HƯỚNG DẪN CHUYỂN ĐỔI TỰ ĐỘNG CÚ PHÁP THÀNH TỐ SANG CÚ PHÁP PHỤ THUỘC TRÊN KHO NGỮ LIỆU NIIVTB

Nguyễn Vương Thịnh

Lưu Hoàng Sơn

{18520367, 18521348}@gm.uit.edu.vn

Đại học Quốc gia TP. Hồ Chí Minh

Trường Đại học Công nghệ Thông tin

Khoa Khoa học Máy tính

Tháng 12, 2021

Mục lục

1 Giới thiệu

Đây là tài liệu hướng dẫn chuyển đổi tự động cho kho ngữ liệu cú pháp thành tố NIIVTB[1]. Nó thể hiện cách thức làm thế nào để chuyển tự động từ một cú pháp thành tố sang cú pháp phụ thuộc.

Tài liệu gồm các phần như sau: phần 2 là phương pháp tìm head của các ngữ trong tiếng Việt, phần 3 sẽ trình bài về việc dán nhãn tự động cho cú pháp phụ thuộc tiếng Việt, phần 4 thể hiện cách khử nhãn NULL và phần 5 sẽ nói về hình thức lưu trữ cây cú pháp phụ thuộc.

2 Xác định head

2.1 Bộ luật tìm head

Chúng tôi sử dụng bộ luật tìm head cho kho ngữ liệu Vietnamese Treebank[2] được trình bày trong [3]. Vì bộ nhãn của Vietnamese Treebank (phụ lục A.1) khác với bộ nhãn NIIVTB (phụ lục A.2), nên chúng tôi thực hiện một vài thay đổi để phù hợp với kho ngữ liệu NIIVTB (bảng 2.1).

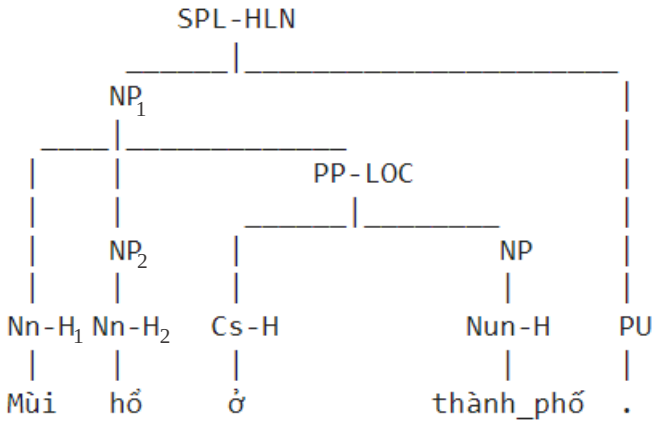
Nhãn	Chiều tìm kiếm	Danh sách ưu tiên
S	→	-H; VP; -PRD; S|SQ|SPL; ADJP; NP; *
SQ	→	-H; VP; QVP; SQ; S; ADJP; NP; QPP; *
SPL	→	-H; VP; SPL; ADJP; NP; *
SBAR	→	-H; VP; S|SQ|SPL; SBAR; ADJP; NP; *
NP	→	-H; NP; Nc|Ncs|Nu|Nun|Nt|Nq|Num|Nw|Nr|Nn; Pd|Pp; VP; *
VP	→	-H; VP; Ve|Vc|D|Vcp|Vv; An|Aa; ADJP; Nc|Ncs|Nu|Nun|Nt|Nq|Num|Nw|Nr|Nn; NP; SBAR; S; R; RP; PP; *
ADJP	→	-H; ADJP; An|Aa; Nc|Ncs|Nu|Nun|Nt|Nq|Num|Nw|Nr|Nn; S; *
RP	←	-H; RP; R; NP; *
PP	→	-H; PP; Cs; VP; SBAR; ADJP; QP; *
QP	→	-H; QP; Nq|Num|Nw; *
MDP	→	-H; MDP; Cs; An|Aa; Pd|Pp; R;*
QNP	→	-H; QNP; NP; Nc|Ncs|Nu|Nun|Nt|Nq|Num|Nw|Nr|Nn; Pd|Pp; *
QADJP	→	-H; QADJP; An|Aa; Nc|Ncs|Nu|Nun|Nt|Nq|Num|Nw|Nr|Nn; Ve|Vc|D|Vcp|Vv; Pd|Pp; *
QRP	→	-H; QRP; Pd|Pp; Cs; *
QPP	→	-H; QPP; Cs; Pd|Pp; *
UCP	→	-H; *
CONJP	→	-H; *

Bảng 2.1 Bộ luật tìm head cho NIIVTB

Trong bảng 2.1, cột đầu tiên là nhãn các loại ngữ, cột thứ hai chỉ chiều tìm kiếm ("→" là từ trái sang phải và "←" là từ phải sang trái). Cột cuối cùng là danh sách ưu tiên của các loại ngữ và độ ưu tiên giảm dần từ trái sang phải. Ngoài ra, các ký hiệu -**<tên nhãn chức năng>** như -**H**, -**PRD** chấp nhận bất kỳ loại nhãn nào miễn được gắn nhãn chức năng chỉ định, ký hiệu | thể hiện mức độ ưu tiên như nhau, và ký hiệu * thể hiện bất kỳ nhãn nào cũng đều thỏa mãn.

Áp dụng bộ luật tìm head không chỉ tìm ra các head của các ngữ mà còn tìm thấy headword của các ngữ đó sau khi khám phá hết tất cả các node trong cây cú pháp thành tố. Ngoài ra, các ngữ còn lại không phải là head sẽ là các dependent.

Ví dụ tìm head và headword của các ngữ trong cây cú pháp thành tố sau:



Hình 2.1 Cây cú pháp thành tố

– Xét mức 0 của cây thành tố gồm node **SPL-HLN**:
+ Node **SPL-HLN** gồm hai node con là **NP** và **PU**.
+ Nhận thấy nhãn **SPL** nằm ở hàng thứ 3 trong bộ luật, thực hiện tìm kiếm từ trái sang phải trên danh sách node con gồm [**NP, PU**], theo danh sách ưu tiên thì **NP** có độ ưu tiên cao hơn nên **NP** sẽ là head cho node **SPL-HLN**.
– Tiếp tục thực hiện các bước tương tự như mức 0 ở các mức 1, 2, 3, và 4. Ta sẽ tìm được tất cả head và headword của các ngữ:

SPL-HLN → NP_1 → $Nn\text{-}H_1$ → Mùi

PU → .

NP_1 → NP_2 → $Nn\text{-}H_2$ → hổ

PP-LOC → Cs-H → ở

NP_3 → Nun-H → thành_phố

2.2 Xác định head của liên ngữ

2.2.1 Phát hiện trường hợp liên ngữ

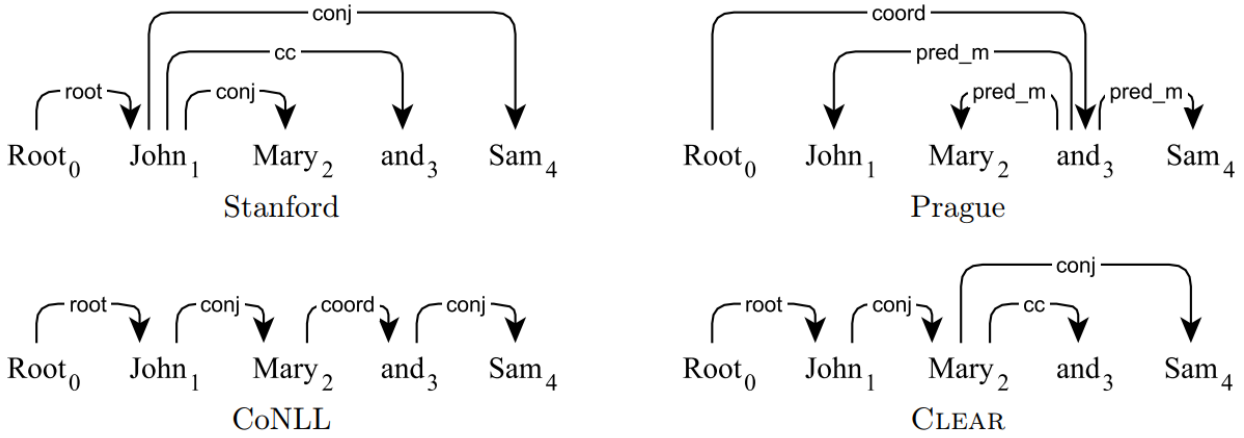
Tiếng Việt có 2 loại liên ngữ là syndetic coordination và asyndetic coordination:

- Trong syndetic coordination, các conjunct sẽ được nối với nhau bởi liên từ (và, hoặc, hay, …)
- Ngược lại, asyndetic coordination sẽ không sử dụng các liên từ.
- Trong tiếng Việt, asyndetic coordination – loại liên ngữ không dùng liên từ thường phổ biến hơn syndetic coordination.

