1a_2...a_n \in T^*$.

Đầu ra: Bảng phân tích T đối với w sao cho t_{ij} chứa A khi và chỉ khi $A \Rightarrow^+ a_1a_2...a_{i+j-1}$

Phương pháp:

1. Tập hợp $t_{i1} = \{A | A \to a_i \in P\}$ với mỗi i. Sau bước này nếu t_{i1} chứa A thì ta có $A \Rightarrow^+ a_i$.

2. Giả sử t_{ij} đã tính $\forall i(1 \leqslant i \leqslant n)$ và $\forall j(1 \leqslant j < j)$.

Tập hợp $t_{ij} = \{A |$ với k nào đó $(1 \leqslant k < j), A \to BC \in P, B \in t_{ik}$ và $C \in t_{i+k,j-k}\}$

Từ điều kiện $1 \leqslant k < j$ suy ra k và $j - k$ nhỏ hơn j. Do đó t_{ik} và $t_{i+k,j-k}$ đã tính trước khi t_{ij} phải tính. Sau bước này, nếu t_{ij} chứa A thì

$$A \Rightarrow BC \Rightarrow^+ a_1a_2...a_{i+k-1}C \Rightarrow^+ a_1a_2...a_{i+k-1}a_{i+k}...a_{i+j-1}.$$

3. Lặp Buớc 2 cho tới khi t_{ij} đã tính với $1 \leqslant i \leqslant n, 1 \leqslant j < n - i + 1$.

Giải thuật phân tích trái CYK

Đầu vào:

Văn phạm $G = (N, T, S, P)$ dạng chuẩn Chomsky, các luật sản xuất (luật sinh) đánh số từ $1, ..., q$. Xâu vào $w = a_1a_2...a_n$.

Bảng phân tích T đối với w.

Đầu ra: Phân tích trái đối với w hoặc thông báo "sai"

Phương pháp:

Ta sử dụng thủ tục đệ quy $SINH(i, j, A)$ để sinh ra phân tích trái đối với suy dẫn:

$$A \Rightarrow^{+} a_1 a_2 ... a_{i+j-1}$$

Thủ tục $SINH(i, j, A)$ xác định như sau:

1. Nếu $j = 1$ và sản xuất thứ m trong P là $A \Rightarrow a_i$, ta viết số thứ tự sản xuất là m.

2. Nếu $j > 1$, gọi k là số nguyên nhỏ nhất $1 \leqslant k < j$ sao cho với $B \in t_{ik}$ và $C \in t_{i+1,k}$ ta có: $A \to BC \in P$ có số thứ tự m.

(Trong trường hợp có nhiều sản xuất dạng $A \to BC$, ta chọn tùy ý một sản xuất với số thứ tự m nhỏ nhất), viết số thứ tự m đó và thực hiện

$$SINH(i, k, b) \text{ và } SINH(i + k, j - k, C).$$

Giải thuật bắt đầu từ $SINH(1, n, S)$ với điều kiện $S \in t_{1n}$, nếu $S \notin t_{1n}$ thì đưa ra thông báo "sai".

Ví dụ ([2])

Cho văn phạm phi ngữ cảnh G với các luật sinh sau:

1. $S \to AA$
2. $S \to AS$
3. $S \to b$
4. $A \to SA$
5. $A \to AS$
6. $A \to a$

Chuỗi nhập vào là $w = abaab$.

Với giải thuật CYK ta có bảng phân tích ở Bảng 1.

5	A,S				
4	A,S	A,S			
3	A,S	S	A,S		
2	A,S	A	S	A,S	
1	A	S	A	A	S
i / j	1	2	3	4	5

1: S -> AA
6: A -> a
4: A -> SA
3: S -> b
5: A -> AS
6: A -> a
2: S -> AS
6: A -> a
3: S -> b

Bảng 1

⇒Đầu ra: 164356263;

Nhận xét

Ta thấy giải thuật CYK tương đối đơn giản và dễ áp dụng, nhưng nếu chúng ta áp dụng trực tiếp CYK cho bài toán phân tích cú pháp tiếng Việt thì CYK sẽ gặp một số nhược điểm sau:

+ CYK chỉ làm việc tốt với các luật dạng chuẩn Chomsky ($A \to BC$ và $A \to \alpha$, với α là ký hiệu kết thúc). Trong khi đó, có nhiều luật cú pháp trong tiếng Việt không thuộc dạng này, ví dụ

⟨danh ngữ⟩ → ⟨phụ tố danh ngữ trước⟩⟨chính tố danh ngữ⟩⟨phụ tố danh ngữ sau⟩.

