BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

ỨNG DỤNG MÔ HÌNH DỊCH MÁY THỐNG KÊ CHO BÀI TOÁN BỎ DẤU CHO VĂN BẢN TIẾNG VIỆT

Sinh viên thực hiện: Đinh Văn Toản

Giáo viên hướng dẫn: Ths . Nguyễn Thị Xuân Hương

Mã số sinh viên: 110584

CHƯƠNG 1: NGÀNH: CÔNG NGHỆ THÔNG TIN

LỜI CẢM ƠN

Đầu tiên em xin chân thành cảm ơn đến các thầy cô giáo khoa Công nghệ thông tin Trường Đại học dân lập Hải Phòng đã tận tình dạy bảo cho em suốt thời gian học tập tại trường.

Em xin gửi lời biết ơn sâu sắc đến Ths.Nguyễn Thị Xuân Hương đã dành rất nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành luận văn tốt nghiệp.

Mặc dù em đã có nhiều cố gắng hoàn thiện luận văn bằng tất cả sự nhiệt tình và năng lực của mình, tuy nhiên không thể tránh khỏi những thiếu sót, nên em rất mong nhận được những đóng góp quý báu của các thầy cô và các bạn.

Hải Phòng, tháng 07 năm 2011.

Sinh viên

Đinh Văn Toản

MỤC LỤC

DANH MỤC HINH VẼ

LỜI NÓI ĐẦU

Chữ viết tiếng Việt của chúng ta có một đặc trưng rất riêng biệt đó là có sự xuất hiện của các dấu thanh và dấu của các ký tự. Điều này giúp cho tiếng Việt "thêm thanh, thêm điệu". Tuy nhiên, chính việc "thêm thanh, thêm điệu" này làm cho việc gõ tiếng Việt trở nên tốn nhiều thời gian hơn.

Trong cuộc sống hiện đại ngày nay, việc sử dụng các ứng dụng công nghệ thông tin để trao đổi và truyền thông tin càng trở lên phổ biến. Hàng ngày, chúng ta đọc và nhận được rất nhiều e-mail, blog, những tin nhắn messenger… nhưng một số trong đó lại được truyền bởi chữ tiếng Việt không dấu. Chúng ta thật là vất vả khi phải vừa đọc vừa đoán nội dung. Chính vì vậy phát triển một công cụ giúp thêm dấu tiếng Việt vào văn bản không dấu là việc rất cần thiết và thú vị.

Hiện nay đã có nhiều nhóm nghiên cứu đã phát triển các phần mềm cho bài toán thêm dấu cho văn bản Tiếng Việt. Có thể kể đến như: VietPad là một chương trình text editor Việt Unicode được phát triển bởi Quân Nguyễn và nhóm phát triển trên http://vietpad.sourceforge.net. viAccess, phần mềm bỏ dấu tiếng việt online tại địa chỉ: http://vietlabs.com/vietizer.html. AMPad của Trần Triết Tâm được nâng cấp của chương trình "AutoMark" có thể chuyển đổi chính xác đến khoảng 80% hoặc hơn.VietMarker, được phát triển bởi nhóm nghiên cứu là giảng viên và sinh viên Học viện Công nghệ Bưu chính Viễn thông, đạt được độ chính xác cao, đến 93%, … Các tiếp cận cho các hệ thống trên chủ yếu sử dụng phương pháp tách từ và so khớp với Từ điển.

Trong đề tài này, chúng tôi hướng đến việc giải quyết bài toán thêm dấu cho văn bản tiếng việt theo mô hình dịch máy thống kê. Dịch máy bằng phương pháp thống kê (Statistical Machine Translation) là một hướng tiếp cận cho dịch máy đã và đang thu hút được rất nhiều sự quan tâm của cộng đồng nghiên cứu trong lĩnh vực xử lý ngôn ngữ tự nhiên hiện nay. Thay vì xây dựng các từ điển, các luật chuyển đổi bằng tay, hệ dịch này tự động xây dựng các từ điển, các quy luật dựa trên kết quả thống kê có được từ dữ liệu. Chính vì vậy, dịch máy dựa vào thống kê có tính khả chuyển cao, có khả năng áp dụng được cho cặp ngôn ngữ bất kỳ.

Luận văn được tổ chức thành 3 chương với nội dung như sau:

Chương 1: Tổng quan về bài toán thêm dấu cho văn bản Tiếng Việt: trong đó trình bày về bái toán và giới thiệu các hệ thống đã có cho bài toán này.

Chương 2: Giới thiệu mô hình dịch máy thống kê cho bài toán thêm dấu cho văn bản tiếng Việt,

Chương 3: Thực nghiệm: là các giới thiệu về việc sử dụng các hệ mã nguồn mở SRILM, GIZA++, MOSES phục vụ cho hệ dịch máy thống kê và các kết quả khi sử dụng hệ này để thêm dấu cho văn bản tiếng Việt,

Và cuối cùng là phần Kết luận.

CHƯƠNG 2: TỔNG QUAN VỀ BÀI TOÁN THÊM DẤU CHO VĂN BẢN TIẾNG VIỆT

2.1.1 Phát biểu bài toán

- Bài toán có thể được phát biểu như sau:
- **Input:** Cho một văn bản tiếng Việt không dấu.
- **OutPut:** Chuyển văn bản không dấu này thành có dấu.
- Sử dụng phương pháp dịch máy thống kê để biên dịch.

2.1.2 Đặc điểm

Với sự xuất hiện của các dấu thanh cũng như dấu của các ký tự, đã làm phong phú thêm cho ngôn từ tiếng Việt, và cũng góp phần tăng độ biểu cảm của tiếng Việt.

Dấu thanh là phần "bất khả phân" trong âm tiết tiếng Việt. Khi loại bỏ dấu thanh, việc hiểu nghĩa từ, gồm một hay nhiều âm tiết kết hợp với nhau, trở nên khó khăn và dễ gây hiểu lầm.

Để thêm dấu, trước tiên, ta cần phải xác định ranh giới từ. Bài toán xác định ranh giới từ đối với văn bản tiếng Việt có dấu đã là một việc thử thách, thì khi không có dấu, việc nhận diện ranh giới từ trong tiếng Việt cũng như một số ngôn ngữ Châu Á khác, một từ chính tả có thể không tương ứng với một "từ" trên văn bản. Đối với các thứ tiếng Châu Âu, ta có thể dễ dàng nhận ra một từ, do các từ được phân cách bởi khoảng trắng. Điều này lại không đúng với tiếng Việt. Trong tiếng Việt, các tiếng_hay còn gọi là âm tiết_được phân cách bởi khoảng trắng, chứ không phải từ.

Sau khi đã nhận diện được ranh giới từ, ta cần phải xác định cho đúng từ có dấu nào có dạng thể hiện không dấu như vậy. Việc xác định này cũng gây nhiều khó khăn, khi từ một từ không dấu có thể có nhiều từ có dấu tương ứng với nó.

Ví dụ 1-1 : Từ không dấu "me" có 3 từ có dấu tương ứng là "mẹ", "mẻ" và "mè".

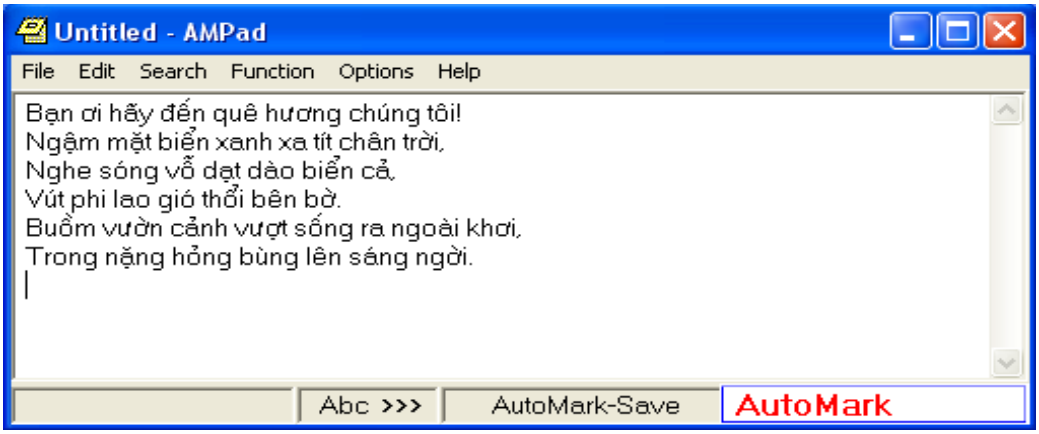
Do đó, sau khi đã giải quyết xong bài toán tách từ tiếng Việt không dấu, ta cần phải giải quyết thêm bài toán xác định từ có dấu thích hợp với từ không dấu đó.

2.2 Giới thiệu công trình đã có

2.2.1 AMPad

AMPad là chương trình chuyển đổi loại tiếng Việt không dấu sang tiếng Việt có dấu thuộc dạng khá chuyên nghiệp. Thực chất nó là bản nâng cấp của chương trình "AutoMark" đã được tác giả Trần Triết Tâm ở Cục thống kê Đà Nẵng tung ra trước đây.AMPad có thể có chuyển đổi chính xác đến khoảng 80% hoặc hơn các đoạn văn dạng chính luận xã hội, hoặc khoa học thường thức… trên các sách báo hiện nay và nó chỉ "chào thua", tức đoán sai đến hơn 50% ở các câu văn thuộc dạng chuyên ngành sâu, hoặc ở các lĩnh vực văn học, thơ ca… với cấu trúc câu vốn quá phức tạp và lắm ngữ nghĩa.

Em đã sử dụng nhiều câu trên nhiều tờ báo để "thử sức" AMPad và công nhận rằng nó là một công cụ "siêu hữu dụng" cho những người đánh máy tiếng Việt dạng "mổ cò". Sau đây là một số ví dụ:



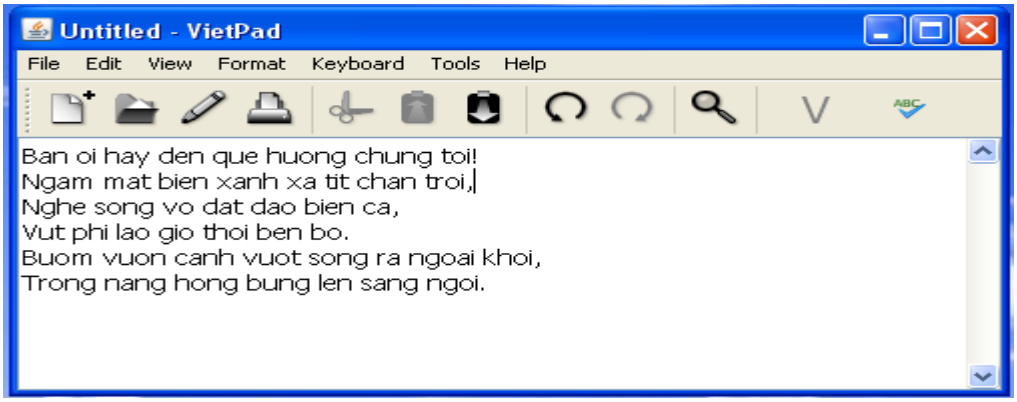
Hinh 1.2.1-1 Thêm dấu tiếng Việt tự động bằng AMPad

Mặc dù vẫn có sai sót nhưng AMPad thực sự là một công cụ tuyệt chiêu gần như "độc nhất vô nhị", không những thật sự có hiệu quả với chính người Việt mà còn là công cụ vô cùng hữu dụng cho những người nước ngoài đang học tiếng Việt.

