Intelligent Systems Principles and Programming

Xiaoqing Zheng
zhengxq@fudan.edu.cn

ELIZA

User:	**ELIZA:**
Men are all alike.	In what way?
They are always bugging us about something or other.	Can you think of a specific example?
Well, my boyfriend made me come here.	Your boyfriend made you come here.
He says I’m depressed much of the time.	I am sorry to hear you are depressed.
It’s true, I am unhappy.	Do you think coming here will help you not to be unhappy?
I need some help; that much seems certain.	What would it mean to you if you got some help?
Perhaps I could learn to get along with my mother.	Tell me more about your family.
My mother takes care of me.	Who else in your family takes care of you?
My father.	Your father.
You are like my father in some ways.	What resemblance do you see?

ELIZA

Word	Rank	Pattern	Outputs
alike	10	?X	In what way?
			What resemblance do you see?
are	3	?X are you ?Y	Would you prefer it if I were not ?Y
	3	?X are ?Y	What if they were not ?Y?
always	5	?X	Can you think of a specific example?
			When
			Really, always?
what	2	?X	Why do you ask?
			Does that interest you?

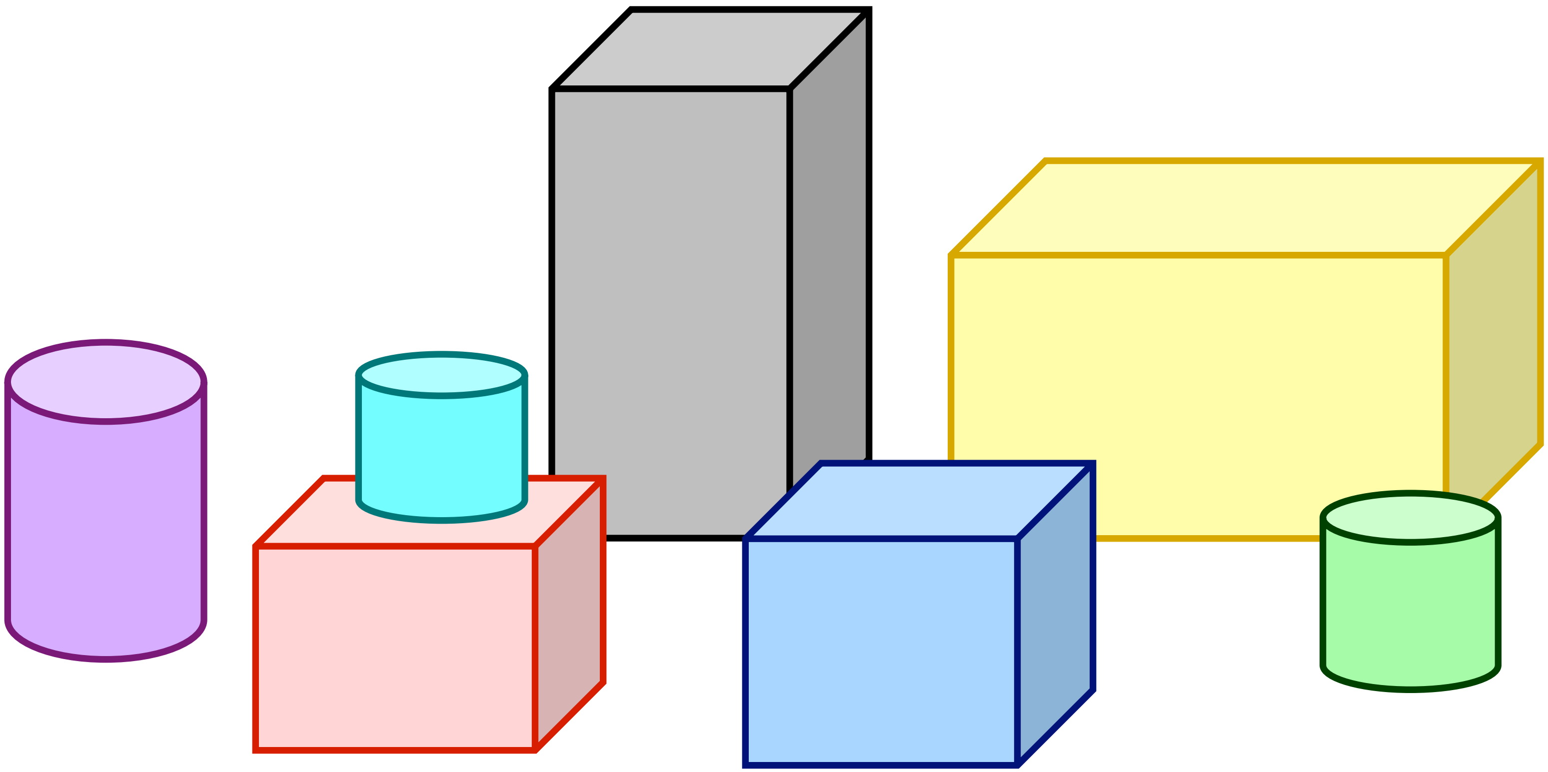
User: Necessity is the mother of invention.
ELIZA: Tell me more about your family.

User: Green the adzabak are the a ran four.
ELIZA: What if they were not the a ran four?

Three major issues

- A large amount of human ***knowledge*** is assumed.
- Language is ***pattern*** based.
- Language acts are the product of ***agents***, either human or computer.

SHRDLU



A blocks world

Different levels of natural language

- ***Prosody*** deals with the rhythm and intonation of language.
- ***Phonology*** examines the sounds that are combined to form language.
- ***Morphology*** is concerned with the components (morphemes) that make up words.
- ***Syntax*** studies the rules for combining words into legal phrases and sentences, and the use of those rules to parse and generate sentences.

Different levels of natural language

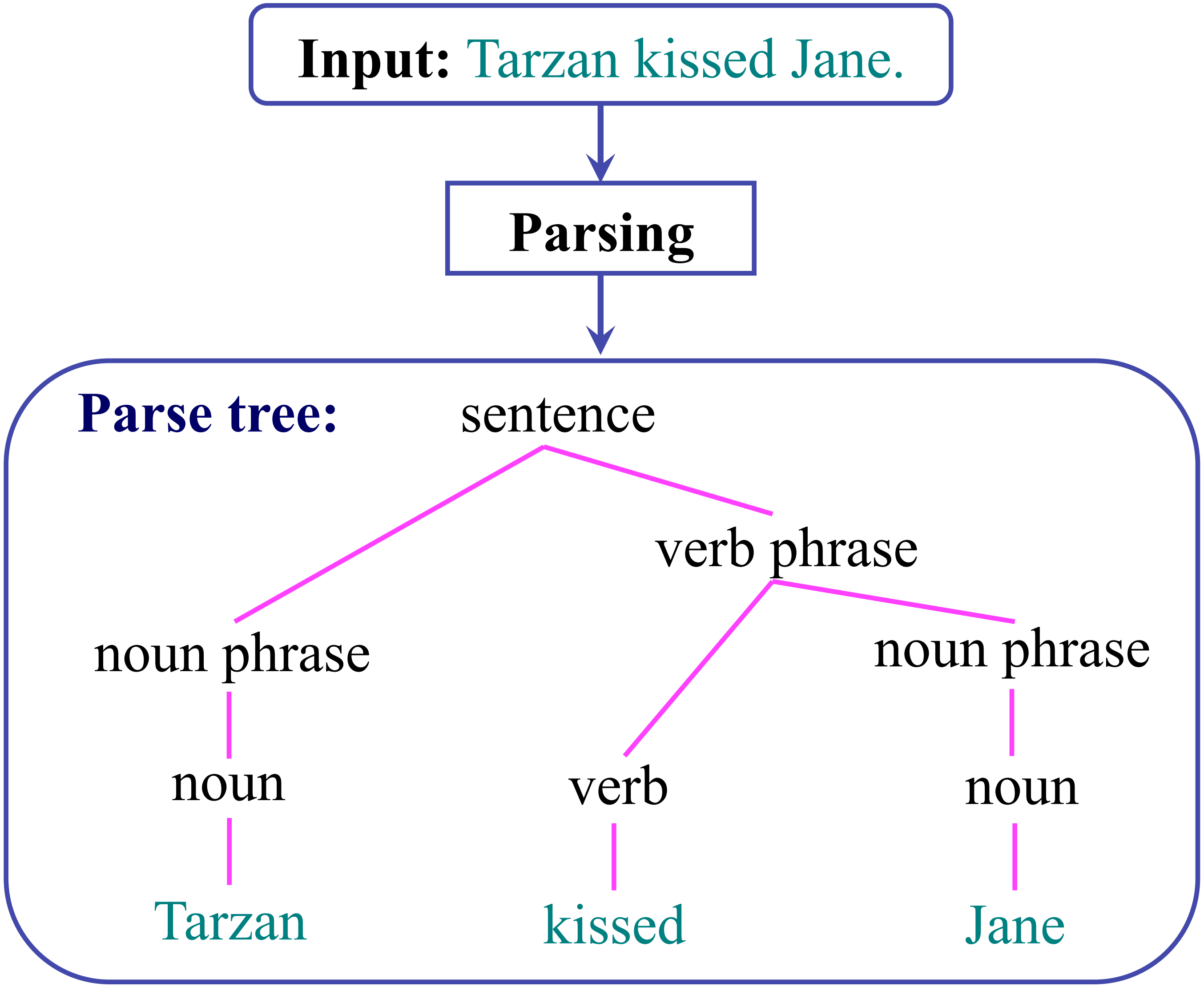
- ***Semantics*** considers the meaning of words, phrases, and sentences and the ways in which meaning is conveyed in natural language expressions.
- ***Pragmatics*** is the study of the ways in which language is used and its effects on the listener.
- ***World knowledge*** includes knowledge of the physical world, the world of human social interaction, and the role of goals and intentions in communication.

Green frogs have large noses.

Green ideas have large noses.

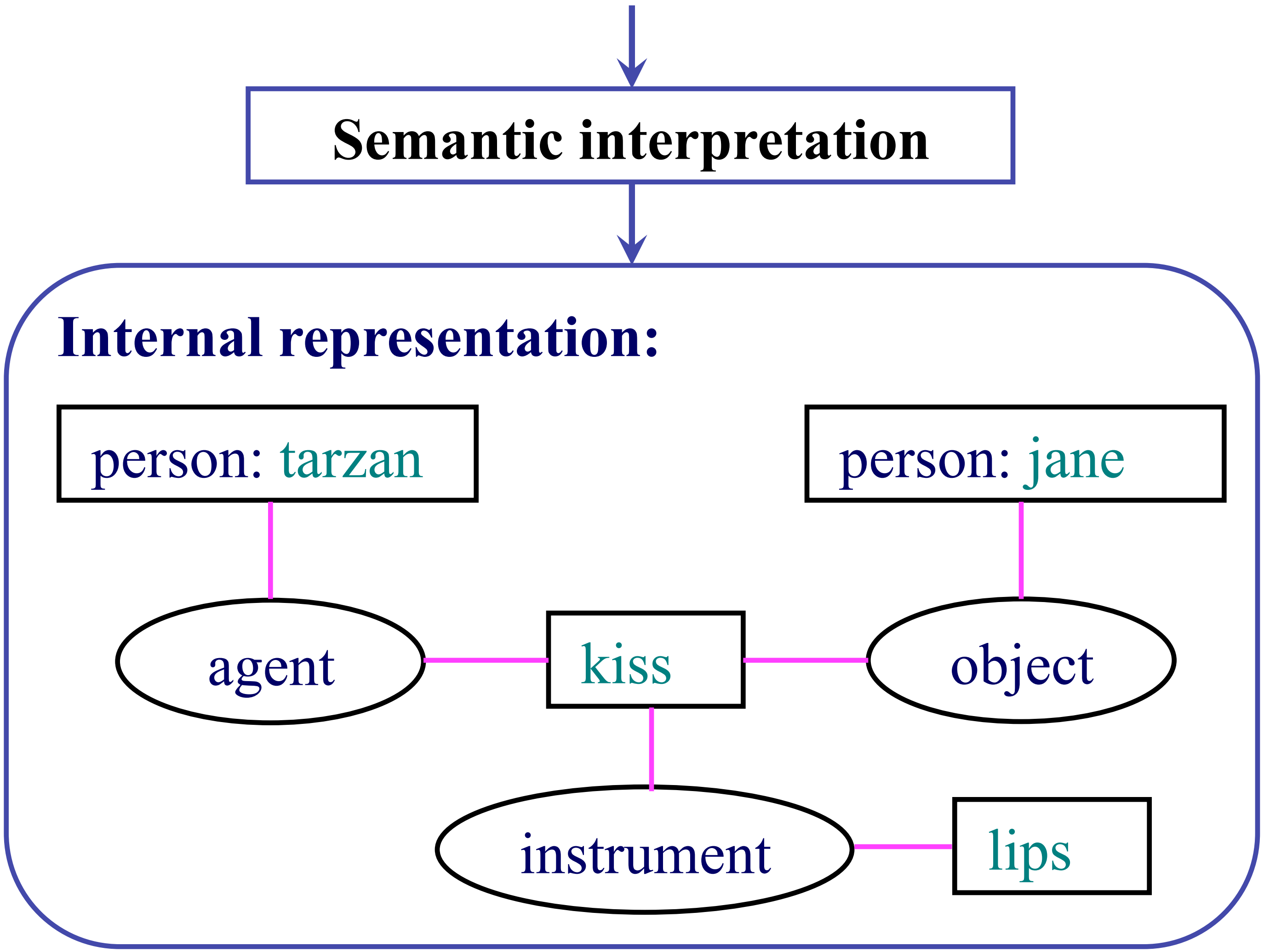
Large have green ideas nose.