+ Khi thực hiện giải thuật phân tích từ trái của bảng phân tích CYK, chúng ta phải tìm lại các luật một lần nữa để dựng cây phân tích. Điều này sẽ làm mất nhiều thời gian thực hiện.

+ Việc chọn tùy ý một sản xuất với số thứ tự m nhỏ nhất sẽ làm mất câu phân tích mà có thể câu phân tích này là trường hợp phân tích đúng với ngữ cảnh văn bản.

+ Tiếng Việt là ngôn ngữ đơn âm tiết nên vấn đề tách từ tiếng Việt sẽ dẫn đến hiện tượng bùng nổ hình thái câu tương ứng với những cách tách từ và gán nhãn từ loại khác nhau.

Như vậy rõ ràng là để có thể áp dụng CYK cho phân tích cú pháp tiếng Việt chúng ta sẽ cần phải tiến hành một số cải tiến nhằm khắc phục các hạn chế trên.

3.2. Cải tiến CYK áp dụng cho bài toán phân tích cú pháp tiếng Việt

3.2.1. Đề xuất mở rộng giải thuật CYK

Nhận thấy tập luật cú pháp tiếng Việt đã xây dựng có nhiều luật không thuộc dạng chuẩn Chomsky, nghĩa là bao gồm các luật mà vế phải không chỉ có dạng bộ đôi $A \to BC$ mà còn có thể có các dạng bộ ba $A \to BCD$, bộ bốn $A \to BCDE$, bộ năm $A \to BCDEF$,...

Để áp dụng giải thuật CYK, ta có thể sử dụng biện pháp là chuyển các luật này về dạng chuẩn Chomsky bằng cách thêm các ký hiệu trung gian ([9]). Ví dụ với luật: $A \to BCDE$, ta cần thêm hai ký hiệu trung gian F và G sao cho: $A \to FG, F \to BC, G \to DE$. Nghĩa là ta đã loại bỏ khỏi tập luật một luật cũ và thêm vào ba luật mới sinh ra. Tuy nhiên cách làm này có các hạn chế sau:

+ Các ký hiệu trung gian này thường là không có ý nghĩa về cú pháp.

+ Việc sinh ra các ký hiệu này và bổ sung các luật mới một cách tùy tiện sẽ làm tăng kích thước và gây nhiễu.

Vì vậy chúng ta không nên sử dụng phương pháp này vì nó không hiệu quả.

Để giải quyết vấn đề này chúng tôi đã tiến hành đề xuất mở rộng giải thuật CYK như sau.

Mở rộng giải thuật CYK

Bước 1: Áp dụng cho các luật dạng $A \to a$

Thực hiện tìm tất các luật dạng $A \to a$ để tìm tập hợp tất cả các ký hiệu $t_{i1} = \{A | A \to a_i \in P\}$ với mỗi i.

Bước 2: Áp dụng cho các luật dạng $A \to B$ (các luật không ở dạng chuẩn Chomsky)

Thực hiện Bước 2 để tìm tất cả các ký hiệu mới từ các luật dạng $A \to B$ và ghi lại. Chính nhờ bước hai này ta sẽ không bỏ sót các luật dạng $A \to B$.

Bước 3: Áp dụng cho các luật dạng $A \to BC[DEF]$ (các luật không ở dạng chuẩn Chomsky)

Ta sẽ không thay đổi gì các luật dạng này, mà tại Bước 3 của giải thuật ta sẽ tìm các luật mà phần đầu của vế phải thỏa mãn luật ở dạng chuẩn Chomsky $A \to BC[DEF]$. Phần dư ra, ký hiệu là wait (tương ứng với [DEF]) sẽ được ghi lại. Chúng ta sẽ xây dựng các nút dữ liệu thay vì sử dụng trực tiếp danh sách từ. Tiến hành đồng thời việc phân tích trên các nút, sử dụng biến đi kèm các nút ghi lại giá trị [DEF].

