**Đại Học Quốc Gia tp HCM – Đại Học CNTT**



Đồ án môn học CS221.L11.KHCL

**TÌM HIỂU VÀ XÂY DỰNG BỘ PHÂN TÍCH CẤU TRÚC NGỮ ĐOẠN**

*Môn học: Xữ lí ngôn ngữ tự nhiên*

*Sinh viên: 18520811 - Trần Kim Hưng*

*18520766 - Trần Thái Hòa*

*18520915 – Hồ Sỹ Đăng Khoa*

*Giảng viên hướng dẫn: T.S. Nguyễn Trọng Chỉnh*

# MỤC LỤC

[MỤC LỤC 3](#_Toc60619825)

[TÓM TẮT 5](#_Toc60619826)

[I. Giới thiệu 6](#_Toc60619827)

[II. Bộ phân tích cú pháp 10](#_Toc60619829)

[III. Thiết kế bộ ngữ liệu 13](#_Toc60619830)

[IV. Thực nghiệm 14](#_Toc60619831)

[V. Kết quả đánh giá 15](#_Toc60619832)

[VI. Kết luận 17](#_Toc60619833)

[THAM KHẢO 18](#_Toc60619834)

# 

# TÓM TẮT

Phân tích cú pháp ngôn ngữ tự nhiên là quá ttrình biến biến đổi những chuỗi các thành phần (tokens), trong xử lí ngôn ngữ tự nhiên thường là các từ thuộc về một ngôn ngữ, thành một cấu trúc biểu diễn các thông tin về mặt cú pháp, đôi khi là một cây phân tích cú pháp (parse tree) hoặc một dồ thị phụ thuộc (dependence graph). Có rất nhiều loại phân tích cú pháp, bao gồm phân tích cú pháp nông (hay còn gọi là chunking), phân tích cú pháp sâu (deep parsing), phân tích cú pháp phụ thuộc (dependence parsing), phân tích cú pháp ngữ đoạn (constituent parsing), phân tích cú pháp ngữ nghĩa (sematic parsing), phân tích cú pháp diễn ngôn (discourse parsing) [1].

Trong xữ lí ngôn ngữ tự nhiên, phân tích cú pháp không phải đích đến cuối cùng nhưng là một bước có thể được sử dụng cho mục đích đến với đích đến. Người bình thường ít khi nào quan tâm đến việc áp dụng thực hiện quá trình phân tích cú pháp cho những kết quả đầu ra của một ứng dụng xử lí ngôn ngữ tự nhiên. Tuy nhiên, những học viên đi theo hướng xử lí ngôn ngữ tự nhiên có thể bổ xung phân tích cú pháp vào quá trình xử lí. Suy luận ra các thông tin về cú pháp có thể cường hoá độ chính xác trong quá trình phân tích văn bản [1].

Trong đồ án này, chúng tôi sẽ xây dựng một bộ phân tích cú pháp với giả thuật CKY và so sánh độ chính xác so với thư viện nltk.

# I. Giới thiệu

Cú pháp ngữ đoạn được xây dựng dựa trên ý tưởng là các nhóm từ có chức năng như một đơn vị, hoặc một ngữ đoạn [2] (Hình. 1.1).

Diagram

Description automatically generated

*Hình.1.1. Minh hoạ cho cấu trúc ngữ đoạn [3].*

Phát triển văn phạn bao gồm việc xây dựng một không gian chứa các ngữ đoạn trong một ngôn ngữ nhất định. Ngôn ngữ được nhóm vào nhau theo quy tắc luôn vòng quanh một cụm danh từ (noun phrase) [2]. Một số cụm danh từ trong tiếng Anh như:

Text

Description automatically generated

*Hình. 1.2. Ví dụ về các cụm danh từ trong ngôn ngữ tiếng Anh [2].*

Bằng chứng cho thấy các cụm danh từ có thể được nhóm vào là chúng đề có thể được thể hiện trong cùng một mô trường cú pháp (syntatic enviroment) ví dụ như đứng trước động từ [2] (Hình. 1.3).

Text

Description automatically generated

*Hình. 1.3. Ví dụ về việc các cụm danh từ ở hình.1.2. có thể được đặt trước động từ [2].*

Một loại bằng chứng khác cho thấy được sự thể hiện của ngữ đoạn đến từ hai khái niệm về cấu trúc tiền tố (*preposed*) và hậu tố (*postposed*). Ví dụ như giới từ chỉ ngày 17 tháng 12 (*September seventeenth*) có thể đặc trong nhiều vị trính khác nhau trong một câu trong tiếng Anh và chúng vẫn có cùng ý nghĩa với nhau như [2]:

On *September seventeenth*, I’d like to fly from Atlanta to Denver

I’d like to fly on *September seventeenth* from Atlanta to Denver

I’d like to fly from Atlanta to Denver on *September seventeenth*

**1.1. Văn phạm phi ngữ cảnh**

Một văn phạm phi ngữ cảnh (Context-Free-Grammar) chứa một bộ các tập luật hoặc những thành phần, mỗi phân đoạn biểu diễn các hướng mà một chuỗi các từ có thể được nhóm lại với nhau [2] (Hình1. 4).

