Part of Speech Tagging

Lê Thanh Hương

School of Information and Communication Technology - HUST

Email: huonglt@soict.hust.edu.vn

Definition

- Part of Speech (POS) tagging: assign each word in a sentence with an appropriate POS.
  - Input: a string of words + a tagset
  - Output: a best tag for each word

Example 1
Example 2
Example 3
Example 4
Example 5

➢ Tagging makes parsing easier

Why POS tagging?

- **Simple**: can be done by many different methods
  - Can be done well with methods that look at local context
  - Though should “really” do it by parsing!
- **Applications**:
  - Text-to-speech: record - N: [‘reko:d], V: [ri’ko:d]; lead – N [led], V: [li:d]
  - Can be a preprocessor for a parser. The parser can do it better but more expensive
  - Speech recognition, parsing, information retrieval, etc.
- **Easy to evaluate** (*how many tags are correct?*)

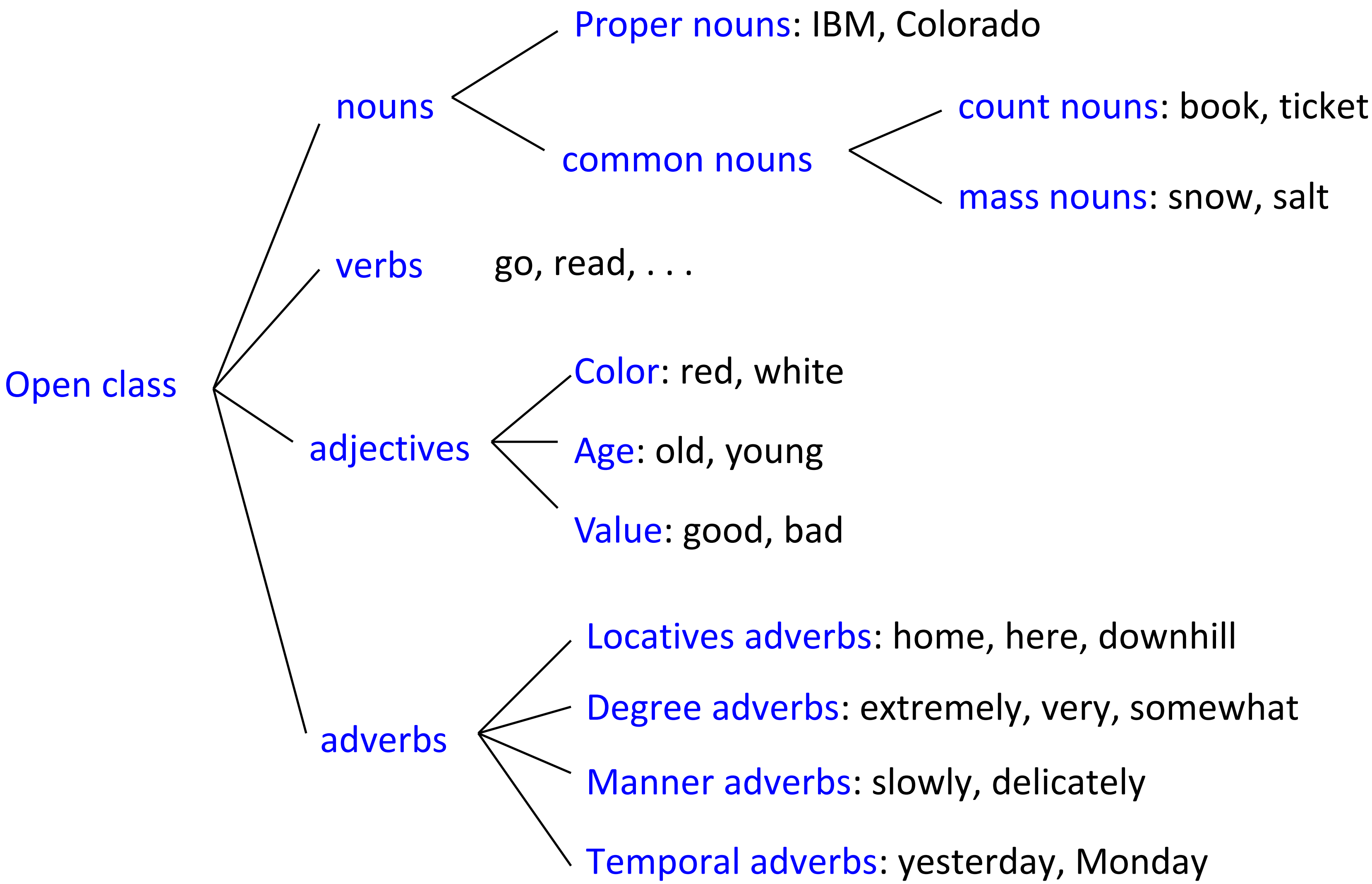
Current Performance

- How many tags are correct?
  - About 97% currently
  - But baseline is already 90%
    - Baseline is performance of stupidest possible method
    - Tag every word with its most frequent tag
    - Tag unknown words as nouns

English POS tag set

- Closed class words:
  - Relatively fixed membership
  - Usually **function** words: short, frequent words with grammatical function
  - Prepositions(Giới từ) : on, under, over,…
  - Particles(Tiểu từ) : abroad, about, around, before, in, instead, since, without,…
  - Articles(Mạo từ) : a, an, the
  - Conjunctions(Liên từ) : and, or, but, that,…
  - Pronouns(Đại từ) : you, me, I, your, what, who,…
  - Auxiliary verbs(Trợ động từ) : can, will, may, should,…
- Open class words
  - Usually **content** words: Nouns, Verbs, Adjectives, Adverbs

English word classes



Open class

nouns

Proper nouns: IBM, Colorado

common nouns

count nouns: book, ticket

mass nouns: snow, salt

verbs go, read, . . .

adjectives

Color: red, white

Age: old, young

Value: good, bad

adverbs

Locatives adverbs: home, here, downhill

Degree adverbs: extremely, very, somewhat

Manner adverbs: slowly, delicately

Temporal adverbs: yesterday, Monday

Tagsets for English

- 87 tags - Brown corpus
- Three most commonly used:
  - ➢ Small: 45 Tags - Penn treebank (next slide)
  - ➢ Medium size: 61 tags, British national corpus
  - ➢ Large: 146 tags, C7

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	“to”	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential ‘there’	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one’s*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	“to”	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential ‘there’	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one’s*	)	Right parenthesis	( ],), }, >)
RB	Adverb				
RBR	Adverb, cor				
RBS	Adverb, sup				
RP	Particle				

I know that blocks the sun.
He always books the violin concert tickets early.
He says that book is interesting.

Example from Penn Treebank

- The grand jury commented on a number of other topics.

⇨ The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

How difficult is POS tagging in English?

- Roughly 15% of word types are ambiguous
- Hence 85% of word types are unambiguous
- *Janet* is always PROPN, *hesitantly* is always ADV
- But those 15% tend to be very common.

➢Problem of POS tagging is to resolve ambiguities, choosing the proper tag for the context.

Main types of taggers

- **Stochastic tagging**: Maximum likelihood, Hidden Markov model tagging

  Pr (Det-N) > Pr (Det-Det)

- **Rule based tagging**

  If <some pattern>

  Then … <some part of speech>

Approaches to Tagging

- **HMM tagging:** 'Use all the information you have and guess’

- **Constrain Grammar (CG) tagging:** 'Don't guess, just eliminate the impossible!’

- **Transformation-based (TB) tagging:** 'Guess first, then change your mind if nessessary!'

Stochastic POS tagging

For a given sentence or word sequence, pick the most likely tag for each word.

How?

- A Hidden Markov model (HMM) tagger:

  Choose the tag sequence that maximizes:

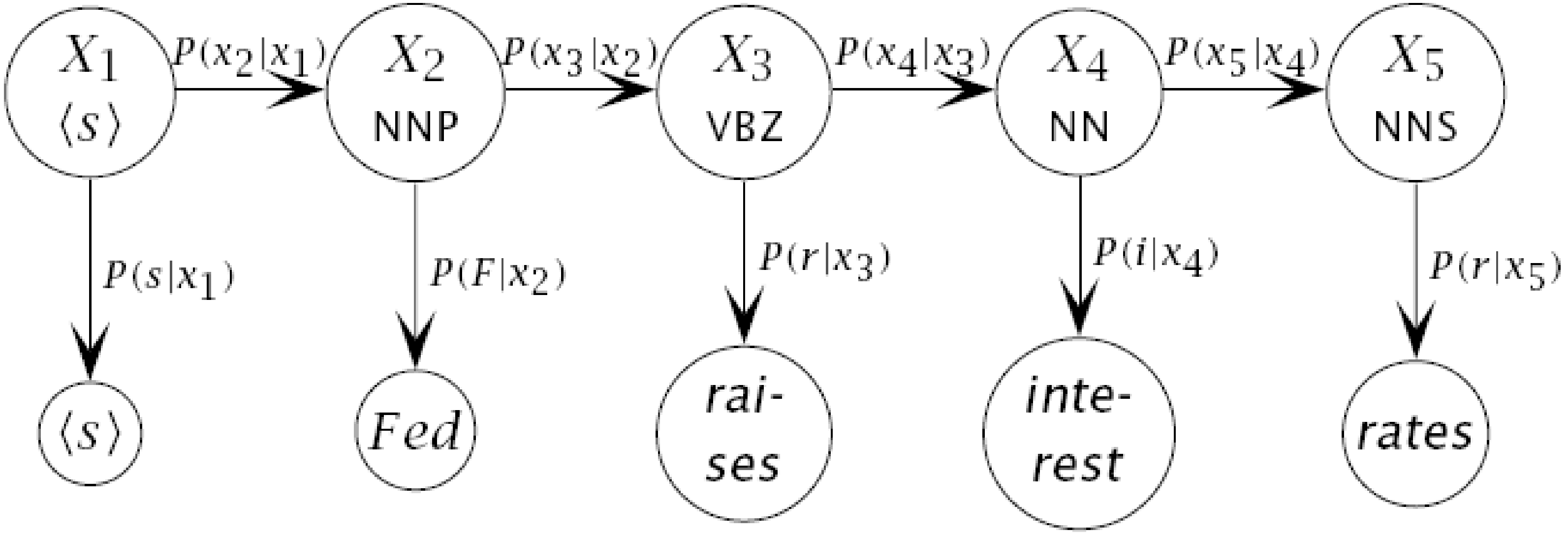
  P(word|tag)•*P*(tag|previous *n* tags)

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

$\Rightarrow$ P(jury|NN) = 1/2

HMMs – POS example

Do supervised training, and then inference to decide POS tags (Bayesian network style)



HMM tagging

- **Bigram HMM Equation**: choose t_i for w_i that is most probably given t_{i-1} and w_i :

  $$t_i = \text{argmax}_j\ P(t_j \mid t_{i-1}, w_i) \qquad (1)$$

- **A HMM simplifying assumption**: the tagging problem can be solved by looking at nearby words and tags.

$$t_i = \text{argmax}_j\ P(t_j \mid t_{i-1})P(w_i \mid t_j) \qquad (2)$$

pr tag sequence (tag co-occurrence)

word (lexical) likelihood

Example

1. Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NN

2. People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN

Suppose we have tagged all but race

- Look at just preceding word (bigram):

  to/TO race/???   NN or VB?

  the/DT race/???

