ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN&TRUYỀN THÔNG

BÙI THANH THUỶ

NGHIÊN CỨU VỀ DỊCH MÁY THỐNG KÊ DỰA VÀO CỤM TỪ VÀ ỨNG DỤNG DỊCH TỪ TIẾNG VIỆT SANG TIẾNG ANH

LUẬN VĂN THẠC SĨ: KHOA HỌC MÁY TÍNH

Thái Nguyên - 2015

LỜI CAM ĐOAN

Tôi xin cam đoan toàn bộ nội dung trong luận văn này do tôi tự nghiên cứu, đọc, dịch tài liệu, tổng hợp và thực hiện. Trong luận văn tôi có sử dụng một số tài liệu tham khảo như đã trình bày trong phần tài liệu tham khảo.

Người viết luận văn

Bùi Thanh Thủy

LỜI CẢM ƠN

Đầu tiên tôi xin gửi lời cảm ơn chân thành đến TS. **Nguyễn Văn Vinh** đã tận tình hướng dẫn, chỉ bảo cho tôi trong suốt quá trình làm luận văn. Em cũng xin cam ơn anh **Trần Hồng Việt**, nghiên cứu sinh Trường đại học công nghệ, giảng viên Trường Đại học Kinh tế kỹ thuật công nghiệp đã giúp đỡ em trong quá trình làm luận văn

Tôi cũng xin gửi lời cảm ơn đến các thầy cô trường Đại học Công nghệ thông tin và Truyền thông – Đại học Thái Nguyên, các thầy cô Viện Công nghệ thông tin đã truyền đạt những kiến thức và giúp đỡ tôi trong suốt quá trình học của mình.

Tôi cũng xin gửi lời cảm ơn tới Ban giám hiệu, Phòng Đào tạo, các đồng nghiệp trường Cao đẳng nghề Phú Thọ, gia đình và bạn bè những người đã động viên tạo mọi điều kiện giúp đỡ tôi để hoàn thành luận văn.

MỤC LỤC

DANH MỤC CÁC HÌNH

MỞ ĐẦU

1.Lý do chon đề tài

Trong quá trình phát triển và hội nhập văn hóa, kinh tế thế giới. Quá trình giao lưu giữa người Việt Nam và người nước ngoài ngày càng nhiều dẫn đến khó khăn trong quá trình giao tiếp và sử dụng văn bản tài liệu tiếng Anh. Hiện nay có nhiều hệ thống tự động dịch miễn phí trên mạng như: google translate, vietgle, vdict, lạc việt,… Những hệ thống này cho phép dịch tự động các văn bản với một cặp ngôn ngữ chọn trước (ví dụ dịch từ tiếng Anh sang tiếng Việt). Điều ấy cho thấy sự phát triển của dịch máy càng ngày càng tiến gần hơn đến ngôn ngữ tự nhiên của con người.

Vào những năm gần đây, dịch máy nói chung, dịch máy thống kê nói riêng được phát triển mạnh và ứng dụng rộng rãi. Kết quả thực tế của hệ thống dịch này rất tốt. Ngôn ngữ của máy dịch ngày càng gần với ngôn ngữ của người. Ngoài ra cùng với hệ thống dịch máy thống kê, các sản phẩm ứng dụng ngày càng nhiều giúp con người trao đổi thông tin dễ dàng hơn, tốc độ nhanh hơn và cùng với nhiều ngôn ngữ hơn.

Hiện nay, phương pháp dịch thống kê dựa trên cụm từ là phương pháp cho kết quả dịch tốt nhất hiện nay. Điều này được thể hiện của qua các hệ dịch máy của Google, Vietgle. Hơn nữa việc dịch giữa tiếng Việt sang tiếng Anh là rất cần thiết khi khối lượng văn bản tiếng Anh ngày càng lớn trong thời kỳ Việt Nam hội nhập sâu rộng với quốc tế.

Chính vì lý do đó, tôi lựa chọn và thực hiện đề tài "Nghiên cứu về dịch thống kê dựa vào cụm từ và áp dụng cho dịch từ tiếng Việt sang tiếng Anh".

2. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Nghiên cứu về các phương pháp, mô hình dịch máy thống kê

- Thử nghiệm và đánh giá kết quả dịch từ tiếng Việt sang tiếng Anh

Phạm vi nghiên cứu:

Đề tài tập trung vào nghiên cứu phương pháp dịch thống kê dựa vào cụm từ và ứng dụng dịch tài liệu, văn bản tiếng Việt, tiếng Anh.

3. Hướng nghiên cứu của đề tài

- Nghiên cứu, tìm hiểu, phân tích về dịch máy thống kê trên cơ sở cụm từ.

- Cài đặt thử nghiệm tối ưu hóa cụm từ bằng hệ dịch máy thống kế Moses

4. Phương pháp nghiên cứu

- Tìm hiểu các hệ dịch tự động đã có để tìm ra các phương pháp dịch máy mà các hệ dịch đang sử dụng.

- Nghiên cứu và đánh giá các phương pháp dịch máy, những ưu điểm và hạn chế, sau đó tìm ra phương pháp có hiệu quả và đề xuất áp dụng cho bài toán đề tài đặt ra.

- Nghiên cứu các phương pháp đánh giá chất lượng dịch máy để đánh giá hiệu quả dịch cho hệ thống đề tài đã xây dựng.

5. Ý nghĩa khoa học của đề tài

Ý nghĩa khoa học:

Dịch máy dựa vào cụm từ là một trong những phương pháp dịch máy hiệu quả nhất hiện nay. Hơn nữa dữ liệu văn bản ngày càng lớn và đa dạng. chính vì vậy nghiên cứu về hệ dịch dựa vào cụm từ và ứng dụng cho dịch Việt – Anh có ý nghĩa khoa học cũng như thực tiễn

6. Cấu trúc luận văn

+ Chương 1: Tổng quan về dịch máy

+ Chương 2: Dịch máy thống kê dựa vào cụm từ và áp dụng cho ngôn ngữ Việt _ Anh

+ Chương 3: Thực nghiêm, đánh giá

+ Kết luận

CHƯƠNG 1 – TỔNG QUAN VỀ DỊCH MÁY

1.1. Khái niệm về hệ dịch máy

1.1.1. Định nghĩa

Các hệ dịch máy (machine translation system-MT) là các hệ thống sử dụng máy tính để dịch từ một thứ tiếng (trong ngôn ngữ tự nhiên) sang một hoặc vài thứ tiếng khác.

Ngôn ngữ của văn bản cần dịch được gọi là ngôn ngữ nguồn, ngôn ngữ của văn bản đã dịch ra được gọi là ngôn ngữ đích.

1.1.2. Vai trò của dịch máy

Hiện nay trên thế giới có khoảng hơn 5000 ngôn ngữ khác nhau, với một số lượng ngôn ngữ lớn như vậy đã gây ra rất nhiều khó khăn trong việc trao đổi thông tin, trong giao tiếp, đồng thời ngăn cản sự phát triển của thương mại và mậu dịch quốc tế.

Với những khó khăn như vậy con người đã phải dùng đến một đội ngũ phiên dịch khổng lồ, để dịch các văn bản, tài liệu, lời nói, ngôn ngữ từ tiếng nước này sang tiếng nước khác. Những công việc đó mang tính chất thủ công, tỉ mỉ đòi hỏi người dịch phải làm mất rất nhiều thời gian và công sức, trong khi khối lượng văn bản cần dịch ngày càng nhiều.

Để khắc phục được những nhược điểm trên con người đã nghĩ đến việc thiết kế một mô hình tự động trong công việc dịch ngôn ngữ, do đó ngay từ khi xuất hiện chiếc máy tính điện tử đầu tiên (năm 1946) người ta đã tiến hành nghiên cứu về dịch máy. Việc đưa ra mô hình tự động cho việc dịch đã và đang được phát triển, mặc dù chưa giải quyết được triệt để lớp ngôn ngữ tự nhiên. Nhưng sự ra đời của chúng đã khẳng định được lợi ích to lớn về mặt chiến lược và phát triển kinh tế, đồng thời các vấn đề liên quan đến dịch máy

cũng là những chủ đề quan trọng của ngành khoa học máy tính, bởi chúng liên quan đến vấn đề xử lí ngôn ngữ tự nhiên, một trong những vấn đề có ý nghĩa nhất mà trí tuệ nhân tạo có khả năng giải quyết. Người ta tin rằng việc xử lí ngôn ngữ tự nhiên trong đó có dịch máy sẽ là giải pháp cho việc mở rộng cánh cửa đối thoại giữa người-máy, lúc đó con người không phải tiếp xúc với máy qua những dòng lệnh cứng nhắc nữa mà có thể giao tiếp một cách trực tiếp với máy.

1.1.3. Sơ đồ tổng quan của một hệ dịch máy

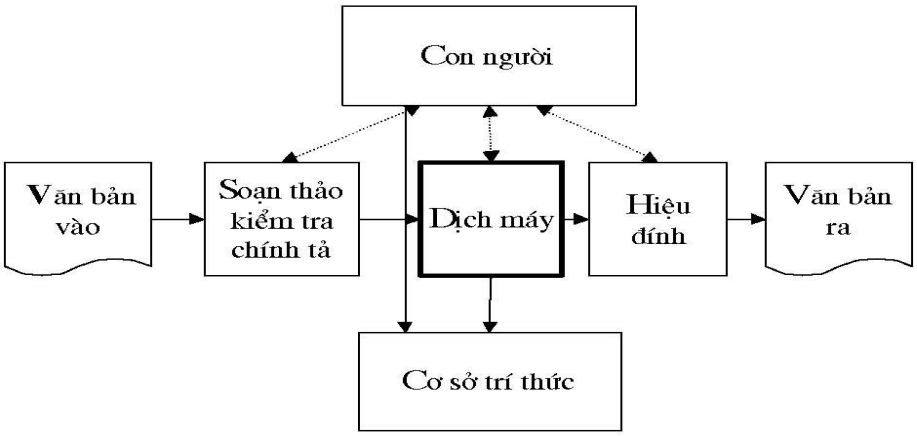
Đầu vào của một hệ dịch máy là một văn bản viết trong ngôn ngữ nguồn. Văn bản này có thể thu được từ một hệ soạn thảo hay một hệ nhận dạng chữ viết, lời nói. Sau đó văn bản có thể được chỉnh sửa lại nhờ khối soạn thảo, kiểm tra chính tả, trước khi đưa vào máy dịch.

Phần dịch máy sẽ chuyển văn bản nguồn thành văn bản viết trên ngôn ngữ đích. Và cũng qua một bộ chỉnh ra để cuối cùng thu được một văn bản tương đối hoàn chỉnh.

Trong quá trình dịch máy, hệ thống thường xuyên phải truy cập đến một khối lượng rất lớn các tri thức dịch. Tri thức dịch thông thường là các loại từ điển bao gồm: từ điển chứa bộ luật về cú pháp, từ điển về từ vựng, từ điển về thông tin ngữ nghĩa vv…..

Trong nhiều hệ thống, có thể có sự tương tác giữa người và máy trong quá trình dịch. Tương tác này thông thường có cả hai chiều (người-máy) và có thể có ở mọi giai đoạn.

Dưới đây là sơ đồ tổng quát của một hệ dịch máy:



Hình 1.1: Sơ đồ tổng quan của hệ dịch máy

1.2. Dịch máy thống kê là gì?

1.2.1. Tổng quan về dịch thống kê

Mục tiêu là dịch một văn bản từ ngôn ngữ nguồn sang ngôn ngữ dịch. Chúng ta có câu văn bản trong ngôn ngữ nguồn ("Tiếng Việt") $v_1^j = v_1, \dots, v_j$. Trong tất cả các câu có thể có trong văn bản đích, chúng ta chọn câu sao cho:

$$V1J = \arg\max p(v1J \mid e1I) \qquad (1.1)$$

1.2.1.1. Mô hình kênh nguồn

Mô hình kênh – nguồn rất tổng quát và có khả năng áp dụng cho nhiều vấn đề khác nhau như nhận dạng tiếng nói, xử lý ảnh, …Về trực giác, kênh nguồn là một kênh truyền thông mà thông tin truyên qua có thể bị nhiễu và khó nhận dạng được thông tin đúng. Giả sử T là thông tin đích ta nhận được từ cuối kênh, nhiệm vụ của chúng ta là phải đoán lại thông tin nguồn S đã truyền đi.

