MỘT CÁCH TIẾP CẬN TRONG PHÂN TÍCH VĂN BẢN TIẾNG VIỆT

ThS. Lê Thanh Hương, Khoa CNTT, Trường ĐHBK Hà nội
PTS. Nguyễn Thanh Thuỷ, Khoa CNTT, Trường ĐHBK Hà nội
PTS. Phạm Hồng Quang, Viện Toán học

Abstract:

This paper represents an approach to construct a Vietnamese syntactic parser using an extension of the context-free grammar. Morphological and syntactic analysis are carried out in parallel in order to reduce possible word combinations and meaning ambiguity. From this point of view, we construct our parsing algorithm and describe some problems in implementing textual processing.

1. Mở đầu

1.1 Vấn đề phân tích văn bản

Việc phân tích văn bản từ trước đến nay thường được chia thành bốn mức:

1. *Phân tích từ vựng:* Quá trình phân tích từ vựng nhằm phân tích hình thái của các từ tạo nên câu, từ đó kiểm tra được tính đúng đắn của âm tiết và từ.
2. *Phân tích cú pháp:* Quá trình phân tích cú pháp đưa ra mô tả về quan hệ và vai trò ngữ pháp của các từ, nhóm từ (hay ngữ) trong câu; đồng thời đưa ra cấu trúc của câu.
3. *Phân tích ngữ nghĩa:* Mục đích của việc phân tích ngữ nghĩa là kiểm tra ý nghĩa của câu có mâu thuẫn với ý nghĩa cả đoạn hay không. Dựa trên mối liên hệ logic về nghĩa giữa các cụm từ trong câu và mối liên hệ giữa các câu trong đoạn, hệ thống sẽ xác định được (một phần) ý nghĩa của câu trong ngữ cảnh của cả đoạn.
4. *Phân tích thực chứng:* Quá trình phân tích thực chứng nhằm xác định ý nghĩa câu dựa trên mối liên hệ của câu với hiện thực. Ý nghĩa thực tế của câu phụ thuộc rất nhiều vào ngữ cảnh diễn ra lời nói. Do vậy, quá trình phân tích thực chứng rất khó thực hiện bằng máy tính. Thông thường, việc phân tích câu thường chỉ dừng ở mức phân tích ngữ nghĩa, còn việc phân tích thực chứng do người dùng tự quyết định.

1.2 Đặc điểm tiếng Việt

1.2.1 Chính tả chưa thống nhất [3]

Chính tả tiếng Việt đã có một hệ thống các quy tắc chuẩn mực. Tuy vậy, vẫn còn một số từ tồn tại nhiều cách viết khác nhau do các sai khác về phương ngữ, vị trí dấu của từ, cách viết danh từ riêng, phiên âm tiếng nước ngoài, ... Ngay với một người cũng có lúc viết thế này, có lúc viết thế khác. Cách viết không thống nhất như vậy sẽ gây nhiều khó khăn trong việc kiểm tra chính tả.

1.2.2 Trật tự các yếu tố trong chuỗi lời nói theo hướng thuận [7]

Trong tiếng Việt, trật tự của các yếu tố trong câu được quy định bằng những vị trí nhất định. Khi vị trí đổi thì nghĩa cũng thay đổi theo.

1.2.3 Vai trò của tổ hợp từ cố định trong câu

Trong tiếng Việt có các tổ hợp từ cố định được gọi là thành ngữ hay quán ngữ (ví dụ "con cà con kê"). Đây là cách nói quen thuộc và được chấp nhận một cách hiển nhiên. Những tổ hợp từ này nhiều khi không tuân theo các qui luật về cú pháp và ngữ nghĩa.

1.2.4 Vấn đề đa nghĩa và nhập nhằng trong ngôn ngữ [6]

Phân tích cú pháp cho chúng ta điểm khởi đầu để tìm ra ý nghĩa của toàn bộ câu. Khi chỉ có một cách phân tích thì việc tìm ra ý nghĩa câu khá đơn giản. Nhưng khi có nhiều cách phân tích, công việc trở nên khó khăn hơn.

Ví dụ: Câu "*Cậu bé đi cùng Hoa là cháu tôi*." có thể hiểu theo hai cách:

Cách 1: Cậu bé // đi cùng (Hoa / là cháu tôi).
Cách 2: (Cậu bé / đi cùng Hoa) // là cháu tôi .

Sự mập mờ nói trên thực chất là mập mờ ngữ nghĩa-cú pháp. Giải pháp xoá bỏ mập mờ có thể giải quyết một phần nhờ vào ngữ nghĩa đoạn. Tuy nhiên, trong nhiều trường hợp, ta không thể dựa vào văn cảnh mà phải có sự can thiệp trực giác của chủ thể phát ngôn.

1.3 Tổng quan về các phần mềm phân tích văn bản hiện có

Hiện nay, ở nước ta, các hệ thống kiểm soát văn bản còn tương đối ít và được cài đặt như một chức năng trong phần mềm soạn thảo. Các phần mềm tiếng Anh như WORDPERFECT, WINWORD có thể tự động sửa lỗi chính tả (chức năng AutoCorrect). Việc sửa lỗi chính tả được thực hiện theo kiểu tương ứng một - một giữa từ viết sai và từ viết đúng, dựa trên bảng liệt kê các từ sai và từ đúng tương ứng (giống chức năng Find and Replace). Việc kiểm tra chính tả (Spelling) được thực hiện dựa trên từ điển bằng cách so sánh các từ trong văn bản và từ điển [2].