- Applying (2): $t_i = \text{argmax}_j \, P(t_j \mid t_{i-1}) P(w_i \mid t_j)$

- Choose tag with greater of the two probabilities:

  P(VB|TO)P(race|VB) or P(NN|TO)P(race|NN)

I/PP know/VBP that/WDT block/NN blocks/NNS?VBZ? the/DT sun/NN.

Calculate Probabilities

Let's consider *P*(VB|TO) and *P*(NN|TO)

- From the Brown corpus

  P(NN|TO)= .021

  P(VB|TO)= .340

  P(race|NN)= 0.00041

  P(race|VB)= 0.00003

- *P*(VB|TO)*P*(race|VB) = 0.00001
- *P*(NN|TO)*P* (race|NN) = 0.000007

➢ *race should be a* VB *after* "TO"

Exercise

$t_i = \text{argmax}_j\, P(t_j \mid t_{i-1}) P(w_i \mid t_j)$

- I know that blocks the sun.
- He always books the violin concert tickets early.
- He says that book is interesting.
  - I/PP know/VBP that/WDT blocks/NNS block/VBP the/DT sun/NN.
  - I/PP know/VBP that/WDT blocks/VBZ the/DT sun/NN.
  - He/PP always/RB books/VBZ the/DT violin/NN concert/NN tickets/NNS early/RB.
  - He/PP says/VBZ that/WDT book/NN is/VBZ interesting/JJ.
  - I know that block blocks the sun.
  - I/PP know/VBP that/DT block/NN blocks/NNS?VBZ? the/DT sun/NN.
- I/PP know/VBP that/WDT block/NN blocks/VBZ the/DT sun/NN.

The full model

- We want the best *sequence* of tags for the whole sentence
- Given the sequence of words, *W*, we want to compute the most probably tag sequence, $T=t_1, t_2, \ldots, t_n$ or,

$$\hat{T} = \arg\max_{T \in \tau} P(T \mid W)$$

(Bayes' Theorem)

$$= \arg\max_{T \in \tau} \frac{P(T)P(W \mid T)}{P(W)}$$

$$= \arg\max_{T \in \tau} P(T)P(W \mid T)$$

Expand this using chain rule

From chain rule for probabilities:

P(A,B) = P(A|B)P(B) = P(B|A)P(A)

P(A,B,C) = P(B,C|A)P(A) = P(C|A,B)P(B|A)P(A)
= P(A)P(B|A)P(C|A,B)

P(A,B,C,D…) = P(A)P(B|A)P(C|A,B)P(D|A,B,C..)

$$P(T)P(W\mid T)=\prod_{i=1}^{n}\underbrace{P(w_i\mid w_1t_1\ldots w_{i-1}t_{i-1}t_i)}_{\text{pr word}}\underbrace{P(t_i\mid w_1t_1\ldots w_{i-1}t_{i-1})}_{\text{tag history}}$$

Trigram assumption

- Probability of a word depends only on its tag

$$P(w_i \mid w_1 t_1 \ldots t_{i-1} t_i) = P(w_i \mid t_i)$$

- Tag history approximated by two most recent tags (trigram: two most recent + current state)

$$P(t_i \mid w_1 t_1 \ldots t_{i-1}) = P(t_i \mid t_{i-2} t_{i-1})$$

Replacing to the equation

P(T)P(W|T) =

$$P(t_1)P(t_2 \mid t_1)\prod_{i=3}^{n} P(t_i \mid t_{i-2}t_{i-1})[\prod_{i=1}^{n} P(w_i \mid t_i)]$$

Estimate Probabilities

- Use relative frequencies from corpus to estimate these probabilities:

$$P(t_i \mid t_{i-1}t_{i-2}) = \frac{c(t_{i-2}t_{i-1}t_i)}{c(t_{i-2}t_{i-1})}$$

$$P(w_i \mid t_i) = \frac{c(w_i, t_i)}{c(t_i)}$$

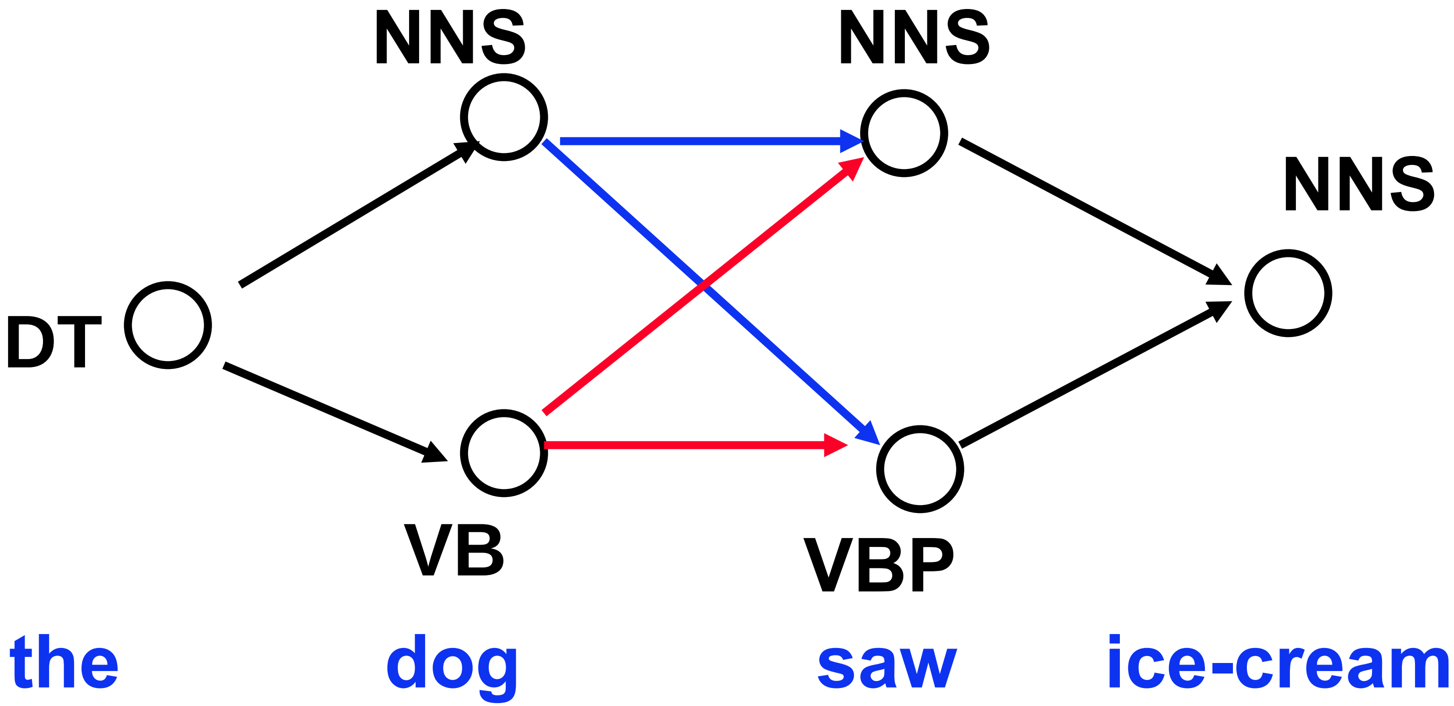
Problem

The problem to solve:

$$\hat{T} = \arg\max_{T \in \tau} P(T)P(W \mid T)$$

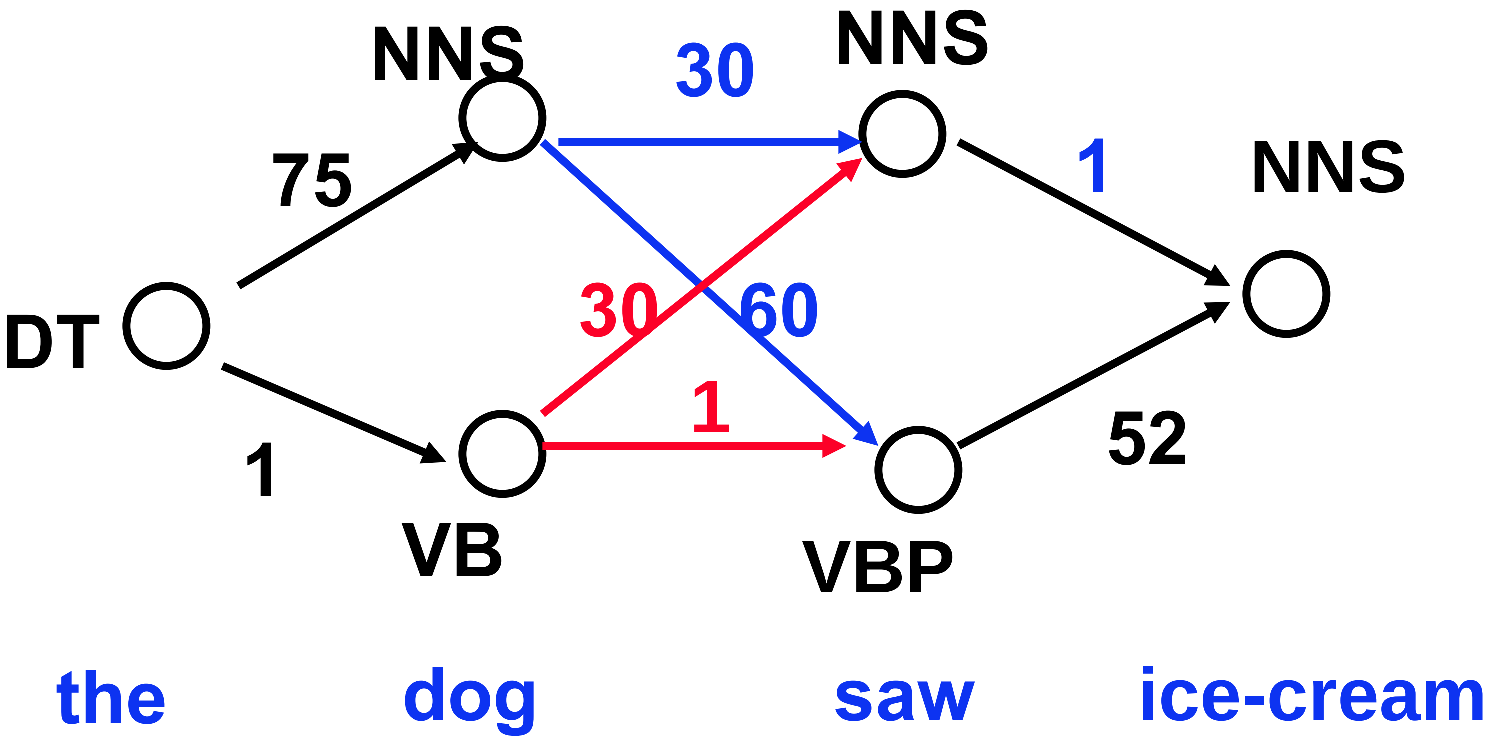
All P(T)P(W|T) can now be computed

Example



How do we find best path?

The counts add scores - we want to find the maximum scoring path



How do we find maximum (best) path?

- We use beam search, as in AI…

1. At each step, k best values ( ) are chosen. Each of the k values corresponds to one possible tagging combination of the visited words. $\hat{T}$
2. When tagging the next word, recompute probabilities. Go to step 1.

