**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**ĐỀ TÀI EP03/05**

**Xây dựng chương trình gán nhãn ngữ nghĩa  
 cho từ tiếng Anh bằng phương pháp fTBL**

GVHD: PGS. TS. Đinh Điền

Nhóm thực hiện:

Ngành Khoa Học Máy Tính – Cao học khóa 23

1. Đỗ Đặng Minh - 13 11 015
2. Huỳnh Công Toàn - 13 11 026
3. Hồ Văn Tấn - 13 11 058

*Tp. Hồ Chí Minh, tháng 06 năm 2014*

**Thông tin nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| **MSHV** | **Họ tên** | **Số điện thoại** | **E-mail** |
| 13 11 015 | Đỗ Đặng Minh | 0168-993-5242 | [masterminh219@gmail.com](mailto:masterminh219@gmail.com) |
| 13 11 026 | Huỳnh Công Toàn | 0121-516-1090 | [alex7huynh@gmail.com](mailto:alex7huynh@gmail.com) |
| 13 11 058 | Hồ Văn Tấn | 090-290-9334 | [tanhv90@gmail.com](mailto:tanhv90@gmail.com) |

Mục lục

[I. Giới thiệu 2](#_Toc391022079)

[II. Giải thuật học cải biến 2](#_Toc391022080)

[1) Hướng tiếp cận của Ramshaw và Marcus 3](#_Toc391022081)

[2) Hướng tiếp cận ICA 3](#_Toc391022082)

[3) Các hướng tiếp cận khác 4](#_Toc391022083)

[III. Fast TBL 4](#_Toc391022084)

[1) Sinh luật 4](#_Toc391022085)

[2) Tổng quát 5](#_Toc391022086)

[3) Đánh giá giải thuật 7](#_Toc391022087)

[IV. Gán nhãn ngữ nghĩa bằng Fast TBL 7](#_Toc391022088)

[V. Phụ lục - Nhãn ngữ nghĩa 8](#_Toc391022089)

[1) Nhãn ngữ nghĩa cơ bản cho danh từ 8](#_Toc391022090)

[2) Nhãn ngữ nghĩa cơ bản cho động từ 8](#_Toc391022091)

[3) Nhãn ngữ nghĩa cơ bản cho tính từ 9](#_Toc391022092)

[VI. Tài liệu tham khảo 9](#_Toc391022093)

# Giới thiệu

Giải thuật học cải biến (TBL) đã được sử dụng để giải quyết nhiều vấn đề trong xử lý ngôn ngữ. Nó cũng đạt được hiệu năng rất cao trong các ứng dụng xử lý ngôn ngữ tự nhiên (NLP).

TBL là một trong những giải thuật máy học thành công nhất. Nó rất linh động, dễ mở rộng và ứng dụng nhiều trong lĩnh vực NLP, như gán nhãn từ loại (Bril, 1995), phân đoạn danh ngữ (Ramshaw và Marcus, 1999), phân tích cú pháp (Brill, 1996), kiểm lỗi chính tả (Mangu và Brill, 1997), khử nhập nhằng ngữ nghĩa, giải quyết nhập nhằng gắn giới ngữ,…

Tuy nhiên nó cũng có một hạn chế rất lớn: thời gian huấn luyện rất dài, đặc biệt là với những bộ ngữ liệu lớn - thường dùng trong NLP. Ví dụ như một bộ gán nhãn từ loại cài đặt bằng TBL phải tốn đến 38 tiếng để huấn luyện một bộ ngữ liệu 1 triệu từ. Hạn chế này càng thể hiện rõ trong các giải thuật ứng dụng TBL và có sử dụng vòng lặp.

Trong báo cáo này, nhóm sẽ trình bày về giải thuật TBL được Eric Brill đề xuất năm 1993 và giải thuật Fast TBL được Grace Ngai và Radu Florian đề xuất năm 2001 nằm cải tiến TBL. Sau đó nhóm sẽ trình bày phương pháp gán nhãn ngữ nghĩa cho từ tiếng Anh bằng phương pháp Fast TBL.