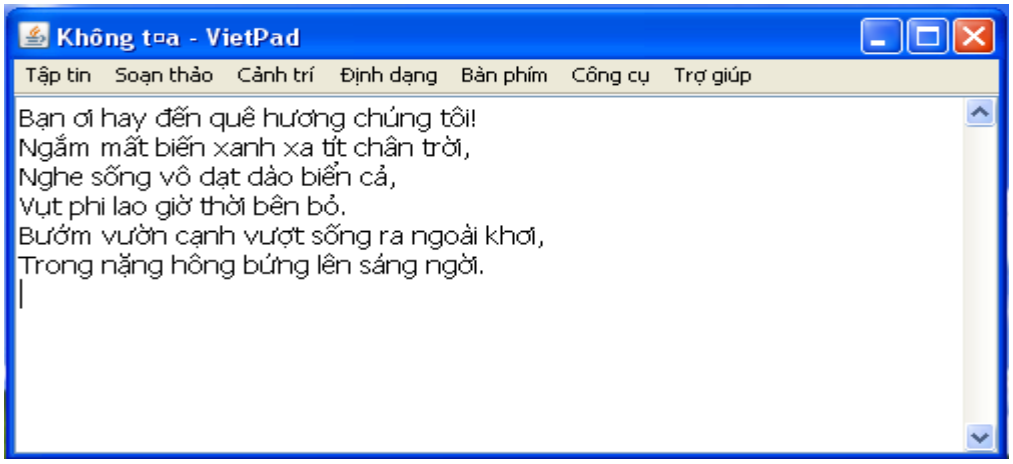
2.2.2 VietPad

VietPad là một chương trình text editor Việt Unicode đầy đủ tính năng có thể chạy trên các máy computer có gắn Java Runtime Environment, như các máy có hệ điều hành Windows, Linux/Unix, Mac OS X, hay Solaris. VietPad được phát triển bởi Quân Nguyễn và nhóm phát triển trên http://vietpad.sourceforge.net. Người sử dụng có thể đánh chữ Việt tương thích với tiêu chuẩn Unicode dùng những cách đánh phổ

thông như lối Telex, VNI, hay VIQR/Vietnet. VietPad hỗ trợ file và text Drag-and-Drop và khả năng bỏ dấu thông minh.



Hình 1.2.2-2 Gõ tiếng Việt không dấu trên VietPad



Hình 1.2.3-3 Văn bản sau khi thực hiện chức năng thêm dấu tiếng Việt của VietPad

2.2.3 viAccent

Phần mềm bỏ dấu tiếng việt online tại địa chỉ:

http://vietlabs.com/vietizer.html

Hình 1.2.3-4 : Gõ tiếng việt không dấu trên viAccent

Kết quả thu được sau khi ấn vào nút thêm dấu:

bạn ơi hãy đến quê hương chúng tôi
ngắm mặt biển xanh xã tít chân trời, nghe sống vợ đất đạo biển cả, vứt phi lao gió thổi bên bờ.
buồm vườn cảnh vượt sống ra ngoài khôi, trong nắng hồng bùng lên sáng ngồi.

2.2.4 VietMarker

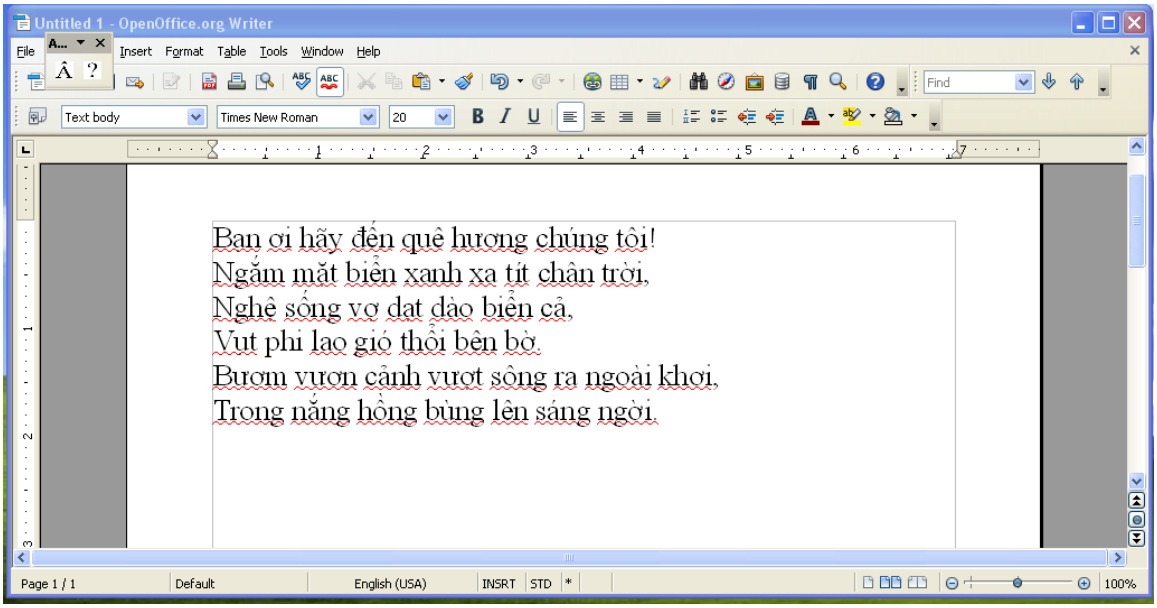
VietMarker, một phần mềm Việt vừa ra mắt sẽ giúp bạn thêm dấu tự động vào các văn bản tiếng Việt không dấu.

VietMarker được phát triển bởi nhóm nghiên cứu là giảng viên và sinh viên Học viện Công nghệ Bưu chính Viễn thông (vietmarker@gmail.com). Bằng việc áp dụng công nghệ mới, việc thêm dấu tự động đạt được độ chính xác cao, đến 93% với đa dạng thể loại văn bản trong các chủ đề, lĩnh vực khác nhau.

Phần mềm được viết bằng ngôn ngữ Java, và được phát triển thành một Add-on dùng cho bộ ứng dụng văn phòng mã nguồn mở Open Office. Chúng tôi lựa chọn giao

diện lập trình ứng dụng dành cho Open Office với ngôn ngữ Java được cung cấp tại http://api.openoffice.org/ để tạo Add-on.

Add-on **Dấu Việt** được cài đặt và sử dụng một cách dễ dàng, thuận tiện với những thao tác đơn giản giúp cho người dùng giảm đáng kể thời gian soạn thảo văn bản, hoặc dịch một cách phù hợp nhất những đoạn văn bản tiếng Việt không dấu sang văn bản có dấu tương ứng.



Hình 1.2.4-5 Văn bản sau khi thực hiện chức năng thêm dấu của Vietmarker

Ngoài ra còn có một số phần mềm thêm dấu tiếng Việt khác như là www.easyvn.com, VnMark…

2.2.5 Hướng tiếp cận trong đề tài

Đề xuất là sử dụng phương pháp dịch máy thống kê để giải quyết bài toán. Sử dụng các luật Bayes để mô hình lại khả năng dịch cho việc dịch một câu không dấu f sang câu tiếng việt e như sau:

$$\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} (\mathbf{f}|\mathbf{e})\ (\mathbf{e})$$

Nó cho thể hiện mô hình ngô ngữ e và mô hình dịch với p(f|e)

CHƯƠNG 3: GIỚI THIỆU MÔ HÌNH DỊCH MÁY THỐNG KÊ

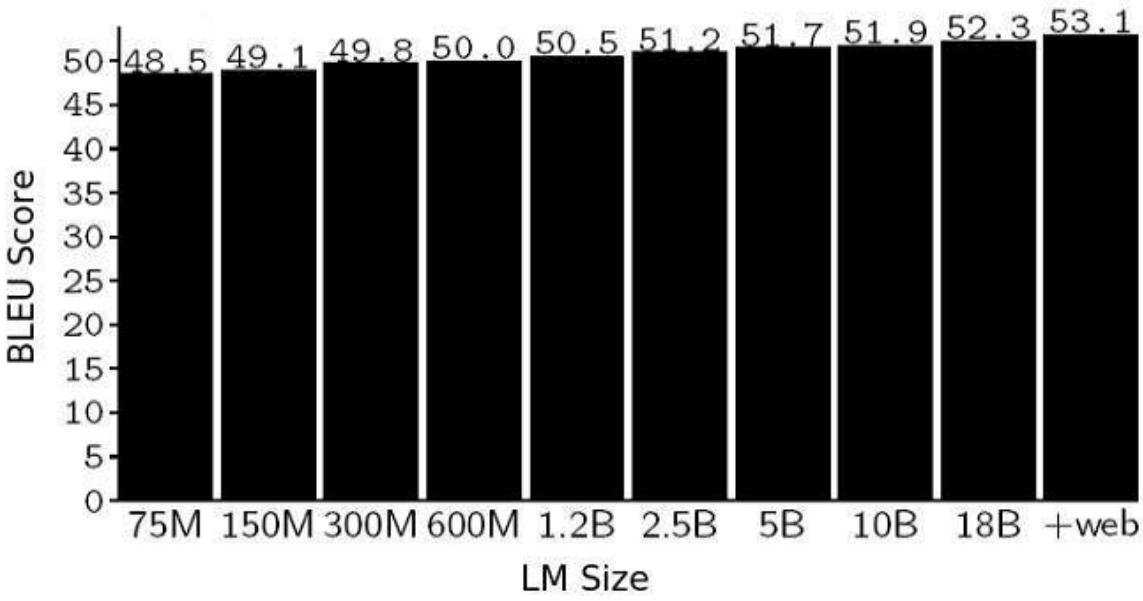
3.1 Giới thiệu

Dịch máy bằng phương pháp thống kê (Statistical Machine Translation) đã chứng tỏ là một hướng tiếp cận đầy đầy tiềm năng bởi những ưu điểm vượt trội so với các phương pháp dịch máy dựa trên cú pháp truyền thống qua nhiều thử nghiệm về dịch máy. Thay vì xây dựng các từ điển, các luật chuyển đổi bằng tay, hệ dịch này tự động xây dựng các từ điển, các quy luật dựa trên kết quả thống kê có được từ dữ liệu. Chính vì vậy, dịch máy dựa vào thống kê có tính khả chuyển cao, có khả năng áp dụng được cho cặp ngôn ngữ bất kỳ. Hệ thống SMT được đề xuất lần đầu tiên bởi Brown năm 1990 sử dụng *mô hình kênh nhiễu* (noisy channel model) và đã phát triển áp đảo trong ngành MT nhiều năm trở lại đây.

Trong phương pháp *dịch trực tiếp*, từng từ được dịch từ ngôn ngữ nguồn sang ngôn ngữ đích. Trong *dịch dựa trên luật chuyển đổi*, đầu tiên chúng ta cần phải phân tích cú pháp của câu vào, rồi áp dụng các luật chuyển đổi để biến đổi cấu trúc câu này ở ngôn ngữ nguồn sang cấu trúc của ngôn ngữ đích; cuối cùng ta mới dịch ra câu hoàn chỉnh. Đối với *dịch liên ngữ*, câu vào được phân tích thành một dạng biểu diễn trừu tượng hóa về ngữ nghĩa, được gọi là "*interlingua*", sau đó ta tìm cách xây dựng câu đích phù hợp nhất với "*interlingua*" này. *Dịch máy thống kê* có cách tiếp cận hoàn toàn khác, khả năng dịch có được là dựa trên các mô hình thống kê được huấn luyện từ các ngữ liệu song ngữ.