- **Advantage**: fast (do not need to check all possible combinations, but only k potential ones).
- **Disadvantage**: may not return the best solution, but only acceptable results.

Accuracy

- Accuracy of this method > 96%
- Baseline? **90%**
    - Baseline is performance of stupidest possible method
    - Tag every word with its most frequent tag
    - Tag unknown words as nouns
- Human: **97%+/- 3%**; if discuss together: **100%**

Suppose we don’t have training data

- Can estimate roughly:
  - start with uniform probabilities,
  - use Expectation Maximization (EM) algorithm to re-estimate from counts
  - try labeling with current estimate
  - use this to correct estimate

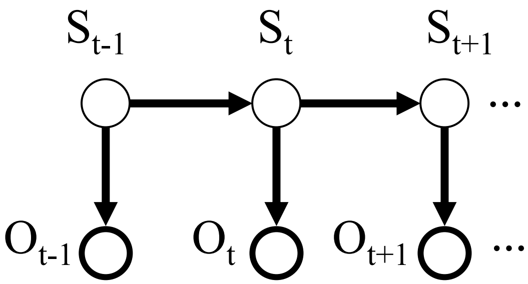
➢ Not work well, a small amount of hand-tagged training data improves the accuracy

POS tagging with CRF

[McCallum, Freitag & Pereira, 2000]
[Lafferty, McCallum, Pereira 2001]

$\vec{s} = s_1, s_2, \ldots s_n$ $\qquad$ $\vec{o} = o_1, o_2, \ldots o_n$

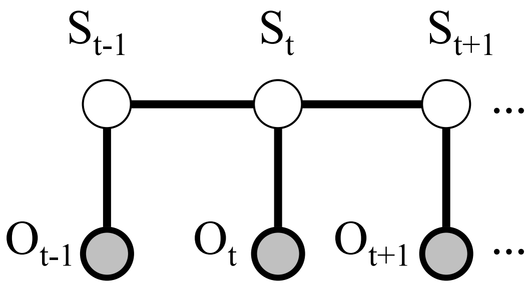
Joint $\qquad P(\vec{s},\vec{o}) = \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$



Conditional

$$P(\vec{s} \mid \vec{o}) = \frac{1}{P(\vec{o})} \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$$

$$= \frac{1}{Z(\vec{o})} \prod_{t=1}^{|\vec{o}|} \Phi_s(s_t, s_{t-1}) \Phi_o(o_t, s_t)$$



(A special case of Conditional Random Fields.)

with $\qquad \Phi_o(t) = \exp\left(\sum_k \lambda_k \boxed{f_k(s_t, o_t)}\right)$

Random features of s,o, and t

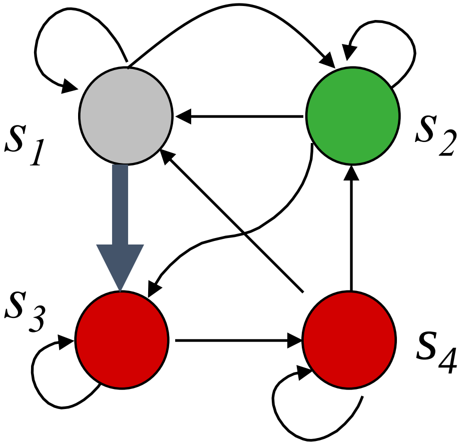
Feature function

Eg. $f_k(s_t, s_{t-1}, \vec{o}, t)$:

$$f_{<\text{Capitalize d}, s_i, s_j>}(s_t, s_{t-1}, \vec{o}, t) = \begin{cases} 1 & \text{if Capitalize d}(o_t) \wedge \mathrm{s_i} = s_{t-1} \wedge \mathrm{s_j} = s_t \\ 0 & \text{otherwise} \end{cases}$$

$\overline{o}$ = Yesterday Pedro Domingos spoke this example sentence.

o_1 o_2 o_3 o_4 o_5 o_6 o_7

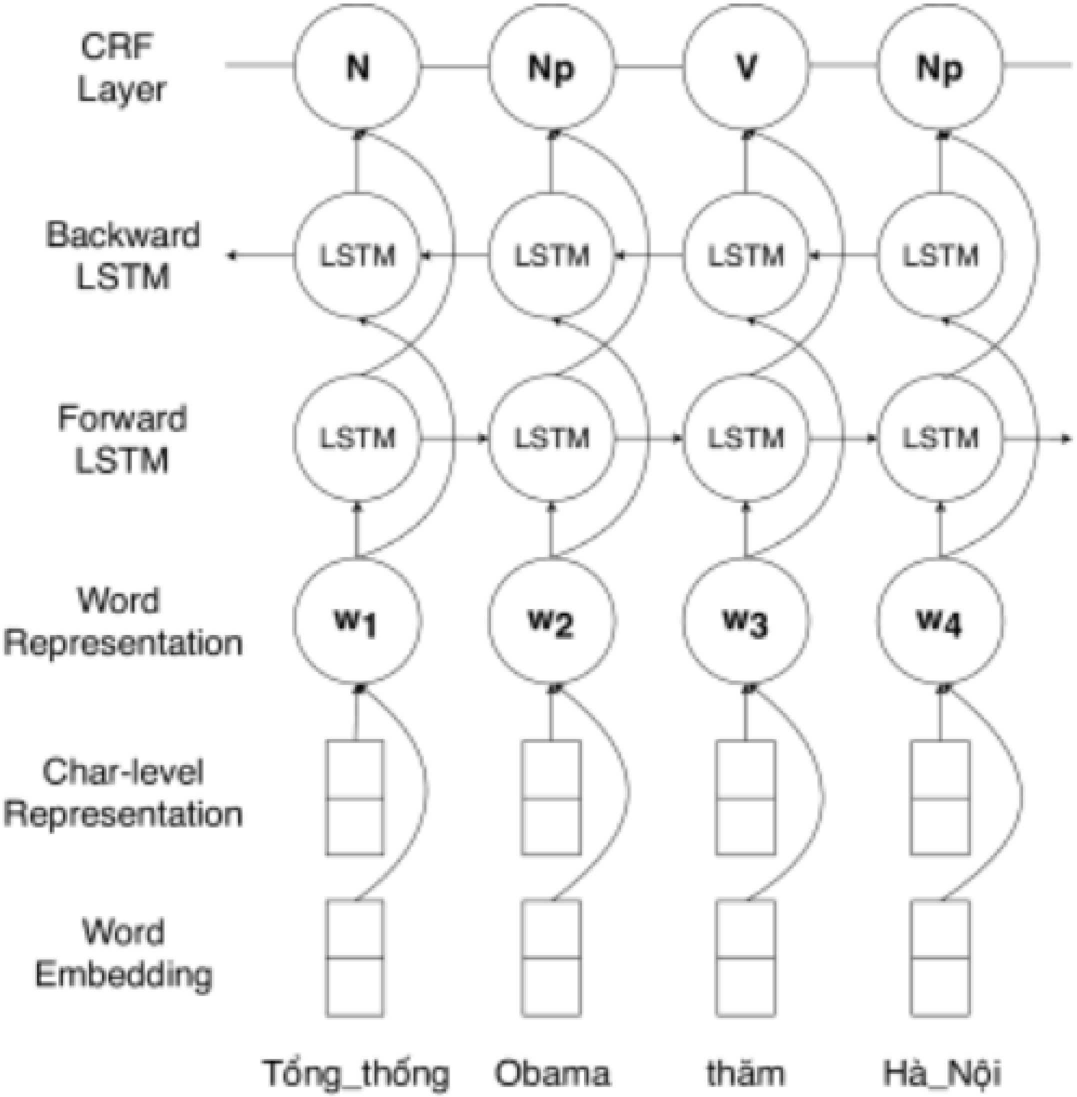


$$f_{<Capitalized, s_1, s_2>}(s_2, s_1, \vec{o}, 2) = 1$$

Features

- Is all capitalized
- Is initial capitalized
- Is a number
- Is a special character
- End with “ing”
- In the location dictionary
- …

POS tagging with biLSTM + CRF



Second approach: transformation-based tagging

Transformation-based Learning (TBL):

- Combines symbolic and stochastic approaches: uses machine learning to refine its tags, via several passes

- Tag using a broadest (most general) rule; then an narrower rule, that changes a smaller number of tags, and so on.

How does the TBL system work?

rules

```
pos:NN>VB <- pos:TO@[-1] o
pos:VB>NN <- pos:DT@[-1] o
....
```

lexicon

```
data:NN
decided:VB
her:PN
she:PN N
table:NN VB
to:TO
```

input

```
She decided to  table her data
NP  VB       TO VB    PN  NN
```

How does the TBL system work?

1. Label every word with its most-likely tag (often 90% right). From Brown corpus:

   P(NN|race)= 0.98

   P(VB|race)= 0.02

2. …expected/VBZ to/TO race/VB tomorrow/NN

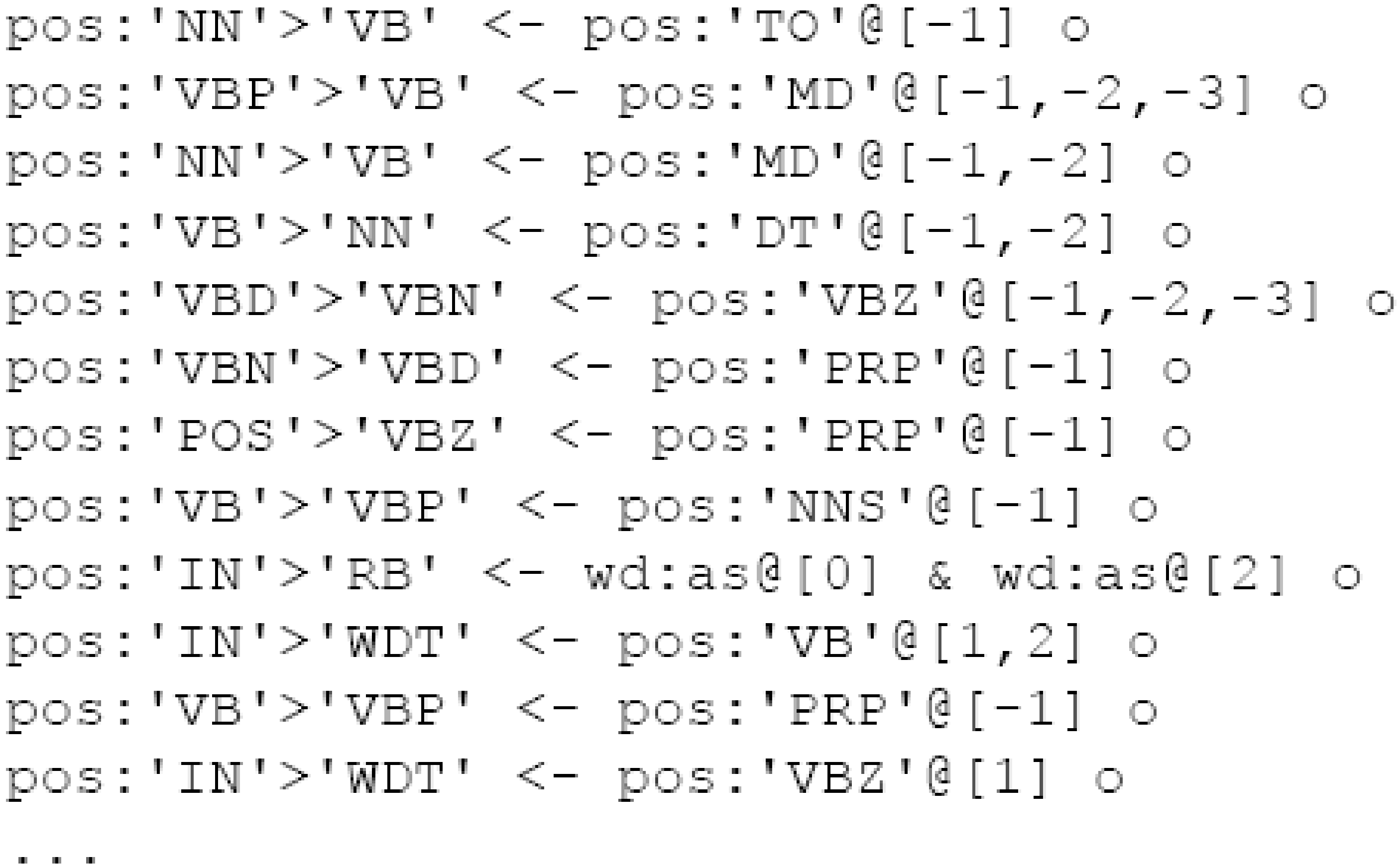
   …the/DT race/NN for/IN outer/JJ space/NN