# Giải thuật học cải biến

Ý tưởng chính của TBL là học một danh sách luật có thứ tự để cải tiến trạng thái hiện tại của tập huấn luyện. Việc khởi gán dựa trên thống kê đơn giản, sau đó áp dụng các luật để sửa các lỗi cho đến khi tập huấn luyện không còn cải tiến được nữa

Các ký hiệu và định nghĩa sau sẽ được dùng trong báo cáo:

* *S*: không gian mẫu.
* *C*: tập phân lớp mẫu.
* *C[s]*: phân lớp gán với mẫu *s*; *T[s]*: phân lớp thật sự của *s*.
* *p*: vị từ do *S* định nghĩa.
* Một luật *r* được định nghĩa là một vị từ - cặp phân lớp *(p, t)*, với được gọi là đích (target) của *r*.
* *R*: tập luật.
* Nếu *r* = *(p, t)* thì biểu thị thay cho *p*, thay cho *t*.
* Một luật *r* = *(, )* được áp dụng vào mẫu *s* nếu và ; *r(s)* là mẫu kết quả.

Để sử dụng TBL, chúng ta cần có:

* Khởi gán lớp. Chúng ta có thể gán lớp phổ biến cho tập huấn luyện hoặc có thể dùng kết quả từ một bộ phân lớp.
* Tập khung luật. Các khung luật này sẽ quyết định vị từ mà các luật sẽ kiểm tra; chúng có tác động lớn vào kết quả của giải thuật.
* Hàm mục tiêu *f*. Hàm mục tiêu của TBL sẽ tối ưu trực tiếp hàm đánh giá. Ví dụ như:

trong đó

Chúng ta chỉ quan tâm đến các luật có tác dụng tích cực, nên từ đây có hai hướng tiếp cận:

* Sinh ra các luật (sử dụng bộ luật mẫu) sửa ít nhất một lỗi () bằng cách kiểm tra tất cả các mẫu chưa chính xác ();
* Tính giá trị cho mỗi luật *r* như , lưu lại giá trị cao nhất; khi tính , bỏ qua đến luật kế tiếp khi:

Khi không còn luật nào cải tiến trạng thái hiện tại của tập huấn luyện hơn một ngưỡng cho trước thì giai đoạn huấn luyện kết thúc. Các luật sau đó sẽ được áp dụng lần lượt theo tập đánh giá theo đúng thứ tự chúng được học. Sau khi áp dụng tất cả các luật, chúng ta sẽ có được phân lớp cuối cùng.

1. Hướng tiếp cận của Ramshaw và Marcus

Một trong những bước tốn nhiều thời gian nhất của TBL là bước cập nhật. Bản chất của giải thuật cần mỗi luật mới chọn áp dụng vào ngữ liệu và trạng thái hiện tại của ngữ liệu cập nhật lại trước khi học luật kế tiếp.

Ramshaw và Marcus (1994) cố gắng giảm thời gian huấn luyện của giải thuật bằng cách cải tiến quy trình cập nhật. Phương pháp của họ là mỗi luật lưu một con trỏ đến các mẫu nó áp dụng, và mỗi mẫu giữ một danh sách các con trỏ đến các luật áp dụng vào nó. Nhờ các danh sách này mà hệ thống dễ dàng:

* Xác định được vị trí mà luật tốt nhất được áp dụng vào ngữ liệu.
* Cập nhật điểm của tất cả các luật ảnh hưởng bởi việc thay đổi trạng thái kho ngữ liệu.

Hai quy trình trên được thực hiện nhiều lần trong quy trình cập nhật, kết quả chỉnh sửa cũng giảm đáng kể thời gian thực thi.

