

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ TRI THỨC**

NGUYỄN THỐNG NHẤT – LÊ MINH SƠN

**GÁN NHÃN PHÂN TÍCH CÚ PHÁP QUAN HỆ
CHO SONG NGỮ ANH VIỆT
THÔNG QUA LIÊN KẾT NGỮ**

LUẬN VĂN CỬ NHÂN TIN HỌC

TP. Hồ Chí Minh – Năm 2003

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ TRI THỨC

NGUYỄN THỐNG NHẤT – 9912053
LÊ MINH SƠN - 9912668

GÁN NHÃN PHÂN TÍCH CÚ PHÁP QUAN HỆ
CHO SONG NGỮ ANH VIỆT
THÔNG QUA LIÊN KẾT NGỮ

LUẬN VĂN CỬ NHÂN TIN HỌC

GIÁO VIÊN HƯỚNG DẪN
GS.TSKH. HOÀNG KIỂM

NIÊN KHOÁ 1999 - 2003

Lời cảm ơn

Trước hết, chúng tôi xin chân thành gửi lời cảm ơn đến GS.TSKH. Hoàng Kiếm, người đã tận tụy dẫn dắt chúng tôi từng bước để hoàn thành bài luận văn này. Chúng tôi cũng chân thành cảm ơn các Thầy Cô trong và ngoài khoa Công nghệ thông tin đã truyền đạt kiến thức quý báu cho tôi trong suốt bốn năm học.

Để hoàn thành bài luận văn này, chúng tôi không thể không nhắc đến sự động viên và chăm sóc của gia đình. Ngoài ra, chúng tôi gửi lời cảm ơn đến những người mà chúng tôi đã có dịp cộng tác và sự ủng hộ tinh thần của bạn bè.

Cuối cùng chúng tôi cũng muốn gửi lời cảm ơn đến Thầy Đinh Điền và các thành viên trong nhóm VCL, những người đã giúp đỡ cho chúng tôi hoàn tất bài luận văn này.

Chúng tôi xin chân thành cảm ơn tất cả.

TP. Hồ Chí Minh, 7-2003

Nguyễn Thống Nhất và Lê Minh Sơn

Nhận xét của giáo viên hướng dẫn

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày thángnăm 2003

Giáo viên hướng dẫn

GS. TSKH. Hoàng Kiếm

Nhận xét của giáo viên phản biện

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày thángnăm 2003

Giáo viên phản biện

MỤC LỤC

LỜI NÓI ĐẦU	1
Chương 1: TỔNG QUAN	1
1.1. Phân tích cú pháp quan hệ.....	1
1.2. Liên kết từ/ngữ.....	1
1.3. Chiều quan hệ cú pháp	3
1.3.1. Chiều nhãn từ loại.....	3
1.3.2. Chiều quan hệ cú pháp.....	4
Chương 2: CÁC CÁCH TIẾP CẬN	5
2.1. Phân tích cú pháp.....	5
2.1.1. Các phương pháp tiếp cận dùng luật phi ngữ cảnh (CFG).....	5
2.1.1.1. Cách tiếp cận từ trên xuống (Top-Down)	5
2.1.1.2. Thuật toán phân tích cú pháp từ trên xuống (Top-Down)	7
2.1.1.3. Cách tiếp cận Từ dưới lên (Bottom-Up)	8
2.1.1.4. Thuật toán phân tích cú pháp Earley	11
2.1.1.5. Mạng ngữ pháp lan truyền	12
2.1.2. Phương pháp TBL (Transformation-Based Error-Driven Learning) ..	15
2.1.3. Phương pháp phân tích cú pháp dựa trên văn phạm TAG	19
2.1.3.1. Văn phạm TAGs.....	19
2.1.3.1.1. Cây sơ cấp.....	19
2.1.3.1.2. Cây phụ trợ	19
2.1.3.2. Các tác tổ trong TAGs.....	20
2.1.3.2.1. Tác tổ thêm vào	20

2.1.3.2.2.	Tác tổ thay thế:.....	21
2.1.3.3.	Những điều kiện kết hợp trên cây	21
2.1.3.4.	Cây rỗng.....	21
2.1.4.	Phương pháp phân tích cú pháp dựa trên nguyên tắc	22
2.1.4.1.1.	Thuyết X-Bar (\bar{X})	23
2.1.4.1.2.	Nguyên lý Theta.....	23
2.1.4.1.3.	Thuyết lọc vai (Case-filter)	23
2.1.4.1.4.	Thuyết kết hợp.....	23
2.1.4.1.5.	Thuyết về tính cục bộ và trường rỗng.....	23
2.1.4.1.6.	Thuyết dịch chuyển.....	24
2.2.	Các cách tiếp cận trong việc liên kết từ/ngữ.....	24
2.2.1.	Char-Align – Hệ thống Termight.....	26
2.2.2.	Phương pháp K-vec	28
2.2.3.	Phương pháp DK-vec	29
2.2.4.	Ánh xạ song ngữ với SIMR.....	30
2.2.5.	Mô hình xác suất với thuật toán IPFP.....	30
2.2.6.	Mô hình dựa vào sự phân lớp (Class-based).....	33
2.2.7.	Mô hình liên kết dựa vào cách tiếp cận dịch máy thống kê (SMT).....	33
2.3.	Các phương pháp chiếu.....	34
2.3.1.	Chiếu nhãn từ loại.....	34
2.3.1.1.	Phương pháp trực tiếp.....	34
2.3.1.2.	Phương pháp Noise-robust.....	34
2.3.1.3.	Phương pháp sử dụng luật tương tác.....	35
2.3.2.	Chiếu quan hệ.....	35
2.3.2.1.	Mô hình xác suất	35
2.3.2.2.	Phương pháp DCA (Direct Correspondence Assumption).....	35
2.3.2.3.	Các phương pháp khác	36

Chương 3: MÔ HÌNH THUẬT TOÁN.....	37
3.1. Phân tích cú pháp dựa trên nguyên tắc.....	37
3.1.1. Khái quát	37
3.1.2. Ý tưởng cơ bản của phương pháp phân tích dựa trên nguyên tắc.....	39
3.1.3. Một số ít những nguyên tắc thay thế cho rất nhiều luật	41
3.1.3.1. Những thành phần cơ bản	41
3.1.3.2. Tham số.....	41
3.1.4. Câu hỏi đặt ra	42
3.1.5. Các nguyên tắc	43
3.1.5.1. Thuyết Xbar (\bar{X} theory)	43
3.1.5.2. Tiêu chuẩn Theta (Theta Criterion).....	44
3.1.5.3. Bộ lọc vai (Case-Filter).....	45
3.1.5.4. Thuyết kết hợp(Binding Theory).....	47
3.1.5.5. Thuyết về tính cục bộ và trường rộng	47
3.1.5.6. Thuyết dịch chuyển	48
3.1.6. Trật tự kết hợp các nguyên tắc	48
3.1.6.1. Dự đoán lỗi trước	49
3.1.6.2. Mô hình động.....	49
3.1.7. Các bước phân tích cú pháp	50
3.1.7.1. Phân tích từ vựng.....	50
3.1.7.2. Phân tích và tìm ra các cây cú pháp thích hợp	50
3.1.7.3. Chọn cây cú pháp thích hợp nhất	55
3.1.7.4. Trọng số	55
3.1.7.5. Chọn cây	55
3.2. Mô hình liên kết từ/ngữ trong song ngữ Anh-Việt.....	56
3.2.1. Giới thiệu mô hình dịch máy thống kê	56
3.2.2. Định nghĩa liên kết từ/ngữ.....	59

3.2.3.	Mô hình ngôn ngữ.....	62
3.2.4.	Mô hình dịch	64
3.2.4.1.	Mô hình 1	67
3.2.4.2.	Mô hình 2.....	69
3.2.4.3.	Một cách đặt vấn đề khác.....	71
3.2.4.4.	Mô hình 3	73
3.2.4.5.	Mô hình 4.....	75
3.2.4.6.	Mô hình 5.....	76
3.2.5.	Thuật toán Ước lượng-Cực đại (Estimation-Maximization Algorithm – viết tắt là thuật toán EM).....	78
3.2.6.	Cải tiến thuật toán EM trong mô hình 3, 4 và 5.....	80
3.2.7.	Tìm liên kết từ tối ưu nhất.....	84
3.2.8.	Cải tiến mô hình liên kết từ để liên kết ngữ	85
3.3.	Chiều kết quả phân tích cú pháp sang Tiếng Việt	89
3.3.1.	Chiều nhận từ loại.....	89
3.3.2.	Chiều quan hệ.....	90
3.3.3.	Sử dụng luật tương tác.....	90
Chương 4:	CÀI ĐẶT THỰC NGHIỆM.....	91
4.1.	Chương trình phân tích cú pháp quan hệ	91
4.1.1.	Phân tích từ vựng	91
4.1.1.1.	Từ điển	91
4.1.1.1.1.	Cấu trúc	91
4.1.1.1.2.	Sự phân loại động từ.....	94
4.1.1.1.3.	Mục từ tham chiếu	96
4.1.2.	Phân tích cú pháp quan hệ.....	97
4.1.2.1.	Từ điển chủ ngữ của động từ.....	97
4.1.2.2.	Mạng cú pháp	98

4.1.2.3.	Sơ đồ lớp	99
4.1.2.4.	Kết quả đầu ra.....	100
4.1.3.	Các thuộc tính	101
4.2.	Chương trình liên kết từ/ngữ.....	102
4.2.1.	Phân tích	102
4.2.1.1.	Phân tích tổng quát.....	103
4.2.1.2.	Phân tích chi tiết.....	104
4.2.1.2.1.	Lưu đồ của mô hình huấn luyện dịch thống kê $P(v e)$	104
4.2.1.2.2.	Lưu đồ của mô hình liên kết ngữ	105
4.2.2.	Thiết kế.....	107
4.2.2.1.	Sơ đồ lớp	107
4.2.2.2.	Danh sách các thuộc tính của từng lớp	108
4.2.2.3.	Danh sách các phương thức của từng lớp	109
4.2.2.4.	Sơ đồ hoạt động tổng thể của các lớp cho quá trình huấn luyện.	111
4.2.3.	Cài đặt các hàm xử lý chính	112
4.2.3.1.	Hàm khởi gán thông số t trong lớp Model1.....	112
4.2.3.2.	Hàm khởi gán thông số a trong lớp Model2.....	112
4.2.3.3.	Vòng lặp EM trong lớp Model1	113
4.2.3.4.	Vòng lặp EM trong lớp Model2	113
4.2.3.5.	Vòng lặp EM trong lớp Model3	114
4.2.3.6.	Tìm liên kết tối ưu nhất trong mô hình 1	115
4.2.3.7.	Tìm liên kết tối ưu nhất trong mô hình 2	116
4.2.3.8.	Tìm liên kết tối ưu nhất trong mô hình 3	117
4.3.	Chiếu kết quả phân tích cú pháp sang Tiếng Việt	117
4.3.1.	Chiếu nhãn từ loại.....	117
4.3.2.	Chiếu quan hệ.....	118
4.3.3.	Sử dụng luật tương tác.....	119

Chương 5: KẾT QUẢ - ĐÁNH GIÁ – KẾT LUẬN – HƯỚNG PHÁT TRIỂN	120
5.1. Chương trình liên kết từ	120
5.1.1. Một số kết quả	120
5.1.2. Giao diện của chương trình thử nghiệm liên kết	124
5.1.3. Đánh giá	125
5.2. Chương trình phân tích quan hệ cú pháp	128
5.2.1. Kết quả	128
5.2.2. Đánh giá	130
5.2.2.1. Ngữ liệu mẫu	130
5.2.2.2. Kết quả đánh giá	131
5.3. Chương trình chiếu kết quả phân tích cú pháp	132
5.3.1. Chiếu kết quả từ loại	132
5.3.2. Chiếu kết quả phân tích quan hệ	134
5.4. Kết luận	134
5.5. Hướng phát triển	135
PHỤ LỤC A: Bảng qui ước các ký hiệu của mô hình dịch máy thống kê	136
PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ	139
PHỤ LỤC C: Bộ nhãn từ loại tiếng Anh	145
PHỤ LỤC D: Các mối quan hệ trong tiếng Anh	147
TÀI LIỆU THAM KHẢO	149

LỜI NÓI ĐẦU

Với sự phát triển như vũ bão của khoa học kỹ thuật như hiện nay, tin học trở thành một nhu cầu không thể thiếu được trong hầu hết các lĩnh vực của đời sống xã hội. Tuy nhiên, việc giao tiếp giữa người và máy không phải lúc nào cũng tự nhiên, thuận lợi. Nguyên nhân chính có lẽ là do có sự khác biệt lớn giữa hai thế giới người và máy. Ngành học xử lý ngôn ngữ tự nhiên ra đời cũng nhằm mục đích xoá đi ngăn cách khác biệt ngôn ngữ giữa người và máy tính.

Tuy nhiên, ngành xử lý ngôn ngữ tự nhiên là một lĩnh vực không dễ. Nó chỉ phát triển mạnh trong mấy thập niên gần đây. Đặc biệt là đối với các ngôn ngữ phổ biến trên thế giới như tiếng Anh, tiếng Hoa, tiếng Pháp... Quá trình nghiên cứu này đã để lại cho nhân loại nhiều thành tựu to lớn. Nhu cầu về kế thừa những thành quả của tiếng Anh để áp dụng cho các ngôn ngữ khác (như là tiếng Việt) là một nhu cầu thiết thực. Để thừa hưởng được những thành quả này, chúng tôi nghiên cứu các kết quả của phân tích cú pháp tiếng Anh và chiếu sang tiếng Việt thông qua liên kết từ/ngữ. Kết quả của việc phân tích cú pháp tiếng Anh và chiếu sang tiếng Việt được làm ngữ liệu cho việc học, giám sát và rút ra các luật chuyển đổi cú pháp giữa hai ngôn ngữ Anh-Việt để phục vụ cho chương trình dịch tự động Anh Việt.

Các bước cơ bản cho việc chiếu kết quả phân tích cú pháp bao gồm ba bước chính: đầu tiên là phân tích cú pháp cho ngôn ngữ nguồn (ở đây là tiếng Anh), sau đó liên kết từ/ngữ, cuối cùng sử dụng kết quả liên kết từ/ngữ để chiếu sang ngôn ngữ đích (ở đây là tiếng Việt). Trong bài luận văn này chúng tôi sẽ trình bày chi tiết các phương pháp cho từng bước xử lý này.

Nội dung của bài luận văn được sắp xếp thành 5 chương như sau:

Chương 1: trình bày khái quát các bước giải quyết vấn đề.

Chương 2: chúng tôi trình bày sơ lược các cách tiếp cận cho các bước xử lý và chọn ra cách tiếp cận tối ưu để nghiên cứu.

Chương 3: giới thiệu mô hình thuật toán chi tiết cho từng bước xử lý chính theo các cách tiếp cận mà chúng tôi đã chọn và được trình bày trong chương 2.

Chương 4: cài đặt cụ thể cho các bước xử lý.

Chương 5: nêu ra một số kết quả và cách đánh giá các kết quả đó, và cuối cùng là kết luận và đưa ra hướng phát triển.

Chương 1: TỔNG QUAN

Các bước cơ bản cho việc chiếu kết quả phân tích cú pháp bao gồm ba bước chính: đầu tiên là phân tích cú pháp cho ngôn ngữ nguồn (ở đây là tiếng Anh), sau đó liên kết từ/ngữ, cuối cùng sử dụng kết quả liên kết từ/ngữ để chiếu sang ngôn ngữ đích (ở đây là tiếng Việt). Trong chương này chúng tôi sẽ giới thiệu sơ lược các bước chính này để độc giả có thể nắm được khái quát các bước xử lý chính này.

1.1. Phân tích cú pháp quan hệ

Muốn có sự giao tiếp bằng ngôn ngữ tự nhiên giữa người và máy, đầu tiên máy tính phải hiểu được ngôn ngữ tự nhiên. Bước đầu tiên để hiểu được một câu, máy phải biết được cấu trúc của câu cũng như quan hệ giữa các thành phần trong câu. Xác định cấu trúc, quan hệ này được gọi là phân tích cú pháp.

Tuy nhiên, muốn phân tích cú pháp thì đầu tiên phải đánh nhãn được từ loại của từng từ trong câu, từ đó mới có thể tổng quát hoá cho máy hiểu được những cấu trúc và những quan hệ ở mức tổng quát có thể được.

1.2. Liên kết từ/ngữ

Vấn đề dịch giữa các ngôn ngữ là vấn đề cổ xưa và rộng rãi. Nhiều nhà nghiên cứu trên thế giới đã và đang làm việc cật lực để tìm ra các phương pháp cho dịch máy tự động. Do đó có nhiều cách tiếp cận khác nhau trong việc dịch tự động. Mặc dù vậy, vấn đề dịch máy vẫn còn là một vấn đề tranh cãi giữa các cách tiếp cận. Có một vài sự bất đồng ý kiến về các phương pháp để thực hiện. Một nhóm các nhà nghiên cứu theo cách tiếp cận cơ sở tri thức (knowledge-based) thì cho rằng để có được chất lượng dịch

cao thì đòi hỏi kiến thức ngôn ngữ học đáng kể và phải có cơ sở kiến thức lớn. Một nhóm khác theo cách tiếp cận thống kê (statistic) thì cho rằng trong thực tế không thể xây dựng một cơ sở tri thức đủ lớn để làm ngữ liệu khả thi, nhưng nếu dựa vào một ngữ liệu song ngữ (tiếng Anh là bilingual corpus, parallel text, hay bitext) lớn để tạo ra một mô hình thống kê thì có thể tạo một hệ thống dịch máy hiệu quả hơn. Còn một nhóm khác nữa thì cho rằng cả hai phương pháp đều có mặt mạnh và mặt yếu riêng của nó, và họ đã đề ra một phương pháp mới bằng cách kết hợp cả hai cách tiếp cận cơ sở tri thức và tiếp cận thống kê, và cách tiếp cận đó được gọi là cách tiếp cận lai (hybrid approach).

Đối với cách tiếp cận cơ sở tri thức thì công việc xây dựng từ điển, xây dựng các luật chuyển đổi hầu hết đều được xây dựng bằng tay bởi các chuyên gia ngôn ngữ. Như vậy, đối với cách tiếp cận này thì đòi hỏi công việc và thời gian rất lớn. Ngoài ra, chúng ta sẽ đặt câu hỏi rằng: “Cơ sở dữ liệu cho từ điển và các luật chuyển đổi bao nhiêu là đủ?”. Và đây là điểm yếu của cách tiếp cận cơ sở tri thức. Đối với cách tiếp cận thống kê thì các công việc xây dựng từ điển và xây dựng các luật chuyển đổi hoàn toàn tự động bằng máy tính. Máy tính sẽ thống kê và rút ra các thông số thống kê tương ứng về từ/ngữ hay cấu trúc giữa hai ngôn ngữ cũng như xác suất dịch giữa hai ngôn ngữ, và xác suất xuất hiện của từ/ngữ đó trong một ngữ cảnh nhất định nào đó. Khuyết điểm của cách tiếp cận này là hoàn toàn dựa vào ngữ song ngữ đã được dịch sẵn bởi con người, vì thế nếu dữ liệu được dịch tốt và ngữ liệu càng lớn thì độ chính xác trong việc thống kê càng cao.

Trong những năm gần đây, dịch máy đã đạt được những thành công nhờ vào công nghệ máy học, và việc học này được dựa vào ngữ liệu song ngữ. Để hệ dịch máy Anh-Việt có thể tiếp cận theo hướng này thì bước đầu tiên trong việc xử lý ngữ liệu song ngữ chính là việc liên kết từ/ngữ của ngôn ngữ nguồn (ở đây là tiếng Anh) với các từ/ngữ của ngôn ngữ đích (ở đây là tiếng Việt). Việc liên kết từ/ngữ không thể đơn thuần tra từ điển song ngữ Anh-Việt, vì sự phong phú trong cách dịch và tính đa nghĩa

của các từ trong cả hai ngôn ngữ. Ngoài ra còn có sự khó khăn rất lớn khác là do sự khác biệt về mặt từ vựng hoá (lexicalization) của hai ngôn ngữ khác biệt về loại hình: giữa tiếng Anh (một thứ tiếng biến hình) với tiếng Việt (một thứ tiếng đơn lập). Trong khuôn khổ bài luận văn này, chúng tôi sẽ trình bày các mô hình dịch máy thông kê để liên kết từ và cụm từ trong văn bản song ngữ Anh-Việt. Các mô hình mà chúng tôi đề cập đến được thực hiện hoàn toàn tự động bằng máy. Ngữ liệu song ngữ mà chúng tôi sử dụng khoảng một triệu câu song ngữ Anh-Việt được nhập từ cách sách song ngữ về khoa học kỹ thuật và đã được đánh liên kết bằng tay. Ngữ liệu này sẽ được đưa vào hệ thống để huấn luyện, tính xác suất, và thử nghiệm. Kết quả có được sau khi qua hệ thống là các câu song ngữ trong ngữ liệu sẽ được liên kết.

Kết quả của việc liên kết từ/ngữ mà chúng tôi thu được trong cách tiếp cận thống kê hết sức quan trọng đối với hệ dịch máy và góp phần không nhỏ cho các hướng tiếp cận khác như: khảo sát sự thay đổi trật tự từ của cây cú pháp tiếng Việt và cây cú pháp tiếng Anh, giải quyết vấn đề nhập nhằng ngữ nghĩa, gán nhãn phân tích cú pháp cho song ngữ Anh-Việt, ... Trong bài luận văn này chúng tôi sẽ trình bày cụ thể ứng dụng kết quả liên kết từ/ngữ cho việc gán nhãn phân tích cú pháp cho song ngữ Anh-Việt.

1.3. Chiều quan hệ cú pháp

Chiều quan hệ cú pháp là sử dụng kết quả liên kết từ/ngữ để ánh xạ kết quả của các mối quan hệ cú pháp đã được đánh nhãn trong tiếng Anh sang tiếng Việt. Quá trình chiếu này chia làm 2 giai đoạn: chiếu nhãn từ loại và chiếu quan hệ cú pháp.

1.3.1. Chiếu nhãn từ loại

Từ kết quả đánh nhãn từ loại trên câu tiếng Anh, thông qua mối liên kết từ/ngữ để đánh nhãn từ loại cho các từ/ngữ trong câu tiếng Việt. Các vấn đề cần giải quyết là:

Trong tiếng Anh, các từ được cách nhau bằng khoảng trắng trong khi đó từ trong tiếng Việt có thể gồm nhiều âm tiết (mỗi âm tiết cách nhau bằng khoảng trắng). Do đó, trước khi đánh nhãn từ loại cho tiếng Việt phải tách từ.

Tiếp theo, thông qua mối liên kết từ/ngữ, nhãn từ loại của tiếng Anh sẽ được chiếu sang tiếng Việt. Tuy nhiên, đây không phải là phép ánh xạ 1-1 bởi vì: hệ thống từ loại trong 2 ngôn ngữ là khác nhau. Ngoài ra, hai ngôn ngữ có sự khác biệt lớn về phong cách trình bày. Do đó, không phải lúc nào cũng tìm ra được sự tương ứng về từ loại giữa hai ngôn ngữ.

1.3.2. Chiếu quan hệ cú pháp

Cũng giống như chiếu nhãn từ loại, kết quả quan hệ cú pháp để chiếu sang tiếng Việt thông qua mối liên kết từ/ngữ. Tuy nhiên, những nhập nhằng do sự khác biệt giữa hai ngôn ngữ sẽ được giải quyết bằng các nhãn từ loại đã được đánh ở bước trước.

Hai bước này có mối quan hệ chặt chẽ, có thể nhờ vào từ loại để làm rõ cho quan hệ cú pháp, ngược lại nhờ vào quan hệ cú pháp có thể làm rõ được những từ bị nhập nhằng từ loại.

Chương 2: CÁC CÁCH TIẾP CẬN

Vấn đề chiếu kết quả phân tích cú pháp từ một ngôn ngữ này sang ngôn ngữ khác là một nhu cầu cần thiết cho các nước mà việc xử lý ngôn ngữ tự nhiên chưa được phát triển mạnh (như các nước đang phát triển trong đó có Việt Nam chúng ta). Do đó, trên thế giới đã có nhiều nhà khoa học nghiên cứu nhiều cách tiếp cận khác nhau cho vấn đề này. Các bước cơ bản để tiến hành công việc chiếu kết quả phân tích cú pháp bao gồm: đầu tiên là phân tích cú pháp cho ngôn ngữ nguồn, sau đó liên kết từ/ngữ, cuối cùng sử dụng kết quả liên kết từ/ngữ để chiếu sang ngôn ngữ đích. Phần đầu chúng tôi sẽ giới thiệu các cách tiếp cận của các cách phân tích cú pháp cho ngôn ngữ nguồn (tiếng Anh), phần hai chúng tôi sẽ giới thiệu các cách tiếp cận của liên kết từ/ngữ (từ tiếng Anh sang tiếng Việt), cuối cùng chúng tôi trình bày các phương pháp chiếu sang ngôn ngữ đích (tiếng Việt).

2.1. Phân tích cú pháp

2.1.1. Các phương pháp tiếp cận dùng luật phi ngữ cảnh (CFG)

2.1.1.1. Cách tiếp cận từ trên xuống (Top-Down)

Phân tích cú pháp theo cách tiếp cận từ trên xuống bắt đầu với kí hiệu S (sentence). Đây chính là cấu trúc cao nhất của một câu và hình thành nên trạng thái ban đầu của cấu trúc câu. Kế tiếp, mỗi kí hiệu trong chuỗi trạng thái hiện tại sẽ được viết lại thành những cấu trúc thấp hơn dựa vào các luật có sẵn tạo thành một *danh sách các kí hiệu*.

Ví dụ : *Câu bắt đầu với kí hiệu S , sau đó nó áp dụng luật $S \rightarrow NP VP$. Danh sách kí hiệu lúc này là $(NP VP)$. Sau đó, kí hiệu NP được xét đến và thoả mãn luật $NP \rightarrow ART N$. Danh sách luật lúc này sẽ là $(ART N VP)$...*

Chương 2: CÁC CÁCH TIẾP CẬN

Quá trình cứ lặp lại một cách đệ quy cho đến khi nào trạng thái của câu bao gồm toàn những kí hiệu kết thúc. Tuy nhiên, đến lúc này, câu nhập vào cũng phải được đưa vào kiểm tra để bảo đảm rằng toàn bộ câu đã được phân tích. Vì vậy, dù gặp phải một danh sách bao gồm toàn những kí hiệu kết thúc nhưng câu vẫn còn từ chưa được phân tích thì cấu trúc tìm được là một cấu trúc sai.

Tuy nhiên, bởi vì từ vựng của một ngôn ngữ là rất lớn cho nên có một loại luật dạng kí hiệu kết thúc \rightarrow từ vựng sẽ là rất lớn. Để tránh gặp phải trường hợp này, người ta đã tách riêng nó thành một từ điển gọi là từ điển từ loại.

Book : N , V

Like : V , RB

...

Do từ điển từ loại đã được tách ra nên trong danh sách luật sẽ không còn luật nào chứa luật từ vựng.

Một ví dụ đơn giản với bộ luật bao gồm 5 luật như sau:

Luật 1	$S \rightarrow NP VP$
Luật 2	$NP \rightarrow ART N$
Luật 3	$NP \rightarrow ART ADJ N$
Luật 4	$VP \rightarrow V$
Luật 5	$VP \rightarrow V NP$

Bảng 2.1. Ví dụ một số luật

Trạng thái của câu bây giờ được định nghĩa thành một cặp : một danh sách kí hiệu và một con số chỉ ra vị trí hiện tại trong câu. Vị trí này được đánh vào giữa 2 từ với 1 là vị trí trước từ đầu tiên (từ số 1).

Ví dụ :

$_1 I _2 eat _3 rice _4$

Và một trạng thái của câu:

$((N VP)2)$

Trạng thái này chỉ ra rằng : chương trình phân tích muốn tìm ra một N (danh từ) và được theo sau bởi một (ngữ động từ), bắt đầu từ vị trí 2. Dựa vào việc kí hiệu đầu tiên trong danh sách kí hiệu có là kí hiệu từ vựng hay không mà trạng thái mới sẽ được hình thành dựa trên trạng thái cũ.

Như vậy trạng thái kế tiếp sẽ là:

$((VP)3)$

Trạng thái này nói lên ý nghĩa là : cần phải tìm một V bắt đầu tại vị trí số 3 trong câu nhập. Nếu kí hiệu đầu tiên là kí hiệu không kết thúc, giống như VP, thì viết lại kí hiệu này bằng luật cú pháp phù hợp.

Trong ví dụ trên, nếu áp dụng luật (4) thì trạng thái kế tiếp sẽ là :

$((V)3)$

trong khi đó, nếu áp dụng luật (3) thì trạng thái kế tiếp sẽ là :

$((VNP)3)$

Thuật toán phân tích bảo đảm rằng tất cả các giải pháp đều được xét tới. Chính vì điều này mà khi có nhiều hơn một trạng thái mới có thể được hình thành thì phải xử dụng tất cả các trạng thái cho phép này. Một kĩ thuật đơn giản được gọi là quay lui theo vết(backtracking). Theo cách tiếp cận này, thay vì chỉ sử dụng một trạng thái có thể thì tất cả các trạng thái đều được xét tới. Lưu các trạng thái mới này thành những trạng thái dự phòng (backup state) rồi sau đó xét qua hết tất cả các trạng thái này. Nếu có một trạng thái nào đó dẫn đến không thể đi tiếp được nữa thì loại nó ra khỏi danh sách.

2.1.1.2. Thuật toán phân tích cú pháp từ trên xuống (Top-Down)

Thuật toán phát sinh ra một danh sách các trạng thái có thể gọi là *possibilities list*. Phần tử đầu tiên trong danh sách được chọn làm trạng thái hiện tại.

Thuật toán bắt đầu với trạng thái khởi tạo là $((S) 1)$ và không có trạng thái dự phóng.

- Bước 1 : Chọn trạng thái hiện tại : phần tử đầu tiên trong possibilities list – gọi là C - được chọn làm trạng thái hiện hành . Nếu danh sách này rỗng thì thuật toán thất bại – không có một cấu trúc nào phù hợp với câu nhập vào.
- Bước 2 : Nếu C rỗng và từ đang xét nằm ở cuối câu thì thuật toán thành công.
- Bước 3 : Ngược lại, phát sinh ra trạng thái mới có thể:
- Bước 4 : Nếu kí hiệu đầu tiên trong danh sách C là một kí hiệu từ vựng (từ loại) và từ trong câu tại vị trí đang xét phù hợp với kí hiệu từ vựng này thì xoá đi kí hiệu đầu tiên trong possibilities list và cập nhật vị trí từ vựng trong câu tăng lên 1.
- Bước 5 : Ngược lại, nếu kí hiệu đầu tiên trong danh sách kí hiệu của C là một kí hiệu không kết thúc (non-terminal) thì phát sinh một trạng thái mới cho mỗi luật mà có thể viết lại kí hiệu không kết thúc đó

2.1.1.3. Cách tiếp cận Từ dưới lên (Bottom-Up)

Giống như tên được gọi, quá trình hình thành cây cú pháp của phương pháp này đi từ mức thấp lên mức cao hay từ lá lên gốc. Điểm khác biệt giữa cách tiếp cận từ dưới lên và từ trên xuống được trình bày ở trên là các mà luật ngữ pháp được sử dụng. Ví dụ khi xét đến luật :

NP → ART ADJ N

Trong hệ thống từ trên xuống, bạn sử dụng luật để tìm NP bằng các tìm kiếm chuỗi ART ADJ N. Ngược lại, trong hệ thống từ dưới lên, từ kết quả hình thành ở bước trước đó, bạn đã có một chuỗi ART ADJ N và bạn gán cho chuỗi này nhãn là NP.

Thao tác cơ bản trong hệ thống từ dưới lên là tìm các chuỗi tuần tự phù hợp với vế phải và thay thế nó bằng vế trái của luật. Bạn có thể sử dụng xây dựng một bộ phân tích cú pháp từ dưới lên đơn giản bằng việc xây dựng hai tiến trình : tiến trình so khớp và tiến trình tìm kiếm. Cũng giống như cách tiếp cận từ dưới lên, trạng thái ban đầu sẽ được khởi tạo và trạng thái cuối cùng dần được hình thành. Tuy nhiên, trạng thái khởi

tạo ở đây là danh sách các từ trong câu và trạng thái thành công (nếu có) là kí hiệu S. Trạng thái thành công có thể được hình thành từ việc tìm và tất cả các cách có thể để :

• Viết lại một từ bằng từ loại có thể có của từ đó

• Thay thế một chuỗi kí hiệu phù hợp với vế phải luật bằng kí hiệu vế trái.

Không may, những thao tác đơn giản trên đây lại có chi phí rất cao bởi vì nó cứ lặp đi lặp lại công việc so khớp chuỗi kí hiệu với vế phải của các luật, điều này tăng gấp bội công việc cần thiết thực sự cần phải làm. Để tránh tình trạng này, cấu trúc dữ liệu gọi là *sơ đồ* (chart) được sử dụng để lưu lại các kết quả của các quá trình so sánh đã được thực hiện để tránh đi việc thực hiện lặp lại này.

Việc so khớp luôn để ý tới một thành phần gọi là khoá (key). Để tìm luật phù hợp với chuỗi, ta chỉ tìm kiếm những luật bắt đầu bằng trường khoá này để tìm ra luật có vế phải trùng khớp với chuỗi kí hiệu.

Giả sử bạn đang phân tích một câu bắt đầu với ART. Kí hiệu ART này được xem như là khoá. Như vậy, có 2 luật được tìm ra phù hợp với khoá là luật (2)(NP → ART N) và luật (3) (NP → ART ADJ N). Để lưu lại dấu vết để có thể biết được trong lần phân tích kế tiếp, sử dụng một dấu chấm () để chỉ ra vị trí đã được xét tới cho đến thời điểm hiện tại. Ta có 2 bản ghi như sau:

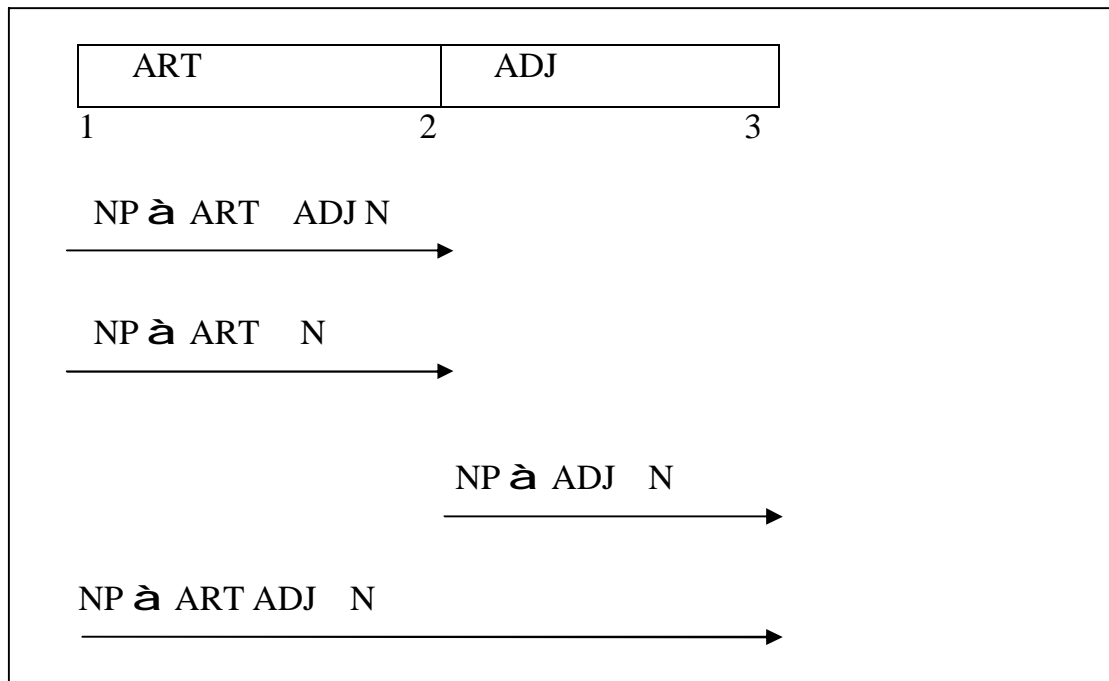
$NP \rightarrow ART \quad ADJ \quad N \quad (2')$

$NP \rightarrow ART \quad N \quad (3')$

Nếu khoá kế tiếp là ADJ thì luật 4 có thể được bắt đầu và bản ghi 2' được thay đổi như sau:

$NP' \rightarrow ART \quad ADJ \quad N \quad (2'')$

Sơ đồ sẽ bảo đảm lưu trữ toàn bộ những luật ứng viên đã được xét. Nó cũng lưu trữ bản ghi của những luật trùng khớp chỉ mới phần đầu. Những bản ghi này được gọi là những cung đang hoạt động. Ví dụ, sau khi tìm ra ART theo sau bởi một ADJ trong ví dụ trước đây, bạn sẽ có một sơ đồ như hình 2.1.



Hình 2.1. Phân tích cú pháp bằng phương pháp Bottom-Up

Ta có thể diễn giải ý nghĩa của sơ đồ trên như sau:

Có 2 luật ứng viên đã hoàn tất là ART từ vị trí 1 đến 2 và ADJ từ vị trí 2 đến 3. Có 4 cung hoạt động tương ứng với 4 luật ứng viên tương ứng với 4 mũi tên trên hình. Chiều của mũi tên là chiều đi từ thấp lên cao (từ lá lên gốc của cây cú pháp). Ý nghĩa của các cung trên hình là

- Ü Có 1 khả năng cho NP xuất hiện tại vị trí 1 cần một ADJ bắt đầu tại vị trí 2
- Ü Có 1 khả năng cho NP xuất hiện tại vị trí 2 cần một N bắt đầu tại vị trí 2
- Ü Có 1 khả năng cho NP xuất hiện tại vị trí 2 cần một N bắt đầu tại vị trí 3
- Ü Có 1 khả năng cho NP xuất hiện tại vị trí 1 cần một N bắt đầu ở vị trí 3

Phép toán cơ bản của phép phân tích cú pháp dựa trên sơ đồ là kết hợp các ứng cử viên đã hoàn tất với các cung đang hoạt động. Một luật mới hoàn tất sẽ được giữ lại trong một danh sách được gọi là nhật kí cho đến khi nó được thêm vào sơ đồ.

2.1.1.4. Thuật toán phân tích cú pháp Earley

Giải thuật phân tích cú pháp là một giải pháp kết hợp 2 phương pháp vừa trình bày ở trên. Trước khi đi chi tiết vào thuật toán, ta sẽ xét lại một số ưu khuyết điểm của từng phương pháp để thấy được lợi điểm khi kết hợp 2 phương pháp lại với nhau.

Ù Thuật toán phân tích cú pháp Top-Down có một ưu điểm là có một tầm nhìn bao quát. Một từ có thể nhập nhằng về từ loại. Tuy nhiên, nếu một từ loại được xét được thấy không có khả năng hình thành một cấu trúc cú pháp hợp lí thì nó sẽ được loại bỏ ngay lập tức và khả năng tiếp theo sẽ được xét đến. Tuy nhiên, bất lợi của thuật toán phân tích này là nó phải so sánh lặp lại nhiều lần những trường giống nhau. Như trong ví dụ trên thì mạo từ a được xét là ART đến 2 lần. Điều này làm tăng chi phí lên một cách không cần thiết.

Û Trong khi đó, thuật toán phân tích cú pháp Bottom-Down chỉ xét một từ với một từ loại chỉ một lần. Tuy nhiên, bởi vì nó xét đến nhãn kí hiệu trước nên tất cả các nhãn của một từ sẽ được xét tới mà không xét ngay đến tình hợp lí của từ loại này. Đây chính là bất lợi của giải thuật phân tích cú pháp từ dưới lên.

Thuật toán phân tích cú pháp Earley sẽ tận dụng lợi điểm của cách tiếp cận từ dưới lên bằng cách đi từ gốc về lá. Tuy nhiên, để tránh phải xét đi xét lại cùng một từ loại cho một từ duy nhất, giải thuật này sẽ đi cùng một lúc tất cả các hướng (tương ứng với các luật ứng viên thoả mãn xét đến thời điểm hiện tại). Đây chính là điểm tương đồng của Earley so với cách tiếp cận từ dưới lên. Như vậy nó đã khai thác được ưu điểm của hai phương pháp trên và cũng đồng nghĩa với việc loại bỏ đi những khuyết điểm của từng phương pháp.

Bây giờ ta sẽ đi tìm cách tiếp cận cụ thể của phương pháp Earley để thực hiện điều vừa trình bày trên.

Thuật toán phân tích Earley:

Gọi n là số từ trong câu. Ta xây dựng các bảng I_i với $i = 0, 1, 2, \dots, n$

- Bước 1 : Đầu tiên bảng I_0 được xác định bằng tất cả các luật có dạng $S \rightarrow A$, thì một trạm có dạng $S \rightarrow A, 0$ sẽ được thêm vào bảng I_0 gọi là một trạm.
- Bước 2 : Nếu $[B \rightarrow A, 0]$ thuộc I_0 , ta thêm $[A \rightarrow B, 0]$ cho tất cả các trạm $[A \rightarrow B, 0]$ thuộc I_0 .
- Bước 3 : Giả sử là $[A \rightarrow B, 0]$ là một trạm trong I_0 , ta thêm vào I_0 , cho tất cả các sản sinh trong P có dạng $B \rightarrow C$, trạm $[B \rightarrow C, 0]$ (miễn là trạm này chưa có trong I_0).
- Bước 4 : Các bảng I_j (với $j = 1 \dots n$) lần lượt được hình thành như sau: Với mỗi trạm $[B \rightarrow a, i]$ trong bảng I_{j-1} mà trong đó $a = a_j$, ta thêm $[B \rightarrow a, i]$ vào bảng I_j . Lặp lại các bước 5 và 6 cho tới khi không còn trạm nào được thêm vào.
- Bước 5 : Giả sử $[A \rightarrow B, i]$ là một trạm trong bảng I_j . Kiểm tra trong bảng I_j xem có những trạm nào có dạng $[B \rightarrow A, k]$ hay không, với mỗi trạm tìm thấy ta thêm $[B \rightarrow A, k]$ vào bảng I_j .
- Bước 6 : Giả sử $[A \rightarrow B, i]$ là một trạm trong bảng I_j . Đối với mọi sản sinh $B \rightarrow C$ trong P , ta thêm $[B \rightarrow C, j]$ vào bảng I_j .

2.1.1.5. Mạng ngữ pháp lan truyền

Cho đến lúc này, ta chỉ mới xét đến một phương pháp biểu diễn ngữ pháp gọi là luật phi ngữ cảnh. Bây giờ ta xét đến một hình thức biểu diễn khác của ngữ pháp được sử dụng rất rộng rãi trong các ứng dụng đó là mạng *ngữ pháp lan truyền*. *Mạng ngữ pháp này dựa trên các **nốt** và **cung***. Có 2 nốt đặc biệt là nốt bắt đầu và nốt kết thúc.

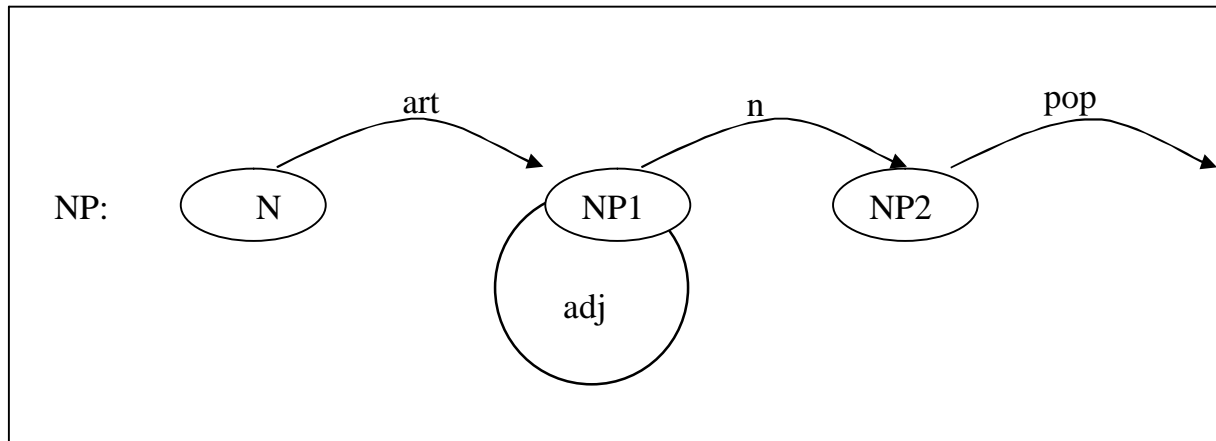
Để đơn giản, ta bắt đầu bằng một ví dụ cụ thể: biểu diễn NP (với các luật đã được trình bày trong phần trước bằng mạng ngữ pháp).

