XÂY DỰNG HỆ THỐNG PHÂN TÍCH CẤU TRÚC ÂM TIẾT TIẾNG VIỆT SỬ DỤNG MÔ HÌNH SDSS

Trần Văn Nam¹

BUILDING A SYSTEM OF ANALYZING VIETNAMESE SYLLABLE STRUCTURE USING SDSS MODEL

Tran Van Nam¹

Tóm tắt – Tham luận giới thiệu một cách tiếp cận xây dựng hệ thống phân tích cấu trúc âm tiết tiếng Việt áp dụng mô hình cấu trúc âm tiết xác định và ràng buộc ngữ nghĩa (SDSS - Semantic Defined Syllable Structure). Cách tiếp cận này cho phép xử lý các vấn đề bùng nổ tổ hợp, loại bỏ những phương án phân tích âm tiết vô nghĩa bằng cách sử dụng các luật cấu tạo âm tiết có ràng buộc ngữ nghĩa. Tham luận cũng đề xuất cách biểu diễn và quản lý luật SDSS cho tiếng Việt dựa trên các đặc điểm riêng của ngôn ngữ này. Hướng tiếp cận hoàn toàn khả thi và có kết quả thử nghiệm đạt độ chính xác cao, loại bỏ kết quả âm tiết được phân tích không có nghĩa, góp phần giải quyết bài toán phân tích văn bản ứng dụng hiệu quả trong xử lý tiếng Việt.

Từ khóa: DSS, SDSS, phân tích cấu trúc âm tiết tiếng Việt.

Abstract – The report presents an approach to building a Vietnamese syllable structure analysis system using Semantic Defined Syllable Structure (SDSS). This approach allows to process combinatorial explosion problems, to eliminate nonsense syllabary options by using semantic-constrained text rules. The report also proposes the way of performing and managing SDSS laws for Vietnamese based on the characteristics of

this language. The approach is completely feasible with the test results in high accuracy by eliminating the nonsense syllables, contributing to solving effective text analysis problems in Vietnamese processing.

Keywords: DSS, SDSS, Vietnamese syllable structure analysis.

I. GIỚI THIỆU

Khi soạn thảo văn bản trên máy tính, người sử dụng thường có nhu cầu được hệ thống tự động kiểm tra và sửa lỗi chính tả, ngữ pháp ngay trên mỗi văn bản của mình. Nhu cầu này luôn được MS Word đáp ứng nhưng chỉ phổ biến cho các văn bản thuần túy tiếng Anh và một số ngôn ngữ châu Âu. Với các ngôn ngữ khác, người sử dụng phải tự tìm cách giải quyết bằng cách thêm, hay tích hợp vào MS Word công cụ kiểm lỗi chính tả và ngữ pháp cho hệ viết của mình. Đối với tiếng Việt, nhu cầu này rất cần thiết trong môi trường soạn thảo văn bản. Để giải quyết vấn đề này, chúng tôi phân chia hệ thống kiểm tra lỗi chính tả và ngữ pháp ra làm ba giai đoạn, giai đoạn thứ nhất là kiểm tra lỗi về âm tiết, giai đoạn thứ hai là kiểm tra về từ vựng, giai đoạn thứ ba là kiểm tra về ngữ pháp của câu trong văn bản. Báo cáo này tập trung tìm hiểu và xử lý phần kiểm tra lỗi âm tiết trong văn bản.

Ở Việt Nam hiện nay, trong quá trình soạn thảo văn bản, người sử dụng mong muốn có hệ thống tự động kiểm tra lỗi chính tả và ngữ pháp của mình nhằm giúp văn bản tránh bị sai lỗi chính tả. Cho đến nay, tuy đã có một số ít chuyên gia ở

^{1,2,3}Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh

Email: namtv@tvu.edu.vn

^{1,2,3}School of Engineering and Technology, Tra Vinh University

trong và ngoài nước nghiên cứu, giải quyết phục vụ cho nhu cầu này, nhưng kết quả đạt được vẫn chưa cao. Đỗ Bá Lâm và cộng sự trình bày về phân tích cú pháp tiếng Việt với mô hình HPSG [1], nhóm nghiên cứu của Nguyễn Tuấn Đăng trình bày về phân tích cú pháp với mô hình PDCG trong tiến trình xử lý tiếng Việt [2].

Để đáp ứng được nhu cầu trên đây, chúng tôi đề xuất hệ thống VSC (VietSpellChecker) có thể tích hợp vào MS Word cho phép kiểm tra lỗi chính tả cho văn bản tiếng Việt dựa trên hướng tiếp cận sử dụng mô hình cấu tạo âm tiết tiếng Việt. Việc kiểm tra lỗi về âm tiết sẽ hỗ trợ người sử dụng biết âm tiết được viết đúng hoặc sai. Mặt khác, đây cũng là bước xử lý quan trọng trong các bài toán hiểu ngôn ngữ tự nhiên. Nó cung cấp một nền tảng vững chắc cho việc xử lý văn bản như truy xuất thông tin, các hệ thống hỏi đáp, dịch máy.... Tùy theo từng ngôn ngữ khác nhau mà nhà nghiên cứu có thể áp dụng từng giai đoạn khác nhau.

