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

-------📖-------

KIỀU CÔNG CHÍNH

TỐI ƯU BẢNG CỤM TỪ ĐỂ CẢI TIẾN DỊCH MÁY THỐNG KÊ

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

Thái Nguyên - 2015

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC BIỂU ĐỒ

DANH SÁCH CÁC TỪ VIẾT TẮT

Viết tắt	Đầy đủ
PB-SMT	Cụm từ base Statistical Machine Translation
SMT	Statistical Machine Translation
PR-Enc	Cụm từ Rank Encoding

MỞ ĐẦU

Hiện nay trên thế giới có khoảng 5650 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. Mặt khác, với việc bùng nổ Internet như hiện nay, có một khối lượng văn bản khổng lồ trên Internet mà phần lớn là bằng tiếng Anh. Do tính đa dạng của nó mà việc hiểu các văn bản này hoàn toàn không dễ chút nào. Do đó việc có một hệ dịch tự động Anh-Việt là hết sức cần thiết. Với những khó khăn như vậy người ta đã phải dùng đến một đội ngũ phiên dịch khổng lồ, để dịch các văn bản, tài liệu, lời nói từ tiếng nước này sang tiếng nước khác. Những công việc đó mang tính chất thủ công, nặng nhọc trong khi khối lượng văn bản cần dịch ngày càng nhiều. Để khắc phục những nhược điểm trên hiện nay có rất nhiều những hệ thống tự động dịch miễn phí trên mạng như: systran, google translate, vietgle, vdict... 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) [1]. Điều ấy cho thấy sự phát triển của dịch máy càng ngày càng tiến gần hơn đến ngôn ngữ tự nhiên của con người.

Ngay từ khi xuất hiện chiếc máy tính điện tử đầu tiên người ta đã tiến hành nghiên cứu về dịch máy. Công 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 ích lợi to lớn về mặt chiến lược và 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 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.

Với sự phát triển mạnh mẽ của dịch máy tự động thì dịch máy thống kê (Statistical Machine Translation) đã chứng tỏ là một hướng tiếp cận đầy tiềm năng bởi ưu điểm vượt trội so với các phương pháp dịch máy dựa trên cú pháp truyền thống. Kết quả thực tế của hệ thống dịch máy thống kê tốt hơn, ngôn ngữ dịch càng ngày càng gần với ngôn ngữ của người, 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. Để dịch hiệu quả thì bảng cụm từ phải lớn chính vì vậy việc lưu trữ và tìm kiếm trong bảng cụm từ là rất quan trọng. Chính vì thế, luận văn này tôi lựa chọn và thực hiện đề tài "**Tối ưu bảng cụm từ để cải tiến dịch máy thống kê**".

CHƯƠNG I: DỊCH MÁY THỐNG KÊ TRÊN CƠ SỞ CỤM TỪ

Hiện nay dịch máy thông kê dựa trên cơ sở cụm từ là một trong những hướng phát triển đang được rất nhiều người quan tâm. Dịch máy thống kê dựa trên cụm từ nhằm mục đích dịch một văn bản từ ngôn ngữ nguồn sang ngôn ngữ đích dựa vào bảng ngữ cụm từ sau khi thực hiện việc gióng hàng từ, gióng hàng thống kê, đảo cụm từ… kết hợp với mô hình ngôn ngữ.

1.1 Ngôn ngữ tự nhiên

Ngôn ngữ tự nhiên là những ngôn ngữ được con người sử dụng trong các giao tiếp hàng ngày nghe, nói, đọc, viết. Mặc dù con người có thể dễ dàng hiểu và học các ngôn ngữ tự nhiên, việc làm cho máy hiểu được ngôn ngữ tự nhiên không phải là chuyện dễ dàng. Sở dĩ có khó khăn là do ngôn ngữ tự nhiên có các bộ luật, cấu trúc ngữ pháp phong phú hơn nhiều các ngôn ngữ máy tính, hơn nữa để hiểu đúng nội dung các giao tiếp, văn bản trong ngôn ngữ tự nhiên cần phải nắm được ngữ cảnh của nội dung đó.

Do vậy, để có thể xây dựng được một bộ ngữ pháp, từ vựng hoàn chỉnh, chính xác để máy có thể hiểu ngôn ngữ tự nhiên là một việc rất tốn công sức và đòi hỏi người thực hiện phải có hiểu biết sâu về ngôn ngữ học. Do đó cần phải tìm ra một phương pháp dịch tư động tối ưu để làm giảm công sức trong vấn đề về dịch ngôn ngữ nói chung.

1.2 Dịch máy

Dịch tự động hay còn gọi là dịch máy là một trong những ứng dụng quan trọng của xử lý ngôn ngữ tự nhiên, là sự kết hợp của ngôn ngữ, dịch thuật và khoa học máy tính. Như tên gọi dịch tự động là việc thực hiện dịch một ngôn ngữ đầu vào (ngôn ngữ này gọi là ngôn ngữ nguồn) sang một hoặc nhiều ngôn ngữ khác (gọi là ngôn ngữ đích) bằng các công cụ, phần mềm trên máy tính đã được lập trình sẵn mà không cần có sự can thiệp của con người.

Do được lập trình sẵn bằng công cụ, thuật toán trên máy tính nên hầu hết việc dịch tự động đều mang tính sát nghĩa, hoặc mang tính tương đối. Ngày nay người ta đã phát triển nhiều phương pháp để tối ưu hóa khả năng dịch của máy tính.

Dịch máy có hai hướng tiếp cận chính đó là:

Hướng luật (Rules-based): dịch dựa vào các luật viết tay. Các luật này dựa trên từ vựng hoặc cú pháp của ngôn ngữ. Ưu điểm của phương pháp này là có thể giải quyết được một số trường hợp dịch nhưng lại mất nhiều công sức và tính khả chuyển không cao.

Thống kê (Statistical) [2]: tạo ra bản sử dụng phương pháp thống kê dựa trên bản dịch song ngữ.

1.3 Dịch máy thống kê dựa vào cụm từ

Dịch máy thống kê: Là một phương pháp dịch máy trong đó các bản dịch được tạo ra trên cơ sở các mô hình thống kê có các tham số được bắt nguồn từ việc phân tích các cặp câu song ngữ. Các phương pháp tiếp cận thống kê tương phản với các phương pháp tiếp cận dựa trên luật trong dịch máy cũng như với dịch máy dựa trên ví dụ. Thay vì xây dựng các từ điển, các quy 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 kho ngữ liệu. Chính vì vậy dịch máy thống kê có tính khả chuyển cao và áp dụng được cho bất cứ cặp ngôn ngữ nào. Ý tưởng đầu tiên của dịch máy thống kê đã được giới thiệu bởi Warren Weaver [2] vào năm 1949, bao gồm cả những ý tưởng của việc áp dụng lý thuyết thông tin của Claude Shannon. Dịch máy thống kê được tái giới thiệu vào năm 1991 bởi các nhà nghiên cứu làm việc tại Trung tâm nghiên cứu Thomas J.Watson của IBM và đã góp phần đáng kể trong sự hồi

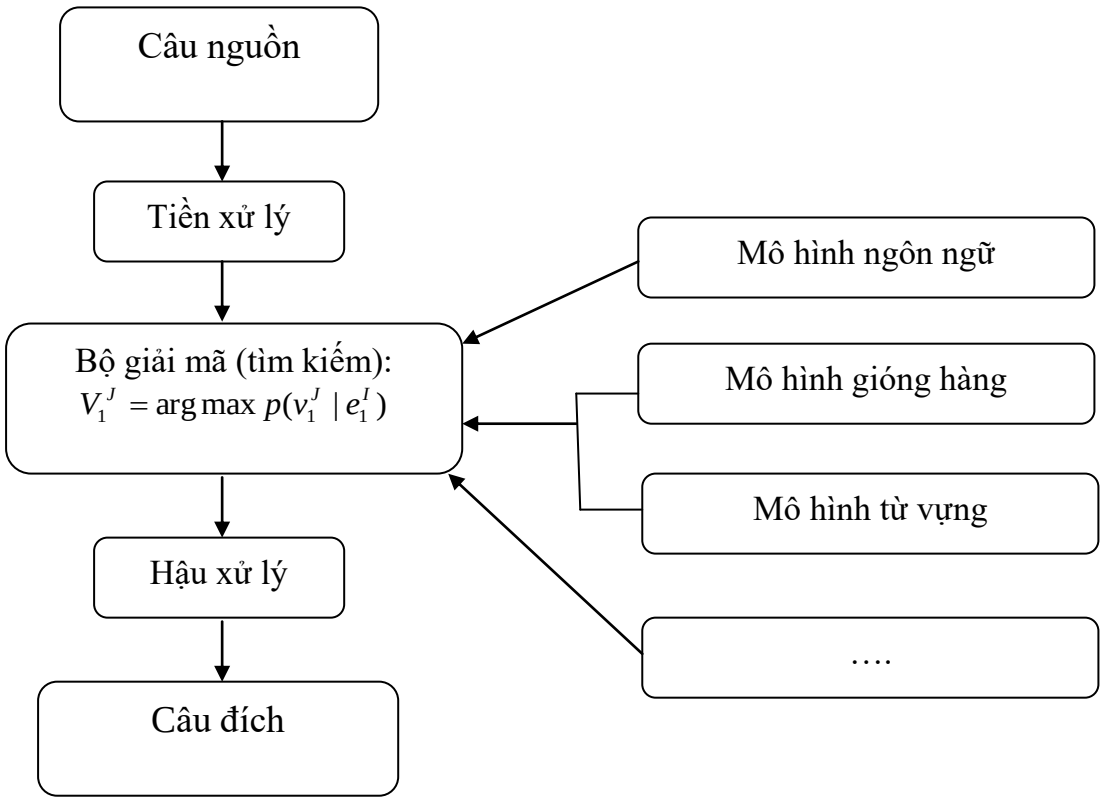
sinh việc quan tâm đến dịch máy trong những năm gần đây. Ngày nay nó là phương pháp dịch máy được nghiên cứu nhiều nhất.

1.3.1 Cơ sở của phương pháp dịch máy thống kê

Mục tiêu là dịch một văn bản từ ngôn ngữ nguồn sang ngôn ngữ đích. Chúng ta có câu văn bản trong ngôn ngữ nguồn ("Tiếng Anh") $e_1^I = e_1,...,e_i$, mà được dịch thành câu văn bản trong ngôn ngữ đích ("Tiếng Việt") $v_1^J = v_1,...,v_j$,. Trong tất cả các câu có thể có trong văn bản đích, chúng ta chọn câu sao cho:

$$V_1^J = \arg\max p(v_1^J \mid e_1^I) \qquad (1.1)$$

Kiến trúc tổng quát của một mô hình dịch thống kê thể hiện trên hình 1.1



Hình 1.1: Sơ đồ của hệ dịch bằng phương pháp thống kê

1.3.2 Gióng hàng từ, gióng hàng thống kê

Gióng hàng xác định ánh xạ i →j = a_i: Từ vị trí i của câu nguồn tương ứng với vị trí j = a_i của câu đích[1]. Việc tìm kiếm được thực hiện dựa vào cực đại biểu thức sau:

$$V = \arg\max_{v_1^j} \left\{ pr(v_1^j) . \sum_{a_1^I} pr(e_1^I, a_1^I / v_1^j) \right\} \qquad (1.2)$$

Do đó, không gian tìm kiếm bao gồm tập tất cả các câu ngôn ngữ đích có thể có v_1^J và tất cả gióng hàng có thể có a_1^I.

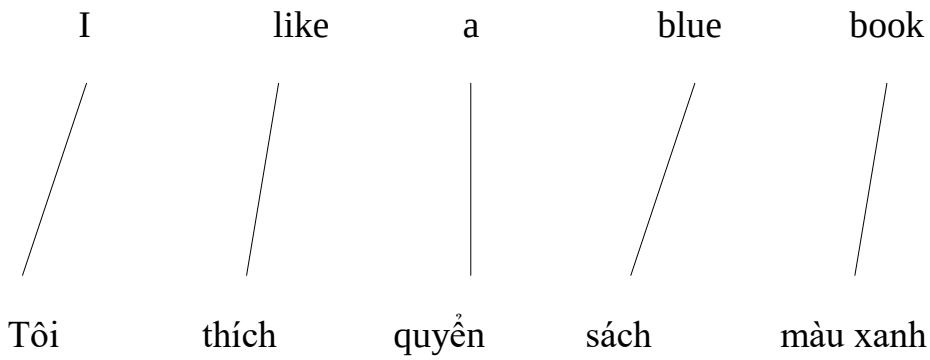
Chúng ta nói rằng cặp xâu kí tự mà xâu này được dịch từ xâu kia từ ngôn ngữ này sang ngôn ngữ khác là cặp xâu dịch. Chúng ta có thể kí hiệu cặp xâu dịch (I like a blue book|Tôi thích quyển sách màu xanh) mà nó biểu diễn là xâu "I like a blue book" (tiếng Anh) được dịch thành "Tôi thích quyển sách màu xanh" (tiếng Việt). Brow và cộng sự [6] đã chỉ ra ý tưởng về việc gióng hàng giữa cặp xâu kí tự dịch như là một sự tương ứng giữa các từ của xâu tiếng Anh với các từ của xâu tiếng Pháp. Điều này ta có thể thấy hoàn toàn tương tự như trong cặp xâu dịch Anh - Việt. Mỗi đương như vậy ta gọi là 1 kết nối. Gióng hàng được biểu diễn bằng đồ thị như hình 1 bằng cách vẽ các đường nối giữa một số từ tiếng Anh và một số từ tiếng Việt.