NP \rightarrow ART NP1

NP1 → ADJ NP1

NP1 → N

Lúc này mạng ngữ pháp sẽ là



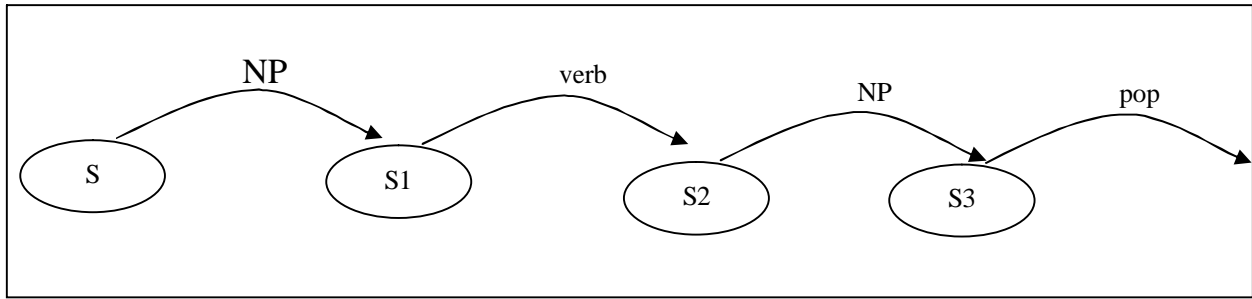
Hình 2.2. Mạng ngữ pháp NP

Trạng thái khởi đầu có nhãn là NP. Bắt đầu tại trạng thái khởi tạo, bạn có thể đi ngang qua một cung nếu từ loại của từ đang xét giống với nhãn của cung đó. Nếu một cung được chấp thuận thì vị trí của từ sẽ được cập nhật đến từ kế tiếp.

Một mạng lan truyền đơn giản như trên gọi là máy trạng thái hữu hạn (finite state machines – FSMs). Loại mạng này là một công cụ đặc lực đối với ngữ pháp đúng quy tắc thông thường nhưng không đủ sức mạnh để mô tả tất cả các ngôn ngữ có thể biểu diễn dưới dạng luật phi ngữ cảnh (CFG). Để có được sức mạnh mô tả của CFGs, ta cần quan tâm đến mạng ngữ pháp đệ quy. Cũng giống như mạng ngữ pháp đơn giản, tuy nhiên mạng ngữ pháp đệ quy có một điểm mới hơn là nó cho phép các cung có thể là một mạng khác (tương ứng với một nhãn không kết thúc) thay vì là một nhãn từ loại).

Như trong ví dụ trong hình 2.3, trường từ loại sẽ được biểu diễn là chữ thường, trường tham chiếu đến một mạng ngữ pháp khác được kí hiệu là chữ in hoa. nốt S và S₁ được nối với nhau bằng một cung NP- đây chính là mạng ngữ pháp được biểu diễn trong hình 2.2.

Chương 2: CÁC CÁCH TIẾP CẬN



Hình 2.3. Mạng ngữ pháp đệ quy

Loại cung	Ví dụ	Ý nghĩa
CAT	Verb	Thành công chỉ khi từ đang xét có thể đánh nhãn CAT
WRD	Of	Thành công chỉ khi từ đang xét giống với WRD
PUSH	NP	Chỉ thành công khi mạng con thành công
JUMP	Jump	Luôn thành công
POP	Pop	Thành công, trả về dấu hiệu thành công của mạng

Bảng 2.2. Ý nghĩa của các cung trong mạng ngữ pháp

Thuật toán phân tích cú pháp Top-Down sử dụng mạng ngữ pháp lan truyền đệ quy:

Các khái niệm:

- Vị trí hiện tại: Con trỏ chỉ đến từ kế tiếp được xét.
- Nốt hiện tại: Nốt đang xét đến trong mạng ngữ pháp.
- Điểm trả về: Một ngăn xếp của những nốt trong mạng khác. Bạn sẽ chỉ tiếp tục nếu mạng này trả về giá trị thành công (pop).

Giống như cách tiếp cận từ trên xuống truyền thống, nhưng thay vì xét tính hợp lệ của một luật thì trong mạng ngữ pháp này, ta xét đến việc có thể di chuyển qua một cung có được hay không

Trường hợp 1 : Nếu tên cung là nhãn từ loại và từ kế tiếp trong câu thuộc nhãn từ loại đó

Thì

Ù Cập nhật vị trí hiện tại tới từ kế tiếp.

Ù Cập nhật nốt hiện tại tới nốt đích của cung hiện tại.

Trường hợp 2 : Nếu cung là dạng cung đưa vào (push) một mạng N
Thì

Ù Thêm đích của cung đến điểm trả về.

Ù Cập nhật nốt hiện tại là nốt đầu tiên trong mạng N.

Trường hợp 3 : Nếu cung thuộc dạng cung đưa ra (pop) và điểm trả về khác rỗng

Thì Xoá phần tử đầu tiên trong điểm trả về và lấy đó làm nốt hiện tại.

Trường hợp 4 : Nếu cung thuộc dạng cung đưa ra (pop) và điểm trả về là rỗng và không còn từ nào bị bỏ đi.

Thì phân tích cú pháp thành công.

2.1.2. Phương pháp TBL (Transformation-Based Error-Driven Learning)

Phương pháp TBL được giới thiệu lần đầu tiên bởi Eric Brill vào năm 1993. Đến năm 1995 thì nó được công bố rộng rãi. Đây là một phương pháp rất mạnh trong lĩnh vực ngôn ngữ học và được áp dụng để giải quyết nhiều bài toán ngôn ngữ khác nhau.

Ý tưởng cơ bản của phương pháp TBL là dựa vào một ngữ liệu đã được đánh nhãn đúng, nó cố gắng tự đi tìm những luật để sửa những lỗi sai theo nguyên lí tham lam. Những luật được rút ra bởi TBL không giống với những luật phi ngữ cảnh ở các phương pháp trước. Nó là những luật tương tác.

Giải thuật TBL có 2 giai đoạn riêng biệt là giai đoạn học và giai đoạn chạy.

Trong giai đoạn học, dựa vào ngữ liệu đã được đánh nhãn, hệ thống sẽ cố tìm ra các luật tương tác để có thể đánh nhãn càng giống càng tốt.

Ngược lại, trong giai đoạn chạy, dựa vào bộ luật đã được rút ra để đánh nhãn cho một tập văn bản chưa được đánh nhãn.

Quá trình học:

Đầu tiên, ngữ liệu đã được đánh nhãn đúng sẽ được bỏ nhãn đi tạo thành một văn bản không có nhãn.

Kế đó, văn bản không có nhãn này sẽ được đánh nhãn ban đầu gọi là nhãn ngây thơ hay nhãn cơ sở. Nhãn ngây thơ có thể là sai rất nhiều theo cách đánh nhãn ngẫu nhiên hay cũng có khi khá chính xác nếu sử dụng một chương trình đánh nhãn nào đó. Trong lĩnh vực phân tích cú pháp, nhãn cơ sở có thể được đánh một cách đơn giản theo cách phân câu thành những ngữ theo nguyên tắc nhị phân.

Có 2 điều cần lưu ý khi đánh nhãn cơ sở:

- ⊘ Không nên sử dụng những đặc trưng của ngôn ngữ, điều này làm giảm tính cơ động của chương trình.

- ⊘ Hãy để cho chương trình tự học ra những luật hữu ích, không nên tốn quá nhiều thời gian để tự xử lí.

Dựa vào các mẫu luật đã được tạo sẵn, các luật ứng viên sẽ được hình thành và được áp dụng vào văn bản đã được đánh nhãn cơ sở. Khác với các luật CFG đã được trình bày trong các chương trước, luật TBL là các luật chuyển đổi dùng để thay đổi nhãn của từ. Quá trình đánh nhãn cú pháp sẽ là quá trình thêm và xoá các nhãn này.

Từng luật ứng viên, khi đưa vào áp dụng thử trên văn bản đã được đánh nhãn cơ sở theo từng câu một. Điểm sẽ được chấm cho mỗi luật làm căn cứ cho việc chọn lựa luật tốt nhất. Quy tắc chấm điểm như sau:

- ⊘ Nếu luật không làm thay đổi gì thì không thay đổi điểm.

- ⊘ Nếu luật sửa đúng thành sai thì cộng một điểm.

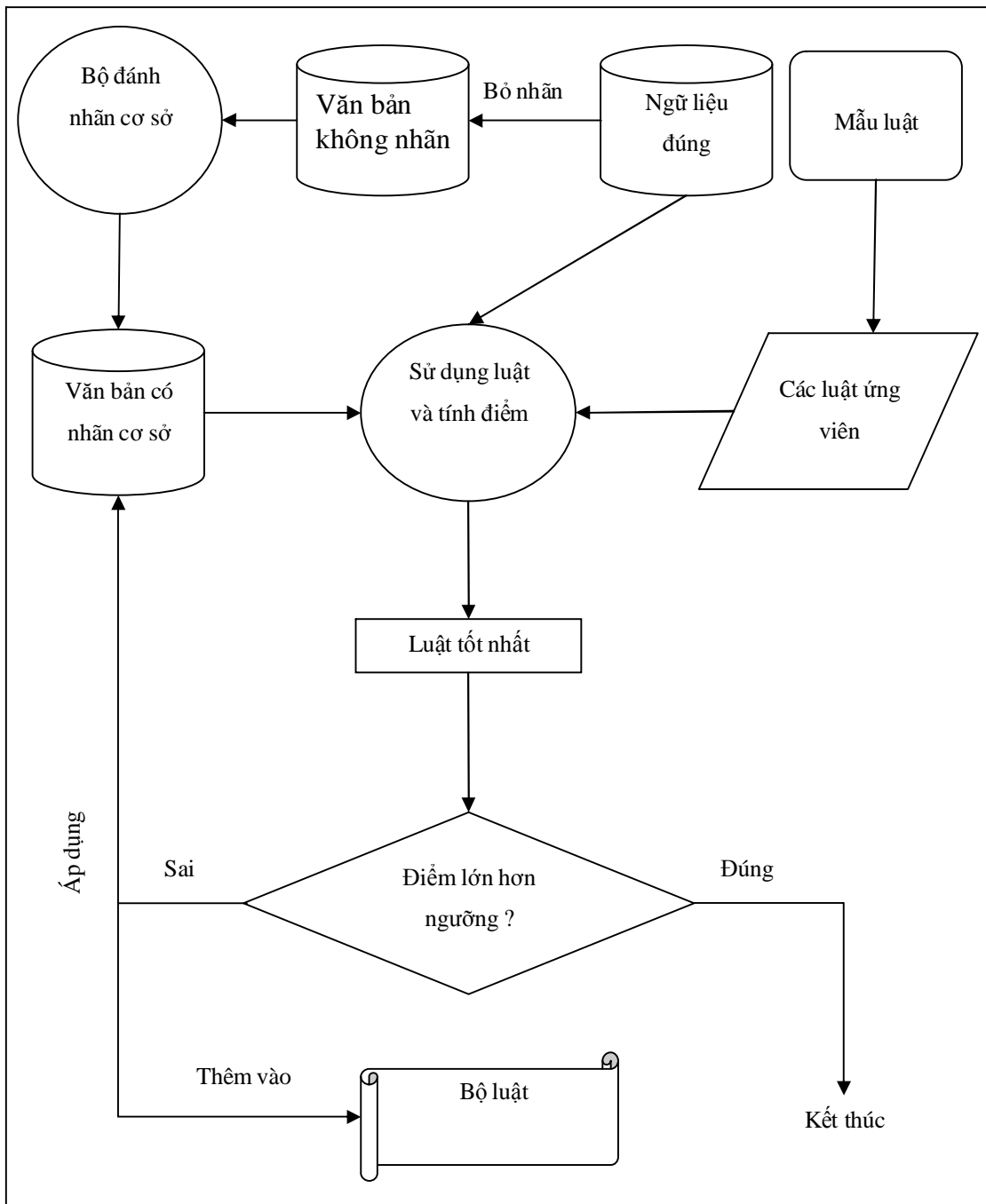
- ⊘ Nếu luật sửa sai thành đúng thì trừ một điểm.

- ⊘ Nếu luật sửa sai thành sai thì không thay đổi điểm.

Sau khi tất cả các luật đã được áp dụng cho tất cả các câu trong văn bản, chọn ra một luật có điểm lớn nhất để giữ lại nếu điểm nó vượt một ngưỡng cho trước. Dùng luật này để sửa nhãn cho văn bản đã được đánh nhãn cơ sở. Bởi vì điểm của luật này luôn dương (bởi ngưỡng là số dương) nên độ chính xác của văn bản bây giờ đã được tăng lên.

Quá trình trên lại được lặp lại: từng luật sẽ được áp dụng thử trên văn bản đánh nhãn cơ sở đã được sửa lại bởi luật được chọn. Luật tốt nhất lại được chọn ra...

Quá trình học sẽ ngừng khi tại một bước, số điểm của luật tốt nhất không vượt quá một ngưỡng cho phép. Nếu ngưỡng này được chọn quá lớn, số luật rút ra sẽ không được nhiều và độ chính xác không được cao. Ngược lại nếu ngưỡng được chọn quá nhỏ sẽ dẫn đến tình trạng quá luyện.



Hình 2.4. Sơ đồ học của TBL

Quá trình chạy (đánh nhãn):

Kết quả của quá trình học là một bộ luật tương tác. Đó là một bộ luật mà sức mạnh nằm ở sự kết hợp của toàn bộ chứ không phải của riêng một luật nào. Thứ tự kết hợp

của các luật là thứ tự được rút ra trong quá trình học. Mỗi luật được rút ra trong một hoàn cảnh đặc biệt là: các luật trước nó đã được đánh nhãn rồi và đó là luật sửa được tốt nhất. Như vậy, độ chính xác sẽ giảm đi rất nhiều nếu một trong số các luật trước nó bị bỏ đi và sẽ không còn ý nghĩa nếu trật tự kết hợp các luật không còn nữa.

Đây chỉ là một tiến trình nhỏ trong quá trình học của TBL trong đó các luật được áp dụng theo trật tự đã được rút ra.

2.1.3. Phương pháp phân tích cú pháp dựa trên văn phạm TAG

Theo phương pháp TAG (Tree Adjoining Grammar-văn phạm nối cây) thì từ vựng của nó được tổ chức thành các cây gọi là cây sơ cấp và các cây phụ trợ, hệ thống sẽ tìm cách kết nối các cây con này thành một cây hoàn chỉnh cho toàn câu.

2.1.3.1. Văn phạm TAGs

Văn phạm TAGs gồm 2 thành phần chính là cây sơ cấp và cây phụ trợ

2.1.3.1.1. Cây sơ cấp

Các cây sơ cấp có đặc điểm sau:

• Mọi nốt lá của cây được đánh nhãn là một thành phần kết thúc hoặc thành phần không kết thúc. Mọi thành phần không kết thúc đều được đánh dấu cho sự thay thế (được kí hiệu thành một mũi tên trên hình).

• Mọi nốt không là nốt lá được đánh nhãn là thành phần không kết thúc. Nếu nốt được từ vựng hoá thì từ vựng sẽ được chèn vào tại nốt tương ứng. Mỗi cây sẽ được đánh một nhãn gọi là supertag.

2.1.3.1.2. Cây phụ trợ

Ngoài các cây sơ cấp còn có một tập các cây phụ trợ có đặc điểm sau:

Ừ Cũng giống như cây sơ cấp, mọi nốt lá của cây được đánh nhãn là một thành phần kết thúc hoặc thành phần không kết thúc. Mọi thành phần không kết thúc đều được đánh dấu cho sự thay thế ngoại trừ một nốt gọi là nốt gốc (foot-node).

Ừ Điểm đặc biệt là nốt gốc có nhãn trùng với nhãn của nốt gốc. Chính điều này làm cho cây phụ trợ mang tính chất đệ quy.

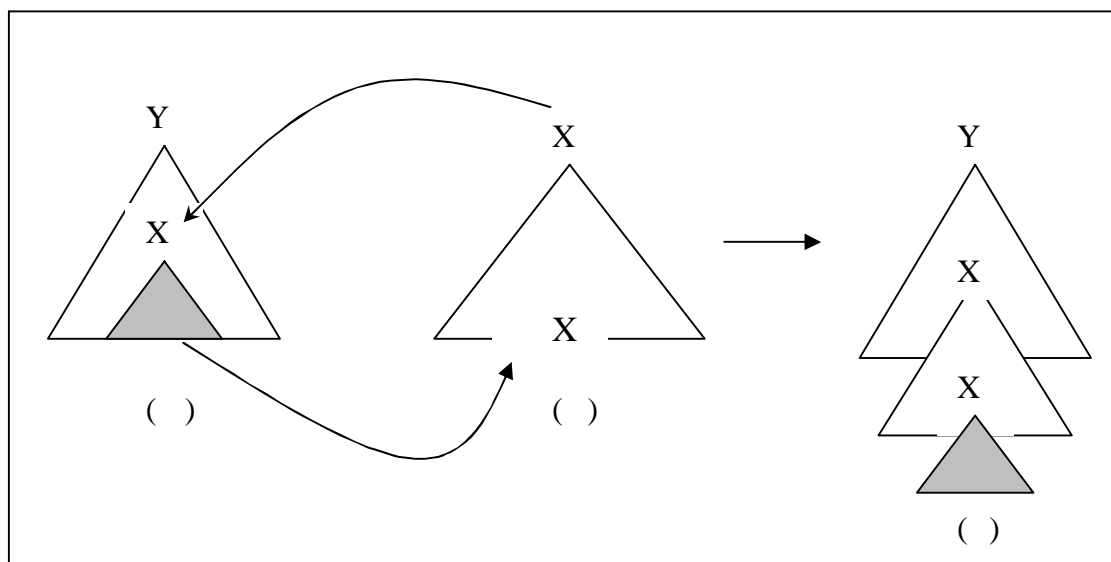
2.1.3.2. Các tác tổ trong TAGs

2.1.3.2.1. Tác tổ thêm vào

Tạo nên cây mới từ một cây phụ trợ (auxiliary tree) và một cây (có thể là cây khởi tạo, hoặc cây phụ trợ, hoặc cây kế thừa).

Giả sử chứa một nốt n không đánh dấu thay thế và có nhãn là X, là một cây là một cây phụ trợ có nốt gốc là X. Tác tổ thêm vào kết hợp cây vào cây để tạo thành một cây mới được tiến hành như sau:

- Ừ Trích ra thành phần con tại nốt n gọi là t
- Ừ Cây phụ trợ được gắn vào tại nốt n.
- Ừ Cây con t được ghép vào nốt gốc của cây .



Hình 2.5. Minh hoạ công việc tác tổ thêm vào

2.1.3.2.2. Tác tổ thay thế:

Tạo nên cây mới bằng cách thay thế nốt được đánh dấu thay thế bằng một cây tương ứng.

2.1.3.3. Những điều kiện kết hợp trên cây

Một cây phụ trợ có thể thêm vào cây tại nốt n nếu :

- Nốt n được đánh nhãn bằng một thành phần không kết thúc và không được đánh dấu cho sự thay thế.

- Nhãn của nốt n giống với nhãn nốt gốc tại cây .

Ngoài ra còn một số điều kiện mở rộng khác được định nghĩa cho sự thêm vào. Đối với mỗi nốt trên cây sơ cấp có 3 điều kiện:

- **Null Adjunction** (kí hiệu là NA) : Cấm sự thêm vào.

- **Obligatory Adjunction** : Bắt buộc phải có sự thêm vào.

- **Selective Adjunction** (kí hiệu SA(T)) : Có thể thêm vào hoặc không.

2.1.3.4. Cây rỗng

Trong tiếng Anh, có những lúc vị trí của các thành phần trong câu có sự thay đổi vị trí. Đó được gọi là sự dịch chuyển (movement). Một ví dụ cho trường hợp này là:

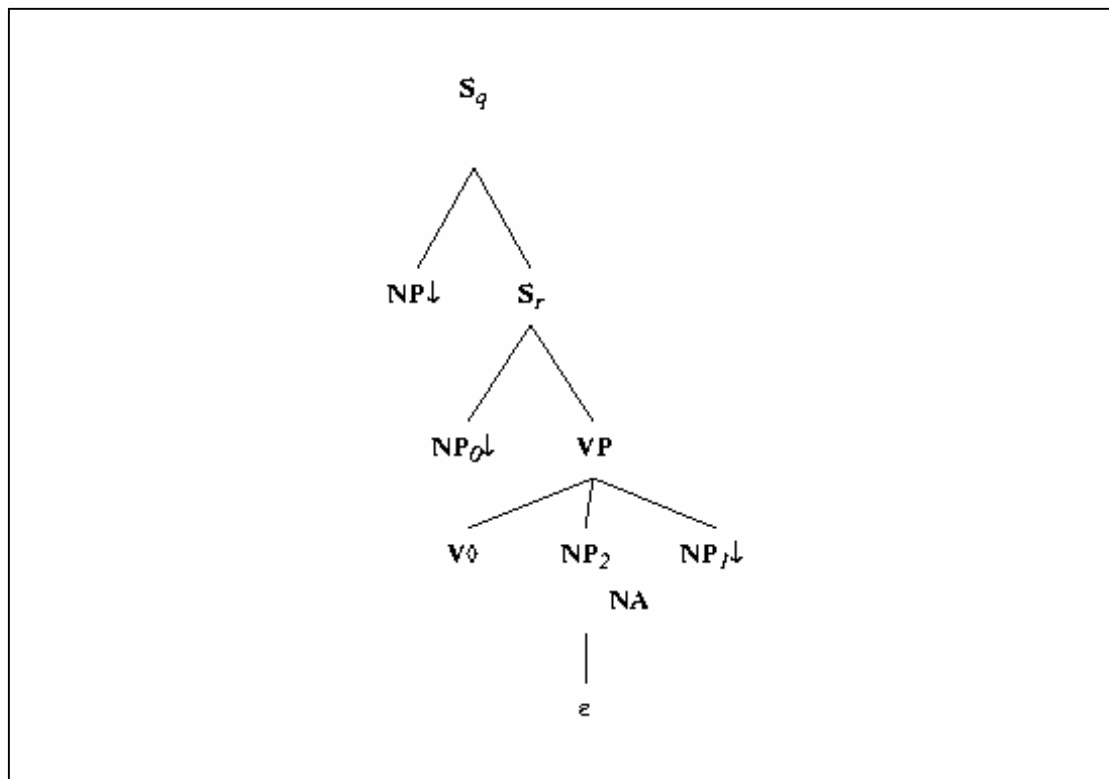
I love Marry **↔** Marry, I love.

Ngoài ra, câu hỏi yes/no, câu hỏi wh-question cũng có sự dịch chuyển của chủ từ so với động từ to be, động từ phụ trợ, của túc từ so với động từ...

Để mô tả sự chuyển vị trí này, người ta sử dụng một nốt đặc biệt gọi là nốt rỗng (epsilon ϵ).

Nốt ϵ sẽ đánh dấu cho một trường cần thay thế nào đó. Trường tương ứng này sẽ có một con trỏ đến trường tương ứng tượng trưng cho sự dịch chuyển vị trí này.

Ví dụ trong câu : “+ “Who did Daina ask a question” (direct object). Khi đó, “who” sẽ đóng vai trò như là túc từ của động từ ask. Như vậy, sẽ có một con trỏ từ “who” đến một nốt rỗng đóng vai trò như là túc từ của động từ ask.



Hình 2.6. Cây cú pháp của câu “Who did you ask a question?”

2.1.4. Phương pháp phân tích cú pháp dựa trên nguyên tắc

Phương pháp phân tích cú pháp dựa trên nguyên tắc dựa trên một ý tưởng khái quát hoá của các luật phi ngữ cảnh. Như đã được trình bày ở các phần trước, muốn bao quát các trường hợp của ngôn ngữ tự nhiên, bộ bộ luật với vài ngàn luật cũng không thể gọi là đầy đủ. Tuy nhiên, xét về nội dung, rất có nhiều luật có một mối tương đồng nào đó. Chính vì vậy người ta nghĩ đến một hệ thống phân tích cú pháp dựa trên một số các nguyên tắc rất ít nhưng lại có khả năng thay thế các luật này.

Hệ thống các nguyên tắc

2.1.4.1.1. Thuyết X-Bar (\bar{X})

Thuyết này mô tả dạng cây cơ bản của ngôn ngữ. Theo thuyết này thì ngôn ngữ có 2 dạng thức (công thức) chính khi xét đến vị trí của từ chính (head-wood) đối với các từ trong cùng một ngữ. Trong tiếng Anh, từ động từ thường đứng đầu trong ngữ động từ, giới từ đứng đầu trong ngữ giới từ nên tiếng Anh thuộc loại ngôn ngữ “từ chính-tham số”. Tuy nhiên, một vài ngôn ngữ lại có cấu trúc ngược lại “tham số-từ chính”

2.1.4.1.2. Nguyên lý Theta

Mô tả tham số cần thiết của mỗi động từ. Mỗi động từ thường có một số tham số đi theo đã được quy định trước. Giống như con người, khi nói lên một động từ, người ta thường nghĩ đến các tham số của nó. Ví dụ : khi ai đó nhắc đến động từ “cho”, người ta thường nghĩ đến “ai cho”? Ai là “người được cho” ? Và cho “cái gì”? Cũng vậy, khi nhắc đến động từ “đi” thì người ta cần biết “ai đi” và “đi đâu”?

Như vậy, mỗi một động từ hình thành xung quanh nó các khoảng chứa trống để điền vào gọi là các tham số.

2.1.4.1.3. Thuyết lọc vai (Case-filter)

Mỗi danh từ trong câu phải được gán một vai. Điều này có nghĩa là mỗi một danh từ trong câu phải giữ một vai trò nhất định nào đó. Chính vì ràng buộc này mà nó có tên là thuyết “lọc” vai.

2.1.4.1.4. Thuyết kết hợp

Mô tả mối liên hệ thay thế của một đại từ cho một danh từ nào đó. Mỗi đại từ phải thay thế cho một từ nào đó. Khi một đại từ được dùng, nó phải thay thế cho một danh từ nào đó đã được nhắc đến trước đây.

2.1.4.1.5. Thuyết về tính cục bộ và trường rộng

Xác định nơi nào một danh ngữ tiềm ẩn (trường rỗng) có thể xuất hiện trong câu. Một danh ngữ tiềm ẩn sẽ không được phát âm nhưng nó giữ một vai trò nhất định trong câu và vì vậy nó cần thiết để có thể hiểu được câu. Tuy nhiên, khoảng cách tương đối giữa danh ngữ tiềm ẩn và danh ngữ thực mà nó cần thay thế không được quá “xa” (liên quan cục bộ).

2.1.4.1.6. Thuyết dịch chuyển

Mô tả cách thức dịch chuyển của các thành phần trong câu. Có 2 loại dịch chuyển là noun và wh trong.

Mạng ngữ pháp.

Ngữ pháp lúc này sẽ được mô tả thành một mạng gọi là mạng ngữ pháp. Quá trình phân tích cú pháp cũng chính là quá trình truyền đi trong mạng. Tuy nhiên, khác với mạng ngữ pháp lan truyền đã được trình bày trong phần trước, quá trình lan truyền bây giờ không còn tuân theo luật nữa mà khi đi qua một cung, các nguyên tắc sẽ được xét đến, nếu thoả điều kiện thì sẽ được truyền qua.

Hàng ngàn luật phi ngữ cảnh sẽ được thay thế bằng 6 nguyên tắc được trình bày trên đây. Chính vì số lượng luật đã giảm đi một cách đáng kể như vậy cho nên mạng ngữ pháp lúc này cũng đơn giản đi nhiều và do đó tốc độ tăng lên một cách đáng kể.

2.2. Các cách tiếp cận trong việc liên kết từ/ngữ

Trong những năm gần đây, vấn đề dịch máy được xem như mục đích lâu dài của ngành khoa học máy tính. Để máy tính dịch được từ một ngôn ngữ này sang một ngôn ngữ khác thì máy tính phải biết các thông tin của cả hai ngôn ngữ đó như: những từ hay cụm từ tương đồng về nghĩa giữa hai ngôn ngữ, ngữ pháp của hai ngôn ngữ, tri thức của ngữ nghĩa và của thế giới thực. Một cách đơn giản cho công việc này là nhờ các nhà ngôn ngữ học nhập các thông tin cần thiết vào trong máy tính. Công việc này

phải đòi hỏi thời gian và công sức rất lớn mà lại không thể tìm ra hết các quy luật tương đồng cũng như dị biệt giữa hai ngôn ngữ đó, tính khách quan lại không cao. Như vậy, các nhà khoa học máy tính và ngôn ngữ học lại tìm một cách giải quyết khác là để cho máy tính học các thông tin của cả hai ngôn ngữ một cách tự động dựa vào một số lượng lớn các cặp câu song ngữ được xây dựng sẵn (ngữ liệu song ngữ là ngữ liệu gồm các cặp câu đã được dịch từ một ngôn ngữ này sang một ngôn ngữ khác một cách gần chính xác). Các nguyên nhân để có thể chứng minh giải pháp máy học có thể giải quyết được vấn đề dịch máy là:

Với sự lớn mạnh của các ngữ liệu song ngữ từ nhiều nguồn khác nhau, nhiều cấp độ chú thích khác nhau, nhiều ngôn ngữ khác nhau, nhiều lĩnh vực khác nhau, ...

Với sự phát triển như vũ bão của công nghệ phần cứng đã lôi kéo theo sự phát triển mạnh mẽ của phần mềm và nó cho phép xử lý một khối lượng lớn thông tin với các thuật toán đòi hỏi chi phí cao.

Một vài con số thống kê cho thấy sự phát triển theo hướng máy học trong lĩnh vực nghiên cứu ngôn ngữ tự nhiên: Vào năm 1990 chỉ có 12,8% các công trình công bố ở hội nghị hàng năm của Tổ chức ngôn ngữ học máy tính (Proceedings of Annual Meeting of the Association of Computational Linguistics) và 15,4 % công trình đăng trên tạp chí Ngôn ngữ học máy tính (Computational Linguistics) liên quan đến hướng nghiên cứu sử dụng tập ngữ liệu, con số này đến năm 1997 là 63,5% và 47,7%.

Cho đến nay, đối với cách tiếp cận máy học thì đã có nhiều đề án nghiên cứu về việc liên kết từ trong song ngữ, và các đề án đó đã đưa ra nhiều phương pháp tiếp cận, và mỗi phương pháp có ưu và khuyết điểm riêng của nó. Các phương pháp liên kết từ trong song ngữ được phân loại như sau:

Hướng tiếp cận dựa trên việc sử dụng từ điển song ngữ. Thuật toán sử dụng một từ điển song ngữ để tra nghĩa của từ và hình thành cặp liên kết từ 1-1 (nếu có) như một cặp dịch tương ứng. Thuật toán này tỏ ra kém hiệu quả bởi vì trong

thực tế thì cách dịch từ một ngôn ngữ này sang một ngôn ngữ khác rất phong phú.

Hướng tiếp cận dựa vào từ cùng nguồn gốc. Phương pháp này chỉ áp dụng được cho cặp ngôn ngữ có cùng nguồn gốc hay cùng loại hình như tiếng Anh-tiếng Pháp, còn đối với cặp ngôn ngữ khác loại hình như tiếng Anh và tiếng Việt thì không thể áp dụng được.

Hướng tiếp cận dựa vào từ điển phân lớp từ theo ý niệm hay ngữ nghĩa của từ. Đây là một phương pháp khá mới, thích hợp với những cặp ngôn ngữ có cách dịch phong phú, nhưng ngược lại đòi hỏi từ điển phân lớp từ phải được xây dựng một cách đầy đủ và phù hợp.

Hướng tiếp cận theo thống kê cổ điển với hai thuật toán tiêu biểu là K-vec và DK-vec.

Hướng tiếp cận theo dịch máy thống kê hiện đại được dựa vào mô hình phức hồi nhiều của tiếng nói. Mô hình này tỏ ra khá hiệu quả, vì nó có thể áp dụng cho nhiều cặp ngôn ngữ khác nhau và nó không cần quan tâm ý niệm về thể giới thực của các ngôn ngữ.

2.2.1. Char-Align – Hệ thống Termight

Hệ thống Termight được xây dựng như là một công cụ để tạo ra từ điển từ song ngữ do Ido Dagan và Ken Church phát triển tại phòng thí nghiệm AT&T Bell. Hệ thống này dựa vào đánh nhãn từ loại (POSTagger) và chương trình liên kết từ Word-Align. Word-Align dựa trên cơ sở là chương trình Char-Align. Char-Align làm việc trên mức ký tự và sử dụng từ cùng nguồn gốc của hai ngôn ngữ để tạo liên kết. Chính vì thế mà nó còn hạn chế bởi lịch sử phát triển ngôn ngữ cũng như nguồn gốc của chúng.

Char-Align là một chương trình được Ken Ward Church phát triển tại phòng thí nghiệm AT&T Bell. Char-Align làm việc trên mức ký tự và dựa vào hướng tiếp cận từ cùng nguồn gốc của Simard, Foster, and Isabelle. Đây là phương pháp sử dụng sự tồn

tại của những cặp từ có cùng nguồn gốc của hai ngôn ngữ. Tác giả đã đề nghị sử dụng những từ cùng nguồn gốc này để cải tiến phương pháp liên kết dựa vào độ dài cơ sở của từ bằng cách định nghĩa một “mức của từ cùng nguồn gốc” như sau:

$$\frac{c}{(n - m)/2} \quad (2.1)$$

với c là số lớn nhất của những từ cùng nguồn gốc trong cặp câu hiện tại, n là số từ trong câu của ngôn ngữ nguồn, và m là số từ trong câu của ngôn ngữ đích.

Từ cùng nguồn gốc được định nghĩa theo nhiều cách khác nhau. Một cách định nghĩa được đưa ra như sau: Nếu hai từ của một cặp từ cùng nguồn gốc có ít nhất một digit hoặc nếu chúng là những dấu chấm câu thì chúng là một cặp từ cùng nguồn gốc khi chúng y hệt nhau. Hoặc một định nghĩa khác là: nếu chiều dài của từ của chúng có ít nhất 4 ký tự đầu tiên và 4 ký tự liên tiếp ở phía sau của chúng giống nhau thì chúng cùng nguồn gốc.

Char-Align sử dụng 4-grams giống nhau để tìm liên kết giữa ngôn ngữ nguồn và ngôn ngữ đích. Theo hướng tiếp cận này, chương trình sử dụng một “ước lượng điểm chia” (dotplot calculation). Nếu có 4-gram tại vị trí x trong tài liệu nguồn, và một 4-gram tại vị trí y của tài liệu đích thì cờ tương ứng trong ma trận 2 chiều xy sẽ được bật. Để cải thiện tốc độ và giảm bớt không gian bộ nhớ, một số đường biên và ước lượng quyết định đã được tạo ra.

Trong bước cuối cùng, những liên kết tốt nhất giữa những điểm đã được tìm thấy. Một số heuristic đã được sử dụng ở đây khi lấy kết quả. Theo cách trọng số trung bình lớn nhất (được tính bằng tổng phần giao nhau cho độ dài của từ) sẽ được xem xét như là đường liên kết tốt nhất.

Tuy nhiên, đối với phương pháp này rất hạn chế (nếu không muốn nói là không khả thi) đối với cặp ngôn ngữ có nguồn gốc khác nhau.

2.2.2. Phương pháp K-vec

Thuật toán K-vec là một hướng tiếp cận theo thống kê cho liên kết từ trong song ngữ được giới thiệu bởi Pascale Fung của Đại học Comlumbia, New York[15].

Bước đầu tiên của phương pháp này là rút ra những từ ứng viên bằng cách tra những từ giống nhau giữa ngôn ngữ nguồn và ngôn ngữ đích. Vì mục đích này nên dữ liệu sẽ được chia ra làm K mảnh. Sau đó, những vector nhị phân K-chiều được tạo ra bởi các từ của ngôn ngữ nguồn và ngôn ngữ đích. Một mảnh nào đó chứa đựng từ trong ngôn ngữ nguồn (tương ứng từ trong ngôn ngữ đích) thì cở trong vector nhị phân đó sẽ được bật. Sau đó, phương pháp xác suất có thể được sử dụng để tìm ra những từ này giống nhau.

Trong phương pháp K-vec, cách tính điểm thông tin tương hỗ được sử dụng, nó được định nghĩa theo công thức:

$$I(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} \quad (2.2)$$

$P(x, y)$ là xác suất của từ x và y được tìm ra trong mảnh tương ứng, $P(x), P(y)$ là xác suất tìm thấy x, y .

Xác suất có thể được ước lượng bằng cách sử dụng số tần số xuất hiện tuyệt đối, $P(x, y)$ sẽ là $\frac{freq(x, y)}{K}$; $P(x), P(y)$ sẽ là $\frac{freq(x)}{K}$ và $\frac{freq(y)}{K}$. Ở đây, $freq(x, y)$ là tần số xuất hiện đồng thời của x và y ; $freq(x)$ và $freq(y)$ là tần số xuất hiện của x và y . Sử dụng đơn vị thông tin tương hỗ này, cặp dịch ứng viên có thể được sắp xếp và những cặp từ có thể nhất sẽ được chọn. Sau đó, những cặp này được sử dụng như điểm tham chiếu để liên kết ngữ liệu song ngữ.

Vấn đề là cách tính điểm thông tin tương hỗ với những từ có tần số xuất hiện thấp thì như thế nào. t-score được sử dụng để lọc ra những giá trị vô nghĩa. Trong trật tự như vậy có đến 95% các từ xuất hiện ít nhất ở 3 mẫu khác nhau.

Vấn đề tiếp theo là chọn K bao nhiêu là tối ưu nhất. Nếu K quá nhỏ thì thông tin tương hỗ sẽ trở nên thiếu chắc chắn. Nếu K quá lớn thì tín hiệu nhận biết sẽ bị mất.

2.2.3. Phương pháp DK-vec

Dựa trên phương pháp K-vec, Pascale Fung and Kathleen McKeown phát triển một thuật toán mới cho liên kết từ được gọi là DK-vec và sử dụng thuật toán quy hoạch động.

Không giống như thuật toán K-vec nội dung được phân thành K mẫu có kích thước giống nhau của những từ có khả năng xuất hiện trong ngôn ngữ nguồn và ngôn ngữ đích được lưu vào trong vector nhị phân. Trong DK-vec, khoảng cách giữa những lần xuất hiện của từ thuộc ngôn ngữ nguồn và ngôn ngữ đích được lưu trữ trong vector gần (recency vectors), Những vector này có thể thay đổi kích thước vì tần số xuất hiện khác nhau (cũng như từ tương ứng của ngôn ngữ nguồn và ngôn ngữ đích).

Những vector này có thể được xem xét như là những dấu hiệu, và thuật toán quy hoạch động được sử dụng để tìm ra dấu hiệu đối sánh trên những mặt phẳng ngôn ngữ nguồn và ngôn ngữ đích.

Thuật toán bắt đầu với việc tìm ra những vector gần cho mỗi từ của ngôn ngữ nguồn và ngôn ngữ đích. Kế tiếp là những cặp dịch ứng viên có thể được tìm thấy, Vì mục đích này, tất cả những cặp xuất hiện lần đầu tiên trong nửa sau của câu sẽ được lọc ra. Hơn nữa, tất cả những cặp mà vector nhỏ hơn một nửa chiều dài của những vector khác sẽ được bỏ đi. Đối với những cặp còn lại, sự khác biệt giữa những vector của chúng sẽ được tính bằng cách sử dụng thuật toán quy hoạch động. Trong bước cuối cùng, các cặp được sắp xếp bởi sự khác nhau tuyệt đối của chúng để tìm ra những cặp từ tương hỗ gần nhất.

2.2.4. Ánh xạ song ngữ với SIMR

Một phương pháp được sử dụng để ánh xạ được gọi là Smooth Injective Map Recognizer (SIMR) được Dan Melamed phát triển ở Đại học Pennsylvania, Philadelphia. Giống như Char_Align, sử dụng từ cùng nguồn gốc trong ngôn ngữ nguồn và ngôn ngữ đích để liên kết từ ở cấp độ ký tự.

Một cặp câu song ngữ được xem như là một không gian song ngữ 2-chiều [Mel96] với vị trí của các ký tự trong ngôn ngữ nguồn và vị trí của các ký tự trong ngôn ngữ đích. Mỗi không gian song ngữ bao gồm những “điểm đúng tương ứng” (True points of correspondence - TPC’s). Bắt đầu với không gian song ngữ nguồn, thuật toán tìm kiếm được gọi là “điểm đúng tương ứng” bằng cách mở rộng không gian tìm kiếm theo hướng tăng dần x và y . TPC’s có thể được nhận ra bằng cách sử dụng từ điển, thuật toán tìm kiếm từ cùng nguồn gốc, ...

Nếu “điểm đúng tương ứng” (TPC) mới được tìm ra và vị trí của nó không cùng giá trị x hoặc y trong không gian song ngữ như trước đó thì TPC mới sẽ dựa trên TPC được tìm ra mới nhất cho bước tìm kiếm kế tiếp.

Sau khi xây dựng được các “điểm đúng tương ứng”, các mối liên kết sẽ được chọn. Như vậy, các điểm TPC lân cận đường chéo trên không gian song ngữ sẽ có khả năng được chọn bằng cách lọc ra những liên kết ứng viên ở một vài bước.

2.2.5. Mô hình xác suất với thuật toán IPFP

Một mô hình liên kết dựa trên xác suất thống kê với thuật toán Iterative Proportional Fitting Procedure (IPFP) đã được giới thiệu. Trong thuật toán này, mỗi cặp câu được biểu diễn như một ma trận. Các dòng biểu diễn ngôn ngữ nguồn, các cột biểu diễn ngôn ngữ đích, và các phần tử trong ma trận chứa tần số liên kết mong muốn cho những từ ở dòng và cột tương ứng. Tần số liên kết mong muốn này chưa được biết và phải được thiết lập.

Có một thuật toán kinh điển trong thống kê được biết đến là Iterative Proportional Fitting Procedure (IPFP) cho việc thiết lập những giá trị sai trong bảng ngẫu nhiên. Bằng cách thay thế giá trị trong ô bởi tỷ lệ của tổng dòng và cột quan sát chia cho tổng dòng và cột được thiết lập. Giá trị ô này sau khi hội tụ sẽ cung cấp ước lượng hợp lý của tần số liên kết mong muốn. Cho một ước lượng khởi đầu cho tất cả các ô, thuật toán IPFP bao gồm những tính toán sau tại lần lặp thứ k đối với mỗi phần tử của ma trận n_{ij} :

$$\begin{aligned} n_{ij}^{(k,1)} &= n_{ij}^{(k-1,2)} * m_i / n_i^{(k-1,2)} \\ n_{ij}^{(k,2)} &= n_{ij}^{(k,1)} * m_j / n_j^{(k,1)} \end{aligned} \quad (2.3)$$

Trong đó:

n_i và n_j là giá trị hiện hành cho biên dòng và cột.

m_i và m_j là tần số xuất hiện giới hạn của dòng và cột quan sát.

Những bước này có thể được lặp đi lặp lại cho tới khi thiết lập biên và đủ gần tới biên quan sát.

Thuật toán IPFP không thực sự lý tưởng cho việc thiết lập những giá trị mong muốn với vài lý do. Tổng số dòng và cột phải giống nhau. Điều này tương đương với việc buộc câu nguồn và câu đích phải cùng độ dài, trong một số trường hợp thì không đúng. Ông Hiemstra giải quyết vấn đề này bằng cách thêm vào những từ trống cho những câu ngắn để cho chúng có cùng độ dài. Việc này bắt buộc cặp từ liên kết phải là 1-1. Trong khi IPFP hầu như luôn luôn hội tụ. Bất kỳ tập giá trị nào thỏa mãn giá trị tần số giới hạn quan sát hội tụ ngay lập tức tới một lời giải hợp lý. Vì thế, lời giải phụ thuộc cao vào giá trị khởi đầu và ước lượng khởi tạo hợp lý cần thiết.

Mô hình của Hiemstra đã thay thế việc tính toán tần số liên kết mong muốn đối với mỗi câu bằng tổng tất cả những tần số để tạo ra giá trị mong muốn chung. Mô tả toán học này được cung cấp ở phía dưới. Ký hiệu: $C_{n,[jk]}^i$ $C_n^i(s_j \quad t_k)$ ám chỉ tới giá trị

mong muốn tại lần lặp thứ i của việc liên kết giữa từ nguồn s_j và từ trong ngôn ngữ đích t_k trong câu n , và I là hàm logic.

Bước 0:

$$C_{[jk]}^0 = \prod_{n=1}^N I(s_j, S_n) * I(t_k, S_n) \quad (2.4)$$

Bước 1: Với mỗi s_j và t_k trong câu $n = 1 \dots N$

$$C_{n,[jk]}^i = C_{[jk]}^i \quad (2.5)$$

$$C_{n,[jk]}^i = IPFP \quad C_{n,[jk]}^{i-1} \quad (2.6)$$

Bước 2:

$$C_{n,[jk]}^i = \prod_{n=1}^N C_{n,[jk]}^{i-1} \quad (2.7)$$

Bước 3:

$$\text{Nếu } \prod_{n=1}^N |C_{[jk]}^{i-1} - C_{[jk]}^i| \leq K \quad (2.8)$$

thì dừng, ngược lại quay lại bước 1.

Từ giá trị toàn cục, chúng ta có thể nhận được từ việc ước lượng khả năng dịch theo hai hướng như sau:

$$P(s_j, t_k) = \frac{C(s_j, t_k)}{\sum_m C(s_j, t_m)} \quad (2.9)$$

$$P(t_k, s_j) = \frac{C(s_j, t_k)}{\sum_i C(s_i, t_k)} \quad (2.10)$$

2.2.6. Mô hình dựa vào sự phân lớp (Class-based)

Mô hình liên kết từ theo hướng tiếp cận dựa vào sự phân lớp từ [4] [8] (Class-based, cũng có thể gọi là ClassAlign) cũng là mô hình liên kết từ theo phương pháp thống kê. Cũng là thống kê nhưng Sue J.Ker và Jason S.Chang đưa ra thuật toán ClassAlign để tính xác suất các cặp liên kết. Thuật toán này tính xác suất $Pr(s/t)$ phụ thuộc vào xác suất dịch từ điển, xác suất sai lệch và công thức tính mức độ tương quan giữa hai lớp từ thuộc hai ngôn ngữ khác nhau. Thuật toán không chỉ sử dụng xác suất dịch từ-từ mà nó sử dụng lặp đi lặp lại thuật toán Estimation Maximization (EM) để ước lượng khả năng dịch. ClassAlign rất thành công trong cặp ngôn ngữ Anh - Hoa, 80% số từ ở ngôn ngữ nguồn được liên kết với từ đích và 90% liên kết đúng.

