University of Science
Computational Linguistics Center
Introduction to Natural Language Processing

Section 3: Corpus-based NLP

Lecturer: Assoc.Prof. Dr. Dinh Dien

Section 3: Rule-based NLP vs. Corpus-based NLP

Case study: Determiner Placement

Task: Automatically place determiners (*a,the,null*) in a text

Scientists in United States have found way of turning lazy monkeys into workaholics using gene therapy. Usually monkeys work hard only when they know reward is coming, but animals given this treatment did their best all time. Researchers at National Institute of Mental Health near Washington DC, led by Dr Barry Richmond, have now developed genetic treatment which changes their work ethic markedly. ”Monkeys under influence of treatment don’t procrastinate,” Dr Richmond says. Treatment consists of anti-sense DNA - mirror image of piece of one of our genes - and basically prevents that gene from working. But for rest of us, day when such treatments fall into hands of our bosses may be one we would prefer to put off.

Rule-based NLP

Relevant Grammar Rules

- Determiner placement is largely determined by:
  - Type of noun (countable, uncountable)
  - Reference (specific, generic)
  - Information value (given, new)
  - Number (singular, plural)
- However, many exceptions and special cases play a role:
  - The definite article is used with newspaper titles (*The Times*), but zero article in names of magazines and journals (*Time*)

Rule-based NLP

Symbolic Approach: Determiner Placement

What categories of knowledge do we need:

- Linguistic knowledge:
  - Static knowledge: number, countability, . . .
  - Context-dependent knowledge: co-reference, . . .
- World knowledge:
  - Uniqueness of reference (*the current president of the US*), type of noun (*newspaper vs. magazine*), situational associativity between nouns (*the score of the football game*), . . .

Hard to manually encode this information!

Statistical Approach: Determiner Placement

Naive approach:

- Collect a large collection of texts relevant to your domain (e.g., newspaper text)
- For each noun, compute its probability to take a certain determiner
  $p(determiner|noun) = \frac{freq(noun,determiner)}{freq(noun)}$
- Given a new noun, select a determiner with the highest likelihood as estimated on the training corpus

Corpus-based NLP

Does it work?

- Implementation
  - Corpus: training — first 21 sections of the Wall Street Journal (WSJ) corpus, testing – the 23th section
  - Prediction accuracy: 71.5%
- The results are not great, but surprisingly high for such a simple method
  - A large fraction of nouns in this corpus always appear with the same determiner
    “the FBI”,“the defendant”, . . .

Determiner Placement as Classification

- **Prediction:** *"the"*, *"a"*, *"null"*
- **Representation of the problem:**
  - plural? (yes, no)
  - first appearance in text? (yes, no)
  - noun (members of the vocabulary set)

Noun	plural?	first appearance	determiner
defendant	no	yes	the
cars	yes	no	null
FBI	no	no	the
concert	no	yes	a

Goal: Learn classification function that can predict unseen examples

Classification Approach

- Learn a function from $X \to Y$ (in the previous example, $\{-1, 0, 1\}$)
- Assume there is some distribution $D(X, Y)$, where $x \in X$, and $y \in Y$
- Attempt to explicitly model the distribution $D(X, Y)$ and $D(X|Y)$

Beyond Classification

Many NLP applications can be viewed as a mapping from one complex set to another:

- Parsing: strings to trees
- Machine Translation: strings to strings
- Natural Language Generation: database entries to strings

Classification framework is not suitable in these cases!

Learning for MT

- Parallel corpora are available in several language pairs
- Basic idea: use a parallel corpus as a training set of translation examples
- Goal: learn a function that maps a string in a source language to a string in a target language

Introduction to Corpus

Definition: Corpus = "a collection of written or spoken texts" (Oxford Dic)

"A corpus is a collection of pieces of language that are selected and ordered according to explicit linguistic criteria in order to be used as a sample of the language"

(Sinclair 1996)

- Translation of "corpus" = "语料库"/yǔ liào kù/(Chinese: ngữ liệu khố); "코퍼스"/kô-po-sư/ (Ko); "コーパス" /kô-pa-zu/ (Jp); corpus (Fr), korpus (Ge), корпус (Ru),...