Trên thế giới, rất nhiều công trình nghiên cứu liên quan về phân tích cú pháp như tác giả Michael Collins với mô hình PCFG [3], nhóm tác giả Christopher D. Manning trình bày về nền tảng xử lý ngôn ngữ tự nhiên [4]. Mô hình PCFG (Probabilistic Context-Free Grammar) của nhóm tác giả Michael Collins [5] xử lý trên tiếng Anh, có các độ đo Precision, Recall và F nằm trong khoảng từ 80% đến 90%. Đặc biệt, công trình nghiên cứu của nhóm tác giả Qaiser Abbas [6] phân tích cú pháp trên ngôn ngữ Urdu với mô hình PDCG (Probabilistic definite Clause Grammar) có các độ đo Precision, Recall và F nằm trong khoảng từ 88% đến 94%. Công trình nghiên cứu của nhóm Đinh Thi Thu Hương [7], “Sử dụng luật cấu tạo âm tiết tiếng Việt hai thành phần trong bài toán kiểm tra chính tả tiếng Việt”, không ràng buộc các biến phụ thuộc của phụ âm và nguyên âm, theo công bố của tác giả đạt độ chính xác là 94%. Từ những kết quả khả quan này, hệ thống VSC của chúng tôi sẽ vận dụng những ưu điểm của từng phương pháp và chỉnh sửa để áp dụng vào việc phân tích cấu trúc âm tiết trong văn bản tiếng Việt. Trong tham luận này, chúng tôi giới thiệu một cách tiếp cận áp dụng mô hình SDSS. Từ DSS cơ bản quản lý các

tập luật cấu trúc âm tiết, SDSS ràng buộc thêm biến ngữ nghĩa để tăng hiệu quả phân tích cấu trúc âm tiết cho các âm tiết trong quá trình xây dựng kho ngữ liệu âm tiết tiếng Việt.

Các khả năng nhập nhằng và sử dụng sai cấu trúc ở bước âm tiết có thể do nhiều nguyên nhân khác nhau mà người dùng đã tạo trong văn bản.

Để giải quyết vấn đề này, SDSS cần đưa thông tin cấu trúc và ngữ nghĩa vào tập luật cấu trúc âm tiết bằng cách sử dụng cấu trúc âm tiết xác định DSS, có ràng buộc thêm biến nghĩa của thành phần âm tiết gọi là SDSS. Cấu trúc này cho phép biểu diễn các mối quan hệ giữa các thành phần và làm tăng ràng buộc kết hợp. Tập luật được tích hợp cấu trúc âm tiết và ngữ nghĩa của SDSS cho phép thực hiện xử lý bùng nổ tổ hợp, nhập nhằng về cấu trúc âm tiết và các âm tiết không đúng cấu trúc trong tiếng Việt.

Phần tiếp theo của bài này được tổ chức như sau: Cách tổ chức biểu diễn SDSS cho tiếng Việt được giới thiệu ở phần 2; Phần 3 trình bày mô hình phân tích cấu trúc ứng dụng SDSS cho tiếng Việt; Các kết quả thử nghiệm được trình bày trong phần 4; Phần 5 kết luận và đề xuất hướng phát triển cho cách tiếp cận này.

II. CẤU TRÚC SDSS

Quá trình phân tích cấu trúc âm tiết của một ngôn ngữ cần có một kho ngữ liệu âm tiết. Kho ngữ liệu âm tiết là một tập hợp các âm tiết ngữ liệu đã được phân tích và chú thích cấu trúc, thường được biểu diễn dưới dạng cấu trúc cây. Ngoài chú thích cấu trúc, đôi khi người ta có thể bổ sung thêm những thông tin về mặt ngôn ngữ học có liên quan khác. Bộ chữ cái tiếng Việt gồm 29 chữ cái bao gồm 17 phụ âm đơn và 12 nguyên âm được chia thành hai dạng chữ hoa và chữ thường [8].

Bảng 1: Bảng chữ cái tiếng Việt

Phụ âm	
Chữ thường	b, c, d, đ, g, h, k, l, m, n, p, q, r, s, t, v, x.
Chữ hoa	B, C, D, Đ, G, H, K, L, M, N, P, Q, R, S, T, V, X.
Nguyên âm	
Chữ thường	i, y, u, e, o, ơ, â, ô, ơ, e, a, ă, o.
Chữ hoa	I, Y, U, E, O, Ơ, Â, Ô, Ơ, E, A, Ă, O.

Chữ tiếng Việt được hình thành từ sự kết hợp giữa các chữ cái và 6 dấu thanh: thanh bằng, thanh sắc (á), thanh huyền (à), thanh hỏi (ã), thanh ngã (ạ) và thanh nặng (ạ).