2.2.7. Mô hình liên kết dựa vào cách tiếp cận dịch máy thống kê (SMT)

Vào năm 1949, Warren Weaver [1] [2] đã đề nghị ứng dụng kỹ thuật thống kê và giải mã. Kỹ thuật này được biết đến từ lý thuyết truyền thông để giải quyết vấn đề sử dụng máy tính trong việc chuyển từ văn bản từ một ngôn ngữ này sang một ngôn ngữ khác. Song sự cố gắng trong lúc đó bị huỷ bỏ bởi lý do triết học và lý thuyết suông. Vào thời điểm đó thì hầu hết các cách tiếp cận đều phải chịu sự bất hạnh về sự thiếu thốn của công nghệ máy tính. Ngày nay, với sự phát triển như vũ bão của công nghệ kỹ thuật số cho phần cứng đã kéo theo sự phát triển về các cách tiếp cận trong việc dịch máy. Trong những thập niên cuối thế kỷ 20, cách tiếp cận thống kê đã được ứng dụng rộng rãi trong các hệ thống, trong đó có hệ thống dịch máy.

Năm 1993, Brown *et al.* [1] [2] [5] đã đưa ra mô hình toán học cho dịch máy thống kê. Ông đã chứng minh bằng lý thuyết xác suất thống kê cho việc dịch máy. Trong đó ông mô tả năm mô hình thống kê rất chi tiết. Các mô hình này dùng để huấn luyện và cho ra những thông số thống kê với độ phức tạp ngày càng tăng dần theo mô hình.

Trong các mô hình đó, chúng tôi ứng dụng chủ yếu từ mô hình 1 đến mô hình 3 và kết với thuật toán Estimation-Maximization (gọi tắt là thuật toán EM) để ước lượng tối ưu các thông số.

Trong bài luận văn này chúng tôi sẽ trình bày chi tiết 3 mô hình đầu của dịch máy thống kê và giới thiệu sơ lược về mô hình 4 và mô hình 5. Ngoài ra, chúng tôi còn trình bày cụ thể hơn việc cải tiến thời gian huấn luyện thông qua việc kết hợp với thuật toán leo đồi (Hill Climbing) [1] [2].

2.3. Các phương pháp chiếu

2.3.1. Chiếu nhãn từ loại

2.3.1.1. Phương pháp trực tiếp

Thông qua mối liên kết từ, mỗi từ loại, mỗi quan hệ trong ngôn ngữ này sẽ được chiếu trực tiếp sang ngôn ngữ kia. Ưu điểm của cách tiếp cận này là nó rất đơn giản. Tuy nhiên nó chỉ thật sự hiệu quả khi các ngôn ngữ này có mối liên hệ gần gũi, tương đồng với nhau.

2.3.1.2. Phương pháp Noise-robust

Ý tưởng cơ bản của phương pháp này xuất phát từ một thực tế là : các từ trong ngôn ngữ như Tiếng Anh, tiếng Pháp.. mỗi từ nó luôn hướng về một nhãn từ loại chính và rất hiếm khi có nhiều hơn 2 nhãn. Từ đó, phương pháp này sẽ làm mịn dần, các nhãn từ loại có xác suất thấp sẽ được loại trừ dần.

Tuy nhiên, phương pháp này đòi hỏi phải có một ngữ liệu tiếng Việt đã được đánh nhãn từ loại để có thể thống kê được chính xác nên không thể áp dụng được.

2.3.1.3. Phương pháp sử dụng luật tương tác

Sự nhập nhằng trong tiếng Việt gặp rất nhiều ở một ít các từ trong tự điển. Tuy nhiên tần số xuất hiện của chúng rất cao. Vì vậy, có thể dễ dàng phát hiện ra các luật đơn giản nhưng tần số xuất hiện lại rất cao.

2.3.2. Chiếu quan hệ

Các ngôn ngữ tuy có cấu trúc khác nhau, nhưng về cơ bản các mối quan hệ ngữ pháp là giống nhau.

2.3.2.1. Mô hình xác suất

Có nhiều mô hình tính theo xác suất [18] khác nhau nhưng phổ biến nhất có lẽ là mô hình n-grams. Theo như mô hình này, đầu tiên một bộ dữ liệu đã có được nhãn quan hệ đúng trên cả 2 ngôn ngữ sẽ được thống kê. Kế đó, dựa vào dữ liệu thống kê, người ta sẽ tính được xác suất xuất hiện của từng bộ quan hệ cho một cặp câu. Bộ quan hệ với xác suất xuất hiện lớn nhất sẽ được chọn.

2.3.2.2. Phương pháp DCA (Direct Correspondence Assumption)

Cho một cặp câu E và F là câu dịch của nhau tương ứng với hai cấu trúc câu $TreeE$ và $TreeF$. Nếu nốt x_E và y_E của cây $TreeE$ liên kết với nốt x_F và y_F của cây $TreeF$. Nếu nốt x_E và y_E có mối liên hệ cú pháp thì tương ứng nốt x_F và y_F cũng có mối liên hệ trực tiếp với nhau.

Vấn đề gặp phải đối với phương pháp DCA:

- Các nốt trong trong cây có thể bị lược bỏ.
- Nốt tương ứng trong câu có thể bị thay đổi vai trò cú pháp.

2.3.2.3. Các phương pháp khác

Mỗi phương pháp đều có ưu và khuyết điểm của nó. Do đó, trong thực tế người ta thường kết hợp nhiều phương pháp đơn giản lại với nhau để phát huy một sức mạnh tổng hợp.

Cuối cùng, các Heuristic để chỉnh sửa lại những sai sót mà các phương pháp trước không tránh được.

Chương 3: MÔ HÌNH THUẬT TOÁN

Như chúng tôi đã giới thiệu trong chương 2, các bước cơ bản để tiến hành công việc chiếu kết quả phân tích cú pháp bao gồm: đầu tiên là phân tích cú pháp cho ngôn ngữ nguồn, sau đó liên kết từ/ngữ, cuối cùng sử dụng kết quả này để chiếu sang ngôn ngữ đích. Trong chương trình này, chúng tôi sẽ lần lượt trình bày chi tiết theo các mô hình thuật toán theo từng bước đã nêu trên. Phần đầu chúng tôi trình bày mô hình thuật toán phân tích cú pháp quan hệ dựa trên nguyên tắc, phần hai chúng tôi trình bày mô hình thuật toán liên kết từ/ngữ dựa trên mô hình dịch máy thống kê, phần ba chúng tôi trình bày mô hình thuật toán của phương pháp chiếu theo lý thuyết DCA kết hợp với phương pháp thống kê và sử dụng luật tương tác.

3.1. Phân tích cú pháp dựa trên nguyên tắc

3.1.1. Khái quát

Từ thuở sơ khai của ngành xử lý ngôn ngữ tự nhiên, người ta đã nghĩ đến các phương pháp phân tích dựa trên luật dẫn (rule-based). Cũng bởi tính đơn giản, hiệu quả của nó mà qua thời gian, nó ngày càng phát triển mạnh hơn.

Một hệ thống dựa trên hệ luật dẫn bao gồm nhiều luật, trong đó mỗi luật được tổ chức dưới dạng:

Nếu điều kiện **thì** hành động (đưa đến một kết quả nào đó).

Đối với con người, luật dẫn gần gũi, dễ hiểu. Đối với máy, nó là một phép toán cơ bản, đơn giản, không thể thiếu được. Chính vì vậy mà khi mở một quyển sách trí tuệ nhân tạo nào ra, bạn đều có thể tìm được một chương nói về luật dẫn. Các hệ chuyên gia mà đầu não là động cơ suy diễn điều được xây dựng dựa trên hệ thống các luật.

Trong lĩnh vực xử lý ngôn ngữ tự nhiên, có rất nhiều phương pháp xem luật dẫn như là một phần tử hoạt động trung tâm, là bộ não điều khiển toàn bộ quá trình hoạt động . Có thể đưa ra một vài ví dụ : phương pháp phân tích cú pháp Top-Down, TBL, ...

Tuy nhiên, theo thời gian, cùng với sự phát triển của khoa học kỹ thuật, thì độ chính xác cần được tăng lên. Để đáp ứng được yêu cầu đó, bộ luật được xây dựng ngày càng công phu (ít nhất là đối với lĩnh vực xử lý ngôn ngữ tự nhiên). Do đó, cùng với tính chính xác, tính bao quát ngày càng tăng thì bộ luật ngày càng trở nên nặng nề, cồng kềnh, khó kiểm soát và giảm tính cơ động. Hơn nữa, một khi bộ luật đã được tạo ra một cách khá hoàn chỉnh, người ta mới có một cái nhìn tổng quát về nó. Người ta đã rút ra được rằng, các luật đã được tạo ra có những nét tương đồng. Các luật này có tuân theo một nguyên tắc nào hay không? Có thể rút ra được những quy tắc từ điều này hay không?

Một ví dụ : trong câu : “The ice-cream was eaten”. Để hiểu được câu này, rất ít hệ thống phân tích cú pháp nào có thể nhận ra được ice-cream là object của eat để mang một ý nghĩa là vật để ăn. Một luật điển hình có thể được tạo ra để nhận dạng trường hợp này như sau :

IF câu có dạng subject + be + verb-ed và không có object

THEN chuyển vai trò của subject như là object.

Kết quả thu được của luật IF THEN mã hoá từ trái qua phải của mẫu câu bị động trong tiếng Anh . Nó chỉ đúng cho tiếng Anh và trong mẫu câu bị động. Có thể nhận thấy ngay rằng, luật rút ra được là rất cụ thể, không mang tính tổng quát. Đó là lý do tại sao khi dùng luật phi ngữ cảnh để mô tả ngữ pháp cho một ngôn ngữ nào có, hàng ngàn luật được đưa ra vẫn là chưa đủ. Từ những nhận xét trên, người ta đã nghĩ đến một phương pháp có thể giải quyết được những khó khăn trên. Phương pháp sử dụng các nguyên tắc.

Có lẽ thật dễ dàng nói rằng nó không giống với các phương pháp phân tích cú pháp thông thường dựa rất nhiều luật (những luật riêng lẻ, cụ thể) hoặc dựa trên những hệ

thống phân tích cú pháp phi ngữ cảnh. Hệ thống phân tích cú pháp dựa trên luật cố gắng mô tả câu bằng những logic dựng sẵn bởi trật tự bề mặt nông của từ, những mẫu câu bị động, và tất cả những gì giống như vậy.

3.1.2. Ý tưởng cơ bản của phương pháp phân tích dựa trên nguyên tắc

Phương pháp phân tích cú pháp dựa trên nguyên tắc là một cách tiếp cận mới trong xử lý ngôn ngữ tự nhiên. Nó đã được phát triển bởi phòng thí nghiệm MIT từ cuối thập niên 80 đầu thập niên 90 [16]. Sử dụng phương pháp này, một tập hợp rất lớn các luật dùng để phân tích cú pháp câu sẽ được thay thế bằng một tập hợp nhỏ, cố định, các nguyên tắc thống nhất.

Cách tiếp cận này đã được áp dụng thực tế vào nhiều vấn đề khác nhau trong xử lý ngôn ngữ tự nhiên :

Áp dụng trên nhiều loại ngôn ngữ khác nhau : English, Spanish, German, và cả ngôn ngữ của thổ dân Úc (một ngôn ngữ mà trật tự từ rất tự do).

Dịch một câu đơn từ ngôn ngữ này sang ngôn ngữ khác.

Tối ưu hoá trong phân tích cú pháp để phân tích tuần tự hoặc song song trên nhiều máy tính.

Ý tưởng cơ bản là thay thế những luật nông cạn này bằng một tập hợp những nguyên tắc cơ bản: sâu sắc hơn, nhỏ hơn, dễ hình tượng, mang tính khái quát hoá cao hơn.

Như đã nói, hệ phân tích cú pháp dựa trên luật có các khuyết điểm :

Quá cứng nhắc : Hệ thống luật được xây dựng ra chỉ áp dụng cho một ngôn ngữ cụ thể. Áp dụng sang một ngôn ngữ khác, hệ thống này phải được xây dựng lại từ đầu. Một giải pháp mềm dẻo hơn được đặt ra?

Quá cụ thể : Chỉ nhìn từng khía cạnh của vấn đề mà không có tầm bao quát.

Dễ phá vỡ, khó giữ vững : Do nó có tính tra khớp các mẫu có sẵn, một mẫu câu mới không được xây dựng sẵn sẽ làm cho câu trở nên trật khớp đôi khi đảo lộn hoàn toàn. Một luật mới thêm vào có thể ảnh hưởng đến nhiều luật đã được xây dựng trước đó.

Quá công kênh : một hệ thống gồm hàng ngàn luật phải chăng là quá lớn. Bạn thử tưởng tượng, một câu rất được đưa vào, hệ thống sẽ không phân biệt câu dài ngắn mà toàn bộ các luật sẽ được áp dụng. Như vậy thì xảy ra trường hợp là có những luật mà tần số xuất hiện rất thấp nhưng lại được xét tới lặp đi lặp lại nhiều lần. Những luật này cũng không thể bỏ đi được bởi vì nếu bỏ đi, bộ luật sẽ không còn đầy đủ, không bao quát được các hiện tượng ngôn ngữ không thường xuyên xuất hiện. Kết quả là bộ luật của bạn sẽ rất công kênh, nặng nề.

Điều gì xảy ra khi hệ thống luật quá lớn? Bởi vì phương pháp dựa trên luật được định nghĩa cho các trường hợp đặc biệt, cụ thể, một hệ thống rất lớn các luật được xây dựng là điều cần thiết. Kết quả là, chỉ cần đưa một câu đơn giản vào, tất cả các luật sẽ lần lượt được đưa ra áp dụng. Điều này tốn chi phí rất nhiều do có những luật rất ít khi được áp dụng, nó chỉ được đưa vào để mô tả đầy đủ các trạng thái bề mặt của một câu trong ngôn ngữ đang phân tích.

Vậy phương pháp phân tích dựa trên nguyên tắc này sẽ giải quyết vấn đề này như thế nào : Phương pháp phân tích dựa trên nguyên tắc sẽ thay thế hệ thống luật khổng lồ này chỉ bằng một số rất ít các nguyên tắc thống nhất. Nó khắc phục những khuyết điểm nêu trên như sau :

Nó không quá cứng nhắc : Những nguyên tắc được xây dựng ra như là những khung sườn, do đó nó không quá phụ thuộc vào ngôn ngữ. Khi áp dụng những nguyên tắc này vào một ngôn ngữ cụ thể, ta chỉ cần thiết lập lại các tham số cần thiết cho phù hợp với ngôn ngữ chứ không cần phải thay thế tất cả. Các nguyên tắc luôn được giữ nguyên.

Nó mang tính khái quát rất cao, không đi vào nhưng vụn vặt tầm thường. Các nguyên tắc này là những tri thức mang tính thống nhất cho các ngôn ngữ.

Nó không cồng kềnh và cũng không dễ bị phá vỡ và mang tính linh hoạt do nó nhỏ gọn và không chi tiết.

3.1.3. Một số ít những nguyên tắc thay thế cho rất nhiều luật

3.1.3.1. Những thành phần cơ bản

Có bao giờ bạn nghĩ đến là mình phải học thuộc tất cả các phân tử trong hoá học hay không? Nếu phải ghi nhớ hết tất cả thì đó quả là một điều hết sức kinh khủng bởi tính đa dạng gần như là vô tận của các chất. Nhưng thử nhìn lại, tất cả các phân tử này đều được xây dựng từ một số lượng hữu hạn các nguyên tử với nhiều cách thức kết hợp khác nhau và đều tuân theo những quy tắc nào đó, đã tạo nên sự đa dạng đó.

Trong toán học hay logic học cũng vậy, chỉ từ một số tiên đề, một hệ thống toán học khổng lồ được xây dựng nên.

Dựa trên ý tưởng này, phương pháp phân tích dựa trên nguyên tắc xem các nguyên tắc và ngữ nghĩa của từ (kết hợp lại) như là các nguyên tử trong hoá học hay các tiên đề trong toán học). Bằng cách liên kết vài chục nguyên tử chúng ta có thể xây dựng được biết bao nhiêu là phân tử (luật và nghĩa của từ) thay vì chúng ta phải liệt kê tất cả các phân tử này. Trong lĩnh vực ngôn ngữ, chúng ta có thể thay thế những hiện tượng xảy ra ở bề mặt của câu bằng những chuỗi suy diễn dài hơn (so với luật chỉ có if then) bằng một số tiên đề cơ bản của ngôn ngữ. Chú ý một điều là điều này sẽ thay thế $n_1 \times n_2 \times \dots$ luật bằng $n_1 + n_2 + \dots$ phần độc lập. Bằng việc kết hợp các nguyên tắc này, với mỗi nguyên tắc có 2, 3 dạng trạng thái thì chúng ta đã mã hoá được hàng ngàn luật.

3.1.3.2. Tham số

Bây giờ ta hãy xét đến tính cơ động của từng phương pháp. Đối với phương pháp dựa trên hệ luật, mỗi khi có sự thay đổi về môi trường làm việc, thay đổi về đối tượng

tác động (như thay đổi ngôn ngữ trong xử lý ngôn ngữ tự nhiên) thì hệ thống luật cần phải được xây dựng lại.

Ngược lại, đối với phương pháp dựa trên nguyên tắc, cũng bởi vì nó là những quy tắc có tính khái quát nên nó có tính cơ động rất cao. Khi có sự thay đổi thì những nguyên tắc trên hầu như không thay đổi mà chỉ cần thay đổi những tham số cho các nguyên tắc này.

Một ví dụ về nguyên tắc Xbar .

Nguyên tắc này phát quy định về trật tự của từ chính (head-word) trong một ngữ. Trong tiếng Anh, một ngữ động từ thường có head-word nằm ở vị trí đầu tiên, một giới từ sẽ nằm ở vị trí đầu tiên trong ngữ giới từ. Như vậy, khi phân tích câu tiếng Anh, tham số về trật tự trong nguyên tắc này được thiết lập là function-argument. Ngược lại, đối với tiếng Nhật và tiếng Đức, tham số này sẽ được thiết lập là argument-function.

Như vậy, nội dung của nguyên tắc không thay đổi. Thay thế những tham số khác nhau cho các nguyên tắc này, chúng ta có thể định nghĩa được những biến đổi theo từng địa phương của ngôn ngữ và ngay cả trên các ngôn ngữ khác.

3.1.4. Câu hỏi đặt ra

Theo như đã trình bày ở trên, phương pháp phân tích cú pháp dựa trên nguyên tắc giải quyết được rất nhiều khuyết điểm của phương pháp sử dụng luật. Tuy nhiên, phương pháp tiếp cận mới này sẽ đặt ra rất nhiều câu hỏi.

Phương pháp này có thể dùng để phân tích ngôn ngữ được không? Từ một tập hợp rất nhỏ các nguyên tắc cố định, chúng có thể mô tả được đầy đủ những biến đổi muôn hình vạn trạng của ngôn ngữ (mà theo lý thuyết là không chỉ cho một ngôn ngữ).

Liệu rằng có thể xây dựng một bộ phân tích cú pháp dựa trên những nguyên tắc này thay vì sử dụng luật.

Liệu rằng nó có hiệu quả hay không?

Tiền đề ngôn ngữ này được xây dựng như thế nào?

Từ những tiền đề của ngôn ngữ này sẽ được kết hợp như thế nào?

3.1.5. Các nguyên tắc

3.1.5.1. Thuyết Xbar (\bar{X} theory)

Mô tả dạng thức của các ngữ hay hình dạng của cây : (function-argument hay ngược lại). Ý chính của thuyết X-theory là:

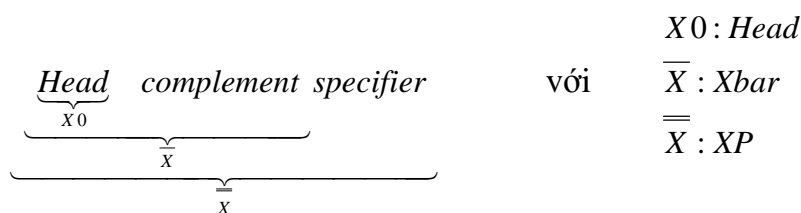
• Mỗi một thành phần ngữ pháp (thành phần không kết thúc) có một phần tử kết thúc trung tâm gọi là head (X_0)

• Phần tử head này kết hợp với complement của nó để tạo thành Xbar, Xbar lại kết hợp với thành phần specifier để tạo thành XP. Với X ở đây là V, N, ..

Phát biểu thứ nhất mô tả mối ràng buộc: khi hai phần tử thì khi hai phần tử có mối quan hệ với nhau thì phải có một phần tử sẽ đóng vai trò trung tâm và là head.

Phát biểu thứ 2 được thể hiện trong cấu trúc của mạng cú pháp.

Có 2 dạng cây căn bản : function-argument và argument-function. Trong tiếng Anh (verb bắt đầu một verb phrase, preposition bắt đầu preposition phrase) nên cây trong tiếng Anh có dạng function-argument.



Hình 3.1. Cấu trúc của Xbar

Ý nghĩa : đây là nền tảng để xây dựng các cấu trúc lớn hơn từ các cấu trúc đơn giản ban đầu.

3.1.5.2. Tiêu chuẩn Theta (Theta Criterion)

Mỗi từ trong câu, dựa vào thuộc tính của chính bản thân nó được thành lập thành một trạm. Thuyết này mô tả về những tham số có thể hoặc là buộc phải có đối với một từ và cách thức kết hợp giữa chúng. Các trạm (item) chỉ được kết hợp lại với nhau khi nó một số thuộc tính (features) nào đó của 2 trạm này là có thể kết hợp được. Chẳng hạn khi ta xét đến một động từ nào đó thì xung quanh nó có một số khoảng trống dành sẵn để điền một số từ thích hợp vào để bổ khuyết cho ý nghĩa của động từ đó : “**who did what to whom**”.

Một ví dụ cụ thể như động từ “eat” chẳng hạn, nó chỉ có thể đi với những thứ có thể ăn được mà thôi trong khi từ “put” lại đi với một nơi hoặc một vị trí nào đó...

Ví dụ : câu “*It loves Mary*”

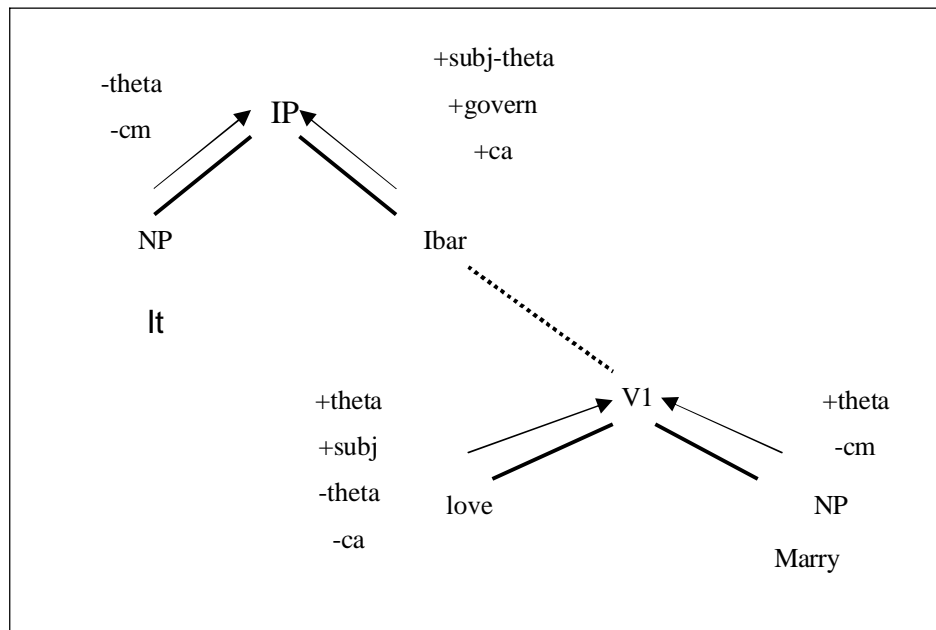
it : có thuộc tính –theta

loves : +theta và +subj-theta

Marry : +theta

do đó : *loves* và *Marry* có thể kết hợp được.

“Ibar” có thuộc tính +subj-theta trong khi “it” có thuộc tính -theta => không thể kết hợp được.



Hình 3.2. Minh hoạ cho câu “It loves Mary” theo tiêu chuẩn Theta

Ý nghĩa: Tạo nên rào cản ngăn cản các sự kết hợp không hợp lệ (Ví dụ: her không thể làm chủ ngữ của câu vì nó không thể tạo nên sự liên kết với động từ của câu).

3.1.5.3. Bộ lọc vai (Case-Filter)

Trước khi phát biểu nguyên tắc này, ta cần quan tâm đến một khái niệm gọi là vai(case) trong ngôn ngữ. Vai thực ra là vai trò của danh từ trong câu.

Một số vai của danh từ như:

- Nominative case: chủ thể của động từ.
- Accusative case: túc từ trực tiếp.
- Oblique case: túc từ bổ nghĩa cho giới từ.

• Phát biểu nguyên tắc

Mỗi noun phrase phải được gán vai (case-assigned) : trong đó A gán vai (case-assign) cho B nếu A nắm vai trò thống trị (governor) B và A phải là một gán vai viên (case-assigner).

Ta sẽ lần lượt làm rõ từng mục từ trong phát biểu trên.

Thế nào là quan hệ thống trị (government)? : A thống trị B nếu A là phần tử ở mức thấp nhất trong mạng ngữ pháp có thể làm chủ B.

Vậy thế nào là làm chủ (m-command)? : Một head daughter của một item biểu diễn hạng mục lớn nhất sẽ làm chủ tất cả các item còn lại ngoại trừ bản thân chính nó.

Gán vai viên là gì? Thực ra nó là một thuộc tính của một mục từ. Thuộc tính này gọi là “ca”. Nếu mục từ nào có thuộc tính này thì được đánh dấu là +ca, ngược lại sẽ có thuộc tính là –ca.

Ü Case-assigner (P, active V, I) có thuộc tính +ca (case-assigner) Not a case-assigned (N, A, passive V) có thuộc tính –ca.

Tuy về lý thuyết, thuyết lọc vai này là hoàn toàn hợp lý. Tuy nhiên, ta phải áp dụng nó vào quá trình phân tích cú pháp là một vấn đề quan trọng. Để loại bỏ những vi phạm của nguyên tắc này, người ta định nghĩa ra các phần tử gọi là rào cản hay nốt chặn (barrier). Các nốt chặn này sẽ “trấn giữ” trên các cung nối liền các nốt trong mạng ngữ pháp (sẽ được trình bày trong phần sau).

Ü Các nốt chặn không cho phép các trạm chưa được gán case đi qua.

Khi có sự kết hợp giữa +govern +ca và –cm thì –cm sẽ bị triệt tiêu.

Ý nghĩa của nguyên tắc này : mỗi một danh từ trong câu phải có một vai trò nhất định.

Bây giờ ta thử xét một vài ví dụ:

Ví dụ 1 : . Có hai câu như sau:

It is likely that John will win.

It is likely that John to win.

Trong câu thứ nhất, John là chủ thể của động từ “win” do đó nó sẽ được gán vai nominative.

Trong câu thứ hai, John không thể được gán vai bởi bất kì một động từ nào thích hợp, do đó câu được xác định là sai ngữ pháp.

Xác định được đại từ thay thế cho danh từ nào trong ngữ cảnh nhất định của câu.

Ví dụ 2: Xét 2 câu sau:

John thinks that he likes ice-cream.

He thinks that John likes ice-cream.

Trong câu thứ nhất, John và he có thể đề cập đến cùng một người trong khi câu thứ 2 thì không thể được.

3.1.5.4. Thuyết kết hợp(Binding Theory)

Xác định được đại từ thay thế cho danh từ nào trong ngữ cảnh nhất định của câu.

Ví dụ : Xét 2 câu sau:

John thinks that he likes ice-cream.

He thinks that John likes ice-cream.

Trong câu thứ nhất, John và he có thể đề cập đến cùng một người trong khi câu thứ 2 thì không thể được.

3.1.5.5. Thuyết về tính cục bộ và trường rộng

Đây là nguyên tắc bảo đảm sự phá vỡ trật tự trong một ngôn ngữ. Thật ra, không có một ngôn ngữ tự nhiên nào có một quy luật rõ ràng, chính xác như ngôn ngữ nhân tạo. Cho nên, nếu chỉ áp dụng các nguyên tắc vừa được trình bày như trên thì chỉ có thể dùng để phân tích những câu văn thật chuẩn, đúng công thức mà thôi. Tuy nhiên, trong ngôn ngữ tự nhiên, việc tính lược một số thành phần hay một vài thành phần đã được nhắc tới ở đâu đó và trong câu hiện tại nó được hiểu ngầm thì rất nhiều.

Chính vì vậy, nguyên tắc này bảo đảm những xác định nơi nào một danh ngữ tiềm ẩn có thể xuất hiện trong câu. Một danh ngữ tiềm ẩn sẽ không được phát âm nhưng nó giữ một vai trò nhất định trong câu và vì vậy nó cần thiết để có thể hiểu được câu.

Ví dụ :

John wants to like ice-cream.

[John wants [to like ice-cream]]

[John wants [to e like ice-cream]]

Câu trên có một danh ngữ tiềm ẩn (được đề cập đến như là node e), nó có vai trò như chủ ngữ của mệnh đề “like ice-cream”. Giống như một đại từ, nó đề cập đến John.

Các danh ngữ tiềm ẩn không thể xuất hiện quá xa từ mà nó đề cập đến và chỉ trong một ngữ cảnh nhất định (Empty category principle).

Ví dụ:

John seems it is certain e to like ice-cream.

John was wanted to e like ice-cream.

Câu thứ nhất ngăn cản sự đồng nhất giữa e và John (không lân cận) trong khi câu thứ 2 thì cho phép.

3.1.5.6. Thuyết dịch chuyển

Đây là nguyên tắc phức tạp nhất trong các nguyên tắc. Nó xuất phát từ hiện tượng : các từ trong câu không nhất định phải xuất hiện theo một trình tự cố định nào đó mà nó có thể hoán đổi vị trí trong câu (câu bị động , wh-question, “Ice-cream, I like” ... là những ví dụ).

Có 2 loại dịch chuyển : danh từ và wh-word.

Một movement không thể di chuyển qua nhiều hơn một rào cản (đối với tiếng Anh vì thực sự con số này phụ thuộc vào ngôn ngữ). Nhìn trên mạng phân tích cú pháp, rào cản (barrier) là các hình chữ nhật nhỏ hình vuông màu đen nằm trên các cung nối.

3.1.6. Trật tự kết hợp các nguyên tắc

Với cùng một số nguyên tố, kết hợp với nhau theo những cách khác nhau sẽ tạo thành các phân tử khác nhau. Điều đó chứng minh rằng trật tự kết hợp đóng vai trò rất quan trọng trong việc áp dụng các nguyên tắc này. Không giống như những luật trong TBL, điểm của các luật là có thể tính được. Các nguyên tắc được áp dụng ở đây không thể lượng giá được mà phải dùng các heuristic. Thực tế đã chứng minh rằng không có

một trật tự nào là tối ưu nhất. Mỗi heuristic chiếm ưu thế trong một số trường hợp nhưng lại sai trong một số trường hợp khác. Sau đây là một số heuristic mà thực tế.

3.1.6.1. Dự đoán lỗi trước

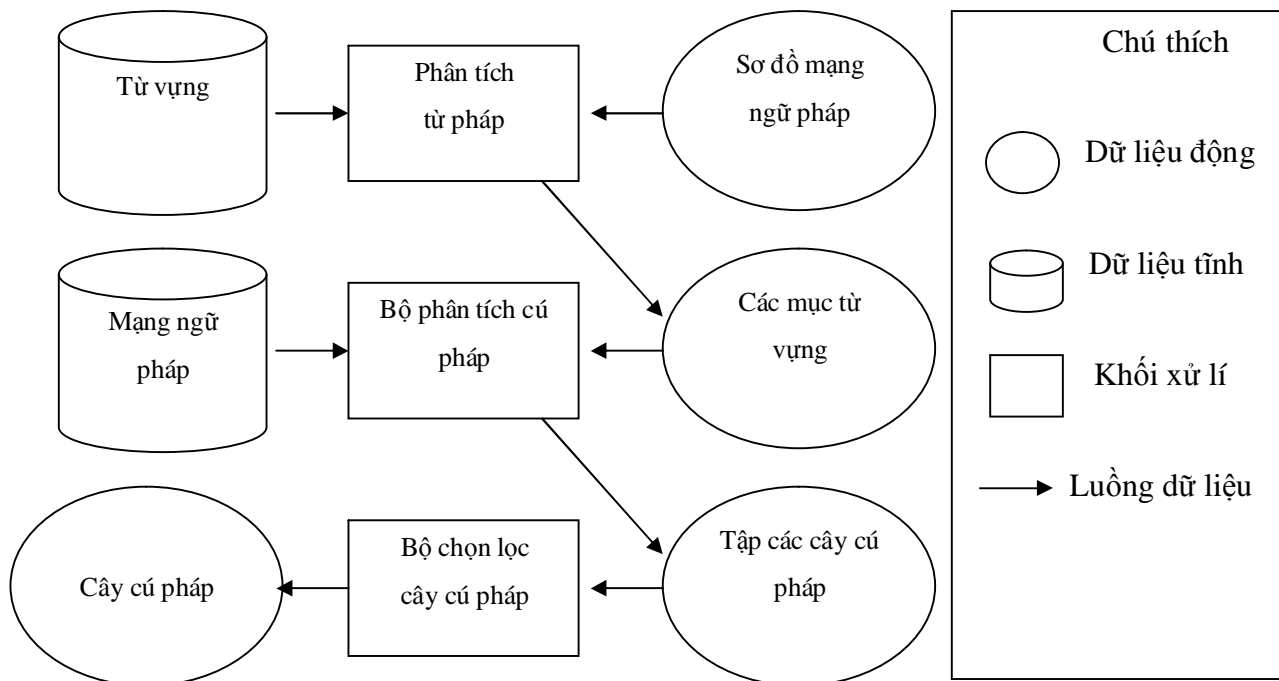
Theo phương pháp này, những thao tác không cần thiết sẽ được ưu tiên loại đi từ đầu. Bởi vì nó không biết cấu trúc một câu theo một dạng là đúng hay sai nên nó sẽ giả sử rằng đây là câu sai cú pháp, rồi sau đó bằng những thông tin của câu, nó sẽ tìm ra sự hợp lý của cấu trúc câu này. Còn nếu như không tìm ra được sự hợp lý này thì cấu trúc đang xét được xem như sai cú pháp và sẽ được bỏ qua. Trong bước kiểm tra đầu tiên này, những hành động nào có chi phí nhỏ sẽ được ưu tiên xét trước.

Cấu trúc	Nguyên tắc
Trace	Empty category và case condition on traces
Intransitive	Case filter
Passive	Theta Criterion Case filter
Non-argument	Theta Criterion
+anaphoric	Binding Theory Principle A
+Pronominal	Binding Theory Principle B

Bảng 3.1. Mô hình kết hợp các nguyên tắc theo phương pháp dự đoán lỗi trước

3.1.6.2. Mô hình động

Nhiều kinh nghiệm đã chứng minh rằng mô hình trật tự tĩnh là không tối ưu. Vì kiến trúc máy tính là phát triển rất nhiều cho nên tùy thuộc vào một nhiệm vụ là phải thực hiện tuần tự hay song song mà tùy tình huống mà áp dụng. Còn một điều nữa là trong việc sắp xếp trật tự các nguyên tắc này, nếu tối ưu hoá về tốc độ và tài nguyên, độ chính xác có thể giảm và ngược lại.

3.1.7. Các bước phân tích cú pháp

Hình 3.3. Sơ đồ hoạt động tổng thể của mô hình phân tích cú pháp dựa trên nguyên tắc

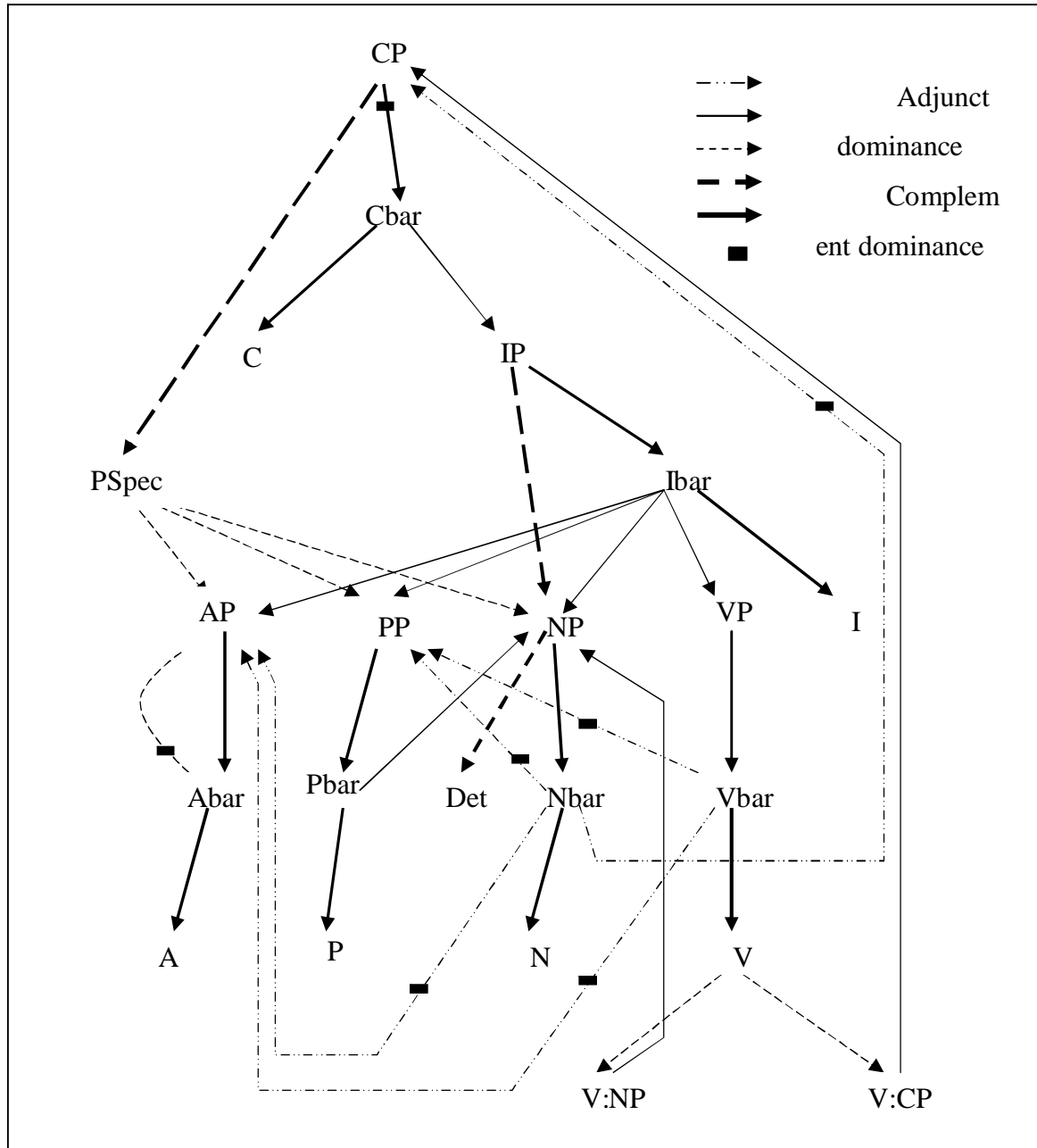
3.1.7.1. Phân tích từ vựng

Từ cần phân tích chỉ đơn giản được tra trong từ điển, nếu có nhập nhằng thì tất cả các nhãn có thể sẽ được chọn. Kết quả của bước này là một từ sẽ được gán nhãn từ loại (POS) cùng với một tập các thuộc tính khác.

3.1.7.2. Phân tích và tìm ra các cây cú pháp thích hợp

Sử dụng thuật toán phân tích cú pháp bằng thông điệp. Đây là thuật toán mở rộng của thuật toán phân tích cú pháp phi ngữ cảnh [17].

Trong phương pháp phân tích cú pháp này, văn phạm được mã hoá thành một mạng gồm nhiều nốt được nối kết lại với nhau. Mỗi node trong mạng biểu diễn một nhãn cú pháp (như N, NBar, NP, CP...). Các đường nối giữa các node trong mạng biểu diễn mối quan hệ giữa các node đó. Có 2 loại quan hệ : quan hệ chính phụ và quan hệ đồng đẳng.



Hình 3.4. Sơ đồ mạng ngữ pháp

Mỗi nốt trong mạng là một thực thể tính toán. Nó giao tiếp với các nốt khác bằng cách gọi đi các thông điệp theo ngược chiều mũi tên (tức là đi từ chi tiết đến phức hợp). Mỗi thông điệp chứa các item. Item là một bộ ba có cấu trúc như sau:

<surface-string, attribute-value, sources>, trong đó:

surface-string: là một khoảng số nguyên từ i đến j biểu diễn dãy các từ trong câu nhập vào.

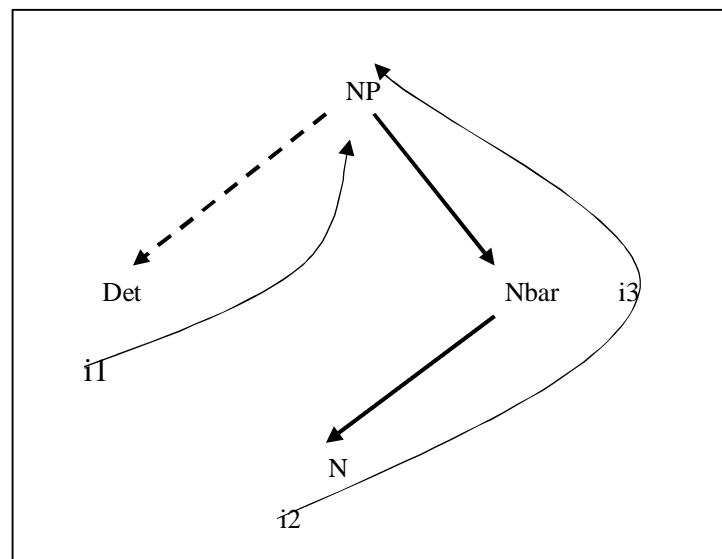
attribute-value: chỉ định thuộc tính cú pháp của nốt gốc biểu diễn bởi item source.

sources: thành phần này là một tập các item của các thành phần con trực tiếp của nó.

Vì thế, truy lần theo vết có thể hoàn chỉnh được cấu trúc cần tìm.

Vị trí của item trong mạng thể hiện được nhãn cú pháp của cấu trúc.

Ví dụ: Trong câu [NP the ice-cream was eaten] biểu diễn bằng một nốt i_4 trong mạng : $\langle [0,1], ((cat -plu (nfrom norm) -cm +theta), \{i_1, i_3\}) \rangle$



Hình 3.5. Biểu diễn một câu trong mạng

Một item biểu diễn một nốt gốc của cấu trúc mà nó chứa đủ thông tin không nốt trung gian nào đạt được.

Tiến trình gởi đi thông điệp bắt đầu bằng việc gởi một thông điệp bắt đầu tới một nốt kết thúc (N,P,...). Item khởi đầu biểu diễn một từ trong câu. Giá trị thuộc tính của item có được từ thông tin từ loại.

Trong trường hợp nhập nhằng ngữ nghĩa, mỗi trường hợp có thể biểu diễn bằng một item.

Khi một nốt nhận được một item, nó sẽ cố kết hợp item với một item khác tạo thành một item mới. Đây là tiền đề cho việc xây dựng lên những cấu trúc lớn hơn.

Hai items : $\langle [i1, j1], A1, S1 \rangle$ và $\langle [i2, j2], A2, S2 \rangle$ có thể được kết hợp với nhau nếu nó thỏa mãn 3 điều kiện sau đây :

Surface-string kế tiếp nhau ($i2 = j1 + 1$).

Thuộc tính A1 và A2 có thể thống nhất lại.

Hai nguồn S1 và S2 không giao nhau $S1 \cap S2 = \emptyset$

Kết quả tạo thành một item mới $\langle [i1, j2], \text{unify}(A1, A2), S1 \cup S2 \rangle$

Các lý thuyết về nguyên tắc (GB) được thực hiện như là một tập các ràng buộc phải được thỏa mãn suốt trong quá trình truyền và kết hợp các item. Những ràng buộc này có thể kết hợp với các nốt hoặc các mối liên kết trên mạng. Ví dụ : mối liên kết từ VP:NP chỉ cho phép NP với case acc đi qua, vì thế mà những danh từ có case là nonactive (như he, she) không thể đi qua.

Theo mặc định, các thuộc tính của mỗi item sẽ gắn kết cùng item khi đi qua một nốt khác theo một đường link. Tuy nhiên một số đường nối sẽ cấm sự lưu thông của một số thuộc tính nhất định.

Câu được parse hoàn chỉnh nếu item được tìm thấy tại nốt IP hoặc CP mà bề mặt chứa toàn bộ câu nhập vào.