Ví dụ: Trong hình 1.2, ta có 5 kết nối: (**I(1) like(2) a(3) blue(4) book(5)|Tôi(1) thích(2) quyển(3) sách(4) màu xanh(5)).**

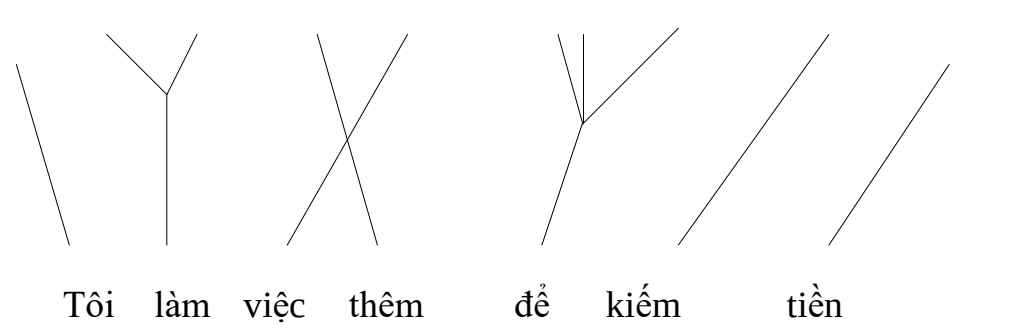
Việc kết nối này có thể là:

- một từ tiếng Anh tương ứng với 1 từ tiếng Việt (hình 1.2)
- một từ tiếng Anh tương ứng nhiều từ tiếng Việt (hình 1.3)
- nhiều từ tiếng Anh tương ứng nhiều từ tiếng Việt (hình 1.4)

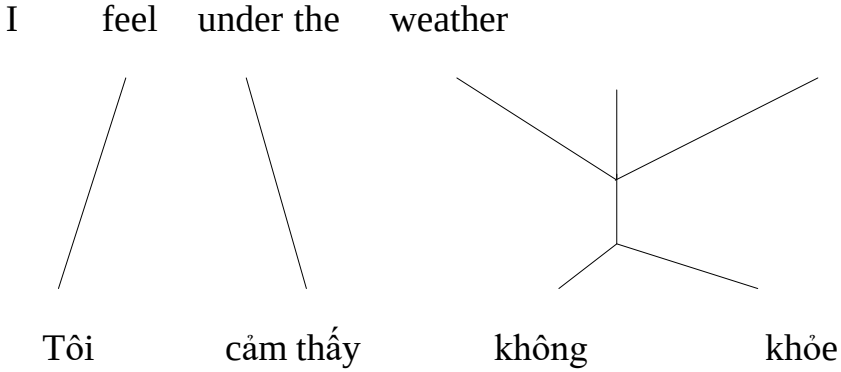
Chúng ta kí hiệu tập gióng hàng của (v/e) là *A(e,v)*. Nếu e có độ dài là I và v có độ dài là J, ta sẽ có I*J liên kết khác nhau giữa J từ tiếng Việt và từ tiếng Anh. Khi đó, số gióng hàng từ cặp câu (v,e) là 2^{IJ}. Nghĩa là số tập con của A(e,v) = 2^{IJ}



Hình 1.2: Gióng hàng với những từ tiếng anh độc lập



Hình 1.3: Gióng hàng với những từ tiếng việt độc lập



Hình 1.4: Gióng hàng tổng quát

Một cách tổng quát, mô hình gióng hàng thống kê phụ thuộc vào tập tham số chưa biết θ mà được học từ dữ liệu huấn luyện. Để biểu diễn sự phụ thuộc của mô hình vào tập các tham số, ta có công thức:

$$\Pr(e,a \mid v) = p_\theta(e,a \mid v) \qquad (1.2)$$

Tính sáng tạo trong mô hình thống kê là phải phát triển các mô hình cụ thể mà nắm bắt các thuộc tính có liên quan của lĩnh vực vấn đề được xem xét. Trong trường hợp của chúng ta, mô hình gióng hàng thống kê phải mô tả mối quan hệ giữa xâu ngôn ngữ nguồn và xâu ngôn ngữ đích tương xứng.

Để huấn luyện tập tham số θ, chúng ta có sẵn corpus song ngữ bao gồm S cặp câu $\{(e_s,v_s) : s = 1, \ldots, S\}$. Với mỗi cặp câu (e_s,v_s), biến gióng hàng được kí hiệu là a. Tập tham số θ được xác định dựa vào cách tiếp cận hợp lý cực đại trong corpus huấn luyện song ngữ:

$$\hat{\theta} = \arg\arg_\theta \prod_{s=1}^{S} \sum_a p_\theta(e,a \mid v) \qquad (1.3)$$

Trong dịch thống kê, chúng ta cố gắng mô hình hóa xác suất dịch Pr(e/v) mà chúng mô tả mối quan hệ giữa câu ngôn ngữ nguồn e và câu ngôn ngữ đích v. Có rất nhiều cách để dịch từ cùng một câu tiếng Anh sang cùng một câu Tiếng Việt. Với mỗi gióng hàng cho ta tương ứng một cách dịch. Vì vậy, ta có công thức quan hệ giữa mô hình dịch và mô hình gióng hàng:

$$P(e \mid v) = \sum_a P(e,a \mid v) \qquad (1.4)$$

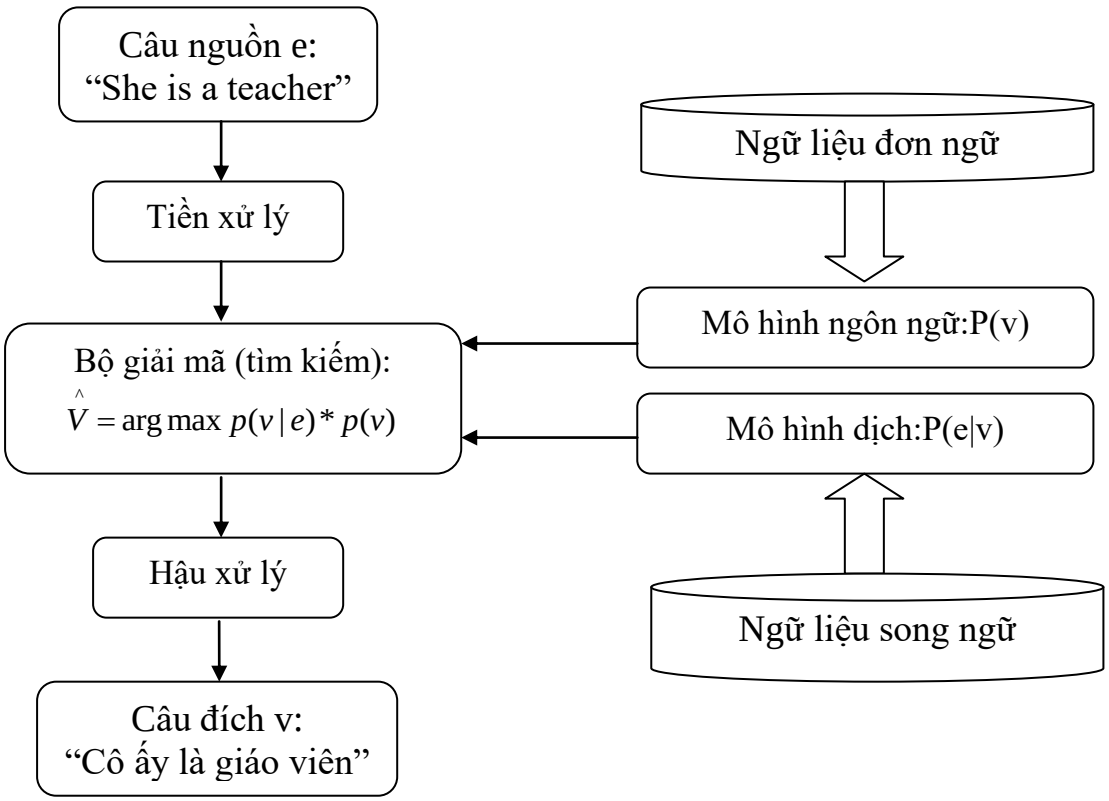
trong đó a là một gióng hàng của cặp (e,v).

1.3.3 Dịch máy thống kê dựa trên cơ sở cụm từ.

Cụm từ (cụm từ) là một nhóm từ kết hợp với nhau tạo thành nghĩa những không đầy đủ.

Nhóm nghiên cứu về dịch máy thống kê ở trường Johns Hopkins đã dựng lên EGYPT [7], một công cụ dịch máy thống kê mã nguồn mở. Trong đó có GIZA++, một công cụ training cho mô hình IBM 1-5, được sử dụng để tạo bảng ánh xạ từ-từ cho nhiều mô hình dịch theo phương pháp bảng cụm từ.

Dịch máy thống kê trên cơ sở cụm từ [5] 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. Việc sử dụng các cụm từ theo nghĩa ngôn ngữ học (tức là dựa trên cú pháp) làm giảm chất lượng của dịch máy bằng phương pháp này.



Hình 1.5: Mô hình dịch từ Tiếng Anh- Tiếng Việt.

Mô hình dịch dựa trên cụm từ thường không thực hiện đúng theo trình tự của phương pháp dựa trên cơ sở từ, mà sử dụng khuôn dạng của bản ghi tuyến tính. Các thành phần như là mô hình ngôn ngữ, mô hình dịch cụm từ, mô hình dịch từ vựng hoặc mô hình đảo cụm đều được sử dụng một cách thích hợp. Khuôn dạng này cho phép tích hợp các tính năng bổ sung như số lượng các từ được tạo ra hoặc số các bản dịch cụm từ được sử dụng.

Trong dịch dựa trên cụm [3], một chuỗi các từ liên tiếp (cụm) được dịch sang ngôn ngữ đích, với độ dài cụm ngôn ngữ nguồn và đích có thể khác nhau. Câu vào được chia thành một số cụm, từng cụm một được dịch sang ngôn ngữ đích, và sau đó các cụm được đảo trật tự theo một cách nào đó rồi ghép với nhau. Cuối cùng ta thu được câu dịch trong ngôn ngữ đích.

Giả sử ta gọi ngôn ngữ nguồn là f và ngôn ngữ đích là e, chúng ta sẽ cố gắng tối đa hóa xác suất Pr(f|e) với mong muốn có được bản dịch tốt nhất. Thực tế là tồn tại rất nhiều bản dịch đúng cho cùng một câu, mục đích của ta là tìm ra câu ngôn ngữ e phù hợp nhất khi cho trước câu ngôn ngữ nguồn f. Dịch dựa vào cụm sử dụng mô hình kênh nhiễu, áp dụng công thức Bayes ta có:

$$\arg\max_e \Pr(e|f) = \arg\max_e \Pr(f|e)\Pr(e)/\Pr(f) \quad (1.5)$$

Do Pr(f) là không đổi đối với e, vấn đề trở thành việc tìm câu e nhằm tối đa hóa Pr(f |e)Pr(e). Việc xây dựng mô hình ngôn ngữ cần sử dụng một ngữ liệu đơn ngữ lớn, trong khi đó mô hình dịch lại cần đến ngữ liệu song ngữ tốt. Bộ giải mã được sử dụng để chia câu nguồn thành các cụm và sinh ra các khả năng dịch có thể cho mỗi cụm nhờ sự trợ giúp của bảng cụm (phrasetable).

Để sinh ra được câu dịch, câu nguồn được chia thành I cụm liên tiếp f_1^I. Chúng ta giả sử rằng phân phối xác suất là như nhau đối với các cụm này. Mỗi cụm f_i trong f_1^I được dịch thành cụm tương ứng trong ngôn ngữ đích e_i.

Các cụm trong ngôn ngữ đích có thể đảo ví trí cho nhau. Quá trình dịch cụm được mô hình hóa bởi phân phối xác suất $p(f_i|e_i)$.

Mô hình đảo cụm thường được mô hình hóa bởi một khoảng cách cơ sở. Đảo cụm thường bị giới hạn bởi sự dịch chuyển số lượng tối đa các từ. Các mô hình đảo cụm thường tuân theo ngữ pháp của ngôn ngữ đích (ví dụ như Tiếng Anh – Tiếng Việt, Với Tiếng Anh thì tính từ nằm trước danh từ, nhưng tiếng Việt thì ngược lại).

1.3.4 Mục đích của việc dịch máy thống kê trên cơ sở cụm từ.

Mục đích chính của việc sử dụng cụm từ trong dịch máy thống kê là để giảm bớt hạn chế của việc dịch máy thống kê trên cơ sở từ [5].

Thông thường với một ngôn ngữ nhất định 1 từ có thể có nhiều nghĩa trong những văn cảnh khác nhau. Việc dịch máy dựa vào dịch từng từ một và sau đó ghép tổ hợp của chúng với nhau thường dẫn đến những kết quả không tốt và phải xử lý một tổ hợp kết quả khá lớn.

Ví dụ : Xét một câu đơn có n từ: AnAn-1….A2A1

Với mỗi từ An, An-1…A1 sẽ có tương ứng Xn, Xn-1, Xn-2… X1 nghĩa

Do vậy với việc dịch trên cơ sở từ thì số ngôn ngữ đích tối đa có thể có sẽ là:

$$(\text{Số Ngôn Ngữ}) = \sum_{i=1}^{n} Xi \qquad (1.6)$$

(chưa sử dụng các thuật toán tối ưu và nén với từ)

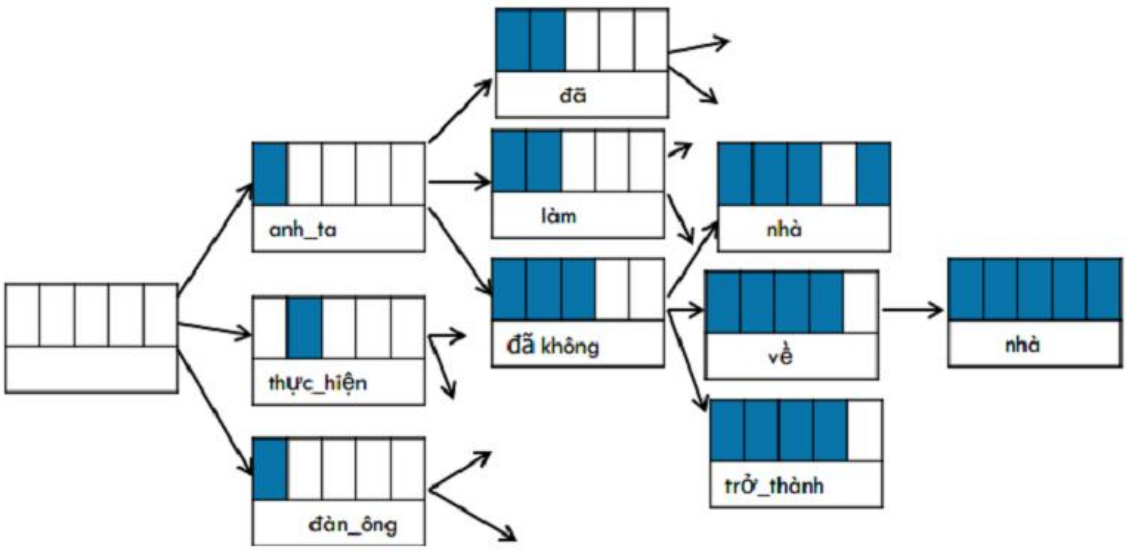
Việc sử dụng cụm từ trong dịch máy sẽ làm tăng độ chính xác của dịch máy đồng thời làm giảm đáng kể thời gian dịch của máy.