Stages



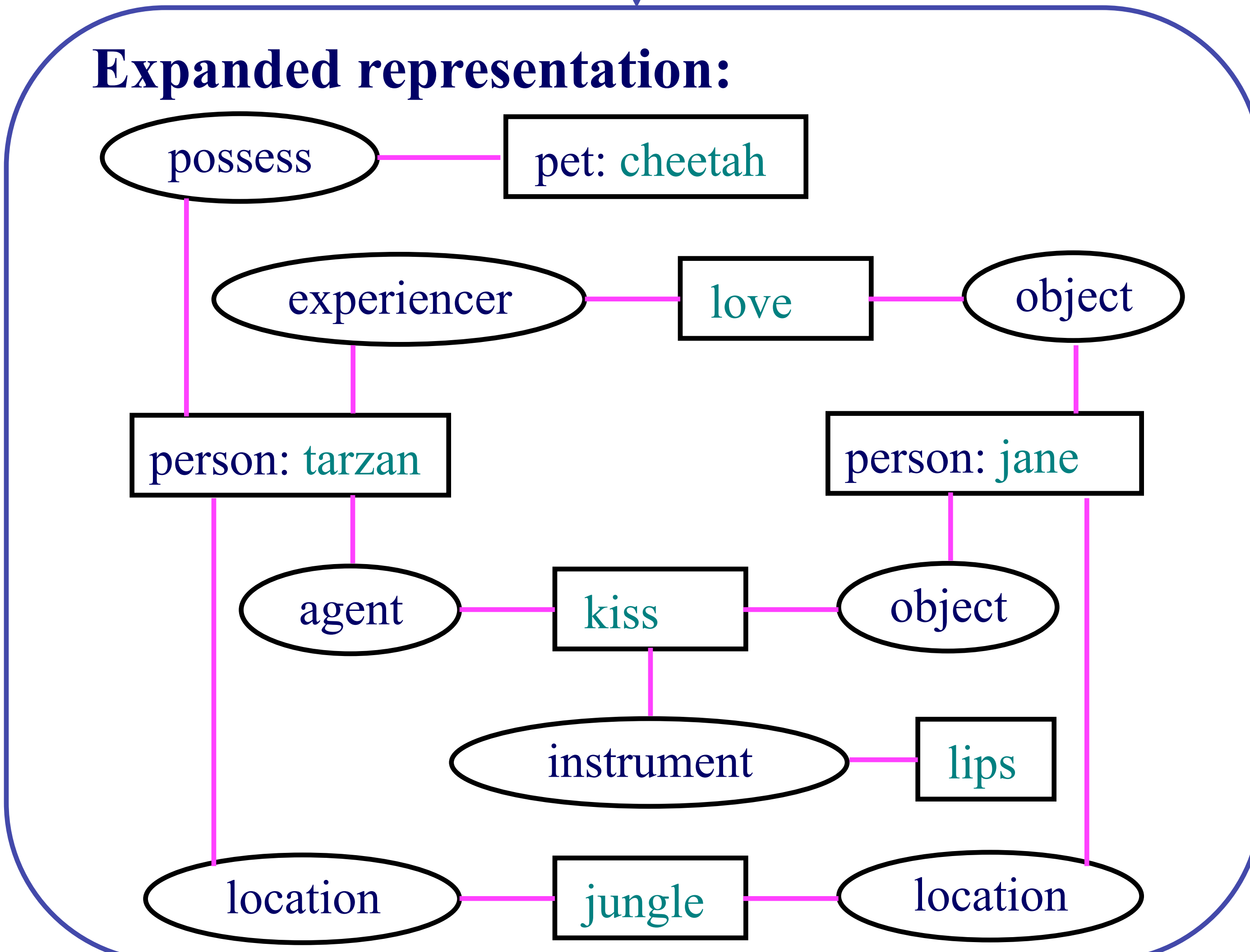
Input: Tarzan kissed Jane.
Parsing
Parse tree:
sentence
verb phrase
noun phrase
noun phrase
noun
verb
noun
Tarzan
kissed
Jane

Stages

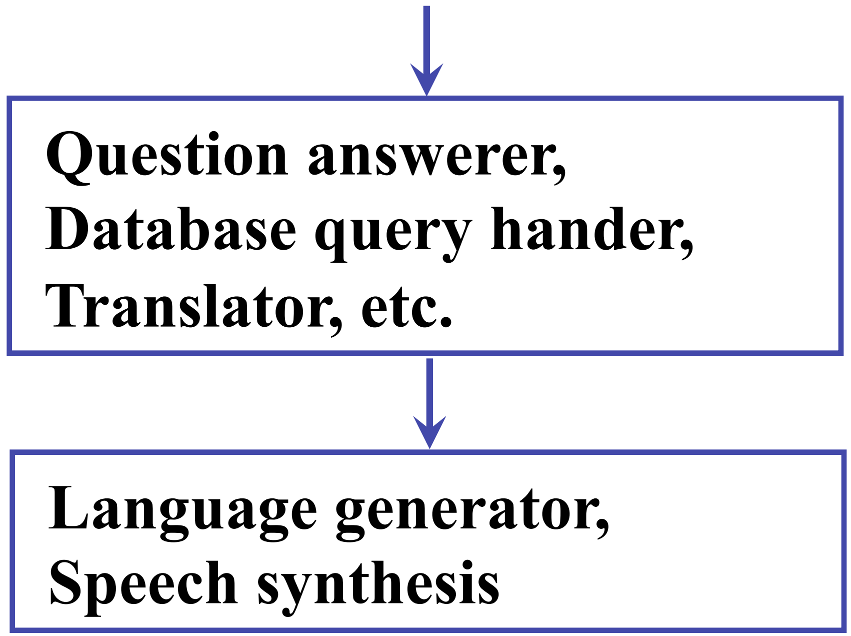


Semantic interpretation
Internal representation:
person: tarzan
person: jane
agent
kiss
object
instrument
lips

Contextual / world knowledge interpretation



Stages


Question answerer,
Database query hander,
Translator, etc.
Language generator,
Speech synthesis

Dialog example

明天下午去看场电影啊?

好呀！去哪里？

我们去学校旁边的万达影城吧。

Chinese word segmentation

步骤一：中文分词

交叉歧义

明天 / 下午 / 去 / 看 / 场 / 电影 / 啊 / ？

好 / 呀 / ！ / 去 / 哪里 / ？

我们 / 去 / 学校 / 旁边 / 的 / 万达 / 影城 / 吧 / 。

命名识别

Challenges of word segmentation

各种歧义

- 组合歧义
  以我 / 个人 / 的名义
  他一 / 个 / 人 / 在家
- 交叉歧义
  从 / 小学 / 到中学
  从小 / 学 / 计算机
- 理解歧义
  美国 / 会 / 采取行动
  美 / 国会 / 采取行动

末知词汇

- 时间、数量
  2012年5月22日
  一市斤
  356克
- 人名、机构名、地名
  李维汉
  阿里巴巴
  上海浦东张江镇
- 外文翻译及缩写
  阿诺德·施瓦辛格
  FDA
- 专用名词缩写
  中航
  央行
- 新词、不规范词
  菜鸟
  菇凉

Challenges of word segmentation

构词方式

- **重叠词**
  AA式：爸爸、宝宝、星星
  AABB式：大大咧咧、形形色色、漂漂亮亮
- **派生词**
  前缀 + 词根：阿姨、老虎、老婆
  词根 + 后缀：椅子、木头、鸟儿
  词根 + 中缀 + 词根：对得起、来得及
- **词转化为短语**
  动宾结构式：鞠躬 → 鞠个躬 → 鞠个九十度的躬
  偏正或联合结构：同学 → 同过三年学
  补充结构式：达到 → 达得/不到
- **短语转化为词**
  ABCD → AC式：土地改革 → 土改、地下铁道 → 地铁
  ABCD → AD式：空中小姐 → 空姐、高等院校 → 高校
  截段简缩：中国南极长城站 → 长城站、复旦大学 → 复旦
  综合简缩：联合国安全理事会 → 安全理事会 → 安理会

Part-of-speech tagging

步骤二：词性标注

我们 / 去 / 学校 / 旁边 / 的 / 万达 / 影城 / 吧 / 。

我们	去	学校	旁边	的	万达	影城	吧	。
PN Pronoun	**VV** Verb	**NN** Noun	**LC** Localizer	**DEG** Associative marker	**NR** Proper noun	**NN** Noun	**SP** Sentence-final particle	**PU** Punctuation

兼类现象

- 动名兼类
  动词：报告上级
  名词：一份报告
- 形名兼类
  形容词：科学价值
  名词：方法不科学
- 形动兼类
  形容词：学校生活很丰富
  动词：丰富业余生活

