

《语义计算与知识检索》研究生课程

句子与篇章级语义计算

万小军

北京大学语言计算与互联网挖掘组

2018年3月28日

<http://www.icst.pku.edu.cn/lcwm/course/sckr2018>

内容

- 概述
- 语义角色标注
- 篇章分析

概述

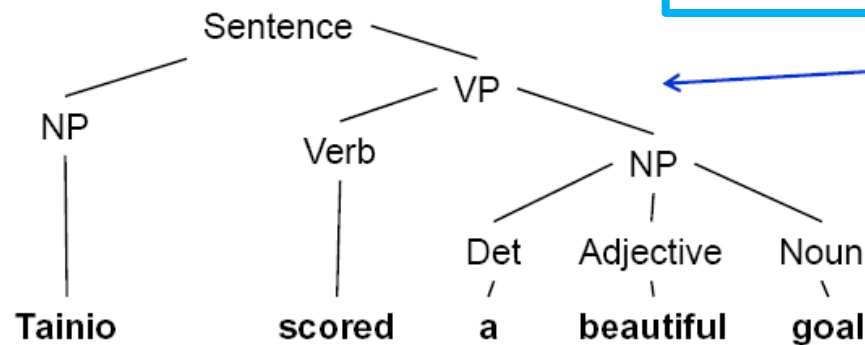


Language
generation



Semantic analysis
(or, understanding)

Tainio scored a beautiful goal!



Syntactic parsing

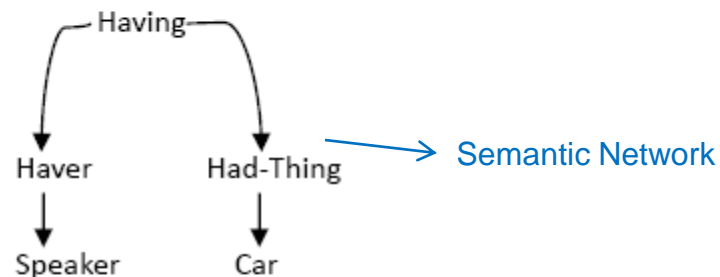
(NP = noun phrase,
VP = verb phrase)

Morphology: score-d

语义表示

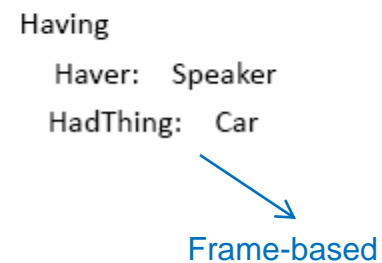
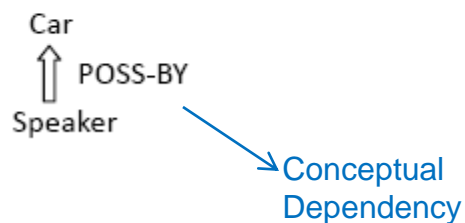
- “I have a car”

$\exists e, y \text{ Having}(e) \wedge \text{Haver}(e, \text{Speaker}) \wedge \text{HadThing}(e, y) \wedge \text{Car}(y)$ → First-Order Logic



$\mathbf{a} = (a_1, a_2, a_3, \dots, a_{n-1}, a_n).$

↓
Semantic Vector



如何计算句子级语义?

- 基于组合语义分析

- 一个句子的意义由其组成成分(例如词语)的意义组合而得到

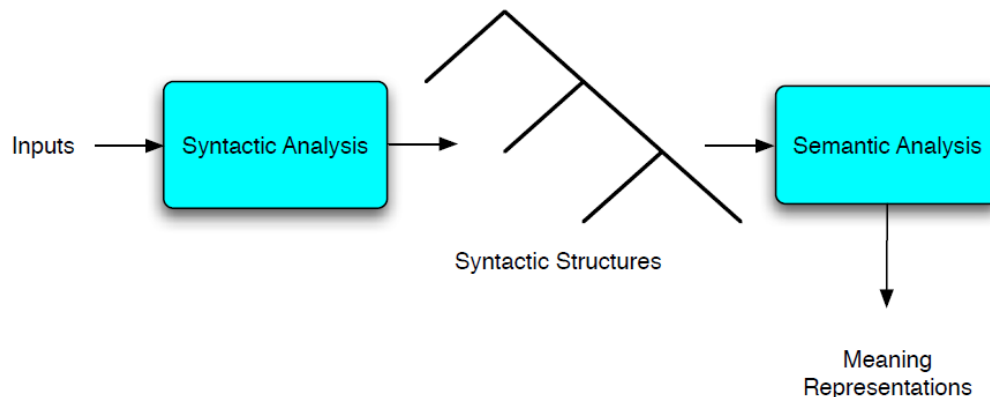
- 句子的句法结构对意义组合有影响

- Tom killed Jack.

- Jack killed Tom.

- 句法驱动的组合语义分析

- 基于词汇和语法信息获取句子意义表达



语义逻辑表示

- 用一阶公式表达句子语义

	First-order Predicate Logic
John loves Mary	$\text{loves}(\text{john}, \text{mary})$
Somebody sleeps	$\exists x \text{ sleeps}(x)$
Somebody loves everybody	1. $\exists x \forall y \text{ loves}(x, y)$ 2. $\forall y \exists x \text{ loves}(x, y)$

Other candidates: Modal logics,
higher-order predicate logics, description logics etc.

一阶谓词逻辑(FOPL)语言

- **基本元素**
 - **常量(Constants)**
 - Refer to specific object in the world: Harry
 - **谓词(Predicates)**
 - Refer to the relations among objects: Serves(Maharani,, VegetarianFood), Restaurant(Maharani)
 - **变量(Variables)**
 - Refer to objects: x, y
 - **逻辑算子、连接词(Logic connectives)**
 - Operators for composing larger representations: \neg , \vee , \wedge , \rightarrow

FOPL语言

- **全称量词(Universals quantifiers)**

- $\forall x (\text{man}(x))$
 - Everybody is a man
- $\forall x (\text{rich}(x) \vee \text{popular}(x))$
 - Everybody is rich or popular
- $\forall x (\text{rich}(x) \wedge \text{popular}(x))$
 - Everybody is rich and popular
- $\forall x (\text{rich}(x) \rightarrow \text{popular}(x))$
 - Everybody who is rich is popular

FOPL语言

- **存在量词(Existential quantifiers)**

- $\exists x (\text{man}(x))$
 - Somebody is a man
- $\exists x (\text{rich}(x) \vee \text{popular}(x))$
 - ...
- $\exists x (\text{rich}(x) \wedge \text{popular}(x))$
 - ...
- $\exists x (\text{rich}(x) \rightarrow \text{popular}(x))$