Tại lần bổ sung bảng phân tích tiếp theo, nếu phần dư ra đó khác rỗng, ta sẽ không đi tìm luật có vế phải dạng AX (X là giá trị đã được tính ở bước trước trong bảng) mà sẽ so sánh ký tự đầu của *wait* với X :

+ Nếu bằng nhau, giá trị A được ghi vào vị trí đang xác định, $wait = wait\backslash$(ký tự đầu của $wait$).

+ Nếu khác nhau, nghĩa là sự kết hợp này không cho kết quả, chương trình sẽ thử một sự kết hợp khác theo giải thuật.

Nếu phần tử tính cuối cùng của bảng có chứa ký hiệu khởi đầu S và $wait(S) = 0$ thì quá trình đoán nhận xâu vào là đúng, ngược lại là sai.

Cụ thể giải thuật CYK mở rộng sẽ được cài đặt như sau.

Đầu vào: Danh sách từ xuất hiện trong câu.

Đầu ra: Không có, có một hay có nhiều trường hợp phân tích cú pháp của câu đó.

1. Xác định tất cả các nhãn từ loại có thể có của mỗi từ trong câu. Tất cả những nhãn này cùng vị trí trong câu được đưa vào danh sách phân tích hiện tại.

2. Trên mỗi nhãn L thuộc danh sách hiện tại, tiến hành tìm tất cả các luật suy dẫn dạng $A \to L$. Nếu A đã có trong danh sách hiện tại thì bỏ qua, nếu chưa có thêm A vào danh sách hiện tại và danh sách mới. Bước 2 được lặp đi lặp lại cho tới khi không có thêm nhãn nào được tìm thấy.

3. Trên danh sách mới và danh sách hiện tại, tiến hành áp dụng giải thuật CYK mở rộng. Cụ thể như sau:

+ Với mỗi nhãn N thuộc danh sách mới, ta tìm tất cả các nhãn thuộc danh sách hiện tại có vị trí đứng ngay sau N trong câu. Với mỗi nhãn N_i này, ta tìm xem có luật nào dạng $X \to NN_i[YZW]$ hay không. Nếu tìm thấy ta tạo một nhãn mới là X. Chú ý là nếu luật suy dẫn tìm được ở trên có vế trái nhiều hơn hai phần tử (ví dụ có thêm YZW) thì YZW phải được ghi lại vào ghi chú của X.

+ Với mỗi nhãn N thuộc danh sách mới và có ghi chú khác rỗng (ví dụ là OP), nếu ta tìm được một luật suy dẫn dạng NO, ta sẽ tạo mới một nhãn N thứ hai, giống hệt nhãn N thứ nhất, tuy nhiên ghi chú = ghi chú $\backslash\{O\} = P$.

4. Sau tất cả các bước trên, kiểm tra xem có nhãn nào mới được tạo ra không. Nếu có, gán danh sách mới = {tất cả các nhãn mới}, danh sách hiện tại = {danh sách hiện tại danh sách mới}, quay lại Bước 2. Nếu không có nhãn nào mới, đến Bước 5.

5. Đưa ra tất cả các nhãn nào có vị trí bắt đầu là đầu câu và kết thúc là cuối câu cùng với quá trình xây dựng lên các nhãn đó.

Ví dụ. Phân tích câu: "Nhân dân đấu tranh chống chiến tranh".

Các luật sinh áp dụng:

1. ⟨Danh từ⟩ → ⟨Danh từ trừu tượng⟩

2. ⟨Phụ tố đối tượng - Động từ ngoại động⟩ → ⟨Danh từ⟩

3. ⟨Phụ tố sau động ngữ động từ ngoại động⟩ → ⟨Phụ tố đối tượng - Động từ ngoại động⟩

4. ⟨Động từ⟩ → ⟨Động từ ngoại động⟩

5. ⟨Phụ tố mục đích⟩ → ⟨Động từ⟩

6. ⟨Phụ tố sau động ngữ động từ ngoại động⟩ → ⟨Phụ tố mục đích⟩ → ⟨Phụ tố sau động ngữ động từ ngoại động⟩

7. ⟨Động ngữ động từ ngoại động⟩ → ⟨Động từ ngoại động⟩ → ⟨Phụ tố sau động ngữ động từ ngoại động⟩

8. ⟨Động ngữ⟩ → ⟨Động ngữ động từ ngoại động⟩

9. ⟨Vị ngữ⟩ → ⟨Động ngữ⟩

10. ⟨Câu⟩ → ⟨Chủ ngữ⟩ → ⟨Vị ngữ⟩

Chúng ta có bảng phân tích của giải thuật CYK cải tiến (Bảng 2).