Để dễ hình dung, ta thử làm một ví dụ sau :

Phân tích câu tiếng Anh nhập vào : “The ice-cream was eaten”.

Đầu tiên item i1 được tạo ra bằng cách tra từ điển từ vựng cho từ “the” và gởi thông điệp đầu tiên đến nốt Det. Tại đây, thông điệp sẽ được tiếp tục truyền bản sao của i1 đến nốt NP.

$i1 = \langle [0, 0], ((cat\ d)), \{\} \rangle$

Tiếp theo, giống như i1, ice-cream sẽ được tra từ điển và sẽ tạo ra được item i2, i2 sẽ được gửi đến N rồi lại được tiếp chuyển đến Nbar.

$i2 = \langle [1, 1], ((cat\ n) - plu (n\ from\ norm) + theta, \{\} \rangle$

$$i3 = \langle [1,1], ((cat\ n) - plu\ (nfrom\ norm) + theta, \{i2\}) \rangle$$

Khi NP nhận được $i3$ từ $Nbar$, $i3$ sẽ được kết hợp với $i1$ để tạo thành một item mới là $i4$.

$$i4 = \langle [0,1], ((cat\ n) - plu\ (nfrom\ norm) - cm + theta, \{i1, i3\}) \rangle$$

Để ý rằng $i4$ có thêm thuộc tính $-cm$, điều đó có nghĩa là nó cần được gán vai (case marked). Sau đó $i4$ sẽ được gửi tới nốt NP.

Từ tiếp theo sẽ được tra từ điển “was” tạo thành item $i5$. $i5$ được gửi tới I rồi gửi tới $Ibar$: $i6$

$$i5 = \langle [2,2], ((cat\ i) - plu\ (per\ 1\ 3)\ (cform\ fin) + be + ca + govern\ (tense\ past)), \{\}\rangle$$

$$i6 = \langle [2,2], ((cat\ i) - plu\ (per\ 1\ 3)\ (cform\ fin) + be + ca + govern\ (tense\ past)), \{i5\}\rangle$$

Từ “eaten” sẽ được tra từ điển. Có 2 khả năng có thể xảy ra: một là ta xem nó như là động từ ở dạng quá khứ phân từ, hai là thể bị động của động từ eat. Giả sử ta xét trường hợp thứ 2. Lúc này item $i7$ sẽ được hình thành.

$$i7 = \langle [3.3], ((cat\ v) + pas\ \{\}) \rangle$$

Từ vựng thuộc về mục từ $V:NP$, trong mục từ này, NP được xem như thành phần bổ nghĩa. Vì vậy, $i7$ sẽ được gửi tới $V:NP$.

Bởi vì $i7$ có thuộc tính $+pas$, một sự dịch chuyển NP (của mục từ $V:NP$) được hình thành.

$$i8 = \langle [3.3], ((cat\ v) + pas + nppg - npcarrrier\ (-npatts\ NNROM)), \{i7\}\rangle$$

$$i9 = \langle [3.3], ((cat\ v) + pas + nppg - npcarrrier\ (-npatts\ NNROM)), \{i8\}\rangle$$

Để ý rằng thuộc tính các thuộc tính $+nppg - npcarrrier$ và $-npatts$ được thêm vào như là những ràng buộc của những nguyên tắc phải tuân theo.

Khi $Ibar$ nhận được $i10$, qua bước trung chuyển đến VP, nó sẽ kết hợp $i10$ và $i6$ để tạo thành $i11$

$$i10 = \langle [3.3], ((cat\ v) + pas + nppg - npcarrrier\ (-npatts\ NNROM)), \{i9\}\rangle$$

$i11 = \langle [2.3], ((cat\ v) + pas + nppg - npcarrrier (-npatts\ NNROM)), (per\ 1\ 3) (cform\ fin) + ca + govern\ (tense\ past)) \{i6, i10\} \rangle$

Khi IP nhận được $i11$, nó sẽ kết hợp với $i4$ từ nốt NP để tạo thành $i12$.

$i12 = \langle [0.3], ((cat\ i) + pas\ (per\ 1\ 3) (cform\ fin) + ca + govern\ (tense\ past)) \{i4, i11\} \rangle$

3.1.7.3. Chọn cây cú pháp thích hợp nhất

Sau khi qua bước phân tích, sẽ có rất nhiều cây khác nhau sẽ được thoả mãn. Để lọc ra được cây tốt nhất, người ta dùng phương pháp chấm điểm. Bộ phân tích sẽ định nghĩa một trọng số cho mỗi cây. Một trọng số sẽ được gán với mỗi từ và mỗi liên kết trong cây cú pháp. Như vậy thì trọng số của cây sẽ là tổng các trọng số của các mối liên kết và các từ tại các nốt lá trong câu.

3.1.7.4. Trọng số

Việc chọn lựa cây này được tổ chức theo cách mà cây có trọng số nhỏ nhất sẽ được chọn ra trước tiên. Sau đó nó sử dụng trọng số này và một con số đã được xác định trước gọi là BIGWEIGHT để chọn cây thích hợp.

3.1.7.5. Chọn cây

Chỉ những cây mà trọng số của nó nhỏ hơn ($\text{minimum weight} + \text{BIGWEIGHT}/2$) được giữ lại.

Trọng số của liên kết và ý niệm được định nghĩa như sau:

Liên kết từ $Xbar$ tới một định ngữ (hoặc bổ ngữ) có trọng số BIGWEIGHT, các liên kết khác có trọng số là 1.

Từ có thể có thuộc tính rare và có thể nhận giá trị : very, very-very. Nếu từ có thuộc tính giá trị (rare,very) thì trọng số của nó là BIGWEIGHT, còn nếu từ có thuộc tính giá trị (rate,very-very) thì trọng số của nó là BIGWEIGHT x 2. Các từ không có những thuộc tính này sẽ có trọng số là 0. (Ghi chú : thuộc tính rate của từ dùng để gán ghép ý nghĩa mức độ thường xuyên vào một từ).

Cây có trọng số càng nhỏ là cây có tần số xuất hiện càng cao.

3.2. Mô hình liên kết từ/ngữ trong song ngữ Anh-Việt

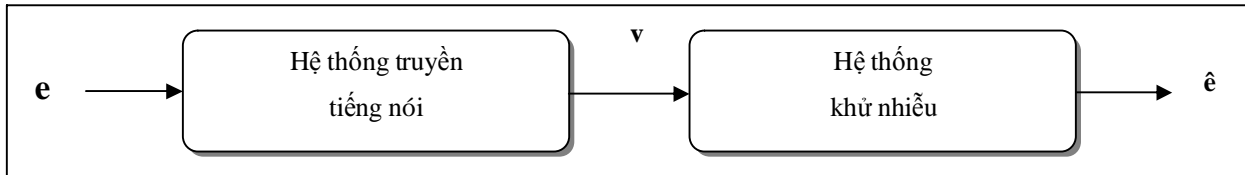
Mô hình liên kết từ/ngữ của chúng tôi dựa vào cách tiếp cận dịch máy thống kê [1] [2] [5]. Trong phần này chúng tôi sẽ trình bày chi tiết về mô hình dịch máy thống kê và ứng dụng nó trong việc liên kết từ trong song ngữ Anh-Việt, chúng tôi không trình bày chi tiết về phần lý thuyết cơ bản của xác suất thống kê, độc giả có thể tham khảo tới [9] [10].

3.2.1. Giới thiệu mô hình dịch máy thống kê

Vào năm 1949, Warren Weaver đã đề nghị ứng dụng kỹ thuật thống kê và giải mã vào trong việc xử lý ngôn ngữ tự nhiên. Những kỹ thuật này được biết đến từ lý thuyết truyền thông để giải quyết vấn đề sử dụng máy tính trong việc chuyển từ văn bản từ một ngôn ngữ này sang một ngôn ngữ khác. Song sự cố gắng trong lúc đó đã bị hủy bỏ một lý do triết học và lý thuyết suông. Vào thời điểm đó thì hầu hết các cách tiếp cận đều phải chịu sự bất hạnh về sự thiếu thốn của công nghệ máy tính. Ngày nay, với sự phát triển như vũ bão của công nghệ kỹ thuật số cho phần cứng đã kéo theo sự phát triển về các cách tiếp cận trong việc dịch máy. Trong những năm cuối của thế kỷ 20, cách tiếp cận thống kê đã được ứng dụng rộng rãi trong các khoa học lĩnh vực, trong đó có lĩnh vực dịch máy.

Mãi đến năm 1993, Brown et al. đã đưa ra mô hình thống kê cho việc dịch máy. Ông đã chứng minh bằng lý thuyết xác suất thống kê [1] cho một hệ huấn luyện và dịch máy hoàn chỉnh. Trong đó ông mô tả năm mô hình thống kê rất chi tiết. Các mô hình này dùng để huấn luyện và cho ra những thông số thống kê dựa trên dữ liệu song ngữ. Trong năm mô hình thống kê đó, chúng tôi đã cài đặt và chạy thử nghiệm ba mô hình đầu cho việc liên kết từ/ngữ của chúng tôi.

Mô hình dịch máy thống kê là mô hình thực dụng, nó được mô phỏng theo hệ thống khử nhiễu tiếng nói (noisy-channel)[2][5]. Hãy hình dung có một hệ thống truyền tiếng nói và truyền một câu tiếng Anh e , nhưng trong quá trình truyền dữ liệu thì câu e bị nhiễu và trở thành câu tiếng Việt v . Như vậy, nhiệm vụ của hệ thống khử nhiễu tiếng nói là phải phục hồi từ câu tiếng Việt v trở về câu tiếng Anh e như ban đầu.



Hình 3.6. Hệ thống truyền và khử nhiễu tiếng nói

Nhưng trong thực tế thì không thể phục hồi hoàn toàn về câu tiếng Anh e được, mà chỉ có thể phục hồi về câu tiếng Anh \hat{e} gần đúng với câu e ban đầu thôi. Toàn bộ hệ thống trong Hình 1. được mô tả bằng biểu thức sau:

$$\hat{e} = \arg \max_e P(e | v) \quad (3.1)$$

với \hat{e} là câu tiếng Anh được phục hồi

e là câu tiếng Anh được truyền

v là câu tiếng Việt bị nhiễu được tạo thành từ câu e

$P(e | v)$ là hàm phân phối xác suất nhiễu

$\arg \max$ là hàm tìm câu e sao cho xác suất nhiễu $P(e | v)$ là lớn nhất

Vấn đề của hệ thống khử nhiễu là làm sao để có được xác suất nhiễu $P(e | v)$. Để có được xác suất này chúng ta phải quan sát hệ thống truyền tiếng nói truyền nhiều câu e khác nhau và thống kê xác suất nhiễu để trở thành câu v . Như vậy, chúng ta phải duyệt qua tất cả các câu tiếng Anh e có thể có, và điều này thì trong thực tế thì không thể thực hiện được vì các câu tiếng Anh e có thể có là vô hạn.

Quay lại vấn đề dịch ngôn ngữ từ một ngôn ngữ này sang một ngôn ngữ khác: giả sử có một người phiên dịch từ tiếng Việt sang tiếng Anh, khi anh ta nhận được một câu v

từ một người Việt, anh ta phải nhẩm trong đầu mình những câu e có thể dịch và anh ta phải tìm ra một câu e sát nghĩa với câu v nhất để làm sao cho người bản địa Anh có thể hiểu được. Đối với con người thì chúng ta thấy vấn đề này không khó, vì chúng ta có thể giới hạn lại các câu e trong một tập hợp nhỏ mà trong đầu chúng ta nhẩm được, nhưng ngược lại đối với máy tính thì không nhẩm được. Do đó, máy tính phải duyệt qua tất cả các câu e có thể có trong thế giới thực.

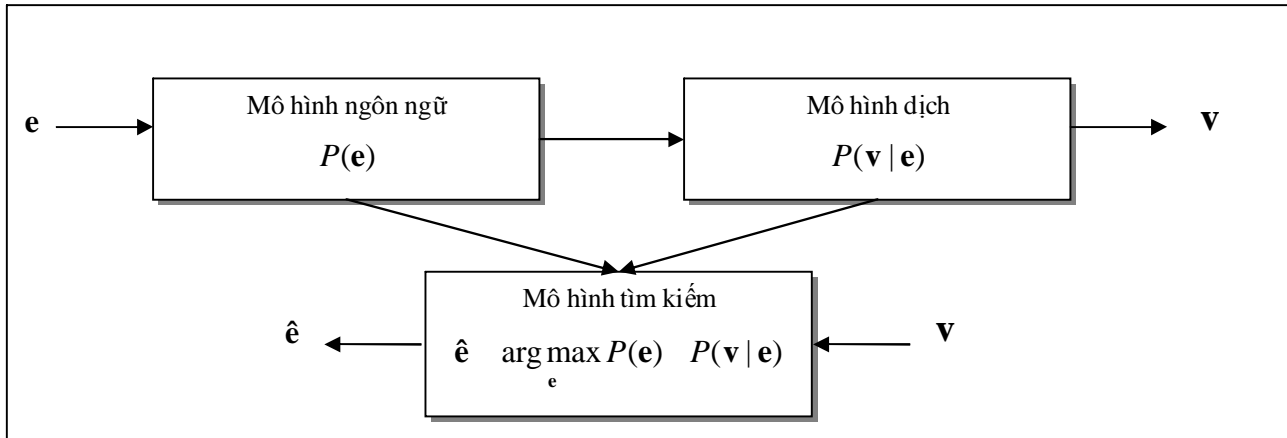
Một lần nữa chúng ta thấy được vấn đề của hệ thống khử nhiễu cũng như trong hệ thống dịch máy là phải tính được xác suất nhiễu (đối với hệ thống truyền tiếng nói) hay xác suất dịch (đối với hệ thống dịch máy). Nhưng thật may mắn là chúng ta có thể giải quyết được vấn đề này thông qua việc sử dụng định lý Bayes để tính xác suất $P(e|v)$ như sau:

$$P(e|v) = \frac{P(e) * P(v|e)}{P(v)} \quad (3.2)$$

Vì mẫu thức trong vế phải của biểu thức (3.2) độc lập so với e nên chúng ta có thể tìm một câu tiếng Anh \hat{e} gần đúng nhất bằng cách làm cho tích của $P(e) * P(v|e)$ là lớn nhất có thể. Như vậy, chúng tôi có thể viết lại biểu thức (3.1) lại như sau

$$\hat{e} = \arg \max_e P(e) * P(v|e) \quad (3.3)$$

Biểu thức (3.3) là biểu thức cơ sở cho dịch máy thống kê. Qua biểu thức (3.3) thì chúng ta nhận thấy rằng công việc trở nên đơn giản hơn và nó tổng quát hoá được ba vấn đề cho việc xây dựng một hệ thống dịch máy thống kê hoàn chỉnh: vấn đề ước lượng xác suất mô hình ngôn ngữ $P(e)$ [1] [11], vấn đề ước lượng xác suất mô hình dịch $P(v|e)$ [1] [2] [5], và vấn đề cuối cùng là vấn đề tìm kiếm một câu tiếng Anh e tốt nhất (có nghĩa là phải tìm tích của $P(e) * P(v|e)$ là lớn nhất). Hình dưới đây thể hiện một hệ dịch hoàn chỉnh với ba mô hình.

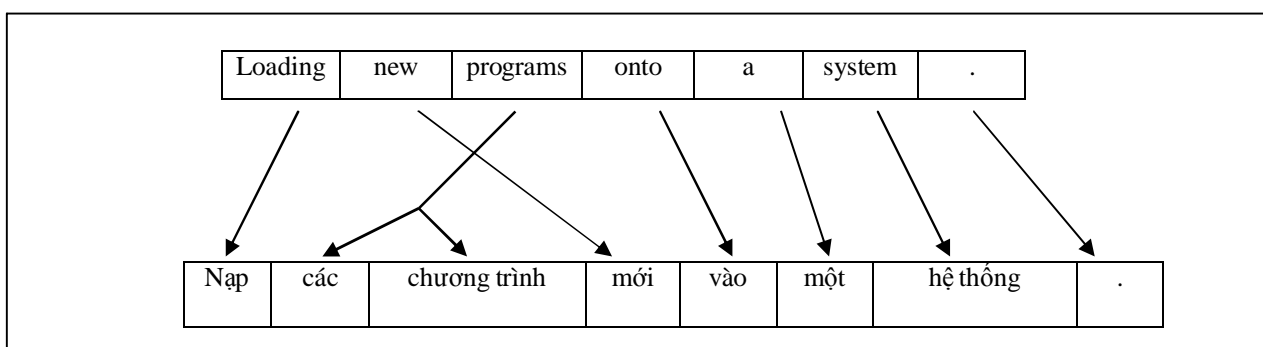


Hình 3.7. Hệ thống dịch máy thống kê hoàn chỉnh. v là câu tiếng Việt cần được dịch, e là câu tiếng Anh giả định sẽ được dịch, và \hat{e} là câu tiếng Anh được dịch tốt nhất.

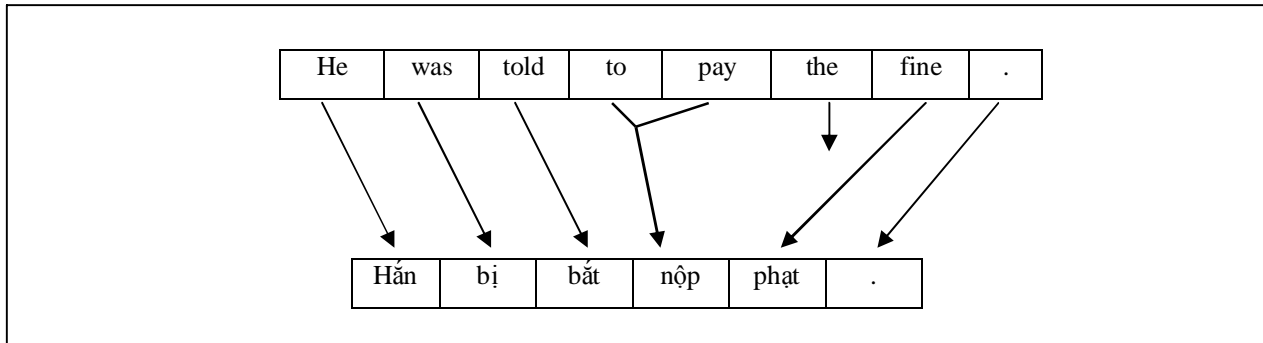
3.2.2. Định nghĩa liên kết từ/ngữ

Theo Brown *et al.* [1] thì liên kết giữa một cặp câu đã được dịch từ một ngôn ngữ nguồn (ở đây là tiếng Anh) sang một ngôn ngữ đích (ở đây là tiếng Việt) chính là xác định mỗi một từ trong câu của ngôn ngữ đích (tiếng Việt) phải được liên kết với một từ liên kết với một từ trong câu của ngôn ngữ nguồn (tiếng Anh) đã phát sinh ra nó.

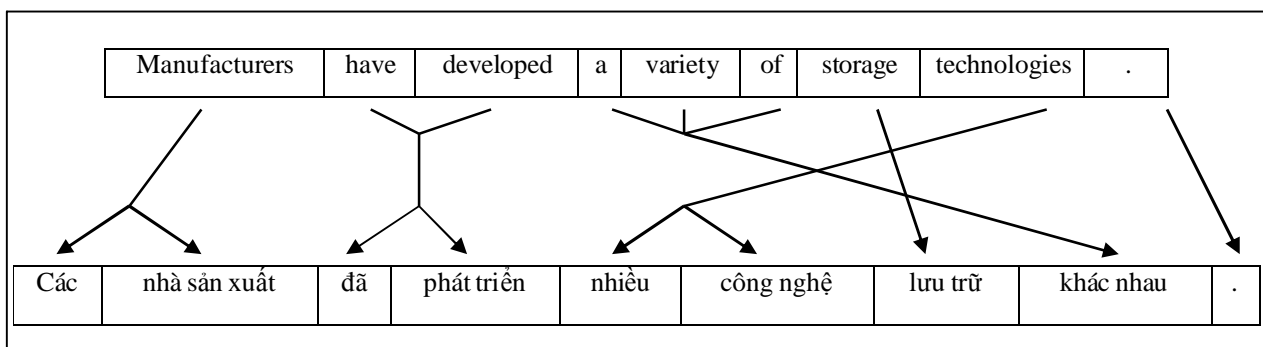
Có ba loại liên kết giữa một cặp câu được dịch từ ngôn ngữ nguồn sang ngôn ngữ đích là: liên kết một-nhiều (một từ tiếng Anh liên kết với nhiều từ tiếng Việt), liên kết nhiều-một (một từ tiếng Việt liên kết với nhiều từ tiếng Anh), liên kết nhiều-nhiều (nhiều từ tiếng Anh được liên kết với nhiều từ tiếng Việt).



Hình 3.8. Liên kết một-nhiều



Hình 3.9. Liên kết nhiều-một



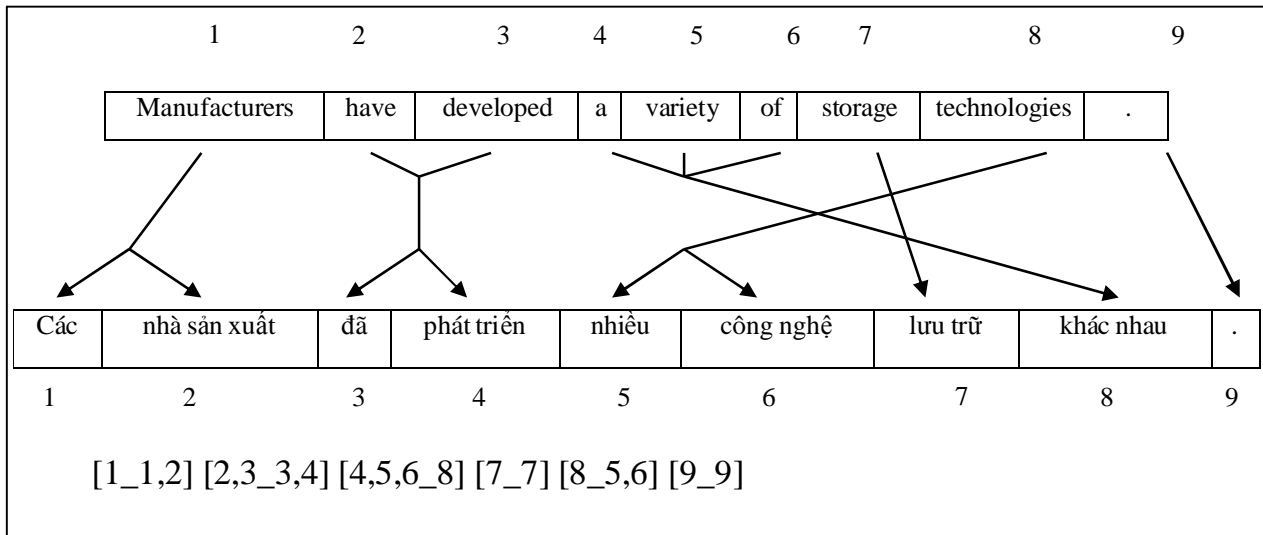
Hình 3.10. Liên kết nhiều-nhiều

Liên kết một-nhiều có nghĩa là một từ tiếng Anh có thể liên kết với một hay nhiều từ tiếng Việt và mỗi từ tiếng Việt chỉ liên kết với duy nhất một từ tiếng Anh. Liên kết nhiều-một thì ngược lại là mỗi từ tiếng Anh chỉ liên kết với duy nhất một từ tiếng Việt và mỗi từ tiếng Việt có thể liên kết với một hoặc nhiều từ tiếng Anh. Liên kết nhiều-nhiều thì kết hợp cả hai liên kết một-nhiều và nhiều-một, một từ tiếng Anh có thể liên kết với một hoặc nhiều từ tiếng Việt và ngược lại một từ tiếng Việt được liên kết với một hoặc nhiều từ tiếng Anh. Nếu một từ trong câu tiếng Anh hoặc câu tiếng Việt thì khi đó chúng tôi có thể nói từ đó được liên kết từ rỗng (NULL-word).

Có rất nhiều cách để biểu diễn một liên kết từ. Trong bài luận văn này chúng tôi có thể biểu diễn liên kết từ giữa một cặp câu được dịch sẵn bằng cách đánh dấu thứ tự của từng từ trong câu tiếng Anh và tiếng Việt, mỗi liên kết chúng ta sẽ biểu diễn trong cặp dấu “[” và “]”, trong cặp dấu này sẽ được liệt kê tất cả các vị trí tiếng Anh được liên

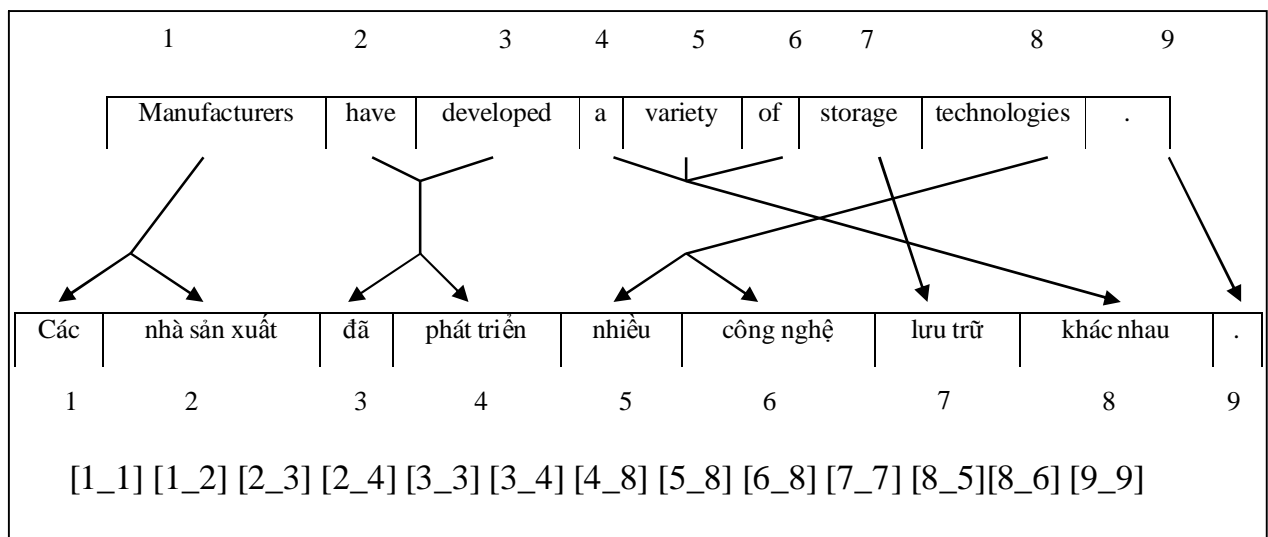
Chương 3: MÔ HÌNH THUẬT TOÁN

kết với các vị trí trong câu tiếng Việt và để phân biệt các vị trí trong câu tiếng Anh với các vị trí trong câu tiếng Việt thì chúng tôi sẽ cho cách nhau bằng dấu “_”. Ví dụ trong Hình 3.10. chúng tôi có thể biểu diễn lại trong hình như sau:



Hình 3.11. Cách biểu diễn của một liên kết cho một cặp câu đã được dịch

Còn cách đơn giản nhất là chúng tôi có thể biểu diễn mỗi kết nối được đặt nằm trong một cặp dấu “[” và “]”. Ví dụ liên kết trong Hình 3.10. còn thể được biểu diễn như sau:



Hình 3.12. Cách biểu diễn đơn giản của một liên kết cho một cặp câu đã được dịch

Chúng tôi qui ước tập hợp tất cả các liên kết từ có thể có của một cặp câu (\mathbf{e}, \mathbf{v}) là $A(\mathbf{e}, \mathbf{v})$. Nếu \mathbf{e} có chiều dài là l và \mathbf{v} có chiều dài là m , khi đó sẽ có $l \cdot m$ kết nối khác nhau giữa chúng bởi vì mỗi một từ trong câu \mathbf{v} (chiều dài là m) có thể có một kết nối với bất kỳ từ nào trong câu \mathbf{e} (chiều dài là l). Như vậy, một liên kết từ được xác định bằng cách chọn các kết nối trong $l \cdot m$ kết nối đó, do đó tập hợp $A(\mathbf{e}, \mathbf{v})$ có tất cả $2^{l \cdot m}$ liên kết từ khác nhau.

3.2.3. Mô hình ngôn ngữ

Chúng tôi qui ước \mathbf{e} là một chuỗi của những từ tiếng Anh $e_1 e_2 \dots e_l$. Mô hình ngôn ngữ $P(\mathbf{e})$ sẽ tính xác suất mà \mathbf{e} sẽ xuất hiện đúng với ngữ pháp tiếng Anh. Nhưng làm sao chúng ta có thể kiểm tra được \mathbf{e} có đúng ngữ pháp hay không? Vấn đề này là vấn đề khó để cho máy tính xử lý. Trong phần này chúng tôi sẽ đưa ra cách giải quyết gần đúng bằng cách thống kê từ ngữ liệu gồm nhiều câu tiếng Anh (với ngữ pháp chuẩn) để rút ra các thông số cho việc tính xác suất $P(\mathbf{e})$.

Theo công thức nhân xác suất, chúng tôi có thể viết lại như sau:

$$P(\mathbf{e}) = P(e_1 e_2 \dots e_l) = P(e_1) P(e_2 | e_1) P(e_3 | e_1 e_2) \dots P(e_l | e_1 e_2 \dots e_{l-1}) \quad (3.4)$$

Nhiệm vụ của mô hình ngôn ngữ là phải ước lượng được mỗi số hạng trong l số hạng ở phía bên phải của biểu thức (3.4).

Nếu chúng ta giả sử rằng $|V|$ là kích cỡ số lượng từ vựng trong từ điển tiếng Anh, thì số trường hợp khác nhau từng đôi một của $e_1 e_2 \dots e_{k-1}$ trong xác suất điều kiện thứ k sẽ là $|V|^{k-1}$. Nếu $|V|$ càng lớn thì số trường hợp khác nhau từng đôi một cũng càng lớn, do đó chúng ta không thể tính một cách chính xác cho một số hạng $P(e_k | e_1 e_2 \dots e_{k-1})$ được. Vì lý do đó, chúng tôi sử dụng mô hình trigram để tính xấp xỉ cho từng số hạng một trong biểu thức (3.4) theo cách sau:

$$P(e_k | e_1 e_2 \dots e_{k-1}) \approx P(e_k | e_{k-2} e_{k-1}) \quad (3.5)$$

Có nghĩa là thay vì chúng tôi xét $k-1$ từ vựng $e_1 e_2 \dots e_{k-1}$ đã xảy ra ở phía trước từ vựng e_k thì chúng tôi giới hạn lại chỉ xét hai từ vựng ở phía trước đã xảy ra thôi. Mỗi bộ ba $(e_{k-2} e_{k-1} e_k)$ được gọi là trigram. Như vậy, từ biểu thức (3.4) chúng tôi có thể viết lại như sau:

$$P(\mathbf{e}) = P(e_1 e_2 \dots e_l) = P(e_1) \prod_{i=2}^l P(e_i | e_{i-2} e_{i-1}) \quad (3.6)$$

Như vậy, số trường hợp mà chúng tôi xét sẽ giảm xuống còn $|V|^3$ trường hợp trigram khác nhau thôi, và trong đó có những trường hợp mà trigram không bao giờ xuất hiện trong ngữ liệu thống kê, khi đó sẽ tồn tại một $P(e_k | e_{k-2} e_{k-1})$ bằng không, và như vậy sẽ dẫn đến $P(\mathbf{e})$ cũng bằng không. Như vậy, chúng ta thấy được một điều vô lý là nếu một câu \mathbf{e} thực tế đúng với ngữ pháp tiếng Anh, nhưng nếu tồn tại một xác suất trigram $P(e_k | e_{k-2} e_{k-1})$ bằng không thì khi đó $P(\mathbf{e})$ cũng bằng không, có nghĩa là câu \mathbf{e} không đúng ngữ pháp. Để tránh được trường hợp này không xảy ra thì chúng tôi sử dụng phương pháp làm trơn bằng cách mô tả cách tính xác suất trigram $P(e_k | e_{k-2} e_{k-1})$ như sau:

$$P(e_k | e_{k-2} e_{k-1}) = \frac{1}{|V|^3} T(e_k | e_{k-2} e_{k-1}) \frac{B(e_k | e_{k-1})}{B(e_k | e_{k-2})} \frac{U(e_k)}{U(e_{k-1})} \quad (3.7)$$

với $\alpha, \beta, \gamma, \delta$: là các hệ số với điều kiện $\alpha + \beta + \gamma + \delta = 1$ và là hằng số

T : là hàm tính xác suất trigram (ba từ liên tiếp nhau)

B : là hàm tính xác suất bigram (hai từ liên tiếp nhau)

U : là hàm tính xác suất unigram (duy nhất một từ) và bằng $1/|V|$

Biểu thức (3.7) được gọi là mô hình trigram đã được làm trơn (smoothed trigram model). Theo cách tính này thì chúng ta không cần quan tâm đến ngữ nghĩa và cú pháp của những từ với nhau. Ngoài ra, để ước lượng $P(\mathbf{e})$ một cách chính xác hơn thì chúng tôi phải kết hợp với mô hình ngữ pháp liên kết [11] (linked grammars model). Đây là một mô hình huấn luyện dùng để tính xác suất ngữ pháp trong từng trigram một mà nó

sẽ mô tả đầy đủ các thông tin của trigram đó. Trong giới hạn của công việc liên kết từ nên chúng tôi không đề cập kỹ mô hình ngôn ngữ này một cách chi tiết. Mô hình ngôn ngữ này dùng để phục vụ cho hệ thống dịch máy thống kê hoàn chỉnh, chúng tôi hy vọng trong tương lai sẽ thực hiện được một hệ dịch máy thống kê hoàn chỉnh và khi đó chúng tôi sẽ mô tả chi tiết hơn.

3.2.4. Mô hình dịch

Trong phần này, chúng tôi sẽ trình bày năm mô hình dịch cùng với các thuật toán cần thiết để ước lượng những thông số của chúng. Mỗi một mô hình có một cách tính xác suất dịch $P(\mathbf{v}|\mathbf{e})$ riêng, mà chúng tôi gọi là “hàm khả năng” của việc dịch cho một cặp câu (\mathbf{v}, \mathbf{e}) . Hàm khả năng là một hàm của một số lượng lớn các thông số mà chúng tôi phải ước lượng trong quá trình thống kê và chúng tôi gọi là huấn luyện (training). Hàm khả năng của một tập hợp của các cặp câu dịch khác nhau chính là tích của những phần tử trong đó. Mục đích của chúng tôi là ước lượng giá trị cho những thông số này bằng cách áp dụng thuật toán Ước lượng-Cực đại (Estimation-Maximization Algorithm – viết tắt là EM Algorithm) vào việc huấn luyện để đạt tới giá trị cực đại cục bộ của hàm khả năng trong một tập hợp những cặp câu dịch mà chúng tôi gọi là dữ liệu huấn luyện. Khi hàm khả năng có nhiều hơn một giá trị cực đại cục bộ, thì khi đó chúng tôi sẽ chọn lựa một giá trị cực đại cục bộ bằng cách tiếp cận một số thuật toán khác mà chúng tôi sẽ trình bày chi tiết trong phần này.

Trong mô hình 1 và 2, đầu tiên chúng tôi chọn chiều dài cho câu tiếng Việt, giả sử rằng tất cả các chiều dài của các câu đều bằng nhau. Sau đó, chúng tôi tìm cách kết nối mỗi vị trí trong câu tiếng Việt tới các vị trí trong câu tiếng Anh. Trong mô hình 1, chúng tôi giả sử rằng tất cả các kết nối cho mỗi vị trí tiếng Việt đều bằng nhau. Do đó, vị trí của các từ trong câu \mathbf{e} và \mathbf{v} không ảnh hưởng đến xác suất $P(\mathbf{v}|\mathbf{e})$. Trong mô hình 2, chúng tôi làm cho thực tế hơn một chút là xác suất của mỗi kết nối phụ thuộc vào vị trí mà nó kết nối và phụ thuộc vào chiều dài của hai câu \mathbf{e} và \mathbf{v} . Vì thế, trong mô hình 2

thì xác suất $P(\mathbf{v}|\mathbf{e})$ sẽ phụ thuộc vào thứ tự của những từ trong \mathbf{e} và \mathbf{v} . Tuy nhiên, hai mô hình 1 và 2 có một hạn chế là chỉ có thể thu nhận được sự tương quan giữa hai cặp từ thường xuyên xuất hiện trong hai ngôn ngữ thôi, còn những cặp từ hiếm hay không xuất hiện thì chúng không thể liên kết được. Chúng tôi sẽ nói rõ hơn phần sau trong hai mô hình này.

Trong mô hình 3, 4, và 5, chúng tôi tạo ra một câu tiếng Việt bằng cách: đầu tiên là chọn số từ trong câu tiếng Việt mà sẽ được kết nối với các từ trong câu tiếng Anh, sau đó xác định những từ tiếng Việt này, và cuối cùng là chọn vị trí thực sự trong câu tiếng Việt mà những từ này sẽ được xuất hiện trong câu tiếng Việt. Bước cuối cùng chính là bước xác định các kết nối giữa câu tiếng Anh và câu tiếng Việt, và trong mỗi mô hình 3, 4 và 5 sẽ có cách xác định kết nối khác nhau. Trong mô hình 3 (cũng giống như mô hình 2) xác suất kết nối phụ thuộc vào vị trí mà nó kết nối và phụ thuộc vào chiều dài của câu tiếng Anh và Việt. Trong mô hình 4, xác suất của kết nối phụ thuộc vào sự xác định các từ tiếng Việt và Anh được kết nối và phụ thuộc vào vị trí của bất kỳ từ nào tiếng Việt nào mà được kết nối với cùng một từ tiếng Anh. Mô hình 3 và 4 vẫn là những mô hình chưa đầy đủ, do đó mô hình 5 sẽ tính xác suất liên kết đầy đủ nhất.

Mô hình 1 và 2 có mô hình toán học đơn giản nhất, vì thế những vòng lặp của thuật toán EM có thể tính chính xác. Chúng tôi có thể thực hiện một cách chính xác của hàm tính tổng qua tất cả các liên kết có thể cho hai mô hình này. Thêm vào đó, mô hình 1 có duy nhất một giá trị cực đại cục bộ vì thế các thông số nhận được thông qua một số vòng lặp của thuật toán EM không phụ thuộc vào điểm bắt đầu của vòng lặp. Chúng tôi sử dụng mô hình 1 để khởi gán ước lượng cho các thông số của mô hình 2. Trong mô hình 2 và những mô hình sau, hàm khả năng không có duy nhất một giá trị cực đại cục bộ, nhưng bằng cách khởi gán ban đầu của mỗi mô hình của các thông số, chúng tôi sẽ ước lượng được những thông số cho các mô hình mà không phụ thuộc vào cách khởi gán ban đầu của chúng tôi cho mô hình 1.

Trong mô hình 3 và 4, chúng tôi phải tạm chấp nhận với các vòng lặp EM gần đúng bởi vì nó không khả thi để tính tổng qua tất cả các liên kết có thể có cho những mô hình này. Mô hình 5 khắc phục được nhược điểm này.

Giữa một cặp câu (\mathbf{e}, \mathbf{v}) thì có thể có rất nhiều cách liên kết khác nhau, do đó để đánh giá độ chính xác của từng liên kết thì chúng tôi phải quan tâm đến xác suất liên kết $P(\mathbf{V} = \mathbf{v}, \mathbf{A} = \mathbf{a}, \mathbf{E} = \mathbf{e})$ gồm bộ ba biến ngẫu nhiên $(\mathbf{V}, \mathbf{A}, \mathbf{E})$. Trong đó, \mathbf{E} là chuỗi tiếng Anh ngẫu nhiên, \mathbf{V} là chuỗi tiếng Việt ngẫu nhiên, \mathbf{A} là một liên kết ngẫu nhiên của chúng trong tập hợp các liên kết có thể có trong $A(\mathbf{e}, \mathbf{v})$; dựa vào xác suất liên kết này chúng tôi tính được xác suất dịch $P(\mathbf{v} | \mathbf{e})$.

Chúng tôi có thể tính xác suất $P(\mathbf{v} | \mathbf{e})$ thông qua xác suất $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ bằng biểu thức như sau:

$$P(\mathbf{v} | \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{v}, \mathbf{a} | \mathbf{e}) \quad (3.8)$$

Hàm tính tổng ở đây duyệt qua tất cả các liên kết có trong tập hợp $A(\mathbf{e}, \mathbf{v})$. Chúng tôi giới hạn việc liên kết từ của cặp song ngữ Anh-Việt theo cách liên kết một-nhiều. Chúng tôi qui ước câu tiếng Anh là $\mathbf{e} = e_1^l = e_1 e_2 \dots e_l$ với l từ, và câu tiếng Việt là $\mathbf{v} = v_1^m = v_1 v_2 \dots v_m$ với m từ, khi đó liên kết từ một-nhiều được ký hiệu là \mathbf{a} giữa cặp câu \mathbf{e} và \mathbf{v} có thể được biểu diễn một mảng có m phần tử $\mathbf{a} = a_1^m = a_1 a_2 \dots a_m$, mỗi phần tử có giá trị từ 0 tới l . Nếu $a_j = 0$ thì tại vị trí thứ j trong câu \mathbf{v} không có kết nối với vị trí nào trong câu \mathbf{e} , nếu $a_j = i, 1 \leq j \leq m, 1 \leq i \leq l$ thì tại vị trí thứ j trong câu \mathbf{v} có một kết nối tới vị trí thứ i trong câu \mathbf{e} .

Theo công thức nhân xác suất, chúng ta có thể tính $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ như sau:

$$P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = P(m | \mathbf{e}) \prod_{j=1}^m P(a_j | a_1^{j-1}, v_1^{j-1}, m, \mathbf{e}) P(v_j | a_1^j, v_1^{j-1}, m, \mathbf{e}) \quad (3.9)$$

3.2.4.1. Mô hình 1

Trong mô hình 1 này chúng tôi giả sử rằng $P(m|\mathbf{e})$ độc lập với \mathbf{e} và m ; và $P(a_j | a_1^{j-1}, v_1^{j-1}, m, \mathbf{e})$ chỉ phụ thuộc vào chiều dài l của câu tiếng Anh; và $P(v_j | a_1^j, v_1^{j-1}, m, \mathbf{e})$ chỉ phụ thuộc v_j và e_{a_j} . Với cách giả sử này chúng tôi có thể đặt như sau:

$$P(m|\mathbf{e}) \quad (3.10)$$

$$P(a_j | a_1^{j-1}, v_1^{j-1}, m, \mathbf{e}) = \frac{1}{(l-1)} \quad (3.11)$$

$$t(v_j | e_{a_j}) = P(v_j | a_1^j, v_1^{j-1}, m, \mathbf{e}) \quad (3.12)$$

Như vậy, $t(v_j | e_{a_j})$ chính là xác suất dịch của một từ vựng e_{a_j} sang v_j , với rằng buộc là $t(v|e) = 1$.

Từ biểu thức (3.9), (3.10), (3.11) và (3.12), chúng tôi có thể viết lại:

$$P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \frac{1}{(l-1)^m} \prod_{j=1}^m t(v_j | e_{a_j}) \quad (3.13)$$

Kết hợp biểu thức (3.8) và (3.13), chúng ta được:

$$P(\mathbf{v} | \mathbf{e}) = \frac{1}{(l-1)^m} \prod_{a_1=0}^l \dots \prod_{a_m=0}^l \prod_{j=1}^m t(v_j | e_{a_j}) \quad (3.14)$$

Mục đích của chúng ta là phải làm cho $P(\mathbf{v} | \mathbf{e})$ lớn nhất, nhưng muốn tính được $P(\mathbf{v} | \mathbf{e})$ thì chúng ta phải tính được thông số t .