Named entity recognition

步骤三：命名实体识别

我们 / 去 / 学校 / 旁边 / 的 / 万达 / 影城 / 吧 / 。

我们	去	学校	旁边	的	万达	影城	吧	。
PN	VV	NN	LC	DEG	NR	NN	SP	PU

机构：电影院

七类实体

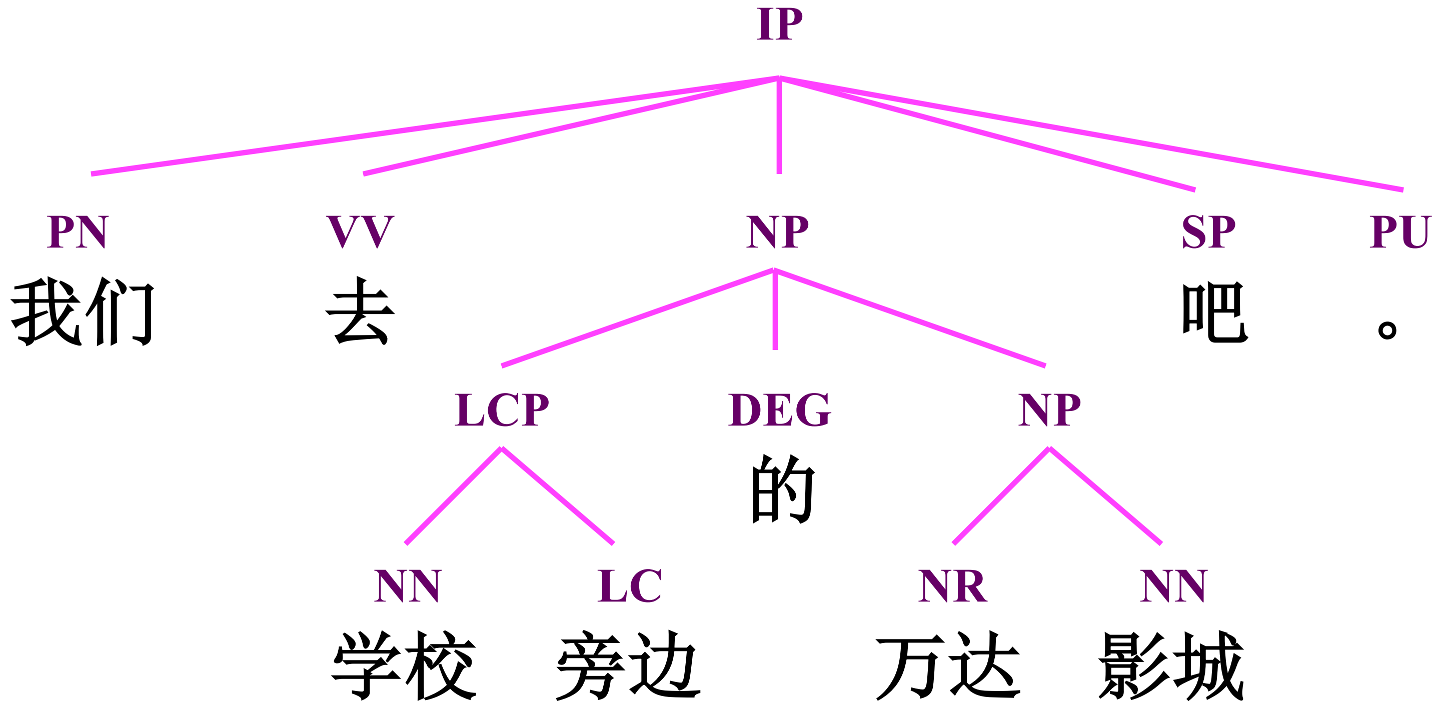
人名
机构名
地名
时间
日期
货币
百分比

困难原因

- 没有明显标志
- 生成规律复杂
- 普通用字混淆

Constituent parsing

步骤四：语法分析



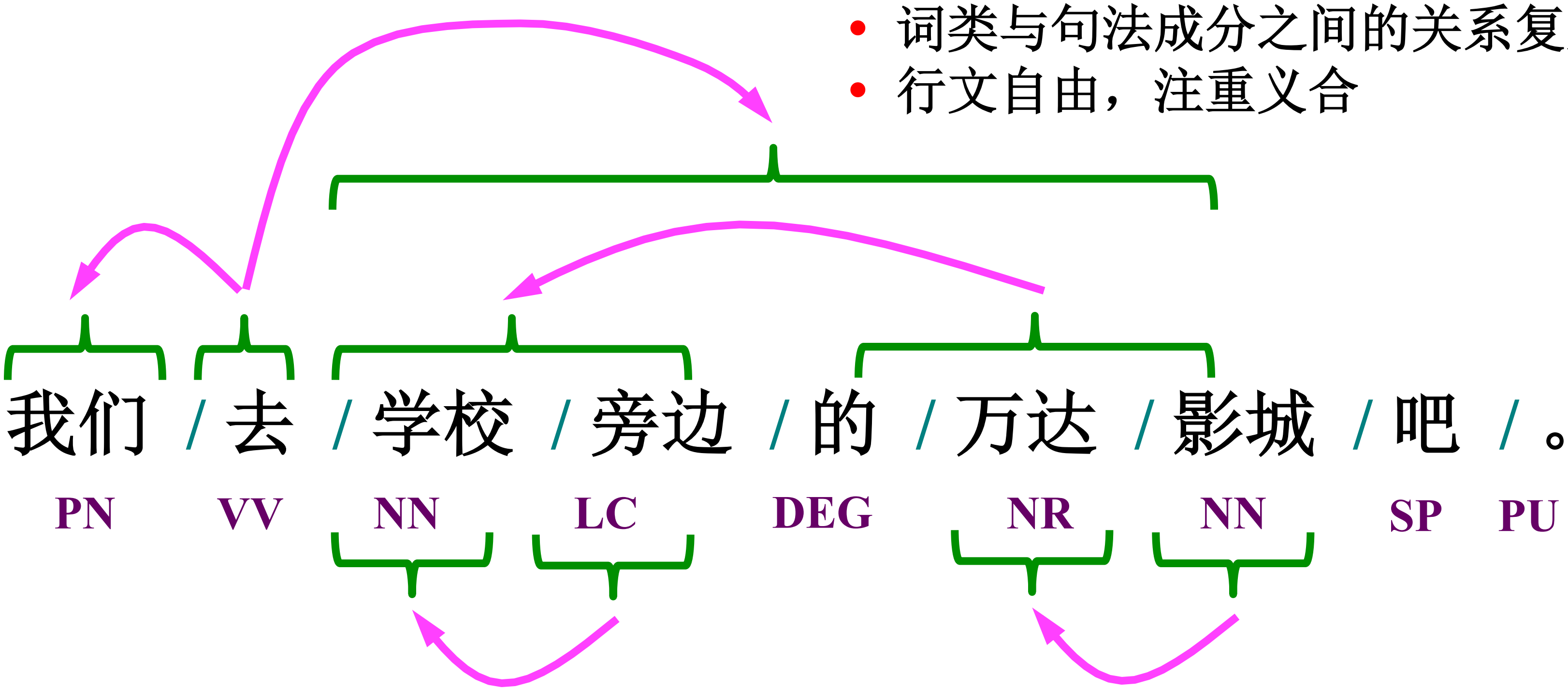
成分语法分析

Dependency parsing

步骤四：语法分析

主要挑战

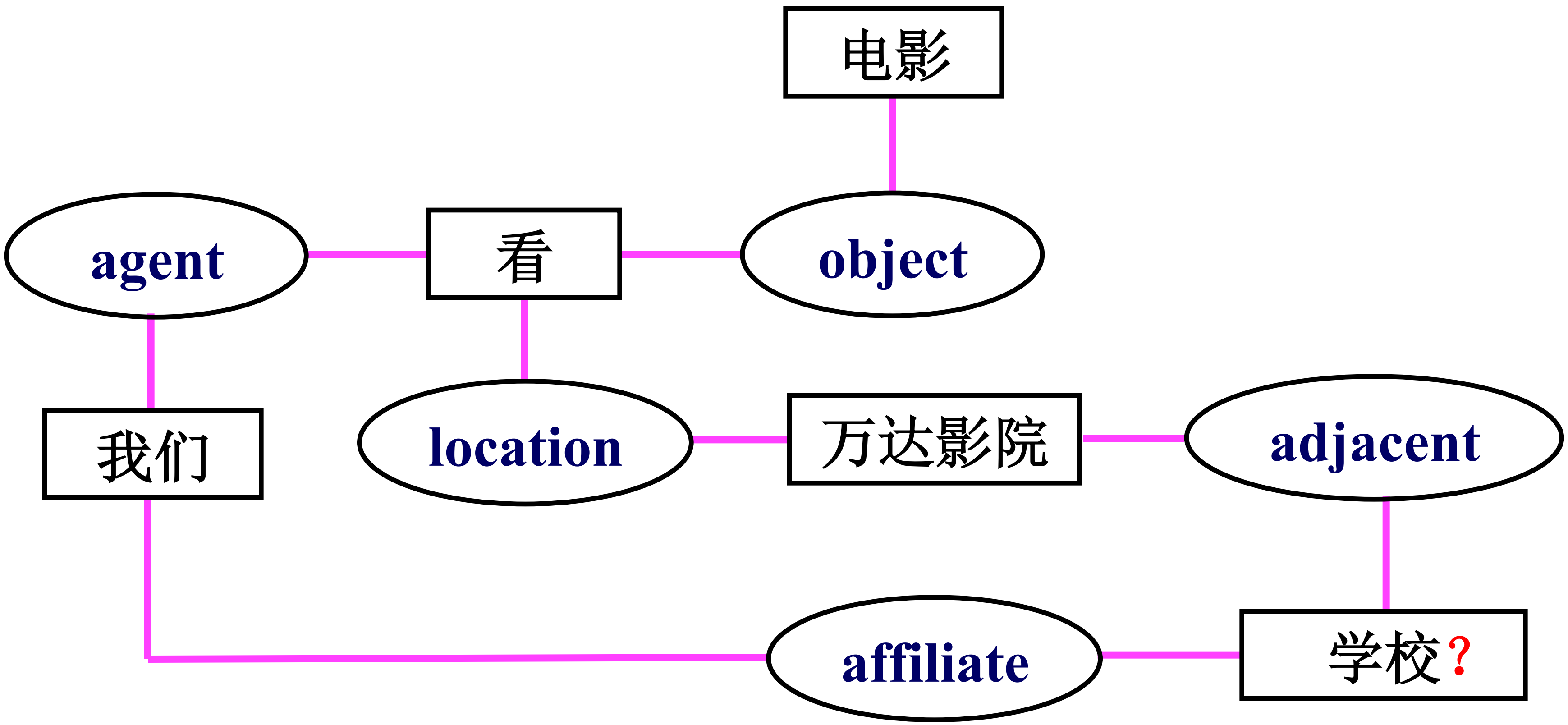
- 不依赖严格的形态变化
- 词类与句法成分之间的关系复杂
- 行文自由，注重义合