A picture containing text

Description automatically generated

*Hình. 1.4. Ví dụ về văn phạm phi ngữ cảnh.*

**1.2. Sự nhập nhằng**

Sự nhập nhằng về cấu trúc sảy ra khi văn phạm có thể gán cho ngữ đoạn nhiều hơn một danh tính trong một câu. Câu nói nổi tiếng của Groucho Marx thủ vai thuyền trưởng Spaulding trong Animal Cracker (Hình. 1.5) được là một trường biểu diễn cho sự nhập nhằng trong cấu trúc. Với từ in my

Diagram

Description automatically generated

*Hình. 1.5.Câu nói của thuyền trưởng Groucho trong bộ Animal Cracker khi phân tích thành cấu trúc ngữ đoạn [2]*

# pajamas đều có thể là một phần của một cụm danh từ (NP) [2].

Sự nhập nhằng về cấu trúc được thể hiện với nhiều hình dạng. Hai sự nhập nhằng thường gặp nhất là **nhập nhằng trong liên kết (attachment ambiguity)** và **nhập nhằng về vị trí (coordination ambiguity)** [2].

Một câu thể hiện sự **nhập nhằng về liên kết (attachment ambiguity)** nếu một ngữ đoạn nhất định có thể được liên kết vào một một cây phân tích cú pháp nhiều hơn một vị trí. Trong ví dụ về câu nói của Groucho Marx, cụm từ *in my pajamas* là một ví dụ về sự nhập nhằng trong liên kết của nhãn PP [2].

**Sự nhập nhằng về vị trí (coordination ambiguity)** thể hiện khi một cụm câu có thể được nối tiếp nhau bởi một từ liên kết như từ *and*. Ví dụ như trong câu old men and women có thể được phân tích theo hướng (old (men and women)) hoặc có thể được phân tích thành (old men) and (women) [2].

# II. Bộ phân tích cú pháp

Trong đồ án về xây dựng bộ phân tích cú pháp ngữ đoạn, chúng tôi đã cài đặt thuật toán CKY.

**Quy hoạch động (Dynamic programming)** cung cấp một hướng giải quyết hiệu quả cho vấn đề nhập nhằng. Trong hướng tiếp cận quy hoạch động, bằng cách lưu lại các kết quả của các bài toán con, ta có thể quay lại và giải quyết tiếp những bài toán con. Trong bộ phân tích cú pháp, các bài toán con được thể hiện dưới dạng những cây văn phạm con được sinh ra trong quá trình phân tích [2].

Hướng tiếp cận quy hoạch động được sinh ra từ bản chất phi ngữ cảnh vốn có của quy tắc văn phạm. Khi một ngữ đoạn được phát hiện ra trong một phân đoạn của dữ liệu đầu vào. Sẽ được lưu trữ và luôn sẳn sàng cho bất kì giai đoạn mở rộng cần đến nó [2]. Từ các bước tính toán trên, bộ phân tích cú pháp có thể tìm ra các cây phân tích có thể có.

**2.1.Văn phạm CNF**

Để giải thuật CKY hoạt động, văn phạm trước hết cần phải được chuẩn hoá thành dạng văn phạm theo quy tắc Chomsky Normal From (CNF). Quy luật của CNF phải giới hạn trong khoảng A → B C hoặc A → w [2]. Các luật có dạng A → B C … sẽ được xử lí theo quy tắc sau:

A → B C γ

Luật trên sẽ được chuyển đổi thành:

A → X1 γ

X1 → B C

A picture containing text

Description automatically generated

*Hình. 2.1. Văn phạm khi được chuẩn hoá thành dạng CNF[2].*

**2.2. Bộ phân tích CKY**

Thuật toán CKY sẽ được biểu diễn dưới dạng mã giã như sau:

Text

Description automatically generated

*Hình. 2.2. Giải thuật CKY [2]*

Với chuỗi văn bản có chiều dài n thuật giải CKY sẽ xây dựng một ma trận tam giác vuôn cận trên với kích thước (n + 1) x (n + 1). Mỗi ô thứ [i j] trong ma trận thể hiện một ngữ đoạn [2] (Hình. 2.2).

