Ứng dụng phương pháp Pointwise vào bài toán tách từ cho tiếng Việt

Lưu Tuấn Anh Yamamoto Kazuhide

Natural Language Processing Laboratory
Department of Electrical Engineering
Nagaoka University of Technology
940-2188, Nagaoka City, Niigata, Japan
{anh, yamamoto}@jnlp.org

***Abstract*— Trong tiếng Việt, dấu cách (space) không được sử dụng như 1 kí hiệu phân tách từ, nó chỉ có ý nghĩa phân tách các âm tiết với nhau. Vì thế, để xử lý tiếng Việt, bài toán tách từ (word segmentation) là 1 trong những bài toán cơ bản và quan trọng bậc nhất. Ngoài tiếng Việt, có khá nhiều các ngôn ngữ châu Á khác cũng cần bước tách từ, ví dụ như : tiếng Nhật, tiếng Trung, tiếng Hàn, … do đó vấn đề này nhận được sự quan tâm rộng rãi và có nhiều hướng tiếp cận khác nhau. Bài viết này sẽ tập trung phân tích hướng tiếp cận pointwise dựa trên máy học SVM : phân loại từng dấu cách một cách độc lập vào 2 loại : SPACE (kí hiệu tách từ) và UNDERSCORE (kí hiệu liên kết 2 âm tiết). Với phương pháp này, chúng tôi đã đạt được độ chính xác 98.2% trong thực nghiệm. Tất cả mã nguồn của nghiên cứu này được ứng dụng để tạo ra công cụ mang tên Đông Du.**

Keywords: xử lý ngôn ngữ tự nhiên, xử lý tiếng Việt, bài toán tách từ, pointwise estimation

I. INTRODUCTION

Trong tiếng Việt, dấu cách không mang ý nghĩa phân tách các từ mà chỉ mang ý nghĩa phân tách các âm tiết với nhau. Ví dụ : từ "đất nước" được tạo ra từ 2 âm tiết "đất" và "nước", cả 2 âm tiết này đều có nghĩa riêng khi đứng độc lập, nhưng khi ghép lại sẽ mang một nghĩa khác. Vì đặc điểm này, bài toán tách từ trở thành 1 bài toán tiền đề cho các ứng dụng xử lý ngôn ngữ tự nhiên khác như phân loại văn bản, tóm tắt văn bản, máy dịch tự động, …

Ngoài tiếng Việt, có khá nhiều các ngôn ngữ khác cũng gặp phải bài toán này, ví dụ như : tiếng Nhật, tiếng Trung, tiếng Hàn, … Mỗi một ngôn ngữ có 1 đặc điểm cú pháp khác nhau, nhưng nhìn chung, hướng tiếp cận chủ đạo ở tất cả các ngôn ngữ này là sử dụng máy học.

Trong phần tiếp theo, chúng tôi sẽ giới thiệu các thuật toán máy học được sử dụng rộng rãi trong bài toán tách từ. Phần 3 sẽ trình bày phương pháp tiếp cận theo hướng pointwise. Phần 4 là kết quả thực nghiệm. Phần 5 sẽ trình bày một số vấn đề mở rộng.

II. SOME MACHINE LENARNING APPROACHS IN WORD SEGMENTATION PROBLEM

Ứng dụng máy học bắt đầu được ứng dụng trong xử lý ngôn ngữ tự nhiên từ khoảng đầu những năm 90, và đã thu được rất nhiều thành tựu quan trọng. Một trong số đó là những ứng dụng cho bài toán tách từ. Trong phần này, chúng tôi sẽ giới thiệu về lịch sử các phương pháp giải quyết bài toán tách từ, và một số ứng dụng với tiếng Việt đã được nghiên cứu.

Phương pháp đầu tiên và thô sơ nhất để giải quyết bài toán tách từ là phương pháp "ghép cực đại". Phương pháp này đơn giản là tạo ra 1 từ điển, và đặt các từ này vào 1 câu sao cho phủ hết được câu đó. Phương pháp này các ưu điểm là rất nhanh, nhưng có rất nhiều hạn chế, ví dụ như độ chính xác thấp, không xử lý được những từ không có trong từ điển.