Ví dụ:

He	**did**	**not**	**go**	**home**
Anh_ta	làm	không	đi	nhà
Nó thực_hiện	không	phải	trở thành	chỗ ở
Đàn_ông	đã	không_đúng	về	quê_hương
nó làm		trở_thành		quê_hương
anh_ta	đã		đi_về	
đã_không				
làm_không_đúng				

Mở rộng không gian giả thuyết:

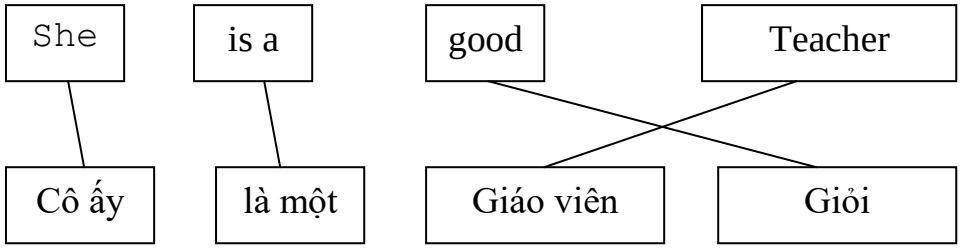
He	Did	Not	Go	Home



Hình 1.6: Mô tả việc giải mã

1.3.5 Đảo cụm từ trong dịch máy thống kê

Đơn vị dịch: Cụm từ, là một chuỗi các từ liên tiếp bất kỳ



- Mỗi cụm tiếng Việt v_j ứng với một cụm tiếng Anh e_i
- $(e_i|v_j)$: xác suất dịch cụm từ

Các cụm từ có thể bị dịch chuyển:

+ $d(start_i\text{-}end_{i-1}\text{-}1)$: xác suất chuyển dịch
+ $start_i$: vị trí đầu tiên của cụm từ tiếng Anh ứng với v_i
+ end_{i-1}: vị trí cuối của cụm từ tiếng Anh ứng với v_{i-1}

⇨ Xác suất p(e|v) :

$$p(e \mid v) = \prod_{i=1}^{i} \varphi(e_i \mid v_i) d(start_i - end_{i-1} - 1) \quad (1.7)$$

1.3.6 Bảng cụm từ trong dịch máy thống kê

Đối với dịch máy thống kê trên cơ sở cụm từ, ta cũng cần phải có một bộ ngữ liệu liên quan đến các cụm từ. Chính vì vậy bảng cụm từ đã được xây dựng. Bảng cụm từ được sử dụng trong dịch máy thống kê dựa trên cụm từ là rất lớn. Kích thước của chúng là một hệ quả trực tiếp của cách tiếp cận bảng cụm từ trong dịch máy thống kê sao cho sự tiên đoán trước có thể truy cập được một cách hiệu quả. Việc tính toán trước sẽ làm tăng lên tổ hợp của cụm từ và cụm từ dư thừa cho bất kỳ cụm từ nào và tất cả các cụm từ con (Subphrase) có thể có được trong bảng cụm từ. Bảng cụm từ được lưu trữ một cách rõ ràng hiện nay là đại diện được sử dụng rộng rãi nhiều nhất các mô hình dịch trong PB-SMT.

Phương pháp được sử dụng trong việc thực hiện tối ưu bảng cụm từ (JunczysDowmunt, 2012a, b) cho Moses (Koehn 2007) [4] có thể được sử dụng để thay thế cho các bảng cụm từ nhị phân hiện tại.

1.4 Mô hình ngôn ngữ

Phương pháp dịch máy thông kê dựa trên xác xuất để xuất hiện ngôn ngữ đích khi cho đầu vào là một ngôn ngữ nguồn. Việc thống kê dựa trên bộ ngữ liệu có sẵn và ta chỉ xác định xác xuất nào là lớn nhất để chọn ra kết quả ngôn ngữ đích phù hợp.

Ví dụ: Khi áp dụng mô hình ngôn ngữ cho tiếng Việt

P[“hôm nay là thứ hai”]=0.003

P[“hai nay là thứ hôm”]=0

Mô hình ngôn ngữ được áp dụng trong nhiều lĩnh vực của xử lý ngôn ngữ tự nhiên có nhiều hướng tiếp cận mô hình ngôn ngữ nhưng chủ yếu được xây dựng theo mô hình ngôn ngữ N-gram.

Mô hình ngôn ngữ N-gram:

*Nhiệm vụ của mô hình ngôn ngữ là cho biết xác suất của một câu w1w2... wm là bao nhiêu. Theo công thức Bayes: P(AB) = P(B|A) * P(A), thì:*

*P(w1w2…wm) = P(w1) * P(w2|w1) * P(w3|w1w2) *…* P(wm|w1w2…wm-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(wm|w1, w2, …, wm-1) = P(wm|wm-n, wn-m+1, …, wm-1)

Nếu áp dụng xấp xỉ Markov, xác suất xuất hiện của một từ (wm) được coi như chỉ phụ thuộc vào n từ đứng liền trước nó (wm-nwm-n+1…wm-1) chứ không

phải phụ thuộc vào toàn bộ dãy từ đứng trước (w1w2…wm-1). Như vậy, công thức tính xác suất văn bản được tính lại theo công thức:

*P(w1w2…wm) = P(w1) * P(w2|w1) * P(w3|w1w2) *…* P(wm-1|wm-n-1wm-n …wm-2) * P(wm|wm-nwm-n+1…wm-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 1 dãy con gồm n phần tử liên tiếp nhau của 1 dãy các phần tử cho trước.

CHƯƠNG II: PHƯƠNG PHÁP TỐI ƯU BẢNG CỤM TỪ

Để tăng chất lượng dịch trong dịch máy thống kê thì bảng cụm từ được sử dụng trong dịch máy thống kê dựa trên cụm từ (PB-SMT) có kích thước rất lớn. Vì vậy, để lưu trữ và tìm kiếm bảng cụm từ một cách hiệu quả là vấn đề đang được nghiên cứu và quan tâm trong dịch máy thống kê. Mục đích của phương pháp sẽ trình bày dưới đây là để mô tả mệnh đề, phương pháp mã hóa làm giảm dữ liệu từ, từ đó giảm đáng kể thời gian dịch máy nhưng vẫn đảm bảo chất lượng dịch một mức nhất định. Phương pháp mã hóa cụm [8] nhằm mục đích giảm sự dư thừa bằng cách khai thác bảng cụm từ như một từ điển nén, sử dụng các mối quan hệ tịnh tiến.

2.1 Quy trình sinh bảng cụm từ

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. Để tạo ra được bảng cụm từ ta sử dụng script train-model.perl trong phần mềm Moses [4], các giai đoạn của thủ tục huấn luyện:

1. chuẩn bị dữ liệu.
2. Chạy bộ công cụ Giza++.
3. Dóng hàng từ.
4. Nhận bảng dịch từ vựng.
5. Chiết xuất cụm từ.
6. Điểm cụm từ.
7. Xây dựng mô hình sắp xếp lại từ vựng.
8. Xây dựng hệ mô hình.
9. Tạo tập tin cấu hình.

ví dụ:

nohup nice ~/Tools/moses/scripts/training/train-model.perl -root-dir ~/Work/50001_utf8/Baseline -corpus ~/Work/50001_utf8/Baseline/data/50001b_train.lower \-f en -e vn -alignment grow-diag-final-and -reordering msd-bidirectional-fe \-lm 0:3:$HOME/tools/Work/50001_utf8/Baseline/lm/5001b.srilm:8 -external-bin-dir ~/Tools/bin >& ~/Work/50001_utf8/Baseline/training.out &

(trong đó corpus -f en -e vn là 2 tệp tin ngữ liệu đầu vào sau tiền xử lý *5001b.srilm* là mô hình ngôn ngữ được huấn luyện ở giai đoạn trên)

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

Cụm từ nguồn	**Cụm từ mục tiêu**	**Điểm**	**Gióng hàng**
carries a normal complement	có thuỷ _ thủ đoàn	0.166667 6.72486e-15 0.0666667 1.87836e-14	1-2
carries a normal complement	có thuỷ _	0.166667 6.72486e-15 0.0666667 2.65348e-07	1-2
carries a normal complement	mỗi tàu có thuỷ _ thủ	0.166667 6.72486e-15 0.0666667 3.77986e-18	1-4

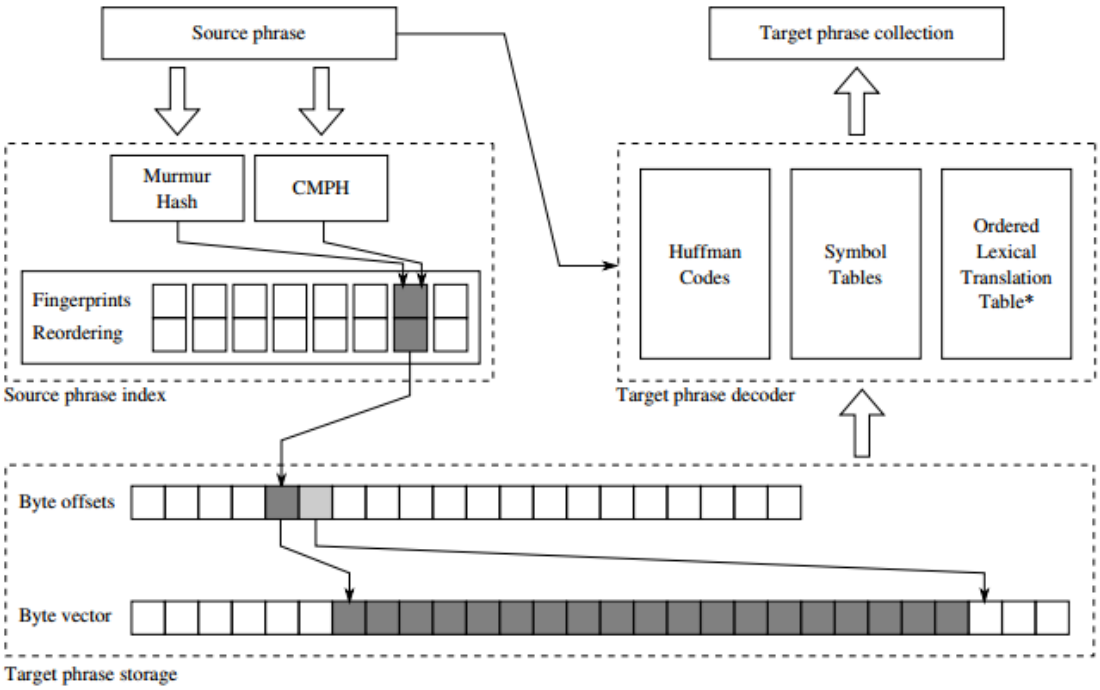
carries a normal complement	mỗi tàu có thuỷ _ thủ đoàn	0.166667 6.72486e-15 0.0666667 6.05912e-22	1-4
carries a normal complement	mỗi tàu có thuỷ _	0.166667 6.72486e-15 0.0666667 8.55947e-15	1-4
carries a normal complement	thuỷ _ thủ	0.111111 6.72486e-15 0.0666667 1.43833e-08	1-1
carries a normal complement	thuỷ _ thủ đoàn	0.125 6.72486e-15 0.0666667 2.30564e-12	1-1
carries a normal complement	thuỷ _	0.0625 6.72486e-15 0.0666667 3.25709e-05	1-1
carries a normal complement	tàu có thuỷ _ thủ	0.166667 6.72486e-15 0.0666667 2.20529e-14	1-3

Bảng 2.1 : Một số phần tử trong bảng cụm từ

2.2 Phương pháp tối ưu bảng cụm từ

2.2.1 Chỉ số cụm từ nguồn

Cấu trúc của các chỉ số cụm từ nguồn lấy cảm hứng bởi Guthrie và cộng sự (2010). Phần quan trọng nhất của chỉ số giống như một chức năng tối thiểu hoàn hảo hàm băm (MPH) mà trong đó gồm một tập S có n cụm từ nguồn đến n số nguyên liên tiếp. Hàm băm này đã được tạo ra với các thuật toán trong thư viện CMPH (Belazzougui và cộng sự, 2009).