Sử dụng luật Bayes, chúng ta có:

$$p(v^J \mid e^I) = \frac{p(e^I \mid v^J) \times p(v^J)}{p(e^I)} \qquad (1.2)$$

Do đó công thức 1.1 tương ứng với:

$$V = \arg\max_{v} p(v^J \mid e^I) = \arg\max_{v} p(v^J) \times p(e^I \mid v^J) \quad (1.3)$$

Cách tiếp cận này được xem như là cách tiếp cận Kênh - Nguồn trong dịch máy thống kê hoặc là " công thức cơ bản của dịch thống kê". Ở đây p(v J) là mô hình ngôn ngữ của ngôn ngữ đích, p(e I | v J) là mô hình đích.

1.2.1.2 Cách tiếp cận Maximum và mô hình gióng hàng

Xác suất p(e I | v J) được phân tích qua biến ẩn được thêm vào. Ta có:

$$p(e_1^I \mid v_1^J) = \sum_{a_1^I} p(e_1^I, a_1^I \mid v_1^J) \quad (1.4)$$

Trong đó p(e1I, a1I | v1J) được gọi là mô hình gióng hàng thống kê và gióng hàng a1I được gọi là biến ẩn.

Gióng hàng xác định ánh xạ i→j=ai : Từ vị trí i của câu nguồn tương ứng với vị trí j= ai của câu đích.

Việc tìm kiếm được thực hiện dực vào cực đại biểu thức sau:

$$V_1^J = \arg\max_{v_1^J} \left\{ p(v_1^J) \times \sum_{a_1^I} p(e_1^I, a_1^I \mid v_1^J) \right\} \quad (1.5)$$

1.2.1.3. Nhiệm vụ trong dịch thống kê

Chúng ta phải giải quyết những vấn đề sau trong việc phát triển hệ thống dịch thống kê:

Mô hình: Chỉ ra cấu trúc trong sự phụ thuộc xác suất để mô hình hóa xác suất dịch $p(e^J)$ hoặc $p(v^J)$.

Huấn luyện: Huấn luyện các tham số mô hình của mô hình dịch thống kê sử dụng dữ liệu huấn luyện: đơn ngữ, song ngữ. Tiêu chuẩn huấn luyện

chuẩn của mô hình dịch máy theo cách tiếp cận kênh-nguồn là tiêu chuẩn hợp lý cực đại mà ở đây chúng ta định nghĩa giá trị tham số tối ưu mà các gía trị tham số tối ưu mà các giá trị này làm cực đại hàm hợp lý trong dữ liệu song ngữ:

$$\hat{d} = \arg\max_{\theta} xp(v_1^J \mid e_1^I)$$

Phụ thuộc vào cấu trúc của mô hình, chúng ta có thể sử dụng tần suất quan hệ hoặc thuật toán tối ưu như thuật toán EM xác định các tham số ẩn của mô hình.

Tìm kiếm: Thực hiện phép tính agrmax theo công thức trong 1.2.1 một cách hiệu quả. Có rất nhiều thuật toán để giải quyết vấn đề tìm kiếm này. Ví dụ như thuật toán qui hoạch động, A*, giải mã ngăn xếp, tìm kiếm ăn tham, ...

Tiền xử lý: Tìm các bước biến đổi thích hợp cho cả ngôn ngữ nguồn và ngôn ngữ đích để cải tiến quá trình dịch.

Trong những nhiệm vụ trên, tri thức ngôn ngữ chỉ cần thiết cho vần đề mô hình và tiền xử lý. Những vấn đề khác là các vấn đề chủ yếu dựa vào toán học và tính toán bao gồm việc phát triển hiệu quả các thuật toán.

1.2.1.4.Ưu điểm của phương pháp dịch thống kê

Cách tiếp cận thống kê có những ưu điểm sau

Dịch máy là vấn đề quyết định: Cho trước những từ trong ngôn ngữ nguồn, chúng ta phải quyết định chọn những từ trong ngôn ngữ đích. Vì vậy, nó tạo cho chúng ta một cảm giác là có thể giải quyết nó bằng định lý quyết định thống kê. Điều đó dẫn đến cách tiếp cận thống kê được đề xuất.

Mối quan hệ giữa đối tượng ngôn ngữ như từ, cụm từ và cấu trúc ngữ pháp thường yếu và mơ hồ. Để mô hình hóa những phụ thuộc này, chúng ta

cần một công thức hóa như đưa ra phân phối xác suất mà nó có thể giải quyết với những vấn đề phụ thuộc lẫn nhau.

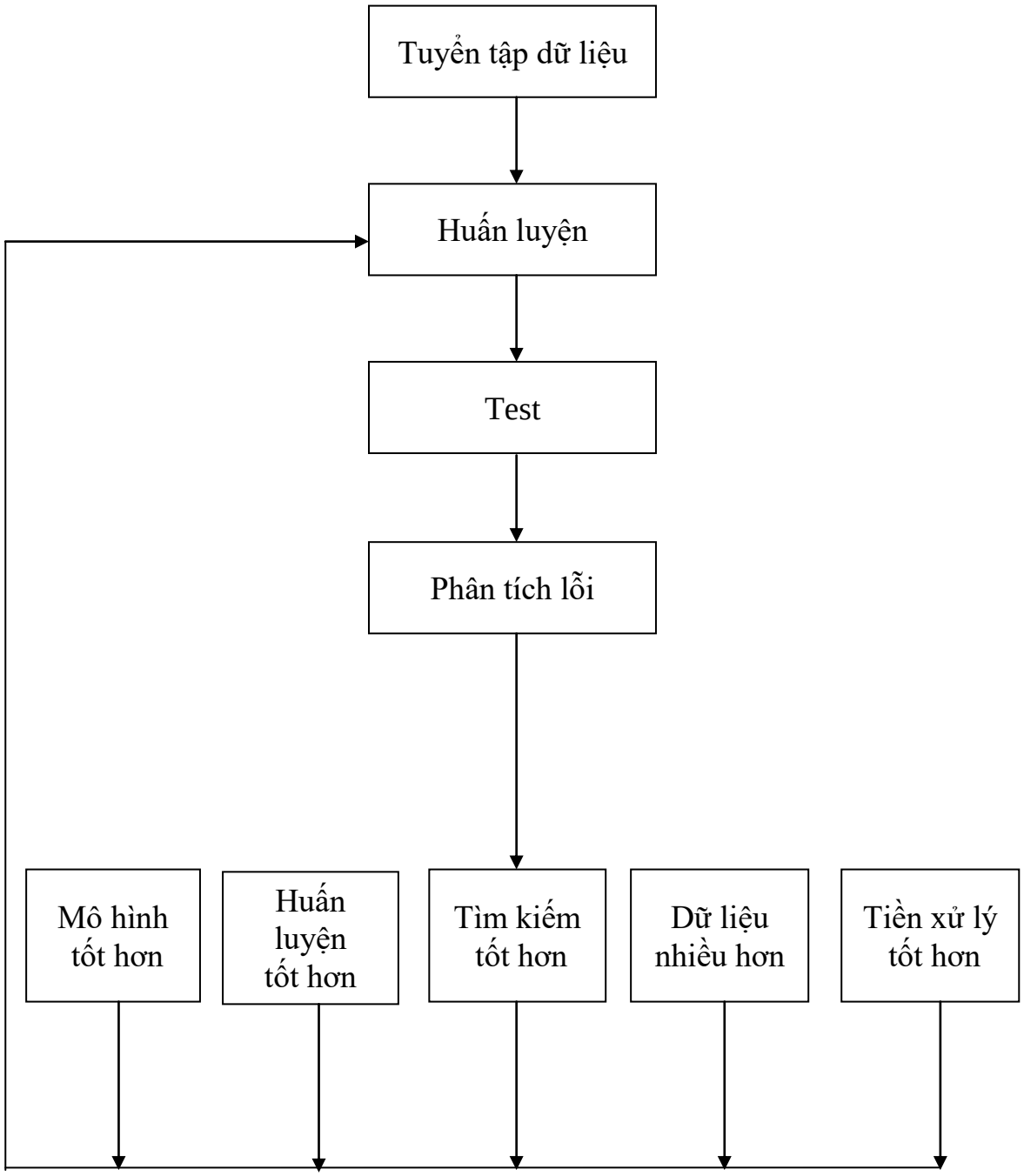
Để thực hiện dịch máy, chúng ta nhất thiết phải kết hợp nhiều nguồn trí thức. Trong dịch thống kê, chúng ta dựa vào toán học để thực hiện kết hợp tối ưu của các nguồn trí thức.

Trong dịch máy thống kê, trí thức dịch được học một cách tự động từ dữ liệu huấn luyện. Với kết quả như vậy, việc phát triển một hệ dịch dựa vào thống kê sẽ rất nhanh so với hệ dịch dựa vào luật.

Dịch máy thống kê khá phù hợp với ứng dụng nhúng mà ở đây dịch máy là một phần của ứng dụng lớn hơn.

Việc đưa ra khái niệm "chính xác" của mối quan hệ ngữ pháp, ngữ nghĩa, văn phong là rất khó khăn nếu không nói là không thể. Vì vậy, việc hình thức hóa vấn đề này càng chính xác càng tốt không thể dựa vào sự giằng buộc bởi các luật mô tả chúng. Thay vào đó, trong cách tiếp cận thống kê, các giả định mô hình được kiểm định bằng thực nghiệm dựa vào dữ liệu huấn luyện.

1.2.1.5. Chu kì phát triển của hệ thống dịch thống kê



Hình 1.2: Chu kì phát triển của hệ thống dịch thống kê

Bước đầu tiên là tập hợp dữ liệu huấn luyện. Ở đây, chúng ta cần thu thập các văn bản song ngữ, thực hiện việc dóng hàng câu và trích lọc ra các

cặp câu phù hợp. Trong bước thứ hai, chúng ta thực hiện huấn luyện tự động hệ thống dịch máy. Đầu ra của bước này là hệ thống dịch máy có hiệu lực.

Tiếp theo hệ thống dịch máy được kiểm tra và việc phân tích lỗi được thực hiện. Dựa vào kiến trúc của hệ thống dịch máy thống kê, chúng ta có thể phân biệt các kiểu lỗi khác nhau: lỗi tìm kiếm, lỗi mô hình, lỗi huấn luyện, lỗi corpus huấn luyện và lỗi tiền xử lý.

Mô hình tốt hơn: Ở đây, mục tiêu là phải phát triển mô hình mà mô hình này mô tả càng nhiều các thuộc tính của ngôn ngữ tự nhiên và các tham số tự do của nó có thể được ước lượng từ dữ liệu huấn luyện

Huấn luyện tốt hơn: Thuật toán huấn luyện thường dựa vào cách tiếp cận hợp lý cực đại. Thông thường, các thuật toán huấn luyện thường cho ta kết quả là tốt ưu địa phương. Do vậy, để làm tốt việc huấn luyện này, cần xây dựng các thuật toán mà kêt quả tối ưu địa phương thường gần với tối ưu toàn cục.

Tìm kiếm tốt hơn: Lỗi tìm kiếm xuất hiện nếu thuật toán tìm kiếm ra câu dịch của câu nguồn. Vấn đề tìm kiếm trong dịch máy thống kê là NP-hoàn thành. Vì vậy, chỉ có các cách tìm kiếm gần đúng để tìm ra câu dịch. Thuật toán hiệu quả là thuật toán mà cân bằng giữa chất lượng và thời gian.

Nhiều dữ liệu huấn luyện hơn: Chất lượng dịch càng tăng khi cỡ của corpus càng lớn. Quá trình học của hệ thống dịch máy sẽ cho biết cỡ của dữ liệu huấn luyện là bao nhiêu để thu được kết quả khả quan.