依存语法分析

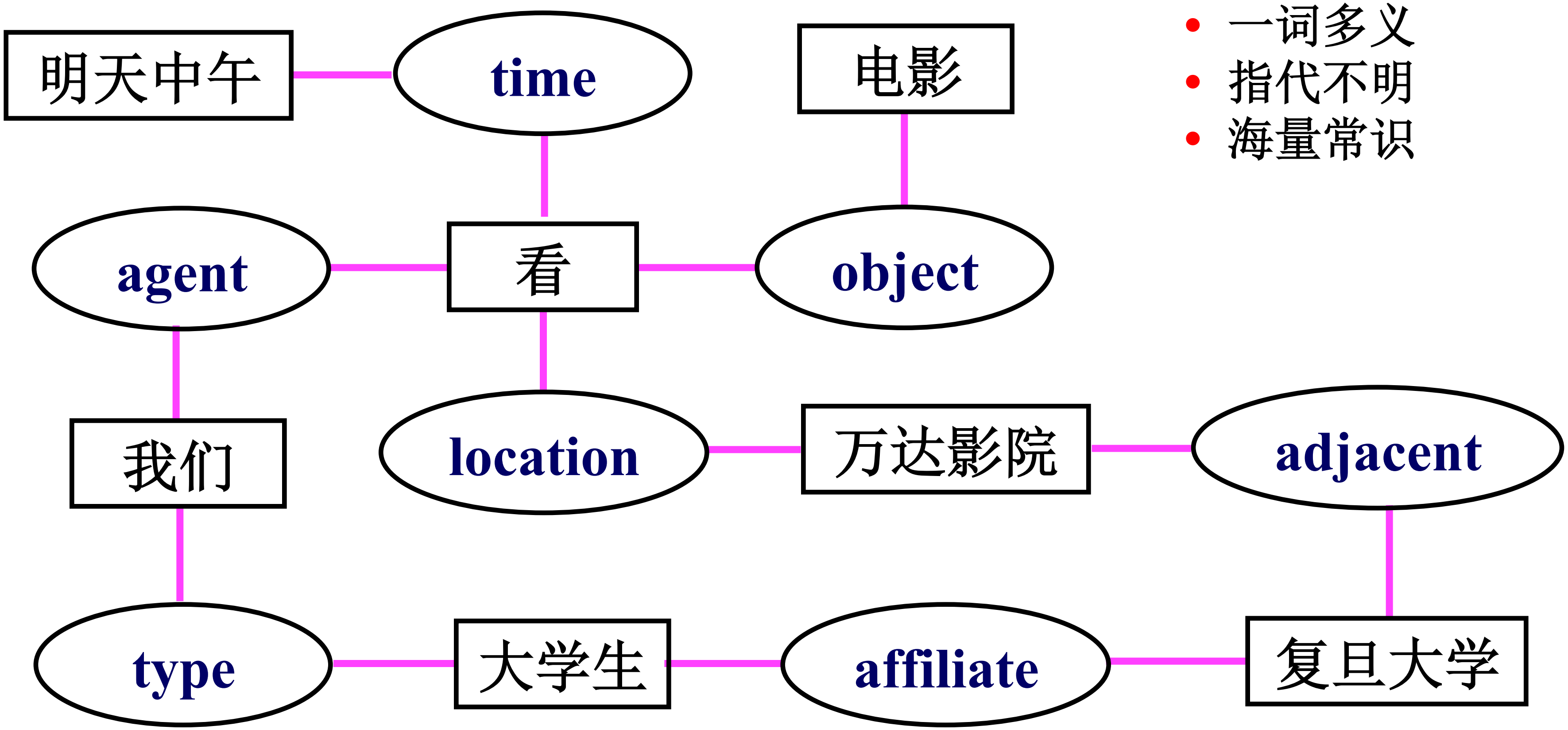
Semantic interpretation

步骤五：语义理解



Background knowledge

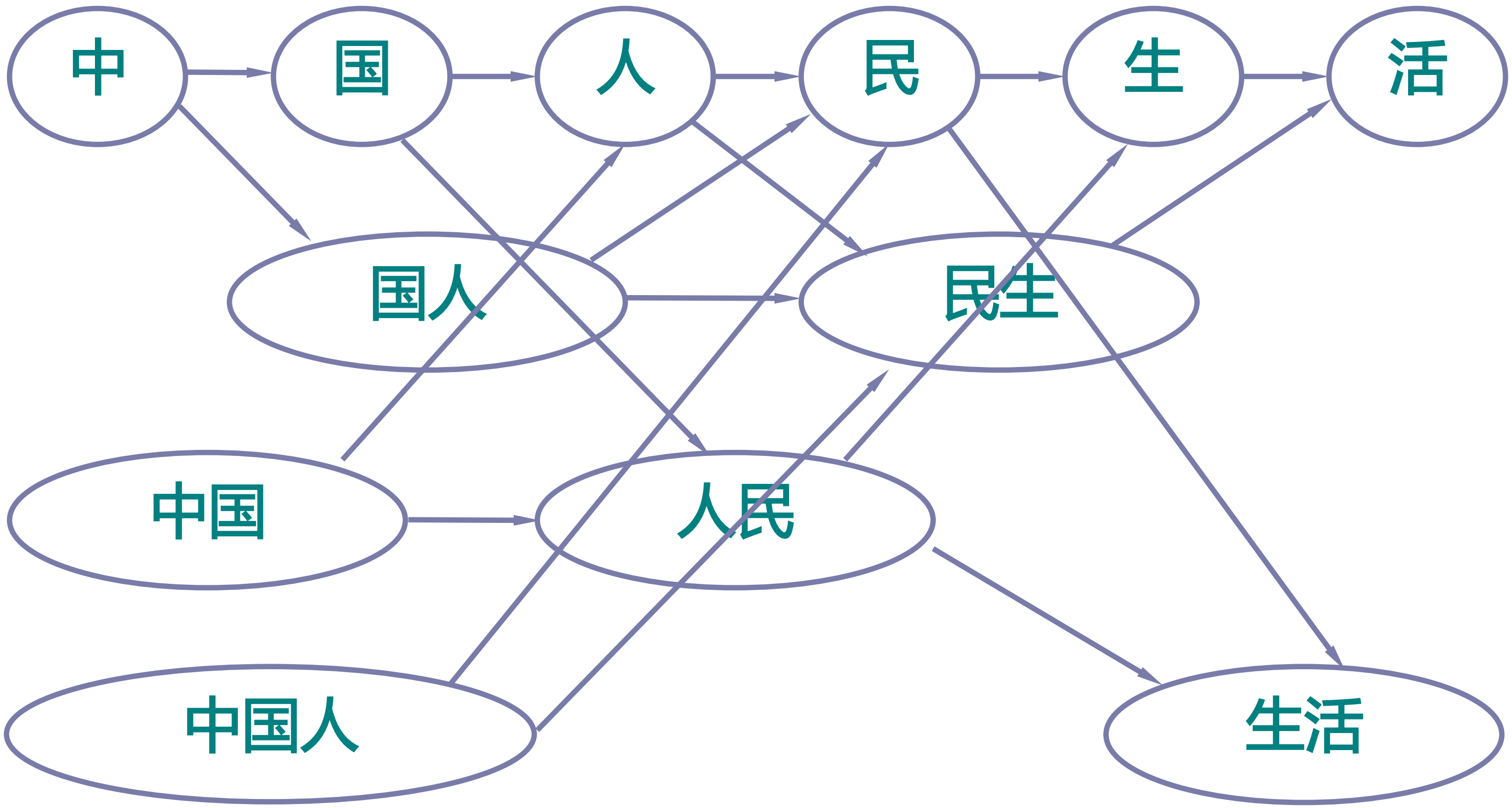
步骤六：背景融合



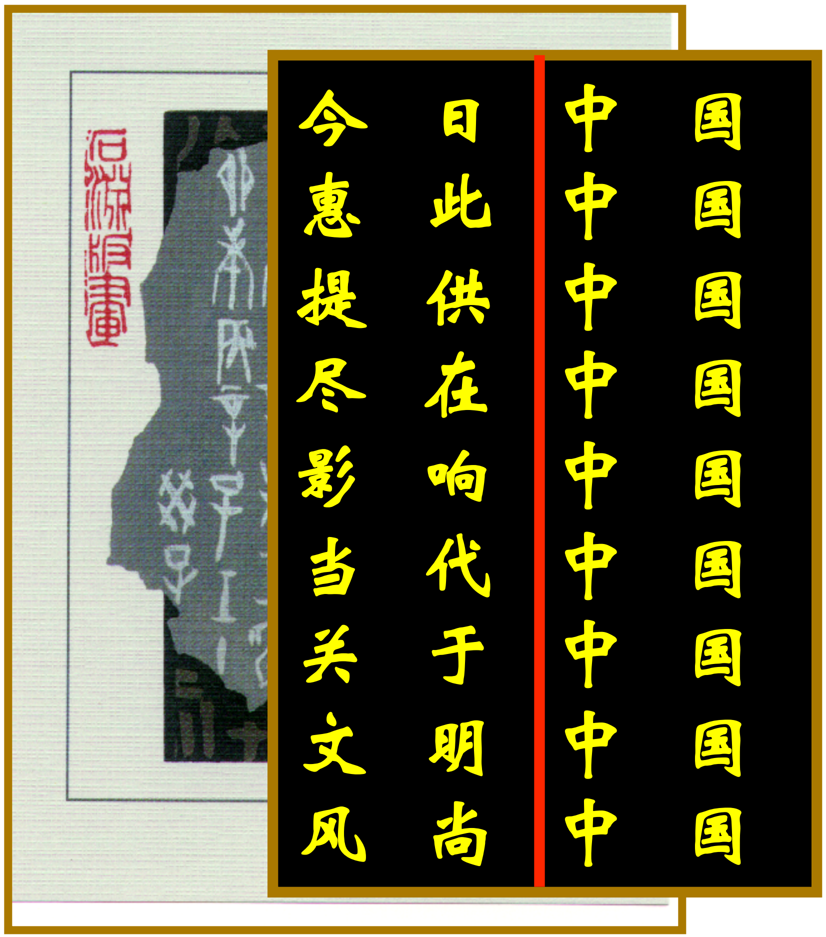
主要困难

- 一词多义
- 指代不明
- 海量常识

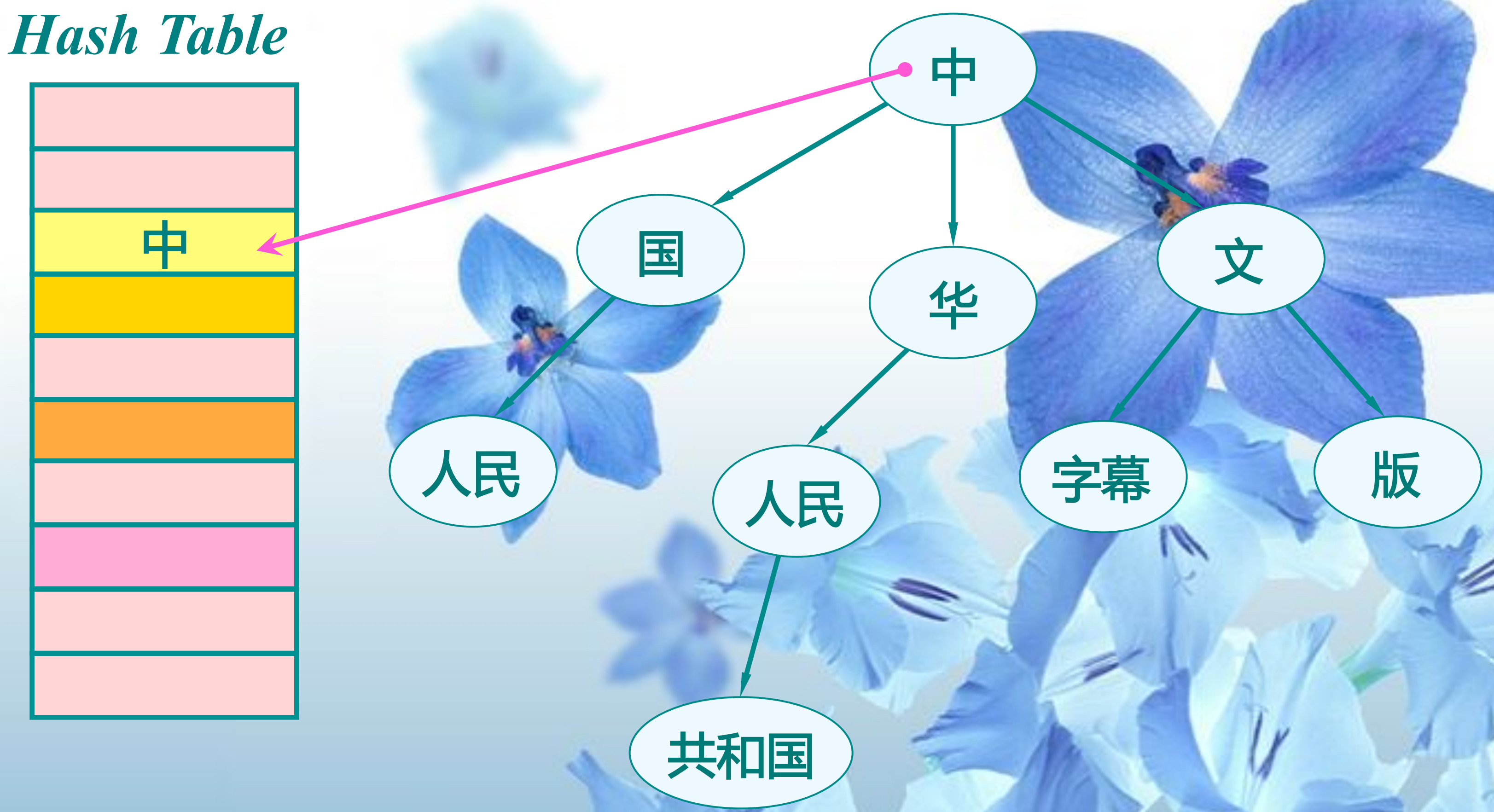
Chinese word segmentation



Unknown word detection



树型词库



Syntax: context-free grammars

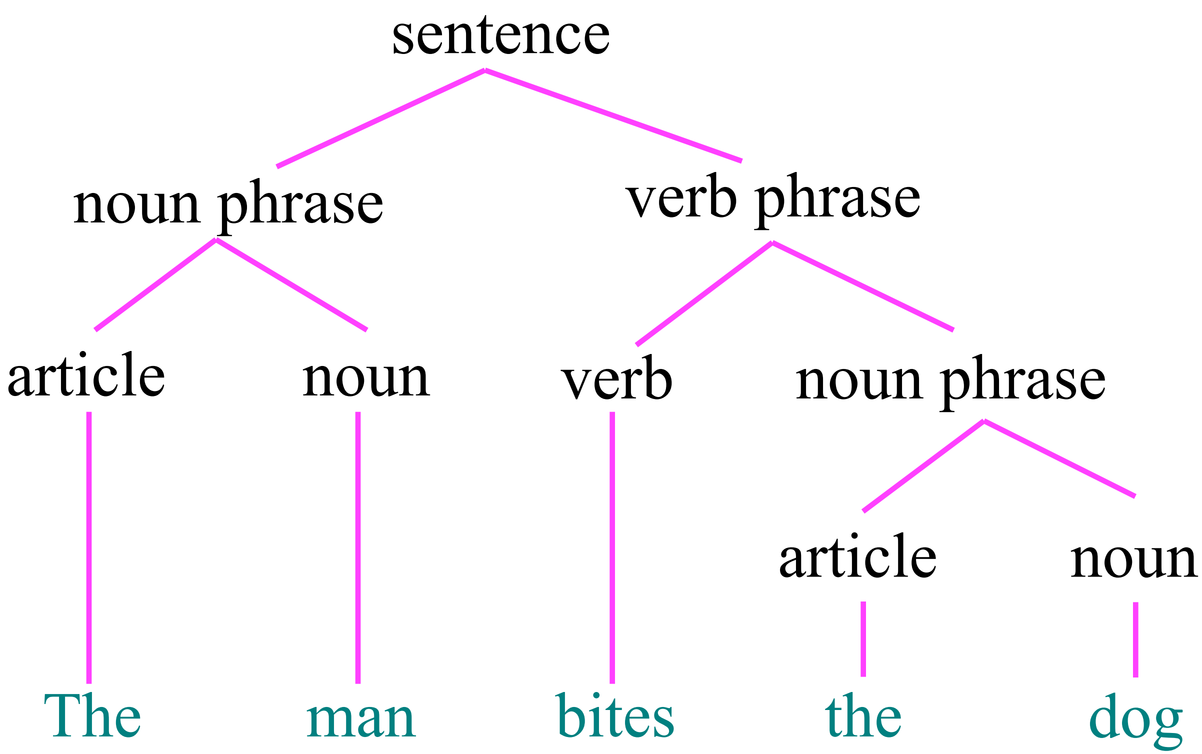
1. sentence → noun_phrase verb_phrase
2. noun_phrase → noun
3. noun_phrase → article noun
4. verb_phrase → verb
5. verb_phrase → verb noun_phrase
6. article → a
7. article → the
8. noun → man
9. noun → dog
10. verb → likes
11. verb → bites

Derivation

The man bites the dog.

String	Apply rule
sentence	**1**
noun_phrase verb_phrase	**3**
article noun verb_phrase	**7**
The noun verb_phrase	**8**
The man verb_phrase	**5**
The man verb noun_phrase	**11**