3. Use transformational (learned) rules:

   Change **NN** *to* **VB** *when the previous tag is* **TO**

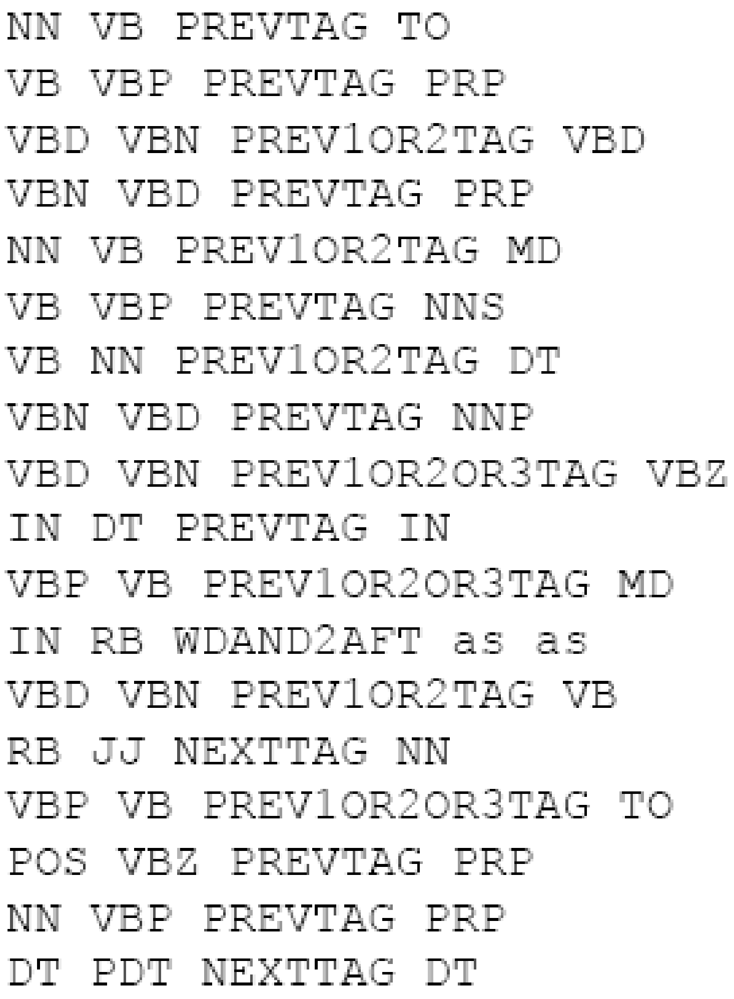
   pos: 'NN'>'VB' ← pos: 'TO' @[-1] o

Rules for POS tagging



```
pos:'NN'>'VB' <- pos:'TO'@[-1] o
pos:'VBP'>'VB' <- pos:'MD'@[-1,-2,-3] o
pos:'NN'>'VB' <- pos:'MD'@[-1,-2] o
pos:'VB'>'NN' <- pos:'DT'@[-1,-2] o
pos:'VBD'>'VBN' <- pos:'VBZ'@[-1,-2,-3] o
pos:'VBN'>'VBD' <- pos:'PRP'@[-1] o
pos:'POS'>'VBZ' <- pos:'PRP'@[-1] o
pos:'VB'>'VBP' <- pos:'NNS'@[-1] o
pos:'IN'>'RB' <- wd:as@[0] & wd:as@[2] o
pos:'IN'>'WDT' <- pos:'VB'@[1,2] o
pos:'VB'>'VBP' <- pos:'PRP'@[-1] o
pos:'IN'>'WDT' <- pos:'VBZ'@[1] o
...
```

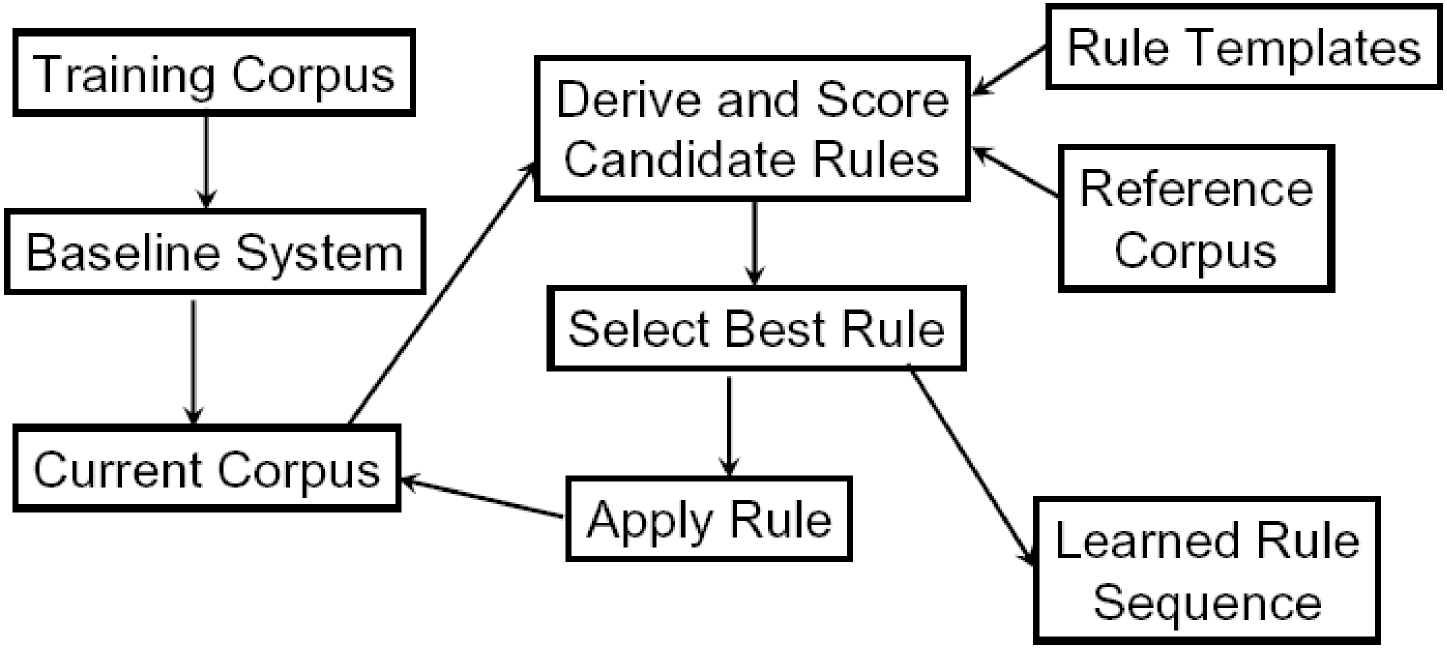
Rules for POS tagging



```
NN VB PREVTAG TO
VB VBP PREVTAG PRP
VBD VBN PREV1OR2TAG VBD
VBN VBD PREVTAG PRP
NN VB PREV1OR2TAG MD
VB VBP PREVTAG NNS
VB NN PREV1OR2TAG DT
VBN VBD PREVTAG NNP
VBD VBN PREV1OR2OR3TAG VBZ
IN DT PREVTAG IN
VBP VB PREV1OR2OR3TAG MD
IN RB WDAND2AFT as as
VBD VBN PREV1OR2TAG VB
RB JJ NEXTTAG NN
VBP VB PREV1OR2OR3TAG TO
POS VBZ PREVTAG PRP
NN VBP PREVTAG PRP
DT PDT NEXTTAG DT
```

…

Learning TB rules in TBL system



Stop when score of best rule falls below threshold.

Various Corpora

- Training corpus

  w0 w1 w2 w3 w4 w5 w6 w7 w8 w9 w10

- Current corpus (CC 1)

  dt vb nn dt vb kn dt vb ab dt vb

- Reference corpus

  dt nn vb dt nn kn dt jj kn dt nn

Rule Templates

- In TBL, only rules that are instances of *templates* can be learned.
- For example, the rules

  tag:'VB'>'NN' ← tag:'DT'@[-1].

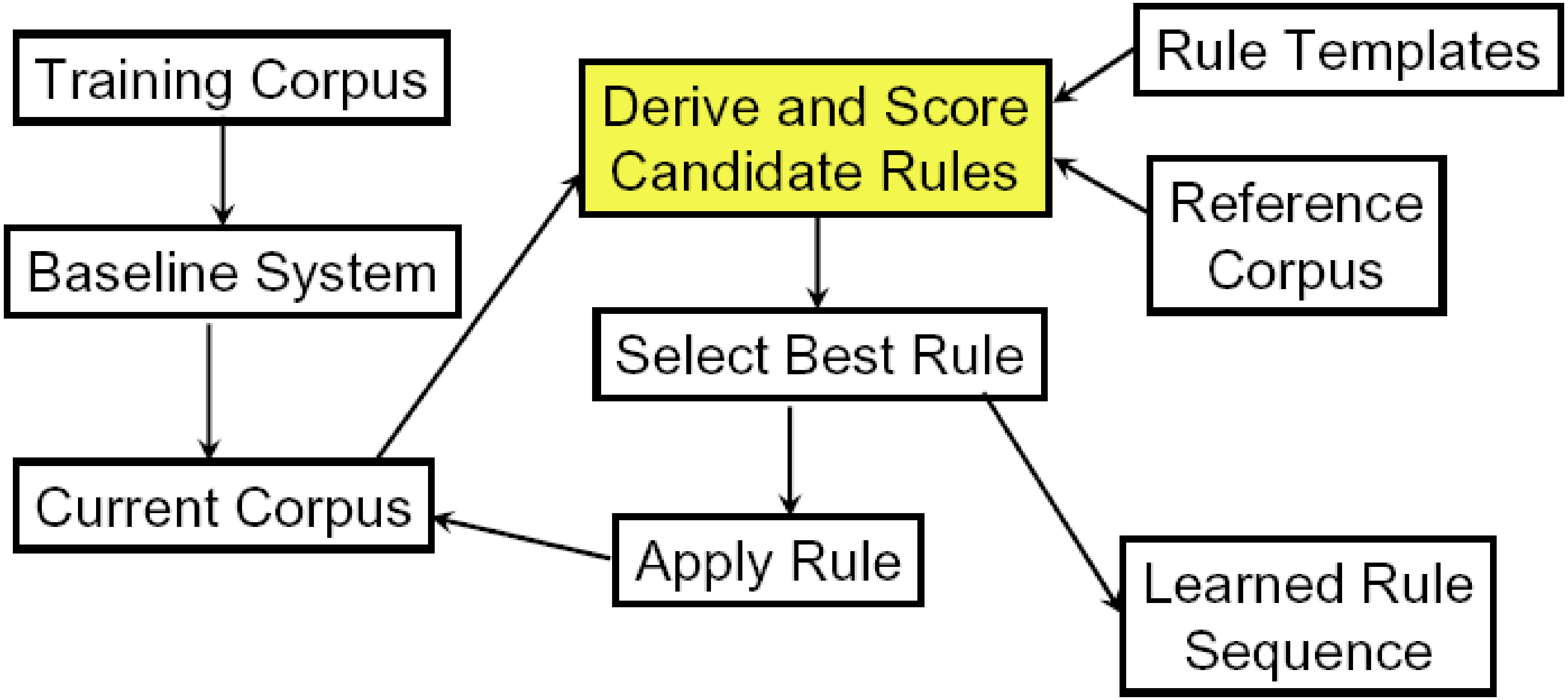
  tag:’NN’>’VB' ← tag:'DT'@[-1].

  are instances of the template

  tag:A>B ← tag:C@[-1].
- Alternative syntax using anonymous variables

  tag:_>_ ← tag:_@[-1].