FOPL语言: 推理规则

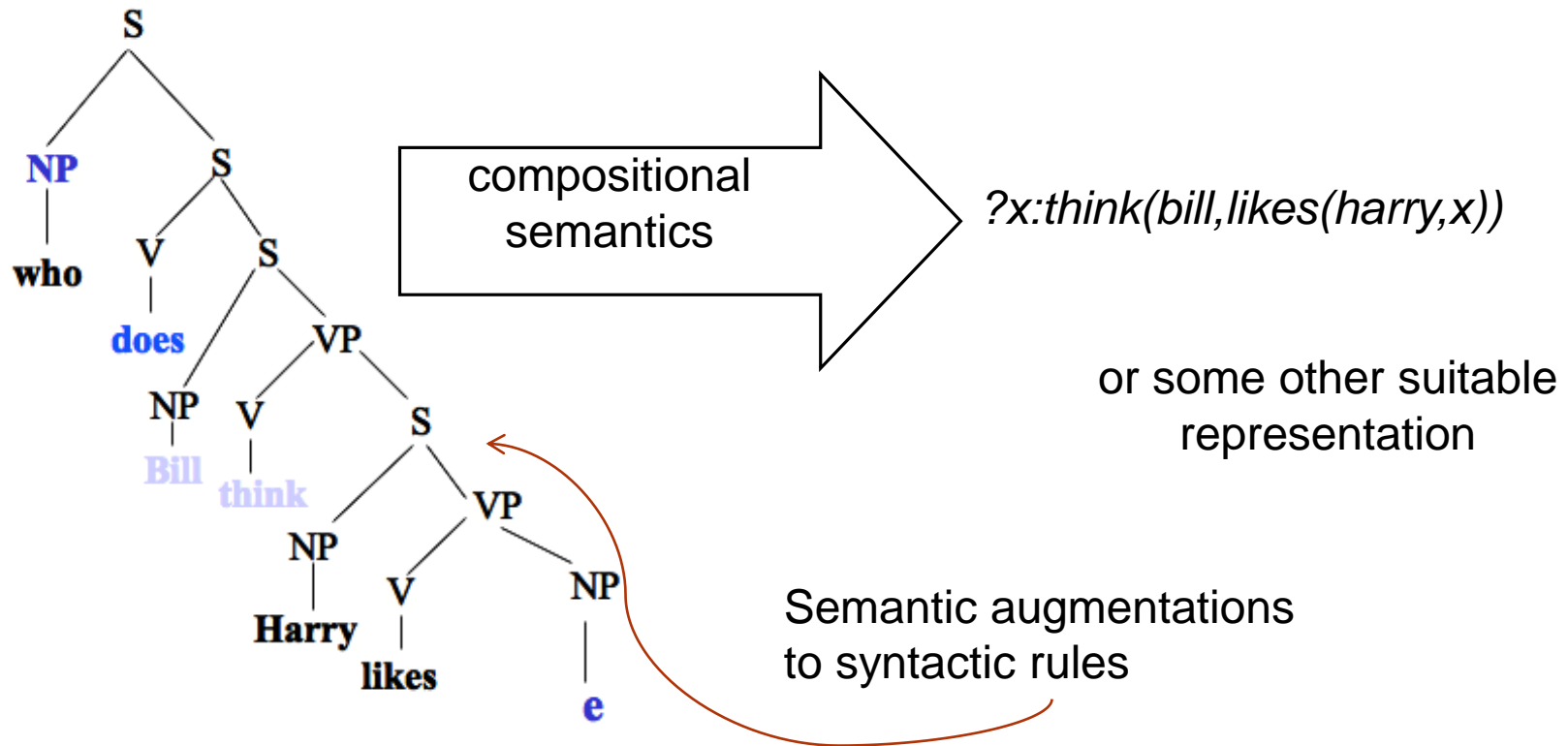
$$\frac{\begin{array}{l} \forall x (\text{man}(x) \rightarrow \text{mortal}(x)) \\ \text{man}(\text{socrates}) \end{array}}{\therefore \text{mortal}(\text{socrates})}$$

$$\frac{\exists x \forall y \text{ loves}(x,y)}{\therefore \forall y \exists x \text{ loves}(x,y)}$$

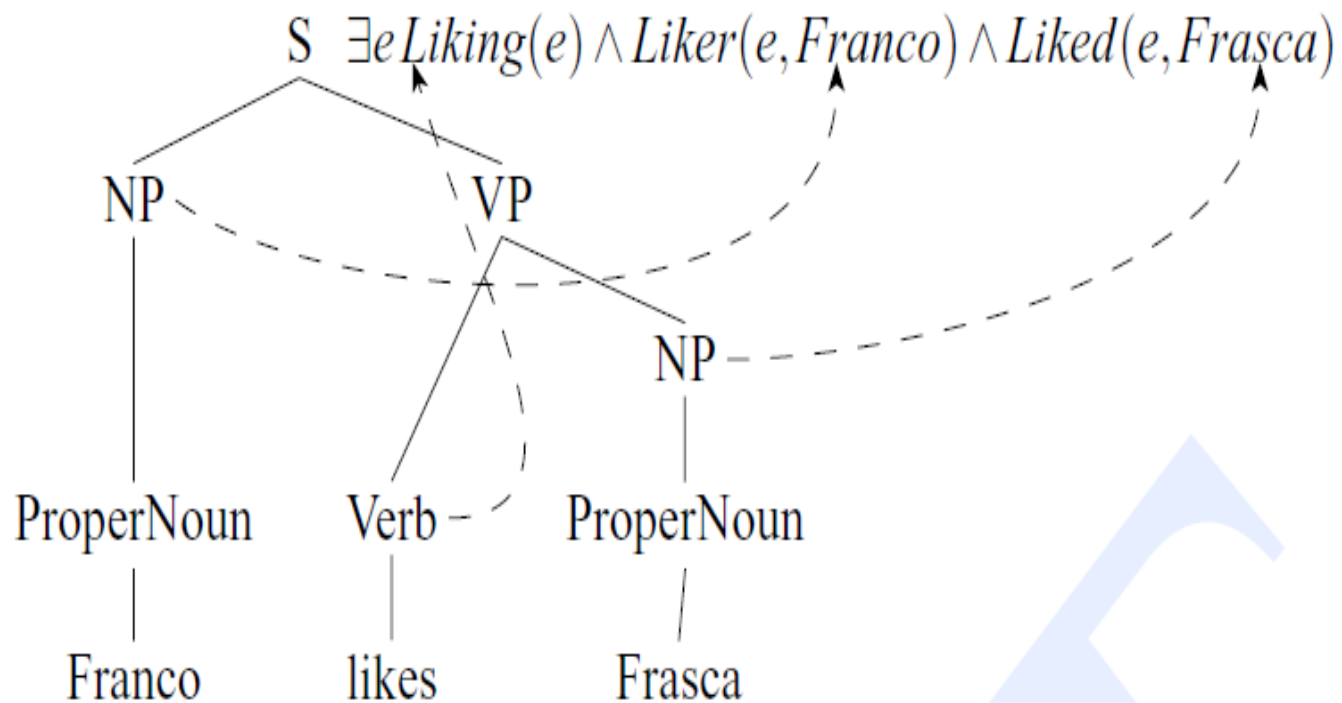
Lambda演算

- $\lambda x (\text{man}(x))$
 - The function that maps men to “true” and non-men to “false”
- $\lambda x (\text{man}(x)) (\text{john})$
 - The function that maps men to “true” and non-men to “false” applied to john
 - $= \text{man}(\text{john})$
 - *function application or beta reduction or lambda conversion*

从句法树到逻辑表达式



从句法树到逻辑表达式



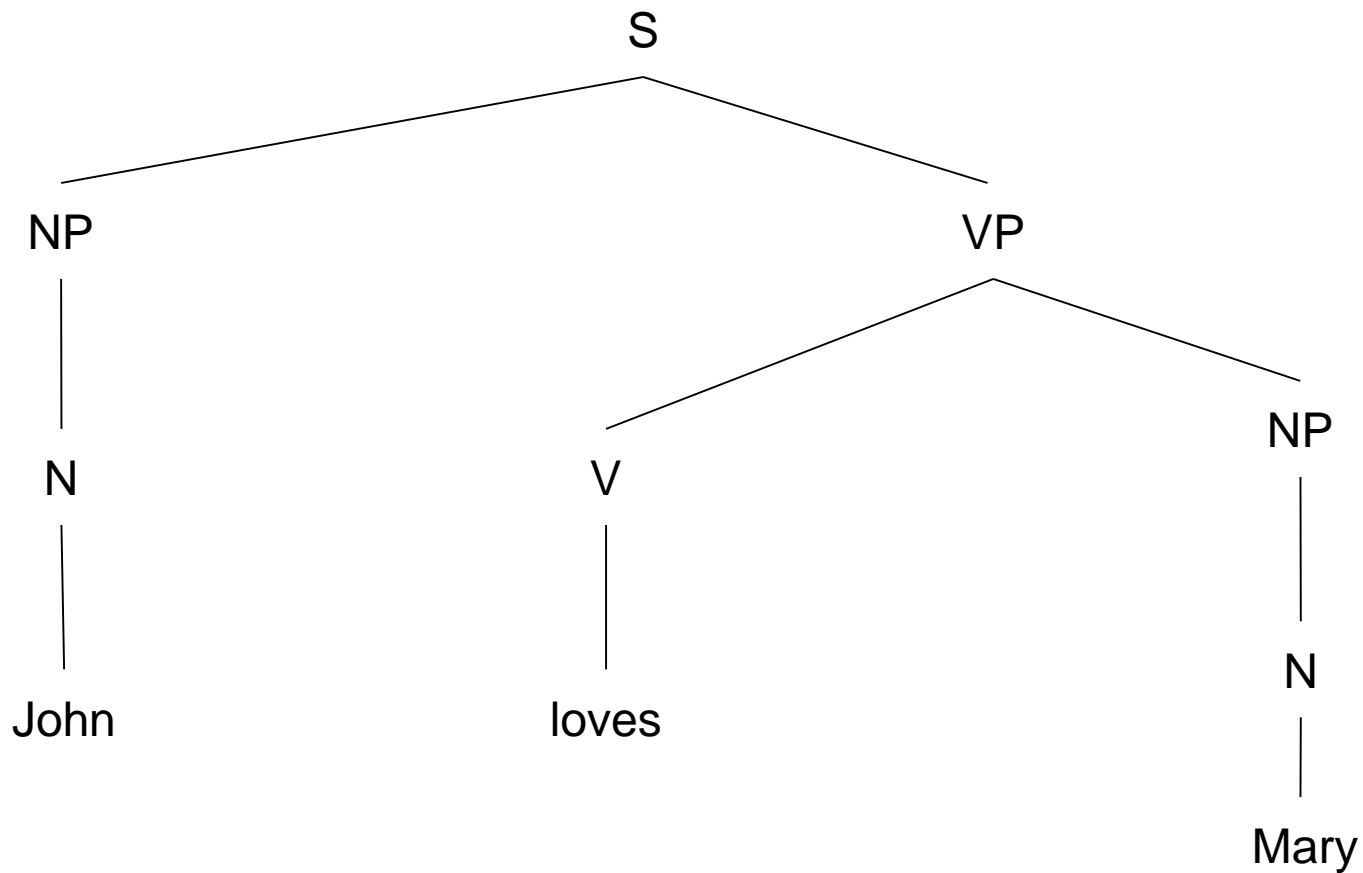
语义解释算法 (简化版)