Tiền xử lý tốt hơn: Hiện tượng ngôn ngữ tự nhiên khác nhau là rất khó xử lý ngay cả trong cách tiếp cận thống kê tiên tiến. Do đó để cho việc sử dụng cách tiếp cận thống kê được tốt thì trong bước tiền xử lý, chúng ta làm

tốt một số việc như: loại bỏ các kí hiệu không phải là văn bản, đưa các từ về dạng gốc của nó, ...

Một đặc tính quan trọng của chu kì phát triển của hệ thống dịch máy thống kê là chúng ta có thể thay đổi hoàn toàn trong vài giờ hoặc vài ngày. Vì vậy, chu kì phát triển được thường xuyên thực hiện. Điều này cho phép cải tiến nhanh hệ thống dịch máy. Thêm vào đó, quá trình phân tích lỗi luôn luôn phụ thuộc vào việc thực hiện cuối cùng của hệ thống dịch máy. Vì vậy, việc quyết định sửa đổi hệ thống có thể trực tiếp dựa vào mục tiêu cuối cùng trong chất lượng của dịch máy.

1.3. Phân loại dịch máy thống kê

1.3.1. Dịch máy thống kê dựa vào từ (word-based)

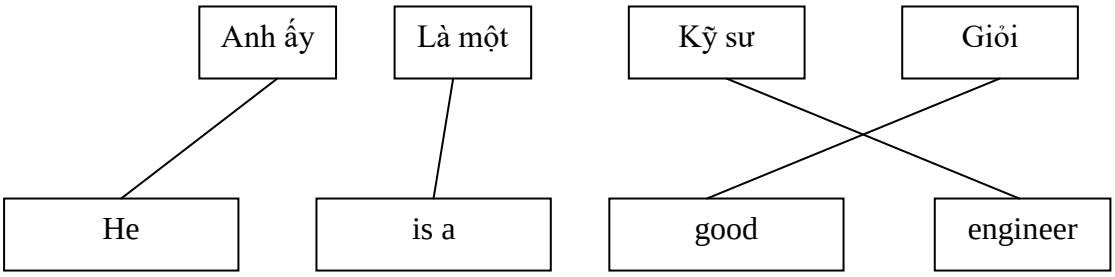
Trong dịch máy thống kê trên cơ sở từ, các đơn vị cơ bản của bản dịch là một từ trong ngôn ngữ tự nhiên.

Dịch máy thống kê trên cơ sở từ không sử dụng rộng rãi ngày nay, thay vào đó là dịch máy thống kê trên cơ sở cụm từ. Hầu hết các hệ thống dựa trên cụm từ sử dụng Giza++ để gióng hàng câu, trích rút ra các cặp câu song ngữ và mô hình ngôn ngữ. Vì những ưu thế của Giza++, hiện nay có một số nỗ lực đưa áp dụng tính toán phân tán trực tuyến cho phần mềm này.

1.3.2. Dịch máy thống kê dựa trên cụm từ (phrase-based)

Dịch máy thống kê trên cơ sở cụm từ có mục đích là để giảm bớt các hạn chế của dịch máy thống kê trên cơ sở từ bằng cách dịch cụm từ, trong đó độ dài cụm từ nguồn và cụm từ đích có thể khác nhau. Các cụm từ trong kỹ thuật này thường không theo nghĩa ngôn ngữ học mà là các cụm từ được tìm thấy bằng cách sử dụng phương pháp thống kê để trích rút từ các cặp câu.

Ví dụ: 1



1.3.3. Dịch máy thông kê dựa trên cú pháp

Dịch máy thống kê trên cơ sở cú pháp dựa trên ý tưởng của dịch các đơn vị cú pháp (phân tích cây của câu), hơn là những từ đơn hay cụm từ (như trong dịch máy thống kê trên cơ sở cụm từ). Ý tưởng này đã xuất hiện từ lâu, tuy nhiên phiên bản thống kê của ý tưởng này chỉ được hình thành khi có những bộ phân tích ngẫu nhiên mạnh mẽ trong những năm 1990.

1.3.4. Một số công cụ và các nhóm nghiên cứu trên Internet về SMT

Hiện có rất nhiều diễn đàn chia sẻ những tài nguyên, công cụ mã nguồn mở hỗ trợ cho hệ dịch máy thống kê. http://www.statmt.org là trang web tiêu biểu giới thiệu đầy đủ các tài liệu, các hội thảo liên quan đến SMT, parallel corpus, mã nguồn liên quan tới dịch máy thống kê được cập nhật một cách thường xuyên.

Các nhóm nghiên cứu về mở về SMT:

Nhóm nghiên cứu về Statistical MT ở trường Johns Hopkins đã dựng lên EGYPT3, một Open source Statistical MT Toolkit. Trong đó có GIZA, một training tool cho mô hình IBM 1-5, được sử dụng để tạo bảng ánh xạ từ-từ cho nhiều mô hình dịch theo phương pháp phrase-based.

Nhóm nghiên cứu về MT của ISI (Koehn, Och and Marcu) cũng sử dụng một Toolkit khác đó là SRILM4 để xây dựng hệ dịch máy nghiên cứu

theo phương pháp Phrase-based Statistical MT Pharaoh [5]. (Koehn cũng là một trong số những người tham gia phát triển hệ dịch Moses sau này).

Và gần đây nhất là sự xuất hiện của Moses [6], một hệ thống nguồn mở phrase-based SMT hoàn chỉnh. Moses thực chất là phiên bản cao hơn của Pharaoh, là phần mềm được nhiều trường đại học, nhóm nghiên cứu nổi tiếng về xử lý ngôn ngữ tự nhiên và dịch máy thống kê như Edinburg (Scotland), RWTH Aachen (Germany), ... tham gia phát triển. Đây là phần mềm có chất lượng khá tốt, khả năng mở rộng cao được dùng để xây dựng nhiều hệ thống dịch thử nghiệm cho nhiều cặp ngôn ngữ như Anh-Czech, Anh-Trung, Anh-Pháp, ... Hệ thống đã được sử dụng làm baseline trong cuộc thi về các hệ thống dịch máy

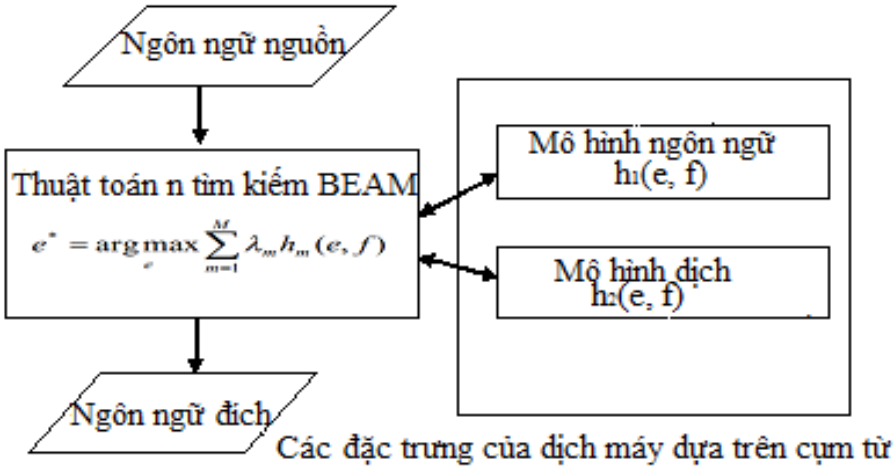
CHƯƠNG 2 – MÔ HÌNH DỊCH MÁY DỰA TRÊN CỤM TỪ VÀ ÁP DỤNG CHO NGÔN NGỮ VIỆT _ ANH

2.1. Giới thiệu mô hình dịch máy dựa trên cụm từ

Dịch máy thống kê: là phương pháp dịch máy trong đó các bản dịch được tạo ra trên cơ sở các mô hình thống kê

Cách tiếp cận thành công nhất trong hệ dịch máy là dịch dựa vào cụm từ, nghĩa là sử dụng cụm từ làm đơn vị nguyên tử. Các cụm từ trong phương pháp này không theo nghĩa của ngôn ngữ học mà là trình tự tiếp giáp của nhiều từ trong một câu. Trong phương pháp này, câu đầu vào của ngôn ngữ nguồn được chia thành một chuỗi các cụm từ, những cụm từ này được ánh xạ một – một để cho ra được các cụm từ của ngôn ngữ đích, thứ tự của các cụm từ trong ngôn ngữ đích có thể được sắp xếp lại. Thông thường các mô hình cụm từ được ước lượng từ các tập từ song song với sự liên kết của từ. Tất cả các cặp cụm từ phù hợp với sự liên kết của từ đều được trích xuất. Xác suất được đưa ra dựa trên số lượng tương đối hoặc xác suất dịch từ vựng.

2.2. Kiến trúc của mô hình dịch dựa trên cụm từ



Hình 2.1. Kiến trúc mô hình dịch dựa trên cụm từ

Từ ngôn ngữ nguồn (Tiếng Việt) dựa vào thuật toán tìm kiếm Beam (thuật toán này sẽ được trình bày ở phần sau) và dựa trên các đặc trưng của hệ dịch máy thống kê dựa trên cụm từ (mô hình ngôn ngữ, mô hình dịch, mô hình đảo cụm,…) để cho ra được ngôn ngữ đích (Tiếng Anh).

2.2.1 Mô hình log-linenear

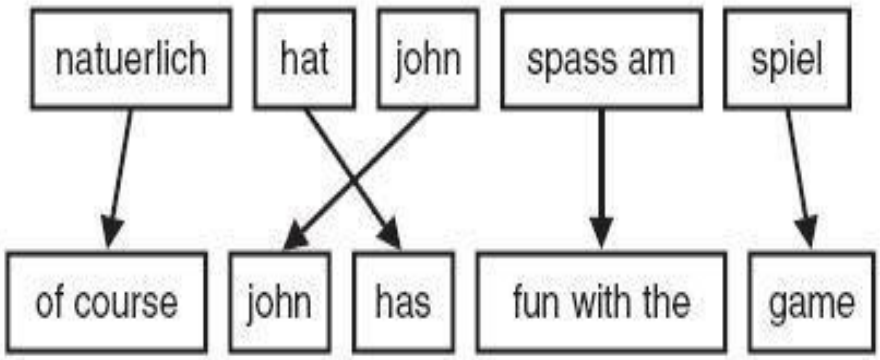
Đầu tiên, chúng ta đưa ra mô hình tiêu chuẩn cho hệ dịch thống kê dựa vào cụm từ. Có rất nhiều biến thể, những biến thể này được xem là sự mở rộng của mô hình tiêu chuẩn này.

2.2.1.1. Mục đích của mô hình dịch dựa trên cụm từ.

Hệ dịch thống kê dựa trên từ có khuyết điểm là không lấy được thông tin ngữ cảnh mà chỉ dựa trên các phân tích thống kê về từ. Mô hình dịch máy thống kê dựa trên cụm từ cải tiến hơn ở chỗ thay vì xử lý trên từ thì xử lý trên cụm từ. Điều này cho phép hệ thống có thể dịch các cụm từ tránh được dịch word-by-word. Vì đôi khi một từ trong ngôn ngữ tiếng Việt có nhiều hơn 1 nghĩa trong ngôn ngữ tiếng Anh.

Cùng xem xét ví dụ dưới đây:

Ví dụ 2:



Câu đầu vào là tiếng Đức được tách ra thành các cụm (với số lượng từ bất kỳ), sau đó mỗi cụm sẽ được dịch sang cụm từ tiếng Anh. Cuối cùng các cụm từ tiếng Anh được sắp xếp lại sao cho đúng với ngữ pháp tiếng Anh. Trong ví dụ trên, 6 từ tiếng Đức được ánh xạ sang 8 từ tiếng Anh và được chia thành 5 cặp cụm từ.

Những cụm từ tiếng Anh phải được sắp xếp lại để động từ luôn đứng sau chủ ngữ. Từ "natuerlich" trong tiếng Đức được dịch chính xác nhất sang tiếng anh là "of course". Để làm được điều này, chúng ta có một bảng dịch để ánh xạ các cụm từ chứ không phải ánh xạ các từ. Bảng có dạng như sau

Translation	Probability p(e|f)
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

Ta thấy xác xuất *p(e|f)* để dịch từ "natuerlich" từ tiếng Đức sang nghĩa "of course" trong tiếng Anh là cao nhất 0.5.