Theo phép biến đổi Lagrange, chúng tôi có thể viết như sau:

$$h(t, \mathbf{e}) = \frac{1}{(l-1)^m} \prod_{a_1=0}^l \dots \prod_{a_m=0}^l \prod_{j=1}^m t(v_j | e_{a_j}) e^{-\sum_v t(v|e)} \quad (3.15)$$

Lấy đạo hàm theo $t(v|e)$ của biểu thức (3.15), chúng ta được:

$$\frac{h}{t(v|e)} = \frac{1}{(l-1)^m} \prod_{a_1=0}^{l-1} \dots \prod_{a_m=0}^{l-1} (v, v_j) (e, e_{a_j}) t(v|e)^{-1} \prod_{k=1}^m t(v_k | e_{a_k}) \quad (3.16)$$

với $(v, v_j) = \begin{matrix} 1, v & v_j \\ 0, v & v_j \end{matrix}$ và $(e, e_{a_j}) = \begin{matrix} 1, e & e_{a_j} \\ 0, e & e_{a_j} \end{matrix}$

Nếu cho vế phải của biểu thức (3.16) bằng không thì chúng ta suy ra được:

$$t(v|e) = \frac{1}{(l-1)^m} \prod_{a_1=0}^{l-1} \dots \prod_{a_m=0}^{l-1} (v, v_j) (e, e_{a_j}) \prod_{k=1}^m t(v_k | e_{a_k}) \quad (3.17)$$

Kết hợp biểu thức (3.13) và (3.17), chúng tôi có thể viết lại là

$$t(v|e) = \frac{1}{e} \sum_{\mathbf{a}} P(\mathbf{v}, \mathbf{a} | \mathbf{e}) \underbrace{\prod_{j=1}^m (v, v_j) (e, e_{a_j})}_{\text{number of times } e \text{ connects to } v \text{ in } \mathbf{a}} \quad (3.18)$$

Như vậy, chúng ta nhận thấy rằng $\prod_{j=1}^m (v, v_j) (e, e_{a_j})$ chính là đếm số lần kết nối từ

vùng e tới từ vùng v trong một liên kết \mathbf{a} của một cặp câu (\mathbf{e}, \mathbf{v}) . Chúng tôi đặt như sau:

$$c(v|e; \mathbf{v}, \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) \prod_{j=1}^m (v, v_j) (e, e_{a_j}) \quad (3.19)$$

với $P(\mathbf{a} | \mathbf{e}, \mathbf{v}) = P(\mathbf{v}, \mathbf{a} | \mathbf{e}) / P(\mathbf{v} | \mathbf{e})$. Nếu chúng tôi thay thế $\frac{1}{e}$ là $\frac{1}{e} P(\mathbf{v} | \mathbf{e})$ thì biểu thức (3.18) có thể được viết lại một cách rút gọn như sau:

$$t(v|e) = \frac{1}{e} c(v|e; \mathbf{v}, \mathbf{e}) \quad (3.20)$$

Biểu thức (3.20) chỉ mới xét trên một cặp câu $(\mathbf{v} | \mathbf{e})$, nhưng trong thực tế thì chúng ta phải xét trên một ngữ liệu huấn luyện lớn gồm nhiều cặp câu $(\mathbf{v}^{(1)} | \mathbf{e}^{(1)})$, $(\mathbf{v}^{(2)} | \mathbf{e}^{(2)})$, ..., $(\mathbf{v}^{(S)} | \mathbf{e}^{(S)})$, vì thế thông số t được tính lại như sau:

$$t(v|e) = \frac{1}{e} \sum_{s=1}^S c(v|e; \mathbf{v}^{(s)}, \mathbf{e}^{(s)}) \quad (3.21)$$

Chúng ta nhận xét một điều là trong biểu thức (3.14) để tính được $P(\mathbf{v}|\mathbf{e})$ thì phải duyệt qua $(l+1)^m$ liên kết có thể có trong cặp câu (\mathbf{e}, \mathbf{v}) . Dẫn đến quá trình tính thông số t cũng phải duyệt qua $(l+1)^m$, nhưng đặc biệt trong mô hình 1 này chúng ta có thể cải tiến được tốc độ bằng cách đặt thừa số chung cho biểu thức (3.14) như sau:

$$\prod_{j=0}^l \dots \prod_{a_m=0}^m t(v_j | e_{a_j}) \prod_{j=1}^m \prod_{i=0}^l t(v_j | e_i) \quad (3.22)$$

Khi đó, biểu thức (3.14) được viết lại như sau:

$$P(\mathbf{v}|\mathbf{e}) = \frac{\prod_{j=1}^m \prod_{i=0}^l t(v_j | e_i)}{(l+1)^m} \quad (3.23)$$

và biểu thức (3.19) được viết lại là

$$c(v | e; \mathbf{v}, \mathbf{e}) = \frac{t(v | e)}{t(v | e_0) \dots t(v | e_l)} \prod_{j=1}^m (v, v_j) \prod_{i=0}^l (e, e_i) \quad (3.24)$$

với $\prod_{j=1}^m (v, v_j)$ có nghĩa là đếm v trong \mathbf{v} , và $\prod_{i=0}^l (e, e_i)$ là đếm e trong \mathbf{e} .

Như vậy, đối với biểu thức (3.24) số lần lặp để đếm sẽ giảm xuống còn $(m+l+1)$ thay vì phải lặp $(l+1)^m$ như trong biểu thức (3.19).

3.2.4.2. Mô hình 2

Trong mô hình 1, chúng tôi không quan tâm đến vị trí xuất hiện của các từ trong câu. Trong mô hình 2 chúng tôi cũng giả sử giống như mô hình 1, ngoại trừ chúng tôi giả sử thêm rằng $P(a_j | a_1^{j-1}, v_1^{j-1}, m, \mathbf{e})$ phụ thuộc vào j, a_j, m , và l , thay vì chỉ phụ thuộc duy nhất vào l như trong mô hình 1. Chúng tôi đặt như sau:

$$a(a_j | j, m, l) = P(a_j | a_1^{j-1}, v_1^{j-1}, m, \mathbf{e}) \quad (3.25)$$

với ràng buộc là

$$\prod_{i=0}^l a(a_j | j, m, l) = 1 \quad (3.26)$$

cho mỗi bộ ba (j, m, l) . Thay vào biểu thức (3.9) chúng ta được

$$P(\mathbf{v} | \mathbf{e}) = \prod_{a_1=0}^l \dots \prod_{a_m=0}^l \prod_{j=1}^m t(v_j | e_{a_j}) a(a_j | j, m, l) \quad (3.27)$$

Khai triển Lagrange chúng ta có

$$h(t, a, \dots) = \prod_{a_1=0}^l \dots \prod_{a_m=0}^l \prod_{j=1}^m t(v_j | e_{a_j}) a(a_j | j, m, l) \quad (3.28)$$

$$e \left(\prod_{v} t(v | e) - 1 \right) = \prod_{j=1}^m \prod_{i=1}^l a(i | j, m, l) - 1$$

Cũng giống như mô hình 1, lấy đạo hàm hai vế của biểu thức (3.28) và cho bằng không thì chúng tôi có thể rút ra được hai hàm đếm. Chúng ta có thêm một hàm đếm mới là $c(i | j, m, l; \mathbf{v}, \mathbf{e})$, hàm sẽ đếm số lần mà một từ bất kỳ ở vị trí j trong \mathbf{v} được nối với một từ bất kỳ ở vị trí i trong \mathbf{e} . Hàm này được khai triển giống như trong mô hình 1 và được tính như sau:

$$c(i | j, m, l; \mathbf{v}, \mathbf{e}) = \frac{\partial}{\partial a} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) (i, a_j) \quad (3.29)$$

Trong mô hình 2 cũng sử dụng lại biểu thức (3.20) và (3.21) để tính xác suất dịch t , và tương tự chúng tôi viết được

$$a(i | j, m, l) = \prod_{jml}^1 c(i | j, m, l; \mathbf{v}, \mathbf{e}) \quad (3.30)$$

và đối với một ngữ liệu huấn luyện gồm nhiều cặp câu thì

$$a(i | j, m, l) = \prod_{jml}^1 \prod_{s=1}^S c(i | j, m, l; \mathbf{v}^{(s)}, \mathbf{e}^{(s)}) \quad (3.31)$$

Cũng tương tự như mô hình 1, mô hình 2 cũng có thể giảm số lần lặp xuống bằng cách đặt thừa số chung và có thể viết lại biểu thức (3.27) lại như sau:

$$P(\mathbf{v} | \mathbf{e}) = \prod_{j=1}^m \prod_{i=0}^l t(v_j | e_i) a(i | j, m, l) \quad (3.32)$$

Khi đó, hai hàm đếm được viết lại như sau:

$$c(v | e; \mathbf{v}, \mathbf{e}) = \prod_{j=1}^m \prod_{i=0}^l \frac{t(v | e) a(i | j, m, l)}{t(v | e_0) a(0 | j, m, l) \dots t(v | e_l) a(l | j, m, l)} (v, v_j) (e, e_i) \quad (3.33)$$

và

$$c(i | j, m, l; v, e) = \frac{t(v_j | e_i) a(i | j, m, l)}{t(v_j | e_0) a(0 | j, m, l) \dots t(v_j | e_l) a(l | j, m, l)} \quad (3.34)$$

3.2.4.3. Một cách đặt vấn đề khác

Để có thể huấn luyện các thông số trong mô hình 1 và 2, chúng tôi giả sử hàng loạt các xác suất có điều kiện trong biểu thức (3.9) và dùng phép biến đổi Lagrange để tìm cách tính các thông số. Nhưng biểu thức (3.9) không phải là cách duy nhất có thể tính hàm khả năng $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$. Mỗi tích số trong vế phải của biểu thức (3.9) tương ứng với việc quá trình phát sinh tự nhiên để tạo lập \mathbf{v} và \mathbf{a} từ \mathbf{e} cho trước. Quá trình tính $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ của biểu thức (3.9) có thể được diễn tả như sau: đầu tiên chúng tôi chọn chiều dài cho \mathbf{v} , kế tiếp chúng tôi quyết định vị trí trong \mathbf{e} được kết nối với v_1 , sau đó chúng tôi lại tiếp tục chọn vị trí trong \mathbf{e} được kết nối với v_2 , và cứ như thế. Đối với mô hình 3, 4 và 5, chúng tôi viết lại hàm tính khả năng $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ bằng những tích số có điều kiện nhưng theo một cách khác.

Trong mô hình 3, 4 và 5, chúng tôi quan tâm về khả năng của một từ vựng tiếng Anh e sẽ được phát sinh bao nhiêu từ vựng tiếng Việt, chúng tôi gọi là số sản sinh ϕ_e , và đây là một biến ngẫu nhiên. Khi huấn luyện xong mô hình 1 và 2, thì chúng tôi tính được xác suất $P(\phi_e)$ của biến ngẫu nhiên này thông qua các liên kết mà hai mô hình này đã liên kết được. Nhưng xác suất này vẫn không đánh giá được khả năng sản

sinh của một từ vựng tiếng Anh e . Trong mô hình 3, 4, và 5, chúng tôi sẽ tính xác suất này một cách chính xác hơn.

Chúng tôi có thể tóm tắt quá trình trong mô hình 3, 4 và 5 như sau. Cho một câu e , đầu tiên chúng tôi chọn khả năng sản sinh của mỗi từ tiếng Anh trong câu này và một danh sách các từ tiếng Việt mà được kết nối tới. Chúng tôi gọi danh sách này (có thể là danh sách rỗng) là *tablet*. Tập hợp các *tablet* là một biến ngẫu nhiên \mathbf{T} , mà chúng tôi gọi là tập hợp *tableau* của e ; danh sách *tablet* cho từ tiếng Anh thứ i là một biến ngẫu nhiên \mathbf{T}_i ; và từ thứ tiếng Việt thứ k trong danh sách *tablet* thứ i cũng là một biến ngẫu nhiên \mathbf{T}_{ik} . Sau khi chọn tập hợp *tableau* của e , chúng tôi hoán vị những từ trong tập hợp *tableau* để phát sinh được câu v . Việc hoán vị này là một biến ngẫu nhiên Π . Vị trí trong v của từ thứ k trong danh sách *tablet* thứ i cũng là một biến ngẫu nhiên Π_{ik} .

Khả năng kết nối cho một tập hợp *tableau*, và việc hoán vị có thể viết là

$$P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{i=1}^L P(\mathbf{T}_i | \mathbf{e}_i) P(\mathbf{v} | \mathbf{T}, \mathbf{e}) \quad (3.35)$$

$$= \prod_{i=1}^L \prod_{k=1}^{|\mathbf{T}_i|} P(\mathbf{T}_{ik} | \mathbf{T}_i, \mathbf{e}_i) P(\mathbf{v} | \mathbf{T}, \mathbf{e})$$

$$= \prod_{i=1}^L \prod_{k=1}^{|\mathbf{T}_i|} P(\mathbf{T}_{ik} | \mathbf{T}_i, \mathbf{e}_i) P(\mathbf{v} | \mathbf{T}, \mathbf{e})$$

$$= \prod_{i=1}^L \prod_{k=1}^{|\mathbf{T}_i|} P(\mathbf{T}_{ik} | \mathbf{T}_i, \mathbf{e}_i) P(\mathbf{v} | \mathbf{T}, \mathbf{e})$$

với \mathbf{T}_i là một mảng các giá trị $T_{i1}, \dots, T_{i|\mathbf{T}_i|}$; và \mathbf{T}_{ik} là một mảng các giá trị $T_{ik1}, \dots, T_{ik|\mathbf{T}_{ik}|}$; và \mathbf{e}_i là viết tắt của e_i .

Nếu tìm được \mathbf{v} và \mathbf{a} thì sẽ xác định được chuỗi \mathbf{v} và liên kết \mathbf{a} . Nhưng trong trường hợp tổng quát thì có một số cặp (\mathbf{v}, \mathbf{a}) sẽ phát sinh cùng một cặp (\mathbf{v}, \mathbf{a}) . Chúng tôi qui ước tập hợp các cặp này là $\langle \mathbf{v}, \mathbf{a} \rangle$. Khi đó, chúng tôi có thể viết

$$P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \frac{P(\mathbf{v}, \mathbf{a} | \mathbf{e})}{|\langle \mathbf{v}, \mathbf{a} \rangle|} \quad (3.36)$$

Số phần tử trong tập hợp $\langle \mathbf{v}, \mathbf{a} \rangle$ là $\prod_{i=0}^l i!$, bởi vì mỗi i thì có $i!$ cách sắp xếp để được một cặp (\mathbf{v}, \mathbf{a}) .

Trong trường hợp tổng quát thì có duy nhất một liên kết trong tập hợp $A(\mathbf{e}, \mathbf{v})$ mà xác suất liên kết $P(\mathbf{a} | \mathbf{e}, \mathbf{v})$ là lớn nhất, và chúng tôi gọi liên kết này là liên kết *Viterbi* của $(\mathbf{v} | \mathbf{e})$ và được qui ước ký hiệu là $V(\mathbf{v} | \mathbf{e})$. Trong thực tế thì không có thuật toán nào để tìm $V(\mathbf{v} | \mathbf{e})$ cho mô hình tổng quát được. Nhưng đối với mô hình 2 (hay cả mô hình 1) thì việc tìm $V(\mathbf{v} | \mathbf{e})$ là một việc dễ dàng. Như đối với mô hình 2 để tìm một liên kết tại vị trí thứ j , chúng tôi chỉ đơn giản là tìm tích số $t(v_j | e_{a_j})a(a_j | j, m, l)$ là lớn nhất có thể. Biểu thức sau mô tả việc tìm một liên kết *Viterbi* trong mô hình 2:

$$j = 1, \dots, m : a_j = \arg \max_i t(v_j | e_{a_j})a(a_j | j, m, l) \quad (3.37)$$

Liên kết *Viterbi* phụ thuộc vào cách tính của từng mô hình, nên chúng tôi sẽ ký hiệu lại liên kết *Viterbi* bằng cách thêm một con số ở phía sau. Ví dụ, liên kết *Viterbi* của mô hình 1 được ký hiệu là $V(\mathbf{v} | \mathbf{e})$, của mô hình 2 là $V(\mathbf{v} | \mathbf{e}; 2)$, ...

3.2.4.4. Mô hình 3

Mô hình 3 dựa vào biểu thức (3.35). Chúng tôi giả sử rằng mỗi vị trí i từ 1 đến l thì $P(\mathbf{v}_i | \mathbf{v}_{1}^{i-1}, \mathbf{e})$ chỉ phụ thuộc vào \mathbf{v}_i và e_i ; và tất cả các vị trí i thì $P(\mathbf{v}_{ik} | \mathbf{v}_{i1}^{k-1}, \mathbf{v}_{i0}^{i-1}, \mathbf{e})$ chỉ phụ thuộc vào \mathbf{v}_{ik} và e_i ; và mỗi vị trí i từ 1 đến l thì $P(\mathbf{v}_{ik} | \mathbf{v}_{i1}^{k-1}, \mathbf{v}_{i1}^{i-1}, \mathbf{v}_{i0}^{l-1}, \mathbf{e})$ phụ thuộc vào \mathbf{v}_{ik}, i, m, l . Các thông số trong mô hình 3 được đặt như sau

$$n(\mathbf{v}_i | e_i) = P(\mathbf{v}_i | \mathbf{v}_1^{i-1}, \mathbf{e}) \quad (3.38)$$

$$t(v | e_i) = P(\mathbf{T}_{ik} = v | \mathbf{v}_{i1}^{k-1}, \mathbf{v}_{i0}^{i-1}, \mathbf{e}) \quad (3.39)$$

$$d(j | i, m, l) = P(\mathbf{v}_{ik} = j | \mathbf{v}_{i1}^{k-1}, \mathbf{v}_{i1}^{i-1}, \mathbf{v}_{i0}^{l-1}, \mathbf{e}) \quad (3.40)$$

Cũng tương tự như mô hình 1 và 2, một liên kết của $(\mathbf{v} | \mathbf{e})$ bằng cách xác định a_j cho mỗi vị trí trong câu tiếng Việt. Khi đó giá trị i được tính bằng cách đếm số phần tử a_j có giá trị là i . Vì vậy,

$$P(\mathbf{v} | \mathbf{e}) = \prod_{a_1=0}^l \dots \prod_{a_m=0}^l P(\mathbf{v}, \mathbf{a} | \mathbf{e})$$

$$= \prod_{a_1=0}^l \dots \prod_{a_m=0}^l \prod_{j=1}^m p_0^{m-2} p_1^0 \dots i! n(i | e_i) \quad (3.41)$$

$$\prod_{j=1}^m t(v_j | e_{a_j}) d(j | a_j, m, l)$$

với

$$t(v | e) = \prod_{j=1}^m d(j | i, l, m) \quad (3.42)$$

$$n(i | e) = \prod_{j=1}^m p_0 p_1 \dots$$

p_0 là xác suất sản sinh với các từ không rỗng (NULL), p_1 là xác suất sản sinh với các từ rỗng (NULL).

Khai triển Lagrange của biểu thức (41) chúng ta có

$$h(t, d, n, p, \dots) = P(\mathbf{v} | \mathbf{e}) \prod_{e_i} e_i^{t(v | e) - 1} \prod_{i=1}^m i^{d(j | i, l, m) - 1} \quad (3.42)$$

$$\prod_{e_i} e_i^{n(i | e) - 1} (p_0 p_1 \dots)$$

Cũng giống như mô hình 1 và 2, chúng tôi có thể lập các hàm đếm là

$$c(v | e; \mathbf{v}, \mathbf{e}) = \prod_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) \prod_{j=1}^m (v, v_j) (e, e_i) \quad (3.43)$$

$$c(j | i, m, l; \mathbf{v}, \mathbf{e}) = \prod_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) (i, a_j) \quad (3.44)$$

$$c(i | e; \mathbf{v}, \mathbf{e}) = \prod_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) \prod_{i=1}^l (i, i) (e, e_i) \quad (3.45)$$

$$c(0; \mathbf{v}, \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) (m - 2 - o) \quad (3.46)$$

$$c(\mathbf{l}; \mathbf{v}, \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) \quad (3.47)$$

Các hàm ước lượng của mô hình 3 là

$$t(v|e) = \prod_{s=1}^S c(v|e; \mathbf{v}^{(s)}, \mathbf{e}^{(s)}) \quad (3.48)$$

$$d(j|i, m, l) = \prod_{s=1}^S c(j|i, m, l; \mathbf{v}^{(s)}, \mathbf{e}^{(s)}) \quad (3.49)$$

$$n(\mathbf{e}) = \prod_{s=1}^S c(\mathbf{e}; \mathbf{v}^{(s)}, \mathbf{e}^{(s)}) \quad (3.50)$$

$$p_k = \prod_{s=1}^S c(k; \mathbf{v}^{(s)}, \mathbf{e}^{(s)}) \quad (3.51)$$

3.2.4.5. Mô hình 4

Thông thường trong một câu tiếng Anh được cấu tạo bằng những cụm từ (phrase) như là những *đơn vị* được dịch sang tiếng Việt. Nhưng một cụm từ tiếng Việt được dịch từ tiếng Anh có thể xuất hiện tại vị trí trong chuỗi tiếng Việt khác với vị trí của cụm từ tương ứng tiếng Anh xuất hiện trong câu tiếng Anh. Sự phân phối xác suất của mô hình 3 không giải thích tốt trường hợp di chuyển của những cụm từ xung quanh những *đơn vị*. Sự di chuyển của một cụm từ dài sẽ có giá trị nhỏ hơn sự di chuyển của cụm từ ngắn hơn bởi vì mỗi từ phải được di chuyển một cách độc lập. Trong mô hình 4, chúng tôi thay đổi cách xử lý của $P(i_k, j | i_1^{k-1}, i_1^j, l_0, l_0, e)$ để làm giảm bớt vấn đề này. Những từ mà được kết nối tới từ rỗng thường không là những dạng cụm từ và vì thế chúng tôi giả sử rằng những từ này được trải ra đồng nhất liên tục trong chuỗi tiếng Việt.

Trong Model 4, chúng tôi thay thế $d(j|i, m, l)$ bởi hai tập hợp của những thông số: một cho việc đặt từ đầu tiên, và một cho việc đặt bất kỳ những từ còn lại. Cho $[i] > 0$, chúng tôi yêu cầu rằng từ đầu tiên thứ i là $_{[i]1}$ và giả sử rằng:

$$P(_{[i]k} = j | _{[i]1}^{[i]-1}, _{0,0}^l, _{0,0}^l, e) = d_1(j = _{i-1} | A(e_{[i-1]}), B(v_j)) \quad (3.52)$$

Ở đây, A và B là những hàm phân lớp của từ tiếng Anh và tiếng Việt. Brown *et al.* mô tả một thuật toán cho việc phân chia từ vựng vào trong 50 lớp tương ứng với thuật toán này. Bằng cách giả sử rằng xác suất mỗi từ chính tả phụ thuộc vào ở mỗi từ vựng trước và phụ thuộc trên sự xác định của từ tiếng Việt đang được đặt. Chúng tôi gọi $j = _{i-1}$ là sự di chuyển của từ đầu tiên thứ i . Nó có thể cả âm lẫn dương. Chúng tôi cho rằng $d_1(_{i-1} | A(e), B(v))$ thì lớn hơn $d_1(_{i-1} | A(e), B(v))$ khi e là một tính từ và v là một danh từ.

Giả định rằng chúng tôi muốn đặt từ thứ k của từ thứ i với $[i] > 0, k > 1$. Chúng tôi giả sử rằng:

$$P(_{[i]k} = j | _{[i]1}^{k-1}, _{[i]1}^{[i]-1}, _{0,0}^l, _{0,0}^l, e) = d_1(j = _{[i]k-1} | B(v_j)) \quad (3.53)$$

3.2.4.6. Mô hình 5

Cả hai mô hình 3 và 4 đều không đầy đủ. Trong mô hình 4, không chỉ có thể một vài từ được coi là hợp lệ trên cả một từ khác, mà còn những từ có thể được đặt trước vị trí đầu tiên hay ở bên kia là vị trí cuối cùng trong câu tiếng Việt. Chúng tôi giải quyết vấn đề này trong mô hình 5.

Sau khi chúng tôi đã đặt những từ $_{[i]1}^{[i]-1}$ và $_{[i]1}^{k-1}$ cho những vị trí còn trống trong chuỗi tiếng Việt, thì rõ ràng $_{[i]k}$ nên được đặt ở một trong những vị trí còn trống này. Trong mô hình 3 và 4 thì không rõ ràng chính xác bởi vì chúng tôi không bắt buộc những ràng buộc này cho những từ chỉ có một đơn vị từ. Đặt $v(j, _{[i]1}^{[i]-1}, _{[i]1}^{k-1})$ là một con số của những chỗ trống để gia tăng và kể cả vị trí j chỉ đặt trước khi chúng tôi đặt $_{[i]k}$.

Chúng tôi viết ngắn gọn lại là v_j . Chúng tôi giữ lại hai thông số, trong mô hình 4, và tiếp tục tham khảo thêm chúng như d_1 và d_{-1} . Chúng tôi giả sử rằng, cho $[i] = 0$, ta có

$$P([i]k = j | [i] = 1, \frac{l}{0}, \frac{l}{0}, e) = d_1(v_j | B(v_j), v_{[i]-1}, v_m - [i] - 1)(1 - (v_j, v_{j-1})) \quad (3.54)$$

Số những chỗ trống là j bằng số những chỗ trống $j-1$ khi và chỉ khi j không là trống. Vì thế, thừa số cuối là 1 khi j là trống và ngược lại là 0. Trong thông số cuối cùng của d_1 , v_m là số chỗ trống còn lại trong chuỗi tiếng Việt. Nếu $[i] = 1$, thì $[i]$ có thể đặt ở vị trí bất kỳ trong những chỗ trống này; nếu $[i] = 2$, thì $[i]$ có thể được đặt ở vị trí bất kỳ nhưng trong những chỗ trống cuối; tổng quát, $[i]$ có thể được đặt trong vị trí bất kỳ nhưng đúng nhất là $[i] = 1$ của những chỗ trống còn lại. Bởi vì $[i]$ phải xuất hiện ở vị trí trái nhất của bất kỳ những từ từ $T_{[i]}$, chúng tôi phải cẩn thận để mà rời khỏi phòng tại vị trí cuối của chuỗi cho những từ còn lại từ bảng *tablet* này. Như với mô hình 4, chúng tôi cho phép d_1 phụ thuộc vào những từ nằm giữa ở phía trước và v_j .

Cho $[i] = 0$ và $k = 1$, chúng tôi giả sử

$$P([i]k = j | [i] = 1, \frac{k-1}{[i]}, \frac{[i]-1}{1}, \frac{l}{0}, \frac{l}{0}, e) = d_{-1}(v_j - v_{[i]k-1} | B(v_j), v_m - v_{[i]k-1} - [i] - k)(1 - (v_j, v_{j-1})) \quad (3.55)$$

Một lần nữa, thừa số cuối có ràng buộc $[i]$ đứng tại vị trí trống, và lại thêm một lần nữa, chúng tôi giả sử rằng xác suất phụ thuộc v_j chỉ khi là lớp của nó.

Như trong mô hình 4, chúng tôi giữ lại chi tiết của việc đếm và tính lại công thức. Không tăng sự định giá khả năng của giá trị gần đúng có thể với mô hình 5 bởi vì một lần dịch chuyển và hoán vị có thể đòi hỏi tính lại cả khối của giá trị của một liên kết từ. Vì thế, khi chúng tôi ước lượng khả năng cho mô hình 5, chúng tôi chỉ lấy những liên kết trong S . Hơn nữa, chúng tôi cắt những liên kết này bằng cách bỏ bất kỳ liên kết a nào mà có $P(a | e, v; 4)$ nhỏ hơn $P(\tilde{b} | (V(v | e; 2) | e, v; 4))$.

3.2.5. Thuật toán Ước lượng-Cực đại (Estimation-Maximization Algorithm – viết tắt là thuật toán EM)

Thuật toán EM là thuật toán huấn luyện cơ bản cho các thông số mà tất cả các mô hình đều sử dụng. Chúng tôi sẽ trình bày chi tiết thuật toán EM để cho độc giả có thể hình dung được một cách cơ bản nhất của việc huấn luyện trong dịch máy thống kê.

Để đơn giản, trước tiên chúng tôi chỉ xét thuật toán EM cho mô hình 1

Khởi gán bảng t

Với mỗi vòng lặp {

 Khởi gán các giá trị của từng phần tử trong bảng đếm tc đều bằng không

 Với mỗi cặp câu (\mathbf{e}, \mathbf{v}) có chiều dài là (l, m) trong ngữ liệu song ngữ {

 Với mỗi liên kết từ \mathbf{a} của (\mathbf{e}, \mathbf{v}) {

$$\text{Tính } P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^m t(v_j | e_{a_j})$$

$$\text{Tính } P(\mathbf{a} | \mathbf{e}, \mathbf{v}) = P(\mathbf{a}, \mathbf{v} | \mathbf{e}) / \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{v} | \mathbf{e})$$

 Với mỗi vị trí j từ 1 đến m { // Cập nhật giá trị tc

$$tc(v_j | e_i) = P(\mathbf{a} | \mathbf{e}, \mathbf{v})$$

 }

 }

}

 Cập nhật bảng t “mới” dựa trên tc

}

Hình 3.13. Mã giả của thuật toán EM cho mô hình 1

Đối với mô hình 2, 3, 4 và 5 thì thuật toán EM cũng tương tự như đối với mô hình 1, chỉ khác biệt ở chỗ tính $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$. Ví dụ như đối với mô hình 2 thì $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ được tính như sau

$$P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^m t(v_j | e_{a_j}) a(a_j | j, m, l) \quad (3.56)$$

Khởi gán bảng t, a

Với mỗi vòng lặp {

Khởi gán các giá trị của từng phần tử trong bảng đếm tc, ac đều bằng không

Với mỗi cặp câu (\mathbf{e}, \mathbf{v}) có chiều dài là (l, m) trong ngữ liệu song ngữ {

Với mỗi liên kết từ \mathbf{a} của (\mathbf{e}, \mathbf{v}) {

$$\text{Tính } P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^m t(v_j | e_{a_j}) a(a_j | j, m, l)$$

$$\text{Tính } P(\mathbf{a} | \mathbf{e}, \mathbf{v}) = P(\mathbf{a}, \mathbf{v} | \mathbf{e}) / \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{v} | \mathbf{e})$$

Với mỗi vị trí j từ 1 đến m { // Cập nhật giá trị tc

$$tc(v_j | e_i) = P(\mathbf{a} | \mathbf{e}, \mathbf{v})$$

}

Tương tự cập nhật cho bảng ac

}

}

Cập nhật bảng t, n “mới” dựa trên tc, ac

}

Hình 3.14. Mã giả của thuật toán EM cho mô hình 2

Tương tự cho mô hình 3, 4 và 5. Chỉ có $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ được tính như sau

$$P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^m p_0^{m-2} p_1^{a_j} \prod_{i=1}^n n_i! n_i(e_i) \prod_{j=1}^m t(v_j | e_{a_j}) d(j | a_j, m, l) \quad (3.57)$$

Thuật toán EM sẽ duyệt qua toàn bộ các cặp câu trong ngữ liệu song ngữ Anh-Việt. Qua mỗi bước lặp nó sẽ tìm được các giá trị tối ưu cho các thông số. Nhưng đối với mô hình 1 và 2 thì dạng thức toán đơn giản và như chúng tôi đã đề cập trong hai mô hình này là số bước lặp giảm xuống bằng cách lấy thừa số chung nên chi phí cho việc duyệt qua toàn bộ các liên kết \mathbf{a} có thể có trong một cặp câu (\mathbf{e}, \mathbf{v}) là $O(l+m+1)$. Nhưng đối với mô hình 3, 4, và 5 thì dạng thức toán khá phức tạp, vì vậy việc duyệt qua tất cả các liên kết \mathbf{a} là điều không thực tế chút nào. Thay vào đó, chúng tôi tìm một phương thức cải tiến khác để hạn chế việc duyệt qua toàn bộ các liên kết \mathbf{a} bằng cách giới hạn trong một tập các liên kết nhỏ hơn, và với hàm đánh giá như trong biểu thức (3.57) thì chúng tôi có thể thông qua một số bước lặp nhỏ để tìm một liên kết tối ưu cho tới khi nó không còn tối ưu được nữa thì ngừng. Cách làm này được gọi là “Leo đồi”.

3.2.6. Cải tiến thuật toán EM trong mô hình 3, 4 và 5

Đầu tiên, dựa vào mô hình 2 chúng tôi tìm một liên kết liên kết tối ưu nhất tìm được trong mô hình 2 (gọi là liên kết *Viterbi*). Sau đó, dựa vào liên kết *Viterbi* này chúng tôi tìm các liên kết láng giềng. Như vậy, chúng tôi chỉ duyệt qua liên kết *Viterbi* và các láng giềng của nó thôi để tính được $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$. Chúng tôi định nghĩa các liên kết láng giềng này như là những “bước nhỏ”. Mỗi bước nhỏ được định nghĩa như là làm một trong hai tác vụ: di chuyển một kết nối tiếng Việt từ một vị trí tiếng Anh sang một vị trí tiếng Anh khác, hay hoán vị hai vị trí kết nối của hai từ tiếng Việt. Khi chúng tôi thực hiện một “bước nhỏ” từ liên kết \mathbf{a} tới liên kết khác \mathbf{a}' , thì chúng tôi tránh được việc phải tính lại toàn bộ $P(\mathbf{a}' | \mathbf{v}, \mathbf{e})$ bằng cách tìm quan hệ nó với $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$.

Đối với việc di chuyển một kết nối trong **a** để thành **a'** thì việc tính quan hệ giữa $P(\mathbf{a}' | \mathbf{v}, \mathbf{e};)$ và $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ như hình sau

j là vị trí trong câu tiếng Việt

m là chiều dài của câu tiếng Việt

a là liên kết cũ

$g = 0, \dots, l$ là giá trị sản sinh cũ

a' là liên kết mới

i là vị trí cũ được liên kết với j

i' là vị trí mới được liên kết với j

từ v_j hiện thời được kết nối với tới $e_{i'}$

Nếu ($i == i'$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = 1.0$$

Nếu ($i > 0$) và ($i' > 0$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = \frac{g(i') - 1}{g(i)} \frac{n(g(i') - 1 | e_{i'})}{n(g(i') | e_{i'})} \frac{n(g(i) - 1 | e_i)}{n(g(i) | e_i)} \frac{t(v_j | e_{i'})}{t(v_j | e_i)} \frac{d(j | i', l, m)}{d(j | i, l, m)}$$

Nếu ($i == 0$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = \frac{t(v_j | e_{i'})}{t(v_j | NULL)} \frac{n(g(i') - 1 | e_{i'})}{n(g(i') | e_{i'})} \frac{d(j | i', l, m)(g(i') - 1)}{1} \frac{g(0)(m - g(0) - 1)}{(m - 2g(0) - 2)(m - 2g(0) - 1)} \frac{p_0^2}{p_1}$$

Nếu ($i' == 0$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = \frac{t(v_j | NULL)}{t(v_j | e_i)} \frac{n(g(i) - 1 | e_i)}{n(g(i) | e_i)} \frac{1}{d(j | i, l, m)g(i)} \frac{(m - 2g(0))(m - 2g(0) - 1)}{(m - g(0))(g(0) - 1)} \frac{p_1}{p_0^2}$$

Hình 3.15. Tính quan hệ xác suất của liên kết **a'** nhận được bằng cách di chuyển một kết nối trong liên kết **a**

Tương tự đối với việc hoán vị hai kết nối trong **a** để thành **a'** thì việc tính quan hệ giữa $P(\mathbf{a}' | \mathbf{v}, \mathbf{e};)$ và $P(\mathbf{v}, \mathbf{a} | \mathbf{e})$ như hình sau

j_1 và j_2 là vị trí trong câu tiếng Việt

m là chiều dài của câu tiếng Việt

\mathbf{a} là liên kết cũ

$g = 0, \dots, l$ là giá trị sản sinh cũ

\mathbf{a}' là liên kết mới

i_1 là vị trí cũ được liên kết với j_1

i_2 là vị trí cũ được liên kết với j_2

i_2 là vị trí mới được liên kết với j_1

i_1 là vị trí mới được liên kết với j_2

Nếu ($i_1 == i_2$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = 1.0$$

Nếu ($i_1 > 0$) và ($i_2 > 0$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = \frac{t(v_{j_1} | e_{i_2})}{t(v_{j_1} | e_{i_1})} \frac{t(v_{j_2} | e_{i_1})}{t(v_{j_2} | e_{i_2})} \frac{d(j_1 | i_2, l, m)}{d(j_1 | i_1, l, m)} \frac{d(j_2 | i_1, l, m)}{d(j_2 | i_2, l, m)}$$

Nếu ($i_1 == 0$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = \frac{t(v_{j_1} | e_{i_2})}{t(v_{j_1} | e_{i_1})} \frac{t(v_{j_2} | e_{i_1})}{t(v_{j_2} | e_{i_2})} \frac{d(j_1 | i_2, l, m)}{d(j_2 | i_2, l, m)}$$

Nếu ($i_2 == 0$) thì

$$\frac{P(\mathbf{a}' | \mathbf{e}, \mathbf{v})}{P(\mathbf{a} | \mathbf{e}, \mathbf{v})} = \frac{t(v_{j_1} | e_{i_2})}{t(v_{j_1} | e_{i_1})} \frac{t(v_{j_2} | e_{i_1})}{t(v_{j_2} | e_{i_2})} \frac{d(j_2 | i_1, l, m)}{d(j_1 | i_1, l, m)}$$

Hình 3.16. Tính quan hệ xác suất của liên kết \mathbf{a}' nhận được bằng cách hoán vị hai kết nối trong liên kết

a

Thuật toán EM trong mô hình 3 có thể được viết lại như sau (Brown et al., 1993a)

[1] [2]:

Bước 1: tính liên kết tốt nhất (gọi là liên kết Viterbi) có được trong mô hình 2

$$\mathbf{a}_0: V(\mathbf{v}|\mathbf{e};2), i: 0$$

Bước 2: trong khi tồn tại một liên kết trong tập hợp láng giềng $N(\mathbf{a}_i)$ có

$$P(\mathbf{a}'|\mathbf{v},\mathbf{e};3) > P(\mathbf{a}_i|\mathbf{v},\mathbf{e};3) \text{ thì}$$

(a) đặt \mathbf{a}_{i+1} là liên kết tốt nhất trong tập hợp láng giềng $N(\mathbf{a}_i)$

(b) $i: i + 1$

Bước 3: với mỗi liên kết trong \mathbf{a} trong tập hợp láng giềng $N(\mathbf{a}_i)$

(a) tính $p: P(\mathbf{a}|\mathbf{v},\mathbf{e})$

(b) for $j:=1$ to m : tăng biến đếm xác suất sai lệch

$$c(j|\mathbf{a}_j,m,l;\mathbf{v},\mathbf{e}): c(j|\mathbf{a}_j,m,l;\mathbf{v},\mathbf{e}) + p$$

(c) for $i:=1$ to l : tăng biến đếm xác suất sản sinh

$$c(i|\mathbf{e}_i;\mathbf{v},\mathbf{e}): c(i|\mathbf{e}_i;\mathbf{v},\mathbf{e}) + p$$

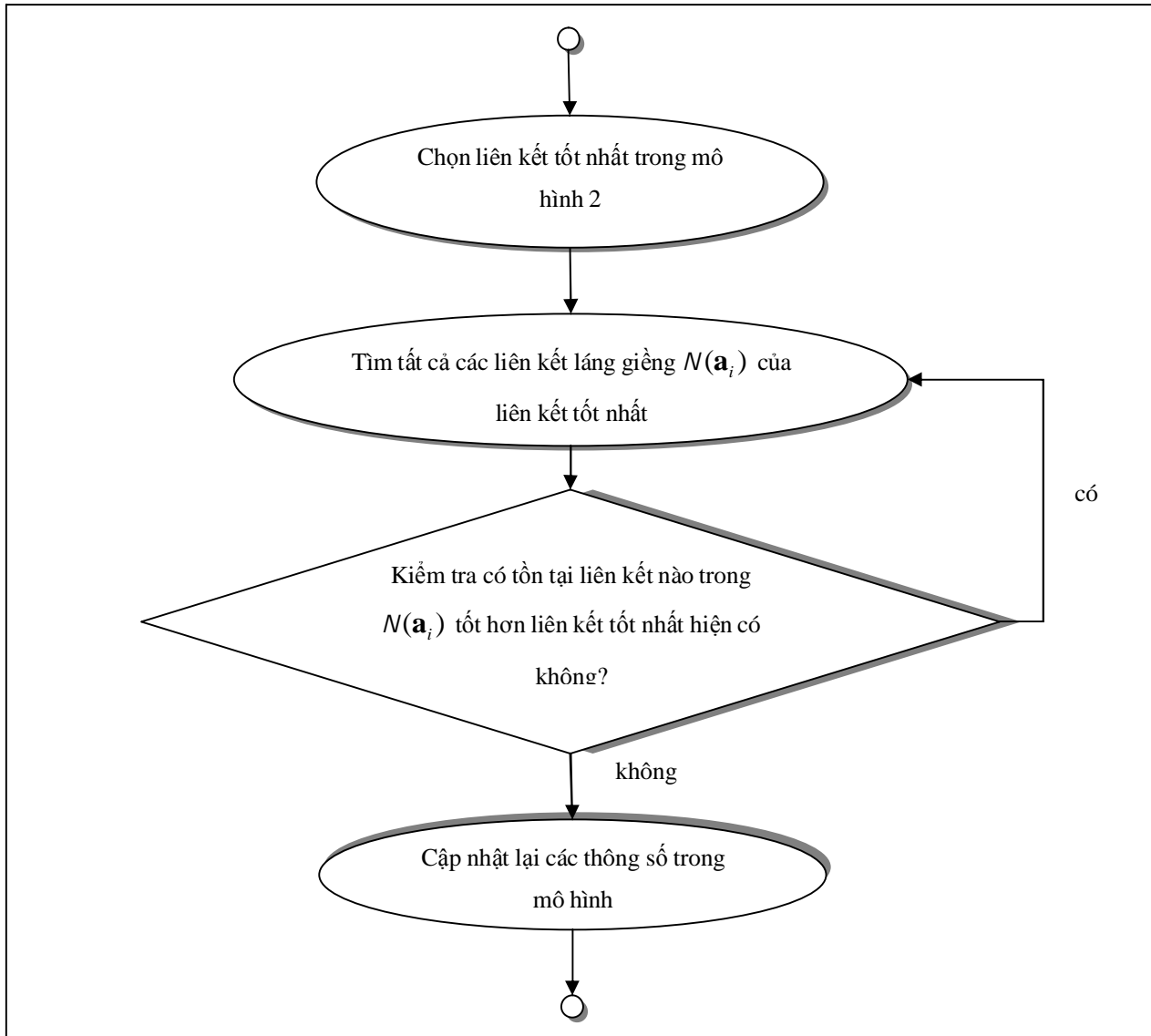
(d) tính lại xác suất liên kết với từ rỗng p_1 và từ không rỗng p_0

$$c(0;\mathbf{v},\mathbf{e}): c(0;\mathbf{v},\mathbf{e}) + p \quad (m \geq 2)$$

$$c(1;\mathbf{v},\mathbf{e}): c(1;\mathbf{v},\mathbf{e}) + p \quad 0$$

Hình 3.17. Thuật toán EM cải tiến trong mô hình 3.

Chúng ta thấy rằng trong thuật toán EM thay vì thực hiện việc vét cạn duyệt qua toàn bộ tất cả các liên kết có thể có trong một cặp câu (\mathbf{e},\mathbf{v}) để tính xác suất liên kết $P(\mathbf{a}_1^m | \mathbf{e}_1^l, \mathbf{v}_1^m)$, thì trong thuật toán Leo đòi hạn chế lại việc vét cạn bằng cách chỉ chọn ra một số liên kết tối ưu nhất và được tóm tắt trong lưu đồ sau:

**Hình 3.18. Lưu đồ thuật toán Leo đổi**

3.2.7. Tìm liên kết từ tối ưu nhất

Như chúng ta đã thấy, thuật toán EM tối ưu dần để tìm liên kết **a** trong quá trình huấn luyện, và dựa vào liên kết **a** này để tính xác suất của các thông số. Trong phần này chúng tôi chỉ giới thiệu lại cách tìm một liên kết **a** tối ưu (hay còn gọi là liên kết *Viterbi*) của một cặp câu (**e,v**) cho trước. Để tìm một liên kết **a** tối ưu của một cặp câu

(\mathbf{e}, \mathbf{v}) thì chúng tôi tìm xác suất $P(\mathbf{a} | \mathbf{e}, \mathbf{v})$ là lớn nhất. Chúng tôi có thể viết cách tìm một liên kết \mathbf{a} tối ưu như sau

$$\hat{\mathbf{a}} = \arg \max_{\mathbf{a}} P(\mathbf{a} | \mathbf{e}, \mathbf{v}) \quad (3.58)$$

Mỗi mô hình thì có cách tìm một liên kết \mathbf{a} tối ưu khác nhau tùy thuộc vào thông số của mỗi mô hình. Đối với mô hình 1 thì cách tìm liên kết tối ưu là

$$j = 1, \dots, m : a_j = \arg \max_i t(v_j | e_{a_j}) \quad (3.59)$$

Đối với mô hình 2 thì cách tìm liên kết tối ưu là

$$j = 1, \dots, m : a_j = \arg \max_i t(v_j | e_{a_j}) a(a_j | j, m, l) \quad (3.60)$$

Đối với mô hình 3 thì cách tìm liên kết tối ưu nhất giống như thuật toán EM cải tiến mà chúng tôi đã giới thiệu ở phần trên.

3.2.8. Cải tiến mô hình liên kết từ để liên kết ngữ

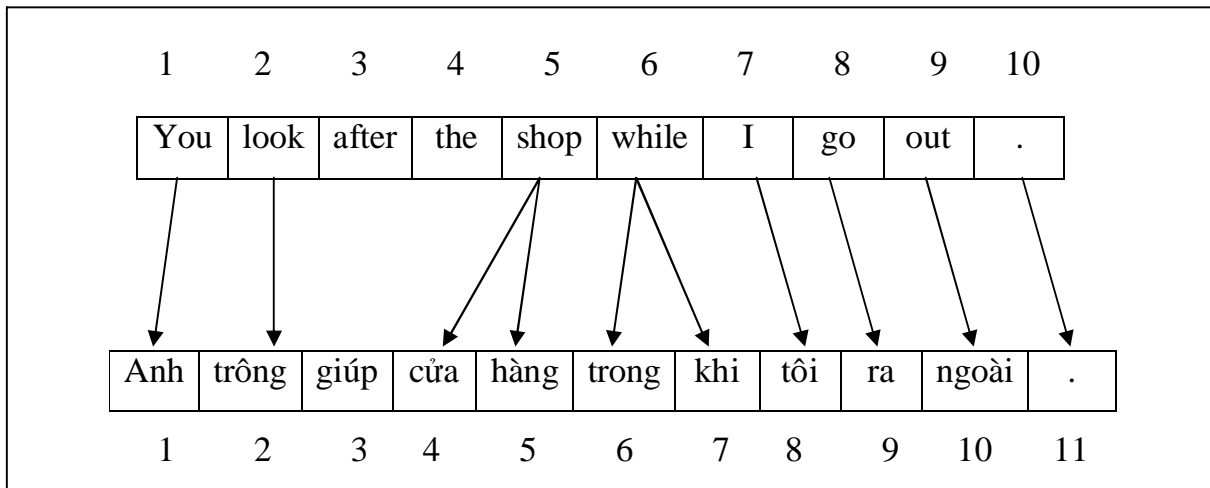
Để đơn giản hoá trong lý thuyết và trong thực tế thì chúng tôi chỉ xét trên liên kết một-nhiều. Như đã nói ở các phần trên chúng tôi xét ngôn ngữ nguồn là tiếng Anh và ngôn ngữ đích là tiếng Việt. Sau khi huấn luyện xong thì mô hình chỉ liên kết được một từ tiếng Anh với một hay nhiều từ tiếng Việt. Để có một liên kết nhiều-nhiều thì chúng tôi huấn luyện cả hai trường hợp. Trường hợp thứ nhất, chúng tôi cho ngôn ngữ nguồn là tiếng Anh và ngôn ngữ đích là tiếng Việt. Trường hợp thứ hai, chúng tôi đảo lại cho ngôn ngữ nguồn là tiếng Việt và ngôn ngữ đích là tiếng Anh. Sau khi huấn luyện xong hai trường hợp thì chúng tôi kết hợp hai kết quả lại với nhau để được một liên kết nhiều-nhiều.