Hình 2.1: Sơ đồ đơn giản hóa bảng cụm từ

Các MPH chỉ được đảm bảo để lập bản đồ các yếu tố được biết đến từ S để nhận dạng chính xác các số nguyên. Nếu một cụm từ nguồn được cho rằng đã không được nhìn thấy trong S trong việc xây dựng các MPH, một số nguyên ngẫu nhiên sẽ được gán cho nó. Điều này có thể dẫn đến các bài tập giả của bộ sưu tập cụm từ mục tiêu đến các cụm từ nguồn vô hình. Guthrie và cộng sự (2010) đề xuất sử dụng một thuật toán hash ngẫu nhiên

(MurmurHash3) trong quá trình xây dựng và lưu trữ các giá trị của nó như dấu vân tay cho mỗi cụm từ trong tập S. Đối với truy vấn, nó cũng đủ để tạo ra các dấu vân tay cho các cụm từ đầu vào và so sánh nó với các dấu vân tay được lưu trữ tại vị trí được trả về bởi hàm MPH. Nếu nó phù hợp, cụm từ đã được nhìn thấy và có thể được tiếp tục xử lý. Đối với 32 bit dấu vân tay có một xác suất 2^{-32} của một cụm từ nguồn vô hình trượt qua.

Vị trí ban đầu của cụm từ nguồn trong tập S có thứ tự được lưu trữ cùng dấu vân tay. Trong Moses, Cụm từ nguồn được truy vấn bằng cách di chuyển các điểm bắt đầu của một cụm từ đến mỗi từ trong câu và tăng chiều dài cụm từ cho đến khi giới hạn độ dài hay kết thúc của câu là đạt. Nếu không có trật tự các vị trí của các MPH được giao là ngẫu nhiên mà có thể làm cho một phiên bản bộ nhớ ánh xạ của các cụm và gần như không sử dụng được. Bảng cụm từ này không chứa bất kỳ đại diện của các cụm từ nguồn bên cạnh các chức năng MPH. Cụm từ nguồn có thể được kiểm tra để đưa vào bộ nhớ nhưng không thu hồi được.

2.2.2 Lưu trữ cụm từ mục tiêu

Việc lưu trữ cụm từ mục tiêu bao gồm một vector byte để lưu trữ bộ sưu tập cụm từ mục tiêu liên tục theo thứ tự họ của cụm từ nguồn tương ứng. Một bộ sưu tập cụm từ mục tiêu bao gồm một hoặc nhiều cụm từ mục tiêu được lưu trữ liên tục. Một cụm từ mục tiêu là một chuỗi các ký hiệu từ mục tiêu tiếp theo là một biểu tượng đặc biệt, một chuỗi dài cố định về điểm số, và một chuỗi các điểm liên tiếp nó được dừng lại bằng một biểu tượng dừng đặc biệt.

Khả năng truy cập ngẫu nhiên được bổ sung bởi các vector byte offset. Đối với mỗi bộ sưu tập cụm từ mục tiêu, nó lưu trữ các byte mà bộ sưu tập

này bắt đầu. Bằng cách kiểm tra bù đắp vị trí kết thúc của một cụm từ mục tiêu tiếp theo thu thập có thể được xác định.

Kích giảm là đạt được bằng cách nén các chuỗi biểu tượng của một bộ sưu tập cụm từ mục tiêu bằng cách sử dụng mã hóa biểu tượng khôn ngoan Huffman (Huffman, 1952; Moffat, 1989). Mục tiêu từ cụm, điểm số, và các điểm liên kết được mã hóa với các bộ khác nhau của mã Huffman được hoán đổi trong mã hóa và giải mã.

Trong khi các vector byte chỉ là một mảng lớn các byte, các vector byte offset là một cấu trúc phức tạp hơn. Thay vì giữ offsets là số nguyên 8-byte sự khác biệt giữa các hiệu số được lưu trữ. Một điểm đồng bộ hóa với các giá trị bù đắp đầy đủ là chèn vào và theo dõi cho mỗi 32 giá trị.

Điều này lần lượt các byte vector vào một danh sách các số khá nhỏ, thậm chí nhiều hơn như vậy khi các mảng byte được nén. Kỹ thuật đảo ngược từ danh sách nén cho chỉ số công cụ tìm kiếm được sử dụng để làm giảm kích thước hơn nữa, Simple-9 mã hóa (Anh và Moffat, 2004) cho sự khác biệt offset và Byte Variable Length mã hóa (Scholer và cộng sư, 2002) cho các điểm đồng bộ hóa. Khi cả hai kỹ thuật sử dụng không gian ít hơn nếu số lượng nhỏ hơn được nén, kích thước của cấu trúc giữ giảm với giảm chênh lệch. Do đó một phương pháp nén tốt hơn cho các mảng byte kết quả tự động trong một byte vector nhỏ hơn. Đối với các bảng cụm từ ban đầu, khoảng 215 triệu offsets sử dụng 260 MB, nhưng chỉ có 220 MB cho các biến thể rankencoded. hơn 30 Mbytes. Mã Huffman được lưu trữ như mã Huffman kinh điển, một đại diện bộ nhớ hiệu quả. Kích thước của các bộ giải mã cụm từ mục tiêu được coi là một phần của kích thước cần thiết để đại diện cho các cụm từ mục tiêu.

2.2.3 Nén ngữ liệu song ngữ

Conley và Klein (2008) đã đề xuất một chương trình mã hóa dữ liệu ngôn ngữ mục tiêudựa trên cở sở liên kết từ và các mối quan hệ tịnh tiến. Cụm từ mục tiêu sẽ được thay thế với các chỉ số bắt đầu và kết thúc của cụm từ tương ứng, chỉ số bản dịch và một con trỏ số nguyên cho mỗi mục tiêu. Cụm từ mục tiêu sẽ được thay thế các điểm với các chỉ số bắt đầu và kết thúc của cụm từ tương ứng, chỉ số bản dịch và một con trỏ số nguyên cho mỗi mục tiêu. Nén được thực hiện bằng việc sử dụng mã hóa Huffman.

Trong khoa học máy tính và lý thuyết thông tin, mã hóa Huffman là một thuật toán mã hóa dùng để nén dữ liệu. Nó dựa trên bảng tần suất xuất hiện các kí tự cần mã hóa để xây dựng một bộ mã nhị phân cho các kí tự đó sao cho dung lượng (số bít) sau khi mã hóa là nhỏ nhất.

Các bước thực hiện nén:

- Đọc file và xác định các ký tự xuất hiện trong file & tần suất của chúng
- Dựng cây mã Huffman
- Dựa vào cây mã thu được mã hóa từng ký tự và ghi vào file nén
- Lưu cây mã vào cuối file nén

Đầu vào.

Tập $A = \{a_1, a_2, \ldots, a_n\}$*, bảng ký tự biểu tượng có kích thước* n*. Trọng lượng* $W = \{w_1, w_2, \ldots, w_n\}$*, Mà là tập hợp của các trọng lượng (dương) của chữ cái (thường là tỷ lệ thuận với xác suất), tức là* $w_i = weight(a_i)$, $1 \leq i \leq n$.

Đầu ra.

$C(A,W) = \{ c_1, c_2, \ldots, c_n \}$ *, Mà là (nhị phân) từ mã, nơi* c_i *là từ mã cho* (a_i), $1 \leq i \leq n$.

Kỹ thuật này hoạt động bằng cách tạo ra một cây nhị phân của các nút, có thể được lưu trữ trong một mảng, kích thước của nó phụ thuộc vào số lượng các biểu tượng n. Một nút có thể là một nút lá hoặc một nút nội bộ. Ban đầu, tất cả các nút là nút lá, trong đó có chứa các biểu tượng riêng của mình, trọng lượng (tần suất xuất hiện) của các biểu tượng và tùy chọn, một liên kết đến một nút cha mà làm cho nó dễ dàng để đọc mã (ngược) bắt đầu từ một nút lá. Các nút nội bộ có trọng lượng biểu tượng là tổng trọng lượng của 2 nút con, liên kết đến hai nút con và liên kết tùy chọn để một nút cha. Như một quy ước chung, bit '0' đại diện cho sau cây con trái và bit '1' đại diện cho sau các cây con phải. Một cây thành phẩm đã lên đến n nút lá và $n - 1$ các nút nội bộ. Một cây Huffman mà bỏ qua những biểu tượng không sử dụng tạo ra độ dài mã tối ưu nhất.

Thuật toán xây dựng cây mã Huffman như sau:

Bước 1: Xác định hai nút "tự do" có trọng số nhỏ nhất.

Bước 2: Nút cha của hai nút này được tạo ra với trọng lượng là tổng trọng lượng của hai nút con.

Bước 3: Nút cha này được thêm vào danh sách các nút. Đánh dấu nút cha là "tự do", hai nút con đánh dấu là "đã xét".

Bước 4: Một trong hai nhánh từ nút cha đến nút con được đánh dấu là "0" và nhánh còn lại được đánh dấu là "1".

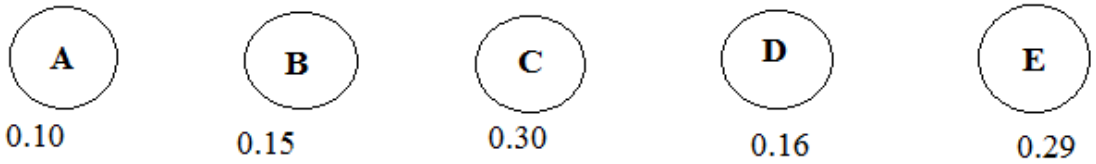
Bước 5: Lặp lại các bước cho đến khi chỉ còn một nút tự do. Nút này chính là nút gốc của cây.

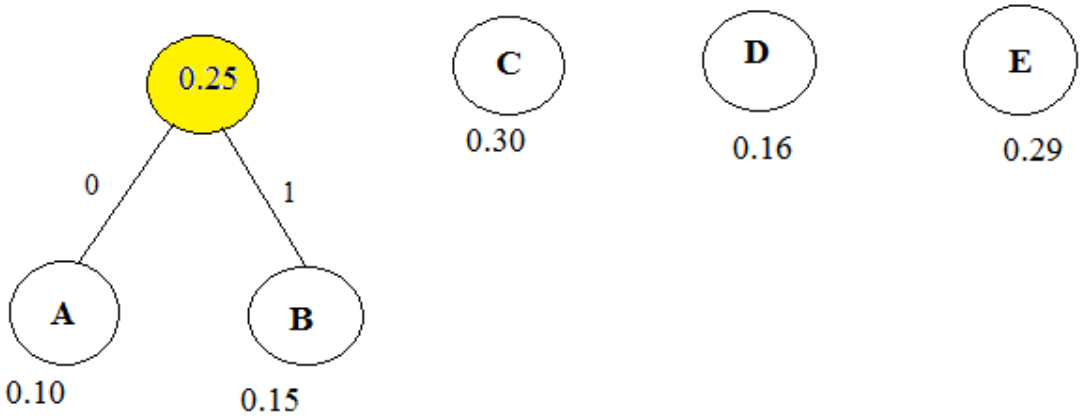
Ví dụ:

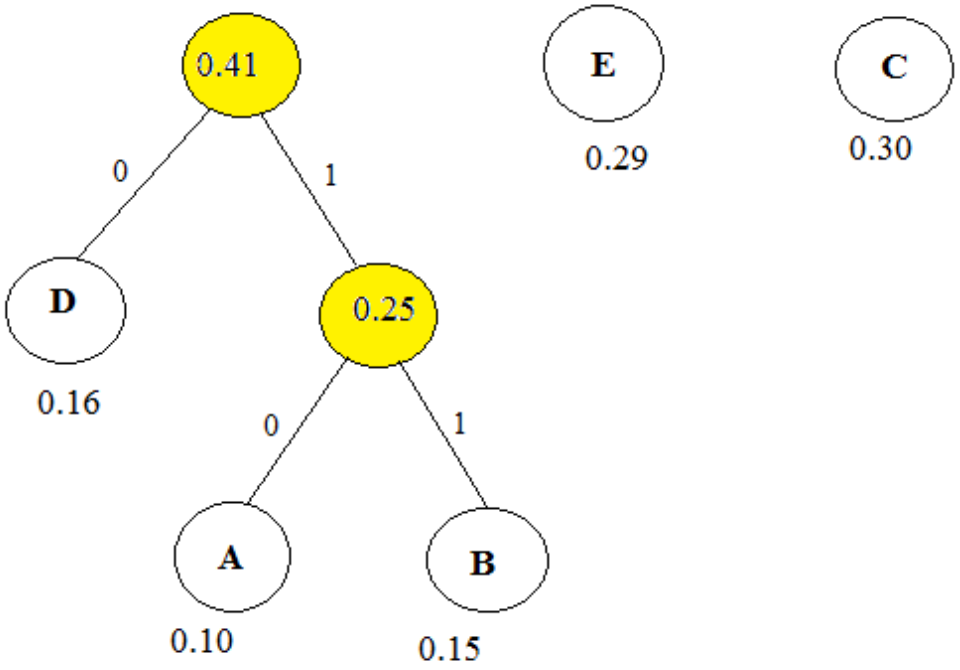
Với bảng tần suất của 5 chữ cái A, B, C, D, E như sau tương ứng là 0.10; 0.15; 0.30; 0.16; 0.29