Hạn chế của phương pháp này là cần bộ nhớ quá lớn. Ví dụ như kho ngữ liệu có 200,000 từ có khoảng 2 triệu luật ở mỗi vòng lặp. Không gian bộ nhớ cần thêm để lưu trữ danh sách con trỏ cho các luật này là khoảng 450 MB, một yêu cầu khá lớn để thêm vào hệ thống.

1. Hướng tiếp cận ICA

Hệ thống ICA (Hepple, 2000) giảm thời gian thực thi đáng kể nhưng phần nào hiệu năng cũng giảm theo.

Hệ thống ICA không cho phép vòng lặp thực hiện trên các luật đã học, bằng cách đưa ra hai giả định sau:

* Mẫu độc lập - một thay đổi trạng thái trong một mẫu (như thay đổi từ loại của một từ) không thay đổi ngữ cảnh của các mẫu lân cận. Trường hợp này thường xảy ra khi giải quyết nhập nhằng gắn giới ngữ. Ngay cả khi gán nhãn từ loại, cũng có thể dùng giả định này nếu luật không áp dụng thường xuyên.
* Luật cam kết - có không quá một thay đổi trạng thái trên mỗi mẫu. Điều này cũng tương đương với một danh sách quyết định (Rivest, 1987), trong đó một mẫu chỉ được chỉnh sửa bởi luật đầu tiên áp dụng vào. Tổng quát hơn, giả định này giải quyết các vấn đề khi độ chính xác khởi điểm cao và thay đổi trạng thái ít xảy ra.

Hệ thống ICA được thiết kế và kiểm tra trên bài toán gán nhãn từ loại, giảm đáng kể thời gian huấn luyện trong khi độ chính xác chỉ giảm rất ít.

1. Các hướng tiếp cận khác

Samuel (1998) đề xuất một cách tiếp cận Monte Carlo cho TBL, trong đó chỉ một phần các luật được chọn ngẫu nhiên cho việc ước lượng ở mỗi vòng lặp. Hệ thống µ-TBL mô tả trong Lager (1999) cũng cố gắng giảm thời gian huấn luyện bằng cài đặt Prolog và phương pháp “lazy learning”.

# Fast TBL

Các tiếp cận trình bày ở đây cũng dựa trên nền tảng của Ramshaw và Marcus (1994): thay vì mỗi lần đều tạo ra luật, thì chúng được lưu vào bộ nhớ cùng với hai giá trị và .

Các ký hiệu sau sẽ được dùng trong suốt phần này:

* và và - các mẫu mà luật áp dụng vào thay đổi chúng thành phân lớp chính xác; vì vậy .
* và và - các mẫu mà luật áp dụng vào thay đổi phân lớp từ chính xác thành không chính xác; vì vậy .

Cho một luật vừa mới học được *b* sẽ được áp dụng vào *S*, mục tiêu là xác định các luật *r* sao cho ít nhất một trong các tập hợp được chỉnh sửa bằng cách áp dụng luật *b*. Rõ ràng là khi cả hai tập hợp không bị chỉnh sửa khi áp dụng luật *b* thì giá trị của hàm mục tiêu cho luật *r* vẫn không đổi.

Phần trình bày khá phức tạp do trong nhiều bài toán NLP, các mẫu không độc lập. Ví dụ như, trong bài toán gán nhãn từ loại, một mẫu phụ thuộc vào 2 mẫu trước và sau của nó. Cho *V(s)* là vùng lân cận của một mẫu - tập các mẫu mà mẫu *s* có thể phụ thuộc vào phân lớp của chúng (quy ước ); nếu mẫu độc lập thì .