Giả sử chúng tôi đã huấn luyện xong hai trường hợp và được các thông số của hai trường hợp này, và bây giờ chúng tôi nhập một cặp câu mới vào. Ví dụ chúng tôi nhập cặp câu

(E) You look after the shop while I go out.

(V) Anh trông giúp cửa hàng trong khi tôi ra ngoài.

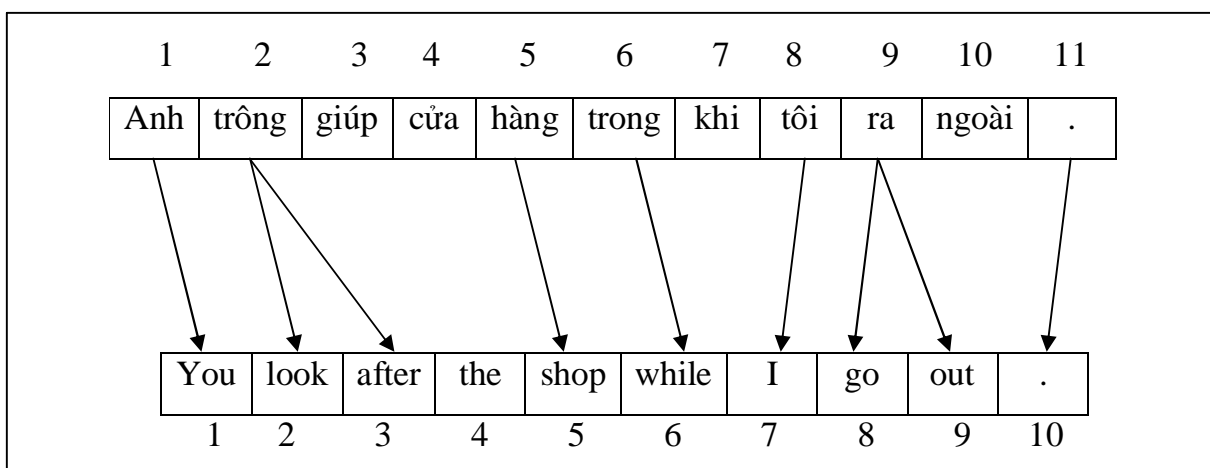
Trường hợp 1, chúng tôi cho ngôn ngữ nguồn là tiếng Anh và ngôn ngữ đích là tiếng Việt thì được kết quả như hình sau



Hình 3.19. Kết quả liên kết một-nhiều, với ngôn ngữ nguồn là tiếng Anh và ngôn ngữ đích là tiếng Việt

$\mathbf{a(E,V)} \quad \mathbf{A1} \quad \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}\} \quad \{1, 2, 0, 5, 5, 6, 6, 7, 8, 9, 10\}$

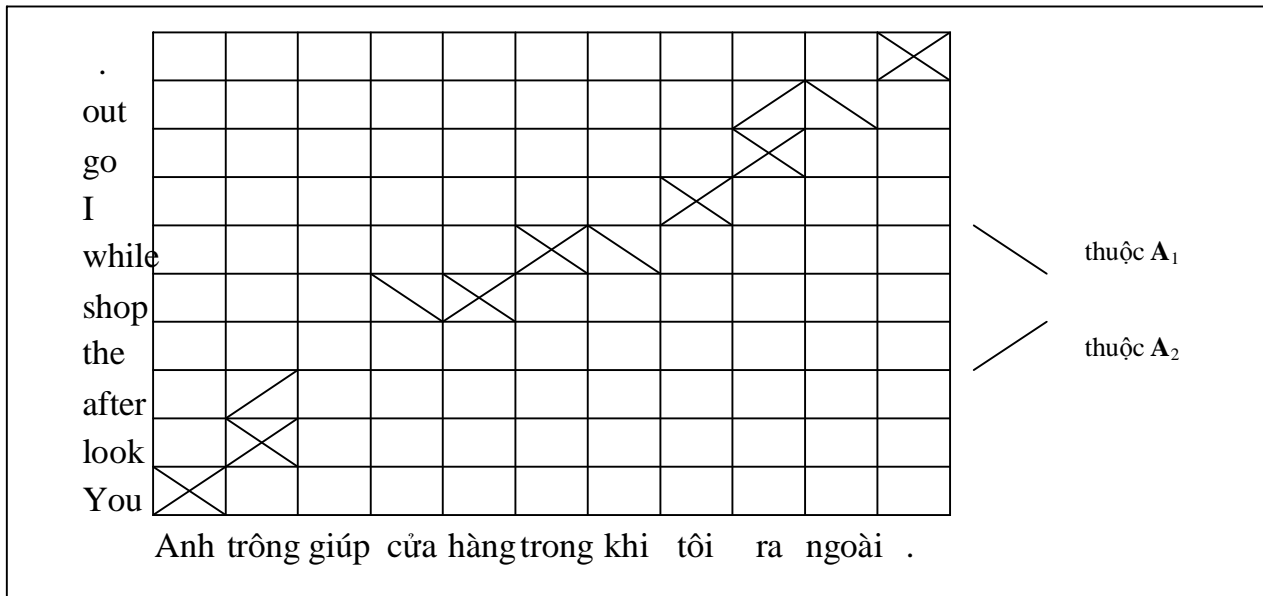
Trường hợp 2, chúng tôi cho ngôn ngữ nguồn là tiếng Việt và ngôn ngữ đích là tiếng Anh thì được kết quả như hình sau



Hình 3.20. Kết quả liên kết một-nhiều, với ngôn ngữ nguồn là tiếng Việt và ngôn ngữ đích là tiếng Anh

$a(V,E) \quad A2 \quad \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}\} \quad \{1,2,2,0,5,6,8,9,9,11\}$

Nếu kết hợp cả hai liên kết $A1$ và $A2$ thì chúng tôi được kết quả như hình sau



Hình 3.21. Kết quả của việc kết hợp cả hai trường hợp

Với hình 15 thì chúng tôi có nhận xét rằng

Nếu giao hai tập hợp $A = A_1 \cap A_2$ thì A chắc chắn đúng.

Nếu hội hai tập hợp $A = A_1 \cup A_2$ thì A chưa chắc chắn đúng.

Vấn đề: làm sao chúng ta có thể *thêm các liên kết đúng vào tập hợp giao*, hoặc *bớt những liên kết sai trong tập hợp hội*?

Chúng tôi đưa ra cách giải quyết vấn đề này như sau

Với $A_1 = \{(a_j, j) | a_j = 0\}$ và $A_2 = \{(i, a_i) | a_i = 0\}$,

Tập hợp hội là $A = A_1 \cap A_2$

Chúng tôi mở rộng tập hợp A bằng cách xét thêm một liên kết (i,j) có trong $A1$ hoặc $A2$ với những điều kiện sau:

⊆ Điều kiện 1: liên kết (i, j) có một liên kết chiều ngang kế cận $(i-1, j)$ hay $(i+1, j)$; hoặc có một liên kết chiều dọc kế cận $(i, j-1)$ hay $(i, j+1)$ đã tồn tại trong A .

⊆ Điều kiện 2: tập hợp $A \cup \{(i, j)\}$ không đồng thời chứa cả hai liên kết kế cận chiều ngang và chiều dọc.

Cứ như thế chúng tôi duyệt qua toàn bộ tất cả các liên kết còn lại có trong A_1 và A_2 .

Xét ví dụ trên chúng tôi có thể viết từng kết quả như sau

$A \quad A_1 \cap A_2 \quad \{(1,1), (2,2), (5,5), (6,6), (7,8), (8,9), (10,11)\}$

$(5, 4)$ thoả điều kiện 1 và 2 $A \cup (5,4) \quad \{(1,1), (2,2), (5,5), (6,6), (7,8), (8,9), (10,11), (5,4)\}$

$(6, 7)$ thoả điều kiện 1 và 2

$A \cup (6,7) \quad \{(1,1), (2,2), (5,5), (6,6), (7,8), (8,9), (10,11), (5,4), (6,7)\}$

$(9, 10)$ không thoả điều kiện 1

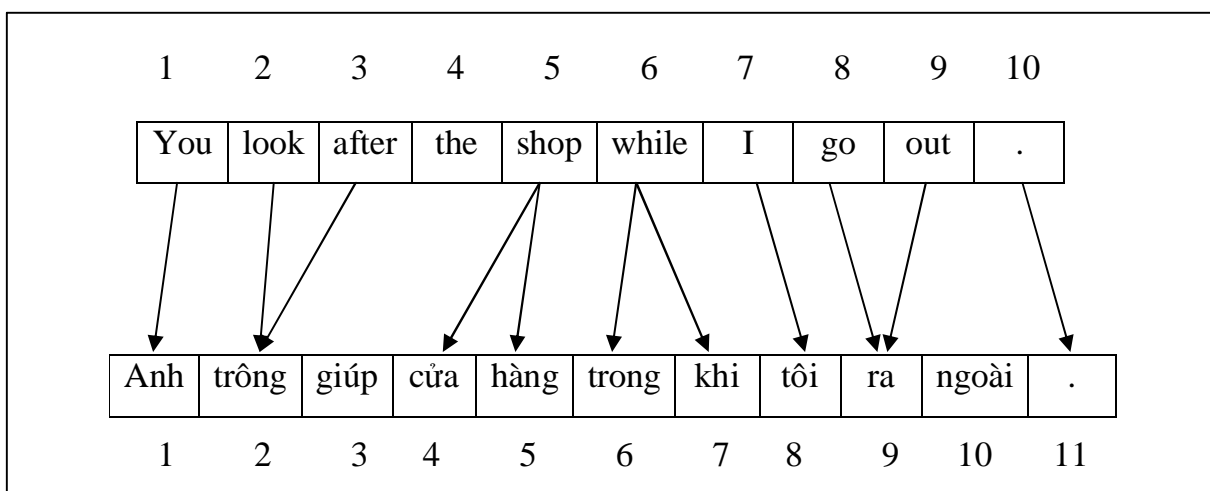
$(3, 2)$ thoả điều kiện 1 và 2

$A \cup (6,7) \quad \{(1,1), (2,2), (5,5), (6,6), (7,8), (8,9), (10,11), (5,4), (6,7), (3,2)\}$

$(9, 9)$ thoả điều kiện 1 và 2 $A \cup (6,7) \quad \{(1,1), (2,2), (5,5), (6,6), (7,8), (8,9), (10,11), (5,4), (6,7), (3,2), (9,9)\}$

$(9, 10)$ thoả điều kiện 1, không thoả điều kiện 2

Kết quả thu được liên kết A được vẽ như sau



Hình 3.22. Kết quả liên kết nhiều-nhiều

3.3. Chiếu kết quả phân tích cú pháp sang Tiếng Việt

Dựa vào kết quả phân tích cú pháp quan hệ trên câu tiếng Anh, thông qua mối liên kết ngữ trên cặp câu Anh-Việt, quá trình chiếu kết quả tiếng Anh sang tiếng Việt được tiến hành từng bước như sau

3.3.1. Chiếu nhãn từ loại

Thông qua mối liên kết từ, các từ loại của tiếng Anh sẽ được chiếu sang tiếng Việt. Tuy nhiên, do hệ thống từ loại Anh-Việt khác nhau nên trước khi gán nhãn này cho từ tiếng Việt thì nhãn từ loại tiếng Anh này sẽ được ánh xạ sang bộ nhãn Tiếng Việt. Phép ánh xạ từ nhãn từ loại tiếng Anh sang Tiếng Việt là một không phải là phép ánh xạ 1-1 mà là phép ánh xạ m-n¹.

Ở bước này, nhập nhằng này được giải quyết chỉ bằng mô hình xác suất. Theo đó, mỗi từ loại tiếng Anh sẽ được chiếu sang tiếng Việt theo nhãn từ loại cao nhất có xác suất cao nhất. Xác suất này được tính theo công thức sau:

$$P(T_{\text{Việt}}) = P_1(T_{\text{Việt}}) * P(T_{\text{Eng}} \rightarrow T_{\text{Việt}})$$

Với $P(T_{\text{Việt}})$: xác suất nhãn $T_{\text{Việt}}$ cho từ tiếng Việt đang xét.

$P_1(T_{\text{Việt}})$: xác suất xuất hiện nhãn $T_{\text{Việt}}$ trong tổng số lần xuất hiện của từ tiếng Việt.

$P(T_{\text{Eng}} \rightarrow T_{\text{Việt}})$: xác suất nhãn T_{Eng} được ánh xạ sang nhãn $T_{\text{Việt}}$.

¹ Ánh xạ m-n có thể hiểu theo 2 nghĩa khác nhau : theo số lượng từ hoặc theo số lượng từ vựng (một từ vựng bên ngôn ngữ này tương ứng với nhiều từ loại trong ngôn ngữ kia). Quan hệ số lượng từ được giải quyết trong phần liên kết ngữ nên ở đây chỉ quan tâm đến quan hệ m-n về mặt số lượng từ vựng

3.3.2. Chiếu quan hệ

Quan hệ sẽ được chiếu trực tiếp sang tiếng Việt thông qua mối liên kết từ theo nguyên lý DCA. Sự khác biệt về cấu trúc ngữ pháp của 2 ngôn ngữ làm cho có những mối quan hệ là thừa, có những mối quan hệ thiếu và có những quan hệ sai. Để giải quyết trường hợp này, chương trình đã dùng các heuristic để sửa những lỗi sai này.

3.3.3. Sử dụng luật tương tác

Sau khi đã được gán nhãn từ loại và quan hệ, câu tiếng Việt sẽ được áp dụng các luật tương tác được rút ra từ quá trình học TBL. Điều này nảy sinh ra một vấn đề là cần ngữ liệu huấn luyện cho tiếng Việt. Một lẽ tất nhiên là sẽ không có ngữ liệu huấn luyện (và vì vậy mới phải sử dụng kết quả của ngôn ngữ khác để chiếu sang). Một cách giải quyết cho sự thiếu hụt này là cách “lấy ngắn nuôi dài”. Đầu tiên, thông qua liên kết từ, các quan hệ trong một văn bản không nhiều các câu tiếng Anh sẽ chiếu trực tiếp sang tiếng Việt. Sau đó các quan hệ này sẽ được sửa bằng tay tạo thành một ngữ liệu vàng có kích thước nhỏ. Sử dụng ngữ liệu này, thuật toán học TBL sẽ rút ra một số luật tương tác. Áp dụng luật này vào chương trình để đánh một lượng văn bản lớn hơn. Bây giờ, kết quả của chương trình sẽ được tăng lên một ít (nhưng không nhiều do ngữ liệu học còn quá ít).

Quá trình học-chỉnh sửa-áp dụng cứ lặp đi lặp lại với khối lượng văn bản ngày càng nhiều và tỉ lệ chính xác ngày càng tăng (vì ngữ liệu lớn sẽ dẫn đến tính tổng quát của luật càng cao). Kết quả cuối cùng là một bộ luật học sẽ được rút ra.

Chương 4: CÀI ĐẶT THỰC NGHIỆM

Dựa trên các mô hình lý thuyết mà chúng tôi trình bày ở chương 3, trong chương này chúng tôi sẽ trình bày chi tiết việc cài đặt các bước xử lý của công việc chiếu quan hệ cú pháp từ tiếng Anh sang tiếng Việt.

4.1. Chương trình phân tích cú pháp quan hệ

Phân tích cú pháp quan hệ ngữ pháp, ngữ nghĩa là đi tìm mối quan hệ giữa những thành phần trong câu. Tuy nhiên, việc phân tích cú pháp phải dựa trên cơ sở phân tích từ loại. Hơn nữa, phân tích cú pháp quan hệ quan tâm nhiều đến ý nghĩa cú pháp của câu. Do đó, việc tổ chức lưu trữ thông tin về ngữ nghĩa và sử dụng nó như thế nào là một vấn đề quan tâm hàng đầu trong phân tích cú pháp quan hệ.

4.1.1. Phân tích từ vựng

Cũng giống như bất kì chương trình phân tích cú pháp nào, bước đầu tiên là phân tích từ vựng hay đánh nhãn từ loại (POS-tagger). Phân tích từ vựng được kết hợp chặt chẽ với phân tích hình thái. Tuy nhiên, trong chương trình này, hai vấn đề này được giải quyết bằng cùng một giải pháp : từ điển.

4.1.1.1. Từ điển

Đây là từ điển được rút trích từ WordNet. Từ điển được tổ chức với đầy đủ thông tin về cấu trúc, về các thuộc tính phục vụ cho việc xem xét đến việc thoả mãn các nguyên tắc khi phân tích cú pháp.

4.1.1.1.1. Cấu trúc

Cũng giống như các từ điển khác, nó cũng được tổ chức thành những mục, mỗi mục tương ứng với một từ cùng với các thông tin mô tả về từ loại của từ. Tuy nhiên, điểm khác biệt ở đây là từ điển được tổ chức rất chi tiết và công phu. Một mặt, nó góp phần làm giảm đi gánh nặng phân tích từ loại. Chính vì từ điển được tổ chức quá chi tiết này mà quá trình phân tích từ loại chỉ là một quá trình tra từ điển mà thôi. Về cơ bản, mỗi một mục từ trong từ điển được tổ chức như sau:

(*<Từ hoặc mục từ>*

(*<chức năng> <Các tham số>*)

(*<chức năng> <Các tham số>*)...

Dòng đầu tiên giới thiệu mục từ. Các dòng tiếp theo là sự mô tả chi tiết các thuộc tính tương ứng với từ ở dòng đầu tiên. Để rõ ràng, ta thử lấy một ví dụ cụ thể để phân tích thật chi tiết. Hãy xét đến từ “want”, từ điển được tổ chức như sau:

(*want*

(*syn (N)*)

(*syn (Ttg)*)

(*syn (Cn.a)*)

(*syn (Cn.t)*)

(*syn (Tn.[pr])*)

(*freq N 24 V_N_C 7 V_N_I 11 V_N_N_P 10 V_N_N_P_A 25*)

)

Nhìn vào đây, ta có thể thấy được từ “want” có 5 loại từ là N, Ttg, Cn.a, Cn.t, Tn.[pr]. Tuy nhiên, thật ra chỉ có 2 loại từ chính là danh từ (loại từ đầu tiên) và động từ

(bốn loại từ sau) ¹. Trường kế tiếp theo mô tả tần số xuất hiện của danh từ và động từ với các dạng tham số khác nhau.

Không chỉ chứa các từ gốc, từ điển còn chứa cả các mục từ dẫn xuất và biến cách của từ. Trở lại ví dụ từ “want”, ta có các mục từ sau:

(*wanted*
(*syn (Aatt)*)
(*freq A 9 N 27*)
)
(*wanting*
(*syn (Aprd)*)
(*syn (Prep)*)
(*freq A 18 Prep 12*)
)...

Như chúng ta vừa tìm hiểu ví dụ trên đây, từ điển từ vựng được tổ chức rất chi tiết. Do đó, việc phân tích từ vựng chỉ đơn giản là quá trình tra từ điển và quét cặn các loại từ trong từ điển. Điều này có vẻ là quá nặng nề, tốn nhiều chi phí. Tuy nhiên, điều này là hoàn toàn ngược lại bởi vì các lý do sau đây:

Thứ nhất: một lần, xét tất cả các từ vựng có thể của từ, ta sẽ không tốn một giải thuật nào để lựa chọn, chọn đi chọn lại.

Thứ hai, việc chọn tất cả này không gây nên một bùng nổ tổ hợp bởi vì trong quá trình phân tích cú pháp (như đã trình bày trong chương 3 (mô hình thuật toán)), các nhãn từ loại không phù hợp sẽ không thể nào lan truyền được trong mạng bởi sự ràng

¹ Chi tiết về việc phân loại động từ ra thành các tiểu loại sẽ được trình bày trong phần 02) Các mục động từ

buộc của các nguyên tắc và ngữ cảnh hiện tại của câu hay nói khác đi là nó không “thích nghi” được với điều kiện “môi trường” (ngữ cảnh) hiện tại của câu.

4.1.1.1.2. Sự phân loại động từ

Như đã giới thiệu trong chương 3, quá trình phân tích cú pháp quan hệ sẽ xác định được mối quan hệ giữa một thành phần được gọi là cha và một thành phần là con. Từ đóng vai trò là cha sẽ đứng ra đại diện cho nhóm các con trong các mối quan hệ. Ở mức câu, động từ sẽ được đưa lên làm trung tâm. Chính vì điều này mà động từ đóng một vai trò rất quan trọng trong việc xác định cấu trúc của câu.

Trong rất nhiều từ điển, động từ được phân lớp dựa vào các tham số bổ nghĩa mà động từ cho phép. Hầu hết các từ điển sử dụng mã để chia động từ ra làm 3 loại chính: nội động từ (intransitivity), ngoại động từ (transitivity) và ngoại động từ 2 túc từ (ditransitivity)¹. Các loại từ mở rộng (cho động từ) này lại được chia nhỏ ra thành 2 loại là động từ có tham số là ngữ danh từ (NP) và động từ có tham số là một mệnh đề (clause).

Bộ mã động từ được chuyển đổi từ bộ mã OALD (Oxford Advanced Learner's Dictionary). Động từ được chia làm 5 nhóm chính : Nội động từ, ngoại động từ, ngoại động từ 2 túc từ, ngoại động từ phức hợp, và động từ liên kết. Mỗi động từ được mã hoá dưới dạng $S a_1.[a_2]$ trong đó S là ký hiệu từ loại của động từ. Có 5 từ loại động từ tương ứng với 5 ký hiệu khác nhau (I , T , D , C , L)². Hai ký hiệu tiếp theo: a_1 và a_2 tượng trưng cho kiểu tham số cho các bổ ngữ của động từ. Nếu động từ có nhiều hơn một tham số thì các tham số này sẽ được liệt kê kế tiếp nhau.

¹ Động từ loại này có 2 túc từ trực tiếp và gián tiếp. Ví dụ động từ give : give something to some one.

² Các ký hiệu này là các chữ cái đầu của các từ tương ứng trong tiếng Anh: I(intransitive), T(transitive), D(ditransitive), C(complex transitive), L(linking verb).

Loại động từ	Các tiểu loại có thể ¹
Nội động từ	I, I _p , I _{pr} , I _{n/pr} , I _t
Ngoại động từ	T _n , T _{n.pr} , T _{n.p} , T _f , T _w , T _t , T _g , T _{n.t} , T _{n.g} , T _{n.i}
Ngoại động từ phức hợp	C _{n.a} , C _{n.n} , C _{n.n/a} , C _{n.t} , C _{n.g} , C _{n.i}
Ngoại động từ 2 túc từ	D _{n.n} , D _{n.pr} , D _{n.f} , D _{n.t} , D _{n.w} , D _{pr.f} , D _{pr.w} , D _{pr.t}
Động từ liên kết	L _a , L _n

Bảng 4.1. Danh mục các tiểu loại động từ theo OALD

Các tham số của động từ có ý nghĩa như sau

Ký hiệu	Ý nghĩa	Ví dụ minh họa
n	Danh từ (noun)	Eat [rice]
f	Mệnh đề kết thúc ²	
g	Mệnh đề có động từ thêm -ing (gerund clause)	Like [driving car]
t	Mệnh đề nguyên thể (infinitive clause)	Want [to stay at home]
w	Mệnh đề kết thúc mà bắt đầu bằng -wh	We concert [who will go there].
i	Mệnh đề nguyên thể không “to” (bare infinitive clauses).	
a	Tính từ (adjectives)	It taste [good]
p	Giới từ (preposition).	
pr	Ngữ giới từ	

Bảng 4.2. Ý nghĩa các tham số của động từ

¹ Sự phân chia các tiểu loại động từ khác nhau dựa vào các loại và vị trí của các bổ ngữ (hay còn gọi là các tham số) của động từ.

² Mệnh đề kết thúc dùng để chỉ cho loại mệnh đề không phải là mệnh đề quan hệ hay nói cách khác, nó là một mệnh đề độc lập (cả về ý nghĩa lẫn cấu trúc).

Tuy nhiên, bảng mã trên đây của OALD cũng đã thể hiện một số nhược điểm nhất định. Cụ thể như : khi chỉ ra rằng động từ được bỏ nghĩa bởi một ngữ giới từ thì ngữ giới từ này không được chỉ định rõ ràng mặc dù động từ này chỉ có một giới từ nhất định nào đó theo sau. Ngoài ra, nó còn nhiều điểm thiếu sót khác như tính tùy chọn (có hoặc không có tham số này), tính lựa chọn (có một trong số các tham số được đưa ra) sẽ được trình bày trong phần tiếp theo.

Vì những lý do này nên chương trình sử dụng bộ mã cải tiến của bộ mã OALD với một số cải tiến cho những thiếu sót của bộ mã này.

Đầu tiên, nó chỉ ra rõ giới từ cụ thể đi theo động từ. Tuy nhiên, nó chỉ được thực hiện trên 8 động từ.

Bộ mã này cho phép có một số tham số tùy chọn (có hoặc không). Ví dụ mã động từ $T_{[n],pr}$ có nghĩa là : ngoại động từ này có thể có một danh từ theo sau và sau đó là một giới từ.

Ví dụ : “ I eat rice at the canteen”.

Có thể viết là : “I eat at the canteen”. Danh từ “rice” là một mục từ bỏ ngữ tùy chọn cho động từ rice.

Một điểm cải tiến khác là nó cho phép có sự chọn lựa nhiều loại tham số khác nhau cho một vị trí bỏ ngữ của động từ.

Ví dụ : T_{fg} có nghĩa là : động từ này có thể theo sau bởi một mệnh đề kết thúc hoặc một mệnh đề với động từ thêm -ing.

4.1.1.1.3. Mục từ tham chiếu

Từ điển có thể chứa nhiều mục từ khác nhau của cùng một từ dưới nhiều hình thái khác nhau, các từ ở mức con có hầu hết các đặc tính của nốt cha. Do đó, nếu lưu trữ thông tin riêng lẻ sẽ dẫn đến dư thừa. Vì vậy mà trong từ điển có tổ chức lưu trữ theo kiểu tham chiếu.

Xét ví dụ sau: Từ *began* là thể quá khứ của từ *begin* sẽ được lưu trữ như sau :

(began

(ref ((cat (vform ed) (tense past))) begin).

ref : cho biết đây là một trường tham chiếu. Các thuộc tính của mục từ này sẽ được kế thừa từ mục từ *begin*.

vform : biến thể động từ.

tense : thì của động từ.

Từ begin ở cuối dòng cho biết mục từ mà nó tham chiếu đến.

Tuy nhiên, ta hãy quay lại ví dụ với từ *want* trước đây, có một sự khác biệt với những gì ta vừa xét. Đó chính là : mục từ *wanted* không kế thừa từ mục từ *wanted*. Nguyên nhân dẫn đến sự khác biệt trên là do từ *wanted* thực ra không phải là một dạng biến cách của động từ *want* bởi vì lúc này *wanted* là một tính từ (Aattt¹).

Quá trình phân tích từ vựng này chính là nền tảng cho quá trình phân tích ở bước tiếp theo : phân tích cú pháp quan hệ

4.1.2. Phân tích cú pháp quan hệ

Phân tích cú pháp quan hệ dựa trên sáu nguyên tắc cơ bản. Mỗi nguyên tắc có một tác động đến một khía cạnh khác nhau cũng như có một cách sử dụng hoàn toàn khác nhau. Có nguyên tắc thì được liệt kê rõ ràng như một danh sách, có nguyên tắc lại được tham số hoá như một biến trong chương trình, có nguyên tắc lại được ẩn giấu trong mạng ngữ pháp... Các cách thức tổ chức, sử dụng của các nguyên tắc sẽ lần lượt được trình bày.

4.1.2.1. Từ điển chủ ngữ của động từ

Để phục vụ cho nguyên tắc theta, mỗi một động từ sẽ được lưu một danh sách các chủ từ có thể có. Điều này bảo đảm cung subj trong mạng chỉ được truyền qua khi nốt

¹ Có 2 loại tính từ là tính từ chỉ thuộc tính (bổ nghĩa cho danh từ) và tính từ đóng vai trò vị ngữ trong câu. Aatt là loại tính từ chỉ thuộc tính (attributive adjectives), loại còn lại là Aprd (predicative adjectives).

danh từ trong mạng có thể làm chủ ngữ của nốt động từ đang xét tương ứng. Tuy nhiên, đây là từ điển thống kê nên cũng không bảo đảm rằng tất cả các trường hợp đều có xuất hiện. Tuy nhiên, kết quả cuối cùng là điểm tổng hợp của các trường hợp nên nếu một nguyên tắc thất bại cũng không chắc chắn rằng đó là một đường đi sai mà chỉ giảm đi xác suất chọn nó mà thôi.

4.1.2.2. Mạng cú pháp

Quá trình phân tích cú pháp là quá trình di chuyển của các trạm đến các nốt trong mạng ngữ pháp thông qua các cung nối. Các nốt trong mạng thể hiện các trường cú pháp. Các cung nối thể hiện cho các mối quan hệ. Mạng cú pháp đóng vai trò trung tâm, đặc trưng cho ngôn ngữ cần phân tích.

Mạng ngữ pháp được tổ chức thành những nốt và cung. Ví dụ về việc tổ chức một nốt trong mạng như sau

```
(def-category I
  (visible-atts (move cform vform auxform tense pro wh inv comp passive))
  (full-pitem-cond
    (or
      (pitem-attvec
        (or (unifiable (-passive))
          (contain (+be))))
      (pitem-preceding-word (in (be is are was were am))))))
  )
```

Đây là nốt I¹ trong mạng ngữ pháp. Nốt có các thuộc tính có thể có trong trường visible-atts. Trường full-pitem-cond chỉ ra điều kiện kết hợp của nốt này trong mạng :

¹ Đây là nốt đại diện cho các động từ tobe

nó chỉ tồn tại ở 2 dạng : kết hợp với động từ dạng bị động (passive) hoặc tồn tại trong câu mà nó đứng trước vị ngữ của câu.

Việc tổ chức các cung phức tạp hơn so với nốt bởi vì mỗi bảng mô tả không phải tương ứng với một cung mà nó chỉ đưa ra các quy tắc dịch chuyển chung với nhiều điều kiện ràng buộc khác nhau. Một ví dụ :

(modifier-cond

(pitem-attvec (and (unifiable ((vform bare)))

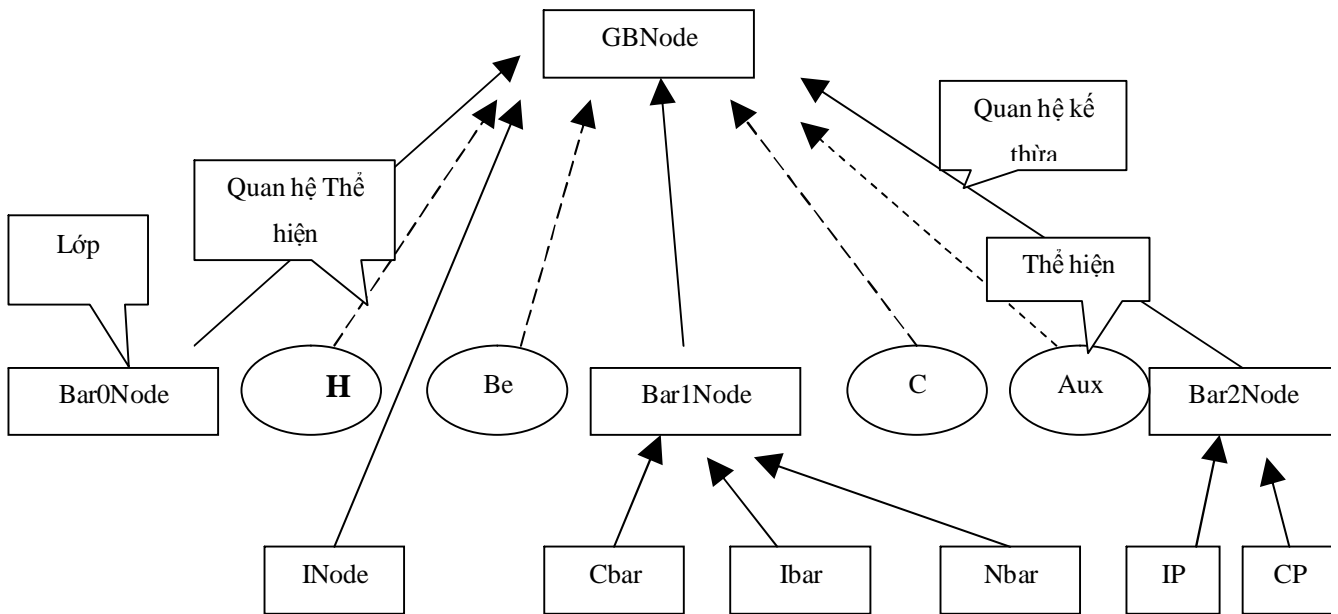
(or (unifiable (~auxform)) (unifiable ((auxform do))))

(not (contain ((move ()) ((slash wh))))))))

Đoạn ví dụ này mô tả rằng, các các trạm chỉ truyền qua được cung khi nó có thể kết hợp với động từ nguyên mẫu không to và phải hoặc là có thể kết hợp được với động từ khiếm khuyết và cuối cùng là không mang thuộc tính move loại wh.

4.1.2.3. Sơ đồ lớp

Sơ đồ lớp được tổ chức theo dạng nốt con thừa kế thuộc tính nốt cha trong mạng ngữ pháp. Ta có thể tóm tắt lại quá trình phân tích cú pháp quan hệ dựa trên nguyên tắc dựa trên sơ đồ sau:



Hình 4.1. Sơ đồ lớp các đối tượng

4.1.2.4. Kết quả đầu ra

Cây phụ thuộc được biểu diễn dưới dạng một danh sách các dòng (hay một bảng). Mỗi dòng biểu diễn một nốt trong mạng ngữ pháp ứng với một quan hệ. Tuy nhiên, bởi vì một nốt có thể có nhiều mối quan hệ nên một nốt có thể được biểu diễn nhiều dòng tương ứng với nhiều quan hệ khác nhau.

Label	Word	Root	Category	Parent's label	Relation	Parent's root	Antecedent
E1		fin	C				
1	I	I	N	2	s	want	
2	want	want	V	E1	i	fin	
E4		I	N	2	subj	want	1
E0		inf	C	2	fc	want	
E2		~	N	E0	s	inf	1
3	to	to	Aux	4	aux	book	
4	book	book	V	E0	i	inf	
E5		I	N	4	subj	book	E2
5	two	two	N	4	obj	book	
6	books	book	N	4	obj2	book	

Hình 4.2. Kết quả phân tích cú pháp quan hệ

Mỗi hàng có cấu trúc gồm 8 trường chính

Ü Label : ID của từ (hoặc từ đại diện cho mệnh đề) cùng với quan hệ đang xét. Nếu được nhắc đến lần đầu thì ID này chính là số thứ tự của từ trong câu. Nếu đây là quan hệ khác cho một từ đã xét trước đó hoặc một từ nào khác (chủ ngữ ẩn hay clause) thì nó có dạng bằng chữ En: với n là một con số.

Ü Word : từ đang xét

Ü Root : từ gốc của từ đang xét.

Ü Category : nhãn từ loại.

Ü Parent's label: ID của từ mà quan hệ này tham chiếu đến .

Ü Relation : quan hệ ngữ pháp.

Ü Parent's root : từ gốc của từ quan hệ đến.

Ü Antecedent : trong trường hợp một từ có nhiều mối quan hệ thì trong lần xuất hiện thứ 2 trở lên (quan hệ thứ 2 trở lên đối với từ này) đây là trường tham chiếu của nó đến lần xuất hiện trước đó.

4.1.3. Các thuộc tính

Cũng xuất phát từ tư tưởng nguyên tắc¹ mà kết xuất của chương trình cũng chia thành nhiều phần nhỏ, riêng biệt với nhau gọi là các thuộc tính. Có tổng cộng 64 thuộc tính² khác nhau trong đó có 38 thuộc tính là thuộc tính nhị phân. Tuy nhiên, không phải tất cả các dòng kết quả đều có đầy đủ các Kết hợp các thuộc tính theo các cách khác nhau sẽ tạo nên những kết quả khác nhau.

Ví dụ : Xét 3 thuộc tính sau:

tense : động từ thuộc thì : quá khứ, hiện tại hay tương lai.

¹ Tư tưởng nguyên tắc là ý muốn nói đến việc nhìn sự vật ở mức tổng quát hoá, là quan niệm xem vô vàn các chất chẳng qua là do kết hợp các nguyên tử mà nên.

² Xem phụ lục F

prog : có phải là thì tiếp diễn hay không

perf : có phải là thì hoàn thành hay không

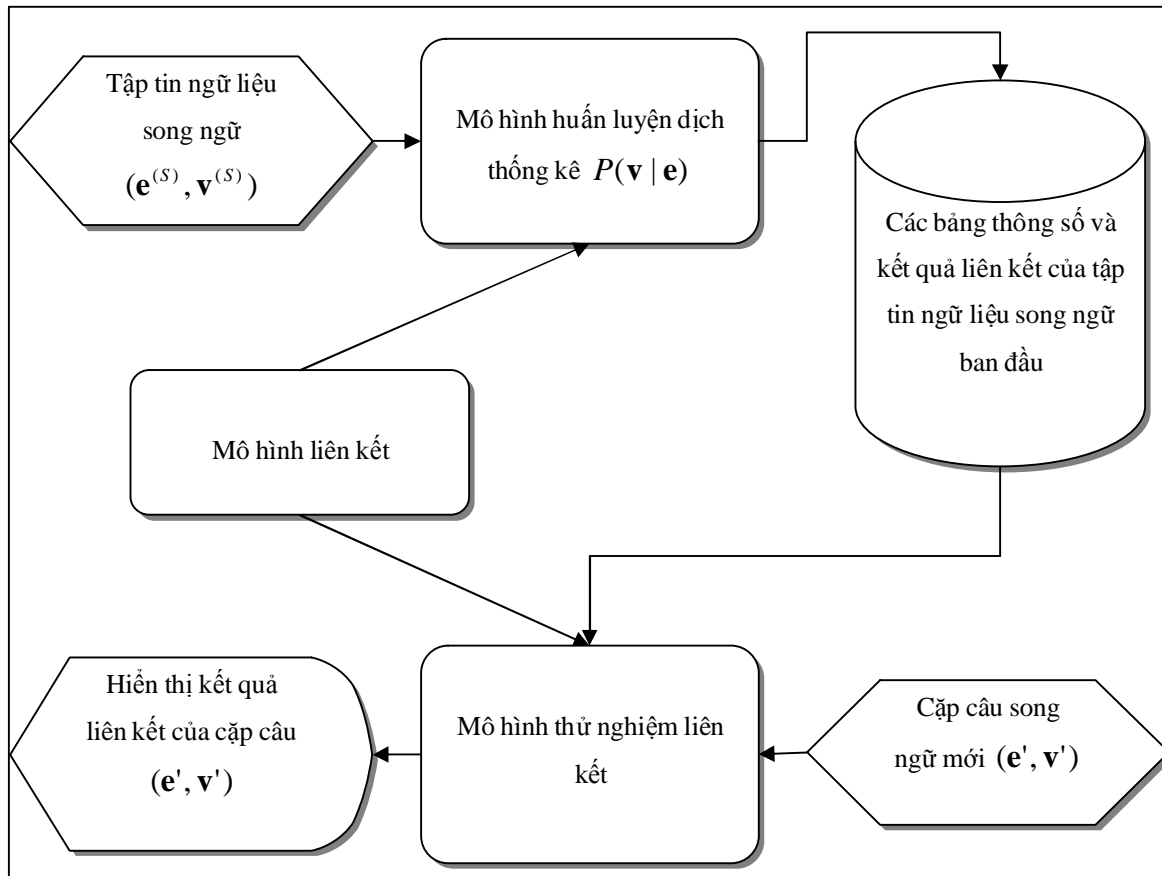
Khi kết hợp 3 thuộc tính này với nhau sẽ cho ra kết quả 12 thì khác nhau trong tiếng Anh.

4.2. Chương trình liên kết từ/ngữ

Trong phần này chúng tôi sẽ phân tích, thiết kế và cài đặt chương trình liên kết từ/ngữ trong song ngữ Anh-Việt dựa trên mô hình dịch máy thống kê.

4.2.1. Phân tích

Trong phần phân tích này chúng tôi phân tích từ một mô hình tổng quát đến mô hình chi tiết của chương trình liên kết từ/ngữ.

4.2.1.1. Phân tích tổng quát

Hình 4.3. Lưu đồ tổng quát của chương trình liên kết từ/ngữ dựa trên dịch máy thống kê

Đầu tiên chúng tôi chuẩn bị ngữ liệu song ngữ Anh-Việt ($\mathbf{e}^{(s)}, \mathbf{v}^{(s)}$), ngữ liệu song ngữ này được đưa vào mô hình huấn luyện dịch thống kê $P(\mathbf{v} | \mathbf{e})$, sau khi mô hình $P(\mathbf{v} | \mathbf{e})$ huấn luyện xong thì sẽ cho ra các bảng thông số và kết quả liên kết của tập tin ngữ liệu song ngữ ban đầu. Để thử nghiệm việc huấn luyện của mô hình dịch thống kê $P(\mathbf{v} | \mathbf{e})$ có tốt không thì chúng tôi phải thử nghiệm liên kết cho những cặp câu song ngữ mới dựa vào kết quả có được của mô hình huấn luyện.

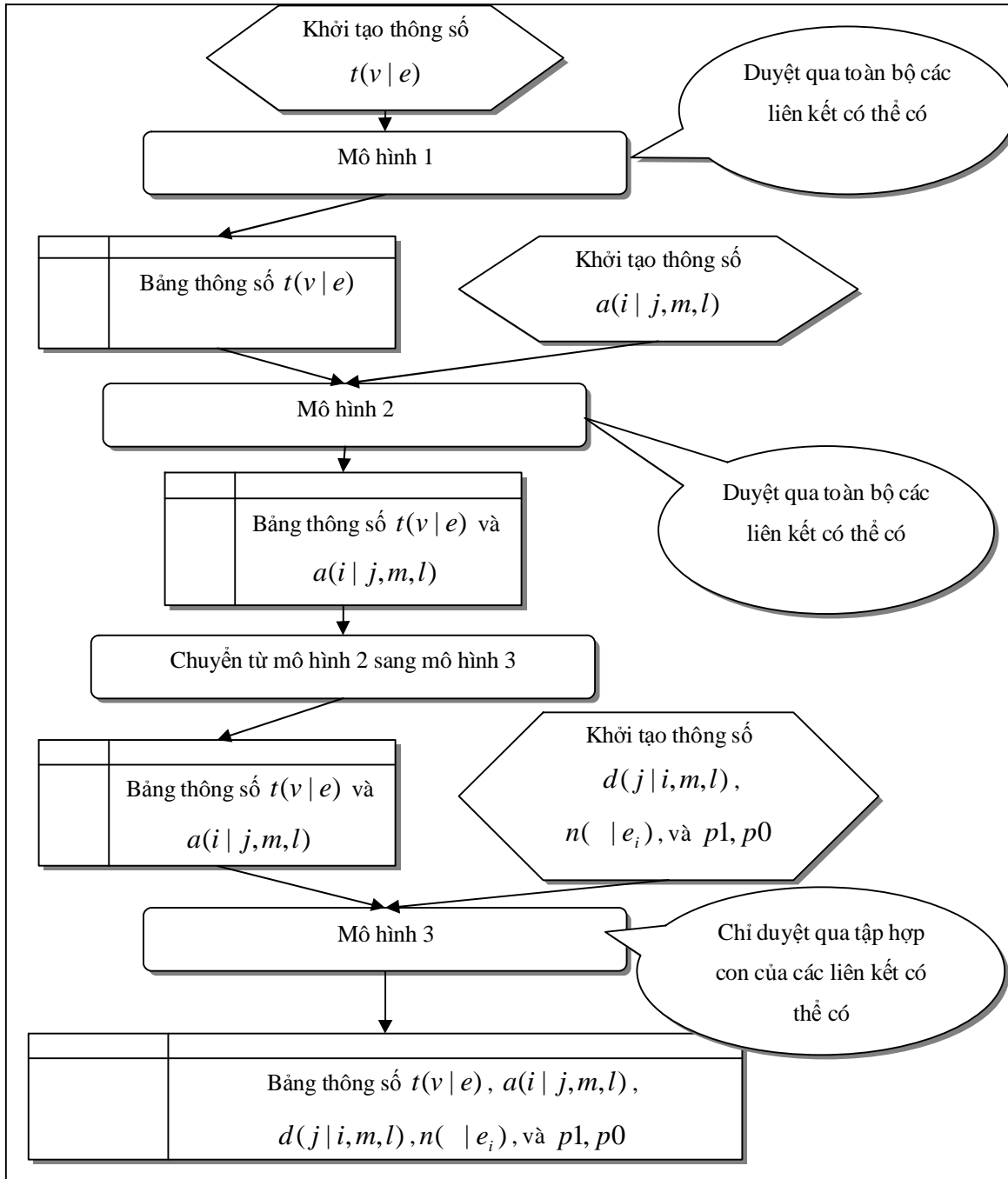
Trong mô hình huấn luyện và mô hình thử nghiệm liên kết chúng tôi lồng mô hình liên kết vào để liên kết các cặp câu song ngữ.