Các trường hợp liên ngữ xuất hiện trong kho ngữ liệu NIIVTB gồm:

- Một ngữ được dán nhãn UCP.
- Một ngữ có ít nhất một node con được gán nhãn Cp hoặc CONJP.
- Tất cả các node con trong một ngữ phải được dán nhãn giống nhau.

2.2.2 Tìm head của liên ngữ



Hình 2.2 Các cách xác định head của Stanford, Prague, CoNLL, CLEAR[4]

Có 4 cách tiếp cận trong việc xác định head cho liên ngữ (hình 2.2). Theo Stanford, head của liên ngữ là conjunct[1] đầu tiên bên trái và các conjunct và liên từ còn lại sẽ làm dependent. Cách của Prague[5] thì cho phép liên từ đầu tiên bên phải làm head cho tất cả conjunct và liên từ còn lại. Hướng tiếp cận của CoNLL thì quy định từ đứng trước sẽ làm head cho từ đứng sau trong cụm liên ngữ. Cuối cùng là CLEAR, với hướng tiếp cận tương tự CoNLL vì CoNLL cho kết quả tốt hơn trên mô hình transition-based[6], nhưng theo CLEAR thì liên ngữ sẽ không làm head vì đảm bảo tính nhất quán của hai trường hợp là

[1] Conjunct là một thành phần của liên ngữ. Ví dụ: "A, B, và C" thì A, B, C là các conjunct và từ "và" là liên từ.

liên ngữ có liên từ hoặc không có liên từ. Hai lý do của CLEAR cũng là cơ sở để chúng tôi quyết định tuân thủ theo cách tiếp cận của CLEAR trong việc xác định head của liên ngữ.

2.3 Thách thức trong việc xác định đúng head

Do sự khác nhau giữa bộ nhãn Vietnamese treebank và NIIVTB, nên đã gây khó khăn trong việc xác định độ ưu tiên của các nhãn. Ví dụ là luật tìm head cho NP thì trong bộ luật của Vietnamese treebank sẽ đưa ra danh sách ưu tiên là [-H; NP; Nc; Nu; Np; N; P; VP;*]. Chúng tôi nhận thấy nhóm danh từ các nhãn Nc, Nu, Np, N thì đều xuất hiện ở cả hai kho ngữ liệu dù ký hiệu khác nhau nhưng ý nghĩa của chúng thì giống nhau. Nhưng NIIVTB có các nhãn khác thuộc nhóm danh từ như Nt, Nq, Num, Nw hay các nhãn nội cấu trúc như Nn_swsp hay Nn_w. Để giải quyết vấn đề xác định độ ưu tiên cho các nhãn khác của NIIVTB, đầu tiên chúng tôi giả sử độ ưu tiên của các nhãn là như nhau thông qua kí hiệu |. Sau khi chuyển đổi xong cú pháp thành tố sang cú pháp phụ thuộc, chúng tôi tiến hành khảo sát trên dữ liệu và đã bổ sung thêm luật phụ để xác định đúng head cho một vài các ngữ (hình 2.2)

Nhãn	Danh sách ưu tiên
S	S|SQ|SPL; ADJP
SBAR	S|SQ|SPL
NP	Nn_swsp|Nn_w; Nn|Nu|Nun|Nt; Num|Nq|Nr; Pd|Pp
Nn_swsp	Ncs|Nc
QP	Nq;Num
VP	Ve|Vc|D|Vcp|Vv
ADJP	Aa

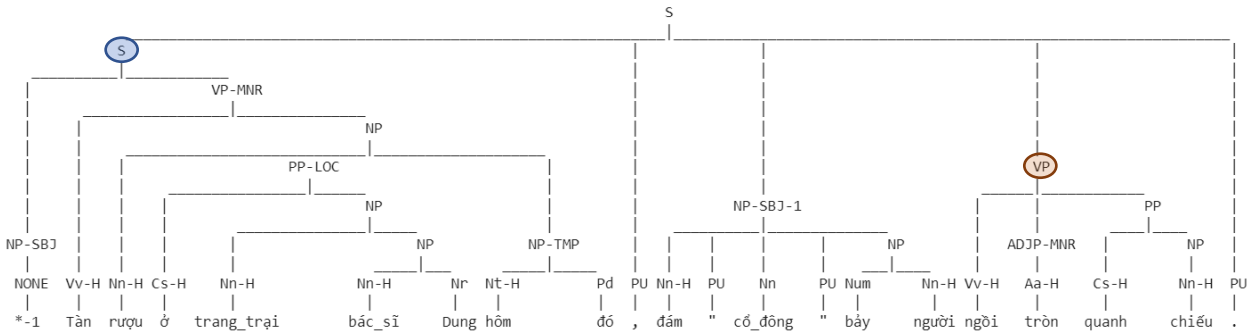
Bảng 2.2 Danh sách phụ tìm head

Để đảm bảo việc xác định đúng head hơn nữa, chúng tôi cũng thử nghiệm thêm bộ luật tìm head

của Stanford[2] và ClearNLP[7] lên kho ngữ liệu NIIVTB. Từ đó, chúng tôi đã phát hiện một vài điểm cần thay đổi ở danh sách ưu tiên của bộ luật tìm head Vietnamese Treebank để phù hợp với kho ngữ liệu NIIVTB.

Ví dụ điển hình là tìm head cho cây cú pháp thành tố (hình 2.3):

[2] Link tham khảo luật tìm head của Stanford: CoreNLP/CollinsHeadFinder.java at 16ac6de8b1d5ecd959170ad78ea965ee5fba89a5 · stanfordnlp/CoreNLP · GitHub



Hình 2.3 Cây cú pháp thành tố

Khi áp dụng bộ luật tìm head Vietnamese Treebank thì head của S sẽ là S(được tô xanh), trong khi áp dụng luật của Stanford thì head của S là VP(được tô cam). Và chúng tôi nhận thấy head đúng ở đây sẽ là VP. Qua đây, chúng tôi đã thay đổi vị trí ưu tiên giữa S và VP trong bộ luật.

3 Dán nhãn cú pháp phụ thuộc

3.1 Lược đồ chú thích nhãn cú pháp phụ thuộc

STT	Nhãn	Mô tả
1	ACOMP	Adjectival complement
2	ADJP_ADVMOD*	Adjective phrase as adverbial modifier
3	ADJUNCT*	Adjunct
4	ADVCL	Adverbial clause modifier
5	AMOD	Adjective modifier
6	AOBJ*	Object of an adjective
7	APPOS	Appositional modifier
8	ASUBJ*	Adjectival subject
9	ATTR	Attribute
10	CC	Coordination
11	CCOMP	Clausal complement
12	CONJ	Conjunct
13	CSUBJ	Clausal subject
14	DEP	Dependent
15	DET	Determiner
16	INTJ	Interjection
17	IOBJ	Indirect object
18	MARK	Marker
19	NC*	Noun classifier
20	NCS*	Special noun classifier
21	NN	Noun compound modifier

22	NP_ADVMOD	Noun phrase as adverbial modifier
23	NSUBJ	Nominal subject
24	NUM	Numeric modifier
25	NUMBER	Number compound modifier
26	OBJ	Object
27	PARATAXIS	Parataxis
28	PCOMP	Prepositional complement
29	POBJ	Object of a preposition
30	PREP	Prepositional modifier
31	PUNCT	Punctuation
32	QUANTMOD	Quantifier phrase modifer
33	RCMOD	Relative clause modifier
34	ROOT	Root
35	SINO*	Sino-Vietnamese
36	SOUND*	Sound
37	VMOD	Verbal modifier
38	VOCATIVE	Vocative
39	VSUBJ*	Verbal subject
40	XCOMP	Open clausal complement

*Bảng 3.1 Danh sách các nhãn cú pháp phụ thuộc cho NIIVTB. Các nhãn mà được theo sau bởi * là các nhãn mới được chúng tôi đề xuất.*

Bảng 3.1 liệt kê các nhãn cú pháp phụ thuộc được sử dụng trong việc chuyển đổi kho ngữ liệu cú pháp thành tố NIIVTB sang cú pháp phụ thuộc. Hầu hết các nhãn chúng tôi mô tả đều được dựa trên cách tiếp cận của Stanford[8], ClearNLP[4], và Universal Dependencies[3]. Ngoài ra, để phù hợp với đặc điểm ngôn ngữ tiếng Việt và kho ngữ liệu NIIVTB, chúng tôi đã bổ sung thêm một vài nhãn mới được theo sau bởi *****.