- 假设所有节点最多有两个子节点 (二分叉)
- 三条规则
 1. 每个词语映射到一个表达式/片段
 - 可从词典中直接查找
 - 需要采用WSD对有歧义的词语进行消歧
 2. 不分叉的节点从其子节点继承意义
 3. 分叉节点应用函数(Functional Application)
 - 如果一个子节点表示一个函数, 另一个子节点表示一个参数, 那么将参数代入函数
- 自底向上自叶节点到根节点应用上述规则获取意义

例子

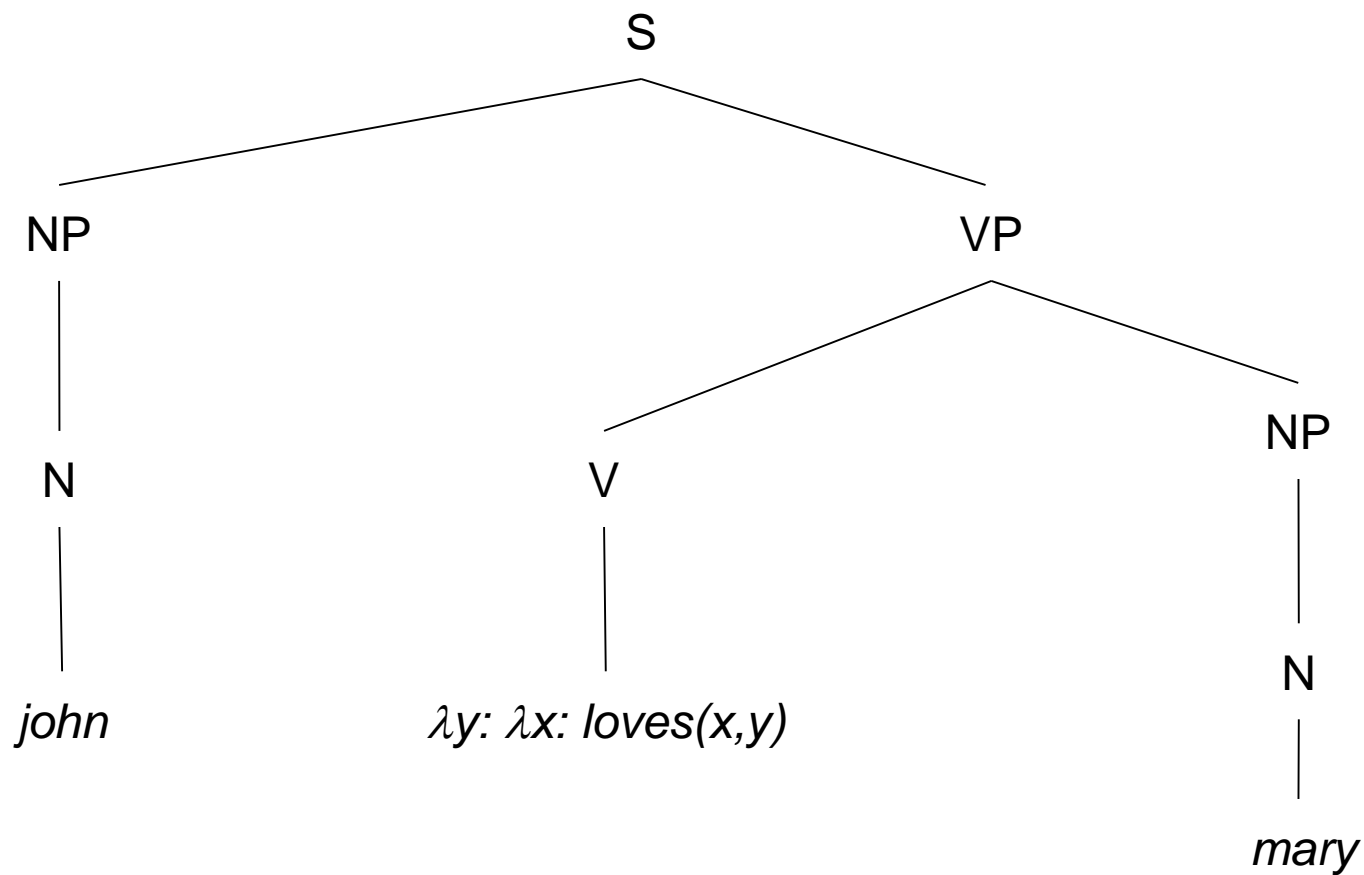
- Proper nouns/ 专有名词 map to a constant
 - $[[\text{John}]]$ = *john*
 - $[[\text{America}]]$ = *america*
- Adjectives, nouns, and intransitive verbs maps to a one-place predicate
 - A predicate is a function that returns a boolean value (true, false)
 - $[[\text{sleeps}]]$ = $\lambda x: \text{sleeps}(x)$
 - $[[\text{man}]]$ = $\lambda x: \text{man}(x)$
 - $[[\text{red}]]$ = $\lambda x: \text{red}(x)$
- Transitive verbs map to a two-place predicate
 - $[[\text{loves}]]$ = $\lambda y: \lambda x: \text{loves}(x,y)$
- Auxiliaries map to the identity function
 - $[[\text{is}]]$ = $[[\text{does}]]$ = $\lambda x: x$