4.2.1.2. Phân tích chi tiết

Chương trình liên kết từ/ngữ gồm có hai mô hình chính là mô hình huấn luyện dịch thống kê $P(\mathbf{v} | \mathbf{e})$ và mô hình liên kết. Trong phần này, chúng tôi sẽ phân tích chi tiết hai mô này và cách thức hoạt động của nó.

4.2.1.2.1. Lưu đồ của mô hình huấn luyện dịch thống kê $P(\mathbf{v} | \mathbf{e})$

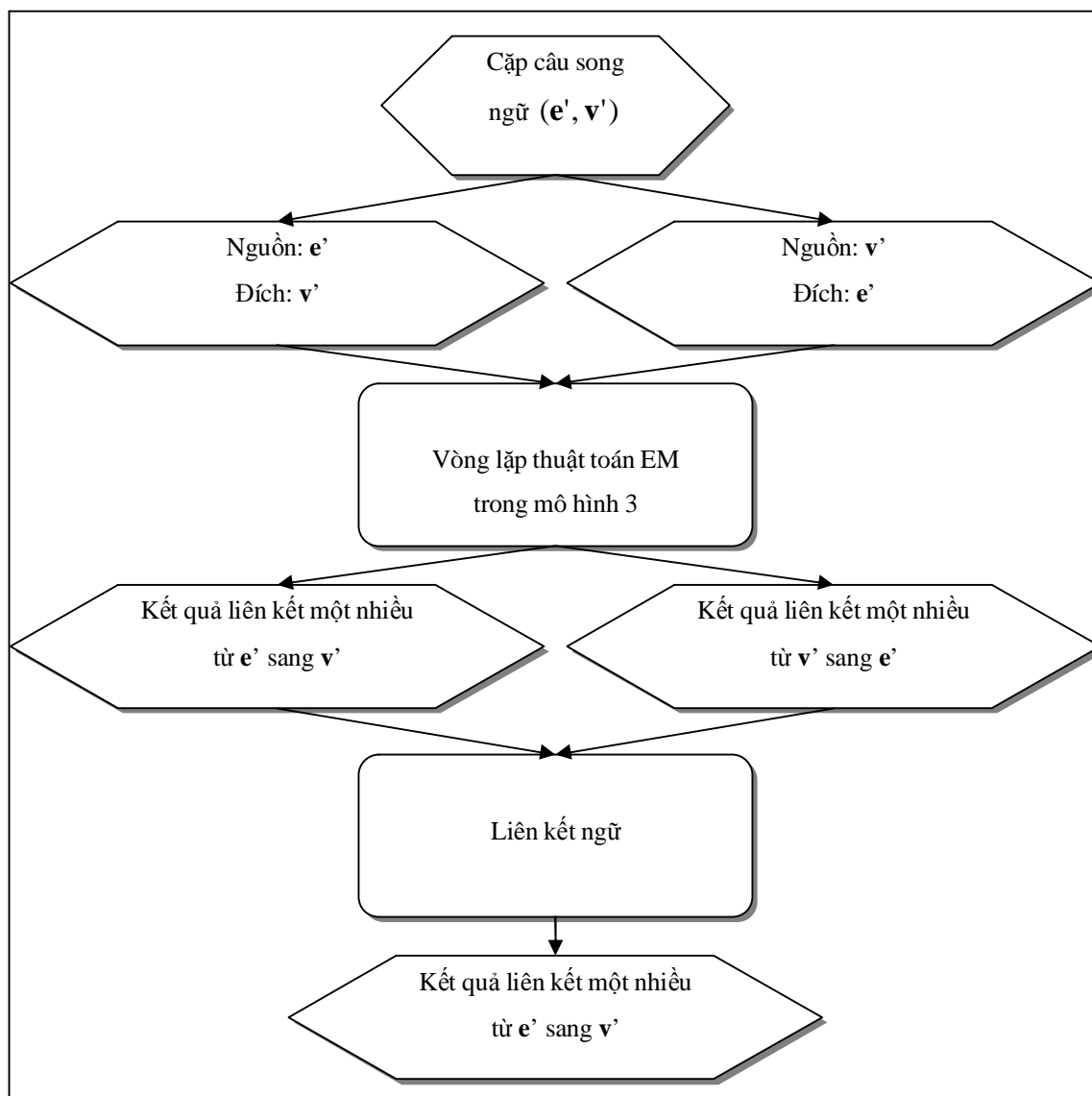
Trước tiên, chúng tôi khởi tạo thông số t cho mô hình 1, và mô hình 1 này duyệt qua toàn bộ các liên kết từ có thể có của cặp từng câu (\mathbf{e}, \mathbf{v}) và trong toàn bộ ngữ liệu song ngữ để tối ưu dần thông số t bằng các vòng lặp của thuật toán EM. Sau khi kết thúc huấn luyện thông số t trong mô hình 1 xong, thì chúng tôi được bảng thông số t tối ưu cục bộ trong mô hình 1 và tiếp theo chúng tôi khởi tạo thông số a để chuẩn bị huấn luyện trong mô hình 2 bằng các vòng lặp thuật toán EM. Sau khi kết thúc mô hình 2 thì chúng tôi được bảng thông số t và bảng thông số a tối ưu cục bộ trong mô hình 2. Tiếp theo, chúng tôi chuyển các thông số đã huấn luyện được trong mô hình 1 và 2 thành các thông số tương ứng với mô hình 3 và khởi tạo thông số d, n, p_0, p_1 để chuẩn bị huấn luyện trong mô hình 3 thông qua thuật toán EM cải tiến. Cuối cùng, sau khi kết thúc mô hình 3 thì chúng tôi thu được các thông số tối ưu cục bộ trong mô hình 3 và kết quả liên kết từ/ngữ.



Hình 4.4. Lưu đồ của mô hình huấn luyện dịch thống kê $P(v | e)$

4.2.1.2.2. Lưu đồ của mô hình liên kết ngữ

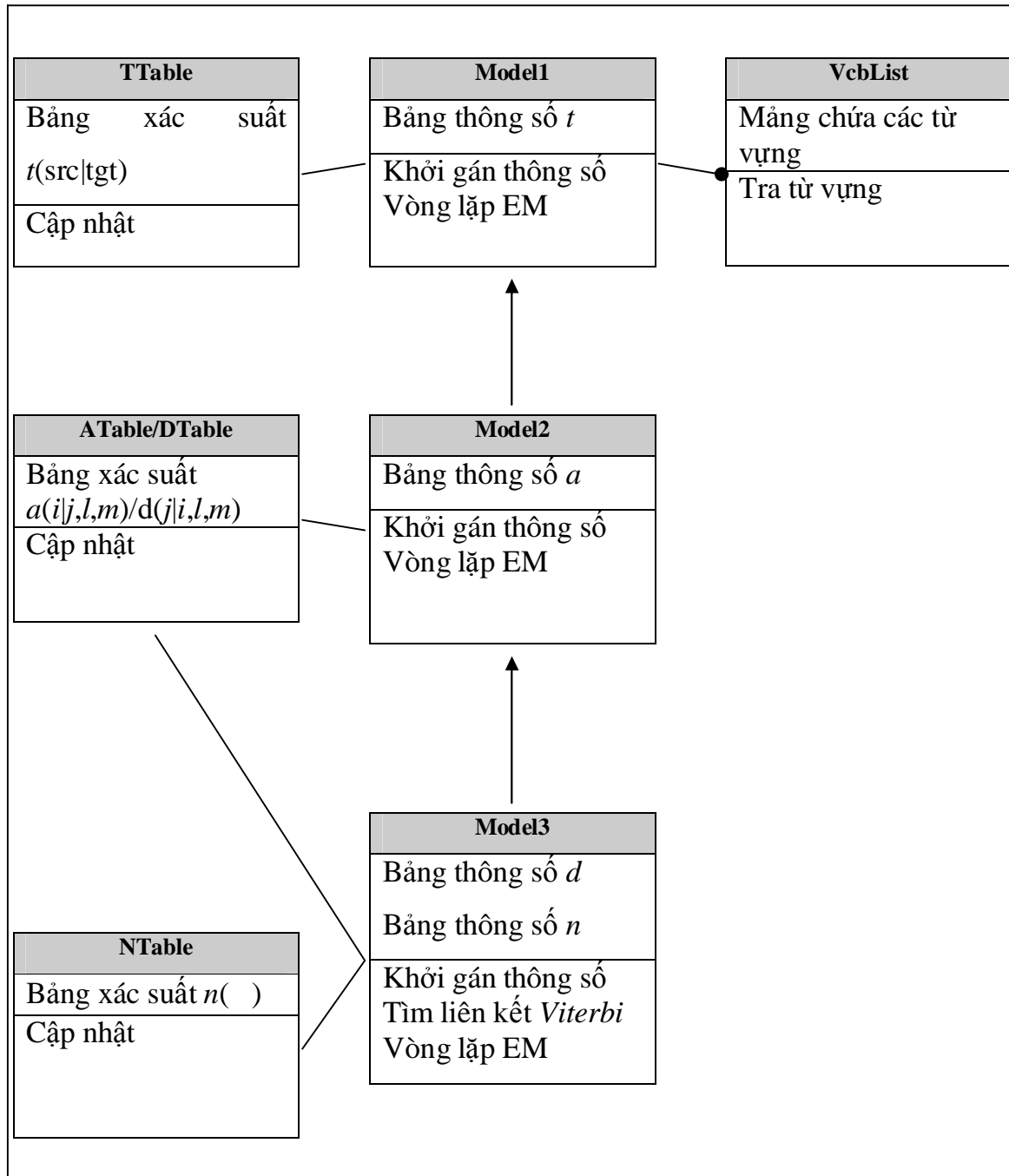
Khi đánh cặp câu song ngữ (e', v') vào thì chúng tôi chia ra làm hai trường hợp. Trường hợp 1 thì chúng tôi cho ngôn ngữ nguồn là tiếng Anh và ngôn ngữ đích là tiếng Việt. Trường hợp 2 thì chúng tôi đảo lại ngôn ngữ nguồn là tiếng Việt và ngôn ngữ đích là tiếng Anh. Trong cả hai trường hợp, chúng tôi cho cặp song ngữ Anh-Việt và Việt-Anh qua vòng lặp thuật toán EM trong mô hình 3 và được hai kết quả liên kết từ từ e' sang v' và ngược lại (từ v' sang e'). Hai kết quả này chúng tôi cho qua mô hình liên kết ngữ và được kết quả liên kết nhiều-nhiều từ e' sang v' .



Hình 4.5. Lưu đồ của mô hình liên kết

4.2.2. Thiết kế

4.2.2.1. Sơ đồ lớp



Hình 4.6. Sơ đồ quan hệ của các lớp chính

4.2.2.2. Danh sách các thuộc tính của từng lớp**Lớp TTable**

STT	Thuộc tính	Ý nghĩa
1	Bảng xác suất $t(src tgt)$	Dùng để lưu trữ các xác suất dịch từ vựng tiếng Anh e sang từ vựng tiếng Việt v

Lớp Model1

STT	Thuộc tính	Ý nghĩa
1	Bảng thông số t	Thể hiện của lớp TTable

Lớp VcbList

STT	Thuộc tính	Ý nghĩa
1	Mảng chứa các từ vựng	Dùng để chứa các từ vựng tiếng Anh hoặc tiếng Việt được mã hoá thành các con số

Lớp ATable/DTable

STT	Thuộc tính	Ý nghĩa
1	Bảng xác suất $a(i j,l,m)/d(j i,l,m)$	Dùng để lưu trữ các xác suất phụ thuộc vị trí liên kết từ câu tiếng Việt v sang câu tiếng Anh e , hoặc ngược lại

Lớp Model2

STT	Thuộc tính	Ý nghĩa
1	Bảng thông số a	Thể hiện của lớp ATable

Lớp NTable

STT	Thuộc tính	Ý nghĩa
1	Bảng xác suất $n()$	Dùng để lưu trữ các xác suất sản

		sinh của một từ vựng tiếng Anh e
--	--	------------------------------------

Lớp Model3

STT	Thuộc tính	Ý nghĩa
1	Bảng thông số d	Thể hiện của lớp DTable
2	Bảng thông số n	Thể hiện của lớp NTable

4.2.2.3. Danh sách các phương thức của từng lớp**Lớp TTable**

STT	Phương thức	Ý nghĩa
1	Cập nhật	Cập nhật các xác suất dịch trong quá trình huấn luyện

Lớp Model1

STT	Phương thức	Ý nghĩa
1	Khởi gán thông số	Dùng để khởi gán giá trị ban đầu cho thông số t và để chuẩn bị cho vòng lặp EM huấn luyện
2	Vòng lặp EM	Dùng để huấn luyện thông số t

Lớp VcbList

STT	Phương thức	Ý nghĩa
1	Tra từ vựng	Dùng để tra từ vựng từ chuỗi sang số và ngược lại

Lớp ATable/DTable

STT	Phương thức	Ý nghĩa
1	Cập nhật	Cập nhật các xác suất phụ thuộc vị trí trong quá trình huấn luyện

Lớp Model2

STT	Phương thức	Ý nghĩa
-----	-------------	---------

Chương 4: CÀI ĐẶT THỰC NGHIỆM

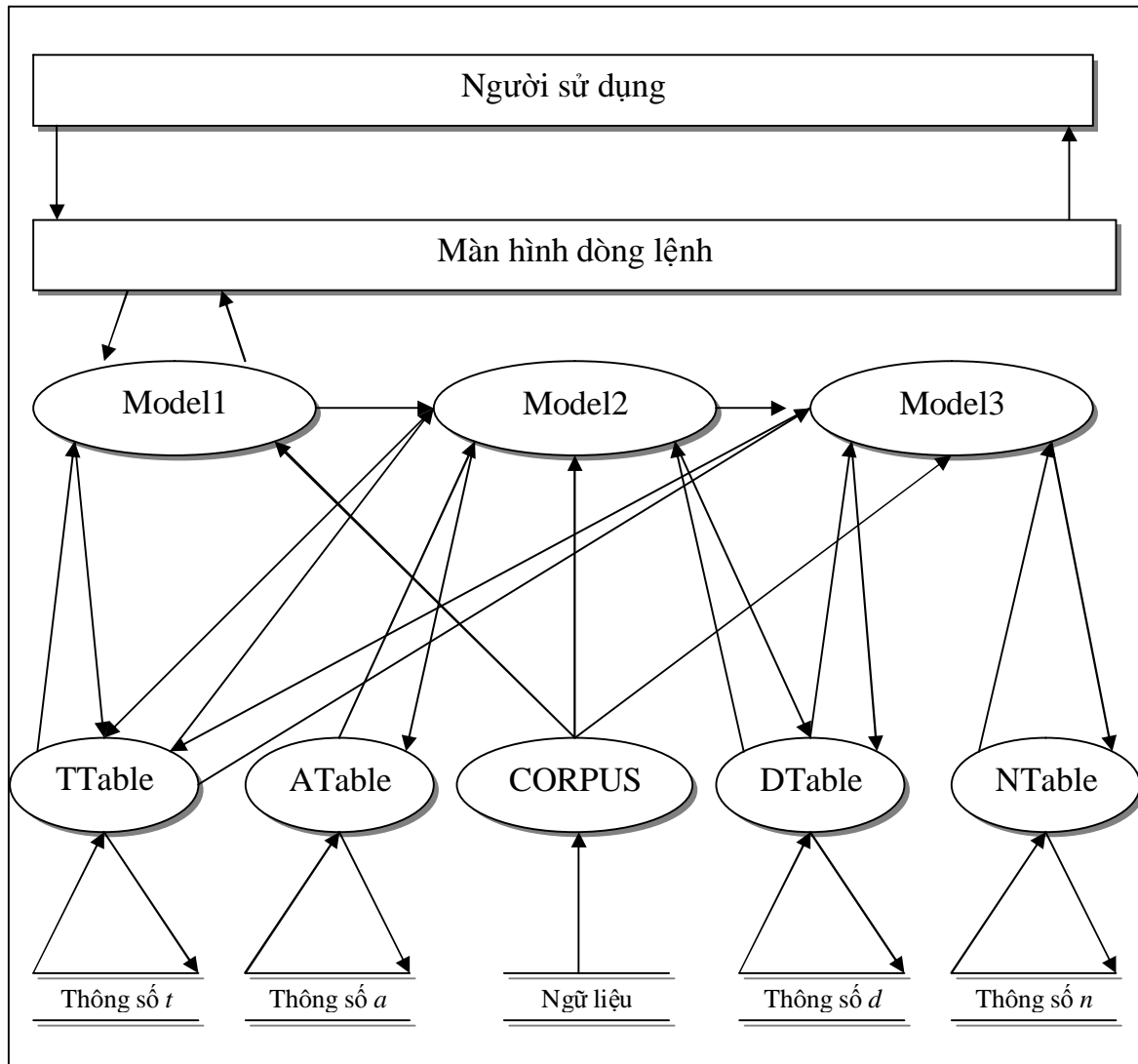
1	Khởi gán thông số	Lấy giá trị t được huấn luyện trong mô hình 1 làm giá trị ban đầu cho thông số t , khởi gán giá trị ban đầu cho thông số a và để chuẩn bị cho vòng lặp EM huấn luyện
2	Vòng lặp EM	Dùng để huấn luyện thông số t và thông số a

Lớp NTable

STT	Phương thức	Ý nghĩa
1	Cập nhật	Cập nhật các xác suất sản sinh trong quá trình huấn luyện

Lớp Model3

STT	Phương thức	Ý nghĩa
1	Khởi gán thông số	Lấy giá trị t được huấn luyện trong mô hình 2 làm giá trị ban đầu cho thông số t , chuyển thông số a trong mô hình 2 thành thông số d trong mô hình 3, khởi gán giá trị ban đầu cho thông số n và để chuẩn bị cho vòng lặp EM huấn luyện
2	Vòng lặp EM	Dùng để huấn luyện thông số t , thông số a , thông số d , và thông số n

4.2.2.4. Sơ đồ hoạt động tổng thể của các lớp cho quá trình huấn luyện

Hình 4.7. Sơ đồ hoạt động tổng thể của các lớp cho quá trình huấn luyện

Lớp Model1, Model2 và Model3 là ba lớp xử lý chính của mô hình 1, 2 và 3.

Các lớp TTable, ATable, DTable, NTable là các lớp lưu trữ và truy cập các thông số cho các mô hình xử lý chính.

Lớp CORPUS là lớp giao tiếp và truy cập các cặp câu ngữ liệu song ngữ từ các mô hình chính với đĩa cứng.

4.2.3. Cài đặt các hàm xử lý chính

4.2.3.1. Hàm khởi gán thông số t trong lớp Model1

Hàm khởi gán thông số t đơn giản gán bởi 1 con số bằng nhau cho tất cả các phần tử trong bảng thông số t . Hàm này được viết mã giả như sau

```
for (1 <= s <= S) { // Duyệt mỗi cặp câu ( $\mathbf{e}^{(s)}, \mathbf{v}^{(s)}$ ) trong ngữ liệu song ngữ
     $l$  = chiều dài của câu  $\mathbf{e}$ ;
     $m$  = chiều dài của câu  $\mathbf{v}$ ;
    uniform = 1.0/ $l$ ;
    for (0 <=  $i$  <=  $l$ ) {
        for (1 <=  $j$  <=  $m$ ) {
             $t(\mathbf{e}[i], \mathbf{v}[j])$  = uniform;
        }
    }
}
```

Hình 4.8. Hàm khởi gán t trong mô hình 1

4.2.3.2. Hàm khởi gán thông số a trong lớp Model2

```
for (1 <= s <= S) { // Duyệt mỗi cặp câu ( $\mathbf{e}, \mathbf{v}$ ) trong ngữ liệu song ngữ
     $l$  = chiều dài của câu  $\mathbf{e}$ ;
     $m$  = chiều dài của câu  $\mathbf{v}$ ;
    uniform = 1.0/( $l+1$ );
    for (1 <=  $j$  <=  $m$ )
        for (0 <=  $i$  <=  $l$ )
             $a(i, j, l, m)$  = uniform;
}
```

Hình 4.9. Hàm khởi gán thông số a trong lớp Model2

4.2.3.3. Vòng lặp EM trong lớp Model1

Vòng lặp EM trong mô hình 1 tối ưu dần thông số t và mã giả được viết như sau:

Khởi gán bảng t

Với mỗi vòng lặp {

Khởi gán các giá trị của từng phần tử trong bảng đếm tc đều bằng không

Với mỗi cặp câu (\mathbf{e}, \mathbf{v}) có chiều dài là (l, m) trong ngữ liệu song ngữ {

Với mỗi liên kết từ \mathbf{a} của (\mathbf{e}, \mathbf{v}) {

$$\text{Tính } P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^m t(v_j | e_{a_j})$$

$$\text{Tính } P(\mathbf{a} | \mathbf{e}, \mathbf{v}) = P(\mathbf{a}, \mathbf{v} | \mathbf{e}) / \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{v} | \mathbf{e})$$

Với mỗi vị trí j từ 1 đến m { // Cập nhật giá trị tc

$$tc(v_j | e_i) = P(\mathbf{a} | \mathbf{e}, \mathbf{v})$$

}

}

}

Cập nhật bảng t “mới” dựa trên tc

}

Hình 4.10. Vòng lặp EM trong lớp Model1

4.2.3.4. Vòng lặp EM trong lớp Model2

Vòng lặp EM trong mô hình 1 tối ưu dần thông số a và t . Mã giả của hàm này được viết như sau:

```

Khởi gán bảng  $t, a$ 
Với mỗi vòng lặp {
    Khởi gán các giá trị của từng phần tử trong bảng đếm  $tc, ac$  đều bằng không
    Với mỗi cặp câu  $(\mathbf{e}, \mathbf{v})$  có chiều dài là  $(l, m)$  trong ngữ liệu song ngữ {
        Với mỗi liên kết từ  $\mathbf{a}$  của  $(\mathbf{e}, \mathbf{v})$  {
            Tính  $P(\mathbf{v}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^m t(v_j | e_{a_j}) a(a_j | j, m, l)$ 
            Tính  $P(\mathbf{a} | \mathbf{e}, \mathbf{v}) = P(\mathbf{a}, \mathbf{v} | \mathbf{e}) / \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{v} | \mathbf{e})$ 
            Với mỗi vị trí  $j$  từ 1 đến  $m$  { // Cập nhật giá trị  $tc$ 
                 $tc(v_j | e_i) = P(\mathbf{a} | \mathbf{e}, \mathbf{v})$ 
            }
            Tương tự cập nhật cho bảng  $ac$ 
        }
    }
    Cập nhật bảng  $t, n$  “mới” dựa trên  $tc, ac$ 
}

```

Hình 4.11. Vòng lặp EM trong lớp Model2**4.2.3.5. Vòng lặp EM trong lớp Model3**

Vòng lặp EM trong mô hình 3 thì khác với vòng lặp EM trong mô hình 1 và 2. Trong vòng lặp EM này chúng tôi không viết lại mã giả cho việc tìm các liên kết láng giềng, vì chúng tôi đã giới thiệu trong chương 3.

Bước 1: tính liên kết tốt nhất (gọi là liên kết Viterbi) có được trong mô hình 2

$$\mathbf{a}_0: V(\mathbf{v}|\mathbf{e};2), i: 0$$

Bước 2: trong khi tồn tại một liên kết trong tập hợp láng giềng $N(\mathbf{a}_i)$ có

$$P(\mathbf{a}'|\mathbf{v},\mathbf{e};3) > P(\mathbf{a}_i|\mathbf{v},\mathbf{e};3) \text{ thì}$$

(a) đặt \mathbf{a}_{i+1} là liên kết tốt nhất trong tập hợp láng giềng $N(\mathbf{a}_i)$

(b) $i := i + 1$

Bước 3: với mỗi liên kết trong \mathbf{a} trong tập hợp láng giềng $N(\mathbf{a}_i)$

(a) tính $p := P(\mathbf{a}|\mathbf{v},\mathbf{e})$

(b) for $j:=1$ to m : tăng biến đếm xác suất sai lệch

$$c(j|\mathbf{a}_j,m,l;\mathbf{v},\mathbf{e}) := c(j|\mathbf{a}_j,m,l;\mathbf{v},\mathbf{e}) + p$$

(c) for $i:=1$ to l : tăng biến đếm xác suất sản sinh

$$c(i|\mathbf{e}_i;\mathbf{v},\mathbf{e}) := c(i|\mathbf{e}_i;\mathbf{v},\mathbf{e}) + p$$

(d) tính lại xác suất liên kết với từ rỗng p_1 và từ không rỗng p_0

$$c(0;\mathbf{v},\mathbf{e}) := c(0;\mathbf{v},\mathbf{e}) + p \quad (m \geq 2 \text{ và } 0)$$

$$c(1;\mathbf{v},\mathbf{e}) := c(1;\mathbf{v},\mathbf{e}) + p \quad 0$$

Hình 4.12. Vòng lặp EM trong lớp Model2

4.2.3.6. Tìm liên kết tối ưu nhất trong mô hình 1

Trong mô hình 1 thì chúng ta dễ dàng tìm ra một câu liên kết tối ưu nhất cho một cặp câu (\mathbf{e}, \mathbf{v}) bằng cách ta tìm xác suất của $P(v_1^m, a_1^m | e_1^l)$ là lớn nhất theo biểu thức sau

$$\hat{a}_1^m = \arg \max_{a_1^m} P(v_1^m, a_1^m | e_1^l) \quad (4.1)$$

hay

$$a_j = \arg \max_i t(v_j | e_i); \quad j = 1, \dots, l; \quad i = 0, \dots, m \quad (4.2)$$

Chúng tôi có thể viết mã giả để tìm liên kết của một cặp câu **e** và **v** cho trước như sau:

```

for (1 <= j <= m) {
    max = 0;
    i_aligned = 0; // Khởi gán liên kết với từ tiếng Anh rỗng (NULL)
    // Tìm vị trí i trong câu e sẽ được liên kết với
    for (0 <= i <= l) {
        if ( max < t( v[j] | e[i] ) ) {
            max = t( v[j] | e[i] );
            i_aligned = i; // giữ lại vị trí trong e tốt nhất
        }
    }
    a[j] = i_aligned; // Gán vị trí sẽ được liên kết
}

```

Hình 4.13. Đoạn mã tìm liên kết tối ưu nhất trong mô hình 1

4.2.3.7. Tìm liên kết tối ưu nhất trong mô hình 2

Trong mô hình 2 thì cũng tương tự như trong mô hình 1, chúng tôi tìm ra một câu liên kết tối ưu nhất cho một cặp câu (**e**, **v**) bằng cách ta tìm xác suất của $P(v_1^m, a_1^m | e_1^l)$ là lớn nhất theo biểu thức số (1) hay

$$a_j = \arg \max_i t(v_j | e_i) a(i | j, l, m); \quad j = 1, \dots, l; \quad i = 0, \dots, m \quad (4.3)$$

Chúng tôi có thể viết mã giả để tìm liên kết của một cặp câu **e** và **v** cho trước như sau:

```

for (1 <= j <= m) {
    max = 0;
    i_aligned = 0; // Khởi gán liên kết với từ tiếng Anh rỗng (NULL)
    // Tìm vị trí i trong câu e sẽ được liên kết với
    for (0 <= i <= l) {
        if ( max < t( v[j] | e[i] ) * a( i | j, l, m ) ) {
            max = t( v[j] | e[i] ) * a( i | j, l, m );
            i_aligned = i; // giữ lại vị trí trong e tốt nhất
        }
    }
    a[j] = i_aligned; // Gán vị trí sẽ được liên kết
}

```

Hình 4.14. Đoạn mã tìm liên kết tối ưu nhất trong mô hình 2**4.2.3.8. Tìm liên kết tối ưu nhất trong mô hình 3**

Trong thuật toán EM được áp dụng trong mô hình 3 cũng đã lồng công việc tìm liên kết tối ưu nhất bằng thuật toán “Leo đồi”. Trong mỗi bước nhỏ thì thuật toán kiểm tra điều kiện $P(\mathbf{a} | \mathbf{v}, \mathbf{e}; 3) > P(\mathbf{a}_i | \mathbf{v}, \mathbf{e}; 3)$, trong một bước nào đó mà không thỏa điều kiện này thì có nghĩa \mathbf{a}_i là liên kết tối ưu nhất. Do đó, trong phần này chúng tôi không viết lại cách tìm liên kết tối ưu nhất.

4.3. Chiếu kết quả phân tích cú pháp sang Tiếng Việt**4.3.1. Chiếu nhãn từ loại**

Trước khi tiến hành chiếu kết quả từ loại từ Anh sang Việt, điều đầu tiên là xây dựng một phép ánh xạ từ vựng giữa 2 ngôn ngữ. Không may thay, sự cách biệt khá lớn giữa hai ngôn ngữ làm cho không những khái niệm từ vựng của hai ngôn ngữ khác

nhau mà khi chiếu sang thì từ loại cũng không được bảo toàn. Do đó, bảng ánh xạ từ vựng này là một phép ánh xạ m-n¹.

Theo mô hình xác suất đã được trình bày, công thức tính xác suất của một từ loại được tính theo công thức sau.

$$P(T_{\text{Việt}}) = P_1(T_{\text{Việt}}) * P(T_{\text{Eng}} \rightarrow T_{\text{Việt}})$$

Với $P(T_{\text{Việt}})$: xác suất nhận $T_{\text{Việt}}$ cho từ tiếng Việt đang xét.

$P_1(T_{\text{Việt}})$: xác suất xuất hiện nhãn $T_{\text{Việt}}$ trong tổng số lần xuất hiện của từ tiếng Việt.

$P(T_{\text{Eng}} \rightarrow T_{\text{Việt}})$: xác suất nhãn T_{Eng} được ánh xạ sang nhãn $T_{\text{Việt}}$.

Một khó khăn gặp phải là làm thế nào biết được các xác suất này khi không có trong tay một ngữ liệu nào bằng tiếng Việt là tuyệt đối chính xác và đủ lớn. Vấn đề này sẽ được nói đến trong phần tiếp theo.

4.3.2. Chiếu quan hệ

Tiếp theo, các quan hệ được chiếu trực tiếp sang tiếng Việt theo nguyên lý DCA. Thử phân tích một ví dụ :

Câu tiếng Anh : My₁ father₂ is₃ a₄ teacher₅

Câu tiếng Việt : Cha₁ của₂ tôi₃ là₃ một₄ giáo_viên₅

Và mỗi liên kết ngữ [1_2,3] [2_1] [3_3] [4_4] [5_5]

Vì giáo viên là một từ trong Tiếng Việt nên chúng tôi đã nối lại tạm xem nó là một từ (từ cách nhau bằng khoảng trắng).

Xét một quan hệ trong câu tiếng Anh : father có mối quan hệ subj với teacher. Thông qua mỗi liên kết từ, ta xác định được 2 từ tương ứng trong tiếng Việt là Cha và giáo viên với mối quan hệ tương ứng là chủ từ.

¹ Ánh xạ m-n có thể hiểu theo 2 nghĩa khác nhau : theo số lượng từ hoặc theo số lượng từ vựng (một từ vựng bên ngôn ngữ này tương ứng với nhiều từ loại trong ngôn ngữ kia). Quan hệ số lượng từ được giải quyết trong phần liên kết ngữ nên ở đây chỉ quan tâm đến quan hệ m-n về mặt số lượng từ vựng

4.3.3. Sử dụng luật tương tác

Sau khi đã được gán nhãn từ loại và quan hệ, câu tiếng Việt sẽ được áp dụng các luật tương tác được rút ra từ quá trình học TBL.

Tuy nhiên, để rút ra được bộ luật tương tác này, chúng tôi đã tiến hành quá trình vừa học vừa sửa chữa. Nguyên nhân chính là chúng tôi không có một ngữ liệu đủ lớn và chính xác để tiến hành học.

Bộ luật sẽ được khởi tạo bằng một số luật được rút ra từ kinh nghiệm. Thực tế cho thấy rằng, điều này làm nâng cao cả tốc độ hội tụ lẫn độ chính xác, hợp lí của các luật rút ra.

Đầu tiên, thông qua liên kết từ, các quan hệ trong một văn bản không nhiều các câu tiếng Anh sẽ chiếu trực tiếp sang tiếng Việt. Các quan hệ này sẽ được chỉnh sửa bằng các luật hiện có trong bộ luật. Sau đó các quan hệ này sẽ được sửa bằng tay tạo thành một ngữ liệu vàng có kích thước nhỏ. Sử dụng ngữ liệu này, thuật toán học TBL sẽ rút ra một số luật tương tác. Các luật này được thêm vào bộ luật. Áp dụng luật này vào chương trình để đánh một lượng văn bản lớn hơn. Bây giờ, kết quả của chương trình sẽ được tăng lên một ít (nhưng không nhiều do ngữ liệu học còn quá ít).

Quá trình học-chỉnh sửa-áp dụng cứ lặp đi lặp lại với khối lượng văn bản ngày càng nhiều và tỉ lệ chính xác ngày càng tăng (vì ngữ liệu lớn sẽ dẫn đến tính tổng quát của luật càng cao). Khi văn bản đã đạt đến một độ lớn nào đó, quá trình chỉnh sửa bằng tay sẽ ngừng lại. Lúc này, chương trình sẽ tự học ra luật rồi lại áp dụng vào chương trình theo kiểu “tự ta sẽ dạy ta”. Để hạn chế điều này, bộ luật được xem xét, chọn lựa bằng tay sau mỗi lần rút ra một bộ luật mới.

Chương 5: KẾT QUẢ - ĐÁNH GIÁ – KẾT LUẬN – HƯỚNG PHÁT TRIỂN

5.1. Chương trình liên kết từ

Chúng tôi đã thu thập một số lượng lớn những cặp câu song ngữ từ nhiều nguồn lĩnh vực khác nhau để làm ngữ liệu huấn luyện các tham số của những mô hình mà chúng tôi đã đề cập trong phần mô hình thuật toán. Các nguồn mà chúng tôi thu thập là:

“Hãy đến với thế vi tính”, Nhà xuất bản Thống kê, do Cadasa biên tập – thuộc lĩnh vực khoa học tin học.

DeDX, những cặp câu song ngữ ví dụ của từ điển Anh-Việt.

LLOCE (Longman Lexicon of Contemporary English).

Susanne (Surface Understudy Analysis Natural English).

5.1.1. Một số kết quả

Trong phần này, chúng tôi trình bày kết quả một số cặp câu mẫu được lấy trong ngữ liệu Cadasa, DeDX, LLOCE và Susanne.

Chúng tôi trình bày kết quả của liên kết từ giống như qui ước trình bày kết quả trong chương 2 mà chúng tôi đã giới thiệu.

Câu tiếng Anh	Câu tiếng Việt	Kết quả
What ₁ is ₂ an ₃ internet ₄	Một ₁ máy_chủ ₂ internet ₃ là ₄	[1_5] [2_4] [3_1] [4_3]

Chương 5: KẾT QUẢ - ĐÁNH GIÁ – KẾT LUẬN – HƯỚNG PHÁT TRIỂN

host ₅ ? ₆	gì ₅ ? ₆	[5_2] [6_6]
Your ₁ instructor ₂ may ₃ direct ₄ you ₅ to ₆ select ₇ different ₈ options ₉ . ₁₀	Giáo_viên ₁ của ₂ bạn ₃ có_thể ₄ hướng_dẫn ₅ bạn ₆ lựa_chọn ₇ các ₈ tùy ₉ chọn ₁₀ khác_nhau ₁₁ . ₁₂	[1_2,3] [2_1] [3_4] [4_5] [5_6] [7_7] [8_11] [9_8,9,10] [10_12]
Name ₁ five ₂ graphic ₃ elements ₄ commonly ₅ found ₆ in ₇ web ₈ pages ₉ .10	Nêu_tên ₁ năm ₂ yếu_tố ₃ đồ_họa ₄ thường ₅ thấy ₆ trên ₇ các ₈ trang ₉ web ₁₀ . ₁₁	[1_1] [2_2] [3_4] [4_3] [5_5] [6_6] [7_7] [8_10] [9_8,9] [10_11]
Creating ₁ a ₂ digital ₃ image ₄ or ₅ manipulating ₆ an ₇ existing ₈ one ₉ can ₁₀ involve ₁₁ a ₁₂ complex ₁₃ array ₁₄ of ₁₅ processes ₁₆ . ₁₇	Tạo_ra ₁ một ₂ ảnh ₃ kỹ ₄ thuật_số ₅ hay ₆ chế_tác ₇ từ ₈ một ₉ ảnh ₁₀ có_sẵn ₁₁ có_thể ₁₂ liên_quan ₁₃ đến ₁₄ một ₁₅ chuỗi ₁₆ phức_tạp ₁₇ các ₁₈ công_việc ₁₉ . ₂₀	[1_1] [2_2] [3_4,5] [4_3] [5_6] [6_7,8] [7_9] [8_11] [10_12] [11_13,14] [12_15] [13_17] [14_16] [16_18,19] [17_20]
This ₁ openness ₂ has ₃ attracted ₄ tens ₅ of ₆ millions_of ₇ users ₈ to ₉ the ₁₀ internet ₁₁ . ₁₂	Tính ₁ mở của ₃ internet ₄ đã ₅ hấp_dẫn ₆ hàng_chục ₇ triệu ₈ người ₉ dùng ₁₀ . ₁₁	[2_1,2] [3_5] [4_6] [5_7] [7_8] [8_9,10] [11_4] [12_11]
Users ₁ are ₂ going ₃ online ₄ for ₅ wide ₆ variety ₇ of ₈ reasons ₉ . ₁₀	Những ₁ người ₂ dùng ₃ kết_nối ₄ trực_tuyến ₅ vì ₆ nhiều ₇ lý_do ₈ khác_nhau ₉ .10	[1_1,2,3] [4_4,5] [7_9] [9_8] [10_10]
The ₁ internet ₂ itself ₃ is ₄ the ₅ pipeline ₆ that ₇ carries ₈ data ₉ between ₁₀ computers ₁₁ . ₁₂	Bản_thân ₁ internet ₂ là ₃ một ₄ kênh ₅ truyền ₆ vận_chuyển ₇ các ₈ dữ_liệu ₉ giữa ₁₀ các ₁₁ máy_tính ₁₂ . ₁₃	[2_2] [3_1] [4_3] [6_5,6] [8_7] [9_8,9] [10_10] [11_11,12] [12_13]

Chương 5: KẾT QUẢ - ĐÁNH GIÁ – KẾT LUẬN – HƯỚNG PHÁT TRIỂN

Most ₁ computers ₂ are ₃ not ₄ connected ₅ directly ₆ to ₇ the ₈ internet ₉ . ₁₀	đa_phần ₁ các ₂ máy_tính ₃ đều ₄ không ₅ nối_kết ₇ trực_tiếp ₈ tới ₉ internet ₁₀ . ₁₁	[1_1] [2_2,3] [3_4] [4_5] [5_6,7] [6_8] [9_10] [10_11]
--	---	--

Bảng 5.1. Một số kết quả liên kết từ trong ngữ liệu Cadasa.

Câu tiếng Anh	Câu tiếng Việt	Kết quả
The ₁ sophisticates ₂ in ₃ the ₄ office ₅ drink ₆ lemon ₇ tea ₈ , ₉ we ₁₀ have ₁₁ coffee ₁₂ . ₁₃	những ₁ người ₂ sành ₃ điệu ₄ trong ₅ văn_phòng ₆ này ₇ uống ₈ trà ₉ chanh ₁₀ , ₁₁ còn ₁₂ chúng_tôi ₁₃ uống ₁₄ cà_phê ₁₅ .16	[1_1] [2_2,3,4] [3_5] [4_7] [5_6] [6_8] [7_10] [8_9] [9_11] [10_13] [12_14,15] [13_16]
He ₁ has ₂ had ₃ many ₄ sorrows ₅ in ₆ his ₇ life ₈ . ₉	anh_ấy ₁ có ₂ nhiều ₃ nỗi ₄ bất_hạnh ₅ trong ₆ đời ₇ . ₈	[1_1] [2_2] [4_3] [5_4,5] [6_6] [8_7] [9_8]
It ₁ was ₂ smart ₃ of ₄ you ₅ to ₆ bring ₇ a ₈ map ₉ . ₁₀	bạn ₁ mang ₂ bản_đồ ₃ theo ₄ thật ₅ là ₆ khôn_ngoan ₇ . ₈	[1_5] [2_6] [3_7] [5_1] [7_2] [9_3] [10_8]
We ₁ have_to ₂ spare ₃ room ₄ for ₅ a ₆ table ₇ . ₈	chúng_tôi ₁ không ₂ còn ₃ chỗ_nào ₄ thừa ₅ để ₆ kê ₇ một ₈ chiếc ₉ bàn ₁₀ . ₁₁	[1_1] [2_2] [3_3,4,5] [5_6] [6_8] [7_9,10] [8_11]
I ₁ wish ₂ we ₃ had ₄ a ₅ spare ₆ room ₇ . ₈	Tôi ₁ ước ₂ gì ₃ chúng_tôi ₄ có ₅ thêm ₆ một ₇ phòng ₈ . ₉	[1_1] [2_2,3] [3_4] [4_5] [5_7] [6_6] [7_8] [8_9]
She ₁ 's ₂ very ₃ special ₄ friend ₅ . ₆	Cô ₁ ta ₂ là ₃ một ₄ người_bạn ₅ rất ₆ đặc_biệt ₇ . ₈	[1_1,2] [2_3] [3_6] [4_7] [5_5] [6_8]
My ₁ speculations ₂ proved ₃ totally ₄ wrong ₅ . ₆	những ₁ điều ₂ suy_đoán ₃ của ₄ tôi ₅ tỏ_ra ₆ hoàn_toàn ₇ sai_lầm ₈ . ₉	[1_4,5] [2_1,2,3] [3_6] [4_7] [5_8] [6_9]
The ₁ runner ₂ spurted ₃ as ₄ he ₅ approached ₆ the ₇ line ₈ .9	vận_động_viên ₁ chạy ₂ tăng ₃ tốc_độ ₄ khi ₅ anh ₆ ta ₇ đến_gần ₈ đích ₉ . ₁₀	[2_1,2] [3_3,4] [4_5] [5_6,7] [6_8] [9_10]

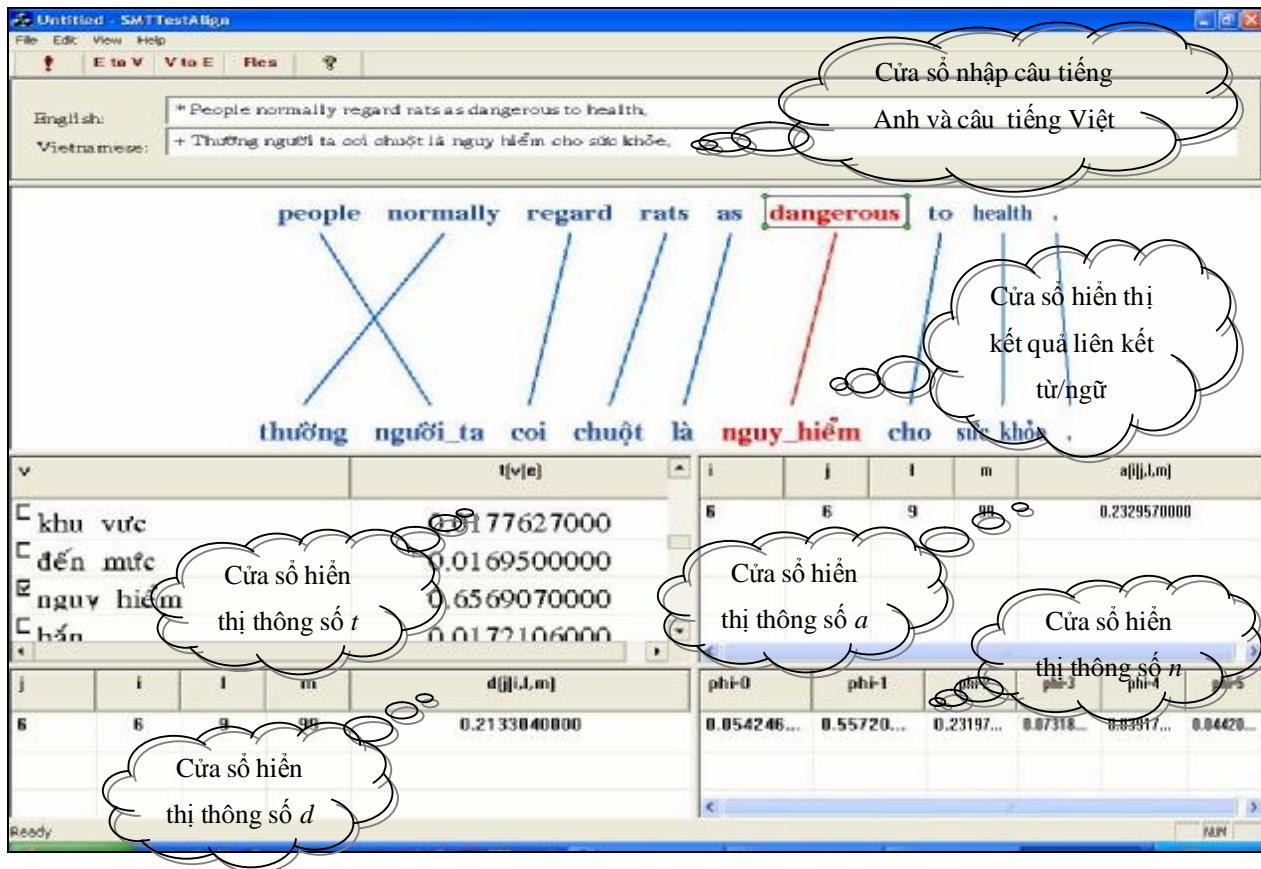
Chương 5: KẾT QUẢ - ĐÁNH GIÁ – KẾT LUẬN –HƯỚNG PHÁT TRIỂN

Bảng 5.2. Một số kết quả liên kết từ trong ngữ liệu DeDX.