A picture containing text, clock

Description automatically generated

*Hình. 2.3. Kết quả một văn phạm được phân tích [2]*

# III. Thiết kế bộ ngữ liệu

Bộ ngữ liệu mà nhóm thiết kế gồm 60 câu đã được phân tích cấu trúc ngữ đoạn một cách thủ công. Các phân tích ngữ đoạn trong câu mà nhóm chuẩn bị theo một quy luật như sau:

(TOP (S (NP <Một dãy các quy luật dẫn đến NP>) (VP <Một dãy các quy luật dẫn đếnVP>)))

Mục tiêu của việc thiết kế như thế này là để khi phân tích ra một cây văn phạm từ tập huấn luyện gồm 60 câu thì các quy luật ở đầu sẽ luôn là:

TOP -> S

S -> NP VP

Với thiết kế như trên, việc tạo ra tập GOLD và tập test có thể thực hiện mộ cách dễ dàng.

# IV. Thực nghiệm

Trước khi thực nghiệm, các bộ ngữ liệu ở tập train sẽ đi qua một bộ phân tích để lấy ra các quy tắc của các cây phân tích cú pháp có trong tập train và sinh ra một bộ văn phạm chứa các quy tắc phân tích.

Sau khi có được bộ văn phạm từ bộ phân tích, nhóm đưa văn phạm vào hai bộ phân tích cú pháp. Bộ phân tích cú pháp CKY do nhóm cài đặt một cách thủ công và sau đó đưa vào một bộ phân tích có sẳn trong thư viện nltk của python là EarleyChartParser.

Cuối cùng, bộ phân tích sẽ đưa các tập test chứa các văn phản chưa được phân tích cú pháp ngữ đoạn vào và nhiệm vụ của hai chương trình sẽ dự đoán cú pháp ngữ đoạn cho các câu thử nghiệm.

Các thực nghiệm chạy trên hệ thống có nền tảng như Hình. 4.1.

Text

Description automatically generated

Hình. 4.1. Nền tảng thực nghiệm của đồ án

# V. Kết quả đánh giá

Sau khi có kết quả phân tích cú pháp từ các bộ phân tích cú pháp, nhóm tiến hành đánh giá giữa hai bộ phân tích.

**5.1. Độ đo đánh giá.**

Nhóm sử dụng các độ đo dánh giá recall, precision và F1 để dánh giá.

Ngữ đoạn trong một tập giả thuyết của một câu s được dán nhãn là đúng khi có một ngữ đoạn trong tập có cùnbg điểm xuất phát, điểm kết thúc và các ký tự không phải thuộc tập cuối (non-terminal symbol) [2]. Các công thức được thể hiện như sau:

Trong lúc thực hiện, nhóm sẽ sử dụng công cụ [3] đã cài đặt dựa trên các lý thuyết trên và đầu ra của công cụ sẽ là các độ đo Recall, Precision và F1.

**5.2. Kết quả đánh giá**

Tập thử nghiệm sẽ gồm 4 câu chưa được phân tích ngữ đoạn và sau khi đưa vào bộ phân tích ngữ đoạn cho ra các dữ liệu sau khi được phân tích ngữ đoạn. Sau khi có kết quả phân tích ngữ đoạn, nhóm sẽ dùng phần mềm [3] để cho các độ đo dánh giá.

**Với CKY:**

**A picture containing text, monitor, black, screen

Description automatically generated**

*Hình.5.1 Kết quả đánh gía CKY trên tập test.*

**Với EarletChartParser:**

**A picture containing calendar

Description automatically generated**

*Hình. 5.2. Kết quả đánh giá của EarletChartParser trên tập test*

# VI. Kết luận

Qua quá trình xây dựng bộ phân tích cú pháp cấu trúc ngữ đoạn, nhóm nhận thấy được để phân tích cú pháp ngữ đoạn cần phải sở hữu một tập luật có thể bao phủ tất cả trường hợp mà một để dẫn đến một phân đoạn trong văn phạm. Từ việc phân tích cú pháp mà nhóm cũng nhận thấy được sự hạn chế trong chương trình mà nhóm cài đặt là chúng không có khả năng phân tích được ra thành cây phân tích cú pháp. Từ đó nhóm có suy nghĩa về việc áp dụng một bộ gán nhãn từ loại vào hỗ trợ cho việc phân tích cú pháp, tuy trong đồ án này, nhóm vẫn chưa hiện thực được việc áp dụng ý tưởng đưa một bộ phân tích cú pháp vào nhưng chúng tôi vẫn mong đây là một hướng suy nghĩ đúng khi nhắc đến vấn đề này.

# THAM KHẢO

[1] Constituent Parsing By Classìication, Joseph Turian, New York University, 2007.

[2] Speech and Language Processing 3rd edition, Dan Jurafsky.

[3] EVALB: https://nlp.cs.nyu.edu/evalb/