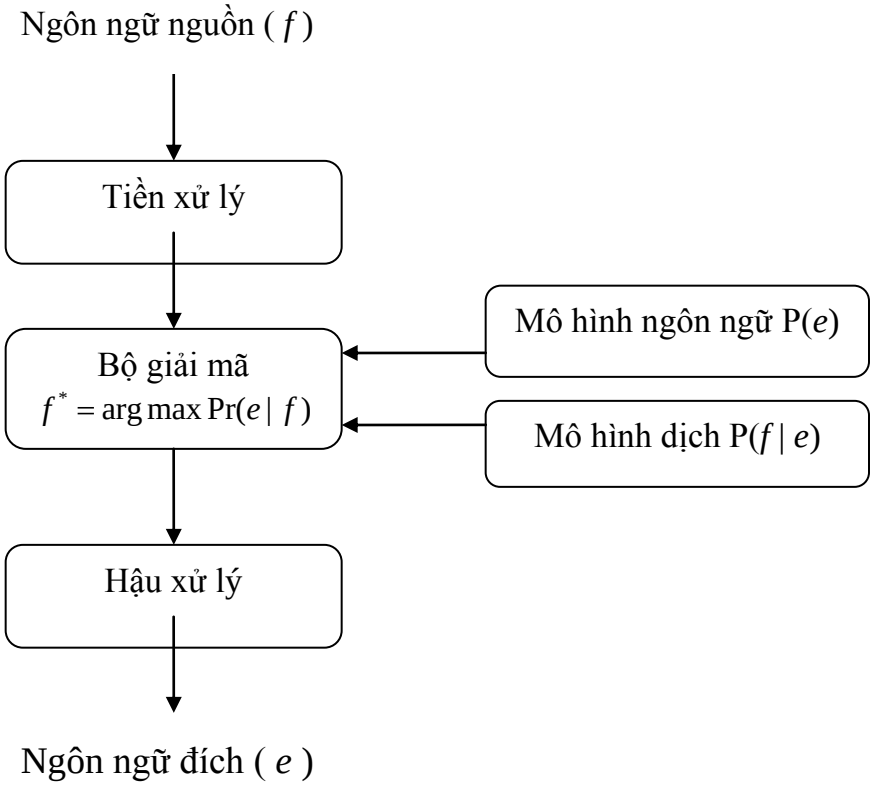
Mô hình của Brown (hay còn gọi là mô hình IBM) biểu diễn quá trình dịch bằng một mô hình kênh nhiễu bao gồm ba thành phần: một *mô hình dịch* (translation model), có nhiệm vụ liên hệ các từ, cụm từ tương ứng của các ngôn ngữ khác nhau; một *mô hình ngôn ngữ* (Language Model), đại diện cho ngôn ngữ đích; một *bộ giải mã* (decoder), kết hợp mô hình dịch và mô hình ngôn ngữ để thực hiện nhiệm vụ dịch.

Thường thì LM được gán trọng số cao hơn các thành phần khác trong hệ thống dịch, bởi vì ngữ liệu đơn ngữ dùng để huấn luyện LM lớn hơn nhiều ngữ liệu song ngữ, do đó có độ tin cậy lớn hơn. Och đã chỉ ra rằng việc tăng kích cỡ của LM cải thiện điểm BLEU – tiêu chuẩn phổ biến để đánh giá chất lượng dịch máy.



2.1.1-6 Tăng kích cỡ LM cải thiện điểm BLEU

Trong mô hình đầu tiên của Brown, mô hình dịch dựa trên kiểu *từ-thành-từ* và chỉ cho phép ánh xạ một từ trong ngôn ngữ nguồn đến một từ trong ngôn ngữ đích. Nhưng trong thực tế, ánh xạ này có thể là một-một, một-nhiều, nhiều-nhiều hoặc một-không. Thế nên nhiều nhà nghiên cứu đã cải tiến chất lượng của SMT bằng cách sử dụng *dịch dựa trên cụm* (phrase-based translation) .



2.2.1-7 Kiến trúc của một hệ thống SMT

3.2 Nguyên lý và các thành phần:

Cho trước câu ngôn ngữ nguồn f, mục tiêu của mô hình dịch máy là tìm ra câu e của ngôn ngữ đích sao cho xác suất P(e|f) là cao nhất.

Có nhiều cách tiếp cận để tính được xác suất P(e|f), tuy nhiên cách tiếp cận trực quan nhất là áp dụng công thức Bayes:

$$P(e|f) = \frac{P(e)P(f|e)}{P(f)}$$

Trong đó P(f|e) là xác suất câu ngôn ngữ nguồn là bản dịch của câu ngôn ngữ đích, còn P(e) là xác suất xuất hiện câu e trông ngôn ngữ. Việc tìm kiếm câu e* phù hợp chính là việc tìm kiếm e* làm cho giá tri P(e*)P(f|e*) là lớn nhất.

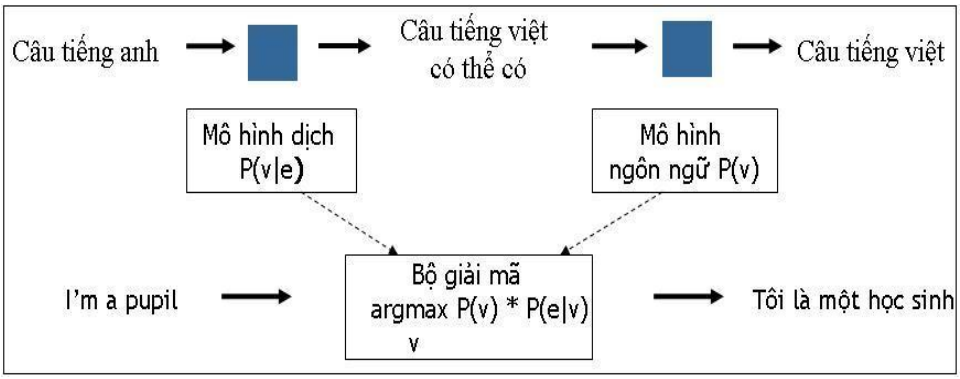
Để mô hình dịch là chính xác, thì công việc tiếp theo là phải tìm ra tất cả các câu e* có thể có trong ngôn ngữ đích từ câu ngôn ngữ nguồn f. Thực hiện công việc tìm kiếm hiệu quả chính là nhiệm vụ của bộ giải mã (decoder). Như vậy, một mô hình dịch máy bao gồm 3 thành phần:

- Mô hình ngôn ngữ: Tính toán được xác suất của câu ngôn ngữ nguồn. Thành phần này chính là mô hình ngôn ngữ.

- Mô hình dịch: Cho biết xác suất của câu ngôn ngữ nguồn là bản dịch từ câu ngôn ngữ đích .

- Bộ giải mã: Tìm kiếm tất cả các câu ngôn ngữ đích e có thể có từ câu ngôn ngữ nguồn f.

Mô hình dịch từ tiếng Anh sang tiếng Việt có thể hình dung thông qua biểu đồ dưới đây:



Hình 2.2-8 Mô hình dịch máy thống kê từ tiếng Anh sang tiếng Việt

3.2.1 Mô hình ngôn ngữ

Mô hình ngôn ngữ (Language Model - LM) là các phân phối xác suất trên một ngữ liệu đơn ngữ, được sử dụng trong nhiều bài toán khác nhau của xử lý ngôn ngữ tự nhiên, ví dụ như: dịch máy bằng phương pháp thống kê, nhận dạng giọng nói, nhận dạng chữ viết tay, sửa lỗi chính tả, …. Thực chất, mô hình ngôn ngữ là một hàm chức năng có đầu vào là một chuỗi các từ và đầu ra là điểm đánh giá xác suất một người bản ngữ có thể nói chuỗi đó. Chính vì vậy, một mô hình ngôn ngữ tốt sẽ đánh giá các câu đúng ngữ pháp, trôi chảy cao hơn một chuỗi các từ có thứ tự ngẫu nhiên, như trong ví dụ sau:

P("hôm nay trời nắng") > P("trời nắng nay hôm")

N-gram:

Nhiệm vụ của mô hình ngôn ngữ là cho biết xác suất của một câu $w_1w_2...w_m$ là bao nhiêu. Theo công thức Bayes: P(AB) = P(B|A) * P(A), thì:

$$P(w_1w_2\ldots w_m) = P(w_1) * P(w_2|w_1) * P(w_3|w_1w_2) *\ldots* P(w_m|w_1w_2\ldots w_{m-1})$$

Theo công thức này, mô hình ngôn ngữ cần phải có một lượng bộ nhớ vô cùng lớn để có thể lưu hết xác suất của tất cả các chuỗi độ dài nhỏ hơn m. Rõ ràng, điều này là không thể khi m là độ dài của các văn bản ngôn ngữ tự nhiên (m có thể tiến tới vô cùng). Để có thể tính được xác suất của văn bản với lượng bộ nhớ chấp nhận được, ta sử dụng xấp xỉ Markov bậc n:

$$P(w_m|w_1,w_2,\ldots, w_{m-1}) = P(w_m|w_{m-n},w_{n-m+1}, \ldots,w_{m-1})$$

Nếu áp dụng xấp xỉ Markov, xác suất xuất hiện của một từ (w_m) được coi như chỉ phụ thuộc vào n từ đứng liền trước nó ($w_{m-n}w_{m-n+1}\ldots w_{m-1}$) chứ không phải phụ thuộc vào toàn bộ dãy từ đứng trước ($w_1w_2\ldots w_{m-1}$). Như vậy, công thức tính xác suất văn bản được tính lại theo công thức:

$$P(w_1w_2\ldots w_m) = P(w_1) * P(w_2|w_1) * P(w_3|w_1w_2) *\ldots* P(w_{m-1}|w_{m-n-1}w_{m-n}\ldots w_{m-2})* P(w_m|w_{m-n}w_{m-n+1}\ldots w_{m-1})$$

Với công thức này, ta có thể xây dựng mô hình ngôn ngữ dựa trên việc thống kê các cụm có ít hơn n+1 từ. Mô hình ngôn ngữ này gọi là mô hình ngôn ngữ N-gram.

Một cụm N-gram là một dãy con gồm n phần tử liên tiếp của 1 dãy các phần tử cho trước (trong bộ dữ liệu huấn luyện)

Các phần tử được xét ở đây thường là kí tự, từ hoặc cụm từ; tùy vào mục đích sử dụng. Dựa vào số phần tử của 1 cụm N-gram, ta có các tên gọi cụ thể:

N = 1: Unigram

N = 2: Bigram

N = 3: Trigram

Công thức tính xác suất thô:

Gọi $C(w_{i-n+1}...w_{i-1}w_i)$ là tần số xuất hiện của cụm $w_{i-n+1}...w_{i-1}w_i$ trong tập văn bản huấn luyện.

Gọi $P(w_i|w_{i-n+1}...w_{i-1})$ là xác suất w_i đi sau cụm $w_{i-n+1}..w_{i-2}w_{i-1}$.

Ta có công thức tính xác suất như sau:

$$P(w_i|w_{i-n+1}...w_{i-1}) = \frac{C(w_{i-n+1}...w_{i-1}w_i)}{\sum_w C(w_{i-n+1}...w_{i-1}w)}$$

Dễ thấy, $\sum_w C(w_{i-n+1}..w_{i-1}w)$ chính là tần số xuất hiện của cụm $w_{i-n+1}...w_{i-1}$ trong văn bản huấn luyện. Do đó công thức trên viết lại thành:

$$P(w_i|w_{i-n+1}...w_{i-1}) = \frac{C(w_{i-n+1}...w_{i-1}w_i)}{C(w_{i-n+1}...w_{i-1})}$$

Tỉ lệ ở vế phải còn gọi là tỉ lệ tần số. Cách tính xác suất dựa vào tỉ lệ tần số còn gọi là ước lượng xác suất cực đại.

Khi sử dụng mô hình N-gram theo công thức trên, sự phân bố không đều trong tập văn bản huấn luyện có thể dẫn đến các ước lượng không chính xác. Khi các N-gram phân bố thưa, nhiều cụm n-gram không xuất hiện hoặc chỉ có số lần xuất hiện nhỏ, việc ước lượng các câu có chứa các cụm n-gram này sẽ có kết quả tồi. Với V là kích thước bộ từ vựng, ta sẽ có V^n cụm N-gram có thể sinh từ bộ từ vựng. Tuy nhiên, thực tế thì số cụm N-gram có nghĩa và thường gặp chỉ chiếm rất ít.

Để khắc phục điều này, người ta đã đưa ra các phương pháp "*làm mịn*" kết quả thống kê nhằm đánh giá chính xác hơn (mịn hơn) xác suất của các cụm N-gram. Các phương pháp "làm mịn" đánh giá lại xác suất của các cụm N-gram bằng cách:

- Gán cho các cụm N-gram có xác suất 0 (không xuất hiện) một giá trị khác 0.
- Thay đổi lại giá trị xác suất của các cụm N-gram có xác suất khác 0 (có xuất hiện khi thống kê) thành một giá trị phù hợp (tổng xác suất không đổi).