Với tiếng Việt, chúng ta có BKED, VIETRES dựa trên luật cấu tạo âm tiết của tiếng Việt để tìm ra các chữ không phải là âm tiết tiếng Việt. VIETBIT sử dụng cách tra cứu theo từ điển từ đơn. Tuy nhiên, các phương pháp trên không thể phát hiện được các từ ghép sai [2]. Phần mềm CADPRO OFFICE 2.0 của trung tâm CadPro dựa trên luật cấu tạo âm tiết và từ điển từ (đơn, ghép). Hệ thống đã bước đầu thực hiện việc xác định các từ trong câu dựa trên từ điển từ tiếng Việt. Hệ thống cho phép chỉnh sai chính tả và đưa ra các gợi ý cần thiết [7].

Nhìn chung, việc kiểm tra chính tả tiếng Việt cho đến nay đã có một số thành công nhất định ở mức âm tiết , mức từ vựng vẫn còn nhiều hạn chế. Trong khi đó, các chương trình kiểm tra cú pháp tiếng Việt rất ít và còn nhiều vấn đề cần được nghiên cứu giải quyết.

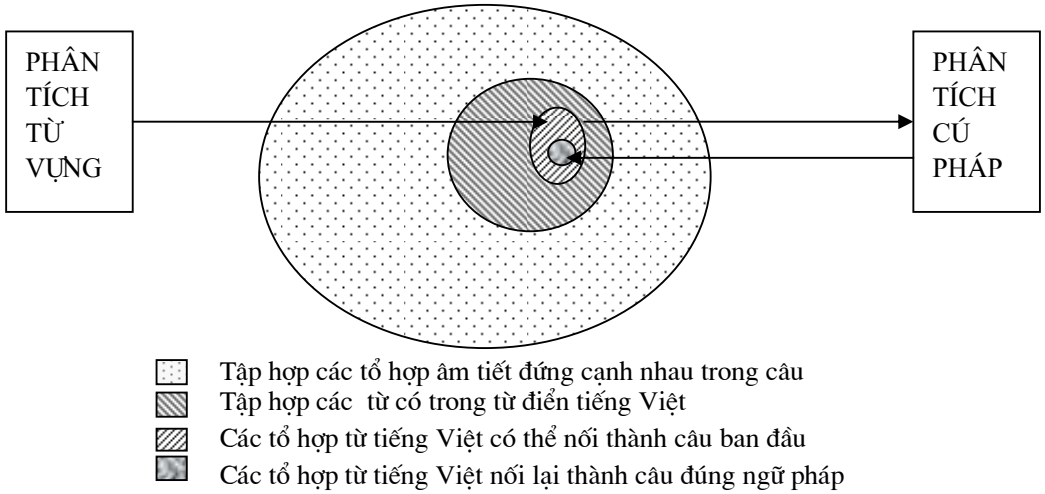
1.4 Cách giải quyết

Với nhóm ngôn ngữ đa âm tiết, khi kết thúc giai đoạn phân tích từ vựng, ta đã xác định được các từ tạo nên câu và hình thái tương ứng của chúng. Nhưng với tiếng Việt, ta không thể làm được việc này vì lý do sau:

Tiếng Việt là ngôn ngữ **đơn âm tiết**, giữa từ và âm tiết không có ranh giới rõ rệt. Do vậy, khi tách được đơn vị nằm giữa các dấu phân cách trong câu, ta vẫn chưa thể xác định được từ. Để xác định từ tiếng Việt, ta phải kết hợp các âm tiết đứng cạnh nhau trong câu thành các tổ hợp âm tiết và so sánh các tổ hợp âm tiết đó với các từ sẵn có của tiếng Việt. Nếu tổ hợp âm tiết đó có tồn tại trong vốn từ vựng tiếng Việt thì mới kết luận được từ đúng.

Có rất nhiều tổ hợp các âm tiết đứng cạnh nhau trong câu, nhưng chỉ một phần nhỏ trong các cách đó tạo ra các từ đúng với ngôn ngữ Việt. Tuy nhiên, một số cách ghép này chỉ đúng trên phương diện từ, còn xét trên phương diện cấu trúc cú pháp câu lại trở thành sai. Điều này cho thấy việc xác định đúng các từ trong câu tiếng Việt không thể tách rời quá trình phân tích cú pháp. Dựa trên quá trình phân tích cú pháp, ta mới có thể xác định một cách chính xác các từ trong câu.

Sau giai đoạn phân tích từ vựng cho ngôn ngữ tiếng Việt, ta thu được dãy các âm tiết đúng và các khả năng kết hợp từ *có thể* tạo nên câu. Tổ hợp các từ đúng với câu trên phương diện ngữ pháp và hình thái tương ứng của chúng chỉ được xác định trong quá trình phân tích cú pháp. Hình vẽ dưới đây sẽ cho ta thấy bức tranh về quá trình xác định tổ hợp từ đúng của một câu tiếng Việt.



Các chương trình phân tích văn bản tiếng Việt từ trước tới nay mới chỉ dựa trên luật cấu tạo âm tiết và từ điển từ mà bỏ qua mối liên hệ chặt chẽ giữa phân tích từ vựng và phân tích cú pháp. Vì vậy, các hệ thống này thường chọn lựa giải pháp tách biệt hai giai đoạn này. *Đầu ra của quá trình phân tích từ*

vựng sẽ là một mảng các cách tách từ có thể mà bỏ qua vai trò cú pháp của chúng trong câu. Đây là một trong những nguyên nhân gây ra hiện tượng bùng nổ tổ hợp trong phân tích cú pháp.

Ta sẽ tập trung vào **PHÂN TÍCH TỪ VÀ PHÂN TÍCH CÚ PHÁP**, *chú trọng tới mối quan hệ qua lại giữa hai giai đoạn này.* Thông qua quá trình phân tích cú pháp, hệ thống không chỉ đưa ra được cách phân tích từ đúng với câu mà cả mô tả hình thái ngữ pháp của câu đó.