Điều quan trọng là ta phải chỉ ra được rằng, những mô hình cụm từ hiện tại đều không bắt nguồn từ bất kỳ khái niệm cụm từ nào trong khái niệm ngôn ngữ. Một trong số các cụm từ đó ở ví dụ trên là "fun with the". Đây là một nhóm bất thường. Hầu hết các lý thuyết cú pháp sẽ phân đoạn câu thành cụm danh từ "fun" và giới từ "with the game".

Tuy nhiên việc dịch “spass am” sang ‘fun with the” là rất hữu ích. Giới từ trong tiếng Đức và tiếng Anh thường không phù hợp với nhau. Nhưng dựa vào bối cảnh nên chúng được dịch như vậy. Từ “am” trong tiếng Đức có nhiều nghĩa trong tiếng Anh. Việc dịch nó sang nghĩa “with the” là bất thường vì nó thường mang nghĩa là “on the” hoặc “at the”, nhưng trong bối cảnh của từ “spass” nên “am” được dịch là “with the”.

Chúng ta thấy được 2 ưu điểm của việc dịch cùm từ thay vì từ. Một là, từ không phải là đơn vị nguyên tử tốt nhất trong việc dịch, do tần xuất ánh xạ một – nhiều (và ngược lại). Hai là, việc dịch một nhóm từ thay vì một từ giúp giải quyết được vấn đề nhập nhằng về nghĩa. Một ưu điểm thứ ba nữa là, nếu chúng có ngữ liệu huấn luyện lớn, chúng sẽ nhớ được những cụm từ hữu ích, đôi khi có thể ghi nhớ bản dịch của toàn bộ câu.

2.2.1.2. Định nghĩa toán học

Đầu tiên, chúng ta áp dụng quy tắc Bayes để chuyển đổi. Ta gọi ebest là kết quả dịch tốt nhất với một câu đầu f, ta định nghĩa như sau:

$$e_{best} = \text{argmax}_e\, p(e|f)$$

$$\text{argmax}_e\, p(f|e)\, p_{LM}(e) \qquad (2.1)$$

Đối với mô hình cụm từ, ta phân tích $p(f|e)$ ra thành:

$$p(f_1^{I} \mid e_1^{I}) = \prod_{i=1}^{I} \phi(\bar{f}_i \mid \bar{e}_i) d(start_i - end_{i-1} - 1) \qquad (2.2)$$

Câu đầu vào f được chia thành I và các cụm $\bar{f}_i$.

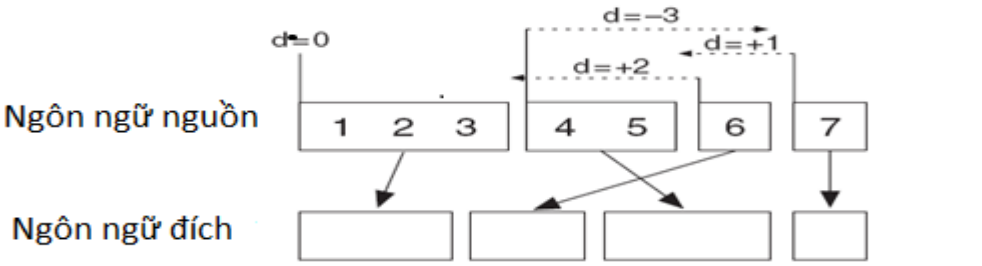
Lưu ý rằng, quá trình phân đoạn này không được mô hình hóa một cách rõ ràng. Điều này có nghĩa là mọi phân đoạn đều bằng nhau.

Mỗi cụm từ $\bar{f}_i$ trong ngôn ngữ nguồn được dịch sang cụm từ của ngôn ngữ đích là $\bar{e}_i$. Đảo cụm được điều khiển bởi “mô hình đảo cụm dựa trên

khoảng cách". Chúng ta xét việc đảo cụm liên quan đến cụm từ trước đó. Ta định nghĩa starti là vị trí của từ đầu tiên trong cụm từ nguồn, cụm mà được dịch sang cụm thứ i trong ngôn ngữ đích, và endi là vị trí kết thúc của từ trong cụm từ nguồn. Khoảng cách đảo cụm được tính bằng starti – endi-1 -1.

Khoảng cách đảo cụm là số từ bị bỏ qua (hoặc về phía trước hoặc phía sau) khi các từ trong ngôn ngữ nguồn không đúng thứ tự. Nếu 2 cụm từ được dịch theo đúng thứ tự thì starti = endi-1 + 1, ví trí của từ đầu tiên trong cụm thứ i cũng là vị trí của từ cuối cùng trong cụm trước đó. Trong trường hợp này, một chi phí đảo cụm d(0) được xác nhận. Hình dưới đây là một ví dụ:

Ví dụ 3:



Cụm	Thống kê	Sự dịch chuyển	Khoảng cách
1	1–3	Điểm bắt đầu	0
2	6	Bỏ qua 4 - 5	+2
3	4–5	Di chuyển quay lại 4 - 6	−3
4	7	Bỏ qua 6	+1

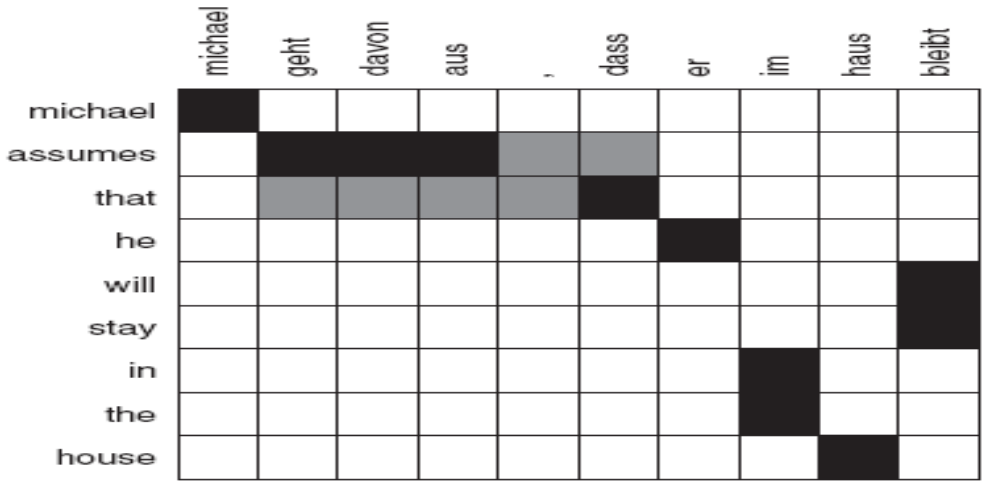
Đảo cụm dựa trên khoảng cách: Khoảng cách đảo cụm được đo từ phía đầu vào của ngôn ngữ nguồn. Trong hình minh họa ở trên, mỗi cụm từ nguồn được chú thích bằng mũi tên trỏ xuống cho thấy sự đảo cụm. Ví dụ cụm từ thứ 2 trong ngôn ngữ đích được dịch bởi từ thứ 6 trong ngôn ngữ nguồn, bỏ qua từ thứ 4 và 5, vậy khoảng cách sẽ là +2.

Xác suất của d được tính như nào? Thay vì ước lượng xác suất đảo cụm từ dữ liệu, chúng ta áp dụng cấp số nhân phân rã hàm chi phí $d(x) = \alpha^{|x|}$ với một giá trị thích hợp của tham số $\alpha \in [0,1]$ để d là một phân bố xác suất hợp lý. Công thức này có nghĩa là, sự dịch chuyển của các cụm từ trên một khoảng cách lớn thì chi phí cao hơn là dịch chuyển ngắn hoặc không dịch chuyển.

Lưu ý rằng, mô hình đảo cụm này tương tự như mô hình đảo từ trong dịch máy thống kê dựa trên cơ sở từ. Chúng thậm chí có thể được huấn luyện xác suất đảo dựa trên dữ liệu, nhưng điều này thường không được thực hiện trong mô hình cơ sở là cụm từ.

2.2.2. Mô hình dịch

Chất lượng của bản dịch trong dịch thống kê dựa trên cụm từ phụ thuộc nhiều vào chất lượng của bảng dịch cụm từ (phrase table). Để xây dựng bảng dịch cụm từ đầu tiên, chúng ta tạo ra gióng hàng từ giữa mỗi cặp câu trong ngữ liệu song ngữ, sau đó trích xuất các cặp cụm từ phù hợp với gióng hàng từ.



Hình 2.2: Ví dụ về mô hình dóng hàng

Với sự gióng hàng từ trong hình 2.2, chúng ta muốn trích xuất ra các cụm từ phù hợp, ví dụ như cụm từ "assumes" trong tiếng Anh với cụm từ "geht davon aus" trong tiếng Đức.

Nếu ta phải dịch một câu tiếng Đức có chứa cụm từ "geht davon aus,dass", thì chúng ta có thể sử dụng cụm từ đã được gióng và được dịch là "assumes that". Các cụm từ hữu ích cho việc dịch có thể dài hoặc ngắn hơn cụm từ trong ví dụ này. Những cụm từ ngắn hơn xảy ra thường xuyên hơn, do đó chúng có khả năng ứng dụng nhiều hơn cho những câu chưa được gặp. Những cụm từ dài thường nắm bắt các ngữ cảnh giúp chúng ta có thể dịch được lượng ký tự lớn hơn cùng một lúc, thậm chí là toàn bộ câu.

Do đó, khi trích xuất các cặp cụm từ, chúng ta phải chọn cả những cụm từ ngắn và cụm từ dài, vì tất cả đều hữu ích. Các cặp cụm từ này được lưu giữ lại trong bảng cụm từ cùng với xác suất $\phi(\bar{f}_i \mid \bar{e}_i)$.

$$\phi(\bar{f}_i \mid \bar{e}_i) = \frac{count(\bar{f} \mid \bar{e})}{\sum_{\bar{f}} count(\bar{f} \mid \bar{e})}$$

Trong đó

Sau quá trình xây dựng mô hình ngôn ngữ ta đi huấn luyện mô hình dịch (Train Model), quá trình này sẽ tạo ra bảng cụm từ (phrase table).

Để tạo ra được bảng cụm từ ta sử dụng script train-model.perl trong phần mềm Moses, các giai đoạn của thủ tục huấn luyện (gióng hàng từ, gióng hàng cụm từ, học mô hình dịch) được gọi trong chương trình, ví dụ:

/path-to-mosesdecoder/scripts/training/train-model.perl -bin-dir bin -external-bin-dir bin -scripts-root-dir bin -root-dir . -corpus corpus -f en -e vn -alignment grow-diag-final-and -reordering msd-bidirectional-fe -lm 0:3:corpus.lm.vn:0

(trong đó corpus -f en -e vn là 2 tệp tin ngữ liệu đầu vào sau tiền xử lý corpus.lm.vn là mô hình ngôn ngữ được huấn luyện ở bước trên)

Một số phần tử trong bảng dịch cụm sau khi được huấn luyện:

” ... họ là một ||| `` ... they ' re a ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” ... họ là ||| `` ... they ' re ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” ... họ ||| `` ... they ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” ... ||| `` ... ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” 006 lịch _ sự nói ||| `` 006 said politely ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” 006 lịch _ ||| `` 006 ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” 006 lịch ||| `` 006 ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” 006 ||| `` 006 ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” ? ||| `` ? ||| 0.428571 0.142857 0.428571 0.714286 0.142857 0.142857

” anh _ ấy giải _ thích ||| `` he explained . ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” anh _ ấy giải ||| `` he explained ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” anh _ ấy nói ||| , `` he said . ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” anh _ ấy nói ||| , `` he said ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” anh _ ấy nói ||| `` he said . ||| 0.818182 0.090909 0.090909 0.818182 0.090909 0.090909

” anh _ ấy nói ||| `` he said ||| 0.428571 0.142857 0.428571 0.714286 0.142857 0.142857

” anh _ ấy phân _ vân ||| `` he wondered . ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” anh _ ấy ||| , `` he ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” anh _ ấy ||| `` he ||| 0.636364 0.090909 0.272727 0.818182 0.090909 0.090909

” ann nói ||| `` ann said . ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” ann nói ||| `` ann said ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” ann ||| `` ann ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” bằng cách đoán mặt _ ||| by guessing ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” bằng cách đoán mặt ||| by guessing ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” cho tối _ nay ? ||| `` for tonight ? ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” cho tối _ nay ||| `` for tonight ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” cho ||| `` for ||| 0.600000 0.200000 0.200000 0.600000 0.200000 0.200000

” chính _ ||| `` ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” chính ||| `` ||| 0.600000 0.200000 0.200000 0.200000 0.200000 0.600000

” có _ nghĩa là ||| , in ||| 0.200000 0.200000 0.600000 0.200000 0.200000 0.600000

2.2.3. Mô hình ngôn ngữ

Mô hình ngôn ngữ là quá trình tính xác suất của ngôn ngữ đích nhằm tính toán ra chuỗi câu đích phù hợp nhất (có xác suất xuất hiện cao nhất). Không chỉ có ích cho thứ tự các từ mà còn có ích cho việc chọn nghĩa giữa các cách dịch khác nhau.