例子



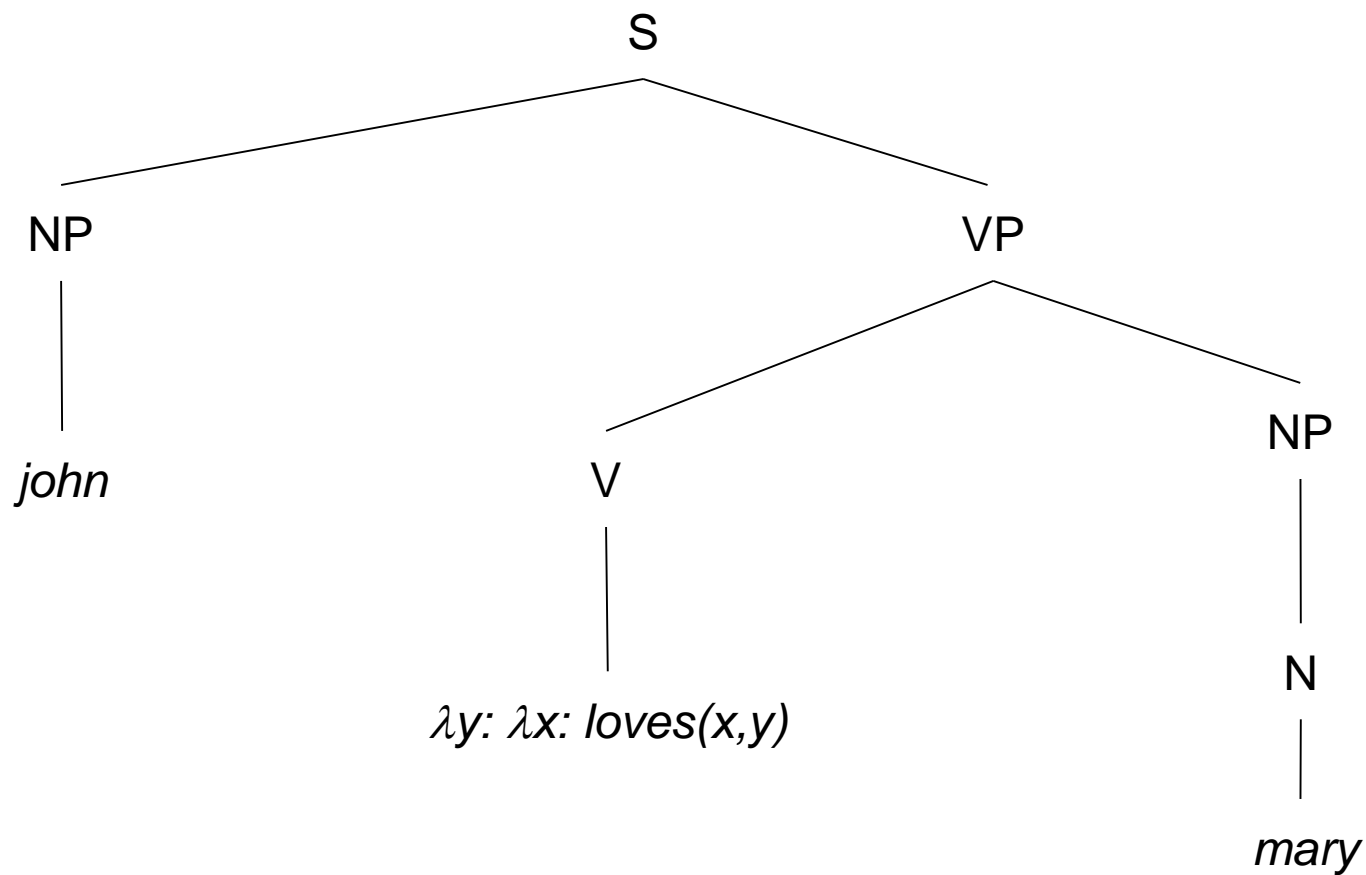
“John loves Mary”

例子



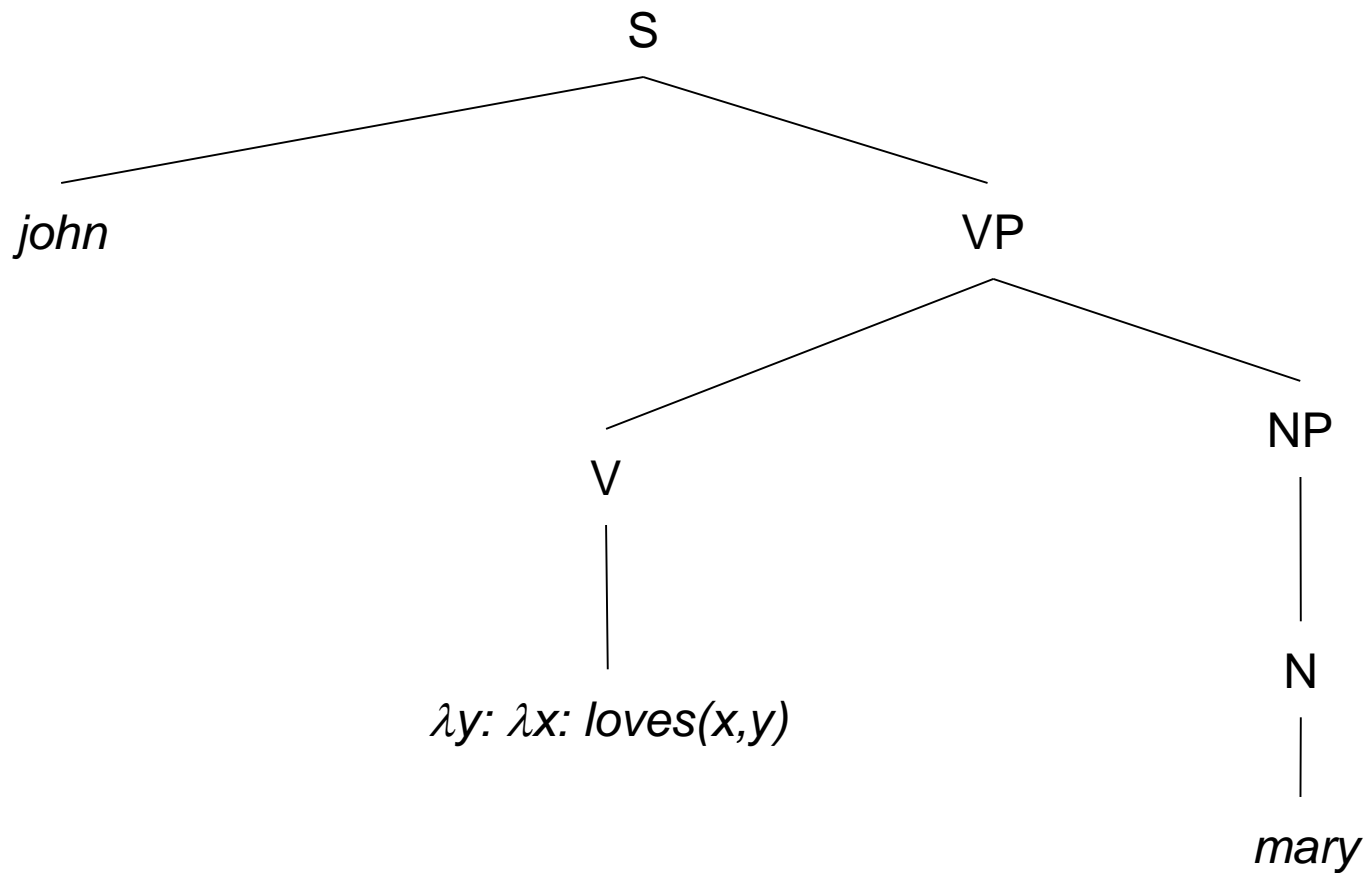
“John loves Mary”

例子



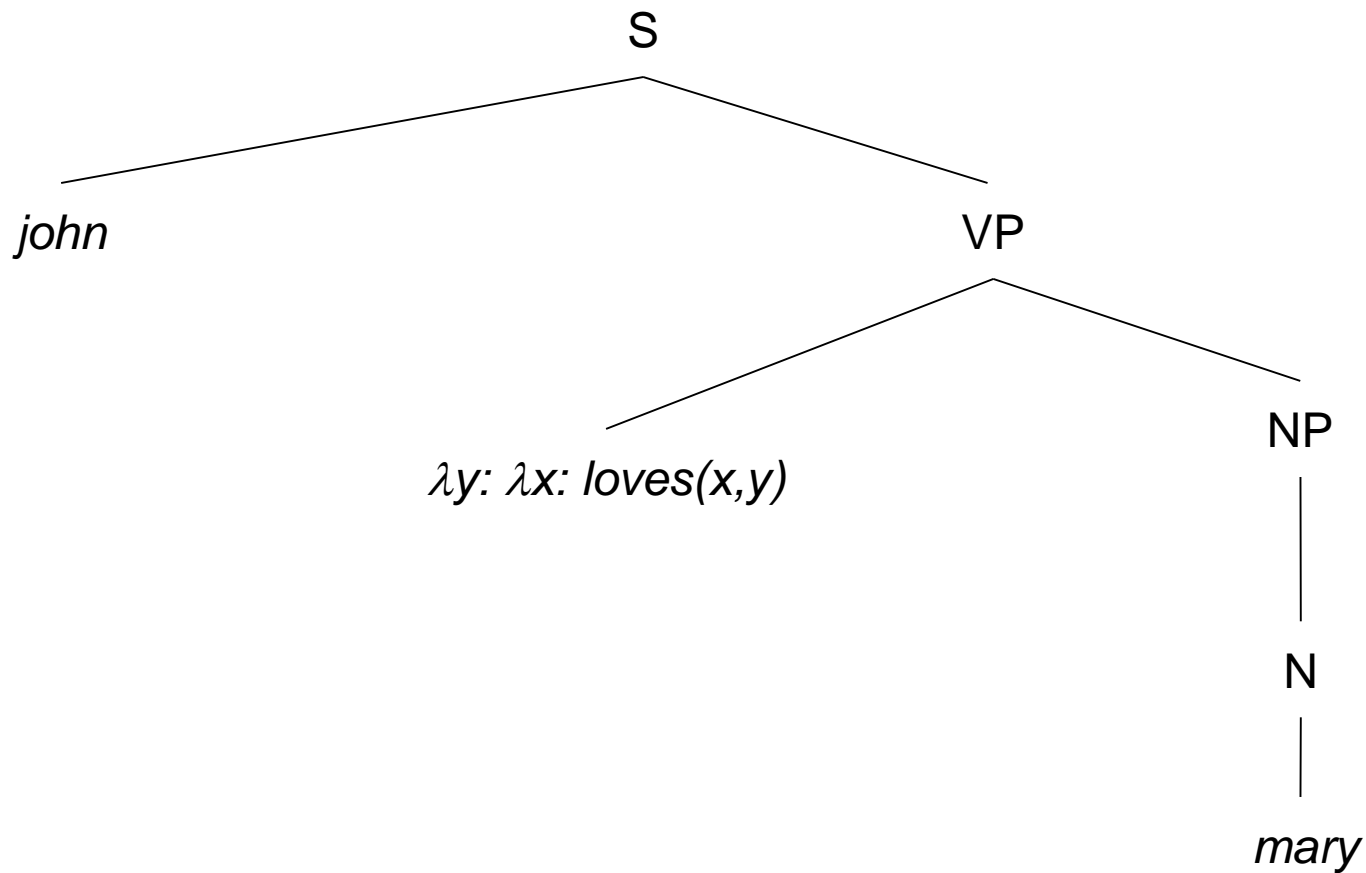
“John loves Mary”