Mục đích của hệ thống phân tích văn bản là đảm bảo tính chính xác về từ và cú pháp, hướng tới phân tích ngữ nghĩa. Thuật toán được xây dựng giải quyết được vấn đề nhập nhằng ngôn ngữ và bùng nổ tổ hợp khi phân tích.

2 Phân tích từ vựng

Để phân tích từ vựng, trước tiên ta phải tiến hành phân tích âm tiết. Một từ tiếng Việt cấu tạo bởi một hay nhiều âm tiết.

2.1 Phân tích âm tiết

Để phân tích âm tiết tiếng Việt, ta phải dựa trên luật cấu tạo âm tiết. Một âm tiết tiếng Việt đầy đủ gồm năm thành phần: nhóm phụ âm tiền âm tiết, bán nguyên âm tròn môi, nhóm nguyên âm, dấu thanh và nhóm phụ âm kết thúc âm tiết. Mỗi âm tiết tiếng Việt không nhất thiết phải có đầy đủ cả năm thành phần trên nhưng bắt buộc phải có nhóm nguyên âm.

Vì số lượng các cụm nguyên âm, phụ âm trong tiếng Việt không quá lớn nên ta có thể liệt kê được. Ngoài ra, các nguyên âm, phụ âm chỉ có thể kết hợp với nhau theo một số quy luật nhất định. Vì vậy, ta có thể phân tích âm tiết dựa trên bảng các nguyên âm, phụ âm trong tiếng Việt và luật cấu tạo âm tiết. Nếu một âm tiết nào đó không phân tích được theo luật cấu tạo âm tiết và bảng các nguyên âm, phụ âm tiếng Việt thì ta có thể kết luật âm tiết đó sai.

2.2 Phân tích từ

Việc phân tích âm tiết dựa trên luật cấu tạo âm tiết chưa đủ để khẳng định được đó đúng là âm tiết tiếng Việt vì có những âm tiết thoả mãn luật cấu tạo nhưng không tồn tại trong vốn ngôn ngữ Việt. Âm tiết chỉ có nghĩa nếu bản thân nó hoặc kết hợp với các âm tiết lân cận để tạo nên từ. Vì vậy, để phân tích tính chính xác của văn bản, sau khi phân tích âm tiết, ta phải tiến hành phân tích từ. Để phân tích từ, trước hết ta phải xác định trong văn bản đâu là từ. Từ là đơn vị có sẵn của ngôn ngữ mà sự hiện diện của nó được mọi người ngầm định. Hầu như không có quy tắc cấu tạo đối với từ. Do vậy, ta phải dùng phương pháp so sánh các từ xuất hiện trong quá trình tách từ trong văn bản với các từ thực tế để kiểm nghiệm từ đó có đúng hay không. Vì số lượng từ vựng tiếng Việt không phải là vô hạn nên ta có thể lưu vào từ điển từ. Các từ mới phát sinh có thể tiếp tục được bổ sung vào sau.

Việc chia câu thành dãy từ phải đảm bảo ba điều kiện: tất cả các thành phần đều là các từ có nghĩa; không có âm tiết nào là thành phần của hai từ khác nhau; chuỗi các từ kế tiếp nhau phải tạo thành câu ban đầu. Nói cách khác, chia câu thành dãy từ nghĩa là cắt câu thành những nhóm âm tiết, mỗi nhóm âm tiết tạo thành một từ có nghĩa trong tiếng Việt. Việc chia này phải đảm bảo không bỏ sót từ.

Nếu quan niệm từ là tổ hợp các âm tiết thì với một câu có n âm tiết, ta sẽ có tối đa $n(n+1)/2$ từ và 2^{n-1} cách tách từ khác nhau. Nếu mỗi từ có tối đa k kiểu từ loại (hay k hình vị), ta sẽ có tối đa $k(k+1)^{n-1}$ cấu trúc hình thái câu khác nhau. Như vậy, ta thấy khi số âm tiết tăng, số cách tách từ và nhất là số câu hình thái tăng lên rất nhanh (theo hàm mũ). Vấn đề đặt ra là phải làm cách nào để giảm số lượng câu hình thái sinh ra trong bước này [4].

Để giải quyết bài toán này, ta sử dụng các phương pháp sau:

- Khi phát hiện thấy một cách tách từ nào đó không phù hợp (trước tiên dựa trên việc kiểm tra đó có phải là từ tiếng Việt hay không), ta phải loại bỏ tất cả các nhánh xuất phát từ cách tách từ đó.
- Dựa trên quan hệ cú pháp giữa các từ trong câu để loại bỏ những trường hợp bất qui tắc.

Ta có thể thấy việc chia câu thành dãy từ không chỉ đơn giản là nhóm các âm tiết cạnh nhau rồi tìm trong từ điển. Mỗi từ đều có một vai trò xác định trong câu. *Để xác định được đâu là từ của câu, cần căn cứ vào mối liên hệ của nó với các đối tượng lân cận.*

Như vậy, ta có thể thấy mối liên hệ chặt chẽ giữa bước chia câu thành dãy từ và bước phân tích cú pháp tiếp theo. Một cách chia có thể đúng hoặc sai theo cú pháp. Việc kiểm tra này do bước phân tích cú pháp quyết định. Bước chia câu sẽ tạo đầu vào cho bước phân tích cú pháp tiếp theo. Trong giai đoạn này, *các thuật toán phân tích trước đây thường liệt kê các trường hợp có thể*. Đây là việc tổ hợp (vô

điều kiện) các từ đứng cạnh nhau trong câu. Như vậy sẽ rất tốn bộ nhớ và thời gian xử lý. Như trên đã nói, mỗi từ trong câu có quan hệ chặt chẽ với các từ khác về mặt cú pháp. Vì vậy, trong giai đoạn này, ta không liệt kê các cách tách câu thành dãy từ. Thay vào đó, ta *chỉ tạo ra một danh sách liên kết tất cả các từ có mặt trong câu tìm thấy trong từ điển từ.* Ta có thể thấy rằng, tuy vốn từ tiếng Việt rất lớn nhưng các từ có thể nằm trong một câu cụ thể nào đó lại khá nhỏ. Do đó, danh sách liên kết nói trên sẽ không quá dài.