Ví dụ: Cho 2 câu

(A) .Tôi tìm thấy sự giàu có trong sân sau nhà tôi

(B) .Tôi tìm thấy sự giàu có ở sân sau của tôi

Quyết định này dịch từ tiếng Việt sang tiếng Anh, cả hai từ “trong” hoặc “ở” đều tương ứng với từ “in”. Nếu trong corpus của chúng ta, giả sử trigram “trong sân sau nhà tôi” xuất hiện 10 lần, trong khi “ở sân sau nhà tôi” không xuất hiện (hoặc khá nhỏ so với “trong sân sau nhà tôi”) thì (A) là câu tốt hơn (được chọn). Điều đó có nghĩa là ta có thể giả quyết vấn đề nhập nhằng ngữ nghĩa chỉ dựa vào ngôn ngữ đích.

Trước khi xây dựng mô hình ngôn ngữ (language model), ngữ liệu đầu vào của quá trình này là tệp tin đơn ngữ của ngôn ngữ đích - tiếng Anh. Ngữ liệu này cần được tiền xử lý (prepare data): phân tích từ tố, tắt chữ hoa ở đầu câu, và tách từ cho file tiếng Việt. Để làm việc này ta sử dụng 2 script: *tokenizer.perl* và *lowercase.perl.*

Sau khi ngữ liệu được tiền xử lý, ta đi xây dựng mô hình ngôn ngữ (Build Language Model). Ta sử dụng script *ngram-count* trong SRILM để xây dựng mô hình ngôn ngữ, mô hình ngôn ngữ chỉ được xây dựng trên ngôn ngữ đích (trong ví dụ này ta xây dựng từ tệp *corpus.vn*) ví dụ:

/path-to-srilm/bin/i686/ngram-count -order 3 interpolate -kndiscount -text corpus.vn -lm corpus.lm.vn (trong đó corpus.vn là tệp ngữ liệu đầu vào của ngôn ngữ đích sau khi tiền xử lý, kết quả của quá trình này được lưu lại vào file corpus.lm.vn)

Ví dụ : Khi xây dựng mô hình ngôn ngữ (language model), ngữ liệu đầu vào của quá trình này là tệp tin đơn ngữ của ngôn ngữ đích - tiếng Anh. tệp tin corpus.vn, có nội dung như sau:

corpus.en

Tell me your argument

It 's not good for business recently

Not good for business ?

I do n't see eye to eye with you

The new policy offers new opportunities

Too risky anyway

It 's necessary to look before you leap

Japanese is easy to learn

You are mistaken about it

As a matter of fact , it 's very hard

There are so many familiar characters and they sound just like Chinese

Ngữ liệu này cần được tiền xử lý (prepare data): phân tích từ tố, tắt chữ hoa ở đầu câu, và tách từ cho file tiếng Anh. Để làm việc này ta sử dụng 2 script: tokenizer.perl và lowercase.perl. Tệp tin kết quả có nội dung như sau:

corpus.vn

tell me your argument

it ' s not good for business recently

not good for business ?

i do n 't see eye to eye with you

the new policy offers new opportunities

too risky anyway

it ' s necessary to look before you leap

Japanese is easy to learn

you are mistaken about it

as a matter of fact , it ' s very hard

there are so many familiar characters and they sound just like Chinese

Sau khi ngữ liệu được tiền xử lý, ta đi xây dựng mô hình ngôn ngữ (Build Language Model)

Các mô hình ngôn ngữ chuẩn, không mất mát được xây dựng sử dụng SRI Language Modelling Toolkit (SRILM). SRILM là một dự án mã nguồn mở bao gồm nhiều chương trình, thư viện C++ và script hỗ trợ trong việc xây dựng và thử nghiệm các mô hình ngôn ngữ cho nhận dạng tiếng nói hoặc các ứng dụng khác. Nó hỗ trợ nhiều kiểu mô hình ngôn ngữ khác nhau dựa trên thống kê về n-gram. SRILM đã được phát triển từ năm 1995 ở Phòng nghiên cứu công nghệ tiếng nói SRI, và vẫn còn đang được tiếp tục sửa chữa, mở rộng bởi nhiều nhà nghiên cứu trong cộng đồng NLP.

Ta sử dụng script ngram-count trong SRILM để xây dựng mô hình ngôn ngữ, mô hình ngôn ngữ chỉ được xây dựng trên ngôn ngữ đích (trong ví dụ này ta xây dựng từ tệp corpus.vn) ví dụ:

/path-to-srilm/bin/i686/ngram-count -order 3 -interpolate -kndiscount -text corpus.vn -lm corpus.lm.vn

(trong đó corpus.vn là tệp ngữ liệu đầu vào của ngôn ngữ đích sau khi tiền xử lý, kết quả của quá trình này được lưu lại vào file corpus.lm.vn)

file corpus.lm.vn có cấu trúc như sau

\data\

ngram 1=22087

ngram 2=211751

ngram 3=56100

\1-grams:

4.035516 worried -0.4450608

-4.648153 worries -0.172538

-4.035516 worry -0.3102582

-4.902032 worrying -0.1273299

-3.913626 worse -0.4014535

-5.133546 worsened -0.1273299

-5.00332 worsening -0.1273299

-5.133546 worsens -0.1273299

-5.00332 worship -0.1273299

-5.00332 worst -0.1273299

-3.804553 worth -0.2521719

-5.00332 worthless -0.2506225

-5.00332 worthwhile -0.2506225

-4.648153 worthy -0.3038294

\2-grams:

-3.463238 ! nam

-3.464096 ! oh

-3.449447 ! quickly

-3.459536 ! role

-3.435275 ! she

-3.21163 ! that

-2.906905 ! there

-2.910975 ! they

-3.410269 ! we

-3.429152 ! well

\3-grams:

-0.03962466 fresh ! </s>

-0.03962466 fun ! </s>

-0.03962466 furious ! </s>

-0.05768845 girl ! </s>

-0.02839333 girlfriend ! </s>

-0.6019399 go ! '

-0.07626353 god ! </s>

…..\end

2.3. Giải mã

2.3.1. Đặt vấn đề

Thuật toán giải mã là vấn đề quyết định trong dịch thống kê. Sự thực hiện của chúng trực tiếp ảnh hưởng tới chất lượng và tính hiệu quả. Với một thuật toán giải mã không đáng tin cậy và hiệu quả, hệ thống dịch thống kê có thể bỏ qua câu dịch tốt nhất ngôn ngữ đích của câu nguồn mặc dù nó được dự đoán đầy đủ bằng mô hình mô tả nó. Có hai vấn đề quan trọng trong giải mã:

- Tối ưu: Có thể thuật toán giải mã tìm ra cách dịch tối ưu như được dự đoán bằng mô hình không?

- Tốc độ: Có thể thuật toán giải mã tìm ra cách dịch tối ưu với một thời gian hiệu quả không?

Chúng ta thường phải có một số biện pháp để cân bằng giữu 2 vấn đề trên. Nếu chúng ta tập trung toàn bộ vào không gian giả thuyết, chúng ta có thể chắc chắn tìm được giả thuyết tối ưu mà được dự đoán bằng mô hình. Tuy nhiên, khi không gian giả thuyết là hàm mũ thì việc liệt kê tất cả các giả thuyết là phi thực tế. Thông thường chúng ta giảm bớt các giả thuyết để tăng tốc độ giải mã và điều này có thể hy sinh vấn đề tối ưu.

Do đó việc nghiên cứu chỉ tập trung vào tìm các thuật toán gần đúng. Có rất nhiều thuật toán như vậy để giải quyết vấn đề này, một trong số các thuật toán đó là thuật toán “Giải mã ngăn xếp nhanh”.

2.3.2. Mô tả thuật toán

Phần còn lại của một hệ dịch máy thống kế là chức năng tìm kiếm câu đích (giải mã). Chức năng của một bộ giải mã là từ câu nguồn E sẽ tìm câu cần dịch V sao cho tích của hai xác suất mô hình dịch và mô hình ngôn ngữ là lớn nhất:

$$V = \arg\max_{v} p(v^J \mid e^I) = \arg\max_{v} p(v^J) \times p(e^I \mid v^J)$$

Đây chính là một bài toán tìm kiếm, quá trình giải mã chỉ là một dạng của bài toán này. Tìm kiếm trong dịch máy dựa theo phương pháp tìm tiếm theo lựa chọn tốt nhất (best-first search), một dạng của tìm kiếm theo kinh nghiệm (heuristic) hay tìm kiếm có thông tin (informed search); các thuật toán này tìm kiếm dựa trên các hiểu biết trong phạm vi của bài toán. Thuật toán tìm kiếm theo lựa chọn tốt nhất sẽ lựa chọn ra một nút n dựa theo một hàm ước lượng là f(n). Chức năng tìm kiếm trong hệ dịch máy thường sử dụng thuật toán A* cũng là một phương pháp tìm kiếm theo chiến lược tìm kiếm theo lựa chọn tốt nhất. A* được đưa vào cho các hệ dịch máy từ năm 1995 bởi IBM, trước đấy thì nó được sử dụng trong các bài toán nhận dạng tiếng nói.

Trong thuật toán A* các trạng thái mà nó đang lưu trữ để tìm kiếm được gọi là stack decoding, một cấu trúc dữ liệu đơn giản cho stack decoding là sử dụng một hàng đợi ưu tiên lưu trữ các giả thuyết dịch cùng với điểm đánh giá của nó.

Trong bài viết này, chúng ta sẽ trình bày về phương pháp dịch được sử dụng cho hệ dịch Pharaoh.

Do chịu sự giới hạn về không gian được sử dụng cho tìm kiếm, chúng ta không cần thiết phải tìm kiếm qua tất cả các câu tiếng anh nguồn, mà chỉ cần quan tâm đến những câu có thể sinh ra được từ câu Tiếng Anh cần dịch E.

Nhằm giảm bớt không gian tìm kiếm, chúng ta sẽ chỉ quan tâm đến các từ hoặc cụm từ mà chúng có thể được dịch ra được từ các từ trong câu E. Công việc này được thực hiện dựa trên việc tìm kiếm trên bảng dịch cụm từ.

function STACK DECODING (source sentence) **return**s target sentence

Initialize *stack* with a null hypothesis

loop do

pop best hypothesis *h* off *stack*

if *h* is a complete sentence **return** *h*

for each *possible* expansion *h'* of *h*

assign a score to *h'*

push *h*' onto stack

Hình 2.3: Thuật toán giải mã A cho dịch máy*

Quá trình tìm kiếm được mô tả như sau. Ban đầu trạng thái tìm kiếm của ta là rỗng. Tiếp theo ta mở rộng nút này bằng cách trên mỗi nút bằng cách từ các từ trong câu tiếng Anh ta tìm các từ tiếng anh có thể dịch ra được từ các từ đấy. Tiếp theo ta chọn nút có đánh giá tối ưu nhất để tiếp tục mở rộng nút này. Quá trình này tiếp tục đến khi nào tìm được câu dịch thỏa mãn.