The man bites noun_phrase	**3**
The man bites article noun	**7**
The man bites the noun	**9**
The man bites the dog	

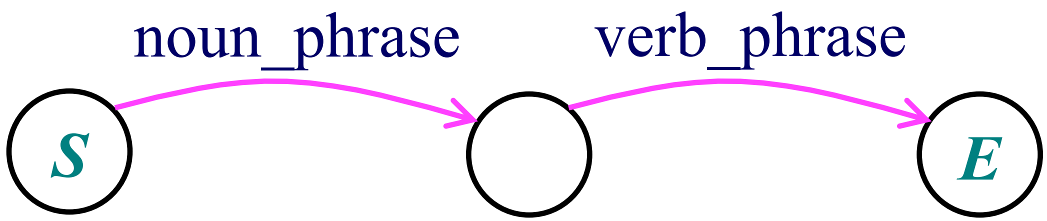
Parse tree



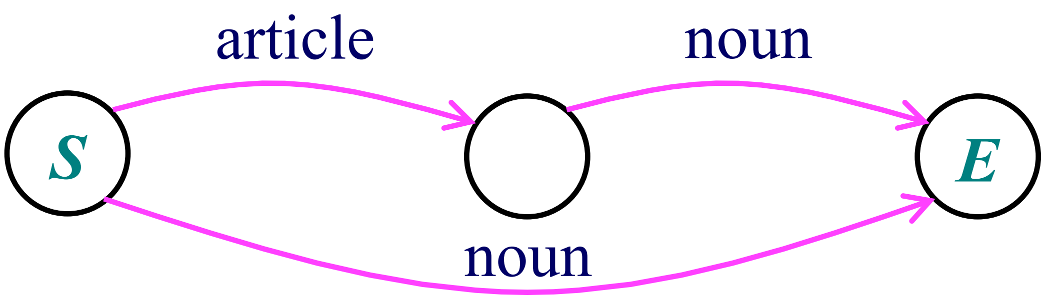
sentence
noun phrase
verb phrase
article
noun
verb
noun phrase
article
noun
The
man
bites
the
dog

Transition network parsers

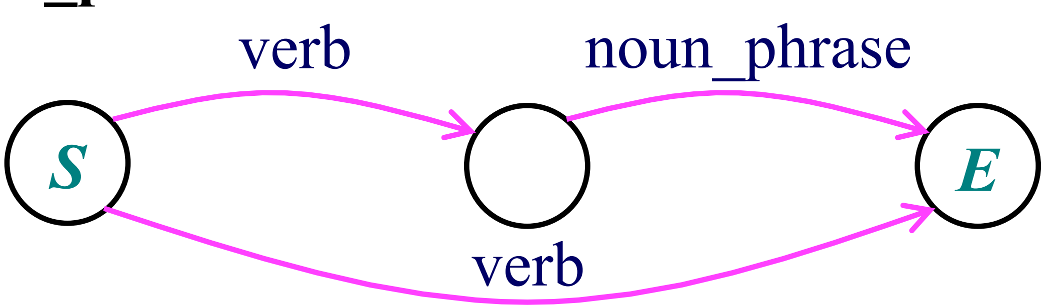
sentence



noun_phrase

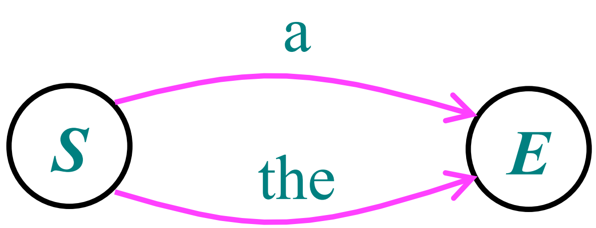


verb_phrase

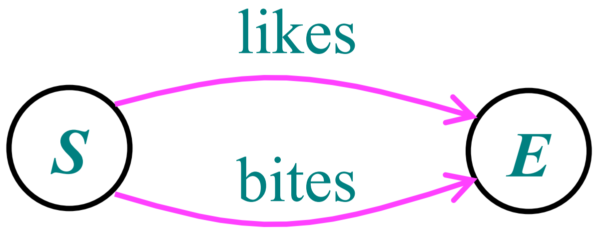


Transition network parsers

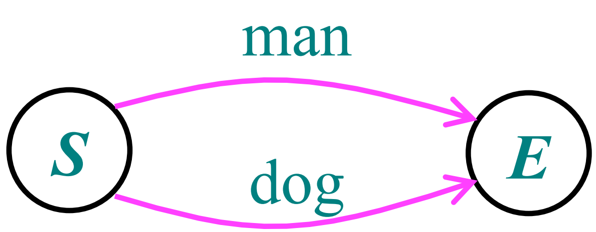
article



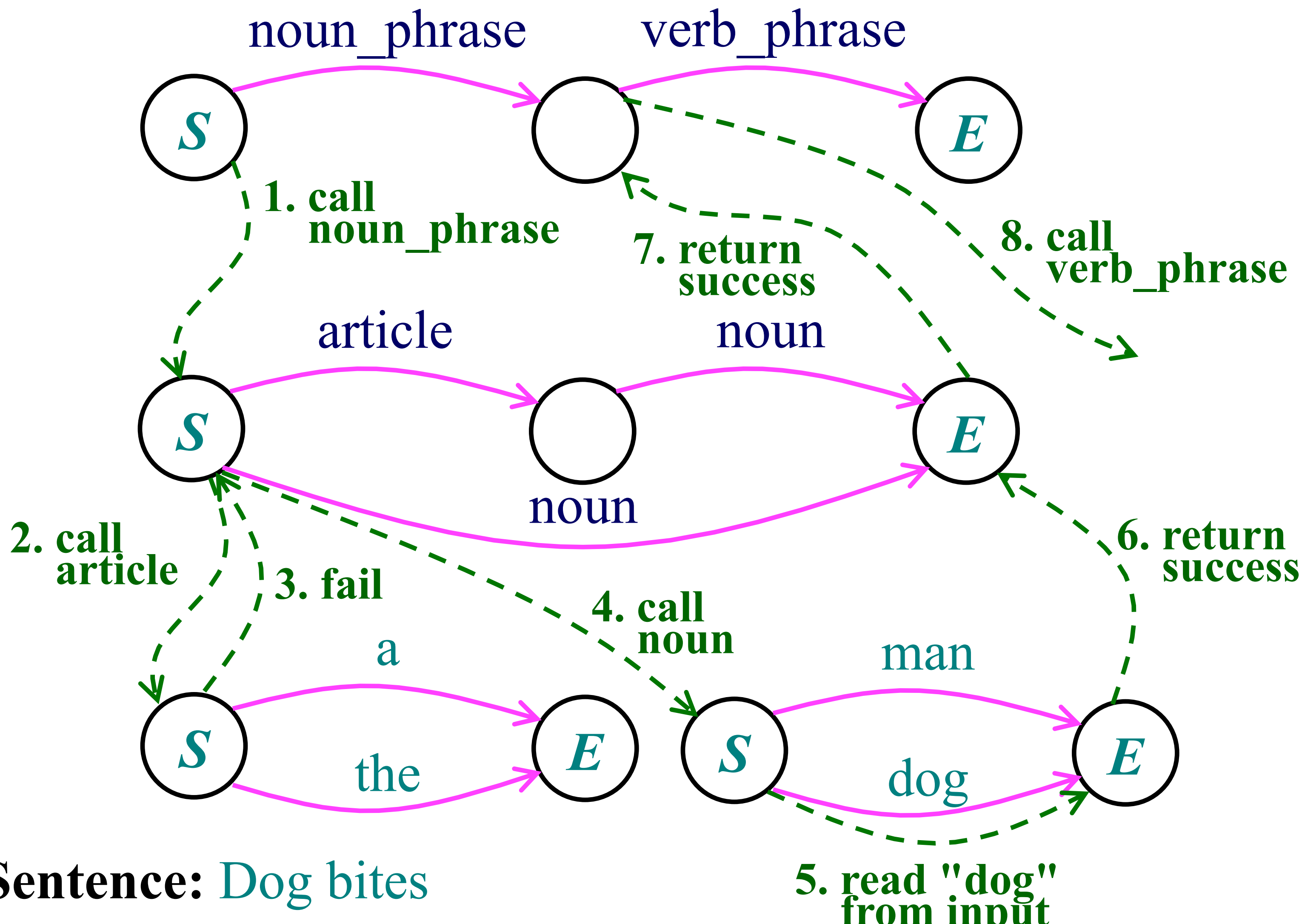
verb



noun



Trace of a transition network parse



Sentence: Dog bites

Context-sensitive grammars

- A ***context-free grammar*** allows rules to have only a ***single*** nonterminal on their left-hand side.