Tiếp sau phương pháp này, là phương pháp đồ thị hoá. Trong phương pháp này, mỗi một cụm âm tiết mà có trong từ điển, sẽ tương ứng với một đỉnh. Trọng số giữa các đỉnh sẽ được tính toán dựa trên dữ liệu huấn luyện, ví dụ như tần số xuất hiện cạnh nhau, khả năng liên kết giữa từ loại của 2 từ, … Phương pháp này cho kết quả khả quan hơn, nhưng vẫn chưa thể đánh giá là tốt. Ngoài ra, phương pháp này cũng có những nhược điểm như : cần những dữ liệu huấn luyện lớn, mất nhiều thời gian thực thi chương trình, … Một số nghiên cứu như [1], [2] có thể coi như 1 bước phát triển nâng cao của phương pháp này. Nhưng kết quả thu được vẫn còn nhiều hạn chế.

Những phương pháp tiếp theo đều dựa trên ý tưởng coi 1 câu là 1 chuỗi các âm tiết (hoặc kí tự nhưng trong tiếng Nhật hay tiếng Trung). Như thế, bài toán tách từ trở thành bài toán gán nhãn cho 1 chuỗi – dấu cách (hay vị trí giữa 2 kí tự) sẽ được gán 1 trong 2 nhãn, tương ứng với có hay không có dấu phân tách từ ở đó.

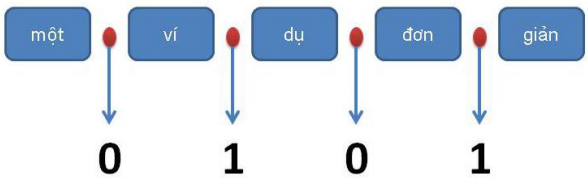


Fig. 1. Ví dụ về sự tương ứng giữa bài toán tách từ và bài toán gán nhãn chuỗi

Những nghiên cứu về ứng dụng máy học trong tách từ tiếng Việt có thể kể đến : phương pháp hidden Markov Model (HMM) [3], conditional random fields (CRF) [4], Maximum entropy (ME) [5]. Nhưng phương pháp này khó có thể trình bày một cách ngắn gọn và dễ hiểu trong 1 bài báo này, vì thế, nếu muốn hiểu rõ hơn, bạn hãy đọc trực tiếp những nghiên cứu đó. Nhìn chung, những phương pháp này cho kết quả khá khả quan (94%~95%) và đơn giản để thực hiện.

Trong những nghiên cứu kể trên, nghiên cứu có độ chính xác cao nhất là của Lê Hồng Phương (97%)[6]. Nghiên cứu này kết hợp các phương pháp máy hữu hạn trạng thái, phân tích dạng chính tắc, và ghép cực đại. Nhược điểm lớn nhất của phương pháp này là không xử lý được những từ mới. Như thế, phương pháp này không sử dụng kĩ thuật học máy. Mã nguồn và công cụ của nghiên cứu này là 1 phần của dự án VLSP [7].

III. POINTWISE ESTIMATION FOR VIETNAMESE WORD SEGMENTATION

A. *Ý tưởng cơ bản*

Pointwise là phương pháp mới được nghiên cứu gần đây. Phương pháp này đang được ứng dụng rộng rãi trong tiếng Nhật và tiếng Trung và thu được những kết quả rất tốt. Ngoài ra, nó còn ứng dụng tốt cho nhiều vấn đề khác nhau trong xử lý ngôn ngữ tự nhiên. Trong tiếng Việt, phương pháp này được ứng dụng trong bài toán thêm dấu cho tiếng Việt không dấu và thu được kết quả khá tốt (gần 95%) [8].

Phương pháp HMM, CRF hay ME có điểm chung là có tham khảo nhãn (hay kết quả) của những nhãn bên cạnh.

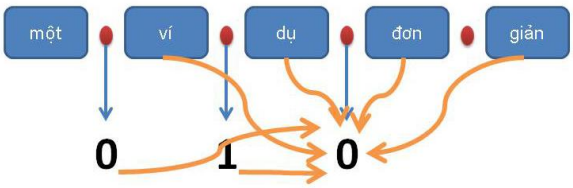


Fig. 2. Ví dụ về việc tham khảo các kết quả trước

Những phương pháp này chỉ thực sự đạt được kết quả tốt khi có một từ điển lớn. Lý do vì những phương pháp này không hiệu quả với những từ mới, và có tham khảo kết quả của các phép gán nhãn trước, nên khi có 1 kết quả sai sẽ kéo theo các kết quả phía sau cũng sai. Hiện tại, từ điển dành cho XLNNTN tiếng Việt lớn nhất chỉ gồm 40,000 từ. Số lượng này là ít nếu so sánh với 150,000 của tiếng Nhật, hay 100,000 của tiếng Trung.