3.2 Bộ luật dán nhãn tự động

Thuật toán 3.1 - attachDependencyLable(P,C,p,c) sẽ dán nhãn cú pháp phụ thuộc cho mối quan hệ p → c chủ yếu dựa trên các nhãn chức năng trong cú pháp thành tố (dòng 1,2,3,5,7,8,15) và các luật chuyển đổi tự động (các dòng còn lại). Chi tiết về các thuật toán con được mô tả ở các mục chú thích bên phải.

Thuật toán 3.1: *attachDependencyLabel(P, C, p, c):*

Đầu vào: Một ngữ C, P là Parent của C, p là head của P, c là head của C
Đầu ra : The dependency label for relation p → c

1: **if** *isSubject(C)* ≠ null **return** *isSubject(C)* # Mục 3.3.3
2: **if** *isAdverbialClauseModifier(C)* ≠ null **return** *isAdverbialClauseModifier(C)* # Mục 3.5.1

[3] Link tham khảo Universal Dependencies: Universal Dependencies

```
3:  if isNounPhraseAsAdverbialModifier(C) ≠ null then                                  # Mục 3.5.2
4:      return isNounPhraseAsAdverbialModifier(C)
5:  if isAdjectivePhraseAsAdverbialModifier(C) ≠ null then                             # Mục 3.5.1
6:      return isAdjectivePhraseAsAdverbialModifier(C)
7:  if isParataxis (C) ≠ null return isParataxis (C)                                   # Mục 3.12.4
8:  if isVocative (C) ≠ null return isVocative (C)                                     # Mục 3.12.5
9:  if isAdjunct (C) ≠ null return isAdjunct (C)                                       # Mục 3.12.1
10: if isPunctuation (C) ≠ null return isPunctuation (C)                               # Mục 3.12.6
11: if isSinoVietnamese (P) ≠ null return isSinoVietnamese (P)                         # Mục 3.12.10
12: if isInterjection(C) ≠ null return isInterjection(C)                               # Mục 3.12.11
13: if isCoordination (C) ≠ null return isCoordination (C)                             # Mục 3.6.2
14: if isMarker( C) ≠ null return isMarker( C)                                         # Mục 3.5.2
15: if isAppositionalModifier(P, C) ≠ null return isAppositionalModifier(P, C)         # Mục 3.8.5
16: if isSound (c) ≠ null return isSound (c)                                           # Mục 3.12.9
17: if isVerbalModifier (P, C) ≠ null return isVerbalModifier (P, C)                   # Mục 3.12.2
18: if isRelativeClauseOrVerbalModifier(P, C) ≠ null then                              # Mục 3.8.4
19:     return isRelativeClauseOrVerbalModifier(P, C)
20: if isNumericModifier(P, C) ≠ null return isNumericModifier(P, C)                   # Mục 3.8.3
21: if isNounCompoundModifier(P, C) ≠ null return isNounCompoundModifier(P, C)         # Mục 3.8.1
22: if isClassifierOrNounCompoundModifier (P, p) ≠ null then                           # Mục 3.12.8
23:     return isClassifierOrNounCompoundModifier (P, p)
24: if isNumberOrQuantifierModifier (P, C) ≠ null then                                 # Mục 3.11
25:     return isNumberOrQuantifierModifier (P, C)
26: if isDeterminer(P, C) ≠ null return isDeterminer(P, C)                             # Mục 3.8.2
27: if isPrepositionalModifier (C) ≠ null return isPrepositionalModifier(C)            # Mục 3.10.3
28: if isPrepComplementOrObject (P, C) ≠ null return isPrepComplementOrObject (P, C)   # Mục 3.10.2
29: if isAdjectiveModifier (P, C) ≠ null return isAdjectiveModifier (P, C)             # Mục 3.9.2
30: if isAttribute(p, C) ≠ null return isAttribute(p, C)                               # Mục 3.4.1
31: if isIndirectObject(P, C) ≠ null return isIndirectObject(P, C)                     # Mục 3.4.3
32: if isObject(P, C) ≠ null return isObject(P, C)                                     # Mục 3.4.2
33: if isObjectOfAdjective (P, C) ≠ null return isObjectOfAdjective (P, C)             # Mục 3.9.1
34: if isAdjectivalComplement (P, C) ≠ null return isAdjectivalComplement (P, C)       # Mục 3.7.1
35: if isClausalComplement (P, C) ≠ null return isClausalComplement (P, C)             # Mục 3.7.2
36: if isClausalOrOpenClausalComplement (P, C) ≠ null then                             # Mục 3.7.3
37:     return isClausalOrOpenClausalComplement (P, C)
38: return DEP
```

3.3 Nhóm nhãn – Chủ ngữ

Nhóm nhãn – Chủ ngữ bao gồm clausal subject (CSUBJ), nominal subject (NSUBJ), verbal subject (VSUBJ), và adjectival subject (ASUBJ).

Thuật toán 3.2: *isSubject(C):*

Đầu vào: Một ngữ C
Đầu ra : CSUBJ, NSUBJ, VSUBJ, hoặc ASUBJ

```
1:  if C has SBJ then
2:      if C is S|SPL|SBAR|SQ return CSUBJ
3:      if C is NP|QNP return NSUBJ
4:      if C is VP return VSUBJ
5:      if C is ADJP return ASUBJ
6:  return null
```

3.3.1 CSUBJ: clausal subject

Clausal subject thể hiện một mệnh đề đóng vai trò như là một chủ ngữ trong câu. Bất kỳ một loại mệnh đề nào mà được gắn nhãn chức năng **SBJ** sẽ được xem là **CSUBJ**.

Ví dụ 1: [Mẹ_con mình cứ buồn, cứ khóc mãi] chẳng ích gì … CSUBJ(ích, buồn)
Ví dụ 2: [Anh ấy bình_phục] là tốt rồi. CSBUJ(tốt, bình_phục)

3.3.2 NSUBJ: nominal subject

Nominal subject là một ngữ danh từ làm chủ ngữ của mệnh đề. Các ngữ danh từ mà được gắn nhãn chức năng **SBJ** sẽ được xem là **NSUBJ**.

Ví dụ 1: [Chúng_tôi] chưa có kế hoạch tuyển người. NSUBJ(có, chúng_tôi)
Ví dụ 2: Ba [mẹ_con] chỉ biết sống dựa vào nhau. NSUBJ(biết, mẹ_con)

3.3.3 VSUBJ: verbal subject

Verbal subject là một ngữ động từ làm chủ ngữ của mệnh đề. Các ngữ động từ mà được gắn nhãn chức năng **SBJ** sẽ được xem là **VSUBJ**.

Ví dụ 1: [Đi học] không mất tiền mà lại có lương. VSUBJ(mất, đi)
Ví dụ 2: [Có] tài thôi chưa đủ. VSUBJ(đủ, có)

3.3.4 ASUBJ: adjectival subject

Adjectival subject là một ngữ tính từ làm chủ ngữ của mệnh đề. Các ngữ tính từ mà được gắn nhãn chức năng **SBJ** sẽ được xem là **ASUBJ**.

Ví dụ 1: [Khó_khăn] nhất là dân cư_trú bất hợp_pháp nhiều quá.
→ ASUBJ(là, Khó_khăn)
Ví dụ 2: [Nóng] dễ_chịu hơn lạnh. ASUBJ(dễ chịu, Nóng)

3.4 Nhóm nhãn – Tân ngữ

Nhóm nhãn – Tân ngữ bao gồm attribute (ATTR), object (OBJ), và indirect object (IOBJ).

3.4.1 ATTR: attribute

Atrribute là một cụm danh từ mà theo sau động từ copula 'là'.

Ví dụ 1: Cát là [nỗi ám_ảnh của dân ngư chài] ATTR(là, nỗi)
Ví dụ 2: Người đàn_ông có nửa đời người là [thợ_săn] ATTR(là, thợ_săn)

Thuật toán 3.3: *isAttribute(p, C):*

Đầu vào: Một ngữ C, p là head của Parent của C
Đầu ra : ATTR hoặc null

```
1: if (p is Vc) and (C is NP|QNP) return ATTR
2: return null
```

3.4.2 OBJ: object

Object là một tân ngữ trực tiếp theo sau một động từ. Tân ngữ là một ngữ danh từ. Các danh từ mà không thuộc trường hợp **IOBJ** hay **ATTR** sẽ được tính là một tân ngữ trực tiếp.