Như vậy, chúng ta đã thấy việc cải tiến trên giúp giảm thời gian phân tích, tính toán và xử lý được các luật phi ngữ cảnh không thuộc dạng chuẩn Chomsky trong văn phạm tiếng Việt.

Bảng 2. Ví dụ về bảng phân tích của giải thuật CYK cải tiến

	Nhân dân 1	***đấu tranh*** 2	***chống*** 3	***chiến tranh*** 4
	Ng	Vt	Vt	Na
1	+ Chủ ngữ (1.1) =>Ng + Noun(1.1) => Ng	+ DongTu (2.1) =>Vt + PhuToMucDich(2.1) =>DongTu	+ DongTu (2.1) =>Vt + PhuToMucDich (3.1) =>DongTu	+ Noun (4.1) => Na +PhuToDoiTuong_DongTuNgoaiDong (4.1)=> Noun. +PhuToSauDongNguDongTuNgoaiDong (4.1) => PhuToDoiTuong_DongTuNgoaiDong
2	0	0	+PhuToSauDongNguDongTuNgoaiDong (3.2) => PhuToMucDich (3.1)+PhuToSauDongNguDongTuNgoaiDong (4.1)	
3	0	+ DongNguDongTuNgoaiDong (2.3) => Vt (3.0) + PhuToSauDongNguDongTuNgoaiDong (3.2) + DongNgu (2.3) => DongNguDongTuNgoaiDong (2.3) + Vị ngữ (2.3) => DongNgu (2.3)		
4	Câu (1.4) => Chủ ngữ (1.1) + Vị ngữ (2.3)			

3.2.2. Một số kĩ thuật và cấu trúc dữ liệu xây dựng

a. Giải quyết bùng nổ hình thái câu

Hiện tượng bùng nổ tổ hợp xảy ra do một từ có thể thuộc vào nhiều từ loại khác nhau, đồng thời quá trình tách từ có thể sinh ra nhiều cách tách từ khác nhau.

Chúng tôi tiến hành giải quyết vấn đề bùng nổ tổ hợp như sau: thay vì việc liệt kê tất cả các hình thái câu rồi sau đó tiến hành tìm cây phân tích trên từng hình thái đó, ta sẽ liệt kê tất cả các từ có trong câu, vị trí xuất hiện đầu và cuối của chúng trong câu, cùng mảng các nghĩa của mỗi từ đó. Sau đó, trên tập hợp từ này, ta tiến hành thực hiện giải thuật như bình thường. Cách làm này đem lại lợi ích rất lớn vì tất cả các tổ hợp từ bị trùng lặp trong các hình thái câu gần giống nhau sẽ được tạo ra chỉ một lần. Lợi ích về bộ nhớ và về thời gian thực hiện đều được nâng cao.

Cụ thể, lấy ví dụ: "Tôi đi học". Câu trên sẽ được tách thành 3 từ: *Tôi, đi, học* và đặt là ba GM (gramaword). Với mỗi GM ta có ba trường: vị trí đầu của từ đó trong xâu, vị trí cuối của từ đó trong xâu và mảng các nghĩa. Trong ví dụ trên: *Tôi* (0, 1, mảng nghĩa "*Tôi*"), *đi* (1, 2, mảng nghĩa của "*đi*"), *học* (2, 3, mảng nghĩa của "*học*"). Xét từ "tôi". Giả sử "*tôi*" có 2 nhãn từ loại là đại từ "*tôi*" và động từ "*tôi*" (tôi vôi), khi đó mảng nghĩa của từ "*tôi*" sẽ có 2 phần tử, mỗi phần tử này sẽ là một lớp gồm 4 trường sau:

+ Trường *mã vế trái* lưu lại vế trái của luật cú pháp: (ví dụ "*chủ ngữ* $\rightarrow$ *đại từ*", thì trường này sẽ là "*chủ ngữ*"). Việc này nhằm mục đích nâng cao tốc độ tìm kiếm (làm khóa

tìm kiếm).