Các phương pháp làm mịn có thể được chia ra thành loại như sau:

- Chiết khấu (Discounting): giảm (lượng nhỏ) xác suất của các cụm Ngram có xác suất lớn hơn 0 để bù cho các cụm Ngram không xuất hiện trong tập huấn luyện.
- Truy hồi (Back-off) : tính toán xác suất các cụm Ngram không xuất hiện trong tập huấn luyện dựa vào các cụm Ngram ngắn hơn có xác suất lớn hơn 0

- Nội suy (Interpolation): tính toán xác suất của tất cả các cụm Ngram dựa vào xác suất của các cụm Ngram ngắn hơn.

3.3 Mô hình dịch:

Mô hình dịch có 3 hướng tiếp cận chính:

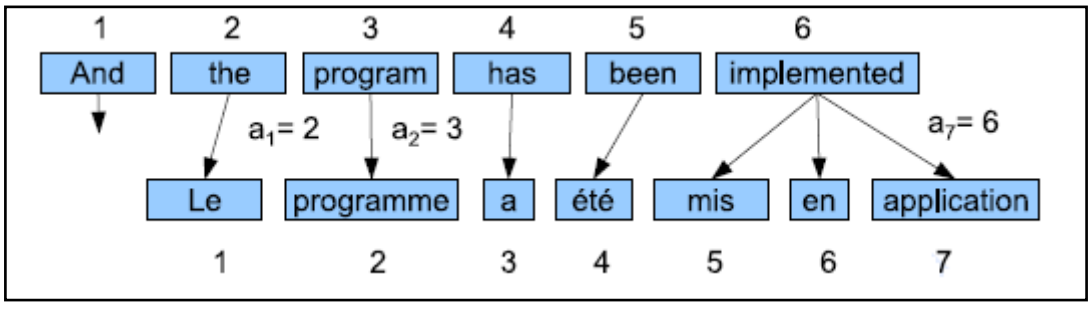
- Mô hình dịch dựa trên từ (word-based)
- Mô hình dịch dựa trên cụm từ (phrase-based)
- Mô hình dịch dựa trên cú pháp (syntax-based)

Cả 3 hướng tiếp cận trên đều dựa trên một tư tưởng. Đó là sự tương ứng giữa hai câu (alignment)

3.3.1 Sự gióng hàng (alignment):

Tất cả các mô hình dịch thống kê đều dựa trên sự tương ứng của từ. Sự tương ứng của từ ở đây chính là một ánh xạ giữa một hay nhiều từ của ngôn ngữ nguồn với một hay nhiều từ của ngôn ngữ đích trong tập hợp các câu văn bản song ngữ.

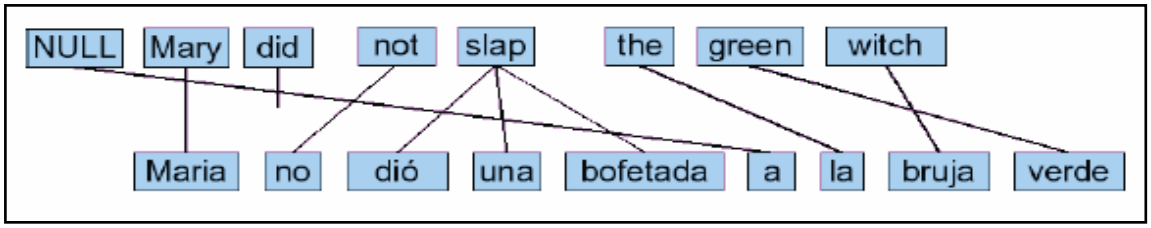
Theo nguyên tắc, chúng ta có thể có mối liên hệ tùy ý giữa các từ của ngôn ngữ nguồn với các từ của ngôn ngữ đích. Tuy nhiên, để cho đơn giản, mô hình dịch máy dựa trên từ (word-based) đưa ra một giả định: mỗi từ của ngôn ngữ đích chỉ tương ứng với một từ của ngôn ngữ nguồn. Nếu áp dụng giả định này, chúng ta có thể biểu diễn một sự tương ứng từ bằng chỉ số của các từ trong ngôn ngữ nguồn tương ứng với từ trong ngôn ngữ đích. Như trong ví dụ ở hình dưới đây có thể biểu diễn một tương ứng từ giữa tiếng Pháp và tiếng Anh bởi một dãy các chỉ số như sau: A = 2, 3, 4, 5, 6, 6, 6.



Hình 2.3.1-9 Sự tương ứng một - một giữa câu tiếng Anh và câu tiếng Pháp

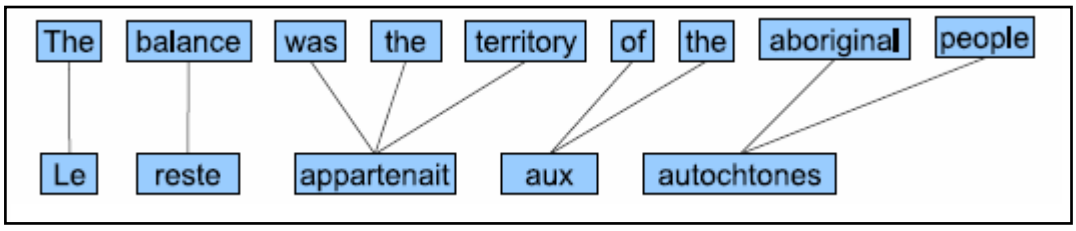
Trong thực tế, có rất nhiều từ ở ngôn ngữ đích không tương ứng với từ nào trong ngôn ngữ nguồn. Để cho tổng quát, ta thêm một từ vô giá trị (null) vào đầu câu ngôn ngữ nguồn và những từ ở ngôn ngữ đích không tương ứng với từ nào sẽ được ánh

xạ với từ vô giá trị đó. Hình 2.3.1-10 ở dưới thể hiện một tương ứng từ giữa hai câu tiếng Anh và tiếng Tây Ban Nha khi cho thêm từ vô giá trị vào đầu câu tiếng Anh.

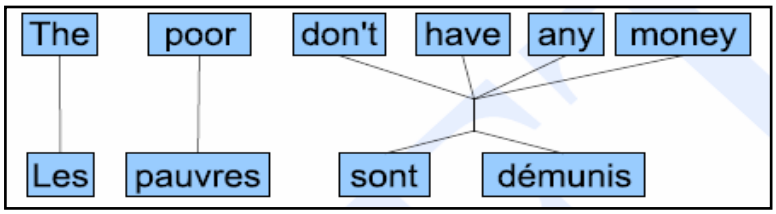


Hình 2.3.1-10 Sự tương ứng giữa câu tiếng Anh với câu tiếng Tây Ban Nha khi cho thêm từ vô giá trị (null) vào đầu câu tiếng Anh

Trong khi mô hình dịch dựa trên từ (word-based) chỉ giải quyết trường hợp một từ của ngôn ngữ đích chỉ tương ứng bởi một từ của ngôn ngữ nguồn, thì mô hình dịch dựa trên cụm từ (pharse-based) có thể giải quyết cả hai trường hợp còn lại là: một từ của ngôn ngữ này tương ứng với nhiều từ của ngôn ngữ kia và nhiều từ của ngôn ngữ này tương ứng với nhiều từ của ngôn ngữ kia. Hình 2.3.1-11 và 2.3.1-12 ở dưới minh họa các tương ứng nói trên.



Hình 2.3.1-11 Sự tương ứng một - nhiều giữa câu tiếng Anh với câu tiếng Pháp



Hình 2.3.1-12 Sự tương ứng nhiều - nhiều giữa câu tiếng Anh với câu tiếng Pháp.

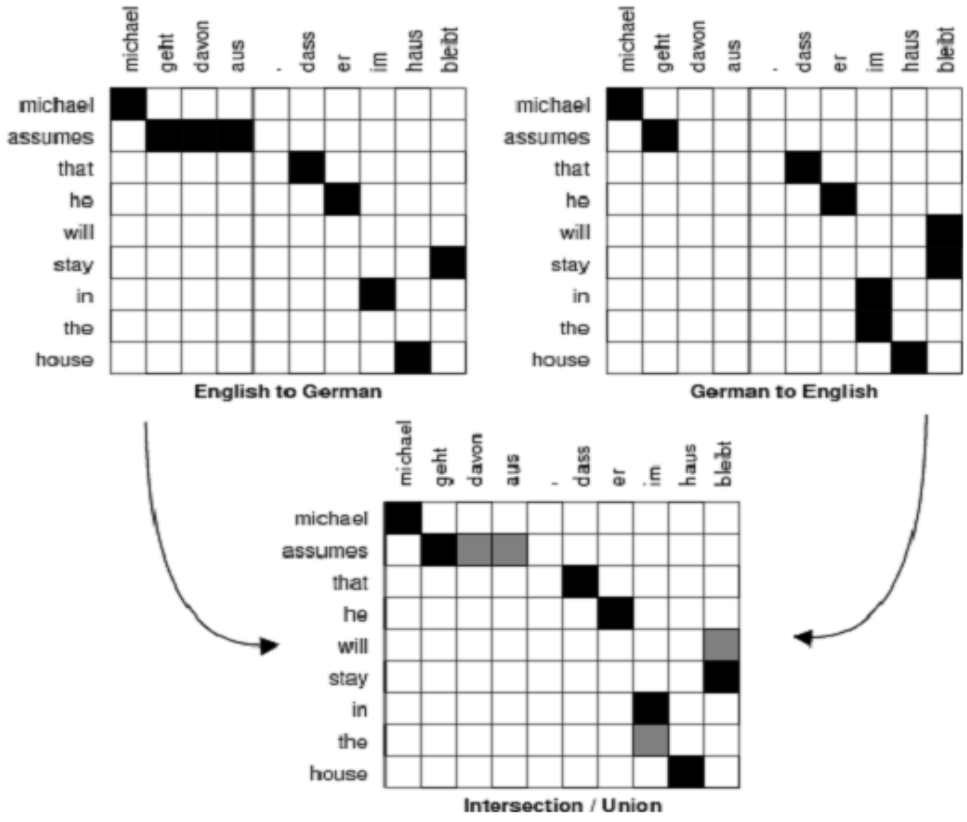
3.3.1.1 Gióng hàng từ

Mô hình gióng hàng từ là cơ sở để chích ra bảng cụm từ từ các văn bản ngôn ngữ song song (parallel corpus). Gióng hàng từ là một chủ đề nghiêm cứu được nhận rất nhiều quan tâm.

GIZA++ là công cụ cơ bản nhất để tạo ra gióng hàng từ. Công cụ này được thực hiện các mô hình cơ bản của IBM là các nghiên cứu của dịch máy thống kê đầu tiên.

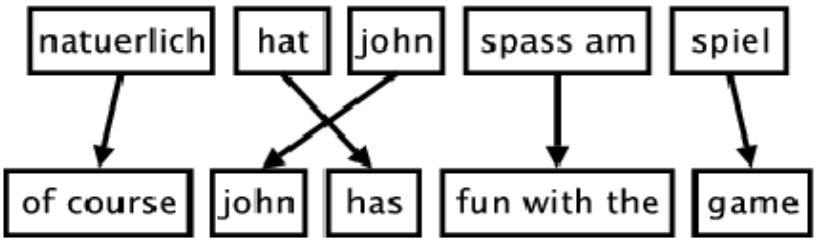
Tuy nhiên các mô hình này có vẫn có một số các hạn chế. Quan trọng nhất là nó cho phép ít nhất một từ tiếng Anh được gióng hàng với một từ nước ngoài.

Trước tiên văn bản song ngữ được gióng hàng song song: ví dụ từ tiếng Anh sang tiếng Tây Ban Nha. Nó sinh ra hai gióng hàng từ mà có thể được điều chỉnh. Nếu chúng ta lấy giao của hai gióng hàng này thì sẽ nhận được một gióng hàng có độ chính xác cao (high-precision alignment) nếu lấy hợp của hai gióng hàng này ta được một gióng hàng có độ lệch cao được minh họa hình dưới



3.3.1.2 Mô hình dịch dựa trên cụm từ

Mô tả quá trình dịch theo cụm từ: đầu vào là các phân đoạn theo các cụm từ của một câu (phrases). Mỗi một cụm từ được dịch sang một cụm từ của tiếng anh, các cụm từ đầu ra có thể sắp xếp lại.