例子



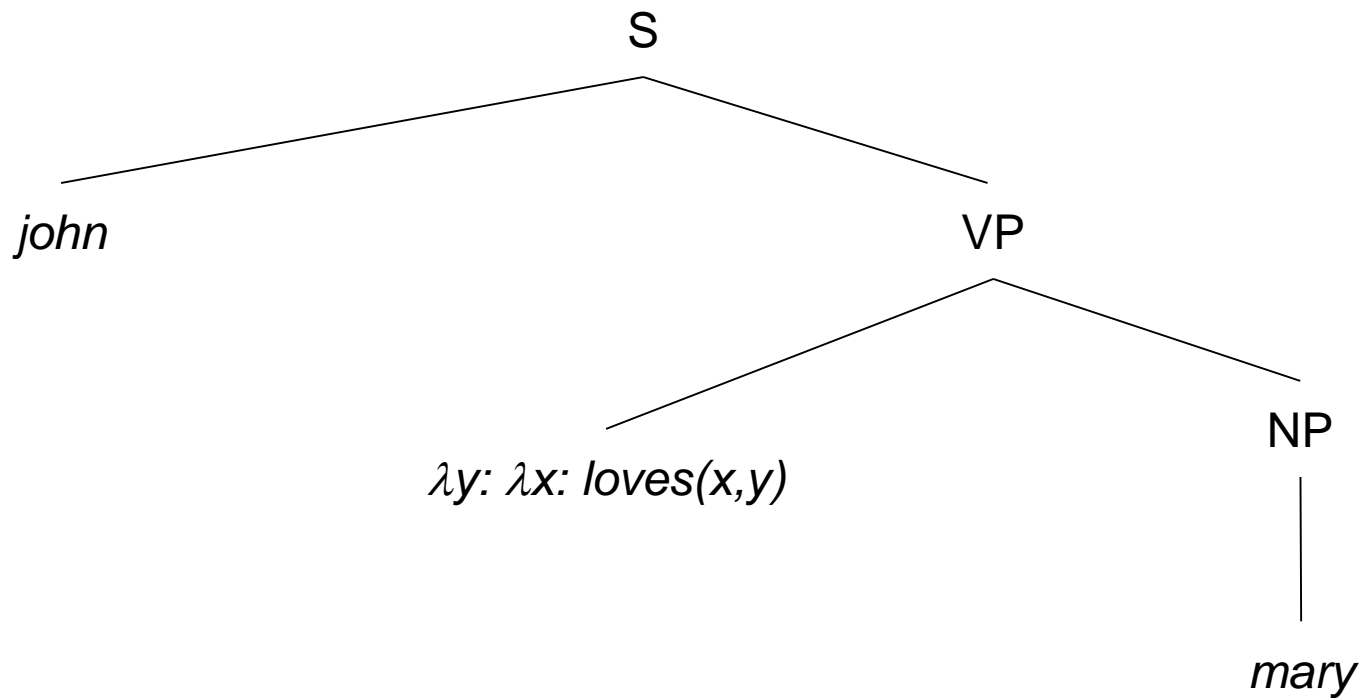
"John loves Mary"

例子



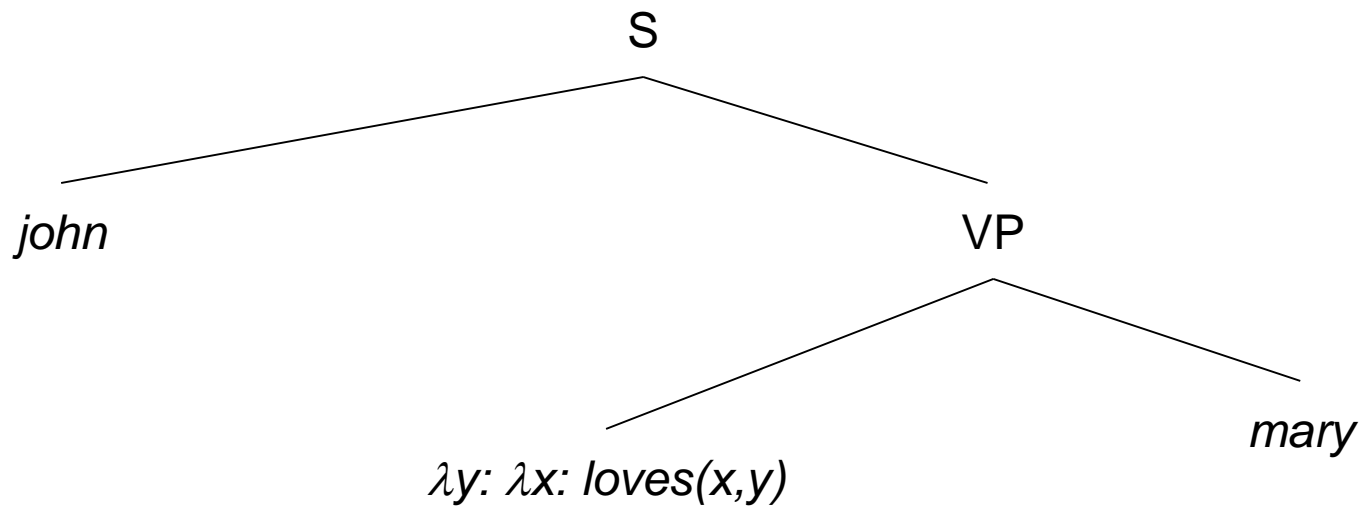
“John loves Mary”

例子



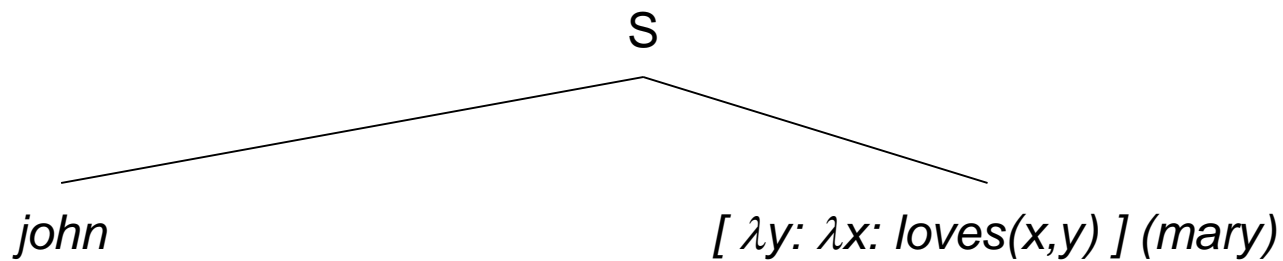
“John loves Mary”