+ Trường *vết*: trường này lưu lại toàn bộ quá trình đã tạo ra nghĩa đang xét. Điều này làm cho công việc dựng và rà soát lại cây tránh việc trùng lặp và nhập nhằng. Ở đây, các *vết* của nghĩa con sẽ bao gồm luôn cả *vết* của nghĩa cha.

+ Trường *wait*: đây là trường phục vụ cho các luật mà vế phải không phải là bộ đôi, ví dụ: với luật $A \rightarrow BC[DEF]$ thì trường này sẽ lưu lại [DEF].

+ Trường *dẫn xuất* xác định dẫn xuất: trường này dùng để đánh dấu xem một nghĩa nào đó đã tạo dẫn xuất hay chưa nhằm tránh được việc lặp lại trong mỗi vòng lặp.

Ngoài ra trong quá trình phân tích cú pháp của câu, chúng tôi nhận thấy việc bùng nổ hình thái của câu cũng liên quan rất nhiều đến việc tổ chức lưu trữ cũng như tìm kiếm từ trong từ điển, nhãn từ. Do đó, trong quá trình phân tích, chúng tôi luôn chú trọng việc đặt các mốc đánh dấu cho việc xét duyệt các luật suy dẫn.Ví dụ như với hai nhãn *GM1* và nhãn *GM2* kết hợp với nhau, thì trong *GM1* phải lưu lại số nghĩa mà nó đã kết hợp với *GM2* cũng như vị trí để bắt đầu duyệt nghĩa (để giảm thời gian tìm kiếm). Đồng thời trong mỗi nhãn cũng phải lưu lại những mã mà chính từ mã ấy đã tạo ra các dẫn xuất dạng $A \rightarrow B$ (ở đây ta đã lưu lại mã B, ghi nhận là nó đã tạo ra dẫn xuất là A), việc này sẽ tránh được sự bùng nổ tổ hợp thừa sau này.

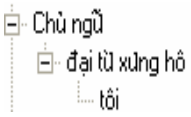
b. Lưu vết dựng cây phân tích

Trong giải thuật CYK, khi dựng lại cây phân tích, hệ thống phải tìm lại các suy dẫn, trong khi quá trình này đã được thực hiện trước đó. Để nâng cao hiệu quả cho giải thuật, chúng tôi tiến hành sử dụng một *vết* đi kèm theo mỗi phần tử của bảng để lưu *vết* của tổ hợp tạo nên nó. Từ tổ hợp này, ta có thể suy đến các ký hiệu kết thúc của xâu đó, việc này giúp tiện lợi cho quá trình tạo cây được nhanh chóng mà không bị nhập nhằng về sau.

Cụ thể, như đã mô tả ở trên, chúng tôi tổ chức các nhãn từ, mỗi một nhãn từ sẽ có các trường ⟨*mã vế trái, vết, wait, dẫn xuất*⟩. Ở đây, mã vế trái chính là mã của thành phần câu đã sinh ra các phần ở trong cây phân tích *treenode*. Ví dụ trong trường hợp nhãn cho từ "tôi" là "*đại từ xưng hô*":

+ *mã vế trái* là "*Chủ ngữ*"

+ phần *vết* là một nút có cấu trúc "*đại từ xưng hô* → *tôi*".



c. Lưu trữ đầy đủ các trường hợp phân tích cú pháp khác nhau

Trong trường hợp một câu có nhiều cách phân tích cú pháp khác nhau, chúng tôi tổ chức một danh sách để quản lý các cách phân tích được nhằm đảm bảo không bỏ sót trường hợp. Điều này có nghĩa là ta sẽ xem xét mọi sản xuất dẫn đến kết quả, không chỉ xem xét sản xuất bất kỳ có số thứ tự nhỏ nhất như trong giải thuật CYK ban đầu.

4. KẾT QUẢ THỬ NGHIỆM

Chúng tôi đã lựa chọn môi trường .NET là môi trường phát triển ứng dụng tiên tiến hiện nay và ngôn ngữ lập trình C# để tiến hành cài đặt mô hình kiểm thử. Ngôn ngữ C# có khả năng xử lý các cơ sở dữ liệu phẳng của bài toán một cách nhanh chóng, đồng thời cung cấp một hệ thống thư viện các hàm, thủ tục cơ sở rất mạnh, thuận tiện, có cơ chế quản lý tìm kiếm và truy xuất phần tử tối ưu, linh hoạt như *DictionaryBase (cấu trúc dữ liệu từ điển),*

HashTable (cấu trúc dữ liệu bảng băm), ArrayList (cấu trúc danh sách mảng) thích hợp cho các bài toán Text Mining.