1. Sinh luật

Cho *s* là một mẫu mà luật phù hợp nhất *b* áp dụng vào (với ). Chúng ta cần xác định các luật *r* bị ảnh hưởng bởi thay đổi . Cho *r* là luật như vậy, thì *f(r)* cần được cập nhật nếu và chỉ nếu tồn tại ít nhất một mẫu như sau:

|  |  |
| --- | --- |
| và hoặc | (1) |
| và hoặc | (2) |
| và hoặc | (3) |
| và | (4) |

Mỗi một điều kiện trên tương ứng với một lần cập nhật cụ thể của hoặc . Điểm chính ở đây là: khi kiểm tra tác động của luật *b* vào mẫu *s*, chỉ mẫu trong tập V(*s*) cần được kiểm tra. Bất kỳ mẫu nào không nằm trong tập có thể bỏ qua vì .

Cho là một mẫu trong vùng lân cận của *s*. Có 2 trường hợp cần kiểm tra - một trường hợp là *b* áp dụng vào , trường hợp còn lại là *b* không áp dụng vào .

**Trường hợp I:** (*b* không thay đổi phân lớp của mẫu ). Chúng ta lưu ý rằng điều kiện và tương đương với

|  |  |
| --- | --- |
|  | (5) |

và công thức và tương đương với

|  |  |
| --- | --- |
|  | (6) |

Các công thức này cho chúng ta một phương pháp sinh ra các luật *r* ảnh hưởng bởi thay đổi :

* Sinh ra tất cả vị từ p (sử dụng khung vị từ) đúng trên mẫu .
* Nếu thì

Nếu thì giảm , trong đó *r* là luật tạo ra bởi vị từ *p* với đích ;

Ngược lại

Nếu thì đối với tất cả các luật *r* có vị từ là *p3* và sẽ giảm ;

Giải thuật sinh ra các luật *r* cần đếm số *good* (công thức (3)) hoặc số *bad* (công thức (4)) tăng (theo công thức (1) hoặc (2)), bằng cách hoán đổi trạng thái và , và đảm bảo rằng thêm tất cả các luật có thể sinh ra (chỉ đối với (3)).

**Trường hợp II:**  (*b* thay đổi phân lớp của mẫu ). Trong trường hợp này, công thức (5) biến đổi thành:

|  |  |
| --- | --- |
|  | (7) |

Trong trường hợp (2) thì đơn giản hơn. Dễ thấy rằng và hàm ý rằng thật vậy, điều kiện cần để mẫu nằm trong tập là được phân lớp chính xác, . Vì nên và . Điều kiện (3) vì vậy tương đương với:

|  |  |
| --- | --- |
|  | (8) |

Giải thuật được chỉnh sửa để thay thế việc kiểm tra và kiểm tra trong công thức (1) và loại bỏ việc kiểm tra đồng thời trong trường hợp (2). Công thức dùng để sinh luật *r* có số đếm tăng (như phương trình (3) và (4)) cũng được tìm tương tự như trường hợp I.

1. Tổng quát

Trong giải thuật, chúng ta giả định rằng tất cả các luật đều có ít nhất một kết quả đầu ra tích cực ( được lưu lại, và điểm số được tính.

|  |
| --- |
| Với tất cả mẫu *s* thỏa mãn , sinh ra tất cả các luật *r* có thể sửa được phân lớp của *s*; tăng .  Với tất cả mẫu *s* thỏa mãn , sinh ra tất cả vị từ *p* với ; với mỗi luật r mà và thì tăng .  **1**: Tìm luật  Nếu (*f(b)* < ngưỡng hoặc ngữ liệu đã học xong) thì ngừng.  Với mỗi vị từ *p*, dặt *R(p)* là luật có vị từ là *p* ().  Với mỗi mẫu *s*, và và :  Nếu thì   * Với mỗi vị từ *p* và   + Nếu thì     - Nếu thì giảm , trong đó  là luật tạo ra bởi vị từ *p* với đích ;   + Ngược lại     - Nếu thì đối với tất cả các luật mà sẽ giảm ; * Với mỗi vị từ *p* mà   + Nếu thì     - Nếu thì tăng , trong đó  ;   + Ngược lại     - Nếu thì đối với tất cả các luật mà sẽ tăng ;   Ngược lại   * Với mỗi vị từ *p* và   + Nếu thì     - Nếu thì giảm , trong đó ;   + Ngược lại     - Với tất cả các luật mà , giảm ; * Với mỗi vị từ *p* mà   + Nếu thì     - Nếu thì tăng , trong đó ;   + Ngược lại     - Với tất cả các luật mà , tăng ;   Lặp lại từ bước **1** |