例子



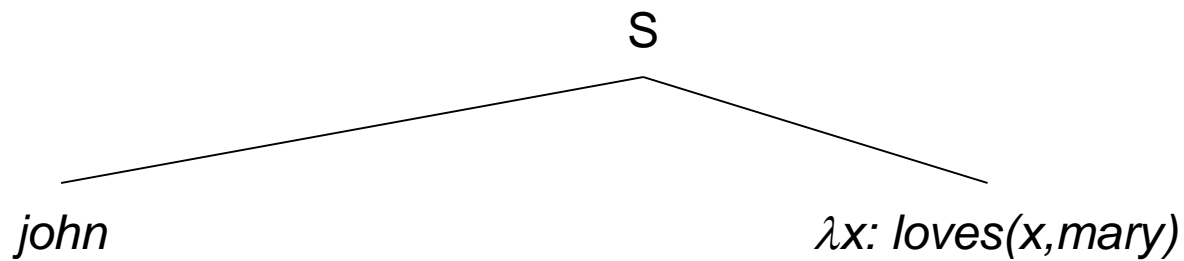
“John loves Mary”

例子



“John loves Mary”

例子



“John loves Mary”

例子

$[\lambda x: \text{loves}(x, \text{mary})] (\text{john})$

“John loves Mary”

例子

loves(john,mary)

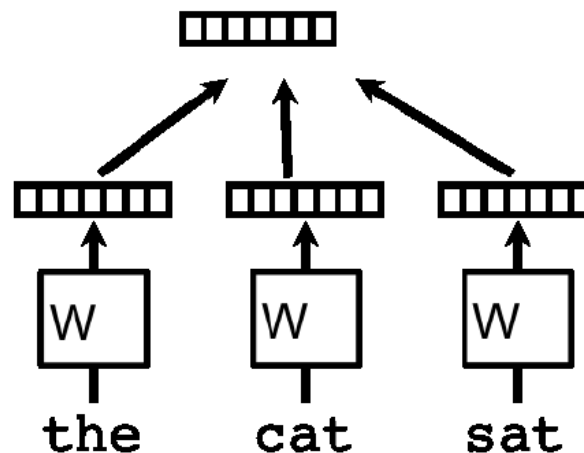
“John loves Mary”

句子语义向量计算

- 非组合语义方法
 - 隐含语义分析
 - 主题模型
- 基于组合语义
 - 不考虑句法结构
 - 词汇语义向量的平均或衔接：

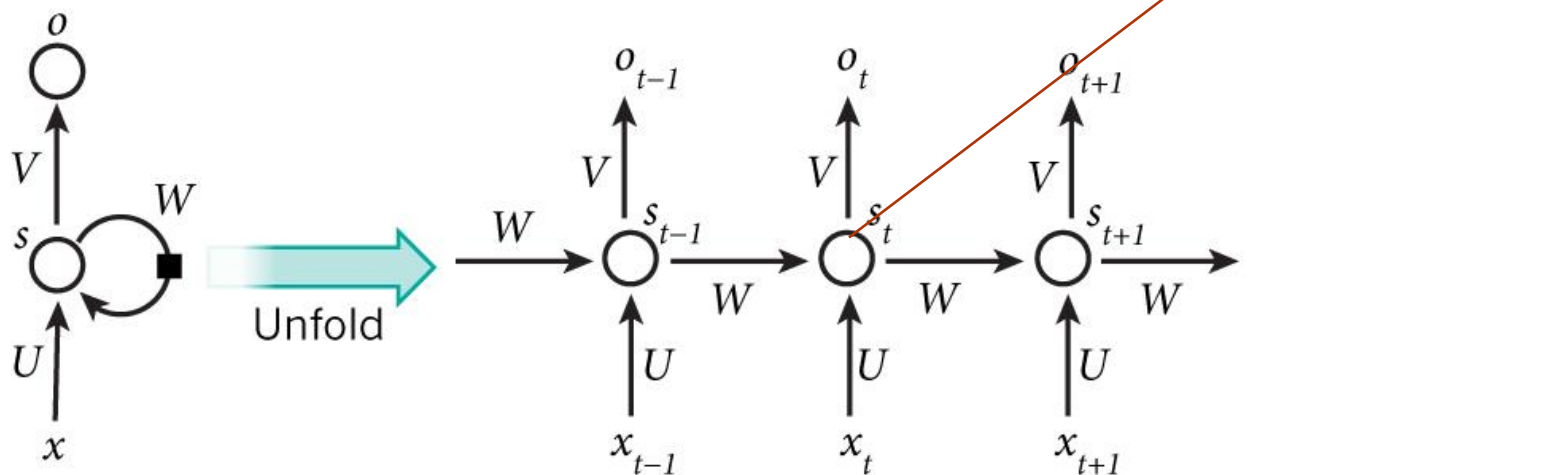
Average/Concatenate

Word Matrix



句子语义向量计算

- 基于组合语义
 - 不考虑句法结构
 - Recurrent Neural Network



$x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots$ 句子中的词序列，每个词用向量表示；

s_t : 时间 t 的隐状态值，可看作 x_1, \dots, x_{t-1}, x_t 的语义表示， $s_t = f(Ux_t + Ws_{t-1})$

f 通常是 \tanh or ReLU

o_t : 时间 t 的输出，跟分类/预测任务有关 $o_t = \text{softmax}(Vs_t)$

句子语义向量计算

- 基于组合语义
 - 不考虑句法结构
 - Convolutional Neural Network

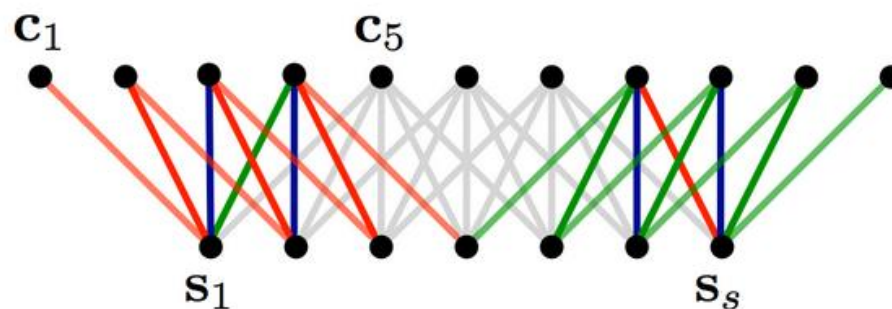
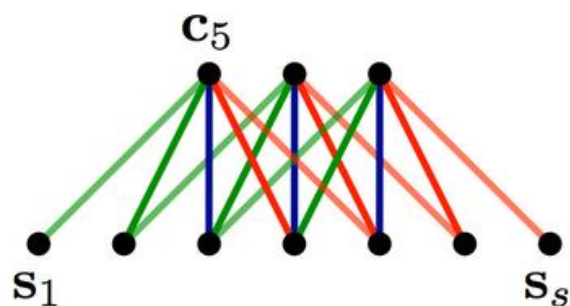
$$c_j = m^T s_{j-m+1:j}$$