2.3 Tổ chức từ điển

Trong giai đoạn phân tích từ vựng, ta dùng đến từ điển từ. Từ điển từ lưu các từ đúng trong tiếng Việt. Từ điển được tổ chức theo kiểu bảng băm. Từ điển được chia thành từng trang theo hai chữ cái đầu của từ. Trong từ điển, ta tổ chức hai danh sách liên kết:

- Một danh sách (có tên Paragraph) dùng để lưu tên các trang (hai chữ cái đầu của các từ trong trang), được sắp theo chiều tăng.
- Một mảng các danh sách (có tên Contence) để lưu nội dung các trang, được sắp theo chiều tăng.

Việc dùng các danh sách thay cho mảng sẽ giúp ta tiết kiệm bộ nhớ và không bị hạn chế bởi số lượng các phần tử.

2.4 Thuật toán tìm từ trong từ điển

Khi tìm từ trong từ điển, ta thường vấp phải một vấn đề là độ dài từ ghép. Một số tài liệu chọn độ dài tối đa là 3. Nhưng như vậy, khi trong câu xuất hiện các tổ hợp từ cố định trong tiếng Việt, hệ thống sẽ có sai sót. Trong chương trình này, ta sẽ không đưa ra một ngưỡng cố định để ghép từ. Thay vì phép kiểm tra từ trong từ điển có trùng xâu vào không, ta sẽ thực hiện phép so sánh từ trong từ điển với phần đầu của xâu vào. Độ dài của từ tìm được trong xâu bằng độ dài của từ tìm thấy trong từ điển. Việc tìm kiếm từ bắt đầu bằng việc tìm trang ứng với hai chữ cái đầu của từ, tiến hành theo kiểu tìm kiếm nhị phân. Khi đã xác định trang, ta lại thực hiện phép tìm kiếm nhị phân trên trang đó để tìm từ.

Đánh giá độ phức tạp của thuật toán tìm từ trong từ điển

Với từ điển có n mục từ được chia làm m trang (theo hai chữ cái đầu), thời gian tìm được trang là $\log_2 m$ (do việc tìm kiếm được tiến hành theo kiểu nhị phân). Với mỗi trang, ta lại tiến hành phép tìm kiếm nhị phân để tìm từ. Thời gian trung bình để tìm từ trong trang là $\log_2(n/m)$. Do đó, thời gian tìm từ trong từ điển là: $\log_2(n/m) + \log_2 m = \log_2 n$ lần.

Như vậy, độ phức tạp của thuật toán là $O(\log_2 n)$.

3 Phân tích cú pháp

3.1 Lựa chọn văn phạm biểu diễn ngôn ngữ tiếng Việt

Theo cách phân loại của Chomsky, văn phạm được chia thành bốn nhóm sau [1]:

Nhóm 0 - ***Văn phạm ngữ cấu***: mọi qui tắc $r \in R$ đều có dạng $r = \alpha \rightarrow \beta$, trong đó $\alpha \in V^+$, $\beta \in V^*$với $V = T \cup N$.

Nhóm 1 - ***Văn phạm cảm ngữ cảnh***: mọi qui tắc $r \in R$ đều có dạng $r = \alpha \rightarrow \beta$, trong đó $\alpha \in V^+$, $\beta \in V^*$ và $|\alpha| \leq |\beta|$. 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. Thuật ngữ "cảm ngữ cảnh" có xuất xứ từ dạng chuẩn của văn phạm này là $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$ với $\beta \neq \varepsilon$, cho thấy một biến A được thay thế bởi một xâu β (khác rỗng) trong ngữ cảnh α_1-α_2.

Nhóm 2 - ***Văn phạm phi ngữ cảnh***: mọi qui tắc $r \in R$ đều có dạng $r = A \rightarrow \theta$, $A \in N$, $\theta \in V^*$. Văn phạm này gọi là phi ngữ cảnh hay ngữ cảnh tự do vì biến A có thể tự do sinh ra xâu θ mà không phụ thuộc vào nó nằm trong ngữ cảnh gì.

Nhóm 3 - ***Văn phạm chính quy***: gồm các qui tắc có dạng $A \rightarrow aB$, $A \rightarrow Ba$, $A \rightarrow a$, với $A, B \in N$, $a \in T$.

Văn phạm lựa chọn để biểu diễn tiếng Việt phải vừa đủ để giải quyết bài toán. Tính chất vừa đủ ở đây có nghĩa là không gian hay phạm vi hoạt động của nó đủ để bao quát các trường hợp của ngôn ngữ tự nhiên, nhưng cũng không nên quá rộng. Theo cách làm truyền thống, người ta thường dùng văn phạm phi ngữ cảnh để biểu diễn ngôn ngữ tự nhiên. Tuy nhiên, trong nhiều trường hợp, văn phạm phi ngữ cảnh không đủ để mô tả ngôn ngữ tiếng Việt mà phải dùng đến văn phạm cảm ngữ cảnh. Ngôn ngữ tự nhiên rất đa dạng, phong phú, có nhiều trường hợp ta không thể tìm được qui luật để biểu diễn nó. Có những trường hợp bất qui tắc mà bản thân văn phạm ngữ cấu (nhóm 0) cũng không đủ mạnh để giải quyết được. Vì vậy, ta không thể hy vọng có thể biểu diễn được mọi trường hợp của ngôn ngữ tự nhiên.