Kết quả thực nghiệm được tiến hành trên máy Pentium IV, 3GHz, 512 MRAM.

4.1. Tập luật cú pháp tiếng Việt xây dựng

Dựa trên quan điểm về ngữ pháp tiếng Việt trong [7, 8] hiện nay chúng tôi đã xây dựng được tập luật bao gồm 438 luật cú pháp cho tiếng Việt cho mô hình văn phạm phi ngữ cảnh.

Các luật đã xây dựng hiện nay bao gồm: các luật cú pháp về từ loại, các luật cú pháp về ngữ loại, một số luật về thành phần câu, một số luật về câu (câu đơn và câu ghép). Các luật cú pháp này được mã hóa và lưu trữ dựa trên lớp *clsDictionaryBase* trong .NET với các phần tử có *key* sẽ là vế phải của luật, còn *value* là một danh sách bao gồm tất cả các vế trái có thể có của luật này. Việc xây dựng các lớp lưu trữ dữ liệu dạng từ điển dựa trên *clsDictionaryBase* giúp tối ưu hóa về tốc độ vì khối lượng từ phân tách là rất lớn, đồng thời giúp dễ dàng thay đổi và hiệu chỉnh.

4.2. Tập dữ liệu thử nghiệm

Chúng tôi tiến hành quá trình kiểm thử giải thuật phân tích cú pháp trên một tập gồm 300 câu tiếng Việt được lấy từ [7, 8], thể hiện các mẫu câu ở các dạng ngữ pháp thông dụng của tiếng Việt. Tập 300 câu này đã được xác định cấu trúc cú pháp sẵn trong [7, 8] để dễ dàng khẳng định tính hiệu quả và chính xác của kết quả phân tích cú pháp.

Ví dụ:

Câu phân tích	Cấu trúc cú pháp
Thầy giáo yêu cầu im lặng	$C(N1) + V1 + V2$
Tôi nhớ đôi mắt ấy	$C(N1) + V1 + N2$

Tập câu phân tích được ghi lại các thông số để tiện cho việc đánh giá kết quả. Một số thông số của tập dữ liệu thử nghiệm này được mô tả trong Bảng 3.

Bảng 3. Một số thông số của tập dữ liệu thử nghiệm

Tập dữ liệu / Thông số	Câu tiếng Việt
Số lượng câu	300 câu
Kích thước câu dài nhất	13 âm tiết
Kích thước câu ngắn nhất	3 âm tiết
Kích thước trung bình một câu	7 âm tiết

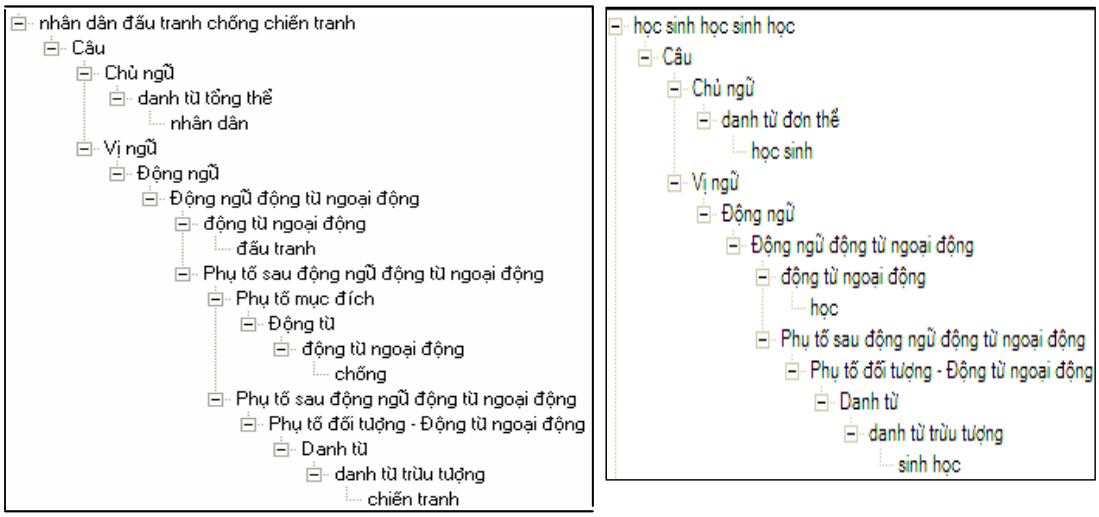
4.3. Kết quả phân tích cú pháp với giải thuật cải tiến CYK