**Hình 1:** Giải thuật Fast TBL

Vì vậy, ở đầu giải thuật, tất cả các luật sửa ít nhất một phân lớp sai cần được sinh ra. Điểm *bad* của các luật cũng được tính: ở mỗi vị trí có phân lớp chính xác, các luật thay đổi phân lớp được tạo ra, như trong trường hợp 4, điểm *bad* được tăng lên. Lưu ý rằng khi điểm *bad* được tính chỉ có những luật có điểm *good* dương mới được dùng để đánh giá. Điều này ngăn việc sinh ra các luật vô ích và tiết kiệm thời gian tích toán.

Số luật kiểm tra được giữ ở mức tối thiểu. Theo cách sinh luật, hầu hết các luật cần phải chỉnh sửa một trong các điểm của nó. Ngoài ra, cần có thêm một phần không gian để truy xuất đến tất cả các luật với vị từ cho trước; phần này nhỏ hơn phần biểu diễn luật. Ví dụ như trong bài toàn phân đoạn văn bản, chỉ cần thêm khoảng 30MB, trong khi cách tiếp cận của Ramshaw và Marcus (1994) cần đến 450MB.

1. Đánh giá giải thuật

Như đã đề cập trước, giải thuật ban đầu còn thiếu sót khiến nó chạy rất chậm, đặc biệt là khi học các luật và điểm số của chúng giảm. Khi luật tốt nhất có điểm cao, nó sẽ không vào đoạn cuối của phần tính điểm. Các luật trong đoạn cuối sẽ bỏ qua khi điểm *bad* được tính do điểm *good* của chúng đủ nhỏ để loại bỏ chúng đi. Tuy nhiên khi luật tốt nhất nằm trong đoạn cuối, nhiều luật có điểm số tương tự không được bỏ qua khi tính điểm *bad*, dẫn tới thời gian thực thi lâu hơn ở mỗi vòng lặp.