Ví dụ 1: Thợ_săn vờn đuổi [hổ dữ] trong rừng rậm. OBJ(vờn, hổ)
Ví dụ 2: 44.000 ha đất nông_nghiệp gặp [khó_khăn về nguồn nước].
→ OBJ(gặp, khó_khăn)

Thuật toán 3.4: *isObject(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : OBJ hoặc null

```
1: if P is VP then
2:     if (C is NP|QNP|Nn) or (C has DOB) return OBJ
3: return null
```

3.4.3 IOBJ: indirect object

Indirect object là một tân ngữ gián tiếp theo sau một động từ. Tân ngữ là một ngữ danh từ. Các danh từ được gán nhãn chức năng **IOB** sẽ tính là các tân ngữ gián tiếp.

Ví dụ 1: Một đầu hè căn nhà được người em cho [vợ_chồng chị Xoan].
→ IOBJ(cho, vợ_chồng)
Ví dụ 2: 80 triệu đồng ủng_hộ [bệnh_nhân nghèo]. IOBJ(ủng_hộ, bệnh_nhân)

Thuật toán 3.5: *isIndirectObject(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : IOBJ hoặc null

```
1: if (P is VP) and (C has IOB) return IOBJ
2: return null
```

3.5 Nhóm nhãn – Trạng ngữ

Nhóm nhãn – Trạng ngữ bao gồm adverbial clause modifier (ADVCL), noun phrase as adverbial modifier (NP_ADVMOD), adjective phrase as adverbial modifier (ADJP_ADVMOD, marker (MARK)

3.5.1 ADVCL: adverbial clause modifier

Adverbial clause modifier là một mệnh đề đóng vai trò như là trạng ngữ.

Ví dụ 1: Làm nhà [vì thương người lận_đận] ADVCL(Làm, thương)

Ví dụ 2: [Đem bán với giá rẻ] nhà anh cũng kiếm được 250.000 – 300.000 đồng.
→ ADVCL(kiếm, đem)

Thuật toán 3.6: *isAdverbialClauseModifier(C):*

Đầu vào: Một ngữ C
Đầu ra : ADVCL hoặc null

```
1: if (C is S|SPL|SBAR) and (C has TMP|MNR|ADV|LOC|PRP|MDP|CND|CNC) then
2:     return ADVCL
3: return null
```

3.5.2 NP_ADVMOD: noun phrase as adverbial modifier

Noun phrase as adverbial modifier là một ngữ danh từ có chức năng như trạng ngữ.

Ví dụ 1: [Ba năm sau], gia_đình anh rơi vào cảnh túng_quẫn
→ NP_ADVMOD(rơi, năm)

Ví dụ 2: [Làng bên], đàn_ông bỏ đi hết NP_ADVMOD(bỏ, Làng)

Thuật toán 3.7: *isNounPhraseAsAdverbialModifier(C):*

Đầu vào: Một ngữ C
Đầu ra : NP_ADVMOD hoặc null

```
1: if (C is NP) and (C has TMP|MNR|ADV|LOC|PRP|MDP|CNC|CND) then
2:     return NP_ADVMOD
3: return null
```

3.5.1 ADJP_ADVMOD: adjective phrase as adverbial modifier

Adjective phrase as adverbial modifier là một ngữ tính từ có chức năng như trạng ngữ.

Ví dụ 1: [Xa_xa], lấp_lánh ánh đèn. ADJP_ADVMOD(lấp_lánh, Xa_xa)

Ví dụ 2: Cách nhà chưa đầy 20 cây_số nhưng [lâu_lâu], anh mới về được.
→ADJP_ADVMOD(về, lâu_lâu)

Đây là nhãn mang đặc trưng riêng của tiếng Việt. Khi so sánh với tiếng Anh thì tiếng Anh có sự biến đổi hình thái của từ như là tính từ thêm đuôi "**-ly**" sẽ thành trạng từ. Nhưng

trong tiếng Việt thì không có sự biến đổi này thay vào đó là tính từ sẽ đóng vai trò như một trạng từ. Một ví dụ về sự khác biệt giữa tiếng Anh và tiếng Việt:

Tiếng Việt: Cô ấy hát quá hay/Adj **Tiếng Anh:** She sings very beautifully/Adv

Thuật toán 3.8: *isAdjectivePhraseAsAdverbialModifier(C):*

Đầu vào: Một ngữ C
Đầu ra : ADJP_ADVMOD hoặc null

```
1: if (C is ADJP) and (C has TMP|MNR|ADV|LOC|PRP|CNC|CND) then
2:     return ADJP_ADVMOD
3: return null
```

3.5.2 MARK: marker

Marker là từ mở đầu cho câu phức.

Ví dụ 1: Anh cho [rằng tháng chín là cơ_hội tốt để kiếm tiền] MARK(là, rằng)
Ví dụ 2: Cuốn sách [mà tôi mua hôm qua rất hay] MARK(mua, mà)
Ví dụ 3: Tôi từng khóc [vì mình quá nghèo] MARK(nghèo, vì)

Thuật toán 3.9: *isMarker(C):*

Đầu vào: Một ngữ C
Đầu ra : MARK hoặc null

```
1: if C is Cs return MARK
2: return null
```

3.6 Nhóm nhãn – Liên ngữ

Nhóm nhãn – Liên ngữ bao gồm conjunct (CONJ) và coordination (CC).

3.6.1 CONJ: conjunct

Conjuct là mối quan hệ giữa 2 conjunct trong liên ngữ. Trong đó, conjunct đầu tiên bên trái sẽ làm head.

Ví dụ 1: Thợ_săn [vờn] [đuổi] hổ dữ trong rừng rậm. CONJ(vờn, đuổi)
Ví dụ 2: Mồm ngậm ngang lưỡi dao [nhọn] [mỏng_dính] [bé] xíu như lá lúa
→ CONJ(nhọn, mỏng_dính), CONJ(mỏng_dính, bé)
Ví dụ 3: Nhiều người thuê [giáo_sư], [bác_sĩ], và [nghệ_nhân] thẩm_định cao
→ CONJ(giáo_sư, bác_sĩ), CONJ(bác_sĩ, nghệ_nhân)

3.6.2 CC: coordination

Coordination là mối quan hệ giữa conjunct và liên từ như và, hoặc, hay,…

Ví dụ 1: Phước [vừa$_1$] bán vé_số nuôi bà [vừa$_2$] dành tiền mua nhà.
→CC(bán, vừa$_1$), CC(bán, vừa$_2$)
Ví dụ 2: 600 chú chim_trời sa lưới [và] bị hóa_kiếp. CC(sa, và)

Thuật toán 3.10: *isCoordination (C):*

Đầu vào: Một ngữ C
Đầu ra : CC hoặc null

1: **if** C is CONJP|Cp **return** CC
2: **return** null

3.7 Nhóm nhãn – Bổ ngữ

Nhóm nhãn – Bổ ngữ bao gồm adjectival complement (ACOMP), clausal complement (CCOMP), open clausal complement (XCOMP).

3.7.1 ACOMP: adjectival complement

Adjectival complement là một ngữ tính từ bổ nghĩa cho động từ.

Ví dụ 1: Chi_hội sẽ thực_hiện một_số kế_hoạch làm [từ_thiện].
→ACOMP(làm, từ_thiện)
Ví dụ 2: Muốn biết [đúng hay sai] tòa phải nhờ hội_đồng thẩm_định
→ACOMP(biết, đúng)
Ví dụ 3: Khả_năng đóng cửa_biển là [rất lớn] ACOMP(là, lớn)

Thuật toán 3.11: *isAdjectivalComplement (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : ACOMP hoặc null

1: **if** (P is VP) **and** (C is ADJP) **return** ACOMP
2: **return** null

3.7.2 CCOMP: clausal complement

Clausal complement là mệnh đề bổ ngữ cho NP|VP|ADJP|RP|S|SQ|QNP|QVP.

Ví dụ 1: Tàu cứu_hộ báo [họ không_thể vào gần tàu bị nạn] CCOMP(báo, vào)
Ví dụ 2: Có người thấy [ông làm việc thiện] CCOMP(thấy, làm)
Ví dụ 3: Ông khẳng_định [(rằng) đây là việc_làm sai_trái của chính_quyền]
→CCOMP(khẳng_định, là)

Thuật toán 3.12: *isClausalComplement (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : CCOMP hoặc null

1: **if** (P is NP|VP|ADJP|RP|S|SQ|QNP|QVP) **and** (C is SQ|SPL) **return** CCOMP
2: **return** null

3.7.3 XCOMP: open clausal complement

Open clausal complement là một mệnh đề bổ ngữ cho NP|VP|ADJP|RP|S|SQ|QNP|QVP. Nhưng mệnh đề này không có chủ ngữ.