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

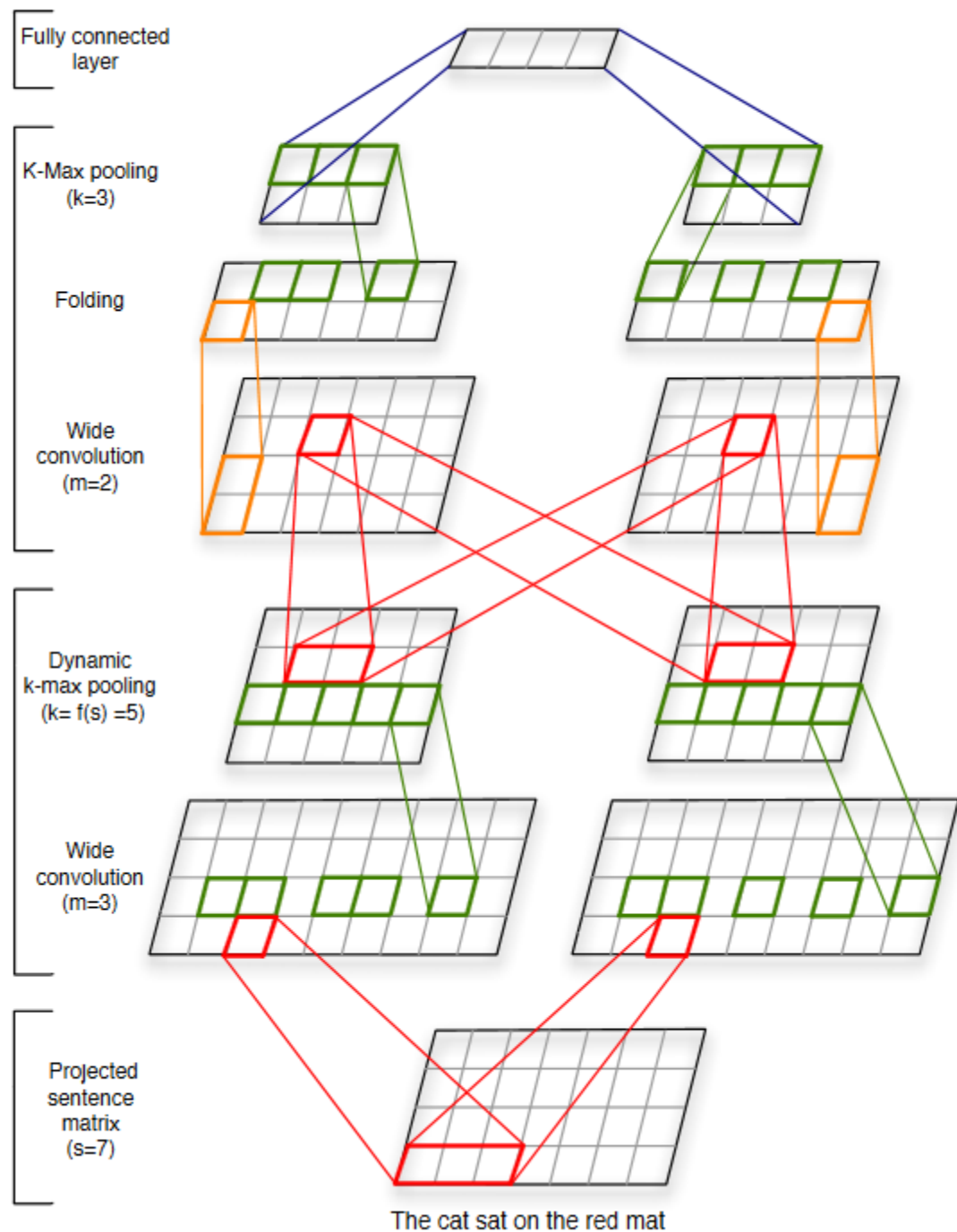


Narrow vs. Wide Convolution. Filter size 5, input size 7. Source: A Convolutional Neural Network for Modelling Sentences (2014)

句子语义向量

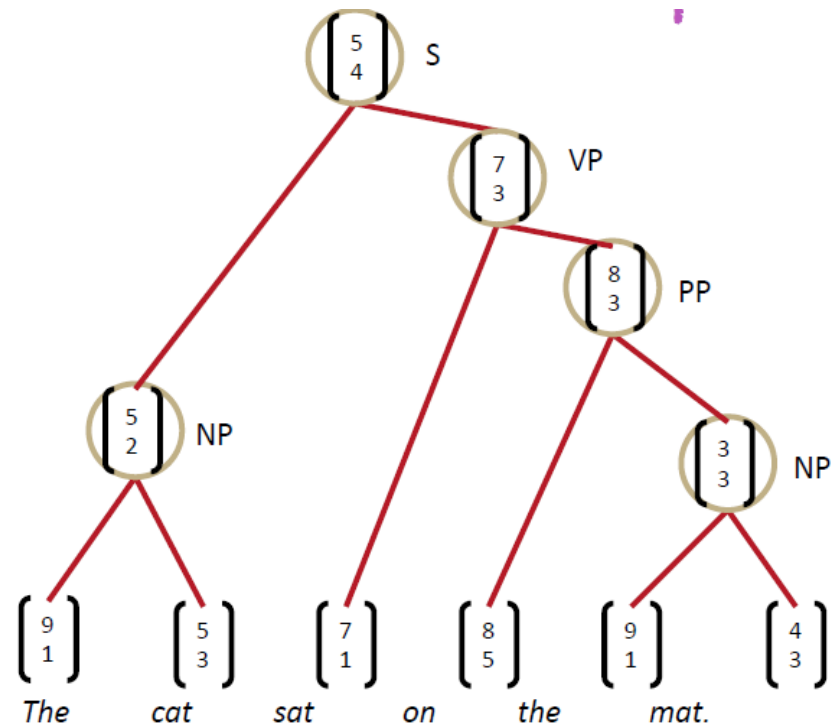
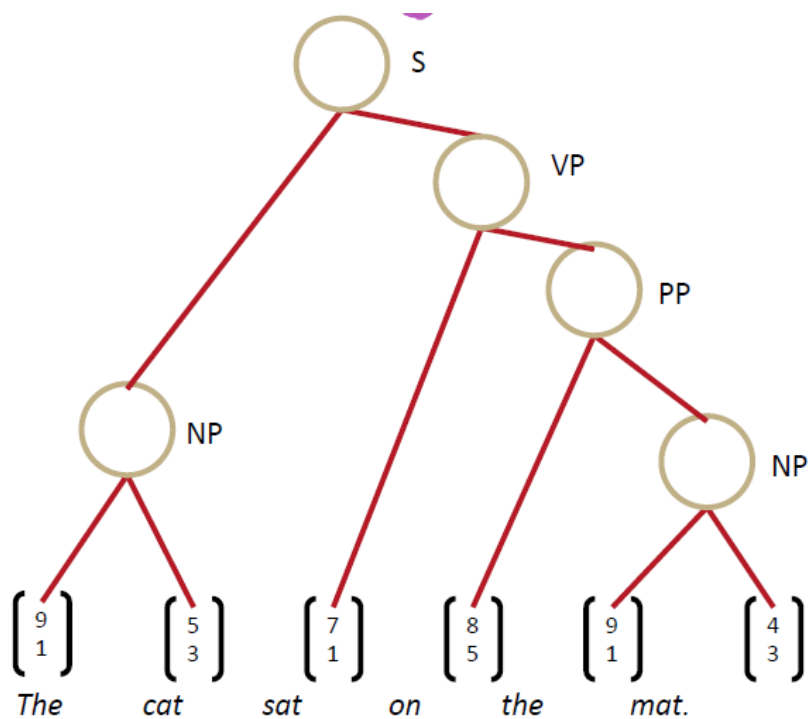
- 基于组合语义
 - 不考虑句法结构
 - Convolutional Neural Network

多层的卷积与池化操作



句子语义向量计算

- 基于组合语义
 - 考虑句法结构



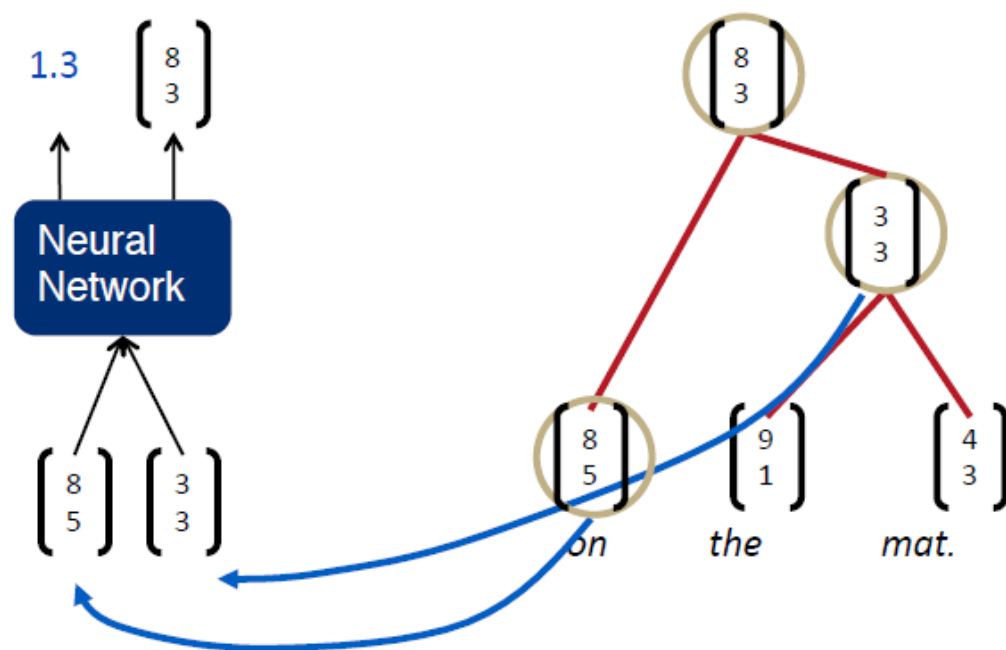
句子语义向量计算

- 基于组合语义
 - 考虑句法结构
 - Recursive Neural Network (RNN)