Thay vào đó, ta sẽ tìm một phương án *khả thi nhất* và *đủ chặt chẽ* để có thể biểu diễn *tối đa* các trường hợp của ngôn ngữ tự nhiên.

Theo [1], việc nhận biết một văn phạm cảm ngữ cảnh phức tạp hơn rất nhiều so với văn phạm phi ngữ cảnh. Vì vậy, ta không dùng văn phạm cảm ngữ cảnh để biểu diễn ngôn ngữ mà ta sẽ tìm các biện pháp ***mở rộng văn phạm phi ngữ cảnh*** để có thể giải quyết được các trường hợp của văn phạm cảm ngữ cảnh và xa hơn nữa [3]. Ở đây, ta sử dụng biện pháp bổ sung thông tin cho các ký hiệu không kết thúc. Thông tin này nhằm kiểm soát mối quan hệ giữa các cụm từ trong câu.

Trong chương trình này, việc này được thể hiện qua hai mức:

- **Bổ sung các ký hiệu không kết thúc**. Bản thân tên gọi của các ký hiệu không kết thúc cũng phản ánh một phần các thông tin về ngữ cảnh của chúng.
- Mỗi ký hiệu không kết thúc được **gắn thêm một thuộc tính**. Thuộc tính này cho biết mối tương quan và ý nghĩa của ký hiệu không kết thúc đó (ứng với một cụm từ) với các thành phần khác trong câu.

3.2 Xây dựng thuật toán phân tích cú pháp

Với văn phạm phi ngữ cảnh, ta có bốn thuật toán phân tích điển hình [5] là thuật toán phân tích Top - Down, thuật toán Bottom - Up, thuật toán Cocke - Younger - Kasami (CYK) và thuật toán Earley. Hai thuật toán phân tích Top - Down và Bottom - Up cài đặt tương đối đơn giản nhưng có độ phức tạp cao (độ phức tạp hàm mũ). Thuật toán Cocke - Younger - Kasami và Earley phức tạp hơn so với hai thuật toán trên nhưng đổi lại, độ phức tạp của chúng nhỏ hơn rất nhiều. Mỗi thuật toán đòi hỏi n^3 thời gian và n^2 bộ nhớ, với n là độ dài của xâu vào. Chương trình phân tích cú pháp này định hướng phân tích theo chiến lược Bottom - Up và từ trái sang nên ta sẽ sử dụng thuật toán CYK làm cơ sở để xây dựng thuật toán phân tích cú pháp. Tuy nhiên, khi phân tích cú pháp tiếng Việt, thuật toán CYK [5] gặp một số hạn chế sau:

- Dạng luật văn phạm: Thuật toán *chỉ làm việc tốt với các luật ở dạng chuẩn Chomsky* (dạng $A \rightarrow BC$ và $A \rightarrow \varpi$ với ϖ là xâu các ký hiệu kết thúc). Trong khi đó, với ngôn ngữ tự nhiên, ta gặp rất nhiều các luật cú pháp không ở dạng này.
- Với thuật toán phân tích trái từ bảng phân tích, để dựng được cây phân tích, ta *phải tìm lại các luật một lần nữa (phải tra lại từ điển)*. Điều này sẽ làm mất thời gian thực hiện chương trình.
- Thuật toán *chỉ đưa ra một cây phân tích*. Việc chọn tuỳ ý một sản xuất với số thứ tự m nhỏ nhất sẽ làm mất các phân tích câu. Đôi khi đó mới là câu phân tích đúng với ngữ cảnh của văn bản.

Chương trình được xây dựng sẽ giải quyết các vấn đề trên.

- <u>Vấn đề thứ nhất: Dang luật văn phạm:</u> Bộ luật của từ điển có các luật mà vế phải không là bộ đôi mà là bộ ba, bộ bốn,... ví dụ $A \rightarrow BCDE$. Cách giải quyết đầu tiên mà ta nghĩ đến là đưa các luật của ta về dạng chuẩn Chomsky. Chẳng hạn với $A \rightarrow BCDE$, ta chuyển thành các luật sau: $A \rightarrow MN$; $M \rightarrow BC$; $N \rightarrow DE$. Với phương pháp này, ta phải sử dụng thêm các ký hiệu không kết thúc M, N làm các ký hiệu trung gian. Đồng thời, ta phải sử dụng một tập luật để thay thế cho luật ban đầu. Phương pháp này có một số nhược điểm sau:
  - Việc bổ sung thêm các ký hiệu trung gian và các luật sẽ làm tăng kích thước bộ nhớ.
  - Các ký hiệu trung gian này thường là các ký hiệu không có ý nghĩa và không có chức năng cú pháp trong câu. Các ký hiệu này không có sẵn trong cơ sở dữ liệu luật cú pháp mà được sinh ra trong quá trình chuẩn hoá các luật. Việc quyết định dạng biểu diễn cho các ký hiệu trung gian này tương đối khó. Nếu sinh các ký hiệu và các luật trung gian một cách tuỳ tiện sẽ làm tăng kích thước và gây nhiễu cho chương trình.
  - Việc phân tách một luật thành nhiều luật nhỏ sẽ làm mất ý nghĩa của luật đó.

  Để không làm tăng số lượng luật, không dùng các ký hiệu vô nghĩa về cú pháp và không làm mất ý nghĩa của luật, ta giải quyết như sau:

  Tại bước 2 của thuật toán CYK, thay vì tìm các luật dạng $A \rightarrow BC$, ta sẽ tìm các luật mà phần đầu vế phải của nó (head) là BC: $A \rightarrow BCDE$. Phần còn lại (tail) sẽ được quản lý thông qua biến expect (đi kèm theo mỗi phần tử của bảng). Tại lượt phân tích sau của bảng phân tích, chương trình sẽ làm phép so sánh $FIRST_1$(expect) với đối tượng muốn ghép. Nếu phần tử được tính cuối cùng của bảng có chứa ký hiệu khởi đầu S và expect(S) = 0 thì xâu vào đúng; ngược lại xâu vào sai.