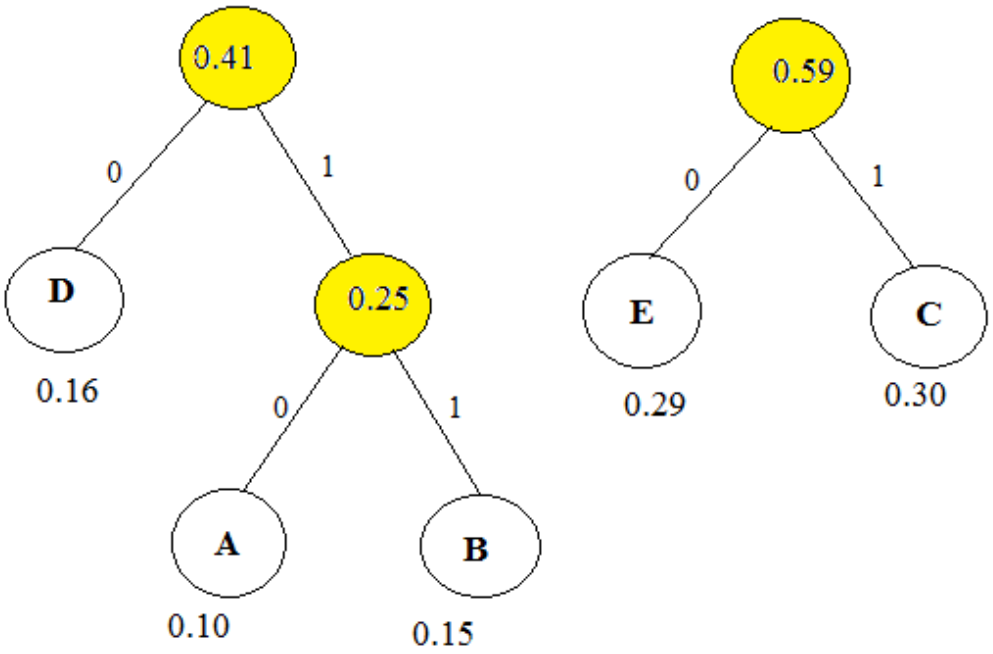
A	B	C	D	E
0.10	0.15	0.30	0.16	0.29

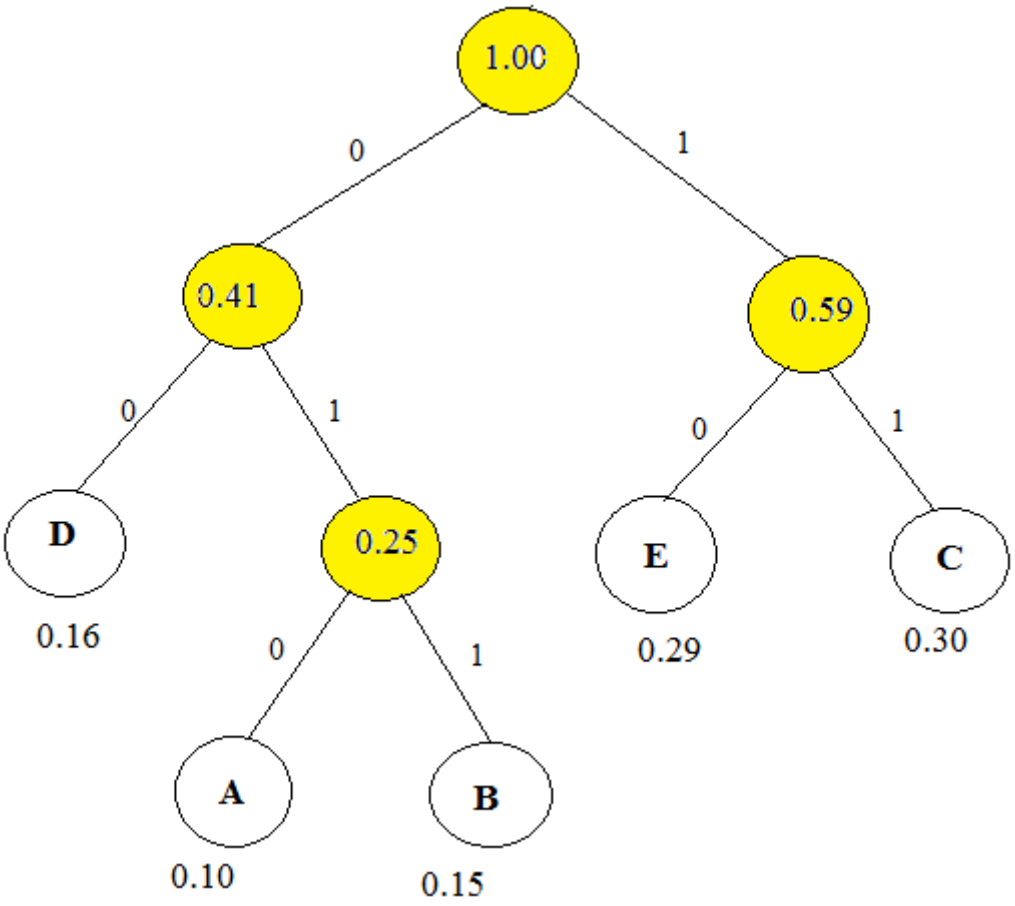
Quá trình xây dựng cây Huffman diễn ra như sau:






0.41
0
1
D
0.16
0.25
0
1
A
0.10
B
0.15
E
0.29
C
0.30


0.41
0
1
D
0.16
0.25
0
1
A
0.10
B
0.15
0.59
0
1
E
0.29
C
0.30



Hình 2.2: Mô tả quá trình tạo cây Huffman

Như vậy bộ mã tối ưu tương ứng là:

A	B	C	D	E
01 0	01 1	11	00	10

Kết quả của Huffman mã hóa cho một mã số với ký tự có trọng lượng nhất định.

Đầu vào ***(A, W)***	Ký tự (a_i)	A	B	C	D	E	**Tổng số**
	Trọng lượng (w_i)	0.10	0.15	0.30	0.16	0.29	= 1
Đầu ra *C*	Mã huffman (c_i)	010	011	11	00	10	
	Chiều dài mã (theo bit) (L_i)	3	3	2	2	2	
	Chiều dài trọng lượng $(L_i w_i)$	0.30	0.45	0.60	0,32	0,58	$L(C)$ = 2,25
Tối ưu	Xác suất (2^{-l_i})	1/8	1/8	1/4	1/4	¼	= 1.00
	Thông tin (theo bit) $(-\mathbf{Log}_2 w_i) \approx$	3,32	2.74	1.74	2,64	1.79	

Bảng 2.2: Bảng mã hóa huffman

2.2.4 Nén bảng cụm từ

Junczys-Dowmunt (2012a) giới thiệu một kiến trúc làm nhỏ bảng cụm từ với việc sử dụng PR-Enc. Cơ bản của việc nén này là sử dụng thuật toán Simple-9 [6],mã hóa biến-byte [9] và mã hóa Huffman (Huffman, 1952) của các từ mục tiêu, điểm và các điểm liên kết.Để giảm được kích thước cho các cụm từ nguồn ta sử dụng một hàm băm với các chỉ số. Juncczys-Dowmunt (2012b) [6] mô tả việc tối ưu của chỉ số cụm từ và tác động của nó đến chất lượng bản dịch. Việc thực hiện tối ưu đã đạt được một kết quả khá tốt với hơn

77% khi so sánh với bảng cụm từ nhị phân Moses với hiệu suất tốt hơn đáng kể.

Mô tả nén sử dụng thuật toán Simple-9: Simple-9 (S9) là một thuật toán đạt được kết quả nén tốt hơn nhiều mã hóa biến byte và cũng đã cải thiện được tốc độ nén. S9 không phải là liên kết byte nó là sự liên kết giữa các từ và bit liên kết. Ý tưởng cơ bản của S9 là cố gắng để đóng gói giống như số nguyên vào một trong 32-bit từ. Mỗi từ được chia thành 4 bit kiểm soát và 28 bit dữ liệu. Những gì chúng ta có thể lưu trữ trong 28 bit:

Lựa chọn (4bit)	Số mã hóa được lưu trữ trong một từ	Chiều dài của mỗi mã trong bit	Bit lãng phí
0000	28	1	0
0001	14	2	0
0010	9	3	1
0011	7	4	0
0100	5	5	3
0101	4	7	0
0110	3	9	1
0111	2	14	0
1000	1	28	0

Bảng 2.3: Bảng tùy chọn mã Simple 9

(giả định cho đơn giản: tất cả các con số mà chúng ta gặp phải cần ít nhất 28 bit). Để làm được điều này, S9 chia mỗi từ thành 4 bit trạng thái và 28 bít dữ liệu.Ví dụ như nếu 7 giá trị tiếp theo mà tất cả chúng đều nhỏ hơn 16, thì chúng ta có thể lưu trữ chúng như dạng 7 giá trị 4-bit, hoặc nếu sau 3 giá trị tiếp theo mà chúng nhỏ hơn 512 thì chúng ta có thể lưu trữ chúng dưới dạng 3 giá trị 9-bit (để lại một bit không dùng). Lưu trữ và lấy số bằng cách sử dụng mặt nạ bit cố định.

Thuật toán Simple 9:

Bước 1: Làm việc 28 số tiếp theo phù hợp với mỗi 1 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 2: Làm việc 14 số tiếp theo phù hợp với mỗi 2 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 3: Làm 9 con số tiếp theo phù hợp với mỗi 3 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 4: Làm việc 7 số tiếp theo phù hợp với mỗi 4 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 5: Làm việc 5 số tiếp theo phù hợp với mỗi 5 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 6: Làm việc 4 số tiếp theo phù hợp với mỗi 7 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 7: Làm 3 con số tiếp theo phù hợp với mỗi 9 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 8: Làm việc 2 số tiếp theo phù hợp với mỗi 14 bit.

- Nếu có: sử dụng trường hợp đó
- Nếu không có thực hiện bước tiếp theo

Bước 9: Làm việc 1 số tiếp theo phù hợp với mỗi 28 bit.

Ví dụ:

Giả sử ta có 1 đoạn string với 8 ký tự (example: “abcdefgh”). Nếu chúng ta đặt vào kiểu mảng thì chúng ta sẽ có mảng với 8 byte.

| a=1 | b=2 | c=3 | d=4 | e=5 | f=6 | g=7 | h=8 |

Ở dạng ASCII

|97|98|99|100|101|102|103|104|

Dạng bit nhị phân tương ứng sẽ là:

00000001|00000010|00000011|00000100|00000101|00000110|00000111|00001000

Với mỗi byte (8bit) thì chúng ta có thể lưu trữ được 2^8=256 ký tự, tuy nhiên trên thực tế với ngôn ngữ giao tiếp thì số ký tự thường chỉ ở mức 24-28 chữ cái (đối với ngôn ngữ la-tinh) do vậy thay vì sử dụng 8bit ta chỉ cần tới 5bit(tối đa 2^5=32) là có thể lưu trữ được 1 chữ cái.

Chuyển về dạng 5 bit thì ta sẽ cắt bỏ 3 bit đầu đi và kết quả sẽ là

00001|00010|00011|00100|00101|00110|00111|01000

Ta có thể gộp dạng trên về dạng:

00001000|10000110|01000010|10011000|11101000

Như vậy ta đã chuyển từ 8 byte về dạng 5 byte trong việc mã hóa string: abcdef

2.2.5 Mã hóa cụm từ

Ý tưởng chung của việc mã hóa cụm tương tự như các phương pháp nén dựa trên từ điển cổ điển. Các cụm từ con (subcụm từs) lặp đi lặp lại được thay thế bằng con trỏ để subcụm từs trong một từ điển cụm từ nên kết quả giảm chiều dài dữ liệu.Việc giải nén dựa trên việc tìm kiếm và chèn lại của các cụm từ con trong ký tự con trỏ. Nếu chúng ta đơn giản hóa những phương thức trên bằng việc xóa tất cả các dữ liệu yêu cầu bên ngoài và chuyển nó tới bảng cụm từ. Thay vì việc nén một bitext với một từ điển các cụm từ, chúng ta nén từ vựng riêng của mình. Việc mã hóa thứ hạng cụm sẽ chia sẻ thuộc tính với mã hóa thứ hạng từ và chỉ ra phương thức nén bitext. Phương pháp nén được thực hiện với một cách khác: Dữ liệu tuần tự được biểu diễn dạng đồ thị cấu trúc giống như dạng cây hoặc là máy tự động.

Mã hóa thứ hạng cụm cũng có thể sử dụng để biểu diễn dữ liệu mục tiêu từ bảng cụm từ nhị phân Moses dựa trên Zens và Ney(2007) vào một cấu trúc đồ thị.

Mã hóa yêu cầu từ bảng cụm từ để có những thông tin liên kết từ. Để thực hiện việc mã hóa có hiệu quả nó có thể tìm các cặp cụm từ liên kết và lấy lại thứ hạng của cụm từ mục tiêu liên quan tới cụm từ nguồn tương ứng. Danh

sách được sắp xếp giảm dần của xác suất dịch P, tức là bản dịch tốt nhất sẽ có cấp bậc là 0, bảng dịch chất lượng càng kém thì có cấp bậc càng cao.

Cho một câu Spanish –English với các cặp cụm từ đã được dóng.

es: Maria no daba una bofetada a la bruja verde

en: Mary did not slap the green witch

Các cặp cụm từ này được đại diện bởi các quadruple (1 bộ 4 chỉ số) bao gồm các chỉ số: vị trí cụm từ nguồn, vị trí cụm từ đích, chiều dài cụm từ nguồn, chiều dài cụm từ đích. Các cặp cụm từ hoàn chỉnh bị loại bỏ vì nó không là các cặp cụm từ con được dóng đúng điều kiện đầu tiên của biểu thức, các cặp cụm từ con phải nằm trong đường bao của các cụm mã hóa.

Zens và các cộng sự (2002) định nghĩa các cụm từ con phù hợp với việc dóng cơ bản với thủ tục tương tự được sử dụng trong suốt quá trình rút các cặp cụm từ khi mô hình dịch được tạo để tránh việc tự tham chiếu.

Các cụm từ con được chèn vào hàng đợi dựa theo thứ tự: "các cụm con được đánh thứ tự giảm dần theo chiều dài, vị trí bắt đầu dịch và sau đó đến chiều dài và vị trí bắt đầu của cụm nguồn".

Với ví dụ trên cặp cụm từ nguồn là hàng đợi:

es: no daba una bofetada a la bruja verde

en: did not slap the green witch

Nó sẽ được kiểm tra để đưa vào bảng xếp hạng cụm từ với thứ hạng tính từ 0.

Cụm từ đích được thay thế bởi 1 ký hiệu con trỏ

es: Maria no daba una bofetada a la bruja verde

en: Mary (0,0,0)

Các giá trị nguyên của bộ 3 này được hiểu như sau:

- Đầu tiên là sự khác nhau giữa vị trí nguồn và đích của cặp cụm con.

-Thứ 2 là khoảng của đường bao cụm nguồn con bên phải từ cuối cụm mã hóa.