Learning TB rules in TBL system



Training Corpus

Baseline System

Current Corpus

Derive and Score Candidate Rules

Rule Templates

Reference Corpus

Select Best Rule

Apply Rule

Learned Rule Sequence

Score, Accuracy and Thresholds

- The *score* of a rule:

$$\text{score}(R) = |\text{pos}(R)| - |\text{neg}(R)|$$

- The *accuracy* of a rule:

$$\text{accuracy}(R) = \frac{|\text{pos}(R)|}{|\text{pos}(R)| + |\text{neg}(R)|}$$

- *Threshold*: the value that a rule must have in order to be considered.
- In *ordinary* TBL, use accuracy threshold < 0.5.

Derive and Score Candidate Rule 1

- Template = tag:_>_ ← tag:_@[-1]
- R1 = tag:vb>nn ← tag:dt@[-1]

CC i	dt	vb	nn	dt	vb	kn	dt	vb	ab	dt	vb
CC i+1	dt	nn	nn	dt	nn	kn	dt	nn	ab	dt	nn

Ref. C	dt	nn	vb	dt	nn	kn	dt	jj	kn	dt	nn

- pos(R1) = 3
- neg(R1) = 1
- score(R1) = pos(R1) - neg(R1) = 3-1 = 2

Derive and Score Candidate Rule 2

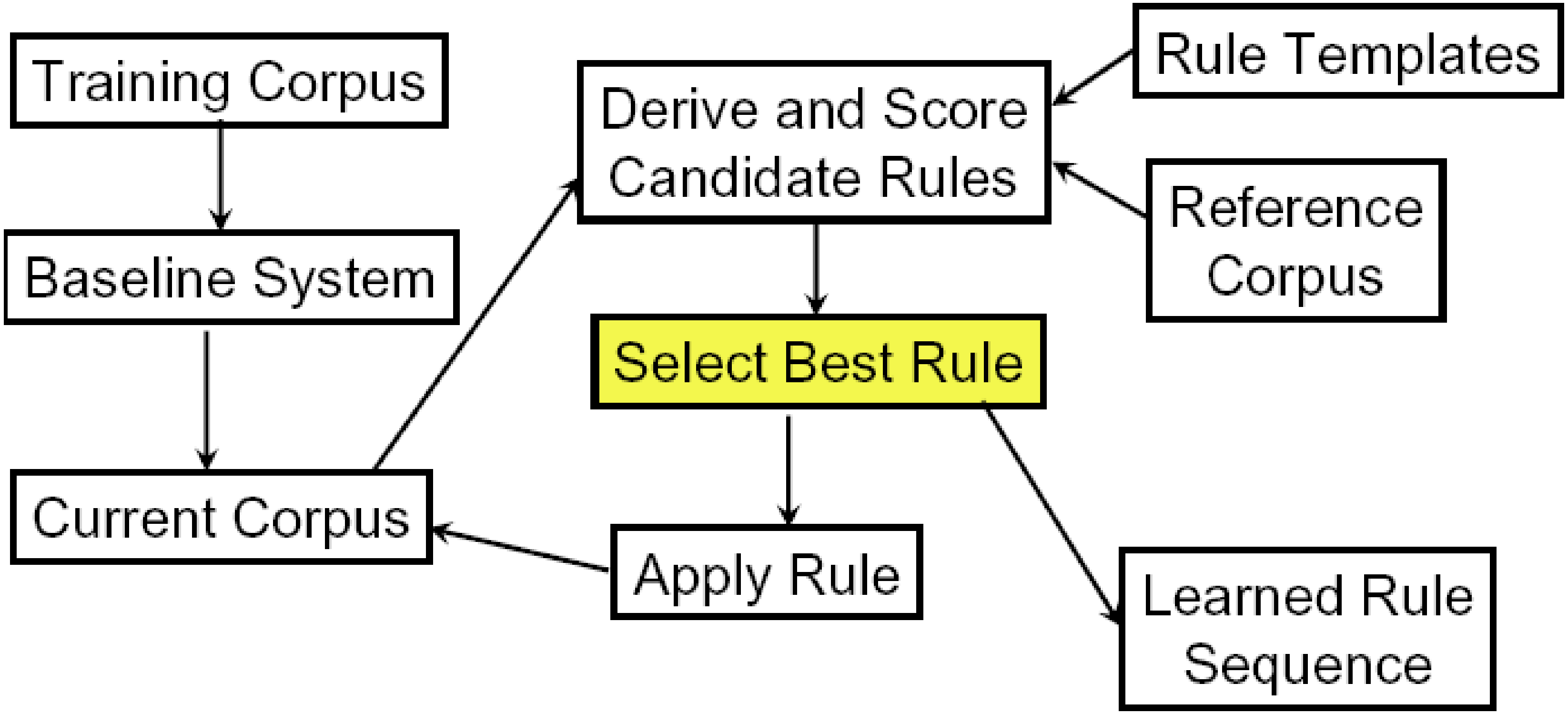
- Template = tag:_>_ ← tag:_@[-1]
- R2 = tag:nn>vb ← tag:vb@[-1]

CC i	dt	**vb**	**nn**	dt	vb	kn	dt	vb	ab	dt	vb
CC i+1	dt	vb	**vb**	dt	vb	kn	dt	vb	ab	dt	vb

Ref. C	dt	nn	vb	dt	nn	kn	dt	nn	kn	dt	nn

- pos(R2) = 1
- neg(R2) = 0
- score(R2) = pos(R2) - neg(R2) = 1-0 = 1

Learning TB rules in TBL system



Stop when score of best rule falls below threshold.

Select Best Rule

- Current ranking of rule candidates

  R1 = tag:vb>nn ← tag:dt@[-1] Score = 2

  R2 = tag:nn>vb ← tag:vb@[-1] Score = 1

  …
- If score threshold =< 2 then select R1
- else if score threshold > 2, terminate.

Select Best Rule Optimizations

- **Reduce redundance rules**: only generate candidate rules that have at least one match in the training data.
- **Incremental evaluation**:
  - Keep track of the leading rule candidate.
  - Ignore rules that has #positive matches < score of the leading rule

Greedy Best-First Search

Evaluation function

h(n) = estimated cost of the cheapest path from the state represented by the node n to a goal state

Advantages of TB Tagging

- Rules can be created/edited manually
- Rules have a declarative, logical semantics
- Simple to implement
- Can be extremely fast (but implementation is more complex)

Error analysis: what's hard for taggers

Common errors (> 4%)

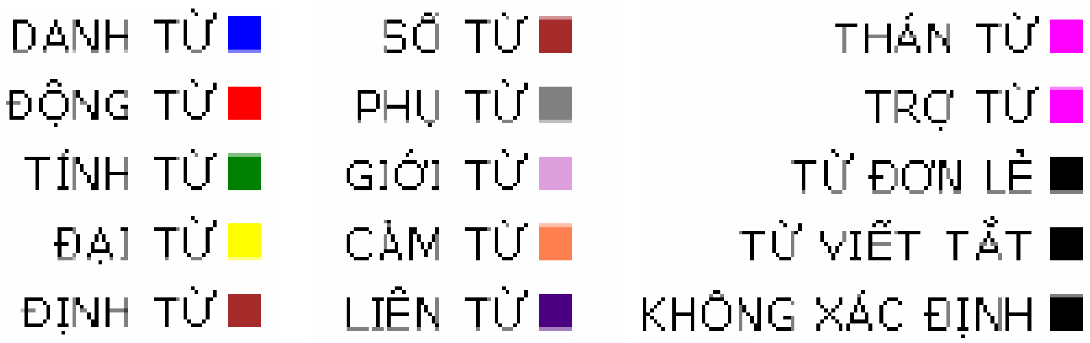
- NN (common noun) vs .NNP (proper noun) vs. JJ (adjective): hard to distinguish; important to distinguish especially for information extraction
- RP(particle) vs. RB(adverb) vs. IN(preposition): all can appear in sequences immediate after verb
- VBD vs. VBN vs. JJ: distinguish past tense, past participles, adjective (*raced* vs. *was raced* vs. *the out raced horse*)

Most powerful unknown word detectors

- 3 inflectional endings (*-ed, -s, -ing*); 32 derivational endings (*-ion,* etc.); capitalization; hyphenation

- More generally:
  - Morphological analysis
  - Machine learning approaches

English POS tagging

Input sentence	Qua những lần từ Sài_Gòn về Quảng_Ngãi kiểm_tra công_việc , Sophie và Jane thường trò_chuyện với Mai , cảm_nhận ngọn_lửa_sống và niềm_tin mãnh_liệt từ người phụ_nữ VN này .
Input sentence with POS tags	Qua những lần từ Sài_Gòn về Quảng_Ngãi kiểm_tra công_việc , Sophie và Jane thường trò_chuyện với Mai , cảm_nhận ngọn_lửa_sống và niềm_tin mãnh_liệt từ người phụ_nữ VN này .
Definition	 DANH TỪ 🟦 ĐỘNG TỪ 🟥 TÍNH TỪ 🟩 ĐẠI TỪ 🟨 ĐỊNH TỪ 🟫 SỐ TỪ 🟫 PHỤ TỪ ▩ GIỚI TỪ 🟪 CẢM TỪ 🟧 LIÊN TỪ 🟪 THÁN TỪ 🟪 TRỢ TỪ 🟪 TỪ ĐƠN LẺ ⬛ TỪ VIẾT TẮT ⬛ KHÔNG XÁC ĐỊNH ⬛