  It will accept sentences such as:

  A men likes a dogs.

- The parser will accept these sentences because the current rules cannot use ***context*** to determine when the singular or plural forms must be coordinated.
- The ***context-sensitive languages*** form a proper superset of the context-free languages. There are defined using context-sensitive grammars which allow more than one symbol on the left-hand side of a rule and make it possible to define a ***context*** in which that rule can be applied.

Context-sensitive grammars

1. sentence → noun_phrase verb_phrase
2. noun_phrase → article number noun
3. noun_phrase → number noun
4. number → singular
5. number → plural
6. article singular → a singular
7. article singular → the singular
8. article plural → some plural
9. article plural → the plural
10. singular noun → dog singular
11. singular noun → man singular
12. plural noun → dogs plural
13. plural noun → men plural

Context-sensitive grammars

14. singular verb_phrase → singluar verb
15. plural verb_phrase → plural verb
16. singular verb → bites
17. singular verb → likes
18. plural verb → bite
19. plural verb → like

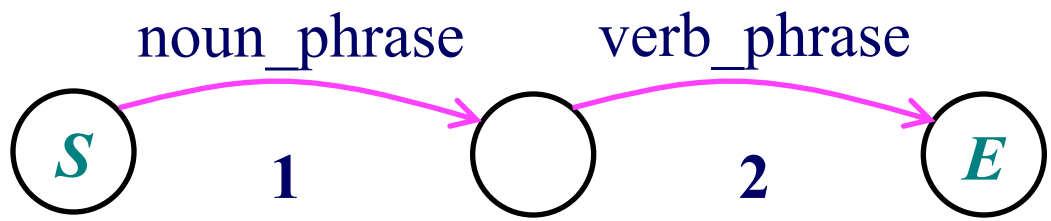
Sentence
noun_phrase verb_phrase
article plural noun verb_phrase
The plural noun verb_phrase
The dogs plural verb_phrase
The dogs plural verb
The dogs bite

Disadvantages

- Context-sensitive grammars increase drastically the ***number of rules*** and nonterminals in the grammar.
- They obscure the ***phrase structure*** of the language that is so ***clearly*** represented in the context-free rules.
- By attempting to handle more complicated checks for agreement and ***semantic consistency*** in the grammar itself, they lose many of the benefits of ***separating*** the syntactic and semantic components of language.
- Context-sensitive grammars do not address the problem of building a ***semantic representation*** of the meaning of the text.

Augmented transition network

sentence



function sentence-1:

begin

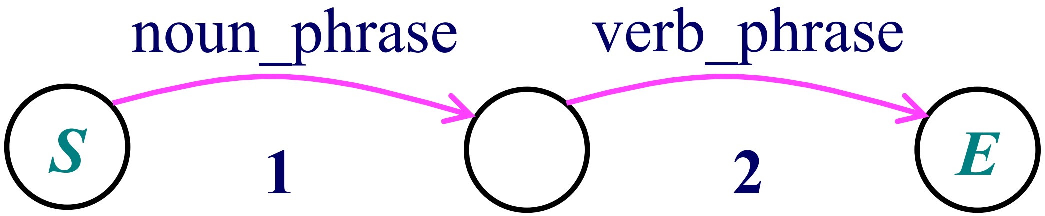
noun_phrase := structure returned by noun_phrase networks;

sentence.subject := noun_phrase;

end

Augmented transition network

sentence




```
function sentence-2:
  begin
    verb_phrase := structure returned by verb_phrase networks;
    if noun_phrase.number = verb_phrase.number
      then begin
        sentence.verb_phares := verb_phrase;
        return sentence
      end
    else fail
end
```

Grammars and logic programming

Grammar

$S \rightarrow NP\ VP$

$s(P1, P3) \text{ :- } np(P1, P2), vp(P2, P3)$

Sentence

John ate the cat

$word(john, 1, 2)$
$word(ate, 2, 3)$
$word(the, 3, 4)$
$word(cat, 4, 5)$

Grammars and logic programming

Lexicon

isart(the)
isname(john)
isverb(ate)
isnoun(cat)

Syntactic category

n(I, O) :- word(Word, I, O), isnoun(Word)
art(I, O) :- word(Word, I, O), isart(Word)
v(I, O) :- word(Word, I, O), isverb(Word)
name(I, O) :- word(Word, I, O), isname(Word)

Grammars and logic programming

Prolog-based representation of grammar

1. $s(P1, P3)$:- $np(P1, P2)$, $vp(P2, P3)$
2. $np(P1, P3)$:- $art(P1, P2)$, $n(P2, P3)$
3. $np(P1, P2)$:- $name(P1, P2)$
4. $pp(P1, P3)$:- $p(P1, P2)$, $np(P2, P3)$
5. $vp(P1, P2)$:- $v(P1, P2)$
6. $vp(P1, P3)$:- $v(P1, P2)$, $np(P2, P3)$
7. $vp(P1, P3)$:- $v(P1, P2)$, $pp(P2, P3)$

Question

$s(1, 5)$

A trace of Prolog-based parse

Current State		Backup States
1.	$s(1, 5)$	
2.	$np(1, P2), vp(P2, 5)$	
3.	$art(1, P2'), n(P2', P2), vp(P2, 5)$	$name(1, P2), vp(P2, 5)$
4.	$name(1, P2), vp(P2, 5)$	
5.	$vp(2, 5)$	
6.	$v(2, 5)$	$v(2, P2), np(P2, 5)$
		$v(2, P2), pp(P2, 5)$
7.	$v(2, P2), np(P2, 5)$	$v(2, P2), pp(P2, 5)$
8.	$np(3, 5)$	$v(2, P2), pp(P2, 5)$
9.	$art(3, P2), n(P2, 5)$	$name(3, 5)$
		$v(2, P2), pp(P2, 5)$
10.	$n(4, 5)$	$name(3, 5)$
11.	*nil*	$v(2, P2), pp(P2, 5)$
		$name(3, 5)$
	John ate the cat	$v(2, P2), pp(P2, 5)$