Vì lý do này, các nghiên cứu sử dụng HMM, CRF và ME cho tiếng Việt đều cho kết quả thấp hơn những nghiên cứu tương tự cho các ngôn ngữ khác.

Pointwise là cách tiếp cận nhằm khắc phục nhược điểm của những phương pháp máy học trên. Trong phương pháp pointwise, các nhãn sẽ được đánh giá một cách độc lập, và không tham khảo kết quả của các nhãn trước đó.

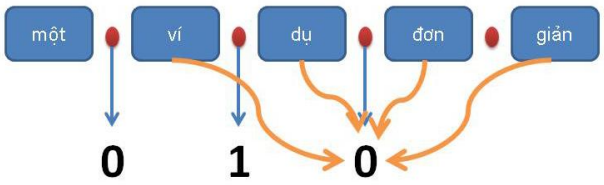


Fig. 3. Ví dụ về việc không tham khảo các nhãn trước

Chính vì việc đánh giá độc lập như thế, mà phương pháp pointwise chỉ cần 1 từ điển vừa phải, và khá hiệu quả khi xác định những từ mới không có trong từ điển. Vì thế, phương pháp pointwise rất phù hợp với những ngôn ngữ không có nhiều dữ liệu như tiếng Việt.

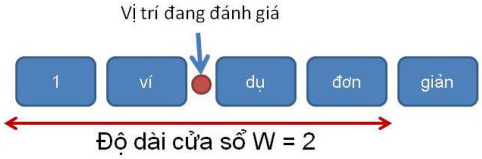
Ngoài ra, vì các vị trí được đánh giá độc lập, các đặc trưng chỉ là thông tin văn bản xung quanh vị trí đó, nên pointwise có thể thực hiện được trên những dữ liệu không đầy đủ. Để thực hiện việc đánh giá các nhãn một cách độc lập trên dữ liệu không đầy đủ, HMM hay CRF cũng có thể thực hiện được, nhưng đòi hỏi rất nhiều thời gian cho quá trình học máy cũng như thực thi trong thực tế [9].

Phương pháp thích hợp nhất để thực hiện việc đánh giá độc lập này là sử dụng Support Vector Machine (SVM). SVM là phương pháp học máy đơn giản nhưng rất hiệu quả cho tập trung vào từng nhãn một cách độc lập, ít bị ảnh hưởng bởi các ví dụ sai trong dữ liệu huấn luyện. Ngoài ra, SVM cũng khá dễ dàng để thực hiện việc chọn lựa đặc trưng (features selection) để giảm kích thước dữ liệu model.

B. *Những đặc trưng được sử dụng*

Phương pháp tiếp cận dạng pointwise sử dụng những thông tin xung quanh vị trí cần đánh giá, và thực hiện một cách độc lập với nhau. Chúng tôi sử dụng 3 dạng đặc trưng cơ bản trong phương pháp pointwise là : n-gram âm tiết, n-gram chủng loại của âm tiết, và đặc trưng từ điển.

- N-gram âm tiết : sử dụng n-gram của những âm tiết xung quanh vị trí đang đánh giá. Ở đây, chúng tôi sử dụng một cửa sổ có độ dài W, và chúng tôi chỉ sử dụng những âm tiết nằm trong cửa sổ này. Với tiếng Việt, có khoảng 70% các từ gồm 2 âm tiết, và 14% các từ gồm 3 âm tiết. Vì lý do này, chúng tôi sẽ sử dụng W là 3. Ngoài ra, n thường là 1 và 2. Trong thực nghiệm, chúng tôi có sử dụng cả n = 3, nhưng kết quả không được cải thiện nhiều, và kích thước file model cũng tăng lên đáng kể.
- N-gram chủng loại của âm tiết : sử dụng chủng loại của các âm tiết trong cửa sổ. Trong nghiên cứu này, chúng tôi định nghĩa 4 chủng loại :
  - Âm tiết viết hoa (U) : những âm tiết tiếng Việt có bắt đầu bằng chữ hoa.
  - Âm tiết viết thường (L) : những âm tiết tiếng Việt chỉ gồm những chữ cái thường.
  - Số (N): gồm các chữ số.
  - Các loại khác (O) : những kí hiệu, tiếng nước ngoài, và những âm tiết không nằm trong 3 loại trên.
- Đặc trưng từ điển : là những từ có trong từ điển. Đặc trưng này sẽ được thuyết minh cụ thể trong ví dụ tiếp theo.