-Giá trị cuối cùng là thứ hạng của cặp cụm con được chọn.

Tất cả các điểm được dóng nằm trong đường bao của cặp cụm con được chọn được loại và tất cả những cặp cụm con chồng lên cặp cụm con đang xét được xóa khỏi hàng đợi.

Chỉ 1 cặp cụm từ còn lại trong hàng đợi là:

Es: Maria

En: Mary

Áp dụng thủ tục tương tự như trên

Các cụm mã hóa sau được tạo ra:

es: Maria no daba una bofetada a la bruja verde

en: (0,8,0) (0,0,0)

các cụm con đích mà không được thay thế được tìm thấy sẽ được giữ lại như những từ plain.

2.2.6 Giải mã cụm từ

Các bộ giải mã cụm từ mục tiêu có chứa các dữ liệu cần thiết để giải mã các luồng byte nén. Nó bao gồm các danh sách nguồn và từ mục tiêu với các chỉ số, các bộ mã Huffman, và nếu Rank Encoding được sử dụng, một bảng dịch từ vựng được sắp xếp. Mã Huffman được lưu trữ như mã Huffman kinh điển, một đại diện bộ nhớ hiệu quả. Kích thước của các bộ giải mã cụm từ mục tiêu được coi là một phần của kích thước cần thiết để đại diện cho các

cụm từ mục tiêu. Trong việc thực hiện cơ bản ba bộ khác nhau của Huffman mã được sử dụng để mã hóa từ mục tiêu, điểm số, và sắp xếp; mã hóa và giải mã dựa trên chuyển đổi giữa ba loại mã Huffman. Đối với những từ cụm từ mục tiêu và liên kết chỉ một biểu tượng dừng đặc biệt đã được thêm vào. Điểm và các điểm liên kết được mã hóa trực tiếp, từ cụm từ mục tiêu có một đại diện trung gian như định danh số nguyên đó đang nhìn lên một bảng từ mục tiêu. Việc thực hiện này được gọi là "đường cơ sở".

Một thủ tục giải mã đơn giản xử lý cây chủ yếu là cây nhị phân với thời gian mũ. Tuy nhiên nếu xem xét tất cả các cụm đích ứng với một câu, một thuật toán quy hoạch động với độ phức tạp tuyến tính cho mỗi cụm có thể được xây dựng.

Moses truy vấn bảng cụm từ xử lý các câu theo kiểu từ trái sang phải bắt đầu với các cụm từ con có chiều dài 1 và tăng dần chiều dài của nó khi đạt đến giới hạn, sau đó chuyển sang từ tiếp theo với chiều dài lại là 1.

Do đó, nếu 1 cụm từ được tìm thấy thì các tiền tố của nó đã được xử lý trước đó. Nếu tất cả các cụm từ truy vấn được cache lại cho việc giải mã và tất cả các cụm được sử dụng cho việc giải mã được cache lại cho việc tìm kiếm thì tổng số bảng cụm từ truy cập là tương tự như trong một bảng cụm từ tuyến tính. Với việc caching một cụm từ đích cho cụm nguồn: “Maria no daba una bofetada” sẽ được tìm thấy ngay lập tức mà không cần duyệt các nhánh còn lại. Các cụm từ con "a la" sẽ vẫn được xử lý, nhưng khi Moses truy vấn cụm từ đó, nó sẽ được lấy từ bộ nhớ cache.

Mọi tập các cụm từ đích cho 1 câu nguồn được tạo. Nếu một cụm đích với thứ hạng cho trước được quan sát thấy trước đó nó sẽ được phục hồi từ cache. Với trường hợp khác nếu một phiên bản giải mã được chuyển từ bảng cụm từ tới DecodeTargetPharse. Cụm con đích giải mã sau đó được nối với

cụm đích hiện tại và điểm dóng cụm được thêm vào. Dóng đầu ra được dịch chuyển phù hợp, kết quả được cache lại.

CHƯƠNG III: ĐÁNH GIÁ THỰC NGHIỆM BẰNG HỆ DỊCH MÁY THỐNG KÊ MOSES

Moses là một hệ thống dịch máy thống kê cho phép chúng ta tự đào tạo mô hình dịch cho cặp câu song ngữ. Tất cả những điều chúng ta cần là thu thập các bản dịch song ngữ.

- ✓ Có khả năng tự động huấn luyện các mô hình dịch
- ✓ Input: bộ dữ liệu song ngữ
- ✓ Thuật toán tìm kiếm: Tìm ra bản dịch tốt nhất có thể

3.1 Môi trường triển khai

Cấu hình phần cứng và phần mềm cài đặt .

-CPU Intel Core i5

-RAM 4GB

-Hệ điều hành Ubuntu 12.04 LTS

Hệ thống dịch máy Moses có thể cài đặt trên các Os khác nhau như Linux, OSX hay Windows. Ở phần demo này chúng ta sẽ cài đặt và chạy các test case trên Linux cụ thể là Ubuntu phiên bản 12.04 LTS.

Các công cụ đi theo:

- ✓ Hệ thống đã cài Boost
- ✓ SRILM
- ✓ CMPH Library

Công cụ xây dựng mô hình dịch: GIZA++, mkcls

3.2 Xây dựng chương trình dịch và thực hiện nén bảng cụm từ.

3.2.1 Chuẩn hóa dữ liệu

Dữ liệu đầu vào cần được chuẩn hóa theo đúng dạng qui định

Việc chuẩn hóa dữ liệu có thể bao gồm những công việc như:

- ✓ Tách từ
- ✓ Tác câu
- ✓ Chuyển sang chữ thường, chữ hoa
- ✓ Loại bỏ từ dư thừa
- ✓ …

Việc chuẩn hóa dữ liệu là một trong những bước tiền xử lý trong hệ dịch máy. Có nhiều phương pháp để chuẩn hóa dữ liệu đầu vào đang được cung cấp miễn phí dưới dạng mã nguồn mở.

3.2.2 Xây dựng mô hình ngôn ngữ, mô hình dịch

SRILM là một gói công cụ để xây dựng mô hình dịch ngôn ngữ. Nó giúp chúng ta xây dựng được mô hình ngôn ngữ trước khi cho vào máy dịch

Sử dụng GIZA++ để xây dựng mô hình dịch và dùng mkcls để ước lượng giá trị cực đại cho mỗi mô hình.

Sau khi chuẩn hóa dữ liệu và đã xây dựng mô hình ngôn ngữ, mô hình dịch việc tiếp theo của chúng ta là dịch máy. Bộ ngữ liệu song ngữ trên các cặp ngôn ngữ khác nhau. Tiến hành dịch và so sánh kết quả.

3.2.3 Nén bảng cụm từ

Cài đặt thêm thư viện CMPH vào hệ thống Moses. CMPH library là một gói thư viện hỗ trợ cho hệ thống Moses trong việc nén bảng cụm từ và sắp xếp lại từ vựng.

./bjam --with-cmph=/path/to/cmph

Trong đó path/to/cmph là đường dến đến thư viện CMPH. Khi cài

CMPH thì cần phải cài các gói thư viện boost hoặc g++.

Ta sử dụng lệnh sau để nén bảng Cụm từ theo PR-Enc sử dụng thuật toán Huffman và simple9 đã cài đặt cùng với thư viện CMPH.

moses/bin/processCụm từTableMin

-in ~/tools/Work/50001_utf8/Baseline/model/phrasetable.gz

-out ~/tools/Work/50001_utf8/Baseline/model/phrasetable -use-alignment -threads 4

Sau khi chạy xong câu lệnh thì hệ thống sẽ tạo cho chúng ta một bảng cụm từ mới có kích thước giảm đi đáng kể (mặc định tên file: phrase_table.minphr).

3.2.4 Đánh giá kết quả dịch

Kết quả dịch máy thống kê có chính xác hay không đều dựa vào các chỉ số dịch máy. Có 2 chỉ số cần quan tâm đó là chỉ số BLEU [10] và chỉ số NIST.

a. 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.

Chỉ số BLEU có giá trị nằm từ 0 đến 1. Chỉ số càng gần 1 thì chất lượng dịch càng tốt, chỉ số càng nhỏ gần tới 0 thì chất lượng dịch càng kém.

BLEU tính điểm bằng cách đối chiếu kết quả dịch với tài liệu dịch tham khảo và tài liệu nguồn. Mặc dù 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 đảm bảo chính xác để đánh giá trên một hệ thống

dịch thống kê. Chính vì vậy, trong luận văn này, điểm BLEU được sử dụng làm tiêu chuẩn đánh giá chất lượng dịch.

Chúng tôi lấy trung bình hình học của các điểm chính xác sửa đổi các văn dữ liệu thử và sau đó nhân kết quả của một yếu tố hình phạt ngắn gọn theo cấp số nhân. Hiện nay, trường hợp gấp là việc bình thường hóa văn bản chỉ được thực hiện trước khi tính toán độ chính xác. Đầu tiên chúng ta tính trung bình hình học của độ chính xác *n-gram* sửa đổi, p_n, sử dụng n-gram đến chiều dài N và trọng lượng tích cực W_N cách tổng hợp một. Tiếp theo, gọi c là độ dài của các cụm từ mục tiêu và r là chiều dài tham khảo dữ liệu hiệu quả. Chúng ta ước tính phạt ngắn gọn BP.

$$BP = \begin{cases} 1, & nếu\ c < r \\ e^{(1-r/c)}, & nếu\ c \geq r \end{cases}$$

Sau đó,

$$\text{BLEU= BP.exp}\left(\sum_{n=1}^{n} w_n\ logp_n\right)$$

Các cách xếp hạng là nhiều hơn ngay lập tức rõ ràng trong việc tính bleu.

$$log\ BLEU = min(1\text{-}r/c,\ 0) + \sum_{n=1}^{n} w_n\ logp_n$$

Trong cơ sở của chúng tôi, chúng tôi sử dụng N = 4 và trọng lượng đồng nhất $w_n = 1/N$.

b. Chỉ số NIST

Về cơ bản phương pháp đánh giá nhờ chỉ số NIST cũng tương tự như chỉ số BLEU nhưng nó cũng có một số khác biệt.

Chỉ số NIST cung cấp thông tin cần thiết để đánh giá trọng số dịch.

3.3 Thực nghiệm và đánh giá kết quả dịch tiếng Anh sang tiếng Việt

Từ bộ dữ liệu gồm hơn 70000 câu tiếng anh và hơn 70000 câu tiếng việt. Sau khi đã training dữ liệu và sinh bảng cụm từ (cụm từ-table.gz)

Ngữ liệu tiếng Việt:

Một ngữ liệu nhỏ đơn ngữ tiếng Việt cũng được sử dụng với mục đích củng thêm cố kết quả với việc thử nghiệm trên nhiều ngữ liệu khác nhau. Ngữ liệu này được xây dựng từ nhiều bài viết trên "Báo Lao động" phiên bản điện tử thuộc nhiều lĩnh vực khác nhau như khoa học, kinh tế, thể thao, văn hóa [1]. Các thống kê về ngữ liệu này được liệt kê trong bảng dưới đây:

Dung lượng	Gzip	Số lượng câu	Số lượng từ	Độ dài trung bình câu
5.88 Mb	1.58Mb	74642	1140470	15.27

Bảng 3.1: Ngữ liệu tiếng việt.

Ngữ liệu tiếng Anh:

Dung lượng	Gzip	Số lượng câu	Số lượng từ	Độ dài trung bình câu
8.12Mb	1.73Mb	74642	1096072	14.68

Bảng 3.2: Ngữ liệu tiếng anh.

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*

Bảng 3.5: Dữ liệu đầu vào

3.3.1 Thực nghiệm dịch với câu đơn giản.

"She is a student"

echo 'She is a student' | ~/moses/bin/moses -f model/moses.ini > out

So sánh kết quả giữa bảng cụm từ gốc và sau khi tối ưu

Tiêu chí so sánh	*Bảng cụm từ gốc*	*Bảng cụm từ đã tối ưu*
Dung lượng bảng	343.0 Mb	43.9 Mb (~ 12,8 %)
Thời gian tai vào bộ nhớ	64,592s	33,550s
Thời gian dịch câu <Chỉ tính từ lúc đã load xong từ bộ nhớ và thực hiện dịch>	0.122s	0.034s

Bảng 3.3: So sánh kết quả dịch máy với một câu đơn.

Như vậy ta thấy kết quả thu được rất khả quan.

Từ bảng cụm từ có dung lượng lên tới 343.0 Mb ta đã thực hiện nén xuống còn 43.9Mb điều này là rất đáng kể trong việc nén dữ liệu. Do đó thời gian load dữ liệu vào hệ thống và thời gian dịch của hệ thống tăng lên đáng kể. Với phương pháp nén bảng cụm từ đã mang lại kết quả khả quan trong việc nén dữ liệu và tăng tốc độ dịch máy. Và đây là một trong những phương pháp khá tốt được nhiều người sử dụng trên thực tiễn.

3.3.2 Thực nghiệm dịch 1 đoạn văn bản từ tiếng Anh-Tiếng Việt

Bước tiếp theo ta sẽ trực tiếp so sánh dịch một văn bản từ tiếng Anh sang tiếng Việt. Dữ liệu đầu vào đều được tối ưu và chuẩn hóa để tăng tốc độ dịch máy. Một số công cụ chuẩn hóa dữ liệu đầu vào có ghi trong phụ lục 2 của khóa luận,tất cả chúng đều được cung cấp dạng mã nguồn mở.

Thí dụ: Đầu vào là một file đã được chuẩn hóa có tên là *500001b_lower.en*

Gõ lệnh sau để dịch file đầu vào và in ra kết quả.