Top-down and bottom-up

The can holds the water

can may be an *AUX*, *V*, or *N*

$S \longrightarrow NP \;\; VP$

$NP \longrightarrow ART \;\; ADJ \;\; N \;\; VP$

$NP \longrightarrow ART \;\; N \;\; VP$

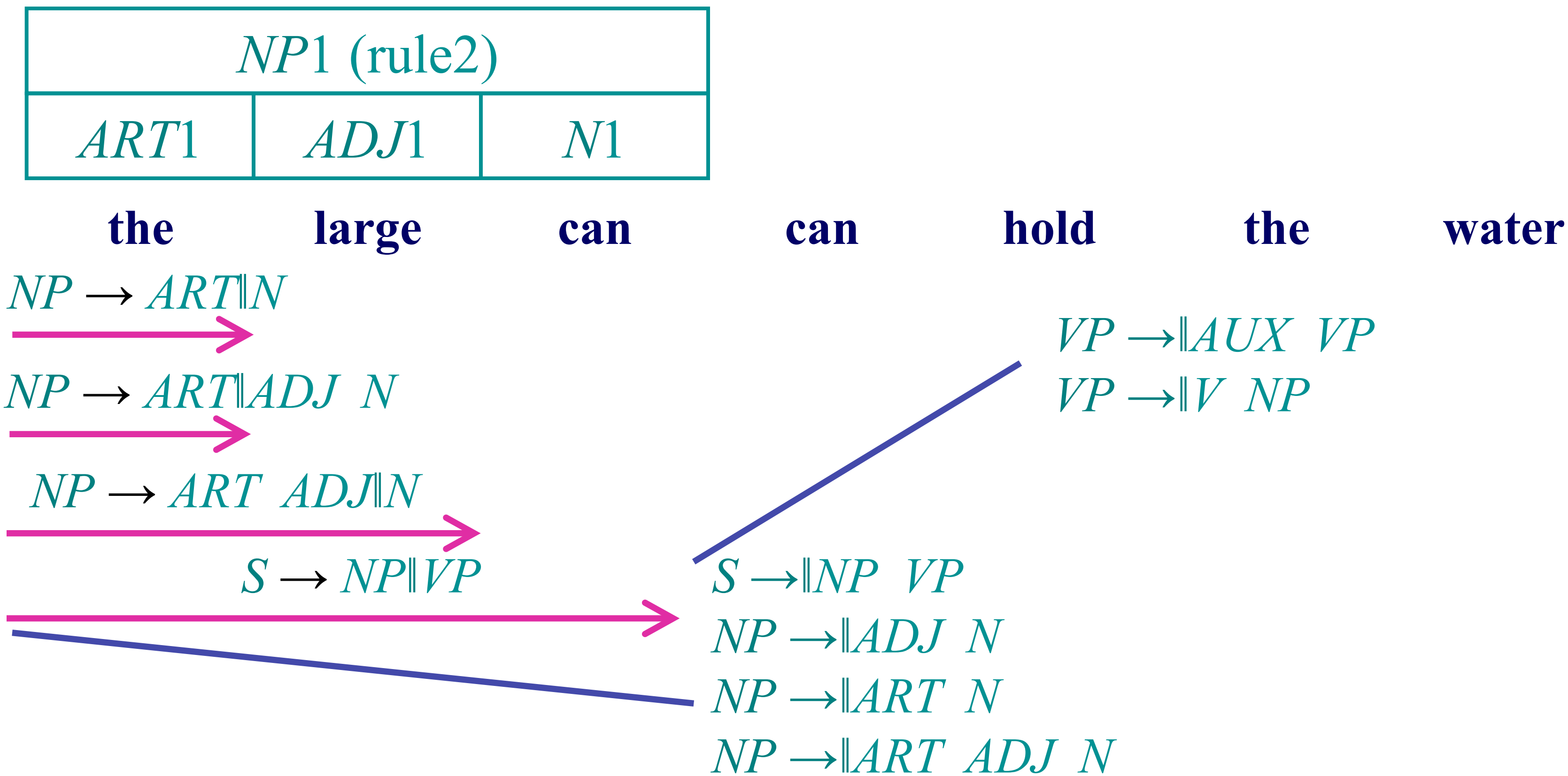
$NP \longrightarrow ADJ \;\; N \;\; VP$

Top-down chart parser

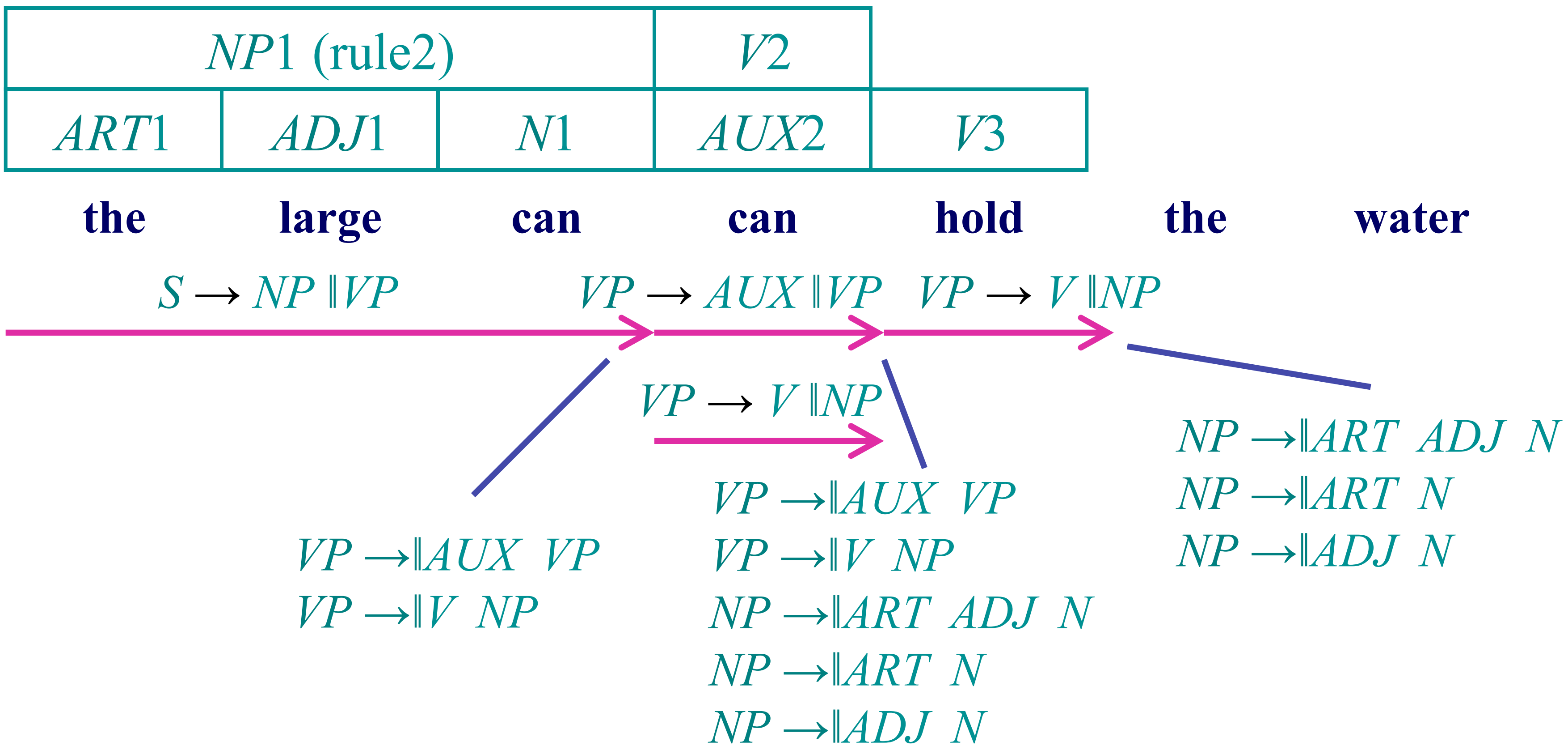
the **large** **can** **can** **hold** **the** **water**

$$S \longrightarrow \| NP \;\; VP$$
$$NP \longrightarrow \| ADJ \;\; N$$
$$NP \longrightarrow \| ART \;\; N$$
$$NP \longrightarrow \| ART \;\; ADJ \;\; N$$

Top-down chart parser



Top-down chart parser



Top-down chart parser

<table>
<tr><td colspan="7">S1 (rule 1 with NP1 & VP2)</td></tr>
<tr><td colspan="3"></td><td colspan="4">VP2 (rule 5 with AUX2 and VP1)</td></tr>
<tr><td colspan="4"></td><td colspan="3">VP1 (rule 6 with V3 and NP2)</td></tr>
<tr><td colspan="3">NP1 (rule2)</td><td>V2</td><td></td><td colspan="2">NP2 (rule3)</td></tr>
<tr><td>ART1</td><td>ADJ1</td><td>N1</td><td>AUX2</td><td>V3</td><td>ART2</td><td>N4</td></tr>
<tr><td>the</td><td>large</td><td>can</td><td>can</td><td>hold</td><td>the</td><td>water</td></tr>
</table>

Dictionary

word: a
part_of_speech: article
root: a
number: singular

word: like
part_of_speech: verb
root: like
number: plural

word: bite
part_of_speech: verb
root: bite
number: plural

word: likes
part_of_speech: verb
root: like
number: singular

word: bites
part_of_speech: verb
root: bite
number: singular

word: man
part_of_speech: noun
root: man
number: singular

Dictionary

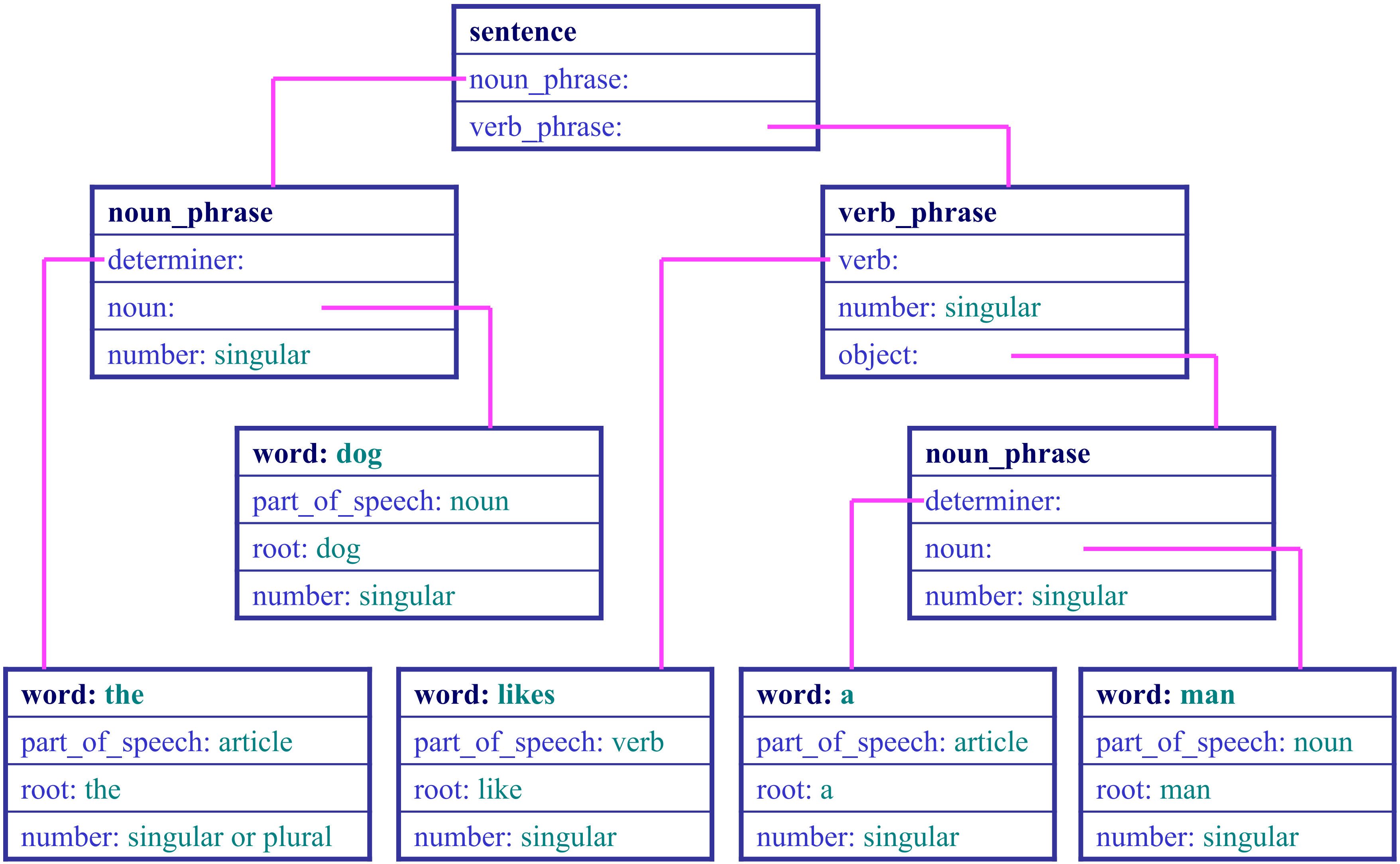
word: dog
part_of_speech: noun
root: dog
number: singular

word: men
part_of_speech: noun
root: man
number: plural

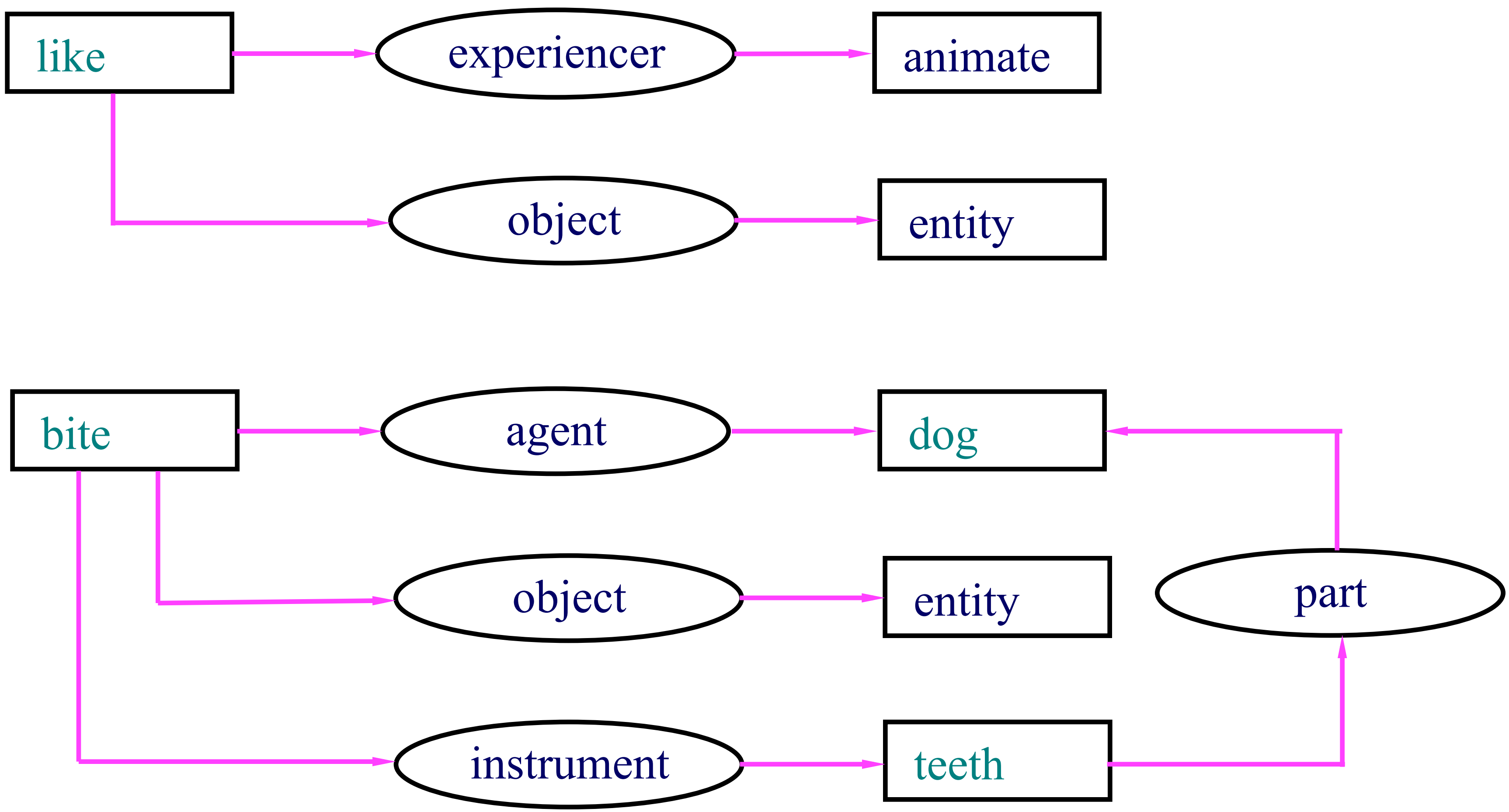
word: dogs
part_of_speech: noun
root: dog
number: plural