Criteria: representativeness, balance, sampling

Corpus Classification

"[…] the term *corpus* as used in modern linguistics can best be defined as a collection of sampled texts, written or spoken, in machine-readable form which may be annotated with various forms of linguistic information"

(McEnery, Xiao and Tono 2006)

Annotation: Raw (unannotated) vs. Annotated (linguistic information: aspects and linguistic units)

Language: monolingual vs. Multilingual

Alignment: parallel vs. comparable

Parallel: text – paragraph – sentence – word alignment

Corpus samples

- PTB (Penn Tree Bank): [Pierre/NNP Vinken/NNP],/, [61/CD years/NNS] old/JJ ,/, will/MD join/VB [the/DT board/NN] as/IN [a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD]./.
- CTB (Chinese Tree Bank): <S ID=12>((IP-HLN (NP-SBJ (NN 外商) (NN 投资) (NN 企业)) (VP (VV 成为) (NP-OBJ (NP (NP-PN (NR 中国)) (NP (NN 外贸))) (ADJP (JJ 重要)) (NP (NN 增长点)))))) </S>

(VTB: Vietnamese Tree Bank): <SEG id="1">
Nguyên_nhân/Nn/O là/Vc/O bão/Nn/O số/Nn/O 10/An/O đang/R/O chịu/Vv/O ảnh_hưởng/Nn/O bởi/Cp/O hệ_thống/Nn/O trục/Nn/O rãnh/Nn/O cao/Aa/O và/Cp/O sự/Nc/O lôi_kéo/Vv/O từ/Cm/O siêu__bão/Nn/TRM_B Melor/Nr/TRM_I ở/Cm/O ngoài/Cm/O khơi/Nn/O Philippines/Nr/LOC_B ./PU/O</SEG