- Vấn đề thứ hai: Vẽ cây phân tích: Trong thuật toán CYK, ta thấy khi dựng lại cây phân tích, hệ thống lại phải tra từ điển để tìm suy dẫn mà việc này thực chất đã được thực hiện trong thuật toán kiểm tra xâu vào. Để tránh lặp lại việc đó, ta sử dụng một biến đi kèm theo mỗi phần tử của bảng để lưu vết về tổ hợp tạo nên nó. Tổ hợp này chỉ bao gồm các phần tử ở mức ngay dưới nó. Từ tổ hợp này, ta có thể suy ngược đến các ký hiệu kết thúc của xâu vào.
- Vấn đề thứ ba: Mất các phân tích câu: Để giải quyết việc này, ta sẽ tổ chức mảng để quản lý các câu phân tích đúng.

Nội dung thuật toán phân tích cú pháp như sau:

Đầu vào của thuật toán phân tích cú pháp là mảng các từ trong câu và hình thái tương ứng của chúng. Các thành phần của mảng sẽ được kết hợp với nhau trên cơ sở các luật cú pháp có trong từ điển. Thuật toán duyệt từ đầu đến cuối mảng. Tại một vị trí, nó sẽ xét các khả năng kết hợp của từ với các từ cạnh nó trong câu. Nếu thoả mãn, tổ hợp từ này sẽ được bổ sung vào mảng. Quá trình này kết thúc khi không thể bổ sung thêm phần tử nào vào mảng.

Khi đã hoàn thành giai đoạn tạo tất cả các cấu trúc ngữ pháp có thể từ xâu vào dựa trên từ điển từ điển từ và từ điển hình thái, dựa trên mảng phân tích cuối cùng, ta có thể biết xâu vào có phải là câu đúng ngữ pháp hay không. Một câu đúng ngữ pháp nếu trong mảng phân tích cuối cùng có phần tử là tổ hợp các từ, từ từ đầu câu đến từ cuối câu. Tổ hợp này có nghĩa cú pháp là câu và không còn expect.

Độ phức tạp của thuật toán

Thuật toán phân tích cú pháp được xây dựng trên cơ sở thuật toán CYK nên có độ phức tạp là $O(n^3)$ với n là số từ trong câu.

3.3 Xử lý nhập nhằng trong phân tích cú pháp

Chương trình không chọn phương án lựa chọn một trong các khả năng mà tìm ra tất cả các cách phân tích có thể, dựa trên bộ luật văn phạm cho tiếng Việt. Các cách phân tích này sẽ tiếp tục được xử lý trong giai đoạn phân tích ngữ nghĩa để tìm ra cách phân tích chính xác nhất. Tuy nhiên, hệ thống cũng có đánh giá trọng số của từng cách phân tích. Trọng số được xác định dựa trên tính đúng đắn về cú pháp và tính đơn giản của câu. Tính đơn giản thể hiện ở chỗ câu được phân tích có tổng số tổ hợp ghép nối và luật cú pháp sử dụng là ít nhất. Nói cách khác, cây phân tích của câu đó có độ dài ngắn nhất. Chương trình phân tích cú pháp sẽ đưa cho người sử dụng câu đúng nhất và đơn giản nhất dựa trên trọng số.

3.4 Khả năng học

Do tiếng Việt không tĩnh tại mà nó không ngừng phát triển nên ta không thể nói đến một từ điển luật cú pháp đầy đủ. Vì vậy, khối phân tích cú pháp nên xây dựng theo kiểu hệ mở, có thể dễ dàng bổ sung thêm các luật. Đối với chương trình này, có hai con đường để nhập liệu vào từ điển cú pháp: hoặc trực tiếp qua phần giao diện nhập liệu cho từ điển, hoặc gián tiếp thông qua việc học của khối phân tích cú pháp.

Tư tưởng về việc học của khối phân tích cú pháp như sau: Khi phân tích câu, hệ thống sẽ tra cứu các luật cú pháp trong từ điển và dựng cây cú pháp. Nếu câu sai, hệ thống sẽ đưa ra câu gần đúng nhất. Lỗi sai có thể xảy ra do xâu vào sai hoặc do từ điển không đủ luật cú pháp. Nếu từ điển thiếu, người sử dụng có thể đưa ra các hướng dẫn cần thiết. Trên cơ sở đó, hệ thống sẽ học và bổ sung thêm luật mới. Với lần phân tích tiếp sau, hệ thống sẽ làm việc với từ điển đã cập nhật.

4 Một số vấn đề trong cài đặt hệ thống

4.1 Tổ chức lưu trữ từ trong điển

Để thuận tiện cho việc phát triển chương trình, ta xây dựng một mô hình chung và các phép xử lý chung cho tất cả các từ điển. Từ điển được xây dựng trên cơ sở một lớp có tên Word_class có các thuộc tính chính sau:

```
class Word_class
{       char *Text;         // dạng văn bản của từ
        int CurDic;         // chỉ số của từ điển chứa từ (do có thể sử dụng cùng lúc nhiều từ điển)
        TIArrayAsVector<Meaning_class> Meanings;        // các hình thái của từ
}
```

Meanings là mảng các phần tử thuộc lớp Meaning_class có các thuộc tính cơ bản sau:

```
class Meaning_class
{       BYTE type;          // mã từ loại của từ
```

```
        char* exp;          // expect
        TISArrayAsVector <Glossarial> Gloss;
}
```

Vì từ loại không nhiều nên ta có thể quản lý chúng qua một bảng mã riêng. Từ loại được chia thành các nhóm riêng, mỗi nhóm gồm các từ có chức năng cú pháp gần giống nhau. Ta không xây dựng từ loại theo kiểu cố định mà có thể bổ sung sau này. Với mỗi từ loại, ta sẽ mã hoá chúng thông qua vị trí tương ứng của từ loại trong bảng (được lưu giữ bằng một ký tự).