A. Cấu trúc âm tiết tiếng Việt

Trước tiên, thực hiện xây dựng kho văn bản thử nghiệm gồm tất cả âm tiết trong 30 báo cáo môn học, văn bản được lựa chọn kỹ liên quan đến chủ đề về công nghệ thông tin. Đó là các báo cáo môn học của sinh viên năm thứ 2 và 3, Trường Đại học Trà Vinh. Các báo cáo đều được sinh viên soạn thảo trong MS Word. Sau đó, chọn ngẫu nhiên 2 phần 3 số báo cáo này sử dụng làm kho ngữ liệu âm tiết và 1 phần 3 số còn lại để phục vụ cho việc thử nghiệm.

Tùy theo người sử dụng có thể phân tích âm tiết tiếng Việt thành những thành phần khác nhau để xử lý, trong báo cáo này, chúng tôi phân chia âm tiết tiếng Việt thành hai thành phần, thành phần phụ âm đầu và phần chính của âm tiết như sau [9]:

Âm tiết		
Phụ âm đầu	Thành phần chính	
	Nguyên âm, dấu thanh	Phụ âm cuối

Phụ âm đầu gồm ba phụ âm: P1, P2, P3, trong đó:

- P1 là tập hợp các phụ âm b, c, d, đ, g, h, k, l, m, n, p, q, r, s, t, v, x và B, C, D, Đ, G, H, K, L, M, N, P, Q, R, S, T, V, X;
- P2 là tập hợp các phụ âm h, g, r và H, G, R;
- P3 là tập hợp các phụ âm h và H.

Thành phần chính gồm nguyên âm, dấu và phụ âm cuối:

Nguyên âm gồm có ba nguyên âm: N1, N2, N3, trong đó:

- N1 là tập hợp các nguyên âm i, y, ư, u, ê, ô, â, ô, e, a, ă, o và I, Y, Ư, U, Ê, Ô, Â, Ô, E, A, Ẫ, O;
- N2 là tập hợp các nguyên âm i, y, u, ê, ô, â, ô, e, a, ă, o và I, Y, U, Ê, Ô, Â, Ô, E, A, Ẫ, O ;
- N3 là tập hợp các nguyên âm i, y, u, ê, a, o và I, Y, U, Ê, A, O.

Nguyên âm có thể kết hợp với dấu tương ứng tùy theo từng trường hợp.

Phụ âm cuối gồm hai phụ âm: P4, P5, trong đó:

- P4 là tập hợp các phụ âm c, m, n, p, t và C, M, N, P, T;
- P5 là tập hợp các phụ âm h, g và H, G.

Các thành phần trong một kho ngữ liệu âm tiết gồm: nhãn phụ âm và nguyên âm, nhãn thành phần âm tiết, nhãn chức năng âm tiết.

Thiết kế nhãn dựa vào công trình của Nguyễn Cao Đàm [10] đã được xây dựng bằng phương pháp thủ công.

Bảng 2: Nhãn phụ âm và nguyên âm trong tiếng Việt

STT	Nhãn	Phụ âm và nguyên âm
1	P1	Phụ âm đầu thứ nhất
2	P2	Phụ âm đầu thứ hai
3	P3	Phụ âm đầu thứ ba
4	P4	Phụ âm cuối thứ nhất
5	P5	Phụ âm cuối thứ hai
6	N1	Nguyên âm thứ nhất
7	N2	Nguyên âm thứ hai
8	N3	Nguyên âm thứ ba

Bảng 3: Nhãn thành phần và âm tiết tiếng Việt

STT	Nhãn	Thành phần trong âm tiết và âm tiết
1	A	Âm tiết
2	Pd	Thành phần phụ âm đầu
3	CH	Thành phần chính
4	N	Thành phần nguyên âm
5	Pc	Thành phần phụ âm cuối

Bảng 4: Các quy tắc âm tiết

STT	Quy tắc âm tiết
1	A -> Pd CH
2	A -> CH

B. DSS cơ bản

DSS dùng để biểu diễn cấu trúc âm tiết trên ngôn ngữ lập trình Prolog. Đối với DSS, không cần phải xây dựng các thuật toán phân tích cấu trúc vì Prolog đã có cơ chế xử lý – cơ chế danh sách hiệu [11].

Bảng 5: Các quy tắc thành phần

STT	Quy tắc âm tiết
Thành phần phụ âm đầu	
1	Pd -> P1
2	Pd -> P1 P2
3	Pd -> P1 P2 P3
Thành phần chính	
1	CH -> N
2	CH -> N Pc
Thành phần nguyên âm	
1	N -> N1
2	N -> N1 N2
3	N -> N1 N2 N3
Thành phần phụ âm cuối	
1	Pc -> P4
2	Pc -> P4 P5

DSS gồm bốn thành phần <N, T, R, S> như sau:

- N là tập hợp các kí hiệu chưa kết thúc.
- T là tập hợp các kí hiệu kết thúc.
- R là tập hợp các quy tắc dẫn có dạng LHS ->RHS.
- S là kí hiệu bắt đầu.