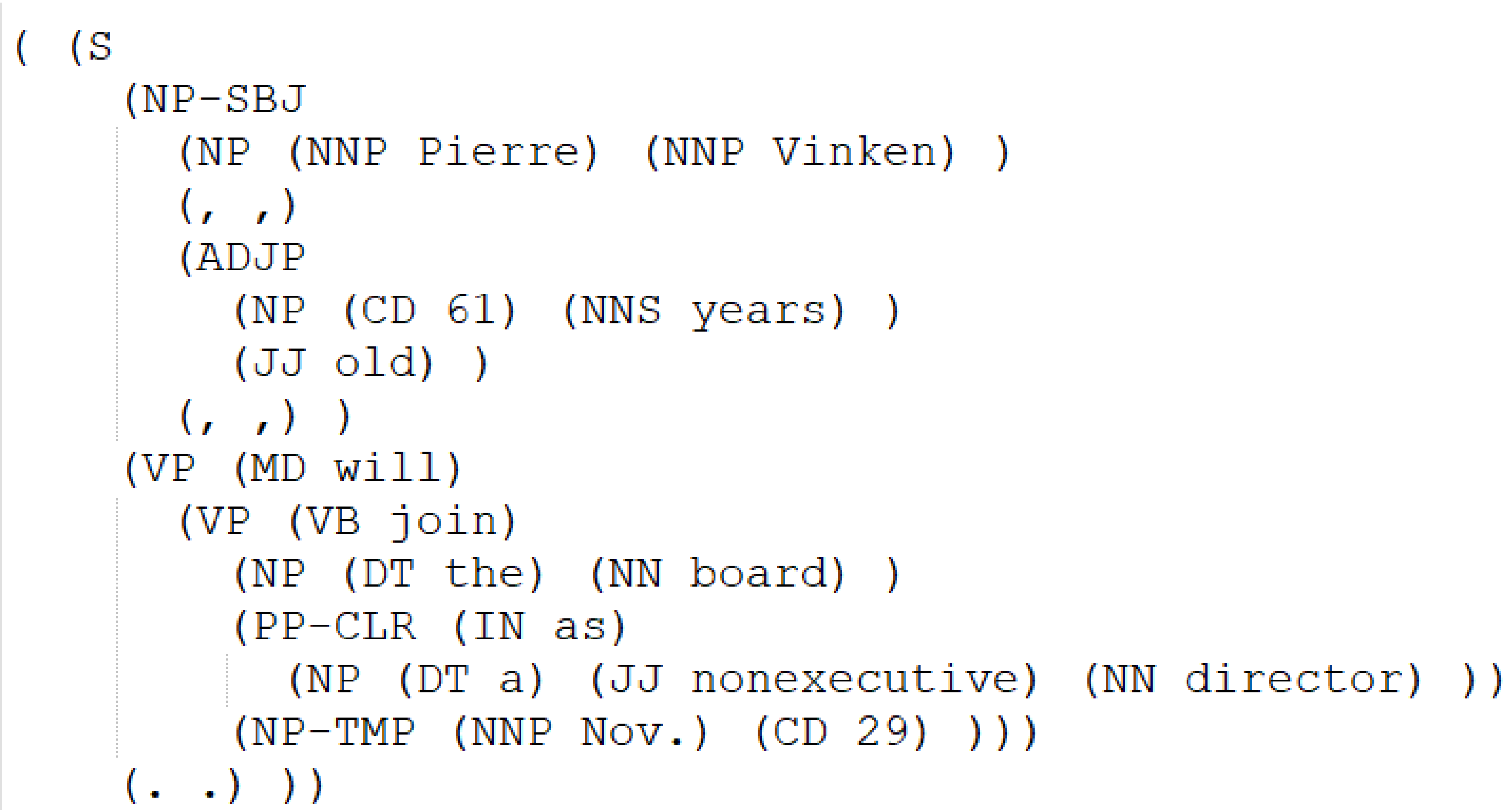
Steps

- Baseline POS tagging
  - Tag every word with all of its possible tags
  - Tag unknown words as nouns
- Decide on the resulting labeling (remove ambiguity)
  - Basing on grammar rules
  - Basing on probability
  - Using neural network
  - Combining probability and grammar constraints

Dataset

- Word dictionary
- Labeled dataset, with grammar rules
- Unlabeled dataset, with POS tags
- Unlabeled dataset, POS tags are automatically generated by statistical computation