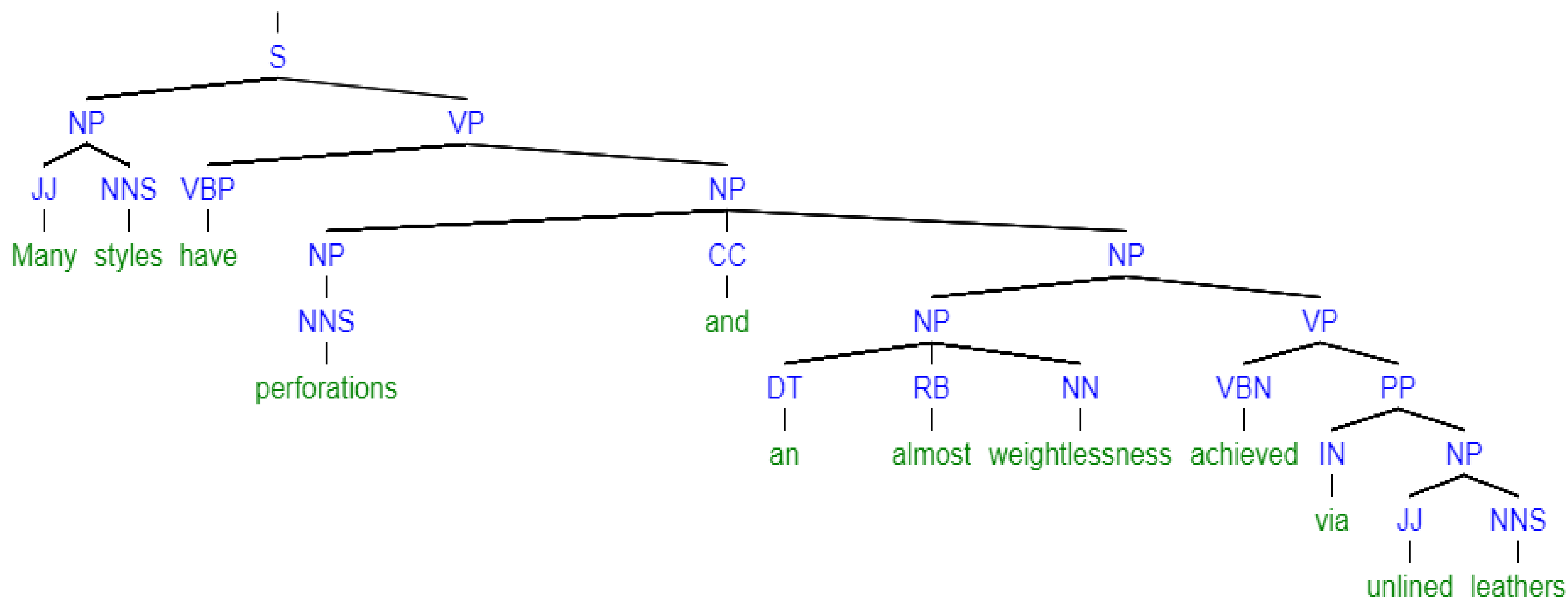
tagset

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 "
PRP	Personal pronoun	*I, you, he*	(	Left parenthesis	[, (, {, <
PRP$	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*			

Figure 5.6 Penn Treebank part-of-speech tags (including punctuation).

```
[ Many/JJ styles/NNS ]
have/VBP
[ perforations/NNS ]
and/CC
[ an/DT almost/RB weightlessness/NN ]
achieved/VBN via/IN
[ unlined/JJ leathers/NNS ]
./.
```

```
( (S
    (NP (JJ Many) (NNS styles) )
    (VP (VBP have)
      (NP
        (NP (NNS perforations) )
        (CC and)
        (NP
          (NP (DT an) (RB almost) (NN weightlessness) )
          (VP (VBN achieved)
            (PP (IN via)
              (NP (JJ unlined) (NNS leathers) )))))))
```



SUSANNE

```
A01:0010a    -    YB       <minbrk>        -        [Oh.Oh]
A01:0010b    -    AT       The      the    [O[S[Nns:s.
A01:0010c    -    NP1s     Fulton   Fulton [Nns.
A01:0010d    -    NNL1cb   County   county .Nns]
A01:0010e    -    JJ       Grand    grand  .
A01:0010f    -    NN1c     Jury     jury   .Nns:s]
A01:0010g    -    VVDv     said     say    [Vd.Vd]
A01:0010h    -    NPD1     Friday   Friday [Nns:t.Nns:t]
A01:0010i    -    AT1      an       an     [Fn:o[Ns:s.
A01:0010j    -    NN1n     investigation   investigation    .
A01:0020a    -    IO       of       of     [Po.
A01:0020b    -    NP1t     Atlanta  Atlanta [Ns[G[Nns.Nns]
A01:0020c    -    GG       +<apos>s        -        .G]
A01:0020d    -    JJ       recent   recent .
A01:0020e    -    JJ       primary  primary .
A01:0020f    -    NN1n     election        election         .Ns]Po]Ns:s]
A01:0020g    -    VVDv     produced        produce [Vd.Vd]
A01:0020h    -    YIL      <ldquo>  -      .
A01:0020i    -    ATn      +no      no     [Ns:o.
A01:0020j    -    NN1u     evidence        evidence         .
A01:0020k    -    YIR      +<rdquo>        -        .
A01:0020m    -    CST      that     that   [Fn.
A01:0030a    -    DDy      any      any    [Np:s.
A01:0030b    -    NN2      irregularities  irregularity     .Np:s]
A01:0030c    -    VVDv     took     take   [Vd.Vd]
A01:0030d    -    NNL1c    place    place  [Ns:o.Ns:o]Fn]Ns:o]Fn:o]S]
A01:0030e    -    YF       +.       -      .O]
A01:0030f    -    YB       <minbrk>        -        [Oh.Oh]
```