Đánh giá tại mỗi nút sẽ giựa theo hai giá trị là giá trị hiện tại và giá trị tương lai. Giá trị hiện tại là tổng xác xuất của các cụm từ đã được dịch trong trong câu giả thiết nó là tích xác suất của mô hình dịch, thay đổi vị trí và mô hình ngôn ngữ.

$$\text{cost}(V \mid E) = \prod_{i \in S} \phi(\bar{v}_i \bar{e}_i) \times d(a_i - b_{i-1}) \times p(V) \qquad (2.3)$$

Giá trị tương lai là đánh giá chi phí về các từ còn lại trong câu Tiếng Anh chưa được dịch khi dịch sang câu Tiếng Việt. Khi kết hợp hai đánh giá này lại ta tìm được một đường đi tối ưu để dịch ra được câu Tiếng Việt. Một phương pháp tìm đường đi trên không gian tìm kiếm để tìm được câu dịch đó là sử dụng thuật toán Viterbi.

Để giảm bớt không gian tìm kiếm của bài toán ta có thể sự dụng thuật toán beam-search pruning. Sau mỗi bước mở rộng, thì chỉ lưu lại n trạng thái có đánh giá tốt nhất. Đây là kĩ thuật được sử dụng trong hệ dịch của Pharaoh.

```
function BEAM SEARCH STACK DECODER (source sentence) returns target sentence
initial hypothesisStack[0..nf]
push initial null hypothesis on hypothesStack[0]

for i    0 to nf-1
      for each hyp hypothesisStack[i]
            for each new_hyp that can be derived from hyp

                  nf[new_hyp]    number of foreign words covered by new_hyp
```

Hình 2.4: Giải thuật tìm kiếm beam sử dụng đa ngăn xếp trong Pharaoh

Trong dịch máy thống kê thì việc đánh giá kết quả khi lựa chọn số từ để dịch là không dễ dàng, vì vậy thay vì dùng một stack ta sẽ dùng m stack trong đấy stack Sm là chứa các giả thiết có thể dịch ra được từ m từ trong câu cần dịch.

2.4. Đánh giá chất lượng dịch

Đánh giá chất lượng các hệ thống dịch có thể được thực hiện thủ công bởi con người hoặc tự động. Quá trình đánh giá thủ công cho điểm cho các câu dịch dựa trên sự trôi chảy và chính xác của chúng. Phần lớn mọi người cho rằng đây là phương pháp đánh giá chính xác nhất. Thế nhưng công việc đánh giá thủ công này lại tiêu tốn quá nhiều thời gian, đặc biệt khi cần so sánh nhiều mô hình ngôn ngữ, nhiều hệ thống khác nhau. Mỗi phương pháp đánh giá đều có ưu nhược điểm riêng. Tuy đánh giá tự động không thể phản ánh được hết mọi khía cạnh của chất lượng dịch, nhưng nó có thể nhanh chóng cho ta biêt: chất lượng của hệ dịch ở tầm nào, có tăng lên hay không sau khi cải tiến hoặc thay đổi một tham số nào đó. Trong thực tế, hai phương pháp này vẫn được sử dụng đồng thời, và điểm BLEU là độ đo chất lượng hệ dịch phổ biến nhất hiện nay, được đề xuất bởi Papineni (Papineni, et al., 2002)

Chỉ số BLEU: Đây là chỉ số đánh giá chất lượng dịch của máy dịch thống kê từ ngôn ngữ này sang ngôn ngữ khác.

Kết quả dịch máy thống kê càng chính xác thì chỉ số BLEU càng cao và ngược lại. Điểm chỉ số BLEU được tính dựa vào việc so sánh câu dịch được với một tập hợp các câu dịch tốt, sau đó lấy giá trị trung bình từ những câu này. tính điểm bằng cách đối chiếu kết quả dịch với tài liệu dịch tham khảo và tài liệu nguồn. Mặc dù Callison-Burch [6] chỉ ra rằng điểm BLEU thường không thực sự tương quan với đánh giá thủ công của con người với các loại hệ thống khác nhau, thế nhưng vẫn có thể khá chính xác để đánh giá trên cùng một hệ thống, hoặc những hệ thống tương tự nhau. Chính vì vậy, trong khóa luận này, điểm BLEU được sử dụng làm thước đo chất lượng dịch, từ đó so sánh các loại mô hình dịch tên riêng khác nhau.

NIST (Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics) là một phương pháp đánh giá chuất lượng của câu dịch. NIST được đưa ra bởi US Nation Institute of Standard and Technology. NIST được dựa trên BLEU, tuy nhiên có thêm một vài cải tiến. Mặc dù vậy, NIST vẫn có những điểm khác so với BLEU. Cùng với BLEU, NIST là một trong những phương pháp đánh giá chất lượng dịch hiện nay.

2.5. Phần mềm mã nguồn mở Moses

Dịch máy thống kê dựa trên cụm từ là một trong những bước tiến trong trong dịch máy thống kê và hiện nay có thể được coi là cho chất lượng dịch tốt nhất. Mô hình dịch máy dựa trên cụm từ được phát triển từ mô hình dịch máy dựa trên từ. Trong mô hình dịch máy dựa trên cụm từ, các đoạn từ liên tục của từ trong câu nguồn được ánh xạ vào các đoạn từ liên tục tương ứng trong câu đích. Trong dịch máy thống kê, chúng ta có 1 câu nguồn s và câu dịch t tương ứng. Mục đích của dịch máy là tìm ra câu :

$$\hat{t} = \arg\max_t p(t \mid s)$$

Trong đó p(t|s) là xác suất mô hình. arg max tìm giá trị trong không gian có thể của các câu dịch t. Đã có rất nhiều các cài đặt của bộ giải mã dịch máy thống kê dựa trên cụm trước khi hệ thống Moses ra đời. Các hệ thống sớm có như Thư viện dóng hàng (ATS) và Pharaoh[5] (Koehn, 2004) được sử dụng trong nghiên cứu rất rộng rãi. ATS có lẽ là một hệ thống đan xen khi các lớp từ được dịch như các phrase nhưng thực tế thì là dịch từ sang từ. Pharaoh thay thế các lớp từ bằng các các lớp từ ngoài (surface word) do vậy loại bỏ được các lớp từ trong giải mã. Cũng có nhiều các hệ giải mã cho mô hình dịch dựa trên cụm từ khác đã sử dụng transducer trọng số hữu hạn trạng thái nhưng lại gặp vấn đề lớn trong việc mô hình việc sắp xếp lại. Mặt khác, có rất nhiều

các hệ thống này lại đi kèm với bảnquyền sử dụng. ATS chưa hề cho ra bản mã nguồn mở, còn với Pharaoh được đưa ra năm 2003 thì cũng là một bản dịch nhị phân cùng tài liệu. Những hạn chế đã được khắc phục phần nào trong hệ thống Moses. Mục đích chính của hệ thống Moses là tạo ra bộ giải mã mở rộng, có thời gian chạy chấp nhận được so với các hệ thống tương tự, dễ sử dụng và là sự lựa chọn tốt hơn cho các nhà nghiên cứu trong lĩnh vực bộ giải mã của dịch máy thống kê.

Moses là hệ dịch máy thống kê cho phép người dùng dễ dàng tạo ra mô hình dịch cho bất cứ một cặp ngôn ngữ nào. Moses cung cấp cả hai loại mô hình dịch là dựa trên cụm và dựa trên cây. Bộ công cụ Moses (Moses Toolkit) là một bộ nghiên cứu dịch học thuật đầy đủ. Nó bao gồm đầy đủ các thành phần để tiền xử lý dữ liệu, huấn luyện mô hình ngôn ngữ và mô hình dịch. Nó cũng bao gồm các công cụ tuning cho các mô hình này sử dụng huấn luyện với lỗi tối thiểu và đánh giá kết quả dịch sử dụng điểm BLEU. Moses sử dụng các chuẩn công cụ ngoài cho một số công việc để tránh sự trùng lặp, như GIZA++[4](Och, et al., 2003) cho gióng hàng từ và SRILM cho mô hình hóa ngôn ngữ.

Bộ giải mã Moses có khả năng chấp nhận các câu đầu vào đơn giản, các mạng rắc rối hoặc đan xen, tương tự như các bộ giải mã dịch máy thống kê khác như MITLL/AFRL hoặc hệ thống ITC-irst. Bộ giải mã đồng thời cùng tạo ra các loại output khác nhau, phân loại từ 1-best tới n-best trong danh sách và từ điển từ.

Moses sử dụng các chuẩn công cụ ngoài với một số công việc để tránh sự trùng lặp, như GIZA++ cho gióng hàng từ và SRILM cho mô hình hóa ngôn ngữ. Đồng thời, bởi các tác vụ thường tốn CPU, bộ công cụ được thiết

kế để làm việc với Sun Grid Engine trong môi trường song song để tăng hiệu quả đầu ra.

Để thống nhất các giai đoạn thử nghiệm, một tiện ích được phát triển để chạy các thử nghiệm lặp lại. Giải pháp này sử dụng các công cụ bao gồm trong Moses và yêu cầu các thay đổi tối thiểu để cài cài đặt.

Bộ công cụ được lưu trữ và phát triển trên sourceforge.net từ khi tạo ra. Moses có một cộng đồng nghiên cứu đang hoạt động .

(Tải về tại: http://sourceforge.net/projects/mosesdecoder/)

Nơi có thể tìm được rất nhiều thông tin về project này. Moses là chủ đề của năm của Johns Hopkin University Workshop về dịch máy [6] (Koehn ,2006). Bộ giải mã là thành phần quan trọng nhất trong Moses. Để làm giảm thiểu các sai sót khi học cho các nhà nghiên cứu, bộ giải mã được phát triển thay thế cho Pharaoh – một bộ giải mã dựa trên cụm rất thông dụng.

Nhằm làm cho bộ công cụ dễ dàng phù hợp với cộng đồng, và việc đóng góp thêm vào dự án này dễ dàng, nhóm phát triển đã giữ các nguyên tắc sau khi phát triển bộ giải mã:

Khả năng tiếp cận

Dễ dàng duy trì

Tính linh hoạt

Dễ dàng cho việc phát triển theo nhóm

Khả chuyển

Nó được phát triển trên nền C++ vì tính hiệu quả, thiết kế theo module và hướng đối tượng.

2.6. Quá trình giải mã

2.6.1. Huấn luyện cực tiểu sai số (MERT)

Mô hình dịch có một số mô hình thành phần (mô hình ngôn ngữ, mô hình đảo từ, các phương pháp tính điểm cụm từ khác nhau, phạt từ). Việc xác định trọng số cho các mô hình thành phần này khó thực hiện bằng tay (thử và sai), trong khi nó lại rất quan trọng với việc tối ưu chất lượng dịch. Quá trình này sẽ tìm ra giá trị tối ưu của các trọng số này. Các công việc trong quá trình này có thể là:

- Lọc bảng cụm từ

Corpus được sử dụng cho MERT thường là nhỏ, trong khi bảng cụm có thể rất lớn. Trong quá trình huấn luyện, văn bản nguồn sẽ được dịch nhiều lần. Để giảm thời gian dịch, bảng cụm từ có thể được lọc dựa vào văn bản nguồn. Việc lọc này được thực hiện bởi chương trình Perl run-filtered-pharaoh.perl.

- Sinh n-tốt nhất (n-best)

Sinh n-tốt nhất tức là bộ decoder đưa ra n câu dịch tốt nhất thay vì chỉ sinh một câu duy nhất.

Ví dụ:

Câu tiếng Việt:

hệ _ thống thông _ tin kế _ toán .

1-best:

the periodic inventory system ,the accounting information ...

Thực hiện MERT cần các câu dịch có thể có của một câu nguồn. Vì số câu như vậy là rất lớn, danh sách n-tốt nhất được sử dụng như một xấp xỉ của không gian các câu dịch.