	N-gram âm tiết	N-gram chủng loại âm tiết
1-gram	-2|1 -1|ví 0|dụ 1|đơn	-2|N -1|L 0|L 1|L
2-gram	-2||1 ví -1||ví dụ 0||dụ đơn	-2||N L -1||L L 0||L L

Fig. 4. Ví dụ về N-gram âm tiết và N-gram chủng loại âm tiết với W = 2

Fig. 5. Ví dụ về đặc trưng từ điển

Trong ví dụ trên, từ "ví dụ" có xuất hiện trong từ điển. Ngoài ra, chúng tôi còn định nghĩa vị trí của những đặc trưng từ điển như sau :

- Nằm bên phải (R) : vị trí đánh giá nằm ở đầu bên phải của 1 từ trong từ điển.
- Nằm bên trái (L) : vị trí đánh giá nằm ở đầu bên trái của 1 từ trong từ điển.
- Nằm ở giữa (I) : vị trí đánh giá nằm ở bên trong 1 từ trong từ điển.

Trong ví dụ trong Fig. 4, vị trí đánh giá nằm trong từ "ví dụ", vì thế sẽ được ghi nhận đặc trưng là "I|ví dụ".

C. *Đặc điểm về dữ liệu huấn luyện*

Một ưu điểm lớn khác của phương pháp Pointwise, là không cần dữ liệu huấn luyện đầy đủ.

Dữ liệu huấn luyện đầy đủ là 1 dữ liệu mà tất cả các câu, các từ trong dữ liệu đó đều phải được tách từ xong. Ví dụ *"1 ví_dụ đơn_giản ."* là 1 dữ liệu huấn luyện đầy đủ.

Dữ liệu huấn luyện không đầy đủ không yêu cầu tất cả các từ phải được tách xong. Cùng với câu ví dụ trên, dữ liệu huấn luyện không đầy đủ có thể là "*1 ví_dụ đơn giản .*". Như thế, chỉ từ "*ví_dụ*" là được tách từ, còn từ "*đơn giản*" thì không được tách.

Trong thực tế, những dữ liệu huấn luyện đầy đủ cần rất nhiều thời gian, công sức và tiền bạc. Ngoài ra, độ chính xác của dữ liệu dạng này cũng là 1 vấn đề. Có nhiều câu, với những quan điểm khác nhau của người thực hiện tách từ, có thể được tách theo những cách khác nhau. Ví dụ câu : "*Ông già đi nhanh quá*" có thể được tách thành "*Ông_già đi nhanh quá*" hoặc cũng có thể tách thành "*Ông già_đi nhanh quá*". Điều này sẽ gây khó khăn và mất thời gian cho người thực hiện tách từ.

Với phương pháp dữ liệu không đầy đủ, chỉ những vị trí chắc chắn chính xác mới được tách. Những vị trí khó phán đoán, có thể để lại. Nhờ thế, dữ liệu không đầy đủ thường đơn giản và ít tốn công sức hơn. Ngoài ra, độ chính xác cũng thường cao hơn.

Phương pháp pointwise thực sự mềm dẻo khi vẫn có thể áp dụng tốt trên cả 2 dạng dữ liệu này.

IV. EXPERIMENTS

A. *Dữ liệu*

Như đã phân tích ở trên, thuật toán máy học thích hợp nhất là thuật toán SVM. Trong thực nghiệm này, chúng tôi sử dụng thư viện LIBLINEAR [10] để giải quyết bài toán phân loại. Thư viện này được sử dụng rộng rãi, và có những ưu điểm rất nổi bật như :

- Tốc độ xử lý rất nhanh
- Có thể phân loại những bài toán có từ hàng triệu đến hàng chục triệu đặc trưng
- Yêu cầu cấu hình máy thấp, máy tính cá nhân thông thường cũng có thể hoạt động được.

Trong phần thực nghiệm này, chúng tôi sử dụng dữ liệu từ dự án VLSP. Dữ liệu bao gồm khoảng 2 triệu âm tiết đã tách từ. Nội dung chủ yếu là những tin tức và xã luận trên các báo.

Chúng tôi chia bộ dữ liệu này thành 2 phần. Trong đó dữ liệu sử dụng để huấn luyện khoảng 70% dữ liệu ban đầu. Khoảng 30% dữ liệu còn lại được sử dụng để đánh giá độ chính xác của chương trình.