Ví dụ 1: Ông dặn hắn [tìm khách mua xương thì được hưởng triệu] →XCOMP(dặn, tìm)
Ví dụ 2: Anh chẳng_thể biết [phải bơi theo hướng nào] XCOMP(biết, phải)
Ví dụ 3: Anh đã quyết_định [đi kiện]. XCOMP(quyết_định, đi)

Thuật toán 3.13: *isClausalOrOpenClausalComplement (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : XCOMP hoặc CCOMP hoặc null

```
1:  if (P is NP|VP|ADJP|RP|S|SQ|QNP|QVP) then
2:     if C is S then
3:        if (C has VP) and (C has the emty subject) return XCOMP
4:        else return CCOMP
5:     if C is SBAR then
6:        let SubC be S in C then
7:        return isClausalOrOpenClausalComplement (P, SubC)
8:  return null
```

3.8 Nhóm nhãn – Danh từ

Nhóm nhãn – Danh từ là các nhãn liên quan về danh từ như noun compound modifier (NN), determiner (DET), numeric modifier (NUM), relative clause modifier (RCMOD), appositional modifier (APPOS).

3.8.1 NN: noun compound modifier

Noun compound modifier là một danh từ hoặc ngữ danh từ bổ nghĩa cho head của một ngữ danh từ.

Ví dụ 1: Mùa [nước nổi] dâng lên có chỗ ngập trên 2,5 m NN(Mùa, nước)
Ví dụ 2: Đất [nông_nghiệp] NN(Đất, nông_nghiệp)

Thuật toán 3.14: *isNounCompoundModifier(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : NN hoặc null

```
1:  if (P is NP|QNP)  and (C is NP|Nr|Nt|Nu|Nun|Nn|ID) return NN
2:  return null
```

3.8.2 DET: determiner

Determiner là mối quan hệ giữa head của một ngữ danh từ và từ hạn định.

Ví dụ 1: [Những] quan_khách lịch_lãm [ấy]
→DET(quan_khách, Những), DET(quan_khách, ấy)
Ví dụ 2: Anh_em bắn lên [mọi] tín_hiệu cấp_cứu có được DET(tín_hiệu, mọi)

Thuật toán 3.15: *isDeterminer(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : DET hoặc null

1: **if (**P is NP|QNP) **and** (C is Nw|Nq|Pd|Pp) **return** DET
2: **return** null

3.8.3 NUM: numeric modifier

Numeric modifier là là một số hoặc ngữ chỉ định lượng bổ nghĩa cho head của một ngữ danh từ.

Ví dụ 1: [Hai] chiều thương_nhớ... NUM(chiều, Hai)
Ví dụ 2: Năm [1963] chị hồi_hương theo gia_đình về VN. NUM(Năm, 1963)
Ví dụ 3: Tổng cộng [150 triệu] đồng. NUM(đồng, triệu)

Thuật toán 3.16: *isNumericModifier(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : NUM hoặc null

1: **if (**P is NP|QNP|Nn) **and** (C is Num|QP) **return** NUM
2: **return** null

3.8.4 RCMOD: relative clause modifier

Relative clause modifier là một mệnh đề quan hệ bổ nghĩa cho một ngữ danh từ.

Ví dụ 1: Khách [mua cao] thường là chỗ quen_thân
→RCMOD(Khách, mua)
Ví dụ 2: Những người [(mà) được ông giúp] đều chí_thú làm_ăn
→RCMOD(người, được)
Ví dụ 3: Tôi là người [(mà) phải chịu ở dưới nước lâu nhất], kinh_hãi lắm…
→RCMOD(người, phải)

Thuật toán **3.17:** *isRelativeClauseOrVerbalModifier(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : RCMOD hoặc VMOD hoặc null

1: **if (**P is NP|QNP) **and** (C is VP) **then**
2: **if** C has the object **return** RCMOD
3: **else return** VMOD
4: **return** null

3.8.5 APPOS: appositional modifier

Appositional modifier là một ngữ danh từ đưa thêm thông tin cho một ngữ danh từ khác đứng phía trước nó.

Ví dụ 1: Nam, anh trai tôi APPOS(Nam, anh)
Ví dụ 2: Bài báo " Con gái Sài Gòn và con gái Hà Nội" đã gây sốt cộng đồng mạng →APPOS(bài, Con)
Ví dụ 3: Bộ phim "Bố Già" APPOS(phim, Bố)

Thuật toán 3.18: *isAppositionalModifier(P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : APPOS hoặc null

```
1: if (P is NP) and (C has HLN|TTL) return APPOS
2: return null
```

3.9 Nhóm nhãn – Tính từ

Nhóm nhãn – Tính từ gồm object of an adjective (AOBJ), adjective modifier (AMOD).

3.9.1 AOBJ: object of an adjective

Object of an adjective là một tân ngữ của tính từ. Tân ngữ thường là một ngữ danh từ.

Ví dụ 1: Ngày nào quán cũng đông [khách]. AOBJ(đông, khách)
Ví dụ 2: Tòa tháp cao [chín tầng]. AOBJ(cao, tầng)
Ví dụ 3: Tôi giỏi [toán]. AOBJ(giỏi, toán)

AOBJ là một nhãn dán mang đặc trưng riêng cho tiếng Việt. Cấu trúc của tiếng Việt trong hình thái học thường có dạng SVO (Chủ từ - Động từ - Túc từ). Tuy nhiên, đôi khi động từ có thể thay thế bằng tính từ SAO (Chủ từ - Tính từ - Túc từ). Để làm rõ đặc điểm này chúng tôi đã bổ sung thêm nhãn AOBJ – túc từ của tính từ như các ví dụ ở trên.

Thuật toán 3.19: *isObjectOfAdjective (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : AOBJ hoặc null

```
1: if P is ADJP then
2:     if (C is NP|QNP|Nn) or (C has DOB) return AOBJ
3: return null
```

3.9.2 AMOD: adjective modifier

Adjective modifier là một ngữ tính từ bổ nghĩa cho một ngữ danh từ.

Ví dụ 1: [Nhiều] học_sinh đi du_học AMOD(học_sinh, Nhiều)
Ví dụ 2: Phán_quyết [cuối_cùng] AMOD(Phán_quyết, cuối_cùng)

Ví dụ 3: Bà trân_trọng từng khoảnh_khắc [quý_báu] bên gia đình. →AMOD(khoảnh_khắc, quý_báu)

Thuật toán 3.20: *isAdjectiveModifier (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : AMOD hoặc null

```
1: if P is NP|QNP then
2:     if (C is NP|QNP|Nn) or (C has DOB) return AOBJ
3: return null
```

3.10 Nhóm nhãn – Giới từ

Nhóm nhãn – Giới từ gồm prepositional complement (PCOMP), object of a preposition (POBJ), prepositional modifier (PREP)

3.10.1 PCOMP: prepositional complement

Prepositional complement là các dependent mà không phải là POBJ nhưng bổ nghĩa cho head của ngữ giới từ.

Ví dụ 1: Đây là cơ_hội tốt để [kiếm tiền] PCOMP(để, kiếm)
Ví dụ 2: Hắn làm đồ_ăn nhựa như [thật] PCOMP(như, thật)

3.10.2 POBJ: object of a preposition

Object of a preposition là một ngữ danh từ bổ nghĩa cho head của ngữ giới từ.

Ví dụ 1: Họ xây nhà tình_thương cho [người nghèo] POBJ(cho, người)
Ví dụ 2: Ông đứng bên tôi với [giọng_nói đầy tự_hào] POBJ(với, giọng_nói)

Thuật toán 3.21: *isPrepComplementOrObject (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : POBJ hoặc PCOMP hoặc null

```
1: if P is PP|QPP then
2:     if C is NP return POBJ
3:     else return PCOMP
4: return null
```

3.10.3 PREP: prepositional modifier

Prepositional modifier là một ngữ giới từ được dùng để bổ nghĩa cho các ngữ khác.

Ví dụ 1: Rất nhiều người cài quốc_ca [vào phần nhạc chuông]. PREP(cài, vào)
Ví dụ 2: Ước_mơ [của chúng] là muốn được đi học. PREP(Ước_mơ, của)

Thuật toán 3.22: *isPrepositionalModifier (C):*

Đầu vào: Một ngữ C
Đầu ra : PREP hoặc null

1: **if** C is PP|QPP **return** PREP
2: **return** null

3.11 Nhóm nhãn – Lượng từ

Nhóm nhãn – Lượng từ gồm number compound modifier (NUMBER), quantifier phrase modifier (QUANTMOD).

Thuật toán 3.23: *isNumberOrQuantifierModifier (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : NUMBER hoặc QUANTMOD hoặc null

1: **if** P is QP **then**
2: **if** C is Num|Nq **return** NUMBER
3: **else return** QUANTMOD
4: **return** null

3.11.1 NUMBER: number compound modifier

Number compound modifier là một số học bổ ngữ cho head của ngữ chỉ định lượng.