Parallel corpus: alignment

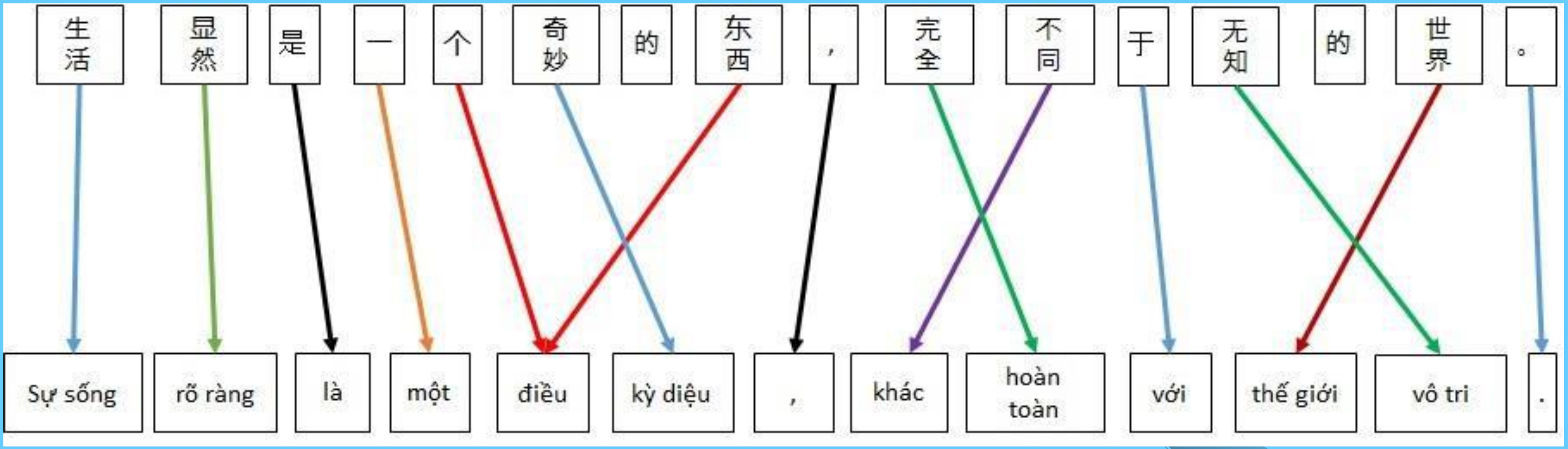
Helicopters can rise straight up into the air and can go straight down. They can stand still in the air. Helicopters do not have wings. A huge whirling propeller, called a rotor, on top of a helicopter provides the lift.

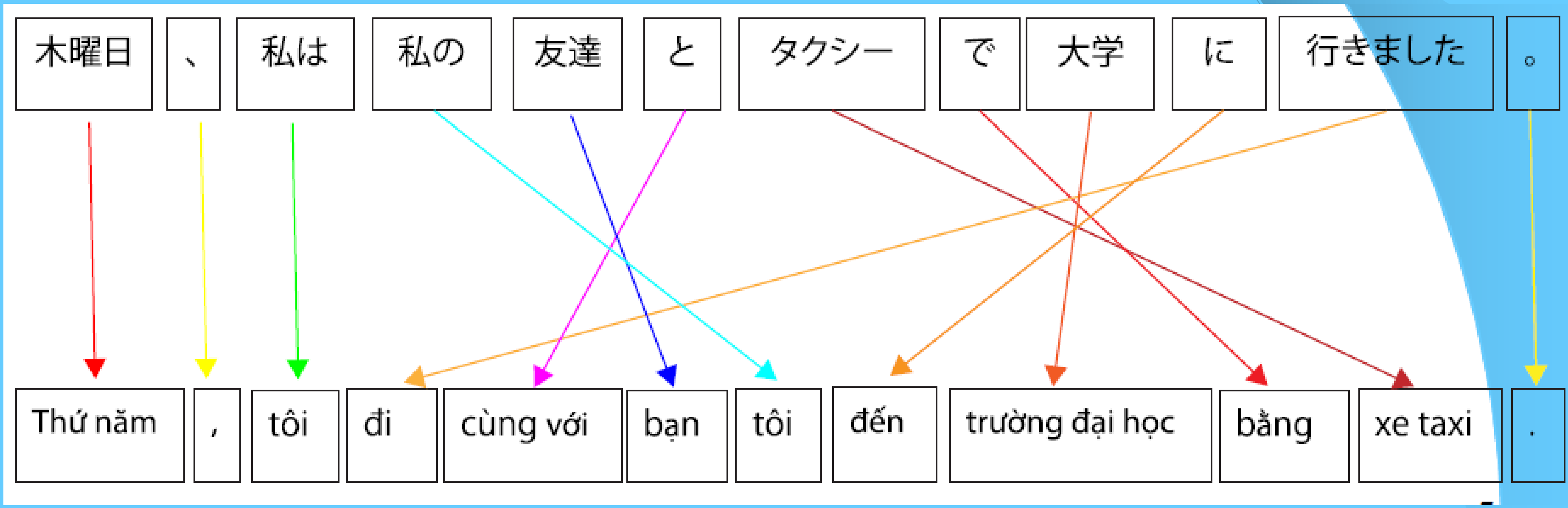
Máy bay trực thăng có thể lên thẳng trên không yà đáp thẳng xuống đất . Chúng có thể đứng yên trên không. Máy bay trực thăng không có cánh, một cánh quạt lớn gọi là chong chóng trên đầu chiếc máy bay cung cấp sức nâng.