TABLE I. TRAINING DATA AND TEST DATA

	Training	***Test***
Size	7.7Mb	2.9Mb
#Syllabels	1,404,406	535,600
#Words	1,071,195	410,088

B. *Kết quả*

Để thực hiện so sánh kết quả, chúng tôi sử dụng công cụ vnTokenizer của tác giả Lê Hồng Minh, phiên bản mới nhất là 4.1.1c, ra ngày 04/08/2010[6][7].

Đối với 1 công cụ tách từ, ngoài yêu cầu về độ chính xác, thì tốc độ xử lý và lượng RAM sử dụng cũng là những yếu tố rất quan trọng. Những yếu tố này cho phép xử lý những văn bản cực lớn, và có thể hoạt động tốt trên những máy tính thông thường.

Kết quả thực nghiệm được tổng kết trong bảng sau :

TABLE II. RESULTS OF TWO METHODS

	vnTokenizer	***DongDu***
Accuracy	97.2%	98.2%
Time	194.672 (s)	26.2 (s)
RAM	19.8Mb	15.1Mb

Theo kết quả này, DongDu cho độ chính xác cao hơn vnTokenizer khoảng 1%. Về tốc độ xử lý, DongDu cũng nhanh hơn vnTokenizer khoảng 8 lần. Ngoài ra, DongDu đòi hỏi lượng RAM ít hơn vnTokenizer.

Những thông số này cho thấy, cách tiếp cận pointwise hiệu quả hơn hẳn các thuật toán khác. Và công cụ DongDu cũng mạnh mẽ và nhanh hơn những công cụ hiện tại.

C. *Lựa chọn đặc trưng*

Như đã trình bày ở trên, chúng tôi sử dụng thư viện LIBLINEAR để thực hiện phân loại.

Để giảm kích thước file dữ liệu và tăng tốc độ chương trình, chúng tôi đã thực hiện việc lựa chọn đặc trưng.

Ở đây, chúng tôi sử dụng phương pháp lựa chọn đặc trưng dựa vào L1 regularized logistic regression [11]. Khi thực hiện việc học máy theo phương pháp này, hầu hết trọng số của những đặc trưng sẽ là 0. Và những đặc trưng đó sẽ bị coi là không cần thiết và có thể loại bỏ.

Sau khi thực hiện lựa chọn đặc trưng, số đặc trưng còn lại chỉ bằng khoảng 1/40 so với ban đầu. Tổng kích thước dữ liệu cũng giảm xuống tương ứng với tỉ lệ trên.

V. EXTENSION

Ưu điểm nổi bật của pointwise là tốc độ xử lý nhanh, đơn giản và dễ hiểu với cả những người mới bắt đầu. Ngoài ra, ưu điểm có thể học máy trên những dữ liệu không đầy đủ cũng là 1 ưu điểm rất quan trọng.

Dựa trên ưu điểm này, ta có thể phát triển thêm nhiều hướng nghiên cứu mới.

A. *Tạo dữ liệu mới*

Giả sử hiện tại ta có 1 corpus A với n âm tiết. Ta sẽ chia corpus này thành 10 phần bằng nhau, mỗi phần có n/10 âm tiết.

Đầu tiên, ta tiến hành tách từ thủ công với 1 phần dữ liệu này. Sau khi hoàn thành, ta sẽ dùng dữ liệu này để tiến hành học máy. Sau đó, với máy học này, ta sẽ tiến hành tách từ bằng máy đối với phần dữ liệu thứ 2. Kết quả của phần dữ liệu này sẽ lại được kiểm tra bằng tay một lần nữa. Vì đã được tách bằng máy, nên lần kiểm tra này sẽ tốn ít thời gian hơn nhiều so với lần đầu tiên. Sau đó, ta gộp dữ liệu ở phần 1 và phần 2 lại, tiến hành học máy, và tiếp tục với những phần sau.

Càng về sau, dữ liệu tách từ xong càng lớn, và độ chính xác của máy học càng cao. Những lần kiểm tra lại bằng thủ công sau cũng sẽ tốn ít thời gian hơn trước.

Cách làm này sẽ tiết kiệm được rất nhiều thời gian và cong sức cho người thực hiện tạo dữ liệu.

B. Tách từ dựa trên phân loại 3 lớp

Đến thời điểm này, ta đã bàn đến phương pháp pointwise phân loại 2 lớp : có hoặc không có dấu phân tách từ. Phần này ta sẽ mở rộng thành 3 lớp : 2 lớp cơ sở, và thêm 1 lớp mới, tạm gọi là lớp nghi vấn.