Ví dụ minh họa DSS đơn giản cho âm tiết “trường”. Nó sẽ phân tích thành dạng như sau: [[[t]P1[r]P2]Pd [[[u]N1[ờ]N2]N [[n]P4[g]P5]Pc]CH]A và cây cấu trúc tương ứng.

Hình 1: Cây cấu trúc âm tiết DSS của âm tiết “trường”.

Các thành phần trong Hình 2:

Pd: Thành phần đầu

CH: Thành phần chính

N: Thành phần nguyên âm

Pc: Thành phần cuối

P1: Phụ âm đầu thứ nhất

P2: Phụ âm đầu thứ hai

P4: Phụ âm cuối thứ nhất

P5: Phụ âm cuối thứ hai

N1: Nguyên âm thứ nhất

N2: Nguyên âm thứ hai

Từ cây cú pháp cơ bản ở Hình 2 được minh họa cấu trúc DSS như sau:

- N = A, Pd, CH, N, Pc, P1, P2, P4, P5, N1, N2.

- T = t, r, n, g, ư, ờ

- R = A -> Pd CH, Pd -> P1 P2, CH -> N Pc, N ->N1 N2, Pc-> P4 P5

- S = A.

Cấu trúc âm tiết DSS được biểu diễn trên Prolog như sau:

```

a(a(Pd,CH)) -->pd(Pd),ch(CH).
pd(pd(P1,P2)) -->p1(P1),p2(P2).
ch(ch(N,Pc)) -->n(N),pc(Pc).
n(n(N1,N2))-->n1(N1),n2(N2).
pc(pc(P4,P5)) -->p4(P4),p5(P5).
p1(p1(t)) -->[t].
p2(p2(r)) -->[r].
p4(p4(n)) -->[n].
p5(p5(g)) -->[g].
n1(n1(u)) -->[u].
n2(n2(o)) -->[o].
  
```

Hình 2: Minh họa cho cấu trúc âm tiết DSS

Khi phân tích một chuỗi a trên văn phạm DSS được thực hiện bởi câu truy vấn:

?- a(T,L,[]).

với {
 T là cấu trúc âm tiết trả về
 L là danh sách chứa các phụ âm và nguyên âm

Ta có thể truy vấn kiểm tra âm tiết “trường”, với câu lệnh $a(T, [t, r, u, ờ, n, q], [])$, kết quả trả về đúng cấu trúc âm tiết như:

$T = a(pd(p1(t), p2(r)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(g))))$.

Danh sách L gồm 6 phần tử $t, r, u, ờ, n, g$. Khi truy vấn một cách tổng quát 6 phần tử trong danh sách L, các cây cấu trúc được trả về như Bảng 6. Tất cả các cây đều đúng cấu trúc, nhưng về nghĩa thì có một số cây có giá trị và một số âm tiết không có giá trị.

Bảng 6: Kết quả truy vấn âm tiết có 6 phần tử trên DSS

Số cây cấu trúc được sinh ra	Có giá trị	Không có giá trị
$T1 = a(pd(p1(t), p2(r)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(h))))$.		x
$T2 = a(pd(p1(t), p2(r)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(g))))$.	x	
$T3 = a(pd(p1(t), p2(r)), ch(n(n1(u), n2(ờ)), pc(p4(m), p5(h))))$.		x
$T4 = a(pd(p1(t), p2(h)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(h))))$.		x
$T5 = a(pd(p1(t), p2(h)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(g))))$.	x	
$T6 = a(pd(p1(m), p2(h)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(g))))$.		x
$T7 = a(pd(p1(b), p2(h)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(g))))$.		x
$T8 = a(pd(p1(d), p2(h)), ch(n(n1(u), n2(ờ)), pc(p4(n), p5(g))))$.		x
$T9 = a(pd(p1(t), p2(r)), ch(n(n1(a), n2(ờ)), pc(p4(n), p5(g))))$.		x
$T10 = a(pd(p1(t), p2(h)), ch(n(n1(e), n2(ờ)), pc(p4(n), p5(g))))$.		x
.....		

Từ Bảng 6 chúng tôi nhận thấy rằng, kết quả truy vấn DSS có rất nhiều cây cấu trúc không có giá trị về nghĩa của âm tiết hoặc thành phần âm tiết trong âm tiết, Mặt khác, khi kho dữ liệu lớn thì số âm tiết sinh ra bởi các luật sinh sẽ rất lớn, dẫn đến bùng nổ tổ hợp. Để khắc phục điều này, DSS thực hiện ràng buộc thêm biến nghĩa cho từng cụm thành phần, thành phần phụ âm và nguyên âm trong âm tiết. Chúng tôi mô tả chi tiết ở phần sau.