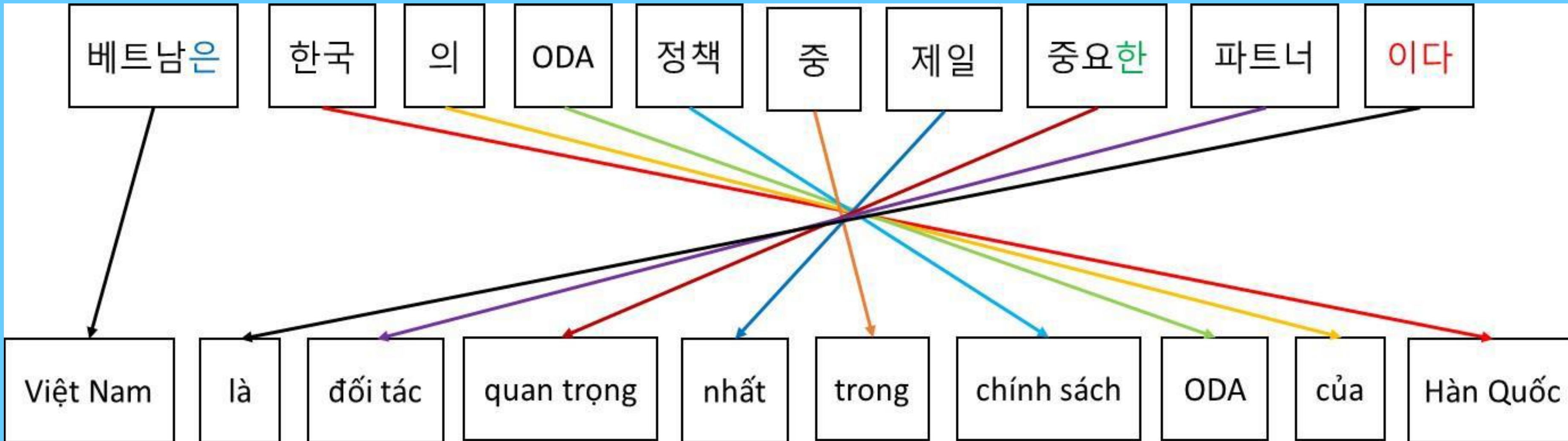
* Helicopters can rise straight up into the air and can go straight down.
\+ Máy bay trực thăng có thể lên thẳng trên không và đáp thẳng xuống đất.