Ví dụ 1: [150] triệu đồng NUMBER(triệu, 150)
Ví dụ 2: Chiếc ôtô của anh ấy trị_giá [hai] tỷ NUMBER(tỷ, hai)

3.11.2 QUANTMOD: quantifier phrase modifier

Quantifier phrase modifier là các dependent bổ nghĩa cho head của ngữ chỉ định lượng.

Ví dụ 1: Mỗi ngày có [đến 500 – 600] chú chim_trời bị sa lưới.
→QUANTMOD(500, đến)
Ví dụ 2: [Cả chục] hầm nuôi cá. QUANTMOD(chục, Cả)

3.12 Nhóm nhãn khác

Nhóm nhác khác bao gồm adjunct (ADJUNCT), verbal modifier (VMOD), dependent (DEP), parataxis (PARATAXIS), vocative (VOCATIVE), punctuation (PUNCT), root (ROOT), noun classifier (NC), special noun classifier (NCS), sound (SOUND), Vietnamese-sino (SINO), và interjection (INTJ).

3.12.1 ADJUNCT: adjunct

Adjunct được dùng để bổ nghĩa cho danh từ, động từ, tính từ, … Chúng gồm trợ từ, trạng từ, từ phủ định.

Ví dụ 1: Ông [luôn] thấy rùng_mình. ADJUNCT(thấy, luôn)
Ví dụ 2: [Thật] kinh_ngạc! ADJUNCT(kinh_ngạc, Thật)
Ví dụ 3: Trời [sắp] mưa ADJUNCT(mưa, sắp)

Thuật toán 3.24: *isAdjunct (C):*

Đầu vào: Một ngữ C

Đầu ra : ADJUNCT hoặc null

1: **if** C is R|RP|QRP **return** ADJUNCT
2: **return** null

3.12.2 VMOD: verbal modifier

Verbal modifier là động từ bổ nghĩa cho danh từ, động từ, tính từ, …

Ví dụ 1: Giá [nấu] [thuê] 5-10 triệu VMOD(Giá, nấu), VMOD(nấu, thuê)
Ví dụ 2: Tôi đi [ăn] phở VMOD(đi, ăn)

Thuật toán 3.25: *isVerbalModifier (P, C):*

Đầu vào: Một ngữ C, P là Parent của C
Đầu ra : VMOD hoặc null

1: **if (**P is VP|ADJP|S*) **and** (C is VP|Ve|Vc|D|Vcp|Vv|VN) **return** VMOD
2: **return** null

Chú ý: S* là một trong bốn loại nhãn cấp mệnh đề (xem trong phụ lục A.2)

3.12.3 DEP: dependent

Dependent là nhãn mặc định khi không xác định được mối quan hệ.

3.12.4 PARATAXIS: parataxis

Parataxis là một cụm từ dùng để bổ sung thêm thông tin, thường được đặc trong 1 cặp dấu(dấu ngoặc kép, dấu phẩy,…).

Chủ nhà, [viện phó một bệnh_viện], gật và đếm tiền. PARATAXIS(Chủ, bệnh_viện)

Thuật toán 3.26: *isParataxis (C):*

Đầu vào: Một ngữ C
Đầu ra : PARATAXIS hoặc null

1: **if** C has PRN **return** PARATAXIS
2: **return** null

3.12.5 VOCATIVE: vocative

Vocative là mối quan hệ được dùng để đánh dấu các ngữ chỉ địa chỉ.

Ví dụ 1: Thấy thương_tâm lắm [chú]! VOCATIVE(thấy, chú)
Ví dụ 2: [Thưa ông], sổ_đỏ đã được giao cho người dân. VOCATIVE(được, ông)
Ví dụ 3: [Dạ], em hiểu rồi. VOCATIVE(hiểu, Dạ)

Thuật toán 3.27: *isVocative (C):*

Đầu vào: Một ngữ C
Đầu ra : VOCATIVE hoặc null

1: **if** C has VOC **return** VOCATIVE
2: **return** null

3.12.6 PUNCT: punctuation

Punctuation là các dấu như . , “” …

Ví dụ 1: Tuyệt quá! PUNCT(Tuyệt, !)

Thuật toán 3.28: *isPunctuation (C):*

Đầu vào: Một ngữ C
Đầu ra : PUNCT hoặc null

1: **if** C is PU|LBRK|RBRK **return** PUNCT
2: **return** null

3.12.7 ROOT: root

Root là mối quan hệ chỉ ra “gốc” của câu. Chỉ số head của mối quan hệ này sẽ được gán là 0.

Ví dụ 1: Mọi người ồn_ào đếm tiền. ROOT(đếm)
Ví dụ 2: Cô ấy rất giỏi ROOT(giỏi)

3.12.8 NC và NCS: noun classifier và special noun classifier

Tương tự các ngôn ngữ Châu Á khác (tiếng Trung, tiếng Thái,…), tiếng Việt cũng có classifier - loại từ đi chung với danh từ (nhãn **NC**). Bên cạnh đó các classifier trong tiếng Việt còn có thể đi chung với cả động từ và tính từ (nhãn **NCS**).

Ví dụ 1: [Người] thanh_niên NC(Người, thanh_niên)
Ví dụ 2: [Nỗi] sợ_hãi NCS(Nỗi, sợ_hãi)
Ví dụ 3: [Việc] đi_lại NCS(Việc, đi_lại)

Thuật toán 3.29: *isClassifierOrNounCompoundModifier (P, p):*

Đầu vào: P là parent của ngữ C đang xét, p là head của P
Đầu ra : NC hoặc NCS hoặc null

1: **if** P is Nn_swsp **then**
2: **if** p is Nc **return** NC
3: **else if** p is Ncs **return** NCS
4: **else return** NN
5: **return** null

3.12.9 SOUND: sound

Sound là mối quan hệ bổ trợ về khía cạnh âm thanh.

Ví dụ 1: Tiếng ống_bơ kêu $coong_1$, $coong_2$…
→SOUND(kêu, $coong_1$), CONJ($coong_1$, $coong_2$)

Thuật toán 3.30: *isSound (c):*

Đầu vào: c là head của C
Đầu ra : SOUND hoặc null

1: **if** c is ON **return** SOUND
2: **return** null

3.12.10 SINO: Sino-Vietnamese

Sino-Vietnamese là các mối quan hệ bổ nghĩa của từ Hán Việt. Đây cũng là nhãn mang đặc trưng riêng cho tiếng Việt.

Ví dụ 1:	Viện [phó]	SINO(Viện, phó)
Ví dụ 2:	[Bất] phương_trình	SINO(phương_trình, Bất)
Ví dụ 3:	Huấn_luyện [viên]	SINO(Huấn_luyện, viên)

Thuật toán 3.31: *isSinoVietnamese (P):*

Đầu vào: P là Parent của C
Đầu ra : SINO hoặc null

1: **if** P is Nn_w|Vv_w|Aa_w|R_w **return** SINO
2: **return** null

3.12.11 INTJ: interjection

Interjection là mối quan hệ thể hiện biểu sự biểu cảm của người nói.

Ví dụ 1:	Chỉ cần như thế [thôi]	INTJ(cần, thôi)
Ví dụ 2:	Mẹ [ơi]!	INTJ(Mẹ, ơi)

Thuật toán 3.32: *isInterjection(C):*

Đầu vào: Một ngữ C
Đầu ra : INTJ hoặc null

1: **if (**C is E|M) **or** (C has MDP)**return** INTJ
2: **return** null

3.13 Thách thức trong việc xây dựng bộ luật dán nhãn cú pháp phụ thuộc

Tài liệu hướng dẫn dán nhãn tự động của chúng tôi được lấy cảm hứng từ tác giả Choi [4]. Tuy nhiên, do tác giả thực hiện chuyển đổi tự động trên các kho ngữ liệu tiếng Anh nên đã có một vài sự khác biệt giữa bộ nhãn cú pháp thành tố tiếng Anh và tiếng Việt. Cụ thể là trong kho ngữ liệu tiếng Việt NIIVTB tồn tại một loại nhãn thể hiện "song" từ loại như

VA, VN, NA (xem chi tiết ở phụ lục A.2). Ví dụ là một từ được gán nhãn VA thì sẽ vừa là động từ vừa là tính từ. Tuy nhiên, trong tài liệu hướng dẫn của tác giả Choi thì tiếng Anh không có loại nhãn này, do đó, chúng tôi đã quyết định là chuyển loại nhãn "song" từ loại thành nhãn từ loại duy nhất. Theo công trình [9],[10], chúng tôi chuyển đổi VA, NA thành nhãn tính từ, và VN thành nhãn động từ.