C. Ràng buộc biến ngữ nghĩa cho DSS

Việc thực hiện ràng buộc ngữ nghĩa cho DSS gọi là (SDSS- Semantic Defined Syllable Struc-

ture) bằng việc thêm vào các biến trong cấu trúc mô tả cụm thành phần và các ràng buộc trong các luật cấu trúc. Khi đó, quá trình phân tích cấu trúc âm tiết sẽ là sự kết hợp giữa luật cấu trúc âm tiết và những ràng buộc ngữ nghĩa. SDSS có hai đặc điểm chính:

- SDSS sử dụng cấu trúc âm tiết và thêm vào các biến để biểu diễn các thông tin về âm tiết. Cấu trúc này thường được mô tả dưới dạng quy tắc và thêm vào các biến trên từng quy tắc, nhằm mô tả các đặc tính cụ thể của thành phần như các thông tin cấu trúc âm tiết và ngữ nghĩa.

- SDSS tích hợp các ràng buộc về cấu trúc âm tiết và ngữ nghĩa vào tập luật. Các ràng buộc này được dùng để kiểm soát các quan hệ cấu trúc âm tiết và ngữ nghĩa giữa các thành phần trong âm tiết.

Để giải quyết vấn đề này, ta cần dựa trên hai vấn đề, thứ nhất là sự kết hợp giữa các thành phần sao cho thích hợp về mặt nghĩa, thứ hai là xét trên từng quy tắc, các biến được áp dụng riêng từng quy tắc.

SDSS sử dụng biến nghĩa cho từng thành phần riêng biệt, điều này đảm bảo tất cả âm tiết đều đúng về nghĩa. SDSS đơn giản để biểu diễn cấu trúc thành phần. SDSS tìm sự gắn kết của phụ âm hoặc nguyên âm là các thành phần kết hợp trên từng thành phần của từng quy tắc. Âm tiết trong ví dụ sau cho thấy cấu trúc này tập trung vào các quy tắc kết hợp cấu trúc âm tiết của nguyên âm và phụ âm. Vì đối với thành phần giữa, nguyên âm là thành phần quan trọng trong âm tiết, có tác dụng gắn kết các thành phần khác trong âm tiết. SDSS ràng buộc biến nghĩa cho phụ âm, nguyên âm hay cụm thành phần của âm tiết “trường” được biểu diễn như sau:

Trong đó:

Quy tắc (1): Biến X ràng buộc thành phần phụ âm đầu Pd được kết hợp với thành phần chính CH của âm tiết.

Quy tắc (2): Biến X ràng buộc thông tin về nghĩa của phụ âm đầu hoặc cụm thành phần đầu trong thành phần phụ âm đầu Pd. Biến X1 ràng buộc phụ âm thứ hai P2 được kết hợp với phụ âm thứ nhất P1 trong cụm thành phần phụ âm đầu.

Quy tắc (3): Biến X ràng buộc cụm thành phần

$a(a(Pd, CH)) \rightarrow pd(X, Pd), ch(X, CH).$	(1)
$pd(X, pd(P1, P2)) \rightarrow p1(X, X1, P1),$ $p2(X1, P2).$	(2)
$ch(X, ch(N, Pc)) \rightarrow n(X, Z, N), pc(Z, Pc).$	(3)
$n(X, Z, n(N1, N2)) \rightarrow$ $n1(X, Z, Y1, N1), n2(Y1, N2).$	(4)
$pc(Z, pc(P4, P5)) \rightarrow$ $p4(Z, Z1, P4), p5(Z1, P5).$	(5)
$p1(tr, X1, p1(t)) \rightarrow [t], \{X1=r\}.$	(6)
$p2(r, p2(r)) \rightarrow [r].$	(7)
$p4(ng, Z1, p4(n)) \rightarrow [n], \{Z1=g\}.$	(8)
$p5(g, p5(g)) \rightarrow [g].$	(9)
$n1(X, Z, Y1, n1(u)) \rightarrow$ $>[u], \{X=tr, Z=ng, Y1=o\}.$	(10)
$n2(o, n2(o)) \rightarrow [o].$	(11)

Hình 3: Minh họa cho cấu trúc SDSS

phụ âm đầu Pd được kết hợp với thành phần chính CH. Biến Z ràng buộc cụm phụ âm cuối Pc với thành phần nguyên âm N trong thành phần chính CH.

Quy tắc (4): Biến X ràng buộc cụm thành phần phụ âm đầu Pd được kết hợp với thành phần chính CH, được truyền từ quy tắc (2). Biến Z ràng buộc cụm phụ âm cuối Pc với thành phần nguyên âm N trong thành phần chính CH, được truyền từ quy tắc (5). Biến Y1 ràng buộc nguyên âm thứ hai N2 được kết hợp với nguyên âm thứ nhất N1 trong cụm thành phần nguyên âm.

Quy tắc (5): Biến Z ràng buộc thông tin về nghĩa của phụ âm cuối hoặc cụm thành phần cuối trong thành phần phụ âm cuối Pc, được truyền từ quy tắc (8). Biến Z1 ràng buộc phụ âm thứ năm P5 được kết hợp với phụ âm thứ tư P4 trong cụm thành phần phụ âm cuối.