Mô hình dịch cụm từ là dựa trên mô hình kênh nhiễu (noisy channel model). Sử dụng các luật Bayes để mô hình lại khả năng dịch cho việc dịch một câu tiếng nước ngoài f sang câu tiếng anh e như sau:

$$\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}}(\mathbf{f}|\mathbf{e})\ (\mathbf{e})$$

Nó cho thể hiện mô hình ngô ngữ e và mô hình dịch với p(f|e)

Trong quá trình giải mã, đầu vào là câu tiếng nước ngoài f được phân đoạn thành I cụm từ

Giả thiết là phân bố xác suất $\bar{f}_1^{\,I}$ đều trên toàn bộ các phân đoạn có thể có.

Mỗi cụm từ tiếng nước ngoài $\bar{f}_i$ trong $\bar{f}_1^{\,I}$ được dịch sang cụm từ tiếng Anh $\bar{e}_i$. Các cụm từ tiếng Anh có thể sắp xếp lại dịch cụm từ được mô hình hóa bởi xác suất phân bố $\phi(\bar{f}_i|\bar{e}_i)$.

Sự sắp xếp lại các cụm đầu ra trong tiếng Anh được mô hình mô bởi phân bố xác xuất bóp méo (distortion probabilityDistribution) $d(start_i,end_{i-1})$, với $start_i$ là vị trí bắt đầu của cụm từ tiếng nước ngoài đã được dịch sang cụm từ tiếng Anh thứ i và

end_{i-1} là vị trí kết thúc của cụm từ tiếng nước ngoài dịch sang cụ từ tiếng Anh thứ i-1.

Chúng ta sử dụng mô hình bóp méo đơn giản

$$d(start_i,end_{i-1}) = \alpha^{|start_i - end_{i-1} - 1|}$$

với giá trị tham số khả năng bóp méo là α. Để xác định kích cỡ của độ dài đầu ra, chúng ta giới thiệu nhân tố ω (được gọi là giá trị từ) cho mỗi từ tiếng Anh đã sinh ra được thêm vào mô hình ngôn ngữ P_{LM} tối ưu hóa quá trình thực hiện thông thường nhân tố này lớn hơn 1.

Tổng quát một câu đầu ra tiếng Anh tốt nhất e_{best} được dịch từ câu tiếng nước ngoài f theo mô hình vừa đề xuất là :

$$e_{best} = \text{argmax}_{\mathbf{e}}(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}}(\mathbf{f}|\mathbf{e})\ p_{LM}(\mathbf{e})\ \omega^{\text{length}(\mathbf{e})}$$

Trong đó p(f/e) được phân chia thành:

$$p(\bar{f}_1^{\,I}|\bar{e}_1^{\,I}) = \Phi_{i=1}^{\,I}\ \phi(\bar{f}_i|\bar{e}_i)\ d(\text{start}_i, \text{end}_{i-1})$$

3.3.1.3 Các phương pháp để học dịch trên cụm từ

Phần lớn các phương pháp được giới thiệu hiện nay sử dụng gióng hàng từ để tạo ra bảng dịch cụm từ (phrase translation table).

Marcu and Wong

Marcu and Wong (EMNLP, 2002) giới thiệu việc tổ chức các tương ứng cụm từ trực tiếp từ văn bản song ngữ. Họ giới thiệu một mô hình khả năng kết nối dựa trên cụm từ sinh đồng thời từ câu của ngôn ngữ nguồn và ngôn ngữ đích trong một văn bản song song.

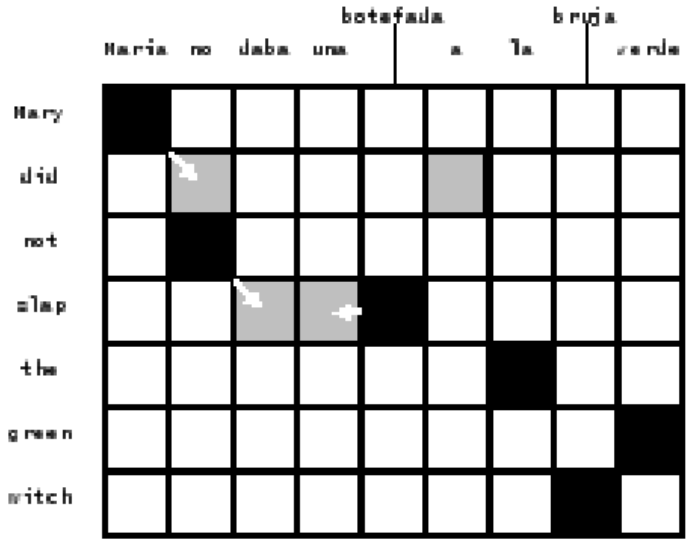
Học ước lượng cực đại (Expectation Maximization) trong hệ thống của Marcu và Wong thực hiện cả trong phân bố khả năng kết nối $\phi(\bar{e}, \bar{f})$, nó thể hiện xác suất cặp cụ từ $\bar{e}$ và $\bar{f}$ được dịch tương đương và phân bố kết nối d(i,j),

thể hiện xác suất một cụm từ tại vị trí i sang một cụm từ vị trí j.

Để sử dụng mô hình này trong hệ thống của chúng ta không quan tâm đến ước lượng xác suất kết nối của Marcu và Wong.

Och và Ney

Och và Ney (Computational Linguistics, 2003) giới thiệu một tiếp cận heuristic để tinh chỉnh các gióng hàng đạt được từ Giza++. Tối thiểu các điểm gióng hàng của phần giao nhau của hai gióng hàng ban đầu được giữ lại. Và tối đa là các điểm của phần hợp hai gióng hàng ban đầu sẽ được xem xét cẩn thận hình sau sẽ minh họa cho phần này:



Điểm giao hình màu đen còn điểm hợp là phần màu xám nhạt.

Och và Ney khai thác không gian giữa phần giao và hợp với phần mở rộng heuristics bắt đầu với phần giao và thêm các điểm gióng hàng vào. Quyết định điểm thêm vào dựa vào giá trị điều kiện :

- Nó có là một điểm gióng hàng tiềm năng hay không từ tiếng Anh sang tiếng nước ngoài?
- Điểm gần với điểm tiềm năng này có phải là các điểm đã được thiết lập không?
- Các điểm gần đó có kề trực tiếp (theo khối), hoặc là theo đường chéo hay không?
- Từ tiếng Anh hoặc từ tiếng nước ngoài là điểm tiềm năng kết nối mà chưa được gióng hàng này có xa không và chúng có đều không được gióng hàng không?
- Xác Suất của các từ cho các điểm tiềm năng là bao nhiêu?

Och và Ney thực hiện song song trong mô tả của họ về các điểm gióng hàng được thêm vào trong phương pháp tinh chỉnh trong Moses thực hiện lại phương pháp này.

Các tiến trình xử lý heuristic như sau:

Bắt đầu với phần giao của hai gióng hàng chỉ thêm một điểm gióng hàng mới nếu tồn tại trong phần hợp của hai gióng hàng đã có (luôn yêu cầu một điểm gióng hàng mới kết nối với tối thiểu với một từ chưa được gióng hàng trước đó).

Trước tiên mở rộng chỉ các điểm gióng hàng liền kề trực tiếp. Kiểm tra các điểm tiềm năng từ góc phải trên của ma trận gióng hàng, kiểm tra các điểm gióng hàng cho từ tiếng Anh đầu tiên, và tiếp tục cho các từ tiếng Anh tiếp theo.

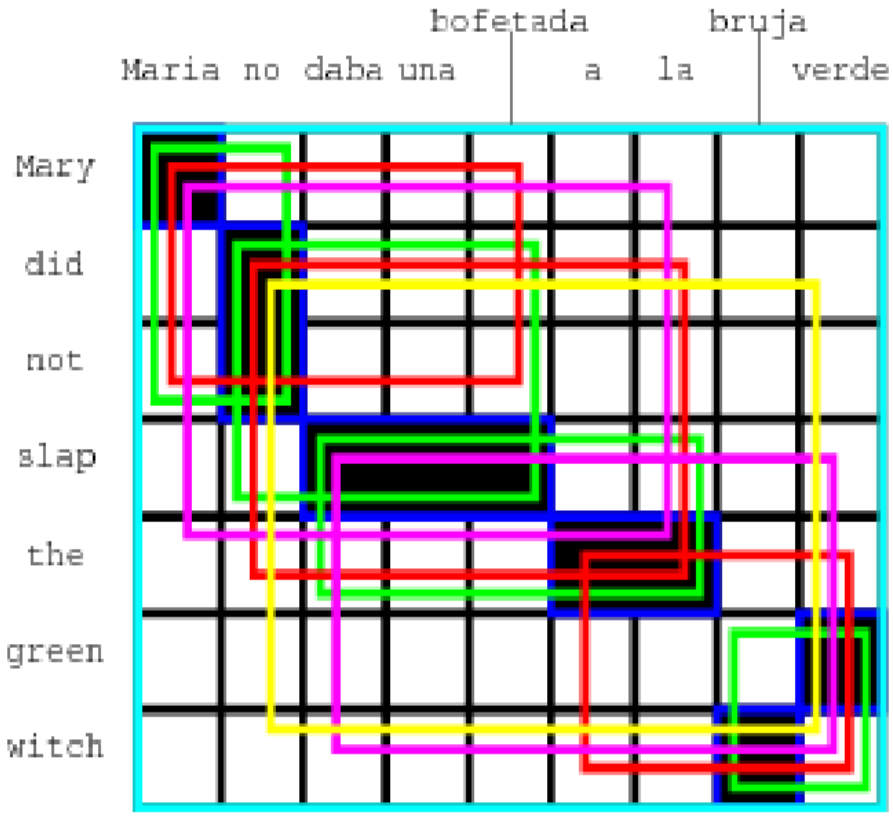
Việc này được lặp lại cho đến khi không còn điểm gióng hàng nào thêm vào nữa.

Cuối cùng chúng ta thêm các điểm gióng không gần kề nhưng với yêu cầu như trên.

Chúng ta thu thập tất cả các cặp cụm từ được gióng hàng mà thành phần của nó là các gióng hàng từ. Các từ trong cặp cụm từ hợp lệ chỉ được gióng hàng với một cụm từ khác và không được gióng hàng với các từ bên ngoài. Tập các cụm từ song song BP được định nghĩa bởi công thức sau(Zens, KI 2002):

$$BP(f_1^J, e_1^J, A) = \{(f_i^{j+m}, e_i^{i+n})\} : \forall (i', j') \in A : j <= j' <= j+m \leftrightarrow i <= i' <= i+m$$

Hình sau hiển thị các cặp cụm từ được thu thập dựa vào định nghĩa này dựa vào đó để gióng hàng



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde,green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada ala, slap the), (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap), (nodaba una bofetada a la, did not slap the), (a la bruja verde, the green witch) (Maria no dabauna bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the greenwitch), (no daba una bofetada a la bruja verde, did not slap the green witch), (Maria no dabauna bofetada a la bruja verde, Mary did not slap the green witch). Cho các cặp cụm từ đã được thu thập chúng ta ước lượng phân bố xác suất dịch cụm từ bằng tần suất sau:

$$\phi(\bar{f}|\bar{e}) = \text{count}(\bar{f},\bar{e})\ \Sigma_{\bar{f}}\ \text{count}(\bar{f},\bar{e})$$

Tillmann (EMNLP, 2003) giới thiệu một biến thể của phương pháp này bắt đầu với các gióng hàng cụm từ dựa trên phần giao nhau của hai gióng hàng Giza và sử dụng các điểm của phần hợp để mở rộng.

Venugopal, Zhang, and Vogel

Venugopal et al. (ACL 2003) cho phép thu thập các cặp cụm từ mà bị vi phạm với các gióng hàng từ. Họ giới thiệu một số phương pháp để tính điểm nhận được tính chắc chắn với gióng hàng từ, như các xác suất dịch từ vựng, độ dài cụm từ,… để tính toán.