Ví du:

mert-moses.pl corpus/tuning/input corpus/tuning/reference bin/moses model/moses.ini --working-dir tuning/ --rootdir scripts/

Câu lệnh này sẽ tạo ra file moses.ini có chứa các tham số tối ưu sau quá trình huấn luyện

Ví dụ về một số tham số của file moses.ini chưa được huấn luyện

distortion (reordering) weight

[weight-d]

0.3

0.3

0.3

0.3

0.3

0.3

0.3

language model weights

[weight-l]

0.5000

translation model weights

[weight-t]

0.20

0.20

0.20

0.20

0.20

Ví dụ về một số tham số của file moses.ini đã được huấn luyện

distortion (reordering) weight

[weight-d]

0.021687

0.139768

0.0502652

0.0364734

0.0326558

0.0561608

0.0750856

language model weights

[weight-l]

0.0663156

translation model weights

[weight-t]

0.0437612

0.0218868

0.0477119

0.384068

Các tham số trong # translation model weights ban đầu đều có giá trị là 0.2, nhưng sau khi huấn luyện đã được thay đổi tối ưu với các giá trị khác nhau.

2.7. Áp dụng với cặp ngôn ngữ Việt – Anh

2.7.1.Xây dựng ngữ liệu (corpus)

Trong xử lý ngôn ngữ tự nhiên bằng thống kê, corpus là tài nguyên không thể thiếu. Có nhiều loại corpus khác nhau, tùy thuộc vào bài toán và phương pháp giải quyết mà yêu cầu loại corpus thích hợp.

Để phát triển hệ thống dịch máy thống kê, chúng ta cần có dữ liệu để huấn luyện (học). Dữ liệu huấn luyện càng lớn thì càng tốt, nên được trích lọc ra từ cùng một lĩnh vực dịch mà hệ thống dịch máy được sử dụng. Dữ liệu sử dụng trong dịch máy là dữ liệu thô và song ngữ.

Bộ dữ liệu huấn luyện nếu thực hiện bằng thủ công thì mất rất nhiều công sức (chi phi đắt). Trong phần này trình bày về corpus và phương pháp xây dựng corpus một cách tự động.

2.7.1.1. Tạo corpus thô

Ở đây chúng ta chỉ cần tạo Corpus thô tiếng Việt, còn Corpus tiếng Anh chúng ta sử dụng Corpus trong Penn Tree Bank.

Download file HTML: có nhiều chương trình download file siêu văn bản từ Internet. Trong đó chúng tôi thấy tốt nhất là TeleportPro. Chương trình này có thể download cả một Website về ổ cứng.

- Lấy text: nếu không muốn viết bộ phân tích file HTML (HTML parser), ta sử dụng COMPONENT đọc file HTML của Microsoft (mshtml).

- Chuẩn hoá: công việc chuẩn hoá bao gồm:

+ Chuyển mã tiếng Việt (nếu cần)

+ Lọc bỏ các file chứa text xấu (trang quảng cáo, tìm việc, v.v.) bằng heuristics.

+ Lọc bỏ các text xấu trong mỗi file (tiêu đề, quảng cáo, v.v.) bằng heuristics.

+ Chuẩn hoá về bỏ dấu thanh (hòa --> hoà, v.v.)

Đánh dấu văn bản: Chúng tôi chỉ đơn giản thực hiện đánh dấu câu và từ. Sau khi cắt câu và phân đoạn từ, câu kết quả được lưu ra file sử dụng các nhãn đánh dấu câu (<S></S>) và từ (#).

Ví dụ:

<S>Phần mềm#máy tính#tự#khắc phục#sự cố#của#IBM</S>

<S>Tập đoàn#IBM#cho biết#sẽ#tung ra#thị trường#các#phiên bản#mới#của#hai#phần mềm#dựa trên#công nghệ#điện toán#tự động#,#góp phần#thực hiện#mục tiêu#xây dựng#công nghệ#tự#sửa#của#ngành công nghiệp#máy tính#.</S>

<S>Hôm qua#,#IBM#bắt đầu#bán#phiên bản#mới#DB2 Version 8#của#phần mềm#cơ sở dữ liệu#.</S>

2.7.1.2. Tạo corpus song ngữ

Một cách tiếp cận hiệu quả và rẻ là thu thập văn bản song ngữ từ Internet. Chúng ta thực các bước sau đây để xây dựng Corpus song ngữ Anh-Việt từ Internet (cũng có thể áp dụng cho các cặp ngôn ngữ khác):

- Download dữ liệu văn bản trên Internet bằng song ngữ Anh-Việt dưới dạng file HTML.

- Thực hiện gióng hàng dữ liệu văn bản ở mức file (tương ứng tên file dữ liệu tiếng Việt tương ứng với tên file dữ liệu tiếng Anh).

- Trích lọc ra tất cả các đoạn text từ các file HTML trên tương ứng Anh-Việt. Tương tự như bước 2 trong việc xây dựng Corpus thô.

- Thực hiện việc gióng hàng đoạn giữa hai ngôn ngữ Anh-Việt, sau bước này, ta được các đoạn song ngữ tương ứng Anh-Việt.

- Thực hiện việc gióng hàng câu, sau bước này ta thu được các cặp câu song ngữ Anh-Việt.

- Từ Corpus song ngữ này, chúng ta loại bỏ các câu sai (kiểm tra thủ công). Chỉ giữ lại các cặp câu mà chắc chắn đúng.

- Thực hiện tiền xử lý đối với cả hai ngôn ngữ Anh-Việt. Công việc này bao gồm: phân tích từ vựng tiếng Anh, phân đoạn tiếng Việt, phân tích hình thái, …

2.7.2. Phân đoạn từ trong corpus tiếng Việt (Segmentation)

Bài toán phân đoạn từ tiếng Việt là cho trước một văn bản tiếng Việt, cần xác định trong văn bản đó ranh giới giữa các từ trong câu. Nhưng khác với một số tiếng nước ngoài như tiếng Anh, thì trong tiếng Việt ranh giới giữa các từ nhiều trường hợp không phải là dấu cách trống. Ví dụ, trong câu nói

“phân_đoạn từ tiếng_Việt là một bài_toán quan_trọng”, chúng ta có thể thấy dấu cách trống không phải là dấu hiệu để nhận ra ranh giới của các từ.

Hiện nay có nhiều phương pháp phân đoạn từ trong tiếng Việt, đó là:

2.7.2.1. Phương pháp Maximum Matching

Phương pháp khớp tối đa (MM-Maximum Matching) hay còn gọi là LRMM-Left Right Maximum Matching. Phương pháp này sẽ duyệt một ngữ hoặc câu từ trái sang phải và chọn từ có nhiều âm tiết nhất có mặt trong từ điển và cứ thực hiện lặp lại như vậy cho đến hết câu.

Dạng đơn giản của phương pháp dùng để giải quyết nhập nhằng từ đơn. Giả sử chúng ta có một chuỗi ký tự C1, C2, …, Cn. Chúng ta sẽ áp dụng phương pháp từ đầu chuỗi. Đầu tiên kiểm tra xem C1 có phải là từ hay không, sau đó kiểm tra xem C1C2 có phải là từ hay không. Tiếp tục thực hiện như thế cho đến khi tìm được từ dài nhất.

Dạng phức tạp: Quy tắc của dạng này là phân đoạn từ. Thông thường người ta chọn phân đoạn ba từ có chiều dài tối đa. Thuật toán bắt đầu từ dạng đơn giản, cụ thể là nếu phát hiện ra những cách tách từ gây nhập nhằng, như ở ví dụ trên, giả sử C1 là từ và C1C2 cũng là một từ, khi đó chúng ta kiểm tra ký tự kế tiếp trong chuỗi C1, C2, ..., Cn để tìm tất cả các đoạn ba từ có bắt đầu với C1 hoặc C1C2.

2.7.2.2. Phương pháp Transformation-based Learning (TBL)

Phương pháp học cải biến (TBL) tiếp cận dựa trên tập ngữ liệu đã đánh dấu. Theo cách tiếp cận này, để huấn luyện cho máy tính nhận biết ranh giới giữa các từ, ta sẽ cho máy “học” các câu mẫu trong tập ngữ liệu đã được đánh dấu ranh giới giữa các từ đúng . Rõ ràng chúng ta thấy phương pháp rất đơn giản, vì chỉ cần cho máy học các tập câu mẫu và sau đó máy sẽ tự rút ra quy

luật của ngôn ngữ và để từ đó sẽ áp dụng chính xác khi có những câu đúng theo luật mà máy đã rút ra. Và rõ ràng để tách từ được hoàn toàn chính xác trong mọi trường hợp thì đòi hỏi phải có một tập ngữ liệu tiếng Việt thật đầy đủ và phải được huấn luyện lâu để có thể rút ra các luật đầy đủ.

2.7.2.3. Phương pháp dựa trên thống kê từ Internet và thuật giải di truyền

Phương pháp tách tách từ tiếng Việt dựa trên thống kê từ Internet và thuật giải di truyền – IGATEC (Internet and Genetics Algorithm based Text Categorization for Documents in Vietnamese) là một hướng tiếp cận mới trong tách từ với mục đích phân loại văn bản mà không cần dùng đến một từ điển hay tập ngữ liệu học nào. Hướng tiếp cận này kết hợp giữa thuật toán di truyền (Genetics Algorithm - GA) với dữ liệu thống kê được lấy từ Internet.

2.7.3. Đánh giá theo dữ liệu huấn luyện

Đánh giá theo dữ liệu huấn luyện là việc ta thay đổi kích thước của tập ngữ liệu đầu vào, việc thay đổi này là quá trình làm tăng hoặc giảm số câu (số dòng) trong cặp ngữ liệu đầu vào đó. Việc thay đổi dữ liệu huấn luyện sẽ làm ảnh hưởng đến mô hình dịch, mô hình ngôn ngữ, … từ đó ảnh hưởng rất lớn đến quá trình đánh giá chất lượng của dịch máy.

2.7.4. Đánh giá theo mô hình gióng hàng từ trong văn bản

Công cụ dùng để gióng hàng từ phổ biến nhất hiện nay là GIZA++. Ban đầu, ngữ liệu song ngữ Anh – Việt được gióng hàng từ cả hai phía, từ tiếng Anh sang tiếng Việt và từ tiếng Việt sang tiếng Anh. Quá trình này tạo ra hai gióng hàng từ. Lấy phần giao hai gióng hàng từ này chúng ta sẽ có gióng hàng từ với độ chính xác cao (high-precision).

Trong dịch máy thống kê, ngoài sử dụng GIZA++ để gióng hàng, người ta còn sử dụng gióng hàng Cross-EMword Aligner (Berkerly). Cross-EMword Aligner là phần mềm gióng mã nguồn mở dựa trên phương pháp gióng hàng Alignment by Agreement. Phương pháp này dựa trên sự quan sát, dự đoán phần giao nhau của 2 mô hình so với từng mô hình riêng rẽ. Sau khi dự đoán cả 2 mô hình thống nhất, thêm một bước thứ ba đó là "thỏa thuận" giữa 2 mô hình này.

CHƯƠNG 3 – THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Công cụ tiền xử lý cho hệ dịch

3.1.1. Môi trường triển khai

Phần cứng: Bộ xử lý Core i5, RAM 3GB, HDD free 250GB

Phần mềm: Hệ điều hành Ubuntu 12.04 64 bit

3.1.2. Chuẩn bị dữ liệu đầu vào cho hệ dịch

Dữ liệu đầu vào là dữ liệu song ngữ Việt – Anh Sử dụng gần 70.000 cặp câu Việt – Anh.

3.1.3. Huấn luyện mô hình dịch

- Sử dụng bộ công cụ mã nguồn mở Moses (đã được trình bày ở chương 3)

- Sử dụng mô hình ngôn ngữ SRILM

- GIZA++ là chương trình dùng để gióng hàng từ và trình tự của các từ trong corpus song ngữ nhằm mục đích liên kết các mô hình phụ thuộc vào lớp từ. GIZA++ thực thi mô hình dóng hàng HMM: Baum Welch training, thuật toán Forward-Backward…; GIZA++ là biến thể của mô hình IBM 3 và 4. GIZA được thiết kế và viết bởi Franz Josef Och.