Quy tắc (6): Nhân nghĩa “tr” cho biết thông tin về nghĩa của cụm thành phần phụ âm đầu Pd. Biến X1 ràng buộc phụ âm thứ hai P2 trong cụm phụ âm đầu Pd với $X1 = r$ được truyền từ quy tắc (7), phụ âm thứ nhất P1 chỉ phù hợp với phụ âm thứ hai P2 là “r”.

Quy tắc (7): Nhân nghĩa phụ âm thứ hai P2 là “r” cho biết thông tin về nghĩa của từ phụ âm có

nhân “r”.

Quy tắc (8): Nhân nghĩa “ng” cho biết thông tin về nghĩa của cụm thành phần phụ âm cuối Pc. Biến Z1 ràng buộc phụ âm thứ năm P5 trong cụm phụ âm cuối Pc với $Z1 = g$ được truyền từ quy tắc (9), phụ âm thứ tư P4 chỉ phù hợp với phụ âm thứ hai P5 là “g”.

Quy tắc (9): Nhân nghĩa phụ âm thứ năm P5 là “g” cho biết thông tin về nghĩa của từ phụ âm có nhân “g”.

Quy tắc (10): Biến $X=tr$ được truyền từ quy tắc (2), quy tắc (2) được truyền từ quy tắc (6). Biến $Z=ng$ được truyền từ quy tắc (5), quy tắc (5) được truyền từ quy tắc (8). Biến Y1 ràng buộc nguyên âm thứ hai N2 trong cụm nguyên âm N với $Y1 = “o”$ được truyền từ quy tắc (11), nguyên âm thứ nhất N1 chỉ phù hợp với nguyên âm thứ hai N2 là “o”.

Quy tắc (11): Nhân nghĩa nguyên âm thứ hai N2 là “o” cho biết thông tin về nghĩa của từ nguyên âm có nhân “o”. Khi đó, phân tích một chuỗi a trên cấu trúc SDSS có ràng buộc biến nghĩa cho âm tiết được thực hiện bởi câu truy vấn:

?- a(T,L,[]).

Truy vấn một cách tổng quát hoặc truy vấn từng âm tiết. Số lượng cây cấu trúc được trả về duy nhất cây đúng về cấu trúc và nghĩa của phụ âm, nguyên âm và cụm thành phần trong âm tiết.

Ví dụ: truy vấn lần lượt các âm tiết có ba dạng chữ khác nhau như: “trường”, “Trường”, “TRƯỜNG”.

?- a(T,[t,r,ư,ồ,n,g],[]).

$T = a(pd(p1(t), p2(r)), ch(n(n1(ư), n2(ồ)), pc(p4(n), p5(g))))).$

?- a(T,['T',r,ư,ồ,n,g],[]).

$T = a(pd(p1('T'), p2(r)), ch(n(n1(ư), n2(ồ)), pc(p4(n), p5(g))))).$

?- a(T,['T', 'R', 'Ư', 'Ồ', 'N', 'G'],[]).

$T = a(pd(p1('T'), p2('R')), ch(n(n1('Ư'), n2('Ồ')), pc(p4('N'), p5('G')))).$

Nhận xét: Khi truy vấn SDSS, số cây trả về đúng chính xác về cấu trúc và nghĩa của phụ âm, nguyên âm và cụm thành phần trong âm tiết, giải quyết được sự bùng nổ tổ hợp và loại bỏ những cây cấu trúc trả về không có giá trị về nghĩa trong âm tiết.

III. MÔ HÌNH HỆ THỐNG PHÂN TÍCH CẤU TRÚC ÂM TIẾT ỨNG DỤNG SDSS

Hình 4: Mô hình xây dựng hệ thống phân tích cấu trúc âm tiết tiếng Việt với SDSS

Bước 1. Thu thập ngữ liệu huấn luyện. Ngữ liệu huấn luyện đã trình bày ở phần II và lựa chọn các dạng âm tiết như “tất cả các kí tự là chữ thường”, “tất cả các kí tự là chữ hoa” và “kí tự đầu âm tiết là chữ hoa, các kí tự còn lại là chữ thường”. Tập hợp tất cả âm tiết đã thu thập được gọi là tập ngữ liệu huấn luyện. Giả sử, tập ngữ liệu huấn luyện có kích cỡ n âm tiết huấn luyện.

Bước 2. Phân tích âm tiết huấn luyện và đưa về định dạng của cây cấu trúc: Sau khi thu thập đủ số lượng âm tiết huấn luyện trên một lĩnh vực cụ thể, bắt đầu phân tích cấu trúc âm tiết cơ bản và gán nhãn cho từng cụm thành phần, phụ âm, nguyên âm của các âm tiết trong tập ngữ liệu huấn luyện.

Bước 3. Cấu trúc SDSS được rút ra từ cây cấu trúc âm tiết: Xây dựng một công cụ để chuyển cây cấu trúc âm tiết sang các quy tắc SDSS.