A Penn Treebank tree



```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```

Difficulty in Vietnamese POS tagging

- Depend on characteristics of each language
- Lack of training data like Brown or Penn Treebank

➢ Difficulty in evaluating results

Approach 1

[Đinh Điền] Dien Dinh and Kiem Hoang, POS-tagger for English-Vietnamese bilingual corpus. HLTNAACL Workshop on Building and using parallel texts: data driven machine translation and beyond, 2003.

- Translate and map information from English POS tags, because:
  - English POS tagger has high accuracy (>97%)
  - Recent success of word alignment methods between language pairs.

[Đinh Điền]

- Build a bilingual English - Vietnamese corpus ~ 5 million words (both English and Vietnamese).
- Tag English POS using Transformation-based Learning – TBL (Brill 1995)
- Align between two languages (accuracy about 87%) to convert English POS tags to Vietnamese.
- The results are manually calibrated to serve as training data for the Vietnamese POS tagger.

[Đinh Điền]

- Advantage:
  - Avoid manual labeling of POS tags by using POS tags from another language.
- Disadvantages :
  - English and Vietnamese are different: word structure, word order, grammatical function of words in sentences → difficulty in alignment
  - Errors accumulate over two stages: (a) assigning English POS tags; (b) alignment between the two languages
  - The POS tags is directly converted from English to Vietnamese is not typical for Vietnamese

Approach 2

- [Nguyen Huyen, Vu Luong] Thi Minh Huyen Nguyen, Laurent Romary, and Xuan Luong Vu, A Case Study in POS Tagging of Vietnamese Texts. The 10th annual conference TALN 2003.
- Basing on the linguistic properties of Vietnamese.
- Building a tagset for Vietnamese based on a general standard of Western European languages, in order to create the label set at two levels:
  - Kernel layer: the most common specification for languages
  - Private layer: extend for a particular language based on the language’s properties

[Nguyen Huyen, Vu Luong]

- Kernel layer: danh từ (noun – N), động từ (verb – V), tính từ (adjective – A), đại từ (pronoun – P), mạo từ (determine – D), trạng từ (adverb – R), tiền-hậu giới từ (adposition – S), liên từ (conjunction – C), số từ (numeral – M), tình thái từ (interjection – I), từ ngoại Việt (residual – X, như foreign words, ...).
- Private layer : extend basing on the above word forms such as countable/uncountable nouns, male/female for pronouns, etc.

Approach 3

- [Phuong] Nguyễn Thị Minh Huyền, Vũ Xuân Lương, Lê Hồng Phương . Sử dụng bộ gán nhãn từ loại xác suất QTAG cho văn bản tiếng Việt. Kỷ yếu Hội thảo ICT.rda'03
- Working on a window of size 3, after adding 2 fake words at the begin and end of the input text.
- The POS tag for each word being out of the window is the final tag.

- w0 w1 w2 w3 w4
- Chúng_tôi bàn về cái bàn.
- PP VB IN NN NN
- ĐaT ĐgT GT DT DT
- ĐaT ĐgT/DT GT/ĐgT DT DT/ĐgT

P_w = P(tag|token) P_c = P(tag|t_1,t_2)

Xét w0

BoS BoS chúng_tôi

- P(chúng_tôi)= P(ĐaT|chúng_tôi) P(ĐaT) = P(ĐaT| BoS BoS)

BoS chúng_tôi bàn

- P(chúng_tôi) = P(ĐaT|chúng_tôi) P(ĐaT) = P(ĐaT|BoS)
- P(bàn) = P(ĐgT|bàn) P(ĐgT) = P(ĐgT|BoS ĐaT)
- P(bàn) = P(DT|bàn) P(DT) = P(DT|BoS ĐaT)

Chúng_tôi bàn về

- P(chúng_tôi) = P(ĐaT|chúng_tôi) P(ĐaT) = P(ĐaT)
- P(bàn) = P(ĐgT|bàn) P(ĐgT) = P(ĐgT| ĐaT)
- P(bàn) = P(DT|bàn) P(DT) = P(DT| ĐaT)
- P(về)=P(GT|về) P(GT) = P(GT| ĐaT DT) P(GT) = P(GT| ĐaT ĐgT)
- P(về)=P(ĐgT|về) P(ĐgT) = P(ĐgT| ĐaT DT) P(ĐgT) = P(ĐgT| ĐaT ĐgT)

P(w).P(c) = P(ĐaT|chúng_tôi) *P(ĐaT| BoS BoS) +
P(ĐaT|chúng_tôi) * P(ĐaT|BoS) + P(ĐaT|chúng_tôi) * P(ĐaT)

POS tagging algorithm [Phương]

1. Read the next word (token)
2. Find the word in the dictionary
3. If not found, assign to that word all possible labels
4. For each possible label
   a. Calculate P_w = P(tag|token)
   b. Calculate P_c = P(tag|t_1 ,t_2), t_1 , t_2 are the corresponding labels of the two words preceding the current word.
   c. Calculate $P_{w,c}$ = P_w * P_c , combining the above two probabilities.
5. Repeat the calculation for the other two words in the window

After each recalculation (3 times for each word), the resulting probabilities are combined to have the overall probability of the label assigned to the word.

[Phương]

- Divide the labeled corpus into 2 sets: training set and test set
- Automatically assign labels to the input text
- Compare the results with the sample data.
- Training time with 32000 words: ~ 30s

[Phương]

- Labeled sentence:

  <w pos="Nc"> **hồi**</w> <w pos="Vto"> **lên** </w> < w pos="Nn"> **sáu** </w> <w pos=","> **,** </w> <w pos="Vs"> **có** </w> <w pos="Nu"> **lần** </w> <w pos="Pp"> **tôi** </w> <w pos="Jt"> **đã** </w> <w pos="Vt"> **nhìn** </w> <w pos="Vt"> **thấy** </w> <w pos="Nn"> **một** </w> <w pos="Nt"> **bức** </w> <w pos="Nc"> **tranh** </w> <w pos="Jd"> **tuyệt** </w> <w pos="Aa"> **đẹp** </w>

Nc - danh từ đơn thể, Vto - ngoại động từ chỉ hướng, Nn - danh từ số lượng, Vs - động từ tồn tại, Nu - danh từ đơn vị, Pp - đại từ nhân xưng, Jt - phụ từ thời gian, Vt - ngoại động từ, Nt - danh từ loại thể, Jd - phụ từ chỉ mức độ, Aa - tính từ hàm chất.

[Phương]

- Sentences from sample corpus

<w pos="Nc"> **back** </w> <w pos="Vto"> **up** </w> < w pos="Nn"> **six** </w> <w pos=","> **,** </ w> <w pos="Vs"> **yes** </w> <w pos="Nu"> **times** </w> <w pos="Pp"> **I** </w> <w pos="Jt"> **did** </w> <w pos="Vt"> **look** </w> <w pos="Vt"> **see** </w> <w pos="Nn"> **one** </w> <w pos="Nt" > **picture** </w> <w pos="Nc"> **picture** </w> <w pos="Jd"> **great** </w> <w pos="Aa"> **beautiful** </w>

- Sentences labeled by the program