~/moses/bin/moses –f model/moses.ini <path_to_file_input> file_out_put

So sánh kết quả giữa bảng cụm từ gốc và sau khi tối ưu:

Tiêu chí so sánh	*Bảng cụm từ gốc*	*Bảng cụm từ đã tối ưu*
Dung lượng file/line	6.8kb/100line	
Thời gian tai vào bộ nhớ	58.329(s)	57.325(s)
Thời gian dịch câu <Chỉ tính từ lúc đã load xong từ bộ nhớ và thực hiện dịch>	121(s)	87(s)

Bảng 3.4: So sánh hai phương pháp dịch với đầu vào là một văn bản

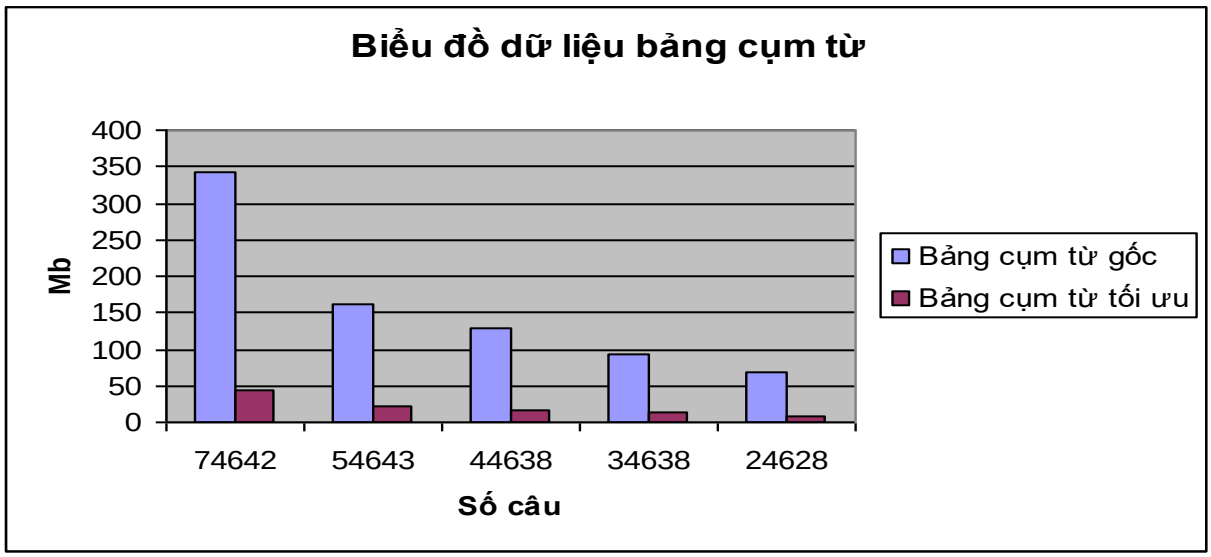
3.3.3 Đánh giá kết quả dữ liệu huấn luyện bảng cụm từ

Dữ liệu huấn luyện được thay đổi theo 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.

Tiêu chí	Số câu	Bảng cụm từ	Bảng cụm từ tối ưu
Dữ liệu huấn luyện	74641 câu	343.0Mb	43.9Mb
	54641 câu	162.7Mb	21.8Mb
	44641 câu	129.9Mb	17.4Mb
	34641 câu	93.0Mb	12.5Mb
	24641 câu	68.0Mb	9.2Mb

Bảng 3.5: So sánh dữ liệu bảng cụm từ gốc và bảng cụm sau khi nén



Biểu đồ 3.1: Biểu đồ so sánh 1.

Nhìn vào biểu đồ 3.1 của bảng cụm từ trước gốc và bảng cụm từ sau khi nén ta thấy dung lượng của bảng được cải thiện đáng kể ~12%. Ở đây xét tập ngữ liệu trên 70.000 câu kích cỡ của bảng cụm từ là 343.0 Mb, giả sử với tập ngữ liệu lên tới 1 triệu câu thì dung lượng bảng cụm từ sẽ lên tới ~5.0 Gb (dữ liệu lớn) nếu không tối ưu thì chúng ta không thể đưa toàn bộ dữ liệu vảo bộ nhớ

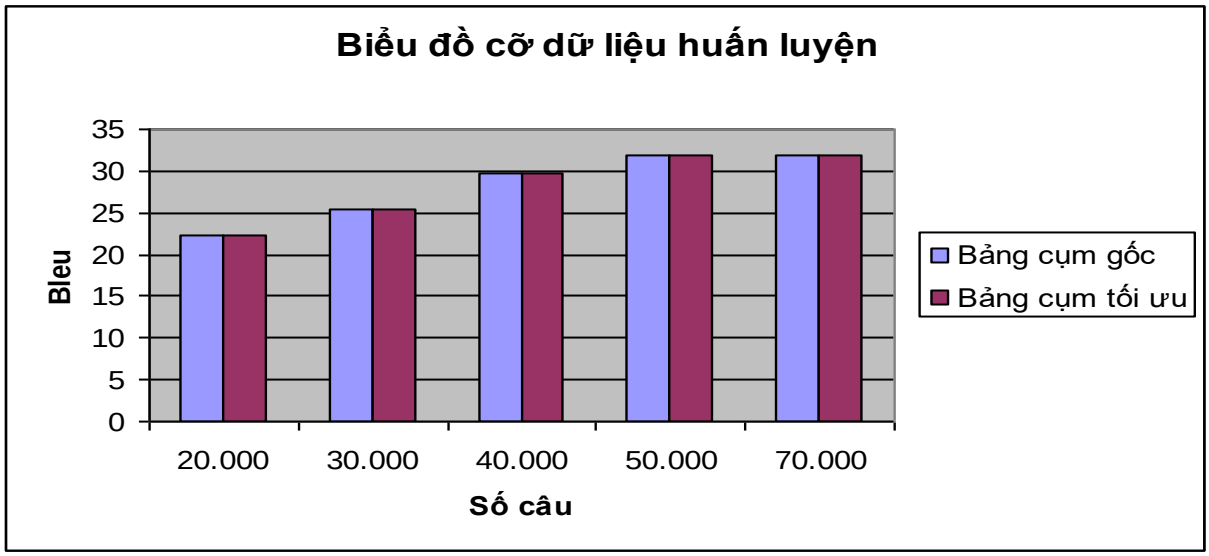
của máy tính. Như vậy, khi kích thước dữ liệu của bảng cụm từ sau khi được nén giảm đi đáng kể so với bảng cụm trước khi được nén. Chúng ta sẽ không cần phải dùng đến một không gian bộ nhớ lớn để lưu bảng cụm từ.

3.3.4 Đánh giá kết quả theo cỡ dữ liệu huấn luyện

Ta thay đổi kích cỡ của dữ liệu huấn luyện lần lượt là 20.000, 30.000,…, 70.000 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 / Điểm Bleu	20.000	30.000	40.000	50.000	70.000
Bảng cụm gốc	22.29	25.39	29.81	31.87	31.95
Bảng cụm tối ưu	22.29	25.39	29.81	31.87	31.95

Bảng 3.7: So sánh điểm BLEU của bảng cụm từ trước và sau khi nén



Biểu đồ 3.2: Biểu đồ so sánh 2.

Bảng 3.7 và biểu đồ 3.2 cho chúng ta thấy rằng, chất lượng dịch của bạng cụm từ và bảng cụm từ tối ưu là như nhau, với cỡ dữ liệu càng lớn thì cho chúng ta

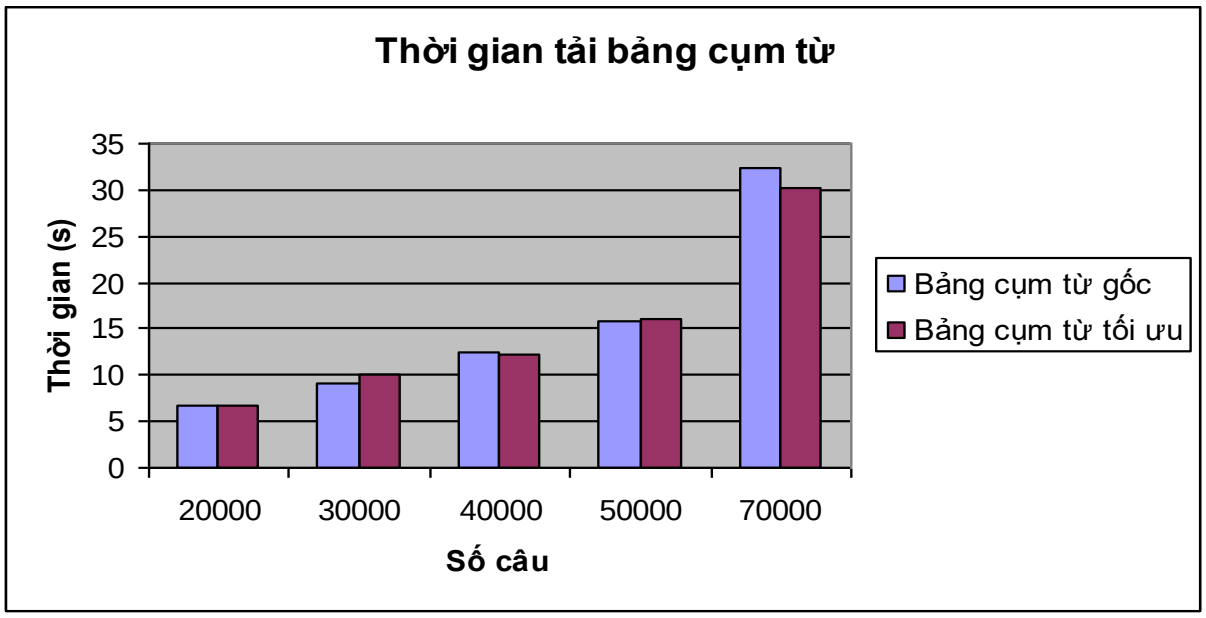
chất lượng dịch càng tốt. Với 20.000 cặp câu của ngữ liệu đầu vào chỉ cho ta điểm BLEU 22.29, đây là số điểm tương đối thấp, nhưng với số lượng 70.000 cặp câu số điểm BLEU là 31.95, đây là điểm khá tốt đối với một hệ dịch máy. Vậy, chúng ta có thể kết luận là, chất lượng của hệ dịch máy phụ thuộc khá nhiều vào kích cỡ dữ liệu được huấn luyện.

3.3.5 Đánh giá kết quả theo thời gian tải bảng cụm từ

Ta thay đổi kích cỡ của dữ liệu huấn luyện lần lượt là 20.000, 30.000,…, 70.000 cặp câu, sau đó thực hiện đánh giá chất lượng dịch dựa vào thời gian tải bảng cụm từ. Thời gian dịch đoạn văn càng nhỏ thì chất lượng dịch càng tốt.

Câu / Thời gian tải	20.000	30.000	40.000	50.000	70.000
Bảng cụm gốc	6.61	9.12	12.47	15.75	32.38
Bảng cụm tối ưu	6.75	9.96	12.29	15.98	30.12

Bảng 3.8: So sánh thời gian tải bảng cụm từ trước và sau khi nén



Biểu đồ 3.3: Biểu đồ so sánh 3

Nhìn vào bảng 3.8 và biểu đồ 3.3 kết quả nhận được là thời gian tải bảng cụm từ lúc tăng lúc giảm. Ta thấy với dữ liệu đầu vào càng lớn thì thời gian sẽ giảm xuống do tấc độ tìm kiếm tăng lên. Trong khi nén thì các cụm từ đã được sắp xếp thành cây do đó tấc độ tìm kiếm sẽ nhanh hơn so với trước khi nén (tìm kiếm tuần tự) . Vậy, chúng ta kết luận thời gian dịch cũng được cải tiến khi nén bảng cụm từ.

KẾT LUẬN

Dịch máy thống kê hiện nay đang rất phát triển trên thế giới, đặc biệt là dịch máy thống kê dựa vào cụm từ. Hướng tiếp cận dịch máy thống kê trên cơ sở cụm từ đã khắc phục được nhiều nhược điểm của dịch máy dựa trên cơ sở từ. Qua ba chương, luận văn đã trình bày về cách tiếp cận dịch máy thông kê dựa trên cụm từ, các phương pháp nén bảng cụm từ và đồng thời áp dụng vào bài toán dịch Anh – Việt. Mặc dù chất lượng dịch chưa cao nhưng khi chúng ta cải tiến mô hình dịch đồng thời huấn luyện với nhiều dữ liệu hơn, chất lượng dịch sẽ được nâng cao rõ rệt. Hơn nữa ta hoàn toàn có thể áp dụng cho chiều dịch Việt – Anh.

Các kết quả đạt được:

- Trình bày về cách tiếp cận dịch máy bằng thống kê trên cơ sở cụm từ.
- Trình bày về quá trình sinh bảng cụm từ trong dịch máy thống kê.
- Áp dụng các phương pháp nén tối ưu bảng cụm từ.
- Xây dựng chương trình thử nghiệm dịch Anh-Việt bằng thống kê dựa trên hệ thống dịch máy Moses.
- Đánh giá kết quả trước và sau khi áp dụng các phương pháp nén bảng cụm từ.