word: the
part_of_speech: article
root: the
number: singular or plural

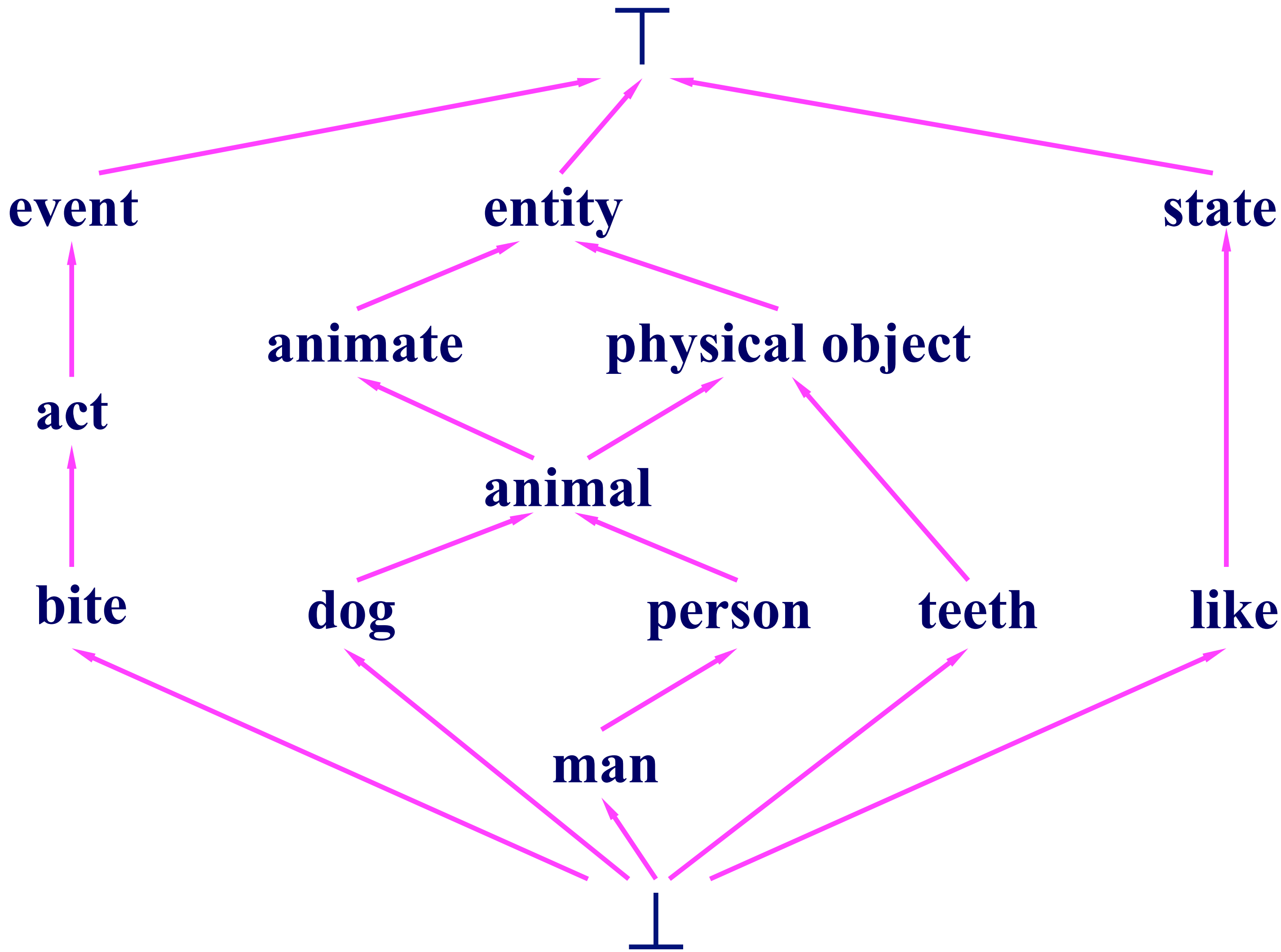
Parse tree


sentence
noun_phrase:
verb_phrase:
noun_phrase
determiner:
noun:
number: singular
verb_phrase
verb:
number: singular
object:
word: dog
part_of_speech: noun
root: dog
number: singular
noun_phrase
determiner:
noun:
number: singular
word: the
part_of_speech: article
root: the
number: singular or plural
word: likes
part_of_speech: verb
root: like
number: singular
word: a
part_of_speech: article
root: a
number: singular
word: man
part_of_speech: noun
root: man
number: singular

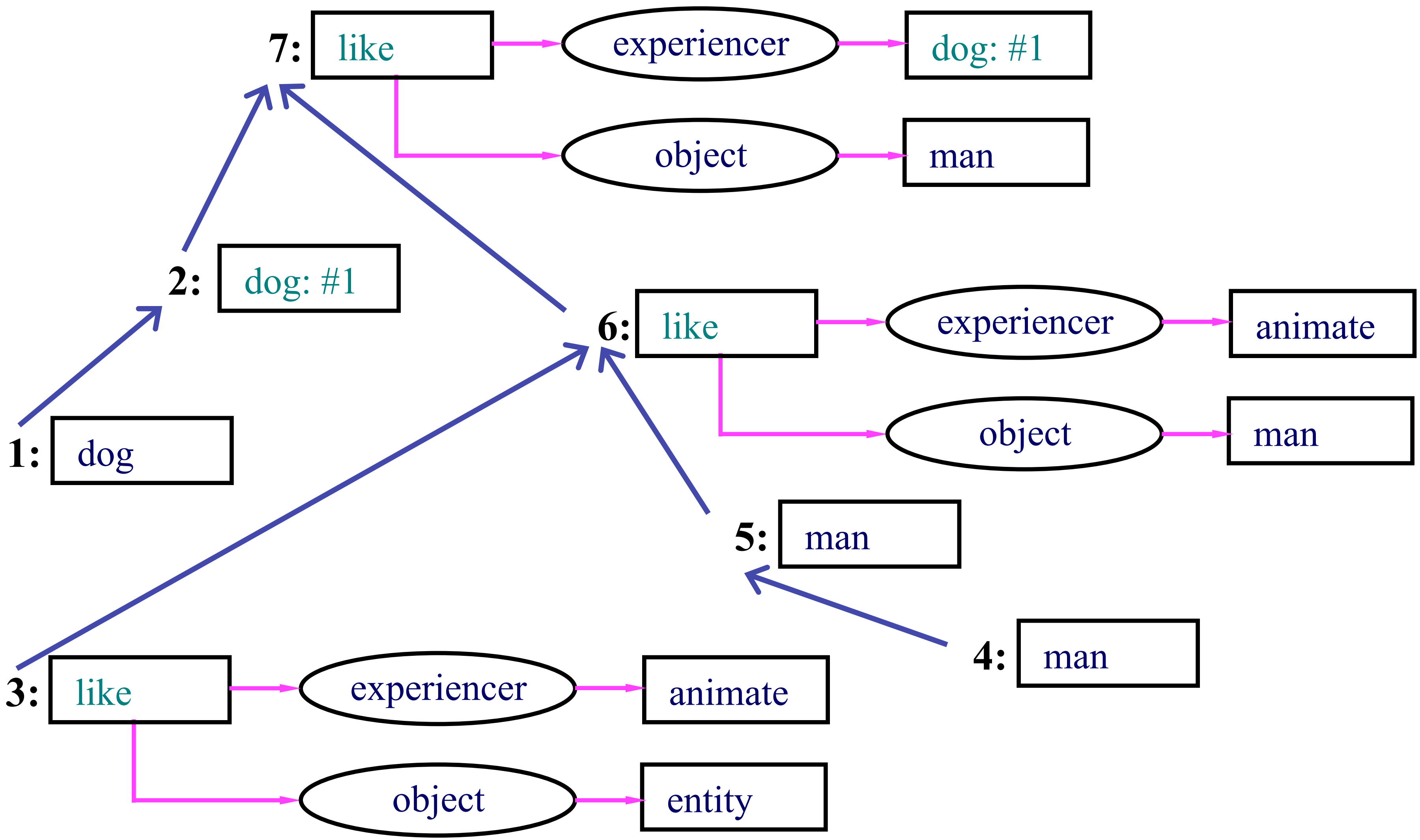
Case frames for the verbs



Type hierarchy



Construct semantic representation



Rationalist and empiricist

All grammars leak

(Sapir 1921)

You shall know a word by the company it keeps

(Firth 1957)

What kinds of things do people say

a. John I believe Sally said Bill believed Sue saw.

b. What did Sally whisper that she had secretly read?

c. Johan wants very much for himself to win.

d. (those are) the books you should read before it becomes difficult to talk about.

e. (those are) the books you should read before talking about becomes difficult.

f. Who did Jo think said John saw him?

g. That a serious discussion could arise here of this topic was quite unexpected.

h. The boys read Mary's stories about each other.

Stochastic tools

Art	is	a	lie	that	lets	us	see	the	truth
Noun	Verb	Article	Noun	Pronoun	Verb	Pronoun	Verb	Article	Noun

We have:

A set of words in our language $S_w = \{ w_1, \ldots, w_n \}$

A set of parts of speech or tags $S_t = \{ t_1, \ldots, t_m \}$

Formally, we want to pick to $t_1, \ldots, t_n$ **maximize:**

$P(T_1 = t_1, \ldots, T_n = t_n \mid W_1 = w_1, \ldots, W_n = w_n)$

Stochastic tools

$P(\ T_1 = t_1,\ \ldots,\ T_n = t_n \mid W_1 = w_1,\ \ldots,\ W_n = w_n\ )$ equal to

$P(\ t_1,\ \ldots,\ t_n \mid w_1,\ \ldots,\ w_n\ ) = P(\ t_1,\ \ldots,\ t_n,\ w_1,\ \ldots,\ w_n\ )\ /\ P(\ w_1,\ \ldots,\ w_n\ )$

Since we maximize this by choosing $t_1,\ \ldots,\ t_n$, we can simplify to:

$P(\ t_1,\ \ldots,\ t_n \mid w_1,\ \ldots,\ w_n\ ) =$

$P(\ t_1)P(\ w_1 \mid t_1\ )P(\ t_2 \mid t_1,\ w_1\ )\ \ldots\ P(\ t_n \mid w_1,\ \ldots,\ w_n,\ t_1,\ \ldots,\ t_{n-1}\ ) =$

$$\prod_{i=1}^{n} P(t_i \mid t_1,\ldots,t_{i-1},w_1,\ldots,w_{i-1})P(w_i \mid t_1,\ldots,t_{i-1},w_1,\ldots,w_{i-1})$$

Stochastic tools

$$\prod_{i=1}^{n} P(t_i \mid t_1,\ldots,t_{i-1},w_1,\ldots,w_{i-1}) P(w_i \mid t_1,\ldots,t_{i-1},w_1,\ldots,w_{i-1})$$

We need to make some useful approximations

$P(\, t_i \mid t_1, \ldots, t_{i-1}, w_1, \ldots, w_{i-1}\,)$ approaches $P(\, t_i \mid t_{i-1}\,)$

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

Plugging these approximations back into the above equation, we get

$$\prod_{i=1}^{n} P(t_i \mid t_{i-1}) P(w_i \mid t_i)$$

Viterbi algorithm

Vector space model

We have a vocabulary V, $w_i \in V$ and $|V| = n$.

A document d_j can be represented as a vector $(tf_{1,j}, tf_{1,j}, \ldots, tf_{n,j})$, where $tf_{i,j}$, called ***term frequency***, is number of occurrences of word w_i in document d_j.

The information that is captured by term frequency is how salient a word is within a given document. The higher the ***term frequency*** (the more often the word occurs) the more likely it is that the word is a good ***description*** of the content of the document.

Vector space model

Because more occurrences of a word indicate higher importance, but not as much relative importance as the undampened count would suggest, them frequency is usually dampened by a function as follows

$$weight(i,j) = \begin{cases} (1+\log(tf_{i,j}))\log\dfrac{N}{df_i} & \text{if } tf_{i,j} \geq 1 \\ 0 & \text{if } tf_{i,j} = 0 \end{cases}$$

where N is the total number of documents, and df_i is number of occurrences of w_i in d_j.

Vector similarity

To do retrieval in the vector space model, documents are ranked according to similarity with the query as measured by ***the cosine measure*** or ***normalized correlation coefficient***.

$$\cos(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}}$$

Research area

- Speech recognition
- Natural language generation
- Speech synthesis
- Information retrieval
- Information extraction
- Spelling correction
- Grammar checking
- Machine translation

Any question?

Xiaoqing Zheng
Fundan University