4 Hậu xử lý

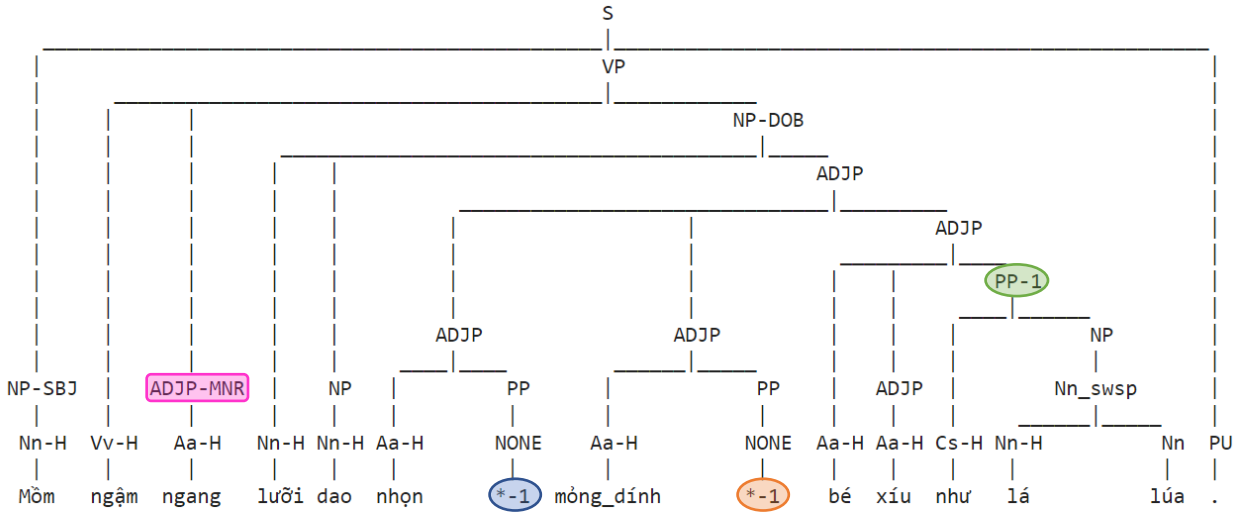
Sau khi chuyển đổi tự động cú pháp thành tố sang cú pháp phụ thuộc, chúng tôi sẽ thực hiện hậu xử lý bao gồm 2 công đoạn là thêm các đặc trưng phụ và khử thành phần NULL.

4.1 Thêm đặc trưng phụ

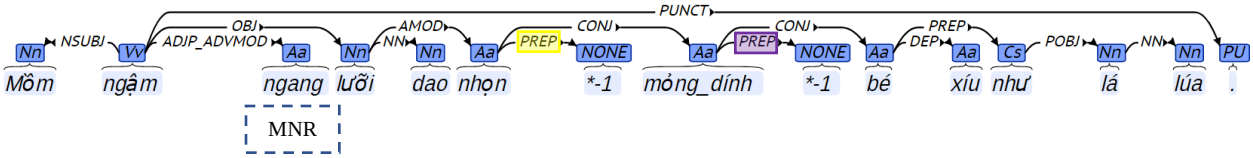
4.1.1 Mối quan hệ thứ hai

Đối với các thành phần NULL mà có chỉ số và đóng vai trò là dependent, thì trước khi khử thành phần NULL, chúng ta phải thay ở vị trí dependent với ngữ mà thành phần NULL trỏ tới. Tuy nhiên đôi khi, ngữ được NULL trỏ tới đã có head nên sẽ vi phạm quy tắc trong cú pháp thành tố là 1 ngữ chỉ được được phép tồn tại một head duy nhất. Do đó, chúng tôi sẽ xem mối quan hệ thay thế NULL sẽ là một đặc trưng phụ và sẽ lưu trữ ở cột 9 trong định dạng CoNLL.

Một ví điển hình khi chuyển cây cú pháp thành tố (hình 4.1), chúng tôi nhận thấy xuất hiện hai thành phần NULL loại * (được tô xanh biển và tô cam). Cả hai thành phần NULL này đều có chỉ số -1 trỏ đến PP-1 (được tô xanh lá). Sau khi chuyển đổi cú pháp thành tố, ta sẽ được cây cú pháp phụ thuộc như hình 4.2. Mục tiêu của chúng tôi bây giờ là thay *-1 ở vị trí dependent trong 2 mối quan hệ PREP(được tô vàng và tím) thành chữ 'như' – head của ngữ mà *-1 trỏ tới, nhưng do từ "như" đã có head là chữ "bé" nên không thể thêm bất kỳ head nào như chữ "nhọn" và "mỏng_dính". Do đó chúng tôi sẽ xem hai mối quan hệ "nhọn → như" và "mỏng_dính → như" là loại quan hệ thứ hai và được lưu ở cột 9 trong định dạng CoNLL-U như bảng 5.1 ở từ có đánh số thứ tự 10.



Hình 4.1 Một cây cú pháp thành tố



Hình 4.2 Một cây cú pháp phụ thuộc

4.1.2 Thêm nhãn chức năng

Chúng tôi cũng xem xét các nhãn chức năng trong cú pháp thành tố như là một đặc trưng phụ và được lưu ở cột 6. Ví dụ cụ thể là hình 4.1 với ngữ ADJP (được tô hồng) được gắn nhãn –MNR nên chúng tôi sẽ lưu nhãn –MNR ở cột 6 định dạng CoNLL-U với từ được đánh số thứ tự 3 như bảng 5.1. Việc lưu nhãn chức năng này thể hiện tính linh hoạt khi có thể thay đổi giữa nhãn cú pháp phụ thuộc và nhãn chức năng. Xét mối quan hệ "ngậm → ngang" (hình 4.2), ta có thể thay thế nhãn ADJP_ADVMOD bằng nhãn MNR.

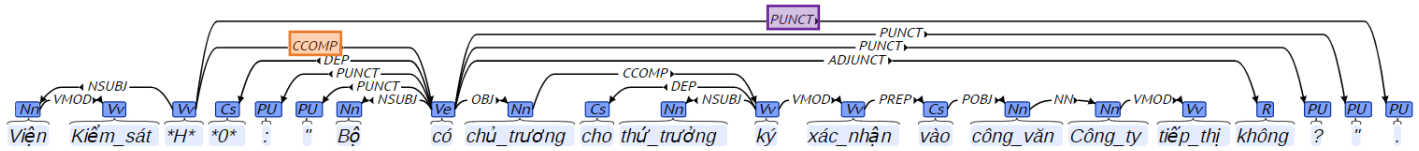
4.2 Khử thành phần NULL

Loại	Mô tả
T	Trace of movement
E	Ellipses without trace
*	Null sentence component as it appears at other position of the sentence
0	Null complementizer
P	Null passive verb
H	Null head word
D	Null post-modifier of a verb
SUM	Null post modifier as it is a part of a sum considered later

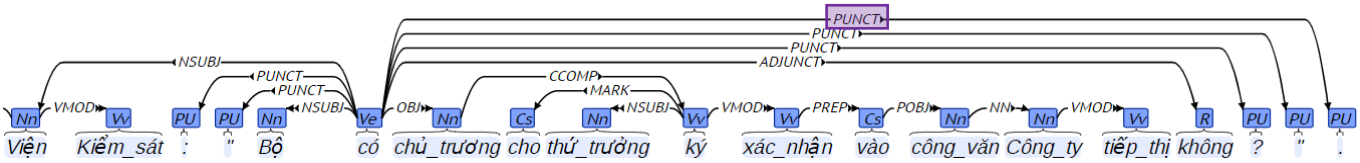
Bảng 4.1 Các loại thành phần NULL trong NIIVTB

Sau khi thêm các đặc trưng phụ, chúng tôi tiến hành loại bỏ thành phần NULL (bảng 4.1). Một vấn đề xảy ra trong quá trình khử thành phần NULL là đôi khi trong mối quan hệ, NULL sẽ làm head. Để xử lý trường hợp này, chúng tôi sẽ tiến hành thay thế thành phần NULL bằng một trong các từ nằm trong mối quan hệ NULL làm head.