Mỗi từ loại lại có thể có nhiều nghĩa loại khác nhau nên ta dùng mảng Gloss để quản lý nghĩa loại của chúng. Gloss là mảng các phần tử thuộc lớp Glossarial gồm các thuộc tính cơ bản sau:

```
class Glossarial
{       BYTE glossarial;        // mã nghĩa loại của từ
        char *Text;             // giải nghĩa từ
}
```

Với từ điển cú pháp, đơn vị cơ bản là luật cú pháp. Các luật cú pháp được xây dựng trên mô hình luật của văn phạm phi ngữ cảnh: <danh ngữ> → <loại từ> <danh từ> hay VT → VP

Khi đó, trường Text của lớp Word_class sẽ lưu VP của luật ; trường type của lớp Meaning_class lưu VT của luật ; trường glossarial của lớp Glossarial lưu nghĩa loại của luật ; trường Text của lớp Glossarial lưu các thông tin phụ thêm (nếu có).

4.2 Mã hoá từ điển

Trước khi ghi một mục từ vào từ điển, ta chuyển chúng sang dạng mã. Mỗi cấu trúc Word_class sẽ được chuyển thành một đoạn text có các ký tự điều khiển. Từ loại và nghĩa loại kiểu byte được lưu dưới dạng một ký tự. Các trường trong Word_class được phân cách nhau bằng một ký tự đặc biệt.

Từ điển cú pháp cũng được mã hoá như trên nhưng riêng với trường Text của Word_class, ta có thể thu gọn hơn nữa không gian lưu trữ. Các từ loại trong Text của Word_class có thể mã hoá bằng một byte. Trong từ điển, nó sẽ được lưu trữ bằng một ký tự. Với các từ thông thường, ta vẫn để nguyên dạng. Để phân biệt giữa từ thông thường và ký tự mã hoá của từ loại, ta dùng các ký tự đặc biệt nằm ngoài khoảng biểu diễn của từ loại và từ.

Chẳng hạn: Nếu mã của <câu> là 107, tương ứng ký tự k ; mã của <danh từ> là 112, tương ứng ký tự p; mã của <loại từ> là 115, tương ứng ký tự s ; dùng ký tự có mã 253 (∫) để báo hiệu bắt đầu chuỗi ký tự và ký tự có mã 254 (∑) báo hiệu bắt đầu chuỗi từ loại (nếu nhiều từ loại đứng liền nhau thì cũng chỉ cần một ký tự báo hiệu đứng trước từ loại đầu tiên mà thôi). Khi đó, "<loại từ> <danh từ>" sẽ được mã hoá thành "∑sp" ; "Nếu <câu> thì <câu>" sẽ được mã hoá thành "∫Nếu∑k∫thì∑k".

Với từ điển cú pháp, trường Text có số lượng từ loại lớn hơn nhiều lần số lượng từ thông thường nên việc mã hoá như vậy sẽ làm giảm kích thước từ điển rất nhiều lần.

5 Kết quả thử nghiệm

Tính chính xác của một hệ thống phân tích văn bản phụ thuộc rất nhiều vào tính đúng đắn và đầy đủ của các từ điển. Hệ thống được xây dựng sử dụng hai từ điển là từ điển hình thái từ và từ điển cú pháp. Người sử dụng có thể bổ sung thêm dữ liệu vào hai từ điển này thông qua giao diện nhập liệu ở dạng hộp thoại. Quá trình thử nghiệm cho thấy tập luật cú pháp phải rất chặt chẽ. Một câu có thể được phân tích đúng với từ điển chỉ có vài luật, nhưng khi bổ sung thêm các luật mới, chương trình lại có thể đưa ra phân tích sai. Trong khi xây dựng và cập nhật tập luật cú pháp, chúng tôi đã có cân nhắc để hạn chế tối đa những trường hợp như vậy.

Chương trình được thử nghiệm trên các trường hợp câu sai, câu đúng, câu đơn, câu ghép. Kết quả đưa ra dưới dạng cây phân tích. Sau đây là một số kết quả:

1. *Ngày nay, các thành tựu trong tin học có đóng góp lớn cho xã hội.*

Kết quả sau khi phân tích từ:

Ngày (danh từ) → ngày nay(danh trạng từ) → nay (đại từ đứng sau) → các (quán từ) → thành (danh từ) → thành tựu (danh từ) → trong (trạng từ đứng trước) → tin (động từ) → tin học (danh từ) → học (động từ) → có (đgtừ cần bổ ngữ) → đóng (đgtừ cần bổ ngữ) → đóng góp (đgtừ cần bổ ngữ) → góp (đgtừ cần bổ ngữ) → lớn (tính từ) → cho (đgtừ cần bổ ngữ; trợ từ) → xã hội (danh từ).

Kết quả sau khi phân tích cú pháp:

Ngày nay (danh trạng từ), các (quán từ) thành tựu (danh từ) trong (trạng từ đứng trước) tin học (danh từ)
danh ngữ kết thúc trái — trạng ngữ
chủ ngữ

// có (đgtừ cần bổ ngữ) đóng góp (đgtừ cần bổ ngữ) lớn (tính từ) cho (trợ từ) xã hội (danh từ).
động ngữ — giới ngữ
động ngữ
vị ngữ

Kết luận : câu đúng cú pháp.