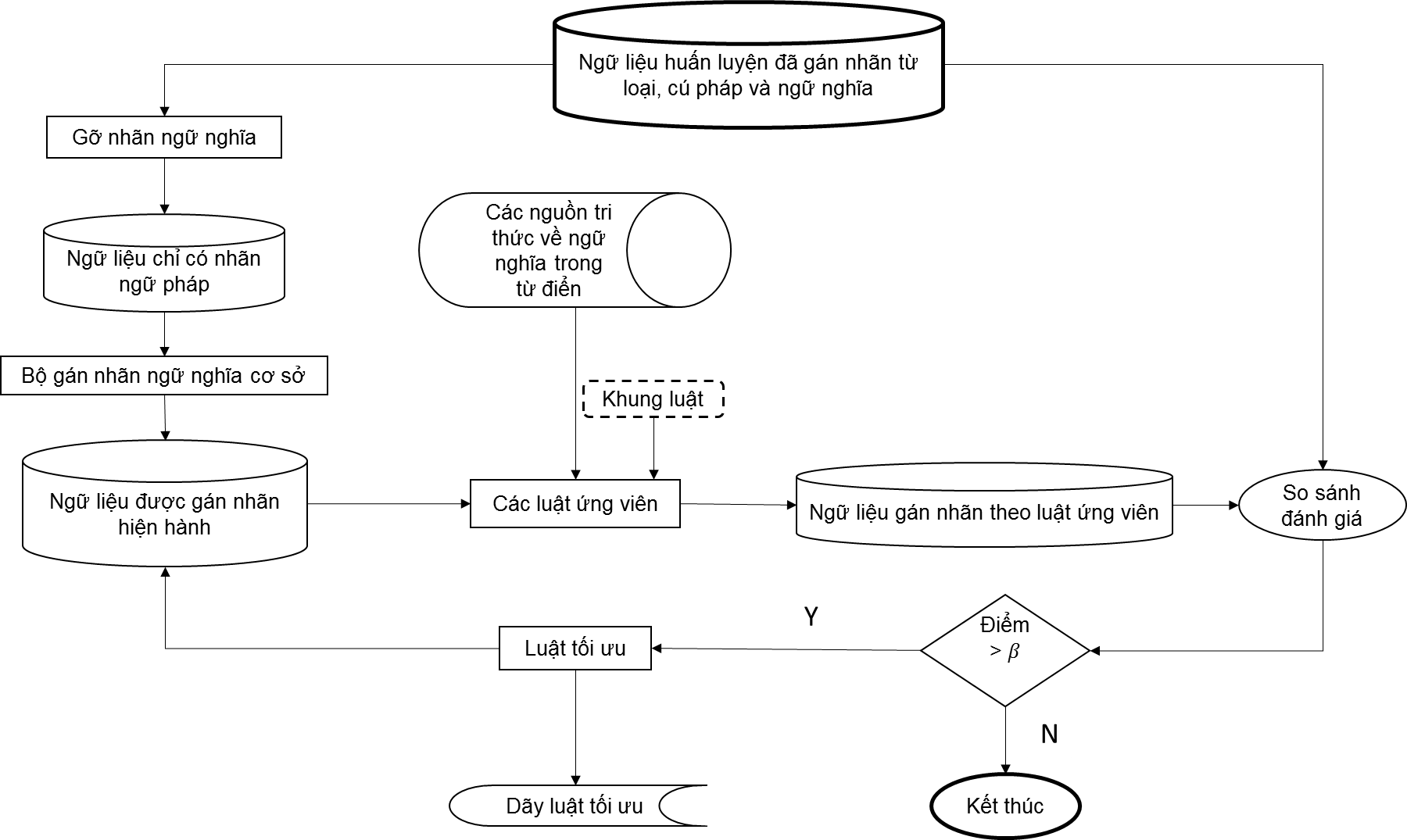
Giải thuật Fast TBL không gặp phải vấn đề này, vì điểm số cập nhật (thay vì phải tính lại) ở mỗi vòng lặp, chỉ các mẫu ảnh hưởng bởi luật học gần nhất mới được cập nhật. Vì số lượng mẫu sẽ giảm dần trong quá trình học, giải thuật này sẽ tăng tốc đáng kể vào cuối giai đoạn huấn luyện. Khi số lượng luật có điểm thấp nhiều hơn đáng kể số luật điểm cao thì thời gian chạy giảm đáng kể.

Kích thước tập huấn luyện cũng ảnh hưởng đến giải thuật. Do dữ liệu lớn thì việc tính điểm càng lâu, nên giải thuật Fast TBL cải tiến thời gian chạy nhanh hơn đáng kể so với giải thuật ban đầu.

# Gán nhãn ngữ nghĩa bằng Fast TBL

Bài toán khử nhập nhằng nghĩa của từ được đưa về bài toán gán nhãn ngữ nghĩa. Một từ nếu có đa nghĩa sẽ được xác định ngay nếu biết được nhãn ngữ nghĩa của nó. Ví dụ như câu “I enter the bank”, danh từ “bank” có nghĩa là “ngân hàng” nếu gán nhãn “HOU” hoặc có nghĩa “bờ sông” nếu có nhãn “NAT”.

Bài toán gán nhãn ngữ nghĩa cho từ tiếng Anh yêu cầu đầu vào là ngữ liệu đã được gán nhãn từ loại và cú pháp chính xác.