3.2. Kết quả thực nghiệm

3.2.1. Dữ liệu đầu vào

Dữ liệu	Ngôn ngữ	Câu	Từ	Độ dài trung bình	Tên tệp tin thực nghiệm
Dữ liệu huấn luyện	Tiếng Anh	74642	1096072	14.68	*50001b_train.en*
	Tiếng Việt	74642	*1140470*	*15.27*	*50001b_train.vn*
	Tiếng Anh	54643	*614578*	*11.24*	*50001b_train.en*
	Tiếng Việt	54643	580754	10.62	*50001b_train.vn*
	Tiếng Anh	44638	498041	11.15	*50001b_train.en*
	Tiếng Việt	44638	463795	10.39	*50001b_train.vn*
	Tiếng Anh	34638	356602	10.29	*50001b_train.en*
	Tiếng Việt	34638	334097	9.64	*50001b_train.vn*
	Tiếng Anh	24638	253886	10.30	*50001b_train.en*
	Tiếng Việt	24638	239951	9.73	*50001b_train.vn*
Dữ liệu điều Chỉnh tham số	Tiếng Anh	201 câu	*2403*	11.95	*50001_dev.en*
	Tiếng Việt	201 câu	2221	11.04	*50001_dev.en*
Dữ liệu đánh giá	Tiếng Anh	500 câu	5620	11.24	*50001_test.en*
	Tiếng Việt	500 câu	5264	10.52	*50001_test.vn*

3.2.2. Quá trình chuẩn bị dữ liệu và huấn luyện

3.2.2.1. Chuẩn bị dữ liệu

~/tools/moses/scripts/tokenizer/tokenizer.perl -l vn
<~/tools/Work/50001_utf8/Baseline/data/50001b_train.vn.1 >
~/tools//Work/50001_utf8/Baseline/data/50001b_train.tok.vn
~/tools/moses/scripts/tokenizer/tokenizer.perl -l fr
<~/tools/Work/50001_utf8/Baseline/data/50001b_train.vn.1 >
~/tools/Work/50001_utf8/Baseline/data/50001b_train.tok.vn
~/tools/moses/scripts/tokenizer/lowercase.perl <
~/tools/Work/50001_utf8/Baseline/data/50001b_train.tok.vn >
~/tools/Work/50001_utf8/Baseline/data/50001b_train.lower.vn
~/tools/moses/scripts/tokenizer/lowercase.perl <
~/tools/Work/50001_utf8/Baseline/data/50001b_train.tok.en >
~/tools/Work/50001_utf8/Baseline/data/50001b_train.lower.en

3.2.2.2. Huấn luyện mô hình ngôn ngữ

~/tools/srilm/bin/i686-m64/ngram-count -order 3 -interpolate -kndiscount -unk -text ~/tools/Work/50001_utf8/Baseline/lm/50001b_train.lower.en -lm ~/tools/Work/50001_utf8/Baseline/lm/5001b.srilm

3.2.2.3. Sinh ra bảng cụm từ

~/tools/moses/scripts/trainning/train-model.perl -roor-dir ~/tools/Work/50001_utf8/Baseline -corpus ~/tools/Work/50001_utf8/Baseline /data/50001b_train.lower \-f vn -e en -alignment grow-diag-final-and -reordering msd-bidirectionnal-fe \-lm 0:3:HOME/Work/50001_utf8/Baseline/lm/50001b.srilm:8 -external-bin-dir ~/tools/bin >& ~/tools/Work/50001_utf8/Baseline/tranning.out &

~/tools/moses/scripts/tokenizer/tokenizer.perl -l en
<~/tools/Work/50001_utf8/Baseline/data/50001_dev.en.1 >
~/tools/Work/50001_utf8/Baseline/data/50001_dev.tok.en
~/tools/moses/scripts/tokenizer/tokenizer.perl -l en
<Work/50001_utf8/Baseline/data/50001_dev.vn.1 >
Work/50001_utf8/Baseline/data/50001_dev.tok.vn
~/tools/moses/scripts/tokenizer/lowercase.perl <
~/tools/Work/50001_utf8/Baseline/data/50001_dev.tok.vn >
~/tools/Work/50001_utf8/Baseline/data/50001_dev.lower.vn
~/tools/moses/scripts/tokenizer/lowercase.perl <
~/tools/Work/50001_utf8/Baseline/data/50001_dev.tok.en >
~/tools/Work/50001_utf8/Baseline/data/50001_dev.lower.en
~/tools/moses/scripts/tokenizer/tokenizer.perl -l fr <
~/tools/Work/50001_utf8/Baseline/data/50001_test.vn.1 >
~/tools/Work/50001_utf8/Baseline/data/50001_test.tok.vn
~/tools/moses/scripts/tokenizer/tokenizer.perl -l en <
~/tools/Work/50001_utf8/Baseline/data/50001_test.en.1 >
~/tools/Work/50001_utf8/Baseline/data/50001_test.tok.en
~/tools/moses/scripts/tokenizer/lowercase.perl <
~/tools/Work/50001_utf8/Baseline/data/50001_test.tok.en >
~/tools/Work/50001_utf8/Baseline/data/50001_test.lower.en
~/tools/moses/scripts/tokenizer/lowercase.perl <
~/tools/Work/50001_utf8/Baseline/data/50001_test.tok.vn >
~/tools/Work/50001_utf8/Baseline/data/50001_test.lower.vn

3.2.2.4. Training tham số của mô hình dịch máy

~/tools/Work/corpus5000 nohup nine
~/tools/moses/scripts/training/mert-moses.pl
~/tools/Work/50001_utf8/Baseline/tuning/50001_dev.lower.vn
~/tools/Work/50001_utf8/Baseline/tuning/50001_dev.lower.en
~/tools/moses/bin/moses
~/tools/Work/50001_utf8/Baseline/moses.ini -mertdir ~/tools/moses/bin/&>
~/tools/Work/50001_utf8/Baseline/tuning/mert.out &
~/tools/moses/scripts/reuse-weights.perl
~/tools/Work/50001_utf8/Baseline/tuning/moses.ini <
~/tools/Work/50001_utf8/Baseline/model/moses.ini >
~/tools/Work/50001_utf8/Baseline/tuning/moses-tuned.ini
~/tools/moses/scripts/training/filter-model-given-input.pl
~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.lower
~/tools/Work/50001_utf8/Baseline/tuning/moses-tuned.ini
~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.lower.vn

3.2.2.5. Dich ra văn bản tiếng anh

:~/tools/Work/50001_utf8/Baseline$ nohup nice ~/tools/moses/bin/moses -config ~/tools/Work/50001_utf8/Baseline/tuning/moses-tuned.ini -input-file ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.lower.vn 1> ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned.output 2> ~/tools/Work/50001_utf8/Baseline/evaluation/tuned.decode.out &
~/tools/moses/bin/moses -config ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.lower/moses.ini -input-file ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.lower.vn 1> ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned-

filtered.output 2> ~/tools/Work/50001_utf8/Baseline/evaluation/tuned-filtered.decode.out &

~/tools/moses/scripts/recaser/recase.perl -model ~/tools/Work/50001_utf8/Baseline/recaser/moses.ini -in ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned-filtered.output -moses ~/tools/moses/bin/moses > ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned-filtered.output.recased

~/tools/scripts/detokenizer.perl -l vn < ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned-filtered.output.recased > ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned-filtered.output.detokenized

3.2.2.6 Đánh giá kết quả

~/tools/Work/50001_utf8/Baseline/plain2sgm -r test ~/tools/Work/50001_utf8/Baseline ~/tools/Work/50001_utf8/Baseline ~/tools/Work/50001_utf8/Baseline/data/50001_test.vn.1 ~/tools/Work/50001_utf8/Baseline/50001_test.vn.sgm

~/tools/Work/50001_utf8/Baseline/plain2sgm -s test ~/tools/Work/50001_utf8/Baseline ~/tools/Work/50001_utf8/Baseline ~/tools/Work/50001_utf8/Baseline/data/50001_test.en.1 ~/tools/Work/50001_utf8/Baseline/50001_test.en.sgm

~/tools/Work/50001_utf8/Baseline/plain2sgm -t test ~/tools/Work/50001_utf8/Baseline ~/tools/Work/50001_utf8/Baseline ~/tools/Work/50001_utf8/Baseline/evaluation/50001_test.tuned-filtered.output ~/tools/Work/50001_utf8/Baseline/50001_test.tuned-filtered.output.sgm

~/tools/Work/50001_utf8/Baseline/mteval-v11b.pl -r
~/tools/Work/50001_utf8/Baseline/50001_test.vn.sgm -s
~/tools/Work/50001_utf8/Baseline/50001_test.en.sgm -t
~/tools/Work/50001_utf8/Baseline/50001_test.tuned-filtered.output.sgm -c

3.3. Đánh gía và phân tích kết quả theo một số tiêu chí như cỡ dữ liệu huấn luyện, chiều tăng của độ dài cụm từ.

3.3.1. Đánh giá và phân tích keeys quả theo cỡ dữ liệu huấn luyện.

Ta thay đổi kích cỡ của dữ liệu huấn luyện lần lượt là 20.000, 30.000,40000, 50.000 , 70000 cặp câu, sau đó thực hiện đánh giá chất lượng dịch dựa vào điểm BLEU. Điểm BLEU càng cao thì chất lượng dịch càng tốt.

Câu	20.000	30.000	40.000	50.000	70.000
Điểm Bleu	8.2	9.5	12.6	14.1	17.7

Bảng 6: So sánh điểm BLEU của bảng cụm từ

4 . Một số kết quả sau khi dịch từ tiếng Việt sang tiếng Anh

xin chào	**hello**
tôi yêu em	**i love you**
tôi được đưa cho cái đĩa	**i was taken for the plate**
kỳ nghỉ mát ở Ai Cập	**a holiday er egypt .**
hôm nay trời mưa	**today it rains**
cửa hàng của tôi	**my shop**
anh tôi đang đi bơi	**my brother is to go swimming**
chị tôi là giáo viên	**my sister is teacher**
Em gái tôi là bác sĩ	**my sister is the teacher**

KẾT LUẬN

Luận văn đã đưa ra phương pháp dịch máy thống kê dựa trên cụm từ là một trong những phương pháp dịch đang được áp dụng rộng rãi trên thế giới. ví dụ như Google, Vietgle, Systran...vvv . nó đã khắc phục được các nhược điểm của dịch máy dựa vào từ và dựa vào luật.Từ mô hình đó tôi đã nghiên cứu và ứng dụng vào dịch ngôn ngữ Việt _ Anh. Mặc dù chất lượng dịch chưa cao, nhưng khi chúng ta cải tiến mô hình dịch đồng thời đưa nhiều dữ liệu nguồn hơn nữa, chất lượng dịch sẽ được nâng lên

1. Các công việc đạt được của luận văn

- Trình bày được tổng quan về hệ dịch máy đặc biệt là dịch máy thống kê dựa vào cụm từ.
- Giải thích được bộ công cụ mã nguồn mở Moses.
- Thử nghiệm mô hình dịch máy và cho kết quả tương đối khả quan

2. Hướng phát triển

Với những kết quả đạt được trong luận văn này, trong tương lai hi vọng sẽ cải thiện được chất lượng dịch và thời gian dịch bằng cách cập nhật các ngữ liệu đầu vào đủ lớn, giảm kích thước của bảng cụm từ, thay đổi một vài tham số để quá trình huấn luyện các mô hình được tốt hơn, cải tiến một số mô hình đảo cụm….

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1] Nguyễn Văn Vinh (2005*). "Xây dựng chương trình dịch tự động Anh-Việt bằng phương pháp dịch thống kê"*. Luận văn Thạc sĩ, Đại học Công nghệ, ĐHQGHN.

[2] Đào Ngọc Tú (2012). " Nghiên cứu về dịch máy thống kê dựa vào cụm từ và thừ nghiệm với cặp ngôn ngữ Anh _ Việt". Luận văn Thạc sĩ Học viện công nghệ bưu chính viễn thông

Tài liệu tiếng Anh

[3] W. Weaver (1955). Translation (1949). In: *Machine Translation of Languages*, MIT Press, Cambridge, MA.

[4] F. Och and H. Ney. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, pp 29(1):19-51

[5] P. Koehn, F.J. Och, and D. Marcu (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*.

[6] Chris Callison-Burch, Miles Osborne and Philipp Koehn (2006), Re-evaluating the Role of Bleuin Machine Translation Research

[7] D. Chiang (2005). A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*.