Dựa vào thư viện LIBLINEAR, ta có thể tính được xác suất chính xác của 1 vị trí. Hay nói cách khác, ở 1 vị trí, ta có thể biết vị trí đó có bao nhiêu phần trăm là dấu phân tách từ, bao nhiêu phần trăm là dấu liên kết từ.

Với những vị trí mà sai khác giữa 2 xác suất này cao hơn 1 giá trị t cho trước, ta sẽ coi là đúng, và phân loại vào 2 lớp có hoặc không có dấu tách từ. Với những vị trí nghi vấn, xác suất giữa 2 lớp này sẽ xấp xỉ nhau. Ta sẽ bảo lưu những vị trí này và không tiến hành học máy trên đó. Sau khi học máy xong, ta sẽ đưa ra những vị trí nghi vấn cao nhất (sai khác giữa 2 xác suất trên càng nhỏ thì nghi vấn càng cao) và thực hiện tách từ thủ công. Khi đó, ta sẽ coi đó là vị trí đúng và tiến hành học máy lại.

Ta sẽ tiếp tục quá trình này đến khi nào đạt được kết quả mong muốn.

Phương pháp này còn có cách gọi khác là cách học máy bán giám sát. Tức là có sự can thiệp ở 1 mức độ nào đó với kết quả của máy học. Phương pháp này thường được đánh giá là hiệu quả và thích hợp trong nhiều vấn đề của xử lý ngôn ngữ tự nhiên.

VI. CONCLUSION

Trong bài viết này, chúng tôi đã trình bày thuật toán tách từ dành cho tiếng Việt dựa theo phương pháp tiếp cận pointwise.

Theo kết quả thực nghiệm, phương pháp này cho độ chính xác cao hơn và tốc độ nhanh hơn hẳn những phương pháp đã được nghiên cứu.

Ngoài ra, tính ứng dụng của phương pháp này cũng rất cao khi có thể tiến hành cả trên dữ liệu đầy đủ lẫn dữ liệu không đầy đủ.

Dựa trên phương pháp này, chúng tôi đã phát triển công cụ mở và miễn phí mang tên DongDu. Với tốc độ xử lý nhanh hơn hẳn và yêu cầu ít RAM hơn, DongDu thích hợp hơn trong việc xử lý dữ liệu lớn. Toàn bộ mã nguồn của DongDu được viết bằng ngôn ngữ C++. DongDu có thể được download từ địa chỉ : http://viet.jnlp.org/dongdu .

REFERENCES

[1] Dinh, D., Kiem, H., Toan, N.V.: Vietnamese Word Segmentation. The 6th Natural Language Processing Pacific Rim Symposium (2001), 749--756s

[2] DD Pham, GB Tran, SB Pham : A hybrid approach to Vietnamese word segmentation using part of speech tags, 2009 International Conference on Knowledge,

[3] Nguyen, P.T., Nguyen, V.V., Le, A.C., Vietnamese word segmentation using hidden markov model (2003), International Workshop for Computer, Information, and Communication Technologies on State of the Art and Future Trends of Information techonologies in Korea and Vietnam

[4] Nguyen, C.T., Nguyen, T.K., Phan, X.H., Nguyen, L.M, Ha, Q.T., Vietnamese word segmentation with CRFs and SVMs: An investigation (2006), Proceedings of the 20th PACLIC, pp.215-222

[5] Dinh, D., Vu, T., A maximum entropy approach for vietnamese word segmentation, 2006, Proceedings of 4th RIVF VietNam, pp.12-16.

[6] Le, H.P, Nguyen, T.M.H, Azim Roussanaly, Ho, T.V, A hybrid approach to Word Segmentaion of Vietnamese texts (2008), Language and atutomata theory and applications 2nd international coference, LATA 2008

[7] VLSP project, Vietnamese Language Processing, http://vlsp.vietlp.org

[8] Luu, T.A, Yamamoto, K., A pointwise approach for Vietnamese Diacritics Restoration, IALP 2012

[9] Tsuboi, Y., Kashima, H. Oda, H., Mori, S., Matsumoto, Y., Training conditional random fields using incomplete annotations, 2008, COLING 08 Proceedings of the 22nd International Conference on Computational Linguistics, 897-904

[10] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification", Journal of Machine Learning Research, 9:1871–1874, 2008.

[11] Okanohara, D., Tsujii, J., Learning combination features with L1 Regularization, 2009, NAACL-Short 09, 97-100