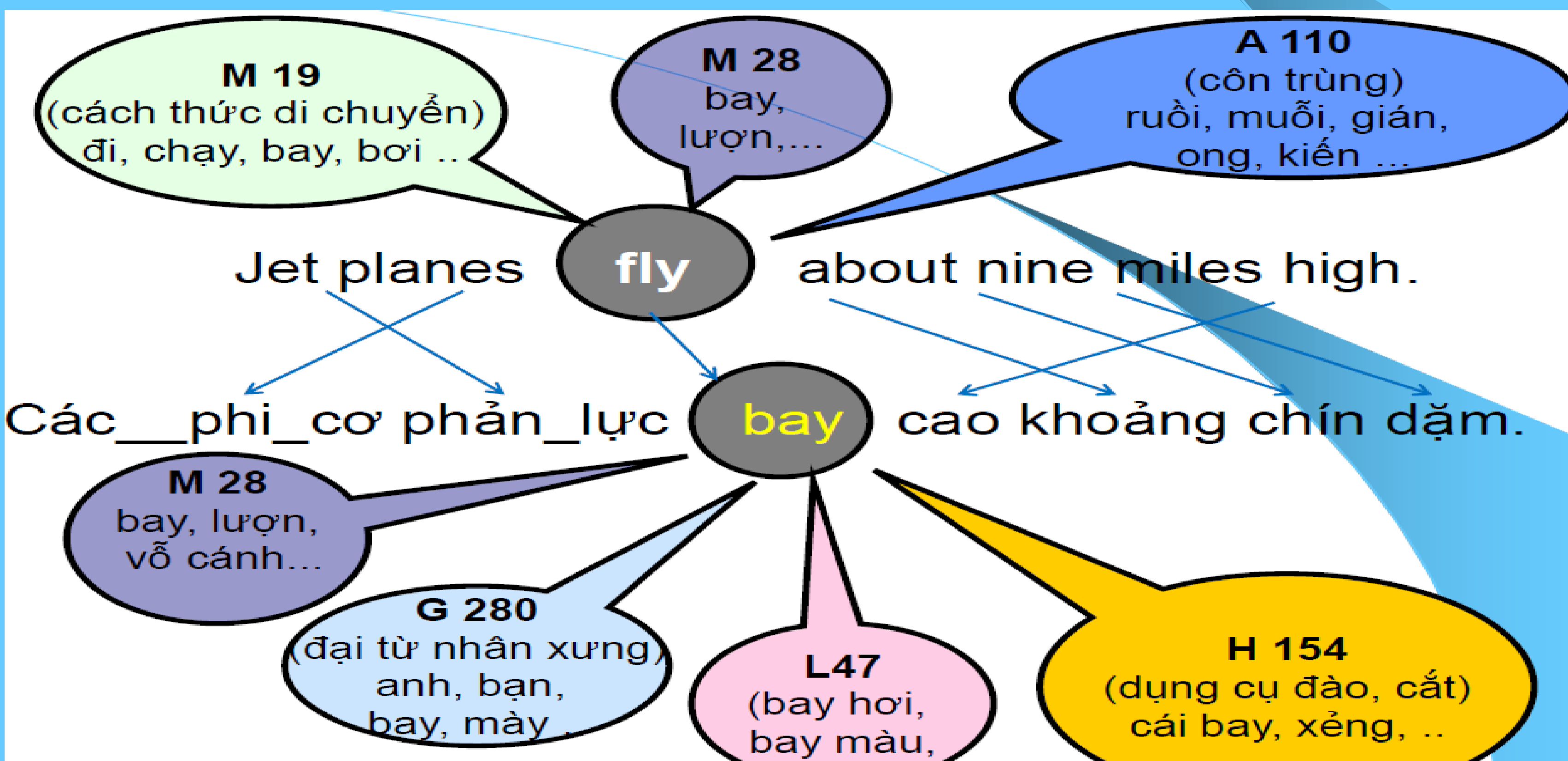
* They can stand still in the air.
\+ Chúng có thể đứng yên trên không.

* Helicopters do not have wings.
\+ Máy bay trực thăng không có cánh.


生活
显然
是
一
个
奇妙
的
东西
，
完全
不同
于
无知
的
世界
。
Sự sống
rõ ràng
là
một
điều
kỳ diệu
,
khác
hoàn toàn
với
thế giới
vô tri
.


木曜日
、
私は
私の
友達
と
タクシー
で
大学
に
行きました
。
Thứ năm
,
tôi
đi
cùng với
bạn
tôi
đến
trường đại học
bằng
xe taxi
.