<w pos="Nc"> **back** </w> <w pos=“Adv”> **up** </w> < w pos="Nn"> **six** </w> <w pos=","> **,** </ w> <w pos="Vs"> **yes** </w> <w pos="Nu"> **times** </w> <w pos="Pp"> **I** </w> <w pos=“JJ”> **did** </w> <w pos="Vt"> **look** </w> <w pos="Vt"> **see** </w> <w pos="Nn"> **one** </w> <w pos="Nt" > **picture** </w> <w pos="Nc"> **picture** </w> <w pos="Jd"> **great** </w> <w pos="Aa"> **beautiful** </w>

- Precision = number of correctly labeled words/total number of labeled words
- Recall = number of correctly labeled words/ total number of correct words

Sentences from sample corpus: (30)

(E Ở)(N số)(M 10)(N phố)(Np Hàng Mành)(Np Hà Nội)(, ,)
(N vợ chồng) (Np Dương Tuấn) (- -) (Np Đặng Hải Lý)(, ,)
(M 26) (N tuổi)(, ,)(V mở)(N lớp) (V dạy)(V viết)(N chữ) (A đẹp)(. .)
(N Lớp học)(E của)(P họ)(X ngày càng)(V thu hút)
(L nhiều)(N học viên)(. .)

Sentences labeled by the program: (30)

(R Ở)(N số)(M 10)(N phố)(Np Hàng Mành)(Np Hà Nội)(, ,)
(N vợ chồng) (Np Dương Tuấn) (- -) (Np Đặng Hải Lý)(, ,)
(M 26) (N tuổi)(, ,)(V mở)(N lớp) (V dạy)(V viết)(N chữ) (A đẹp)(. .)
(N Lớp học)(C của)(P họ)(R ngày càng)(A thu hút)
(A nhiều)(N học viên)(. .)

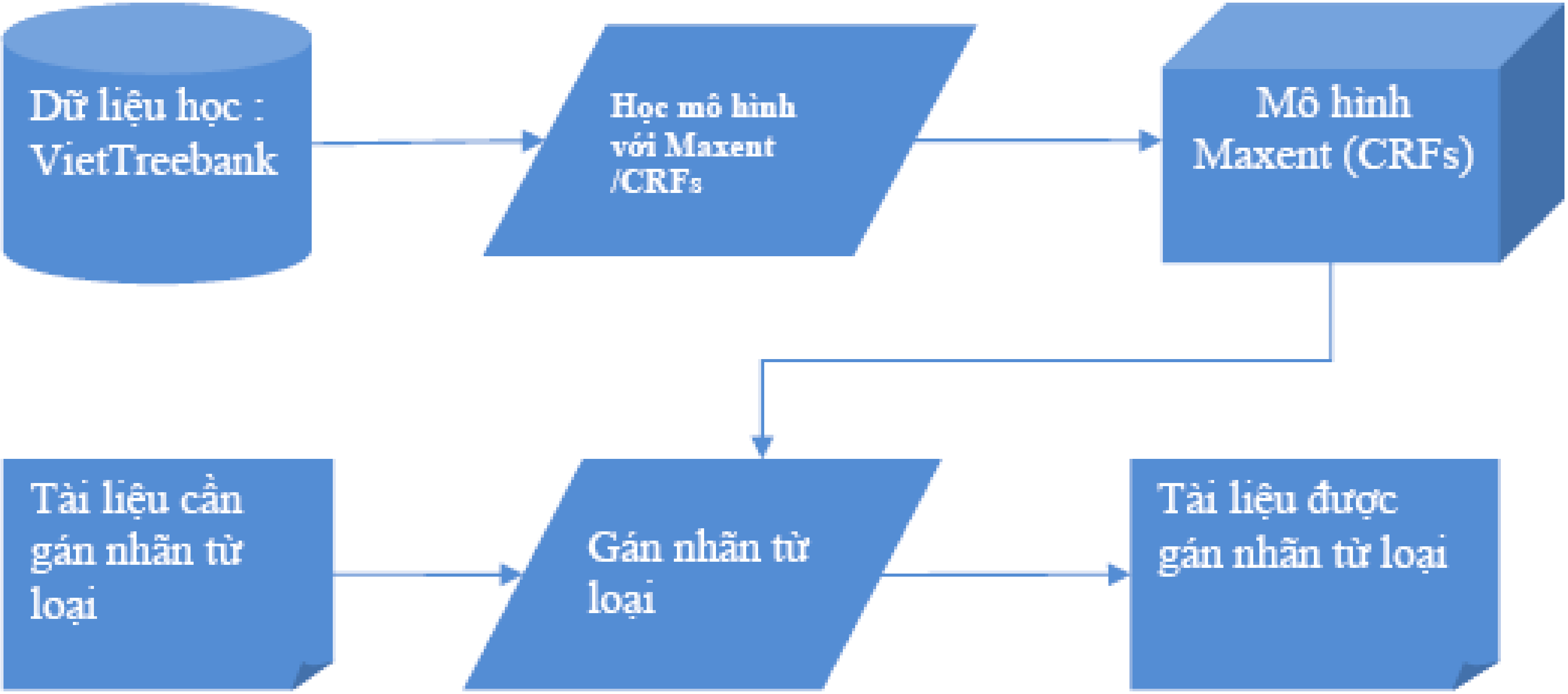
[Phương]

- Result:
  - ~94% (9 vocabulary tags and 10 labels for symbol types)
  - ~85% (48 vocabulary tags and 10 labels for symbol types)
- Without using a lexical dictionary (using only the sample labeled corpus), the results are only ~80% and ~60%, respectively.

Approach 4

- Phan Xuân Hiếu (2009). Công cụ gán nhãn từ loại tiếng Việt dựa trên Conditional Random Fields và Maximum Entropy JvnTagger.
- Basing on Maximum Entropy (MaxEnt) and Conditional Random Fields (CRFs).
- Training set: Viet Treebank corpus, more than 10,000 Vietnamese sentences labeled by language experts.

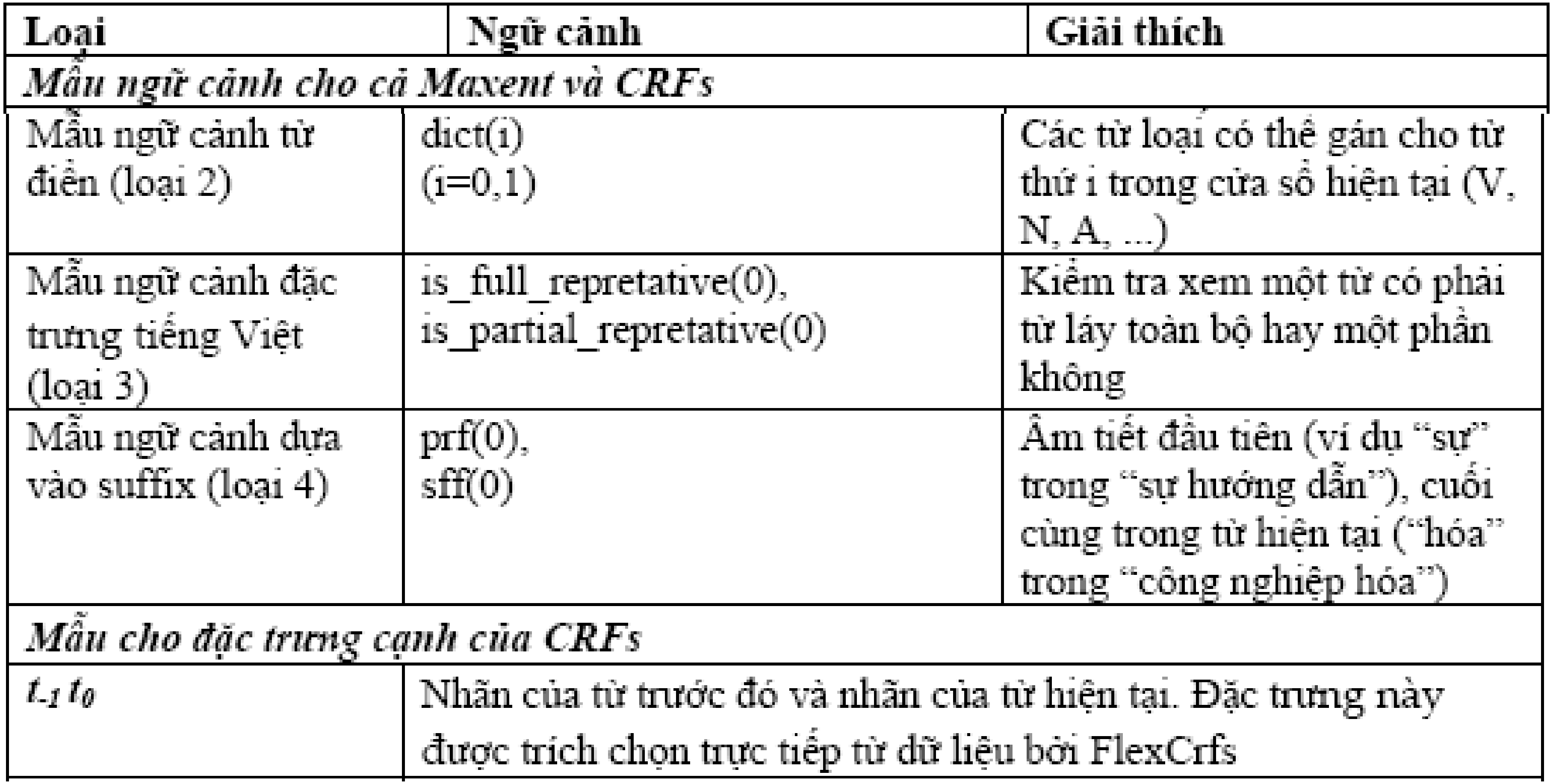
Học mô hình gán nhãn từ loại


Dữ liệu học : VietTreebank
Học mô hình với Maxent /CRFs
Mô hình Maxent (CRFs)
Tài liệu cần gán nhãn từ loại
Gán nhãn từ loại
Tài liệu được gán nhãn từ loại

Feature extraction

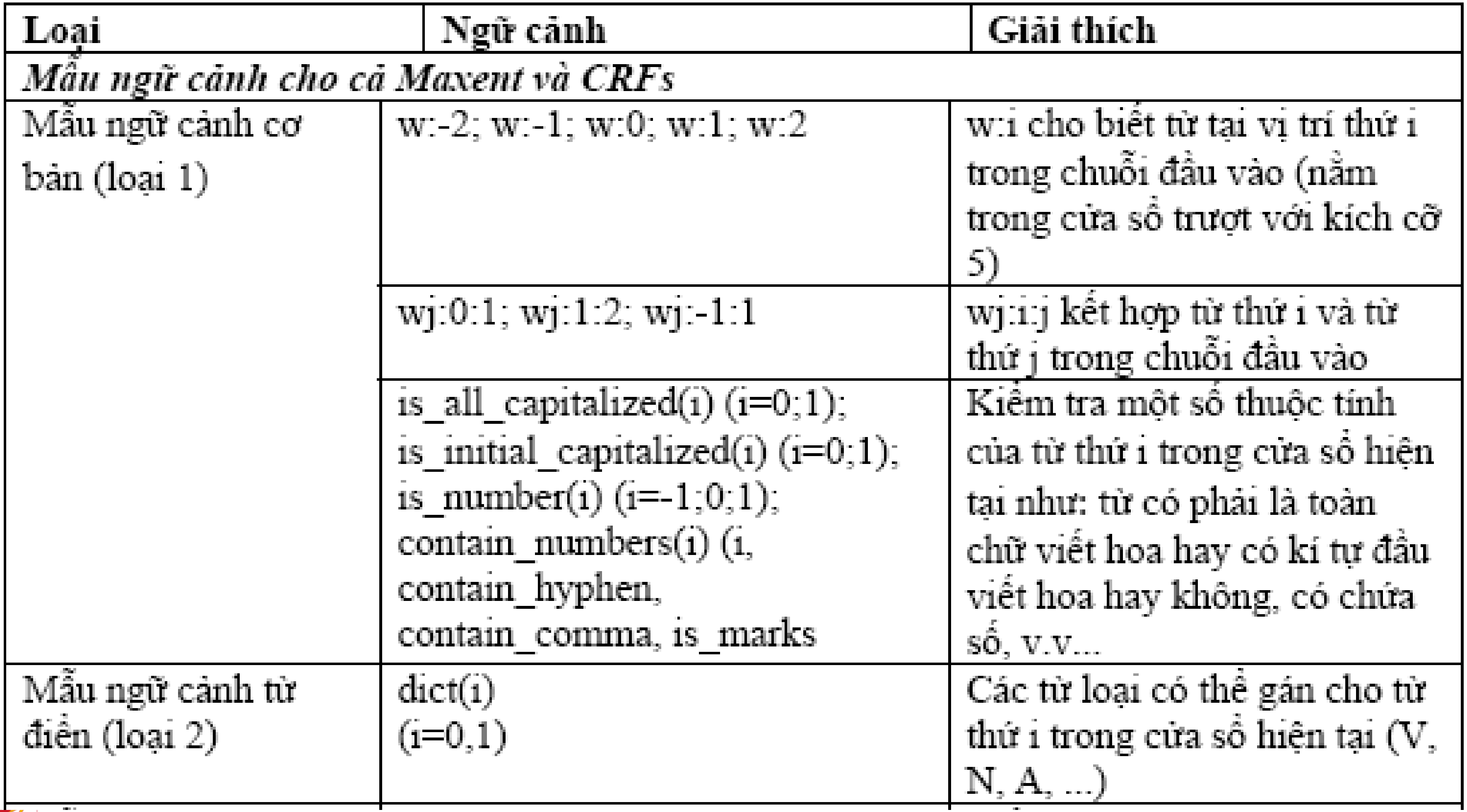
- ... thường trò_chuyện với Mai ...
- It is necessary to determine the POS tag for the word “trò_chuyện”, the characteristics:
  - The word “trò_chuyện” in the dictionary often appears with which POS tag?
  - What is the word “trò_chuyện” usually labeled as? Is it a verb?
  - What does the word “thường” before the word “trò_chuyện” usually suggest?
  - What does the word “với” after “trò_chuyện” suggest? Does it suggest that it is preceded by a verb?
  - What does the combination of the two words “với Mai” suggest, perhaps the previous word (“trò_chuyện”) should be a verb?

Context for feature extraction



Loại	Ngữ cảnh	Giải thích
Mẫu ngữ cảnh cho cả Maxent và CRFs		
Mẫu ngữ cảnh từ điển (loại 2)	dict(i) (i=0,1)	Các từ loại có thể gán cho từ thứ i trong cửa sổ hiện tại (V, N, A, ...)
Mẫu ngữ cảnh đặc trưng tiếng Việt (loại 3)	is_full_repretative(0), is_partial_repretative(0)	Kiểm tra xem một từ có phải từ láy toàn bộ hay một phần không
Mẫu ngữ cảnh dựa vào suffix (loại 4)	prf(0), sff(0)	Âm tiết đầu tiên (ví dụ “sự” trong “sự hướng dẫn”), cuối cùng trong từ hiện tại (“hóa” trong “công nghiệp hóa”)
Mẫu cho đặc trưng cạnh của CRFs		
$t_{-1}\ t_0$	Nhãn của từ trước đó và nhãn của từ hiện tại. Đặc trưng này được trích chọn trực tiếp từ dữ liệu bởi FlexCrfs	

Context for feature extraction

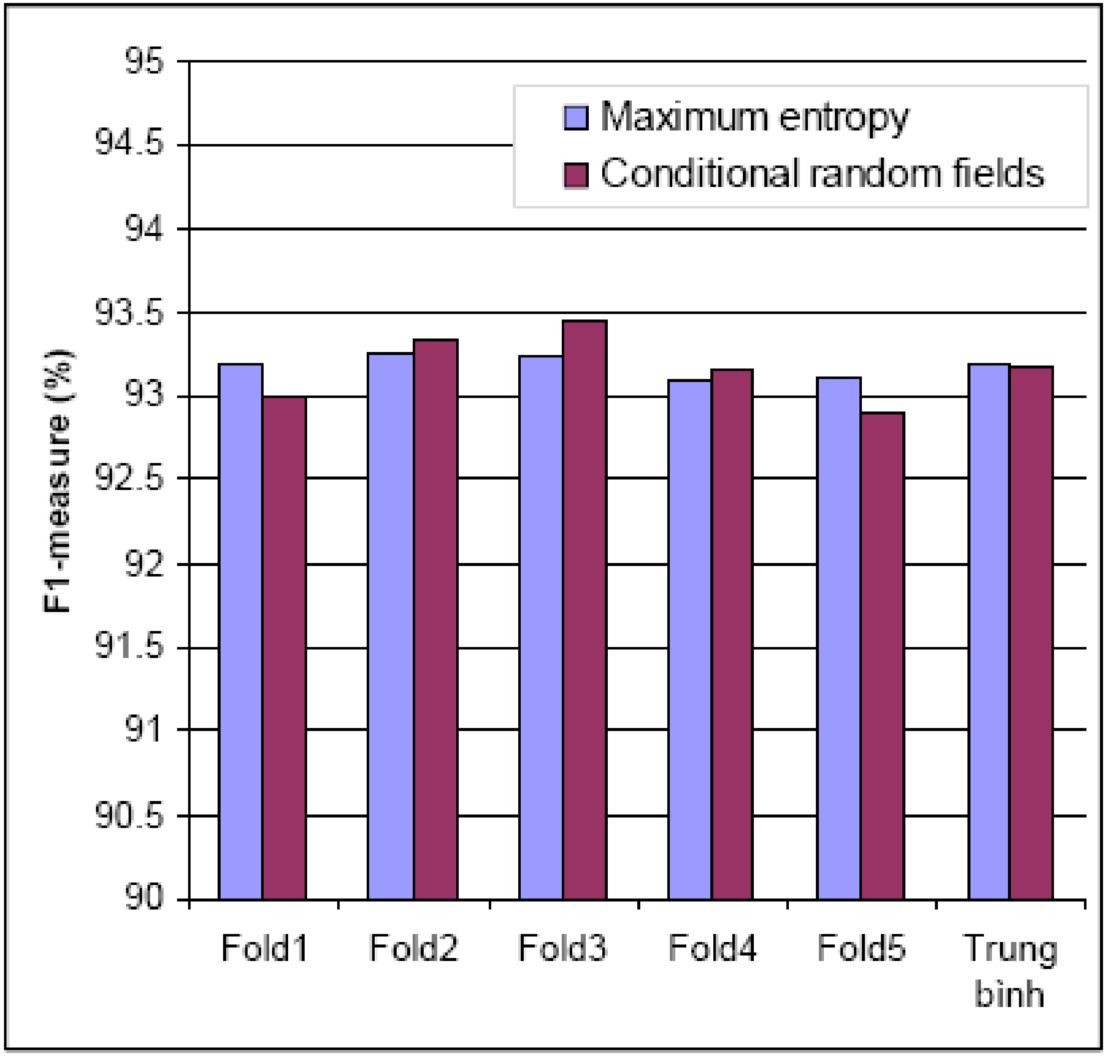


Loại	Ngữ cảnh	Giải thích
Mẫu ngữ cảnh cho cả Maxent và CRFs		
Mẫu ngữ cảnh cơ bản (loại 1)	w:-2; w:-1; w:0; w:1; w:2	w:i cho biết từ tại vị trí thứ i trong chuỗi đầu vào (nằm trong cửa sổ trượt với kích cỡ 5)
	wj:0:1; wj:1:2; wj:-1:1	wj:i:j kết hợp từ thứ i và từ thứ j trong chuỗi đầu vào

	is_all_capitalized(i) (i=0;1);
	is_initial_capitalized(i) (i=0;1);
	is_number(i) (i=-1;0;1);
	contain_numbers(i) (i,
	contain_hyphen,
	contain_comma, is_marks
		Kiểm tra một số thuộc tính của từ thứ i trong cửa sổ hiện tại như: từ có phải là toàn chữ viết hoa hay có kí tự đầu viết hoa hay không, có chứa số, v.v...

Mẫu ngữ cảnh từ điển (loại 2)	dict(i) (i=0,1)	Các từ loại có thể gán cho từ thứ i trong cửa sổ hiện tại (V, N, A, ...)

Performance using MaxEnt and CRFs



Vietnamese POS tag set

idPOS	symbolPOS	vnPOS	enPOS
1	N	danh từ (DT)	noun
2	V	động từ (ĐgT)	verb
3	A	tính từ (TT)	adjective
4	M	số từ (ST)	numeral
5	P	đại từ (ĐaT)	pronoun
6	R	phụ từ (PT)	adverb
7	O	giới từ (GT)	preposition
8	C	liên từ (LT)	conjunction
9	I	trợ từ	auxiliary word
10	E	cảm từ	emotivity word
11	Xy*	từ tắt	abbreviation
12	S	yếu tố từ (bất, vô…)	component stem
13	U	không xác định	undetermined

Vietnamese POS subtag set

idPOS	idSub	symbol	vnPOS	enPOS
	POS	POS		
1	1	Np	danh từ riêng	proper noun
1	2	Nc	danh từ đơn thể	countable noun
1	3	Ng	danh từ tổng thể	collective Noun
1	4	Na	danh từ trừu tượng	abstract noun
1	5	Ns	danh từ chỉ loại	classifier noun
1	6	Nu	danh từ đơn vị	unit noun
1	7	Nq	danh từ chỉ lượng	quantity noun
2	8	Vi	động từ nội động	intransitive verb
2	9	Vt	động từ ngoại động	transitive verb
2	10	Vs	động từ trạng thái	state verb
2	11	Vm	động từ tình thái	modal verb
2	12	Vr	động từ quan hệ	relative verb
3	13	Ap	tính từ tính chất	property adjective
3	14	Ar	tính từ quan hệ	relative adjective
3	15	Ao	tính từ tượng thanh	onomatopoetic adjective
3	16	Ai	tính từ tượng hình	pictographic adjective

Vietnamese POS subtag set

idPOS	idSub POS	symbol POS	vnPOS	enPOS
4	17	Mc	số từ số lượng	cardinal numeral
4	18	Mo	số từ thứ tự	ordinal numeral
5	19	Pp	đại từ xưng hô	personal pronoun
5	20	Pd	đại từ chỉ định	demonstrative pronoun
5	21	Pq	đại từ số lượng	quality pronoun
5	22	Pi	đại từ nghi vấn	interrogative pronoun
6	23	R	phụ từ	adverb
7	24	O	giới từ	preposition
8	25	C	liên từ	conjunction
9	26	I	trợ từ	auxiliary word
10	27	E	cảm từ	emotivity word
11	28	Xy	từ tắt	abbreviation
12	29	S	yếu tố từ (bất, vô…)	component stem
13	30	U	không xác định	undetermined