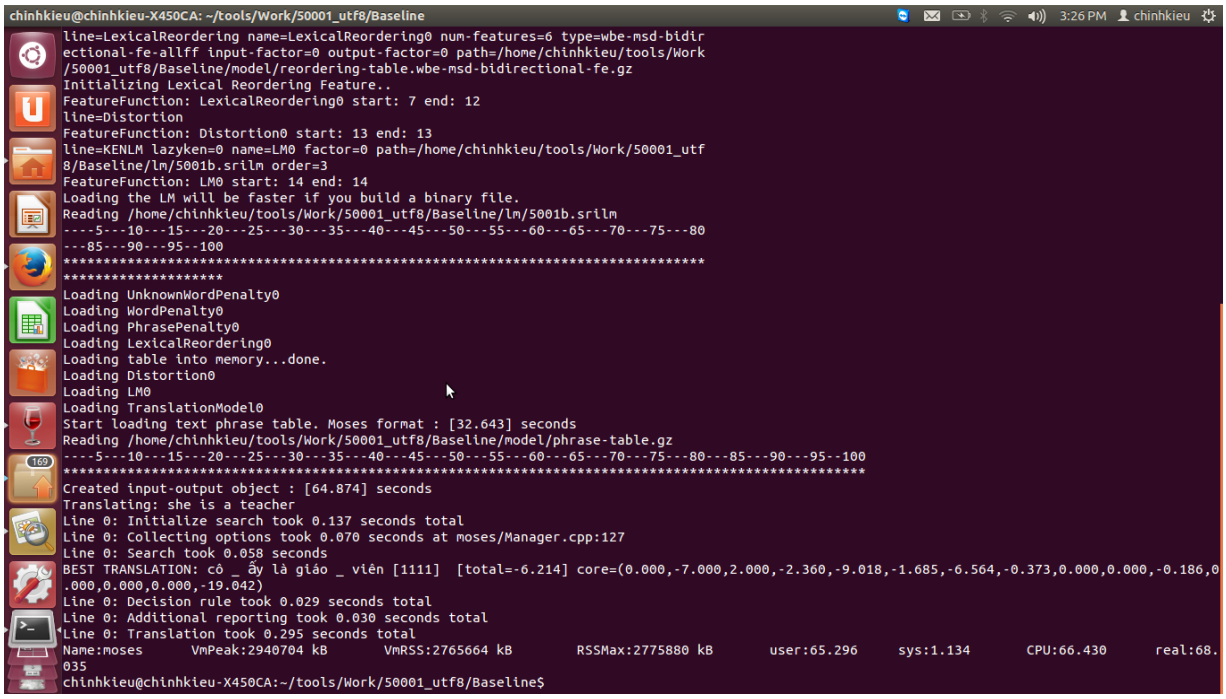
Hướng phát triển:

- Thử nghiệm với dữ liệu đa dạng hơn và lớn hơn.
- Tìm hiểu thêm về các phương pháp nén bảng cụm từ.
- Cải tiến thuật toán giải nén (decoding) để cho hiệu quả hơn.
- Áp dụng cho chiều dịch từ Việt – Anh.

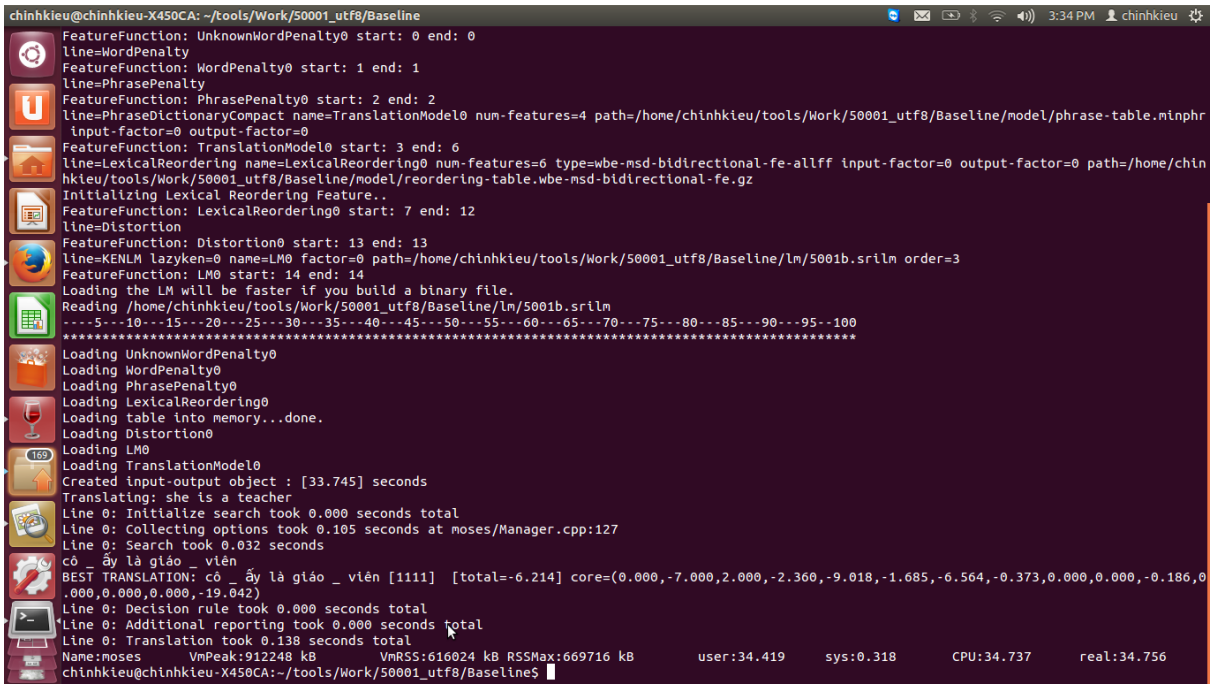
PHỤ LỤC

Luận văn nêu ra một trong những phương pháp quan trọng trong dịch máy thống kê với hệ thống Moses. Với việc tập dữ liệu các ngôn ngữ là rất lớn và việc xử lý với lượng dữ liệu như vậy tương ứng với thời gian dịch sẽ tăng. Do vậy việc tối ưu dữ liệu là hướng phát triển hàng đầu trong dịch máy. Một điều quan trọng nữa là hầu hết các hệ dịch máy đều là online do đó nhu cầu về thời gian ngắn đặt lên hàng đầu. Chúng ta không thể để clients đợi hàng tiếng để dịch một câu từ ngôn ngữ này sang ngôn ngữ khác được. Với việc mã hóa bảng cụm từ, nén bảng cụm từ…. Chúng ta đã có cái nhìn khái quát về một trong những phương pháp phổ biến trong hệ dịch máy. Điều này cũng giải thích nhiều câu hỏi mà nhiều người thường hay đặt ra. (Tại sao một số từ điển trên điện thoại-máy tính chỉ có vài chục Mb mà có thể dịch tương đối tốt !). Đồng thời luận văn cũng trình bày một cách khái quát về việc cài đặt và sử dụng hệ thống dịch máy Moses một trong những hệ thống ổn định và đem lại chất lượng tốt, và cái thư viện công cụ có liên quan như SRILM ,CMPH,BOOST….

1. Kết quả dịch máy đối với câu đơn giản.

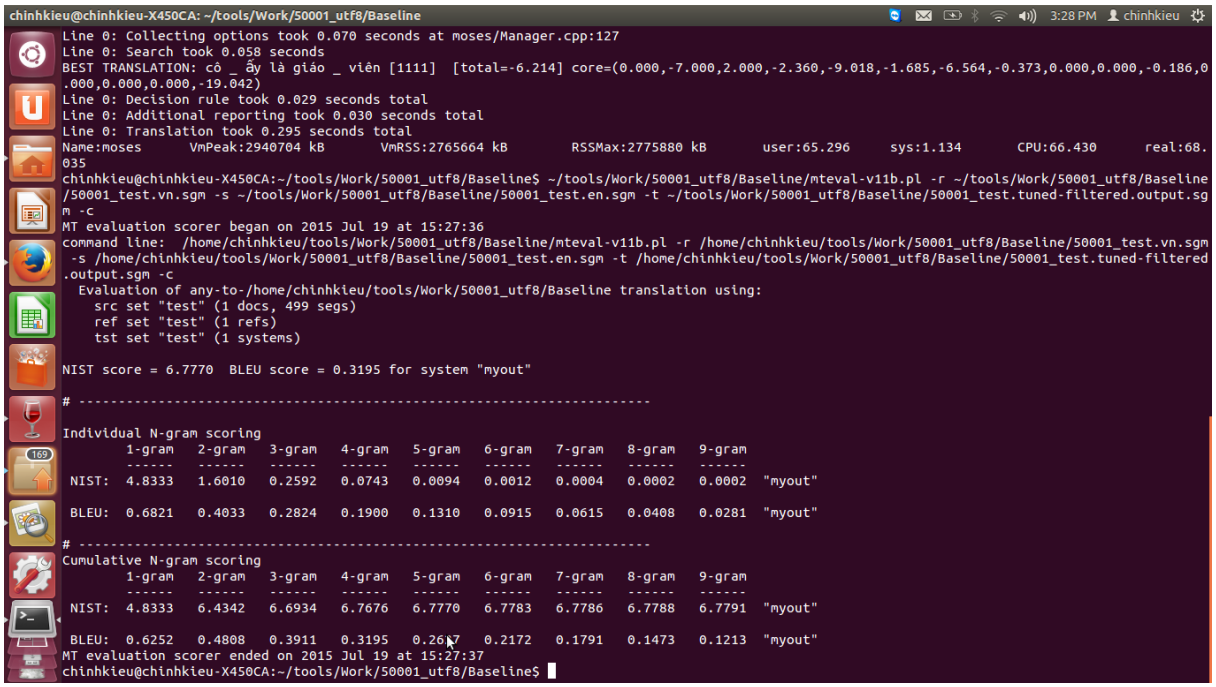


Hình 3.1: Dịch câu đơn giản với bảng cụm từ gốc

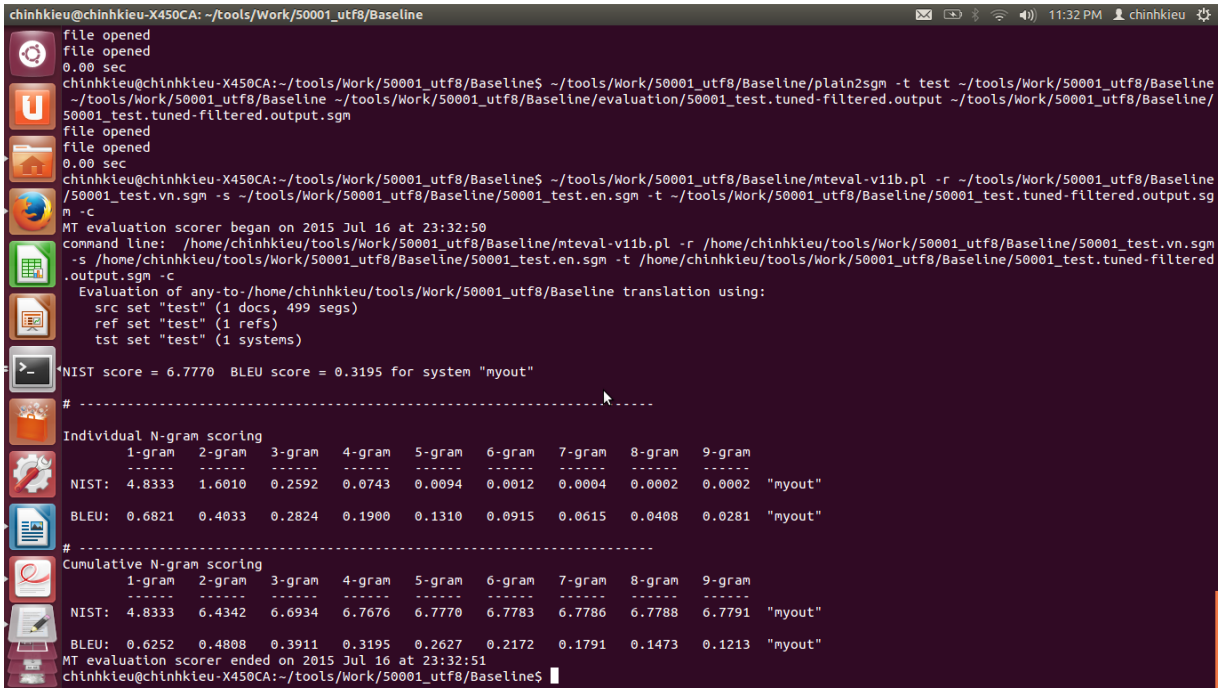


Hình 3.2: Dịch câu đơn giản với bảng cụm tối ưu

2. Kết quả dịch máy đối với bộ dữ liệu.



Hình 3.3: Điểm Bleu bộ dữ liệu bảng cụm từ gốc



Hình 3 .4: Điểm Bleu bộ dữ liệu bảng cụm từ tối ưu

3. Một số công cụ tiền xử lý thường được hay sử dụng trong hệ dịch.

Công cụ tiền xử lý.

Bộ tokenizer: Sử dụng bộ Tokenizer trong bộ ngữ liệu Europarl corpus

do Koehn[9] phát triển

Bộ tách từ: Sử dụng công cụ JvnSegmenter được cung cấp dưới dạng mã nguồn mở do nhóm Phan Xuân Hiếu, Nguyễn Cẩm Tú phát triển sử dụng kỹ thuật Conditional Random Field. Chất lượng bộ tách từ là 94%

Bộ gán nhãn từ loại (Postagger): Sử dụng công cụ CRFTagger được cung cấp dạng mã nguồn mở do nhóm Phan Xuân Hiếu, Nguyễn Cẩm Tú phát triển sử dụng kĩ thuật Conditional Random Field. Chất lượng của bộ gán nhãn này theo tác giả cung cấp lên tới 97%.

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.

Tài liệu tiếng Anh

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

[3] P. Koehn, F.J. Och, and D. Marcu (2003). Statistical phrase table 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)*.

[4] Koehn, P, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst (2007), Moses: Open Source Toolkit for Statistical Machine Translation. ACL 2007, Demonstration Session, Prague, Czech Republic

[5] Philipp Koehn, Franz Josef Och, Daniel Marcu (2003), "*Statistical Bảng cụm từ Translation*", In proceedings of NAACL.

[6]. Brown, P., Cocke, J., Pietra, S. D., Jelinek, J., Lafferty and Roossina, P. (1990), "*A statistical approach to machine translation*", Computational Linguistics, 16(2), pp. 79-85.

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

[8] Marcin Junczys-Dowmunt (2012). Phrasal Rank-Encoding: Exploiting phrase Redundancy and Translational Relations for phrase Table Compression.

[9] Franz Joset Och and Hermann Ney (2002), Discriminative training and maximum entropy models for statistical machine translation, In Processdings of the 40th Anuual Meeting of the ACL, pages 295-302, Philadelphia, PA

[10] Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. (2002), BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the ACL, pages 311-318, Philadelphia, PA