Zhang et al. (2003) giới thiệu phương pháp gióng hàng cụm từ dựa trên các gióng hàng từ và cố gắng tìm các phân đoạn duy nhất của các cặp câu tương tự như là Marcu và Wong. Điều này cho phép họ ước lượng được phân bố xác suất kết nối mà nó có thể không quan trọng trong các phân bố xác suất điều kiện.

Vogel et al. (2003) nhận xét hai phương pháp và chỉ ra rằng kết hợp các bảng cụm từ được sinh ra bằng các phương pháp khác nhau sẽ cải thiện được kết quả.

3.4 Giải mã (Decode)

Bộ giải mã được phát triển đầu tiên cho mô hình dịch cụm từ được giới thiệu bởi Marcu và Wong, sử dụng phương pháp leo đồi. Nhưng phương pháp này không đủ cho việc dịch cụm danh từ (Koehn, PhD, 2003). Bộ giải mã thực hiện một tìm kiếm theo chùm (beam search) tương tự công việc của Tillmann (PhD,2001) và Och (PhD,

2002). Bắt đầu bằng việc định nghĩa các khái niệm cơ bản của các lựa chọn dịch mô tả cơ chế hoạt động của beam search và các thành phần cần thiết của nó và các ước lượng giá trị tương lai và các khái niệm về sinh danh sách n-best.

3.4.1 Translation Options

Cho một xâu các từ đầu vào, số các cụm từ được dịch có thể được áp dụng gọi mỗi một bản dịch cụm từ có thể là một lựa chọn dịch được minh họa ở hình sau để dịch một câu tiếng Tây Ban Nha

Các lựa chọn dịch này được thu thập trước bất kì giải mã nào được đưa ra điều này cho phép tìm kiếm nhanh hơn các tham khảo toàn bộ bản dịch cụm từ trong quá trình giải mã. Các lựa chọn dịch được lưu trữ với các thông tin sau:

- Từ tiếng nước ngoài đầu tiên được bao trùm
- Từ tiếng nước ngoài cuối cùng được bao trùm.
- Bản dịch cụm từ tiếng Anh.
- Xác suất bản dịch cụm từ.

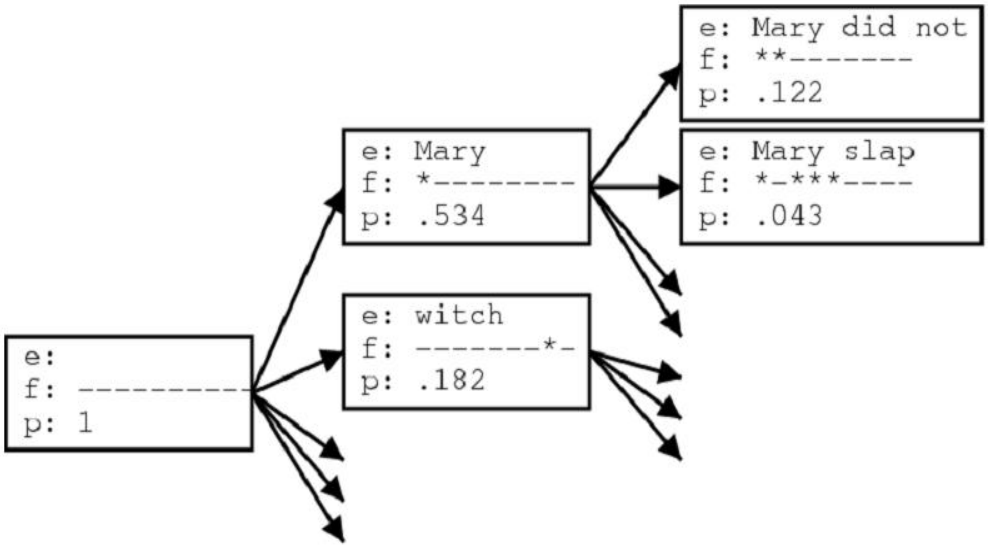
Do các bảng dịch cụm từ có thể rất lớn so với bộ nhớ lên chúng ta có thể giới hạn các lựa chọn dịch để phù hợp với khả năng tính toán. Chúng ta có thể sinh ra bản dịch cụm từ dựa vào nhu cầu chỉ bao gồm các lựa chọn dịch hợp lý cho câu đầu vào đã cho.

Bằng cách này có thể không cần sinh ra các bản dịch cụm từ đầy đủ.

3.4.2 Thuật toán cơ bản (Core Algorithm)

Giải mã dựa trên cụm từ được phát triển dựa vào thuật toán tìm kiếm chùm (beam search algorithm), câu tiếng Anh đầu ra được sinh từ trái sang phải trong các dạng của các giả thiết.

Tiến trình này được minh họa bởi hình sau:



Bắt đầu từ giả thuyết đầu tiên, mở rộng đầu tiên là từ tiếng nước ngoài Maria, nó được dịch là Mary. Từ tiếng nước ngoài được đánh dấu là đã được dịch (đánh dấu bởi dấu hoa thị). Chúng ta có thể mở rộng bằng các giả thuyết bằng việc dịch từ nước ngoài ví dụ như là lựa chọn từ bruja.

Chúng ta có thể sinh các giả thuyết mới từ giả thuyết đã mở rộng. Cho giả thuyết đầu tiên mở rộng ta sinh ra một giả thuyết mới bằng cách dịch từ no bằng did not.

Chúng ta có thể mô tả tìm kiếm chùm như sau. Trạng thái khởi tạo là không có từ tiếng nước ngoài nào được dịch sang các từ tiếng Anh được sinh ra. Một trạng thái mới được tạo ra bằng cách mở rộng từ tiếng Anh đầu ra với một bản dịch cụm từ của các từ đầu vào tiếng nước ngoài vẫn chưa được dịch.

- Chi phí hiện tại của trạng thái mới là chi phí của trạng thái ban đầu nhân với chi phí dịch, chi phí bóp méo, chi phí cho mô hình ngôn ngữ của bản dịch cụm từ được thêm vào.
- Mỗi một trạng thái tìm kiếm (giả thuyết) được thể hiện bởi:
- Một liên kết ngược trở lại với trạng thái trước đó tốt nhất (cần thiết cho việc tìm kiếm bản dịch tốt nhất của câu bằng giải thuật quay lui thông qua các trạng thái tìm kiếm).
- Các từ tiếng nước ngoài được bao trùm đến hiện tại.
- Các từ tiếng Anh cuối cùng được sinh ra (cần thiết cho việc tính toán mô hình ngôn ngữ tiếp theo).
- Cụm từ tiếng nước ngoài cuối cùng được bao trùm (cần thiết tính toán cho các chi phí bóp méo tương lai).
- Các chi phí đến hiện tại.
- Một ước lượng của chi phí tương lai (được tính toán trước và lưu trữ phù hợp).

Các trạng thái cuối cùng trong việc tìm kiếm là các giả thuyết được bao trùm trên toàn bộ từ tiếng nước ngoài. Trong số các giả thuyết có chi phí thấp nhất (xác suất cao nhất) được chọn là các bản dịch tồi nhất.

Thuật toán này cho đến nay có thể sử dụng để tìm kiếm tuyệt đối toàn bộ các khả năng dịch. Phần tiếp theo sẽ mô tả làm thế nào có thể tối ưu việc tìm kiếm bằng cách loại bỏ các giả thuyết mà phần đường dẫn của nó không cho bản dịch tốt nhất. Chúng ta giới thiệu khái niệm cơ bản của các trạng thái có thể so sánh được, cho phép chúng ta định nghĩa một cụm của các giả thuyết tốt nhất và cắt bỏ các giả thuyết không phù hợp trong cụm này

3.4.3 Kết hợp lại các giả thuyết (Recombining Hypotheses)

Kết hợp lại các giả thuyết là cách tốt để giảm không gian tìm kiếm hai giả thuyết được kết hợp lại nếu thỏa mãn :

- Các từ nước ngoài được bao trùm đến hiện tại.
- Hai từ tiếng Anh cuối cùng được sinh ra.
- Cụm từ cuối cùng tiếng nước ngoài được bao trùm.

Nếu có hai đường dẫn được bắt đầu đến hai giả thuyết mà thỏa thuận thỏa mãn các thuộc tính trên chúng ta chỉ giữ lại một giả thuyết có chi phí thấp hơn, ví dụ giả thuyết với chi phí tối thiểu đến hiện tại. Giả thuyết kia không phải một phần của đường dẫn cho bản dịch tốt nhất chúng ta loại bỏ nó.

Chú ý rằng giả thuyết cấp thấp có thể là một phần của đường dẫn cho bản dịch tốt nhất thứ hai điều này là quan trọng trong việc sinh danh sách n giả thuyết tốt nhất.

3.4.4 Tìm kiếm chùm (Beam Search)

Chúng ta sẽ ước lượng có bao nhiêu giả thuyết được sinh ra trong quá trình tìm kiếm. Xem xét các giá trị có thể cho các thuộc tính các giả thuyết duy nhất, chúng ta có thể ước lượng cận trên của số các giả thuyết bởi N.

$$2^{n_f} |V_e|^2 \bar{n_f}$$

Trong đó n_f là số các từ tiếng nước ngoài và $|V_e|$ là kích thước của các từ vựng tiếng Anh. Thực tế số các từ tiếng Anh có thể được sinh ra nhỏ hơn rất nhiều so với $|V_e|$. Vấn đề chính là sự bùng nổ hàm mũ 2^{n_f} xác định khả năng của các từ tiếng nước ngoài được bao trùm bởi một giả thuyết.

Trong tìm kiếm chùm chúng ta so sánh các giả thuyết bao trùm cùng số lượng các từ tiếng nước ngoài và cắt bỏ các giả thuyết cấp dưới. Chúng ta có thể dựa trên việc xem xét các giả thuyết cấp dưới nào nằm trong đánh giá của mỗi giả thuyết tiếp theo. Tuy nhiên việc này nhìn chung là một tiêu chuẩn tồn tại khi nó nghiêng về tìm kiếm bản dịch đầu tiên dễ tìm thấy trong phần đầu của câu.

Ví dụ nếu có ba cụm từ tiếng ngoài mà dễ dàng dịch sang một cụm từ tiếng Anh thông thường, điều này có thể có chi phí nhỏ hơn việc dịch ba từ riêng rẽ sang các từ tiếng Anh. Việc tìm kiếm này được ưa dùng hơn cho bắt đầu với câu mà dễ dàng phân chia và giảm các thay thế một cách dễ dàng.

Độ đo của chúng ta trong việc cắt bỏ các giả thuyết trong tìm kiếm chùm không chỉ bao gồm các chi phí đến hiện tại mà còn gồm ước lượng chi phí tương lai. Ước lượng chi phí tương lai này thiên về các giả thuyết mà sẵn sàng được bao trùm các phân chia khó của câu và có thể dễ dàng phân chia trái, và loại bỏ các giả thuyết mà được bao trùm các phần chia đầu tiên dễ dàng.

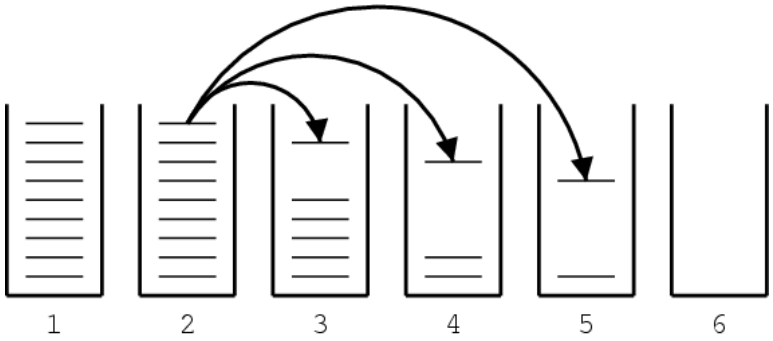
Cho chi phí đến hiện tại và ước lượng chi phí tương lai, chúng ta cắt tỉa các giả thuyết nằm bên ngoài chùm. Kích thước của chùm được xác định bởi các ngưỡng và lược đồ cắt tỉa. Ngưỡng cắt tỉa giả thuyết liên quan cắt bỏ giae thuyết với xác suất nhỏ hơn nhân tố α của các giả thuyết tốt nhất ((vd , $\alpha = 0.001$)). Lược đồ cắt tỉa giữ lại một số n chắc chắn của các giả thuyết (vd: n = 100).