Câu tiếng Anh	Câu tiếng Việt	Kết quả
He ₁ could ₂ see ₃ some ₄ men ₅ and ₆ horses ₇ on ₈ the ₉ hillside ₁₀ . ₁₁	Cậu ₁ ấy ₂ có_thể ₃ trông_thấy ₄ một_số ₅ đàn_ông ₆ và ₇ ngựa ₈ bên ₉ sườn ₁₀ đồi ₁₁ . ₁₂	[1_1,2] [2_3] [3_4] [4_5] [5_6] [6_7] [7_8] [8_9] [10_10,11]
Man ₁ and ₂ the ₃ monkey ₄ have ₅ many ₆ things ₇ in ₈ common ₉ . ₁₀	Loài_người ₁ và ₂ loài ₃ khi ₄ có ₅ nhiều ₆ điểm ₇ tương_đồng ₈ . ₉	[1_1] [2_2] [4_3,4] [5_5] [6_6] [7_8] [10_7] [11_9]
I ₁ 'm ₂ sure ₃ there ₄ are ₅ mice ₆ in ₇ that ₈ old ₉ house ₁₀ . ₁₁	Tôi ₁ tin ₂ là ₃ có ₄ chuột ₅ trong ₆ căn_nhà ₇ cũ_nát ₈ đó ₉ . ₁₀	[1_1] [2_3] [3_2] [4_4] [6_5] [7_6] [8_9] [9_8] [10_7] [11_10]
People ₁ normally ₂ regard ₃ rats ₄ as ₅ dangerous ₆ to ₇ health ₈ . ₉	Thường ₁ người_ta ₂ coi ₃ chuột ₄ là ₅ nguy_hiểm ₆ cho ₇ sức_khỏe ₈ . ₉	[1_2] [2_1] [3_3] [4_4] [5_5] [6_6] [7_7] [8_8] [9_9]


Bảng 5.3. Một số kết quả liên kết từ trong ngữ liệu LLOCE.

5.1.2. Giao diện của chương trình thử nghiệm liên kết



Hình 5.1 Giao diện của chương trình thử nghiệm liên kết.

Giao diện thử nghiệm liên kết sử dụng các kết quả của các thông số sau khi huấn luyện để liên kết các cặp câu mới. Giao diện đơn giản, dễ sử dụng. Giao diện bao gồm 6 cửa sổ chính: cửa sổ nhập câu tiếng Anh và câu tiếng Việt, cửa sổ hiển thị kết quả liên kết từ/ngữ, cửa sổ hiển thị thông số t , cửa sổ hiển thị thông số a , cửa sổ hiển thị thông số d và cửa sổ hiển thị thông số n .

Trên thanh công cụ có 4 nút chức năng chính là: nút chạy chương trình thử liên kết , nút hiển thị kết quả liên kết từ câu tiếng Anh sang câu tiếng Việt **E to V**, nút hiển thị kết quả liên kết từ câu tiếng Việt sang câu tiếng Anh **V to E** và nút hiển thị kết quả cuối cùng **Res**.

5.1.3. Đánh giá

Chúng tôi đã chạy thử nghiệm trên từng ngữ liệu liệt kê ở trên và mỗi nguồn có một kết quả được liệt kê trong bảng sau:

Tên ngữ liệu	Số cặp câu	Số từ vựng tiếng Anh	Số từ vựng tiếng Việt	Kết quả đúng (%)
Cadasa	8583	7634	5806	81,2
DeDX	92732	23540	19254	85,4
LLOCE	28741	3254	2654	73,5
Susanne	4739	5423	6543	79,7

Bảng 5.4. Kết quả của liên kết từ/ngữ dựa trên bốn ngữ liệu khác nhau

Ngoài ra, chúng tôi còn kết hợp hai ngữ liệu Cadasa và DeDX thì kết quả thật bất ngờ 90,2% đúng so với ngữ liệu vàng, trong đó kết quả liên kết của Cadasa đúng 93,3% và của DeDX là 87,1%. Có kết quả này thì cũng dễ hiểu bởi vì mô hình huấn luyện dựa trên mô hình thống kê, do đó nếu ngữ liệu càng nhiều thì độ chính xác của liên kết từ/ngữ càng cao.

Ngoài cách đánh giá chúng tôi còn đánh giá kết quả của chương trình thông qua độ phức tạp và thời gian huấn luyện của thuật toán qua từng vòng lặp EM của mỗi mô hình huấn luyện. Độ phức tạp (Perplexity) là một hàm đánh giá độ chính xác gần đúng dựa trên xác suất. Đối với độ phức tạp huấn luyện thì chúng tôi dựa vào hàm đánh giá xác suất dịch từ một câu tiếng Anh sang một câu tiếng Việt $P(\mathbf{v}^{(s)} | \mathbf{e}^{(s)})$ qua hàm đánh giá như sau

$$Perplexity = 2^{\frac{\sum_{s=1}^S \log_2 P(\mathbf{v}_s | \mathbf{e}_s)}{N}} \quad (5.1)$$

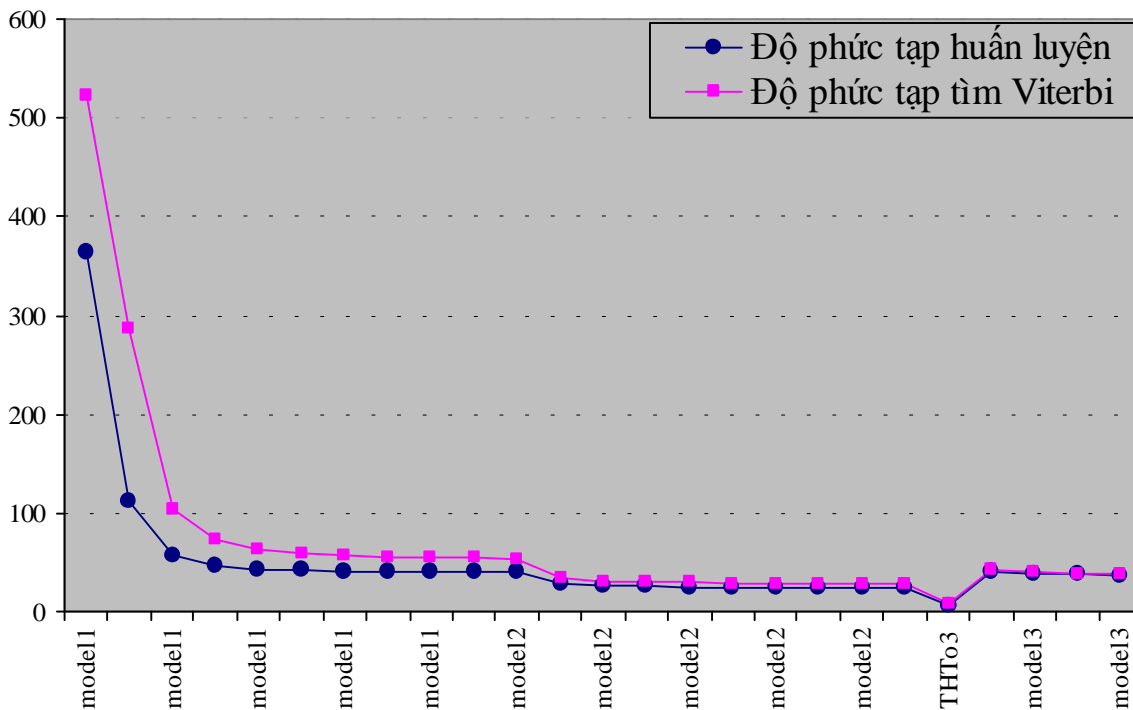
với S là tổng số cặp câu trong ngữ liệu huấn luyện, N là tổng số từ tiếng Anh và tiếng Việt trong ngữ liệu, *Perplexity* là giá trị của độ phức tạp.

Nếu chúng ta muốn xác suất dịch một cặp câu cao thì khi đó độ phức tạp *Perplexity* phải thấp.

Đối với độ phức tạp tìm liên kết tốt cho mỗi cặp câu trong ngữ liệu chúng tôi dựa vào hàm đánh giá xác suất liên kết của một cặp câu (\mathbf{e}, \mathbf{v}) cho sẵn qua hàm đánh giá như sau

$$Perplexity = 2^{\frac{\sum_{s=1}^S \log_2 P(\mathbf{a}|\mathbf{v}^s, \mathbf{e}^s)}{N}} \quad (5.2)$$

Hình sau biểu diễn biểu đồ đánh giá độ phức tạp của thuật toán EM qua mỗi vòng lặp với ngữ liệu huấn luyện là Cadasa và DeDX.

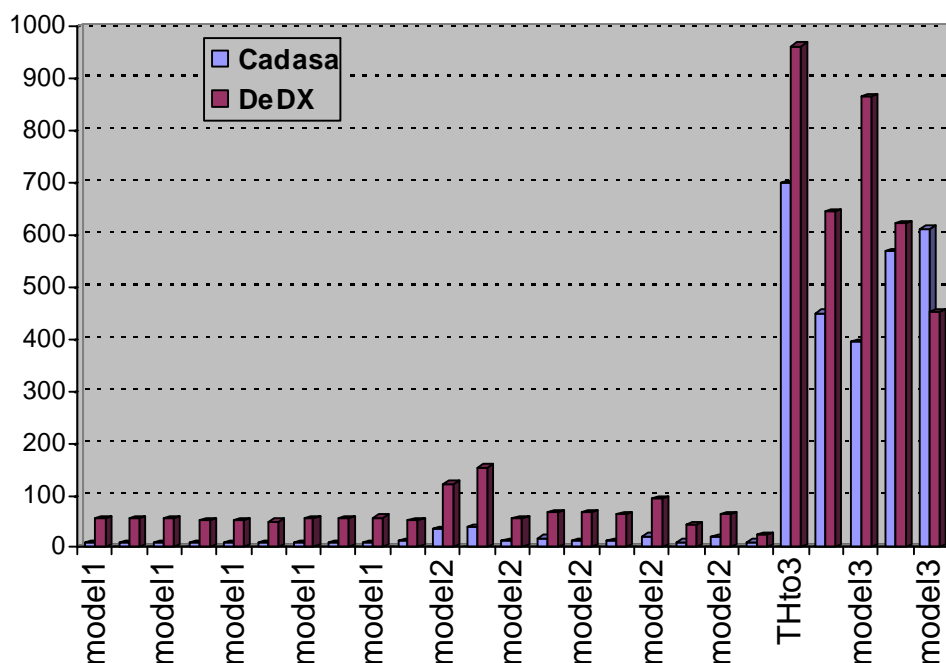


Hình 5.2. Biểu đồ độ phức tạp của quá trình huấn luyện và quá trình tìm liên kết tối ưu nhất (liên kết Viterbi) của ngữ liệu huấn luyện Cadasa kết hợp với ngữ liệu DeDX. Trục ngang biểu diễn mỗi vòng lặp, trục đứng biểu diễn độ phức tạp.

Qua biểu đồ trên, chúng tôi có nhận xét là: đường biểu diễn độ phức tạp giảm nhanh qua các vòng lặp huấn luyện của thuật toán EM trong mô hình 1 (model 1) và mô hình

2 (model 2) nhưng đến mô hình 3 thì độ phức tạp lại tăng lên. Điều đó có nghĩa là các vòng lặp EM trong mô hình 1 và 2 thì có hiệu quả hơn là các vòng lặp trong mô hình 3. Do đó, số vòng lặp EM trong mô hình 1 và 2 chúng tôi cho nhiều hơn số vòng lặp trong mô hình 3.

Chúng tôi còn đánh về mặt thời gian của thuật toán EM qua mỗi vòng lặp. Chúng tôi ghi nhận được thời gian huấn luyện trên ngữ liệu Cadasa và DeDX của thuật toán EM qua mỗi vòng lặp như biểu đồ sau



Hình 5.3. Biểu đồ thời của quá trình huấn luyện của ngữ liệu huấn luyện Cadasa với ngữ liệu DeDX.

Trục ngang biểu diễn mỗi vòng lặp, trục đứng biểu diễn thời gian được tính bằng đơn vị giây.

Qua biểu đồ này chúng ta thấy rằng các vòng lặp của thuật toán EM trong mô hình 1 và mô hình 2 có thời gian chạy ít hơn nhiều so với thời gian chạy của các vòng lặp trong thuật toán EM trong mô hình 3. Một lần nữa chúng ta thấy rằng độ phức tạp của thuật toán EM trong mô hình 1 và mô hình 2 đơn giản hơn mô hình 3. Ngoài ra, chúng ta còn nhận thấy rằng thời gian các vòng lặp của thuật EM trong mô hình 1 và 2 tương đối ổn định hơn trong các vòng lặp của mô hình 3. Điều này chứng tỏ rằng thuật toán

EM trong mô hình 3 không ổn định do áp dụng thuật toán “Leo đồi” mà chúng tôi đã giới thiệu trong chương mô hình thuật toán.

5.2. Chương trình phân tích quan hệ cú pháp

5.2.1. Kết quả

Chương trình có khả năng đánh nhãn từ loại và xác định các quan hệ trong câu. Ngoài thông tin về quan hệ giữa những từ trong câu, chương trình còn có thể rút trích được các thông tin về từ gốc của từ, nhãn từ loại của từ, loại mệnh đề, thì của động từ trong câu...

Label	Word	Root	Category	Parent's label	Relation	Parent's root	Antecedent
E1			U	*			
E0		fin	C	*			
1	All	.	PreDet	2	pre	animal	
2	animals	animal	N	3	s	need	
3	need	.	V	E0	i	fin	
E2		animal	N	3	subj	need	2
4	water	.	N	3	obj	need	

Hình 1 Minh hoạ kết quả phân tích cú pháp quan hệ

Mỗi một dòng trong bảng kết quả là một cặp tương ứng (từ vựng – quan hệ) ngoại trừ dòng đầu tiên là dòng biểu diễn "nốt quản lý". Dòng đầu tiên không có bất kì nội dung nào (không có từ loại, quan hệ, thuộc tính), nó chỉ dùng để trường cấp dưới tham chiếu đến.

Trường Label là một trường đánh nhãn địa chỉ cho từng dòng và dùng để tham chiếu cho trường Parent's Label. Nếu nó mang giá trị số thì đây cũng chính là vị trí của từ trong câu.

Trường word dùng để chỉ đến từ đang xét, nó chỉ có nội dung của từ khi được xét đến trong lần đầu tiên. Từ lần xuất hiện thứ 2 trở đi (của một từ), trường này bằng rỗng và được tham chiếu đến lần xuất hiện trước đó bằng trường *antecedent*.

Trường root là từ gốc của từ đang xét.

Trường Category mang ý nghĩa từ loại.

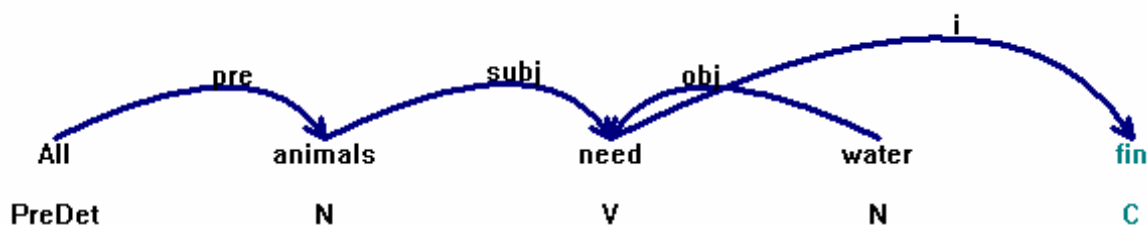
Trường Parent's Label dùng để tham chiếu đến nốt cha cho quan hệ đang được xét (quan hệ trong cùng dòng này).

Trường relation dùng để chỉ tên của mỗi quan hệ giữa word và parent.

Trường parent's root chỉ ra từ gốc của nốt cha.

Trường antecedent là trường tham chiếu cho lần xuất hiện thứ 2 trở đi (như đã được trình bày).

Để dễ hình dung, ta hãy biểu diễn kết quả trên dưới dạng trực quan thành các cung và các nốt, bỏ đi. Bỏ đi phần lớn các thông tin, chỉ giữ lại 2 trường thông tin chính là từ loại và quan hệ.



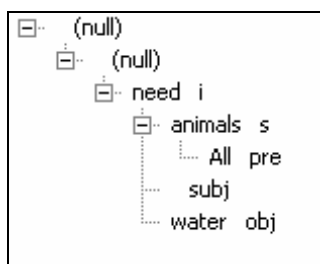
Hình 2 Các mối quan hệ được tìm thấy trong câu "All animals need water"

Giải thích kết quả : Mỗi từ trong câu sẽ hình thành một nốt. Riêng nốt cuối cùng không phải là một từ trong câu (nốt fin). Đây là nốt đại diện cho cấu trúc cả câu. Mỗi nốt sẽ được đánh một nhãn từ loại, nhãn này được đặt dưới các nốt. Các cung nối biểu diễn cho các mối quan hệ giữa các nốt với nhau, chiều của mũi tên hướng từ “con” về “cha”. Nốt nào có cung nối nên nốt fin sẽ là nốt gốc của cây. Nốt fin được đánh nhãn từ loại là C (clause).

Dựa vào cung nối subj, ta biết được animals là chủ thể của hành động need. Tương tự như vậy, water là túc từ của động từ need. Động từ được đưa lên làm trung tâm.

Quan hệ giữa động từ need và nốt fin (quan hệ i) là một quan hệ phân lớp. Nó chỉ ra rằng đây là một mệnh đề kết thúc (xem bảng 4.2).

Để thấy rõ được cách tổ chức phân cấp, ta hãy xem biểu diễn kết quả này theo cách thứ 3: cây quan hệ



Hình 3 Cây quan hệ

Theo cách biểu diễn quan hệ này, ta có thể thấy rõ được việc tổ chức phân cấp của câu. Động từ sẽ được đưa lên làm trung tâm (động từ need). Sau đó là animals và water có mối quan hệ chủ ngữ bề mặt (s) và túc từ (obj) của động từ need. Hãy xét đến quan hệ subj của động từ need, nốt con trong quan hệ này không được nhìn thấy. Tuy nhiên, khi nhìn trở lại kết quả cách biểu diễn dạng bảng, ta có thể biết nốt con này là animals (trường antecedent bằng 2 trong quan hệ subj). Như vậy, animals vừa là chủ ngữ bề mặt vừa là chủ từ thực sự của động từ need.

5.2.2. Đánh giá

5.2.2.1. Ngữ liệu mẫu

Với chương trình phân tích cú pháp quan hệ được trình bày như trên, chúng tôi đã tiến hành đánh giá trên tập ngữ liệu Susanne. Susanne là tập ngữ liệu con của ngữ liệu Brown bao gồm 64 file được chia đều làm 4 thể loại khác nhau (A, G, J, N), mỗi thể loại có 16 file.

A : Các bài phóng sự

G : Tiểu sử, hồi kí.

J : Sách kĩ thuật, bài giảng.

N: Văn chương, tiểu thuyết.

Ngữ liệu Susanne không những đánh nhãn từ loại cho từ cho ngữ mà nó còn đánh nhãn cho vai trò ngữ pháp của từ như chủ ngữ, vị ngữ, túc từ... Ví dụ như, khi một ngữ được đánh nhãn là Ns:s thì có nghĩa là đây một ngữ danh từ và nó đóng vai trò là chủ ngữ trong câu. Chính thông tin này đã giúp chúng ta có thể xác định được tính đúng đắn của các mối quan hệ rút ra từ chương trình phân tích cú pháp.

5.2.2.2. Kết quả đánh giá

Xét về tốc độ chương trình: chương trình phân tích cực kì hiệu quả. Thời gian phân tích trung bình của chương trình là 500 từ/một giây trên máy Pentium III 800 MHz với 128M RAM. So với chương trình phân tích cú pháp dựa vào thống kê (chương trình NLPParser) thì tốc độ nhanh hơn gấp 20 lần.¹²

Bây giờ ta hãy khảo sát đến tính chính xác của chương trình. Khi đánh giá kết quả phân tích, người ta thường đưa ra 2 tiêu chuẩn:

• Độ chính xác (precision): Tỷ lệ phần trăm quan hệ trong kết quả được tìm thấy trong ngữ liệu đúng.

• Độ trùng khớp (recall): tỉ lệ phần trăm quan hệ trong ngữ liệu đúng được tìm thấy trong kết quả.

Từ 2 tiêu chuẩn đánh giá trên, chúng tôi đã cho phân tích trên toàn bộ Susanne và kết quả thu được là : chương trình có độ chính xác là 89% và có độ trùng khớp là 79%. Tuy nhiên, đây chỉ là kết quả mang tính tương đối bởi vì thực tế có sự bất tương đồng giữa kết quả thu được và quan hệ đúng thật sự

¹² Khi cho phân tích trên toàn bộ ngữ liệu Susanne, chương trình chỉ mất 12 phút, trong khi đó, chương trình NLPParser mất đến 4 giờ.

Ü Điểm bất đồng thứ nhất là về quan niệm tách từ. Điều này dẫn đến sự khác nhau về số lượng từ, số lượng quan hệ. Ví dụ, đối với các từ cấu gạch ngang (hyphen), tùy theo quan niệm mà ta có nên tách ra thành 2 từ hay không¹³

Ü Điểm bất đồng thứ 2 là việc một câu có thể hiểu theo nhiều cách khác nhau. Ví dụ, khi xét câu "I saw a man with a telescope", ta có thể hiểu theo 2 cách khác nhau

[I saw a man] with a telescope.

I saw [a man with a telescope].

Do sự khác biệt này mà đôi khi có sự bất đồng nhất giữa kết quả chương trình và ngữ liệu mẫu tuy cả 2 đều đúng.

5.3. Chương trình chiếu kết quả phân tích cú pháp

5.3.1. Chiếu kết quả từ loại

Đây là một số kết quả chiếu từ loại sang tiếng Việt.

Bạn/N đã/J từng/J xem/V một/N cảnh/N kỳ_thú/A trên/C phim/N hay/C đã/J từng/J xem/V một/N bức/N tranh/N mà/C trông/V như/J thật/A đến_nỗi/A bạn/N nghĩ/V là/V một/N bức/N ảnh/N chưa/J ?/Q

Và/C bạn/N có/J ngõ_ngang/A khi/N biết/V được/V rằng/C những/N điều/N đó/P được/V làm/V trên/C máy_tính/N không/J ?/Q

Nếu/C có/A ,/Q bạn/N thì/C chắc_chắn/V chẳng/J phải/V mình/N đâu/J ./Q

Chúng_ta/N sẽ/J không/J hết/A ngạc_nhiên/V vì/C những/N kết_quả/N hoàn_hảo/A nhờ/V sựN giúp_đỡ/N của/C máy_tính/N và/C chúng_ta/N sẽ/J thú_vị/A bởi/C sự/N phức_tạp/N của/C nó/P ./Q

¹³ Một số ví dụ về từ có chứa dấu gạch ngang : overage(hàng hoá dư),over-age (quá tuổi), screw-bolt(tua vít).

Vì/C lý_do/N này/P ./Q nhiều/A người/N cho/V rằng/V máy_tính/N thật/J khó/A hiểu/V và/C khó/A sử_dụng/V ./Q

Tuy_nhiên/C ./Q hầu_hết/N chúng_ta/N không/J hiểu/V rằng/V ./Q cơ_bản/N máy_tính/N là/V một/N thiết_bị/N đơn_giản/A và/C tất_cả/N các/N máy_tính/N đều/J có/V nhiều/A sự/N đồng_nhất/N ./Q

Hầu_hết/A các/N máy_tính/N từ/C lớn_nhất/A cho_đến/C nhỏ_nhất/A đều/J thao_tác/V dựa/V vào/V các/N qui_tắc/N căn_bản/A như_nhau/A ./Q

Tất_cả/N chúng/N đều/A được/V xây_dựng/V trên/C các/N kiểu/N bộ/N phận_cấu_thành/N cơ_bản/A như_nhau/A và/C đều/A cần/V phải/V có/V các/N chỉ_dẫn/N để/C điều_khiển/V chúng/N hoạt_động/V ./Q

Là/C bước/N đầu_tiên/A để/C hiểu/V và/C học/V cách/N sử_dụng/N máy_tính/N ./Q bài_học/N này/P cung_cấp/V cho/V bạn/N một/N cái/N nhìn/N cơ_bản/A về/C loại/N máy/N hấp_dẫn/A này/P ./Q

Chúng_ta/N sẽ/J học/V về/C các/N kiểu/N phần_cứng/N mà/C tất_cả/N các/N hệ_thống/N máy_tính/N đều/J sử_dụng/V ./Q và/C các/N kiểu/N phần_mềm/N vận_hành/V chúng/N ./Q

Chúng_ta/N cũng/J sẽ/J thấy/V rằng/C nếu/C không/J có/A người_sử_dụng/N -/Q người/J nào/P đó/P như/C bạn/N -/Q thì/C một/N hệ_thống/N máy_tính/N sẽ/J thực_sự/A không/J đầy_đủ/A ./Q

Liệt_kê/V bốn/N phần/N của/C một/N hệ_thống/N máy_tính/N ./Q

Xác_định/V bốn/N kiểu/N phần_cứng/N máy_tính/N ./Q

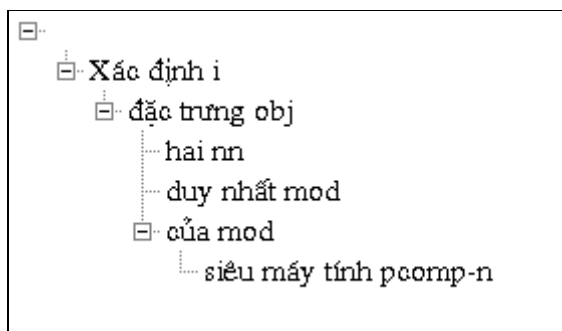
Liệt_kê/V năm/N đơn_vị_do/N dung_lượng/N bộ_nhớ/N và/C bộ/N lưu_trữ/N của/C máy_tính/N ./Q

Cho/V hai/N ví_dụ/N về/V các/N thiết_bị/N nhập/A và/C xuất/A ./Q

5.3.2. Chiều kết quả phân tích quan hệ

Kết quả của tất cả các bước trước là cơ sở để tiến hành chiều kết quả quan hệ.

Xét một ví dụ : “Xác định hai đặc trưng duy nhất của siêu máy tính”. Các quan hệ được chiếu hình thành nên cây cú pháp quan hệ trên tiếng Việt như sau:



Hình 4 Kết quả ánh xạ cây quan hệ của câu “Xác định hai đặc trưng duy nhất của siêu máy tính”.

5.4. Kết luận

Mặc dù lý thuyết về dịch máy thống kê đã có từ rất lâu nhưng trong những năm gần đây thì nó mới được ứng dụng rộng rãi cho các hướng nghiên cứu khác nhau. Trong đó nó được ứng dụng mạnh mẽ trong dịch máy. Trong bài luận văn này chúng tôi chỉ mô tả một phần ứng dụng của dịch máy thống kê cho việc liên kết từ. Dịch máy thống kê có một điểm mạnh ở chỗ là không cần biết về thế giới thực nhưng ngược lại nó có nhược điểm ở chỗ là bị phụ thuộc vào dữ liệu song ngữ rất nhiều. Nếu dữ liệu song ngữ được dịch càng chính xác thì kết quả của nó càng cao.

Kết quả của việc liên kết từ mà chúng tôi thu được từ cách tiếp cận dịch máy thống kê hết sức trọng đối với hệ dịch máy và góp phần không nhỏ cho các hướng nghiên cứu khác như: khảo sát sự thay đổi trật tự từ của cây cú pháp tiếng Việt và cây tiếng Anh, giải quyết vấn đề nhập nhằng ngữ nghĩa, ...

5.5. Hướng phát triển

Vì trong thời gian hạn chế nên chương trình chúng tôi còn một số hạn chế, và hướng phát triển của chương trình của chúng tôi bao gồm những công việc như sau

Tính xác suất dịch phụ thuộc vào ngữ cảnh (Brown et al., 1991; Berger et al., 1996; Melamed, 1998a).

Thêm vào bộ tìm hình thái học (Morphology) cho tiếng Anh. Bộ tìm hình thái học sẽ giảm các kính cỡ của từ vựng tiếng Anh xuống và nâng cao hiệu suất thống kê.

Khởi gán các liên kết tốt hơn cho các mô hình 3, 4 và 5 để công việc đếm của chúng chỉ duyệt qua một tập con các liên kết nhỏ hơn.

Xây dựng một hệ thống dịch tự động từ tiếng Việt sang tiếng Anh dựa vào mô hình dịch máy thống kê hoàn chỉnh.

Xây dựng mạng ngữ pháp cho tiếng Việt để cài đặt một chương trình phân tích cú pháp độc lập.

PHỤ LỤC A: Bảng qui ước các ký hiệu của mô hình dịch máy thống kê

Ký hiệu	Ý nghĩa
e	Từ vựng tiếng Anh
\mathbf{e}	Câu tiếng Anh
\mathbf{E}	Câu tiếng Anh ngẫu nhiên
l	Chiều dài câu tiếng Anh \mathbf{e}
L	Chiều dài câu tiếng Anh \mathbf{E} ngẫu nhiên
i	Vị trí trong \mathbf{e} , $i = 0, 1, \dots, l$
e_i	Từ thứ i trong \mathbf{e}
e_0	Từ rỗng
e_1^i	$e_1 e_2 \dots e_i$
v	Từ vựng tiếng Việt
\mathbf{v}	Câu tiếng Việt
\mathbf{V}	Câu tiếng Việt ngẫu nhiên
m	Chiều dài câu tiếng Việt \mathbf{v}
M	Chiều dài câu tiếng Việt \mathbf{V} ngẫu nhiên
j	Vị trí trong \mathbf{v} , $j = 1, 2, \dots, m$
v_j	Từ thứ j trong \mathbf{v}
v_1^j	$v_1 v_2 \dots v_j$

PHỤ LỤC A: Bảng qui ước các ký hiệu của mô hình dịch máy thống kê

a	Liên kết từ
a_j	Vị trí trong e được kết nối tới vị trí thứ j trong v của liên kết a
a_1^j	$a_1 a_2 \dots a_j$
i	Số vị trí trong v được kết nối với vị trí j trong e
i_1	$1 \ 2 \dots i$
	Tập hợp tableau – mỗi dãy các bảng tablet, và bảng tablet là một dãy các từ tiếng Việt
i	Bảng tablet thứ i trong
i_0	$0 \ 1 \dots i$
i	Chiều dài của i
k	Vị trí bên trong bảng tablet, $k = 1, 2, \dots, i$
ik	Từ thứ k của i
	Việc hoán vị của những vị trí trong tập hợp tableau
ik	Vị trí trong v cho từ thứ k của i cho việc hoán vị
i_1^k	$i_1 \ i_2 \dots i_k$
$V(\mathbf{v} \mathbf{e})$	Liên kết tối ưu nhất Viterbi của cặp câu (\mathbf{e}, \mathbf{v})
$N(\mathbf{a})$	Tập hợp liên kết láng giềng của a
$A(e)$	Lớp của từ tiếng Anh e
$A(v)$	Lớp của từ tiếng Việt v
d_j	Sự dịch chuyển của một từ trong v
	Những vị trí trống trong v
c_i	Vị trí trung bình trong v của những từ được kết nối tới vị trí thứ i của e

PHỤ LỤC A: Bảng qui ước các ký hiệu của mô hình dịch máy thống kê

$t(v e)$	Xác suất dịch từ (cho tất cả các mô hình)
$(m l)$	Xác suất chiều dài của một cặp câu (mô hình 1 và 2)
$n(\cdot e)$	Xác suất sản sinh (mô hình 3, 4 và 5)
p_0, p_1	Xác suất sản sinh của của từ tiếng Anh rỗng e_0 (mô hình 3, 4 và 5)
$a(i j, l, m)$	Xác suất liên kết vị trí từ j sang i (mô hình 2)
$d(j i, l, m)$	Xác suất liên kết vị trí từ i sang j (mô hình 3)
$d_1(j A, B)$	Xác suất dịch chuyển của từ đầu tiên trong bảng tablet (mô hình 4)
$d_{-1}(j B)$	Xác suất dịch chuyển của các từ khác từ đầu tiên trong bảng tablet (mô hình 4)
$d_1(j B, \cdot)$	Xác suất dịch chuyển của từ đầu tiên trong bảng tablet (mô hình 5)
$d_{-1}(j B, \cdot)$	Xác suất dịch chuyển của các từ khác từ đầu tiên trong bảng tablet (mô hình 5)

PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ

Stt	Tên đặc tính	Mô tả
Thuộc tính nhị phân		
1	3sg	Ngôi thứ 3 số ít (dùng cho cả động từ, danh từ, đại từ)
2	Allcap	Tất cả kí tự viết
3	Adv	trạng từ
4	Appo	Appositive
5	Att	attributive or predicative
6	Be	To be
7	Bare	bare clause (trái với complement clause)
8	Cap	phải được viết hoa (Vd : từ I)
9	Cm	-cm means the source needs case marking. This attribute is used to implement Case Filter, which states that all overt noun phrases must be case marked.
10	Cmp	Tính từ có thể so sánh (vd: “hot” thì có, nhưng “national” thì không phải.)
11	Cn	compound noun (Vd : “army hut”).
12	control	(Không dùng)
13	Ct	Countable noun
14	Det	Deteminer.
15	Easy	giống từ easy (difficult, tough) : bởi vì nó khá đặc biệt

PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ

16	free_rel	Free relative clause
17	genitive	Genitive pronoun (Vd : their, Kim's)
18	govern	(không dùng)
19	Group	Danh từ số ít lẫn số nhiều (committee, crew, fish)
20	Guest	to indicate a phrase is an adjunct
21	have	A form of have
22	head_final	vị trí của head ở cuối cùng (trong tiếng anh thì giá trị mặc định là sai). Ngược lại trong SOV thì đúng.
23	inv	chỉ sự đảo vị trí của aux verb và subject (trong câu hỏi)
24	last_conj	And/or are +last_conj, either/both are -last_conj
25	neg	Negation (E.g couldn't, isn't)
26	nilto	verbs that are followed by an covert 'to' have +nilTo (E.g., help, wanna, gotta)
27	opt	optional argument (For example, the object of 'cook' is optional.)
28	perf	indicates a verb or a clause has the perfective aspect
29	plu	Plural
30	pn	Proper noun
31	postnom	Post nominal adjective
32	prd	Predicative
33	pro	PRO subject .
34	prog	A clause has +prog if it has progressive aspect
35	pron	Pronoun
36	ref	reference entry used to deal irregular verbs
37	refl	reflective pronoun. Examples: myself, themselves

PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ

38	wh	Wh-element
Thuộc tính liệt kê		
39	auxform	the forms auxiliary verb (allowable-values to can could dare do did does may might would should must will ought shall have_to be_going_to need had_better)
40	cat	major category
41	Pform	Preposition form
Thuộc tính 8 giá trị (Disj8)		
42	Vform	Inflection of verb (allowable-values bare s ed ing)
43	Role	allowable-values : subj subject scsubj subject of small clause obj object obj2 second object sc small clause fc full clause dest destination desc description pcomp-n complement of preposition mod modifier expletive
44	Rare	The rarity of lexical items (allowable-values very very_very))
45	pred	type of predicate (allowable-values n v a p c))
46	barred	barred as these types modifiers (allowable-values ba

PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ

		<p>aa bv av)</p> <p>This attribute is mostly used by adverbs</p> <p>ba : it cannot be used before an adjective</p> <p>aa : it cannot be used after an adjective</p> <p>bv: it cannot be used before a verb</p> <p>av : it cannot be used after a verb</p>
47	slash	types of movement (allowable-values np wh)) (không dùng)
48	check	(Không dùng)
49	nform	(there are 3 allowable-values : there it norm))
50	case	<p>The case of NP (allowable-values acc nom dat gen))</p> <p>acc: accusitive case, assigned to nouns that are the objects of verbs and prepositions</p> <p>nom : nominative case, assigned to nouns at subject positions</p> <p>dat : dative case</p> <p>gen : genitive case</p>
51	cform	<p>The form of clauses</p> <p>(allowable-values fin inf npsc apsc ppsc vpsc)):</p> <p>fin: finite clause, e.g., I think [the key is lost]</p> <p>inf: infinitive clause,</p> <p>e.g., I wanted [to sleep]. I believe [him to be a good candidate]</p> <p>npsc: small clause where the predicate is a noun phrase</p>

PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ

		<p>e.g., I consider [him a good candidate]</p> <p>ppsc: small clause where the predicate is a prepositional phrase</p> <p>e.g., I consider [it in good condition]</p> <p>apsc : small clause where the predicate is a adjectival phrase</p> <p>e.g., I consider [it good]</p> <p>vpsc: small clause where the predicate is a -ing verb phrase</p> <p>e.g., I saw [them leaving the garden]</p>
52	Comp	<p>The type of complimentizers</p> <p>(allowable-values : for that whether if other none))</p> <p>The above words are the only ones that have the comp attribute.</p>
53	Person	<p>(allowable-values 1 2 3))</p> <p>(per 1): I, me, my, we, us, our ...</p> <p>(per 2): you, your, yours,</p> <p>(per 3): he, she, they, them, ...</p>
53	Tense	<p>Allowable-values : present past future pastfut</p> <p>The tense attribute specifies the tense of a clause.</p> <p>The tense of infinitive and small clauses are undefined.</p>
Thuộc tính dịch chuyển		
54	whform	Form of wh-element

PHỤ LỤC B: Các thuộc tính trong phân tích cú pháp quan hệ

Thuộc tính vết (Trace)		
57	trace	Indicate the position of the trace
Thuộc tính chuỗi (String)		
58	nốt	Corresponding nốt in the grammar network
Thuộc tính Véc-tơ		
59	sem	The semantic properties of the word
Thuộc tính ngăn xếp (Stack)		
60	args	The list of arguments of a word (including the subject).
61	move	The list of descriptions of the moved elements. Số đối số của một động từ có thể nhận ra ở trường này (intransitive verb, transitive verb V_N, V_N_N, V_N_N_N ...).

PHỤ LỤC C: Bộ nhãn từ loại tiếng Anh

Tên từ loại	Ý nghĩa	Ví dụ	Ghi chú
Det	Determiners		
PreDet	Pre-determiners	All, as much as, even, just, only...	
PostDet	Post-determiners	I'll see you next Friday.	
NUM	numbers		
C	Clauses		
I	Inflectional Phrases		
V	Verb and Verb Phrases		
N	Noun and Noun Phrases		Có thể phân biệt noun và pronoun bằng thuộc tính pro
NN	noun-noun modifiers	operating system software	
P	Preposition and Preposition Phrases		
PpSpec	Specifiers of Preposition Phrases	back, up, dead	
A	Adjective/Adverbs		Có thể phân biệt Adj/Adv dựa vào

PHỤ LỤC C: Bộ nhãn từ loại tiếng Anh

			thuộc tính "adv"
Have	have	I have gone there.	
Aux	Auxiliary verbs, e.g. should, will, does, ...		
Be	Different forms of be	is, am, were, be, ...	
COMP	Complementizer	The book that/COMP you lent me is very old.	
VBE	be used as a linking verb.	I am hungry	
V_N	verbs with one argument (the subject), i.e., intransitive verbs	I eat rice.	Những từ loại này không có trong cột từ loại mà nằm trong thuộc tính "move" (cột cuối cùng)
V_N_N		I send him one dollar.	
V_N_I	verbs with two arguments, i.e., transitive verbs		

PHỤ LỤC D: Các mối quan hệ trong tiếng Anh

Stt	Tên quan hệ	Ý nghĩa	Ví dụ
1	appo	Quan hệ giải thích	ACME president, --appo-> P.W. Buckman
2	aux	Quan hệ giữa trợ động từ và động từ	should <-aux-- resign
3	be	Quan hệ giữa tobe và động từ	"is <-be-- sleeping"
4	c	Quan hệ giữa đại từ quan hệ và mệnh đề quan hệ	"that <-c-- John loves Mary"
5	comp1	first complement	I do it once I leave.
6	det		the hat
7	gen		Mary's father.
8	have		have gone
9	i	the relationship between a C clause and its I clause	
10	inv-aux		inverted auxiliary: "Will <-inv-aux-- you stop it?"
11	inv-have		inverted have: "Have <-inv-have-- you slept"
12	mod	the relationship	The book in the desk is mine.

PHỤ LỤC D: Các mối quan hệ trong tiếng Anh

		between a word and its adjunct modifier	
13	pnmod		Sales of passenger cars grew 22 % from a year earlier to 361,376 units
14	p-spec		The purchasing managers also said that orders turned up in October after four months of decline .
15	pcomp-c	clausal complement of prepositions	We have no useful information on whether users are at risk.
16	pcomp-n	nominal complement of prepositions	I meet him at school
17	post	post determiner	I will see you next Friday.
18	pre	pre determiner.	All book.
19	pred	predicate of clause	There is a book on the table.
20	rel	relative clause	The man who you met yesterday is my brother.
21	vrel	passive verb modifier of nouns	The book bought by my father is a good book.

TÀI LIỆU THAM KHẢO

- [1] Brown, Peter, Stephen Della-Pietra, Vincent Della-Pietra, and Robert Mercer. *The mathematics of statistical machine translation: Parameter estimation*. Computational Linguistics, 19(2): 263-311 (1993).
- [2] Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, David Yarowsky. *Final Report*. JHU Workshop, (1999).
- [3] Kevin Knight. *A Statistical MT Tutorial Workbook*. Prepared connection with the JHU workshop, (April 30, 1999).
- [4] Dinh Dien, Hoang Kiem, Van Toan, Quoc Hung, Phu Hoi, Thuy Ngan, Xuan Quang, *Word alignment in English – Vietnamese bilingual corpus*, Proceedings of International Conference on East-Asia Language Processing and Internet Information Technology 2002. Hanoi, VietNam, (2002).
- [5] Đinh Điền, Nguyễn Thống Nhất, Nguyễn Thái Ngọc Duy. *Cách tiếp cận dịch máy thống kê cho hệ dịch tự động Việt-Anh*. Tạp chí Phát triển Khoa học Công nghệ, tập 6 (1&2-2003)
- [6] Đinh Điền, Nguyễn Thống Nhất, Nguyễn Thái Ngọc Duy. *Cách tiếp cận dịch máy thống kê cho việc liên kết từ trong song ngữ Anh-Việt*. Hội nghị Trường Khoa học Tự nhiên – ĐHQG TP. HCM, Lần thứ 3 (9-2002)

- [7] Đinh Điền, Nguyễn Văn Toàn, Ngô Quốc Hưng, Nguyễn Lưu Thủy Ngân, Đỗ Xuân Quang, Phạm Phú Hội. *Cách tiếp cận dựa trên sự phân lớp cho việc liên kết từ Anh-Việt*. Hội nghị kỷ niệm 25 năm thành lập, Trung tâm khoa học tự nhiên và công nghệ quốc gia, Viện công nghệ thông tin, (2001).
- [8] Ngô Quốc Hưng và Phạm Phú Hội. Luận văn cử nhân tin học: *Liên kết từ trong song ngữ Anh Việt (ứng dụng trong khảo sát trật tự từ)*, Trường Đại học Khoa học tự nhiên. (2002).
- [9] Đặng Hùng Thắng. *Mở đầu về lý thuyết xác suất và các ứng dụng*. Nhà xuất bản giáo dục – Việt Nam (1999).
- [10] Allen, Arnold O. *Probability, Statistics and Queueing Theory*, Academic Press, New York, NY (1978).
- [11] Lafferty, John, Daniel Sleator, Davy Temperly. Grammatical trigrams: a probabilistic model of link grammar. Proceedings of the 1992 AAAI Fall Symposium on Probabilistic Approaches to Natural Language (1992).
- [12] Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. *Fast Decoding and Optimal Decoding for Machine Translation*. In 39th Annual Meeting of the Association for Computational Linguistics. ACL (2001).
- [13] Y. Wang and A. Waibel. *Decoding algorithm in statistical machine translation*. In Proc. ACL (1997).
- [14] K. Knight. *Decoding complexity in word-replacement translation models*. Computational Linguistics, 25(4) (1999).
- [15] Ker S. J. and Chang J. S. *A class-based Approach to Word Alignment*, Computational Linguistics, 23/2, page 313-343 (1997).

- [16] Dekang Lin. *Principle-Based Parsing without Overgeneration*. In Proceeding of ACL-93, pages 112-120, Columbus, Ohio, 1993.
- [17] Dekang Lin. *Principar-an efficient, broad-coverage, principle-based parser*. In proceedings of COLING-94, pages 482-488. Kyoto, Japan, 1994.
- [18] David Yarowsky and Grace Ngai, 2001. *Inducing multilingual pos taggers and np bracketter via robust projection across aligned corpora*. In Proc. Of NAACL-2001, pages 200-207.