Để tạo điều kiện cho việc đánh giá tính hiệu quả của giải thuật cải tiến CYK đề xuất trên, chúng tôi đã tiến hành cài đặt hai giải thuật phân tích cú pháp: giải thuật cải tiến CYK đề xuất ở trên và giải thuật Early cải tiến trong [6].

Giải thuật Early cải tiến ([6]) giải quyết việc phải duyệt qua các luật sinh dư thừa của Early ban đầu bằng cách lập luật sinh với dạng riêng. Tất cả các luật sinh của văn phạm phi ngữ cảnh $G(V, W, S, P)$ đều thuộc vào một trong hai dạng: $A \to \alpha, \alpha \in V^*$ và $A \to a, a \in W$. Khi đó các luật dư thừa sẽ là các luật có vế phải chỉ là một ký tự kết thúc ($A \to a$) hoặc luật có vế phải là các ký tự không kết thúc mà không dẫn đến đệ quy trái ($A \to \alpha$). Vì thế giải thuật Early cải tiến ([6]) chỉ còn hai bước, bỏ qua bước quét do dạng luật sinh đã giúp phát

hiện luật dư thừa trong giai đoạn đón nhận. Vấn đề bùng nổ tổ hợp cũng được giải quyết như sau: đầu tiên 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 phân tích. Khi kiểm tra một chuỗi thất bại sẽ 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.

Quá trình thực nghiệm giải thuật CYK cải tiến ở trên và Early cải tiến ([6]) tiến hành trên hai máy có cùng cấu hình, với cùng tập 438 luật cú pháp tiếng Việt đã xây dựng và thử nghiệm trên cùng tập dữ liệu 300 câu tiếng Việt. Một số ngữ liệu khác được sử dụng bao gồm: từ điển từ vựng tiếng Việt hơn 70.000 từ, từ điển từ loại tiếng Việt 37.000 từ ([10]).



Hình 3. Hai ví dụ về kết quả phân tích cú pháp bằng giải thuật CYK cải tiến

Với đầu vào là một câu tiếng Việt cần phân tích cú pháp, kết quả kiểm thử đầu ra của hai giải thuật là tập các cách phân tích được coi là hợp lệ của câu đó. Chúng tôi giới hạn kết quả thử nghiệm không đưa ra phương án phân tích được hệ thống lựa chọn là tốt nhất, nguyên nhân là do việc giải quyết nhập nhằng giữa các cách phân tích này là một bài toán lớn khác và phụ thuộc vào nhiều yếu tố quan trọng mà hiện nay vẫn chưa được giải quyết hiệu quả.

Kết quả kiểm thử thực nghiệm của hai giải thuật CYK và Earley như sau (Bảng 4).

Bảng 4. So sánh kết quả phân tích cú pháp giữa CYK cải tiến và Earley cải tiến

Tập dữ liệu 300 câu	CYK cải tiến	Early cải tiến
Số trường hợp phân tích đưa ra ở kết quả (trung bình)	3	8
Số câu đưa ra được trường hợp phân tích đúng	227	216
Số câu không đưa ra được trường hợp phân tích đúng	73	84
Hiệu suất	75%	72,6%
Tốc độ trung bình	0.406 s	0.521 s

Nhận xét

+ Kết quả phân tích cú pháp bằng giải thuật Early cải tiến ([6]): đưa ra số trường hợp phân tích câu nhiều hơn, độ chính xác thấp hơn, thời gian phân tích lâu hơn.

+ Kết quả phân tích cú pháp bằng giải thuật CYK cải tiến: có độ chính xác cao hơn không nhiều tuy nhiên thời gian phân tích nhanh hơn. Đặc biệt giải thuật đã giải quyết được khá hiệu quả sự bùng nổ tổ hợp trong quá trình phân tích, nên số lượng trường hợp phân tích câu đưa ra cuối cùng so với Early ít hơn rất nhiều (trung bình chỉ khoảng 30% so với Early).