베트남은
한국
의
ODA
정책
중
제일
중요한
파트너
이다
Việt Nam
là
đối tác
quan trọng
nhất
trong
chính sách
ODA
của
Hàn Quốc


M 19
(cách thức di chuyển)
đi, chạy, bay, bơi ..
M 28
bay,
lượn,...
A 110
(côn trùng)
ruồi, muỗi, gián,
ong, kiến ...
Jet planes
fly
about nine miles high.
Các__phi_cơ phản_lực
bay
cao khoảng chín dặm.
M 28
bay, lượn,
vỗ cánh...
G 280
(đại từ nhân xưng)
anh, bạn,
bay, mày ..
L47
(bay hơi,
bay màu,
H 154
(dụng cụ đào, cắt)
cái bay, xẻng, ..

Transformation-Based Learning

Introduction

- An ‘error-driven’ approach for learning an ordered set of rules
- Adds annotations/classifications to each token of the input
- Developed by Brill [1995] for POS tagging
- Also used for other NLP areas, e.g.
  - ➢ text chunking [Ramshaw and Marcus 1995; Florian et al. 2000]
  - ➢ prepositional phrase attachment [Brill and Resnik 1994]
  - ➢ parsing [Brill 1996]
  - ➢ dialogue act tagging [Samuel 1998]
  - ➢ named entity recognition [Day et al. 1997]

Required Input

For application:

- The input to annotate:
  POS: *Recently, there has been a rebirth of empiricism in the field of natural language processing.*

Additionally for training:

- The correctly annotated input ('truth'):
  POS: *Recently/RB ,/, there/EX has/VBZ been/VBN a/DT rebirth/NN of/IN empiricism/NN in/IN the/DT field/NN of/IN natural/JJ language/NN processing/NN ./.*

Preliminaries

- Templates of admissible transformation rules (triggering environments)
- An initial-state annotator

  POS:

  Known words: Tag each word with its the most frequent tag.

  Unknown words: Tag each capitalized word as proper noun (NNP); each other word as common noun (NP).
- An objective function for learning

  POS: *Minimize the number of tagging errors.*

Transformation Rules

Rewrite rules: what to replace

POS: $t_i \rightarrow t_j$; $* \rightarrow t_j$ *(replace tag t_i / any tag by tag t_j)*

Triggering environment: when to replace

POS:

Non-lexicalized templates:

1. The preceding (following) word is tagged t_a.
2. The word two before (after) is tagged t_a.
3. One of the two preceding (following) words is tagged t_a.
4. One of the three preceding (following) words is tagged t_a.
5. The preceding word is tagged t_a and the following word is tagged t_b.
6. The preceding (following) word is tagged t_a and the word two before (after) is tagged t_b.

Lexicalized templates:

1. The preceding (following) word is w_a.
2. The word two before (after) is w_a.
3. One of the two preceding (following) words is w_a.
4. The current word is w_a and the preceding (following) word is w_b.
5. The current word is w_a and the preceding (following) word is tagged t_a.
6. The current word is w_a.
7. The preceding (following) word is w_a and the preceding (following) tag is t_a.
8. The current word is w_a, the preceding (following) word is w_b and the preceding (following) tag is t_a.

Learning Algorithm

1. Generate all rules that correct at least one error.
2. For each rule:
   (a) Apply to a copy of the most recent state of the training set.
   (b) Score the result using the objective function.
3. Select the rule with the best score.
4. Update the training set by applying the selected rule.
5. Stop if the score is smaller than some pre-set threshold T; otherwise repeat from step 1.

Rules Learnt

The first rules learnt by Brill's POS tagger (with examples):

#	From	To	If
1	NN	VB	previous tag is TO
	to/TO conflict/NN→ NB		
2	VBP	VB	one of the previous 3 tags is MD
	might/MD vanish/VBP→ VB		
3	NN	VB	one of the previous two tags is MD
	might/MD not reply/NN→ VB		
4	VB	NN	one of the previous two tags is DT
	the/DT amazing play/VB→ NN		

Tagging Unknown Words

Additional rule templates use character-based cues:
Change the tag of an unknown word from X to Y if:

1. Deleting the prefix (suffix) x, $|x| \leq 4$, results in a word.
2. The first (last) 1–4 characters of the word are x.
3. Adding the character string x, $|x| \leq 4$, as a prefix (suffix) results in a word.
4. Word w appears immediately to the left (right) of the word.
5. Character z appears in the word.