Lưu ý rằng kiểu cắt tỉa không phải là rủi ro (trái ngược với sự tái tổ hợp). Nếu ước lượng chi phí tương lai là không đủ, chúng tôi có thể loại bỏ những giả thuyết trên đường dẫn đến bản dịch chi phí tốt nhất. Trong một phiên bản đặc biệt của tìm kiếm chùm, tìm kiếm A, ước lượng chi phí tương lai được yêu cầu là được phép, nghĩa là nó không bao giờ ước lượng vượt quá chi phí trong tương lai. Sử dụngtìm kiếm tốt nhất đầu tiên và chấp nhận heuristic cho phép cắt tỉa là rủi ro. Trong thực tế, loại cắt tỉa này không đủ để giảm không gian tìm kiếm.

Hình dưới đây đưa ra giả mã cho thuật toán chúng ta sử dụng cho tìm kiếm chùm.. Đối với mỗi số từ nước ngoài được bao trùm, một chồng giả thuyết được tạo ra. Giả thuyết ban đầu được đặt trong ngăn xếp cho các giả thuyết không có từ nước ngoài nào được bao trùm. Bắt đầu với giả thuyết này, các giả thuyết mới được sinh ra bởi việc gán cho các bản dịch cụm từ mà đã bao rùm các từ nước ngoài không được sử dụng trước đó. Mỗi giả thuyết có nguồn gốc được đặt trong một ngăn xếp dựa trên số lượng từ nước ngoài mà nó bao trùm..

```
initialize hypothesisStack[0 .. nf];
create initial hypothesis hyp_init;
add to stack hypothesisStack[0];
for i=0 to nf-1:
for each hyp in hypothesisStack[i]:
for each new_hyp that can be derived from hyp:
nf[new_hyp] = number of foreign words covered by new_hyp;
add new_hyp to hypothesisStack[nf[new_hyp]];
prune hypothesisStack[nf[new_hyp]];
find best hypothesis best_hyp in hypothesisStack[nf];
output best path that leads to best_hyp;
```

Tiến hành thông qua các ngắn xếp giả thuyết, thông qua mỗi giả thuyết trong ngăn xếp, sinh những giả thuyết mới cho giả thuyết này và đặt chúng vào ngăn xếp thích hợp. Sau khi một giả thuyết mới được đặt vào một ngăn xếp, ngăn xếp có thể phải được cắt tỉa bởi ngưỡng hoặc cắt tỉa lược đồ, nếu nó quá lớn. Cuối cùng, giả thuyết tốt nhất của những giả thuyết nó bao gồm tất cả các từ nước ngoài là trạng thái cuối cùng của bản dịch tốt nhất. Từ đó có thể đọc ra các từ tiếng Anh của bản dịch bằng cách làm theo các liên kết ngược trở lại trong mỗi giả thuyết.



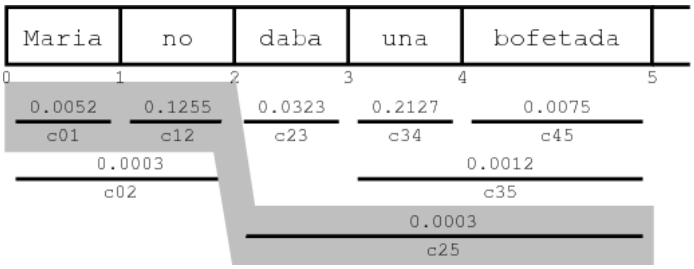
Để loại trừ các giả thuyết từ chùm chúng ta không chỉ phải xem xét chi phí cho đến hiện tại, mà còn phải ước lượng chi phí tương lai. Trong khi nó có thể tính toán chi phí rẻ nhất có thể trong tương lai cho mỗi giả thuyết, đây là tính toán rất tốn kém mà nó sẽ đánh bại mục đích của việc tìm kiếm chùm.

Chi phí trong tương lai gắn liền với những từ nước ngoài chưa được dịch. Trong khuôn khổ của mô hình dựa trên cụm từ, không chỉ những từ đơn lẻ có thể được dịch riêng lẻ, mà còn là các từ liên tiếp như một cụm từ.

Mỗi hoạt động dịch mang một chi phí dịch, các chi phí mô hình ngôn ngữ, và chi phí bóp méo. Đối với ước lượng chi phí trong tương lai, chúng ta chỉ xem xét chi phí mô hình dịch và mô hình ngôn ngữ. Chi phí mô hình ngôn ngữ thường được tính bằng một mô hình ngôn ngữ trigram. Tuy nhiên, chúng ta không biết các từ tiếng Anh trước đó cho một phép dịch. Vì vậy, chúng ta xấp xỉ gần đúng chi phí này bằng cách tính toán chi phí mô hình ngôn ngữ cho các từ tiếng Anh được tạo ra đơn lẻ. Điều đó có nghĩa, nếu chỉ có một từ tiếng Anh được sinh ra, chúng ta lấy xác suất unigram của nó. Nếu hai từ được tạo ra, chúng ta xác suất unigram của từ đầu tiên và xác suất bigram của từ thứ hai, và tiếp tục như vậy.

Đối với một chuỗi của các lựa chọn dịch chồng nhiều lần các từ nước ngoài tồn tại. Chúng ta đã mô tả làm thế nào tính toán chi phí cho mỗi tùy chọn dịch. Cách rẻ nhất để dịch chuỗi các từ nước ngoài bao gồm các tùy chọn dịch rẻ nhất. Chúng ta xấp xỉ chi phí cho một đường dẫn thông qua các lựa chọn bản dịch của các sản xuất của chi phí cho mỗi tuỳ chọn.

Hình dưới minh họa cho khái niệm này. Các tùy chọn bản dịch bao trùm các từ nước ngoài liên tiếp khác nhau và mang theo một chi phí được ước lượng: C_{ij}. Chi phí của đường dẫn thông qua chuỗi của các tùy chọn bản dịch là $c_{01}c_{12}c_{25} = 1{,}9578 * 10^{-7}$.



Các đường dẫn rẻ nhất cho một chuỗi các từ nước ngoài có thể được tính nhanh với thuật toán quy hoạch động. Lưu ý rằng nếu những từ nước ngoài không được bao trùm đến hiện tại là hai (hoặc hơn) các chuỗi không được kết nối của các từ nước ngoài, chi phí được kết hợp đơn giản chỉ là sản phẩm của các chi phí cho mỗi chuỗi liên tiếp nhau. Vì chỉ có n (n +1) / 2 các chuỗi liên tiếp cho n từ, các ước lương chi phí tương lai cho các chuỗi này có thể dễ dàng đươc tính toán lại và được lưu trữ cho mỗi câu đầu vào. Tìm các chi phí tương lai cho một giả thuyết có thể được thực hiện rất

nhanh chóng bằng cách tra cứu bảng. Điều này được xem là lợi thế hơn về tốc độ vượt qua tính toán chi phí tương lai.

3.4.5 Sinh danh sách N-giá trị tốt nhất (N-Best Lists Generation)

Thông thường, chúng ta hy vọng các bộ giải mã cung cấp cho chúng ta bản dịch tốt nhất cho một đầu vào theo mô hình. Nhưng đối với một số ứng dụng, chúng ta cũng có thể quan tâm đến bản dịch tốt nhất thứ hai, bản dịch tốt nhất thứ ba,…

Một phương pháp phổ biến trong nhận dạng giọng nói, cũng đã xuất hiện trong dịch máy là: trước tiên sử dụng một hệ thống dịch máy như bộ giải mã của chúng ta như là một mô hình cơ sở để sinh ra một tập hợp các bản dịch ứng cử cho mỗi câu đầu vào. Sau đó, thêm các đặc trưng được sử dụng để tính lại chi phí các bản dịch này.

Một danh sách n- best là một cách để đại diện cho các bản dịch ứng cử. Như một tập hợp các bản dịch có thể cũng có thể là đại diện bởi các đồ thị từ (Ueffing et al., EMNLP 2002) hoặc các cấu trúc rừng trên toàn bộ cây phân tích cú pháp (Langkilde, EACL 2002). Những cấu trúc dữ liệu thay thế cho phép đại diện nhỏ gọn hơn của một tập hợp lớn hơn nhiều của các ứng cử. Tuy nhiên, điều khó khăn hơn nhiều để phát hiện và tính điểm các thuộc tính toàn bộ trong cấu trúc dữ liệu đó.

Các cung **trong đồ thị Tìm kiếm**

Trong quá trình mở rộng trạng thái. Các giả thuyết được sinh ra và được mở rộng để liên kết chúng tạo thành một đồ thị. Các đường dẫn nhánh ra khi có các lựa chọn dịch nhiều khả năng cho một giả thuyết mà từ đó nhiều giả thuyết mới có thể được sinh ra. Đường dẫn tham gia khi các giả thuyết được kết hợp lại.

Thông thường, khi chúng ta kết hợp lại các giả thuyết, chúng ta chỉ đơn giản là loại bỏ các giả thuyết tồi yếu, vì nó không thể là một phần của con đường tốt nhất thông qua đồ thị tìm kiếm (nói cách khác, là một phần của bản dịch tốt nhất).

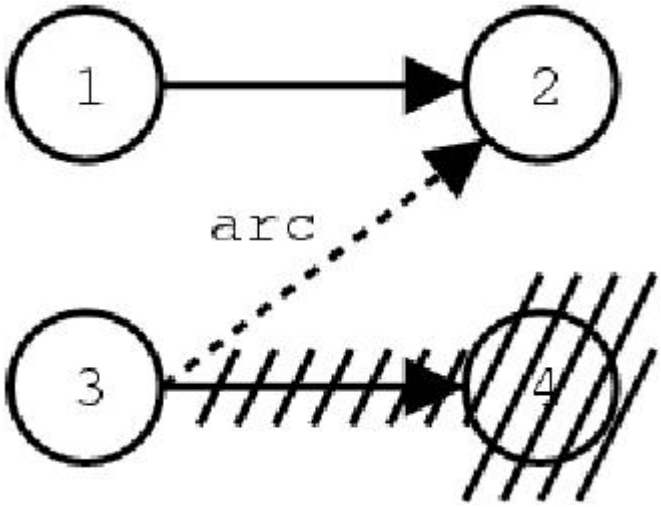
Nhưng kể từ khi chúng ta cũng quan tâm đến bản dịch tốt nhất thứ hai, chúng ta không thể chỉ đơn giản là loại bỏ thông tin về giả thuyết đó. Nếu chúng ta sẽ làm điều này, đồ thị tìm kiếm sẽ chỉ chứa một đường dẫn cho mỗi giả thuyết trong các ngăn xếp giả thuyết cuối cùng (trong đó bao gồm các giả thuyết mà bao trùm tất cả các từ nước ngoài).

Nếu chúng ta lưu trữ thông tin mà có nhiều cách để đạt được một giả thuyết, số lượng các con đường có thể cũng sẽ nhân dọc theo đường dẫn khi chúng ta thăm lại thông qua đồ thị.

Để giữ cho các thông tin về các đường dẫn được trộn, chúng ta lưu giữ hồ sơ của các kết hợp như vậy bao gồm:

- nhận dạng của giả thuyết trước đây
- nhận dạng của giả thuyết chi phí thấp hơn
- chi phí từ trước đến chi phí giả thuyết cao hơn.

Hình dưới đây cho một ví dụ cho thế hệ của vòng cung như vậy: trong trường hợp này, các giả thuyết 2 và 4 là tương đương đối với các tìm kiếm heuristic. Do đó, giả thuyết 4 đã bị xóa. Nhưng khi chúng ta muốn giữ cho thông tin về đường dẫn từ giả thuyết 3 đến 2, chúng ta lưu trữ bản ghi về vòng cung này. Cung này cũng bao gồm chi phí thêm từ giả thuyết 3 đến 4. Lưu ý rằng chi phí từ giả thuyết 1 đến giả thuyết 2 không được lưu trữ, vì nó có thể được tính toán lại từ cấu trúc dữ liệu giả thuyết.