+ Các trường hợp hai giải thuật phân tích sai phần nhiều do tập luật chưa xử lý được những trường hợp câu phức tạp. Một phần nguyên nhân là do tập luật cú pháp tiếng Việt xây dựng chưa đầy đủ, luật cú pháp chưa đủ bao quát hầu hết các quy tắc trong ngôn ngữ tiếng Việt, một phần khác là do thiếu ngữ liệu tiếng Việt cho quá trình xử lý.

5. KẾT LUẬN

Với những nghiên cứu trình bày ở trên, chúng tôi đã xây dựng một tập luật cú pháp tiếng Việt trong mô hình văn phạm phi ngữ cảnh. Tập luật xây dựng được hiện tại tuy chưa đầy đủ nhưng đã đi sâu vào từng từ loại, ngữ loại giúp việc phân tích cú pháp được chi tiết và rõ ràng. Bên cạnh đó, chúng tôi cũng đã đề xuất mở rộng giải thuật CYK và một số cải tiến quá trình phân tích, lưu trữ dữ liệu trung gian trong quá trình đoán nhận câu. Những cải tiến này giúp giải quyết hai hạn chế trong CYK là luật cú pháp không ở dạng văn phạm chuẩn Chomsky và bùng nổ tổ hợp, giúp nâng cao hiệu quả giải thuật CYK hơn để áp dụng trong bài toán phân tích cú pháp tiếng Việt.

Tiếp theo, chúng tôi sẽ tiếp tục hoàn thiện quá trình mô hình hóa tiếng Việt với tập luật cú pháp đầy đủ và chính xác hơn. Bên cạnh đó, vấn đề giải quyết nhập nhằng trong phân tích cú pháp cũng sẽ được nghiên cứu kết hợp với mô hình phân tích cú pháp dựa trên xác suất PCFG (Problistic Context Free Grammar) nhằm đưa ra một phương pháp giải quyết hiệu quả.

TÀI LIỆU THAM KHẢO

[1] Dick Grune, *Ceriel Jacobs.Parsing Techniques: A Practial Guide*, Ellis Horwood, Chichester, England, 1990.

[2] Chung, Sei Kwang, "The Cocke-Younger-Kasami Algorithm", http://klpl.re.pusan.ac.kr/seminar/.

[3] Bernd Kiefer, "Chart Parsing of Context-Free Grammars", http://www.coli.uni-saarland.de.

[4] Richard Sproat, LING 406: Introduction to Computational, http://catarina.ai.uiuc.edu/L406/Lectures.

[5] Martin Rajman, Natural language processing computationnal linguistics, http://www.di.uevora.pt.

[6] Nguyễn Gia Định, Trần Thanh Lương, Lê Viết Mẫn, Một số cải tiến giải thuật Early cho việc phân tích cú pháp trong xử lý ngôn ngữ tự nhiên, *Tạp chí Khoa học Đại học Huế* (25) (2004) 45–51.

[7] Nguyễn Chí Hòa, *Ngữ pháp tiếng Việt thực hành*, NXB Đại Học Quốc Gia, 2004.

[8] Nguyễn Tài Cẩn, "Ngữ pháp tiếng Việt: Tiếng - Từ ghép - Đoản ngữ". Tài liệu dùng cho sinh viên, nghiên cứu sinh và bồi dưỡng giáo viên ngữ văn, in lần thứ 5, Nhà xuất bản ĐHQGHN, 1998.

[9] Lê Thanh Hương, Phân tích cú pháp tiếng Việt, "Luận văn tốt nghiệp thạc sĩ" ĐHBK Hà Nội, 2000.

[10] Nguyễn Thị Minh Huyền, Vũ Xuân Lương, Lê Hồng Phương, Sử dụng bộ gán nhãn từ loại xác suất QTAG cho văn bản tiếng Việt, *Hội thảo khoa học quốc gia lần thứ nhất về Nghiên cứu phát triển và ứng dụng Công nghệ thông tin và truyền thông ICT.RDA'03*, 22-23/10/2003.

[11] http://www.vietlex.com/research/ITCra03POSTagging.html.

Nhận bài ngày 4 - 12 - 2005
Nhận lại sau sửa ngày 1 - 11 -2006