Một số phân tích khác:
2. Các ứng dụng trong lĩnh vực này rất phong phú. KL: câu đúng cú pháp
3. Các ứng dụng trong lĩnh vực này phong phú rất. KL: câu sai cú pháp
4. Trên cơ sở nghiên cứu đặc điểm ngôn ngữ tiếng Việt, luận văn đã sẽ đề xuất phương án nhằm cải thiện các vấn đề trên. KL: câu sai cú pháp

6 Kết luận

Vấn đề phân tích văn bản nói riêng và xử lý ngôn ngữ tự nhiên nói chung là một vấn đề khó khăn và phức tạp. Để tạo ra một sản phẩm phần mềm tương đối khả quan, cần có sự nghiên cứu lâu dài và có hệ thống của con người trên cả hai vực tin học và ngôn ngữ học. Nếu các nhà tin học làm việc theo chủ quan cảm tính của mình thì sản phẩm tạo ra sẽ không thể đem ứng dụng thực tế. Mặc dù đây là lĩnh vực mà trên thế giới đã được nhiều người nghiên cứu nhưng bên trong nó vẫn còn vô số những vấn đề chưa tìm được cách xử lý hữu hiệu. Đề tài này là một nghiên cứu để góp phần giải quyết các vấn đề phức tạp đó.

6.1 Đánh giá kết quả

Đề tài xây dựng một phương pháp phân tích văn bản tiếng Việt. Việc kiểm tra chính tả từ trước đến nay đã có nhiều công trình nghiên cứu và đã có thành công nên ta không tập trung nhiều vào phần này. Phân tích cú pháp là vấn đề trọng tâm được đưa ra nghiên cứu và giải quyết. Trong phần phân tích cú pháp có những điểm chính sau:

- Việc phân tích từ và phân tích cú pháp được tiến hành đan xen để hạn chế bùng nổ tổ hợp, tăng tốc độ thực hiện chương trình; đồng thời tìm được cách phân tích chính xác các từ tạo nên câu.
- Văn phạm biểu diễn ngôn ngữ tự nhiên là một mở rộng của văn phạm phi ngữ cảnh để bao quát cả những hiện tượng trong văn phạm cảm ngữ cảnh. Để thực hiện việc này, luận văn gắn vào mỗi ký hiệu không kết thúc các thuộc tính kiểm soát sự tương ứng văn phạm của các bộ phận trong câu (văn phạm thuộc tính).
- Thuật toán phân tích cú pháp là một cải tiến của thuật toán CYK. Việc mở rộng văn phạm FNC và cải tiến thuật toán CYK cho phép mở rộng khả năng phân tích câu và tăng tính chính xác của kết quả phân tích cú pháp.
- Từ điển được mã hoá nhỏ gọn và tổ chức thuận tiện cho việc tìm kiếm. Trong từ điển, mỗi mục từ là một tổ hợp các tri thức từ. Các thông tin này góp phần làm chính xác quá trình phân cú pháp. Chúng đặc biệt quan trọng trong quá trình phân tích ngữ nghĩa.
- Xây dựng thuật toán tìm từ ghép trong văn bản hiệu quả hơn, giải quyết được vấn đề độ dài từ ghép.

Quá trình thử nghiệm cho thấy chương trình chạy khá nhanh và cho kết quả tương đối chính xác.

6.2 Hướng phát triển

Bài toán phân tích văn bản là một vấn đề khá lớn. Trong thời gian tương đối ngắn, người thực hiện không thể chuyển tải tất cả các mục đích vào chương trình. Sau đây là các vấn đề mà chúng tôi muốn tiếp tục phát triển trong thời gian tới:

- Đưa thêm các modul tiền xử lý để giảm thời gian phân tích cú pháp; phát triển khả năng học của chương trình.
- Tăng cường khả năng phân tích và sửa lỗi văn bản.
- Nghiên cứu và xây dựng khối phân tích ngữ nghĩa.
- Trên cơ sở việc xử lý một ngôn ngữ, tiến hành xây dựng hệ thống dịch tự động.

Tài liệu tham khảo

1. Nguyễn Văn Ba, Ngôn ngữ hình thức, Trường Đại học Bách khoa Hà nội, 1993.
2. Nguyễn Tùng Giang, Một cách tiếp cận kiểm tra chính tả trong các hệ soạn thảo tiếng Việt, Luận văn tốt nghiệp, Hà nội, 1995.
3. Lê Khánh Hùng, Báo cáo tổng kết đề tài "Hoàn thiện và đưa vào thực tế Hệ thống dịch tự động Anh-Việt để biên dịch tài liệu chuyên ngành, Hà nội, 3-1998.
4. Lê Thanh Hương, Phân tích cú pháp tiếng Việt, Luận văn cao học, Hà nội, 1999.
5. Vũ Lục, Phân tích cú pháp, Trường Đại học Bách khoa Hà nội, 1990.
6. Hoàng Trọng Phiến, Ngữ pháp tiếng Việt, Nhà xuất bản Đại học và Trung học chuyên nghiệp, Hà nội, 1980.
7. Phạm Hồng Quang, Dự án xây dựng phần mềm dịch máy Nhật-Việt, Hà nội, 12-1998.
8. Nguyễn Kim Thản, Nghiên cứu về ngữ pháp tiếng Việt (tập II), Nhà xuất bản Khoa học, Hà nội, 1964.
9. Trịnh Nguyên Thiều, Tóm tắt luận án phó tiến sĩ "Phương pháp và thuật toán nhận hiểu câu tiếng Việt cho các hệ thống điều khiển tự động", Odessa - 1998.
10. Nguyễn Công Tú, Một phương pháp kiểm tra chính tả tiếng Việt trên máy tính, Luận văn thạc sĩ, Hà nội, 1998.
11. Viện Ngôn ngữ học, Nhiều tác giả, Chủ biên Lưu Vân Lăng, Những vấn đề ngữ pháp tiếng Việt, Nhà xuất bản Khoa học Xã hội, Hà nội, 1988.