CHƯƠNG 4: THỰC NGHIỆM

4.1 Cấu hình và hệ điều hành.

- CPU Core i3 2.1 GHz
- RAM 2G
- Hệ điều hành Ubuntu 11.04
- SWAP 5G

4.2 Các công cụ sử dụng.

4.2.1 Bộ công cụ xây dựng mô hình ngôn ngữ - SRILM:

SRILM là bộ công cụ để xây dựng và áp dụng các mô hình ngôn ngữ thống kê , chủ yếu là để sử dụng trong nhận dạng tiếng nói, gắn thẻ thống kê và phân khúc, và dịch máy thống kê. Bộ công cụ này được phát triển bởi "Phòng thí nghiệm và nghiên cứu công nghệ giọng nói SRI" từ năm 1995, có thể chạy trên nền tảng Linux cũng như Windows.

SRILM bao gồm các thành phần sau:

Một tập hợp các thư viện C++ giúp cài đặt mô hình ngôn ngữ, hỗ trợ cấu trúc dữ liệu và các chức năng tiện ích nhỏ.

Một tập hợp các chương trình thực thi thực hiện nhiệm vụ xây dựng mô hình ngôn ngữ, đào tạo và thử nghiệm mô hình ngôn ngữ trên dữ liệu, gắn thẻ hoặc phân chia văn bản, ...

Bộ công cụ SRILM có rất nhiều chương trình con, để xây dựng mô hình thêm dấu cho văn bản tiếng việt ta sử dụng chương trình chính sau :

4.2.1.1 Ngram-count:

Chương trình Ngram-count thống kê tần số xuất hiện của các cụm Ngram. Kết quả của việc thống kê được ghi lại vào một tệp hoặc sử dụng chúng để xây dựng mô hình ngôn ngữ.

4.2.2 Bộ công cụ xây dựng mô hình dịch máy thống kê – MOSES:

Moses là một hệ thống dịch máy thống kê cho phép người dùng xây dựng các mô hình dịch cho bất kỳ cặp ngôn ngữ nào với đầu vào là một tập hợp các văn bản song ngữ, được nhiều trường đại học

- tệp **moses.ini** chứa các tham số cho bộ giải mã như: đường dẫn đến tệp **phrase-table,** đường dẫn đến tệp chứa mô hình ngôn ngữ, số lượng tối đa cụm từ của ngôn ngữ đích được dịch bởi một cụm từ của ngôn ngữ nguồn, ...

Để xây dựng được mô hình dịch thống kê, ta có thể sử dụng script: **train-model.perl** với một số tham số sau:

--root-dir -- cài đặt thư mục gốc nơi lưu trữ các tệp đầu ra

--corpus -- tên của tệp văn bản huấn luyện (bao gồm cả 2 ngôn ngữ nguồn và đích)

--e -- đuôi mở rộng của tệp văn bản huấn luyện ngôn ngữ đích

--f -- đuôi mở rộng của tệp văn bản huấn luyện ngôn ngữ nguồn

--lm -- language model: <factor>:<order>:<filename> : thiết lập file cấu hình mô hình ngôn ngữ theo định dạng đã trình bày trong phần

--max-phrase-length -- độ dài lớn nhất của các cụm từ lưu trữ trong tệp **phrase-table**

- Công cụ gióng hàng GIZA++

4.2.3 Các bước huấn luyện dich và kiểm tra.

- Chuẩn hóa dữ liệu
- Dữ liệu được chia làm 2 loại
- Dữ liệu song ngữ :
- Văn bản tiếng Việt không dấu
- Văn bản tiếng Việt có dấu
- Dữ liệu đơn ngữ
- Văn bản tiếng Việt có dấu
- Xây dựng mô hình ngôn ngữ
- Xây dựng mô hình dịch
- Dịch máy
- Đánh giá kết quả dịch

4.2.4 Chuẩn hóa dữ liệu.

Sử dụng bộ Bộ công cụ được download từ trên mạng gồm các công cụ Scripts, **Tokenizer, lowercase,..** phục vụ cho việc chuẩn hoá dữ liệu như: tách từ, tách câu, chuyển sang chữ thường, …

4.2.5 Xây dựng mô hình ngôn ngữ.

Sử dụng công cụ SRILM để xây dựng mô hình ngôn ngữ. Sử dụng văn bản đơn ngữ Tiếng Việt để xây dựng mô hình ngôn ngữ. Kết quả sau khi xây dựng mô hình ngôn ngữ tri-gam:

Bảng thống kê n-gram

\data\

ngram 1=6773

ngram 2=162282

ngram 3=92846

\1-grams:

-2.6378 ! -0.9554

-3.523143 " -0.2592531

-4.542355 $ -0.1713233

-3.916798 % -0.2839231

-2.823495 ' -0.4149792

-2.759014 (-0.3535762

-2.827075) -0.4755327

4.2.6 Huấn luyện mô hình:

- Sử dụng hệ dịch MOSES kết hợp với công cụ dóng hàng GIZA++
- Sử dụng tệp văn bản song ngữ phục vụ cho bài toán bỏ dấu. Thực chất là dịch tệp tiếng nước ngoài gọi là tệp nguồn .en chính là văn bản tiếng Việt không dấu sang tệp đích là ngôn ngữ tiếng Việt .vn.
- Mô hình dịch (phrase-table).

❖ **Kết quả trong file phrase-table :**

! ' ! . ||| ! ' ! . ||| 1 1 1 1 2.718 ||| ||| 5 5

! ' ! ||| ! ' ! ||| 1 1 1 1 2.718 ||| ||| 5 5

! ' ' Do la mot viec ||| ! ' ' Đó là một việc ||| 1 1 1 0.767028 2.718 ||| ||| 1 1

! ' ' Do la mot ||| ! ' ' Đó là một ||| 1 1 1 0.767028 2.718 ||| ||| 1 1

! ' ' Do la ||| ! ' ' Đó là ||| 1 1 1 0.769353 2.718 ||| ||| 1 1

! ' ' Do ||| ! ' ' Đó ||| 1 1 1 0.797297 2.718 ||| ||| 1 1

! ' ' Duoc , neu anh ||| ! ' ' Được , nếu anh ||| 1 1 1 0.897503 2.718 ||| ||| 1 1

! ' ' Duoc , neu ||| ! ' ' Được , nếu ||| 1 1 1 0.924477 2.718 ||| ||| 1 1

! ' ' Duoc , ||| ! ' ' Được , ||| 1 1 1 0.986842 2.718 ||| ||| 1 1

! ' ' Duoc ||| ! ' ' Được ||| 1 1 1 0.986842 2.718 ||| ||| 1 1

! ' ' may cung the , ||| ! ' ' mày cũng thế , ||| 1 1 1 0.0870583 2.718 ||| ||| 1 1

! ' ' may cung the ||| ! ' ' mày cũng thế ||| 1 1 1 0.0870583 2.718 ||| ||| 1 1

! ' ' may cung ||| ! ' ' mày cũng ||| 1 1 1 0.127426 2.718 ||| |||

4.2.7 Kết quả dịch

Đầu vào : văn bản tiếng Việt không dấu	Kết quả trả ra đưa vào hệ thống thêm dấu	Văn bản chính xác
toi ngo rang co ta khong noi cho toi biet su_that .	toi ngờ rang co ta không noi cho toi biết sự_thật .	tôi ngờ rằng cô ta không nói cho tôi biết sự thật .
toi bi dau bao_tu du_doi .	toi bi đâu bao_tử dữ_dội .	tôi bị đau bao tử dữ dội .
toi hoan_toan tin_tuong vao tai_nang cua cac bac_si .	toi hoàn_toàn tin_tưởng vào tài_năng cua các bác_sĩ .	tôi hoàn toàn tin tưởng vào tài năng của các bác sĩ .
toi luc_nao cung thich nghe_noi ve	toi lúc_nào cung thích nghe_nói ve	tôi lúc nào cũng thích nghe nói về chính phủ

chinh_phu my . nhung co_phan 2 bang anh bay_gio dang_gia 2.75 bang . 10 bang anh la du tien xang cho cuoc hanh_trinh cua chung_toi .	chính_phủ my . nhung cổ_phần 2 bang anh bây_giờ đáng_giá 2.75 bang . 10 bang anh la du tiền xăng cho cuộc hành_trình cua chúng_tôi .	mỹ . những cổ phần 2 bảng anh bây giờ đáng giá 2.75 bảng . 1. 10 bảng anh là đủ tiền xăng cho cuộc hành trình của chúng_tôi.

4.2.8 Đánh giá kết quả dịch

- Chỉ số BLEU

Individual N-gram scoring

1-gram 2-gram 3-gram 4-gram 5-gram 6-gram 7-gram 8-gram 9-gram

------ ------ ------ ------ ------ ------ ------ ------ ------

BLEU: 0.0968 0.0006 0.0001 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 "ref"

Chỉ số BlEU:Là chỉ số đánh giá chất lượng dịch của máy dịch thống kê từ ngôn ngữ này sang ngôn ngữ khác. Nếu kết quả gần giống với cách hiểu tự nhiên thì chất lượng dịch càng tốt. Điểm BlEU được tính bằng cách so sánh những câu cần dịch với một tập hợp các tham chiếu dịch tốt. Sau đó lấy ra giá trị trung bình tương ứng điểm số riêng lẻ này. Chỉ số này nằm trong khoảng 0 đến 1. Nếu càng gần1 thì chất lượng dịch càng tốt (sát nghĩa).

KẾT LUẬN

Bài toán thêm dấu tiếng Việt vào văn bản không dấu đã có nhiều phương pháp tiếp cận và cho độ chính xác cao. Tuy nhiên, trong đồ án này hướng tới một tiếp cận khác sử dụng mô hình dịch máy thống kê cho việc thêm dấu cho văn bản tiếng Việt không dấu. Kết quả của đồ án đã thực hiện được mục tiêu như trong phần giới thiệu nhưng do thời gian có hạn, nên kết quả thực nghiệm còn chưa đạt được theo mong muốn. Tuy nhiên, luận văn cũng đạt được một số kết quả:

Về lý thuyết:

Tìm hiểu về bài toán thêm dấu cho văn bản tiếng việt chưa có dấu.

Tìm hiểu, nghiên cứu mô hình dịch máy thống kê với tiếp cận cho bài toán thêm dấu cho văn bản tiếng Việt

Về thực nghiệm:

- Sử dụng bộ công cụ mã nguồn mở Moses, GIZA++, SRILM, … để xây dựng mô hình dịch máy thống kê. Cài đặt và ứng dụng được mô hình dịch máy thống kê cho bài toán bỏ dấu cho văn bản tiếng Việt.
- Đánh giá kết quả.

Do thời gian có hạn, nên kết quả của thực nghiệm còn hạn chế. Trong tương lai, tôi sẽ tiếp tục nghiên cứu để cải thiện chất lượng cho mô hình bằng cách xây dựng mô hình n-gram cho văn bản tiếng Việt để sử dụng hiệu quả hơn và kết hợp với từ điển và các phương pháp nhằm cải thiện chất lượng dịch.

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo Tiếng Việt

[1]. **Thắng, Tô Hồng.** NGRAM. s.l. : Khóa luận tốt nghiệp Trường đại học Công Nghệ, 2007.

Tài liệu tham khảo Tiếng Anh

[1]. **Thắng, Tô Hồng.** Building language model for vietnamese and its application, graduation thesis. 2008.

[2]. **Brown, P. F, Cocke J., Della Pietra V., Della Pietra S., Jelinek F., Lafferty J. D., Mercer R. L., and Roossin P. S.** *A statistical approach to machine translation.* s.l. : Computational Linguistics, 1990.

[3] http://www.statmt.org/moses/

[4] MOSES Statistical Machine Translation System User Manual and Code Guide.

Philipp Koehn pkoehn@inf.ed.ac.uk University of Edinburgh