Ví dụ điển hình là thành phần NULL *H* (hình 4.3) làm head cho 2 mối quan hệ " *H* → có" (được tô cam) và " *H* → ." (được tô tím). Chúng tôi giải quyết bằng cách chọn từ "có" trong mối quan hệ với *H* làm head và các dependent còn lại có liên quan đến *H* sẽ phụ thuộc vào từ "có" như dấu "." . Kết quả khử nhãn NULL được thể hiện ở hình 4.4



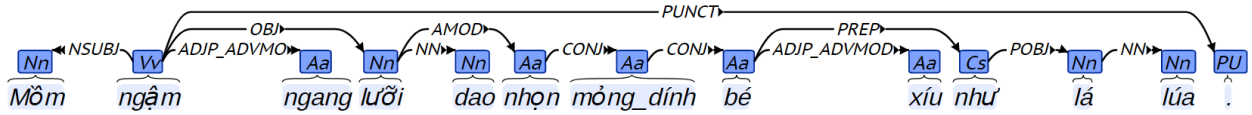
Hình 4.3 Một cây cú pháp phụ thuộc



Hình 4.4 Một cây cú pháp phụ thuộc

5 Lưu cây cú pháp phụ thuộc

Chúng tôi tổ chức lưu dữ liệu cây cú pháp phụ thuộc dưới định dạng CoNLL-U[4]. Mỗi cây cú pháp phụ thuộc sẽ có một mã ID, định dạng gồm 10 cột như hình 5.2 là hình thức lưu trữ dữ liệu cho cây cú pháp phụ thuộc (hình 5.1)



Hình 5.1 Cây cú pháp phụ thuộc

```
# ID = 1
1   Mồm          _   Nn   _   _     2   NSUBJ          _                  _
2   ngậm         _   Vv   _   _     0   ROOT           _                  _
3   ngang        _   Aa   _   MNR   2   ADJP_ADVMOD    _                  _
4   lưỡi         _   Nn   _   _     2   OBJ            _                  _
5   dao          _   Nn   _   _     4   NN             _                  _
6   nhọn         _   Aa   _   _     4   AMOD           _                  _
7   mỏng_dính    _   Aa   _   _     6   CONJ           _                  _
8   bé           _   Aa   _   _     7   CONJ           _                  _
9   xíu          _   Aa   _   _     8   ADJP_ADVMOD    _                  _
10  như          _   Cs   _   _     8   PREP           6:PREP|7:PREP      _
11  lá           _   Nn   _   _     10  POBJ           _                  _
12  lúa          _   Nn   _   _     11  NN             _                  _
13  .            _   PU   _   _     2   PUNCT          _                  _
```

Hình 5.2 Định dạng CoNLL-U của cây cú pháp phụ thuộc

[4] Link tham khảo CoNLL-U: CoNLL-U Format (universaldependencies.org)

6 Tài liệu tham khảo

[1] Q. T. Nguyen, Y. Miyao, H. T. T. Le, and N. T. H. Nguyen, "Ensuring annotation consistency and accuracy for Vietnamese treebank," *Lang Resources & Evaluation*, vol. 52, no. 1, pp. 269–315, Mar. 2018, doi: 10.1007/s10579-017-9398-3.

[2] P.-T. Nguyen, X.-L. Vu, T.-M.-H. Nguyen, V.-H. Nguyen, and H.-P. Le, "Building a large syntactically-annotated corpus of Vietnamese," in *Proceedings of the Third Linguistic Annotation Workshop on - ACL-IJCNLP '09*, Suntec, Singapore, 2009, pp. 182–185. doi: 10.3115/1698381.1698416.

[3] Le-Hong P., Nguyen T. M. H., Nguyen P. T., and Phan T. H., "Automated Extraction of Tree Adjoining Grammars from a Treebank for Vietnamese," p. 17.

[4] J. D. Choi and M. Palmer, "Style Constituent to Dependency Conversion," p. 53.

[5] M. Čmejrek, J. Cuřín, and J. Havelka, "Prague Czech-English Dependency Treebank: Any Hopes for a Common Annotation Scheme?," in *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, Boston, Massachusetts, USA, May 2004, pp. 47–54. Available: https://aclanthology.org/W04-2708

[6] "(PDF) Graph Transformations in Data-Driven Dependency Parsing." https://www.researchgate.net/publication/200179365_Graph_Transformations_in_Data-Driven_Dependency_Parsing.

[7] J. Choi and M. Palmer, "Robust Constituent-to-Dependency Conversion for English," Dec. 2010.

[8] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," p. 28.

[9] H. Leung, R. Poiret, T. Wong, X. Chen, K. Gerdes, and J. Lee, "Developing Universal Dependencies for Mandarin Chinese," in *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)*, Osaka, Japan, Dec. 2016, pp. 20–29. Available: https://aclanthology.org/W16-5403

[10] T. Tanaka *et al.*, "Universal Dependencies for Japanese," p. 8.

A Nhãn cú pháp thành tố

A.1 Kho ngữ liệu Vietnamse Treebank

POS tags			
N	Noun	R	Adjunct
Np	Proper noun	L	Determiner

Nc	Classifier noun	M	Quantity
Nu	Unit noun	E	Preposition
Ny	Abbreviated noun	C	Conjunction
Nb	Borrowed noun	I	Exclamation
V	Verb	T	Particle
Vb	Borrowed verb	Y	Abbreviation
A	Adjective	S	Affix
P	Pronoun	X	Un-definition/Other

* Các loại dấu câu: **. , ; : - " /** và **...**

* Hai nhãn LBKT, RBKT lần lượt là: left bracket và right bracket

Clause level tags	
S	Sentence
SQ	Question
SBAR	Subordinate clause

Phrase level tags			
NP	Noun phrase	YP	Abbreviation phrase
VP	Verb phrase	WHNP	Wh-noun phrase
AP	Adjective phrase	WHAP	Wh-adjective phrase
RP	Adjunct phrase	WHRP	Wh-adjunct phrase
PP	Prepositional phrase	WHPP	Wh-prepositional phrase
QP	Quantity phrase	WHVP	Wh-verb phrase
MDP	Modal phrase	WHXP	Wh-undefined phrase
UCP	Unlike coordinated phrase	XP	Undefined phrase

Function tags			
H	Head	TMP	Temporal
SUB	Subject	LOC	Location
DOB	Direct object	DIR	Direction
IOB	Indirect object	MNR	Manner
TPC	Topicalization	PRP	Purpose
PRD	Predicate	CND	Condition
EXT	Extent	CNC	Concession

VOC	Vocatives	ADV	Adverbial

A.2 Kho ngữ liệu NIIVTB

POS tags			
Nc	Classifier noun	An	Ordinal number
Ncs	Special classifier noun	Aa	Other adjectives
Nu	Unit noun	Pd	Demonstrative pronoun
Nun	Special unit noun	Pp	Other pronouns
Nt	Noun of time	R	Adjunc
Nq	Nouns imply the quantity but not numbers	Cs	Prepositions and conjunctions introducing a SBAR
Num	Quantifiers can write by numbers	Cp	Other conjunctions
Nw	Quantifiers indicating the whole	NA	Noun-Adjective
Nr	Proper noun	VN	Noun-Verb
Nn	Other nouns	VA	Verb-Adjective
Ve	Existing verb	ON	Onomatopoeia
Vc	Copula “là” verb	PU	Punctuation
D	Directional co-verb	ID	Idioms
Vcp	Comparative verb	E	Exclamation word
Vv	Other verbs	M	Modifier word
Sv	Sino-Vietnamese	FW	Foreign word
X	Unidentified words	LBRK	Left bracket
RBRK	Right bracket		

Internal structure tags			
Nn_w	A combination of a Sino-Vietnamese and a noun	Vv_swsp_Rt	Repetition form of verb
Vv_w	A combination of a Sino-Vietnamese and a verb	Aa_swsp_Rt	Repetition form of adjective

Aa_w	A combination of a Sino-Vietnamese and a adjective	Nn_swsp_Rt	Repetition form of noun
R_w	A combination of a Sino-Vietnamese and a adjunct	ON_swsp_Rt	Repetition form of sound
Nn_swsp	Noun sub-phrase		

Clause level tags	
S	Sentence
SQ	Question
SBAR	Subordinate clause
SPL	Special sentence

Phrase level tags			
NP	Noun phrase	UCP	Unlike coordinated phrase
VP	Verb phrase	QNP	Wh-noun phrase
ADJP	Adjective phrase	QADJP	Wh-adjective phrase
RP	Adjunct phrase	QRP	Wh-adjunct phrase
PP	Prepositional phrase	QPP	Wh-prepositional phrase
QP	Quantity phrase	QVP	Wh-verb phrase
MDP	Modal phrase	CONJP	Conjunction phrase

Function tags			
H	Head	TMP	Temporal
SBJ	Subject	LOC	Location
DOB	Direct object	LGS	Logical subject
IOB	Indirect object	MNR	Manner
TPC	Topicalization	PRP	Purpose
PRD	Predicate that is not VP	CND	Condition
EXC	Exclamative sentence	CNC	Concession
VOC	Vocatives	ADV	Adverbial
CMP	Complement	MDP	Modal phrase
PRN	Parenthetical	HLN	Headline
TTL	Title	CMD	Imperative sentence