**Hình 2:** Sơ đồ huấn luyện bộ gán nhãn ngữ nghĩa từ tiếng Anh trong Fast TBL

**Gán nhãn cơ sở (baseline):**

Chúng ta tiến hành gán nhãn cơ sở dựa trên thông tin về tần suất xuất hiện của nhãn *c* đối với những từ *w* mang từ loại *p* trong bộ ngữ liệu huấn luyện: . Từ loại *p* là từ loại chính xác của từ *w* được xác định sau khi qua tầng gán nhãn từ loại. Đối với từ mới (không có trong ngữ liệu huấn luyện), thông tin về tần suất nhãn ngữ nghĩa này được lấy từ nguồn tri thức WordNet. Cách gán nhãn cơ sở như trên đã thể hiện việc sử dụng nguồn tri thức về từ loại, tri thức về chủ đề và tri thức về tuần suất ngữ nghĩa.

**Khung luật (template):**

Các khung luật sẽ sử dụng các nhân tố từ các nguồn tri thức có liên quan đếnviệc khử nhập nhằng ngữ nghĩa (ngoài những nguồn tri thức đã được sử dụng trong phần gán nhãn cơ sở ở trên), cụ thể như sau:

* Sử dụng tri thức về ràng buộc ngữ nghĩa và quan hệ cú pháp:
* Sử dụng tri thức về ngôn từ:

**Dãy luật tối ưu:**

Sau khi được huấn luyện, hệ thống này sẽ rút ra được dãy luật tối ưu có dạng như sau:

1. ()
2. ()
3. ()
4. ()
5. ()

Các nhãn ngữ nghĩa được gán nói trên cho mỗi từ phải nằm trong danh sách bộ nhãn ngữ nghĩa có thể có của từ đó với từ loại xác định của nó. Kết quả gán nhãn có thể được đánh giá bằng cách so sánh với bộ ngữ liệu vàng như EVC.

# Phụ lục - Nhãn ngữ nghĩa

Phụ lục này chứa hệ thống nhãn ngữ nghĩa chung cho cả từ tiếng Anh và từ tiếng Việt được dùng trong luận án tiến sĩ ngôn ngữ của thầy Đinh Điền. Ngoài ra còn có một số hệ thống nhãn ngữ nghĩa khá phổ biến hiện nay như LLOCE, WordNet, CoreLex,… nhưng không được trình bày ở đây.

Do nội dung của báo cáo này tập trung vào phần gán nhãn ngữ nghĩa cho từ tiếng Anh bằng phương pháp Fast TBL nên nhóm tham khảo chủ yếu vào hệ thống nhãn ngữ nghĩa dưới đây.

1. Nhãn ngữ nghĩa cơ bản cho danh từ
2. Nhãn ngữ nghĩa cơ bản cho động từ
3. Nhãn ngữ nghĩa cơ bản cho tính từ

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Nhãn** | **Mô tả** | **Ý nghĩa** |
| 1 | ACOL | color | Các tính từ về màu sắc |
| 2 | ASIZ | size | Các tính từ về kích thước |
| 3 | ATME | time | Các tính từ về thời gian |
| 4 | ASPC | space | Các tính từ về không gian |
| 5 | ASTR | strength | Các tính từ về sức mạnh |
| 6 | ADEG | degree | Các tính từ về mức độ |
| 7 | AFEA | feature | Các tính từ về đặc điểm, nội dung |
| 8 | AREF | referent | Các tính từ bổ nghĩa sở chỉ |
| 9 | AREL | relation | Các tính từ quan hệ |

# Tài liệu tham khảo

Tiếng Anh:

1. **Grace Ngai, Radu Florian**, *Transformation-based Learning in the Fast Lane*, 2001

Tiếng Việt:

1. **Đinh Điền**, *Xử lý ngôn ngữ tự nhiên*, Nxb Đại học Quốc gia Thành phố Hồ Chí Minh, 2006

Website:

1. <http://www.cs.jhu.edu/~rflorian/fntbl>