Unknown Words: Rules Learnt

#	From	To	If
1	NN	NNS	has suffix **-s**
	rules/NN $\rightarrow$ *NNS*		
4	NN	VBN	has suffix **-ed**
	tagged/NN $\rightarrow$ *VBN*		
5	NN	VBG	has suffix **-ing**
	applying/NN $\rightarrow$ *VBG*		
18	NNS	NN	has suffix **-ss**
	actress/NNS $\rightarrow$ *NN*		

Training Speedup: Hepple

Disallows interaction between learnt rules,
by enforcing two assumptions:

Sample independence: a state change in a sample does not change the context of surrounding samples

Rule commitment: there will be at most one state change per sample

→ Impressive reduction in training time, but the quality of the results is reduced (assumptions do not always hold)

‘Lossless’ Speedup: Fast TBL

1. Store for each rule r that corrects at least one error:
   - $good(r)$: the number of errors corrected by r
   - $bad(r)$: the number of errors introduced by r
2. Select the rule b with the best score.
   Stop if the score is smaller than a threshold T.
3. Apply b to each sample s.
4. Considering only samples in the set $\bigcup_{\{s|b\,\text{changes}\,s\}} V(s)$, where $V(s)$ is the set of samples whose tag might depend on s (the ‘vicinity’ of s; $s \in V(s)$):
   - Update $good(r)$ and $bad(r)$ for all stored rules, discarding rules whose $good(r)$ reaches 0.
   - Add rules with a positive $good(r)$ not yet stored.

   Repeat from step 2. [Ngai and Florian 2001]

Prepositional Phrase Attachment

Samples:
1. *I* ***[VB*** *washed]* ***[NP*** *the shirt]* ***[PP*** *with soap and water]*.
2. *I* ***[VB*** *washed]* ***[NP*** *the shirt]* ***[PP*** *with pockets]*.

Task: Is the prepositional phrase attached to the verb (sample 1) or to the noun phrase (sample 2)?

Approach: Apply TBL to 4-tuple of base head words (tag tuple as either *VB* or *NP*):

1. *wash shirt with soap*
2. *wash shirt with pocket*

Rules: Templates consider the words in the tuple and their semantic classes (WordNet hierarchy)

Evaluation

POS tagging:

	Regular TBL	Fast TBL	Hepple
Accuracy	96.61%	96.61%	96.23%
Time	38:06h	17:21min	6:13min

Prepositional Phrase Attachment:

	Regular TBL	Fast TBL	Hepple
Accuracy	81.0%	81.0%	77.8%
Time	3:10h	14:38min	4:01min

Scaling on input data:

Fast TBL: linear

Regular TBL: almost quadratic

Advantages

- Can capture more context than Markov models
- Always learns on the whole data set – no ‘divide and conquer’ ➔ no data sparseness:
  - ➢ Target evaluation criterion can be directly used for training, no need for indirect measures (e.g. entropy)
  - ➢ No overtraining
- Can consider its own (intermediate) results on the whole context ➔ More powerful than other methods like decision trees [Brill 1995, sec. 3]

More Advantages

- Can do any processing, not only classification:
  - Can change the structure of the input (e.g. parse tree)
  - Can be used as an postprocessor to any annotation system
- Resulting model is easy to review and understand
- Very fast to apply – rule set can be converted into a finite-state transducer [Roche and Schabes 1995] (for tagging and classification) or finite-state tree automaton [Satta and Brill 1996] (for parsing and other tree transformations)

… and Disadvantages

- Greedy learning so the found rule sequence might not be optimal
- Not a probabilistic method:
  - ➢ Cannot directly return more than one result (*k*-best tagging can be added but is not built-in [Brill 1995, sec. 4.4])
  - ➢ Cannot measure confidence of results (through [Florian et al. 2000] estimate probabilities by converting transformation rule lists to decision trees and computing distributions over equivalence classes)

TBL flowchart