Bước 4. Xây dựng tập quy tắc SDSS Sau khi có quy tắc SDSS trong tập ngữ liệu huấn luyện, ta tiến hành gom các quy tắc giống nhau nhằm giảm thời gian thực hiện quay lui để tạo thành tập quy tắc SDSS.

Bước 5. Hoàn thiện hệ thống phân tích cấu trúc âm tiết.

Đối với cấu trúc SDSS, không cần phải xây dựng thuật toán phân tích cấu trúc vì SDSS sử dụng cơ chế danh sách hiệu có sẵn trong Prolog. SDSS chỉ việc xây dựng các vị từ như sau: công

việc là tìm tất cả các cây cấu trúc thành phần ứng cử của một âm tiết.

Trường hợp âm tiết không phân tích được (mẫu âm tiết chưa có trong tập ngữ liệu huấn luyện), hệ thống cập nhật vào tập ngữ liệu huấn luyện, tiến hành thiết kế cây cấu trúc âm tiết cơ bản.

IV. CHẠY THỬ NGHIỆM VÀ ĐÁNH GIÁ GIẢI PHÁP

Dưới đây là một số mẫu câu thử nghiệm, sắp xếp theo thứ tự từ đơn giản đến phức tạp dần.

Bảng 7: Một số kết quả thử nghiệm phân tích cấu trúc âm tiết

STT	Âm tiết	Kết quả	
		DSS	SDSS
1	bộ	1	1
2	Môn	1	1
3	công	1	1
4	nghệ	1	1
5	thông	1	1
6	tin	1	1
7	NAM	1	1
8	Phol	0	0
9	Trường	1	1

Kí hiệu kết quả của DSS và SDSS trong Bảng 7. Kết quả trả về 0 là không có giá trị, 1 là có giá trị.

Bảng 7 trên là kết quả thử nghiệm với tập test. DSS và SDSS đều trả về kết quả 0 của âm tiết thứ 8, vì bộ luật của DSS và SDSS không có cấu trúc âm tiết dạng này. Đây là âm tiết vay mượn, tập luật có thể bổ sung trong tương lai. Âm tiết thứ 10 là âm tiết “không có giá trị về nghĩa” của cụm thành phần trong âm tiết, nhưng DSS nhận diện có giá trị do cụm thành phần phụ âm đầu Pd gồm P1 và P2. Do các luật sinh (Pd->P1, P2) nên DSS không quan tâm về nghĩa của cụm thành phần, nhưng SDSS phát hiện âm tiết “Trường” sai về nghĩa.

Qua việc thử nghiệm với số lượng âm tiết chuẩn, chúng tôi đã tiến hành so sánh các pha phân tích cấu trúc: cơ sở DSS, phân tích cơ sở có ràng buộc biên nghĩa SDSS. Dưới đây là các kết quả so sánh:

Bảng 8: Bảng so sánh giữa các pha phân tích

STT	Mẫu âm tiết	DSS		SDSS	
		Kết quả trả về	Thời gian	Kết quả trả về	Thời gian
1	Kgoa	1	0.0019	0	0.0003
2	KỠ	1	0.0008	1	0.0002
3	Thuật	1	0.0030	0	0.0007
4	Và	1	0.0003	1	0.0001
5	CÔNG	1	0.0031	0	0.0007
6	Nghệ	1	0.0014	1	0.0007
7	Đại	1	0.0018	1	0.0002
8	Họt	1	0.0034	0	0.0002
9	Trà	1	0.0234	1	0.0002

Nhận xét: Qua Bảng 8, chúng ta có thể thấy được hiệu quả của việc giải quyết loại bỏ phân tích âm tiết không có giá trị nhờ SDSS.

Kết quả phân tích cho thấy, đối với SDSS, sự loại bỏ phân tích âm tiết không có giá trị đã được giải quyết. Bộ phận tích cho kết quả tương đối khả quan.

Tốc độ xử lý nhanh hơn rất nhiều so với các pha phân tích cơ sở.

Các pha phân tích cơ sở, hầu như chỉ làm việc được với các âm tiết có số lượng thành phần cùng loại ký tự in hoa hoặc ký tự thường và sự phân tích âm tiết không có giá trị là tương đối lớn.

Chúng tôi so sánh hiệu năng của các pha phân tích: phân tích cơ sở với DSS, ràng buộc biến nghĩa với SDSS.

Bảng 9: So sánh hiệu năng các pha phân tích cấu trúc âm tiết

Các tiêu chí so sánh	Pha phân tích cấu trúc âm tiết	
	DSS	SDSS
Độ chính xác	90.8%	99.6%
Tốc độ trung bình cho một âm tiết	0.00405s	0.00036s

Chúng ta có thể thấy, qua Bảng 9, pha phân tích cấu trúc sau khi áp dụng mô hình SDSS cho việc loại bỏ phân tích âm tiết không có giá trị thì số cây cấu trúc sinh ra giảm đi, do ràng buộc biến nghĩa làm hạn chế luật sinh. Nhưng xét tốc độ phân tích, việc kết hợp SDSS đã cho hiệu năng rất cao, tốc độ nhanh hơn rất nhiều và hoàn toàn không còn trường hợp phân tích âm tiết không có giá trị.

V. KẾT LUẬN

Trong nghiên cứu này, trình bày được kết quả nghiên cứu:

Áp dụng mô hình cấu trúc âm tiết xác định có ràng buộc biến ngữ nghĩa SDSS đầu tiên trên tiếng Việt.

Kỹ thuật ràng buộc đối số đã loại bỏ đi những hướng phân tích âm tiết không có giá trị về nghĩa và làm tăng hiệu suất hệ thống phân tích cấu trúc âm tiết. Bên cạnh đó, các kỹ thuật xử lý cấu trúc thành phần làm giảm thời gian chạy của hệ thống một cách đáng kể.

Đưa ra mô hình biểu diễn âm tiết theo cấu trúc SDSS. Mô hình này tập trung vào việc giải quyết bùng nổ tổ hợp. Đồng thời, mô hình này cũng cho phép mô tả mối quan hệ ràng buộc giữa các thành phần khác và loại bỏ phương án phân tích âm tiết không có giá trị.

Xây dựng mô hình biểu diễn luật chứa các ràng buộc cấu trúc âm tiết và ngữ nghĩa. Mô hình này dựa trên sự mở rộng của luật trong DSS cơ sở, bổ sung thành phần biến nghĩa ràng buộc các thành phần khác trong âm tiết phù hợp về nghĩa của cụm thành phần. Việc đưa ra tập luật của SDSS cho phép bao phủ ràng buộc giữa các thành phần dựa trên thông tin ngữ nghĩa.

Pha phân tích cấu trúc âm tiết của SDSS đạt độ chính xác cao hơn so với nghiên cứu Đinh Thị Phương Thu và cộng sự [7], vì không có cùng chung dữ liệu nguồn, như công bố của tác giả là 94%. Khi đó, SDSS đã đạt được độ chính xác 99.6%.

Trong thời gian tới, SDSS cần bổ sung các cấu trúc âm tiết mới vào tập luật để nâng cao khả năng phân tích. Tập luật mới cần phân tích được các loại âm tiết đa dạng hơn như âm tiết vay mượn tiếng nước ngoài hay từ viết tắt.

TÀI LIỆU THAM KHẢO

- [1] Đỗ Bá Lâm, Lê Thanh Hương, "Implementing a Vietnamese syntactic parser using HPSG", International Conference on Asian Language Processing (IALP), 2008.
- [2] Dang Tuan Nguyen, Kiet Van Nguyen and Tin Trung Pham, "Implementing A Subcategorized Probabilistic Definite Clause Grammar for Vietnamese Sentence Parsing", International Journal on Natural Language Computing (IJNLC), Vol. 2, No. 4, August 2013,

pp. 1-19. ISSN : 2278 - 1307 [Online]; 2319 - 4111 [Print].

- [3] Michael Collins, “*Probabilistic Context – Free Grammars(PCFGs)*”, Fall 2011, Note on PCFGs.
- [4] Christopher D. Manning, Hinrich schüze, “*Foundations of Statistical Natural Language Processing*”, MIT Press, 1999.
- [5] Michael Collins, “*Probabilistic Context – Free Grammars(PCFGs)*”, COMS W4705, Natural Language Processing, Fall 2011.
- [6] Qaiser Abbas, Nayyara Karamat, Sadia Niazi, “*Development of Tree-bank Based Probabilistic Grammar for Urdu Language*”, International Journal of Electrical & Computer Sciences (IJECS), Vol. 9, No. 9, 2009.
- [7] Đinh Thị Phương Thu, Huỳnh Quyết Thắng, Nguyễn Văn Lợi, “*Sử dụng luật cấu tạo âm tiết tiếng Việt trong hai thành phần trong bài toán kiểm tra chính tả tiếng Việt*”, P.83-92. Tạp chí BCVT & CNTT kỳ 3 10/2007.
- [8] Nguyễn Tài Cẩn, “*Ngữ pháp tiếng Việt, Tiếng - Từ ghép - Đoản ngữ*”, Nhà Xuất bản Khoa học Xã hội, 1975.
- [9] Hoàng Trọng Phiến, “*Ngữ pháp tiếng Việt - Câu*”, Nhà Xuất bản ĐHQG Hà Nội, 2008.
- [10] Nguyễn Cao Đàm, “*Ngữ pháp tiếng Việt*” (câu đơn hai thành phần), Nhà Xuất bản ĐHQG Hà Nội, 2008.
- [11] Nguyễn Tuấn Đăng, Nguyễn Trọng Chính, “*Các kỹ thuật trong xử lý ngôn ngữ tự nhiên*”, Nhà Xuất bản Đại học Quốc gia TP. Hồ Chí Minh, 2017.