Input: 两个子节点的语义表达

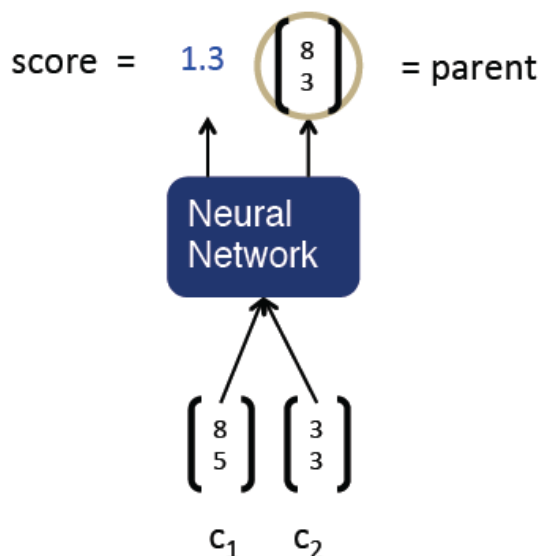
Outputs:

1. 两个子节点合并后父节点的语义表达
2. 父节点的权值(跟任务有关)



句子语义向量计算

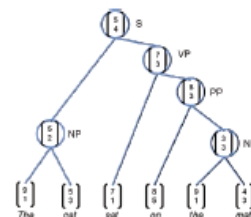
- 基于组合语义
 - 考虑句法结构
 - Recursive Neural Network (RNN)



$$\text{score} = U^T p$$

$$p = \tanh \left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b \right)$$

Same W parameters at all nodes of the tree

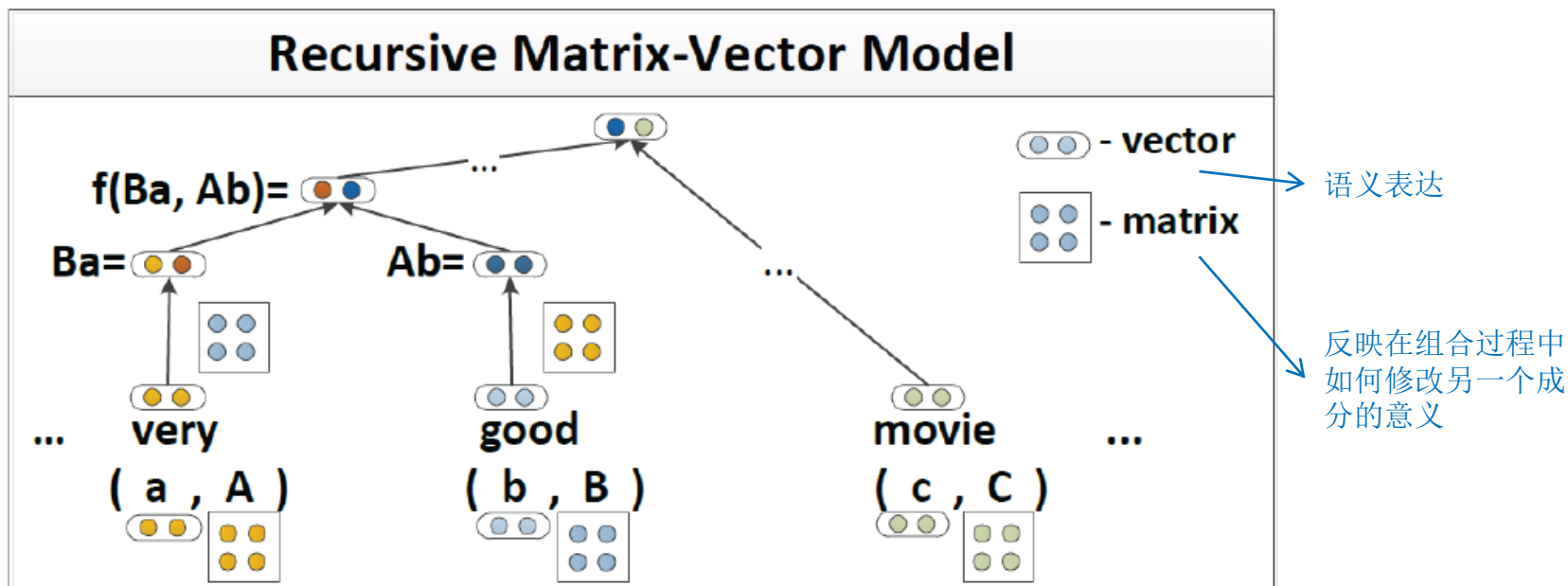


句子语义向量计算

- 基于组合语义
 - 考虑句法结构
 - 进一步: Matrix-Vector Recursive Neural Network

$$p = \tanh \left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b \right)$$

$$p = \tanh \left(W \begin{bmatrix} C_2 c_1 \\ C_1 c_2 \end{bmatrix} + b \right)$$



句子语义向量计算

- 更多细节可参考如下博士论文

[Recursive Deep Learning for Natural Language Processing and Computer Vision](#), Richard Socher

PhD Thesis, Computer Science Department, Stanford University, 2014.

如何计算篇章级语义？

- 可以采用与组合语义类似的思想
 - 可考虑句子语义与篇章结构
- 但已有相关研究极少，缺乏相关理论及实验研究
- 实际上一般简单采用词袋模型表达篇章语义
- 篇章层面的研究大多集中在篇章结构分析上：如篇章分割、指代消解

语义角色标注

句法变形

Yesterday, Kristina hit Scott with a baseball

Scott was hit by Kristina yesterday with a baseball

Yesterday, Scott was hit with a baseball by Kristina

With a baseball, Kristina hit Scott yesterday

Yesterday Scott was hit by Kristina with a baseball

Kristina hit Scott with a baseball yesterday

Agent (hitter)	Patient hit	Instrument of hitting	Temporal adjunct
----------------	-------------	-----------------------	------------------

语义角色标注(SRL)

- 一种浅层语义分析技术
- 确定作为动词或谓语变元的名词短语所扮演的语义角色

agent patient source destination instrument

施事 受事 来源 目的 工具

- John **drove** Mary from Austin to Dallas in his Toyota Prius.
- The hammer **broke** the window.

语义角色

- **多种语义角色被提出，最常用的如下：**
 - Agent/施事: Actor of an action
 - Patient/受事: Entity affected by the action
 - Instrument/工具: Tool used in performing action.
 - Beneficiary/受益者: Entity for whom action is performed
 - Source/来源: Origin of the affected entity
 - Destination/目标: Destination of the affected entity
 - ...

语义角色的应用

- 对很多任务都有用
- 问答系统
 - “Who” questions usually use Agents
 - “What” question usually use Patients
 - “How” and “with what” questions usually use Instruments
 - “Where” questions frequently use Sources and Destinations.
 - “For whom” questions usually use Beneficiaries
 - “To whom” questions usually use Destinations
- 信息抽取
- 知识获取
- 文档摘要
- ...

语义角色的应用

- 信息抽取
 - 从文本中抽取特定类型的信息

London gold fell \$4.70 to \$308.45.



Slot	Filler
Product	London gold
Price change	-\$4.70
Current price	\$308.45

语义角色的应用

- 信息抽取
 - 识别语义角色，然后通过规则基于语义角色结果获得特定类型信息

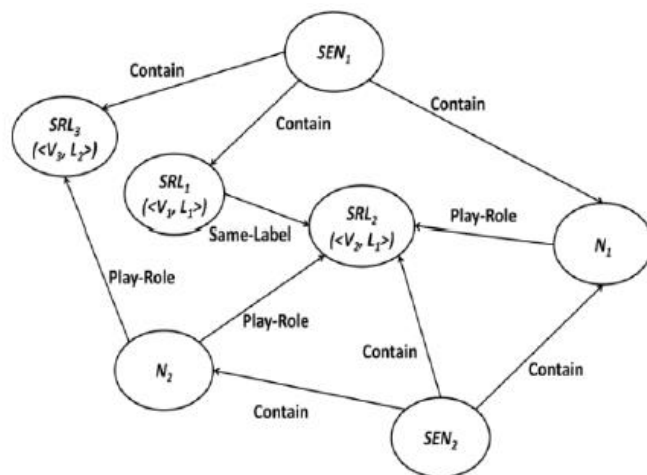
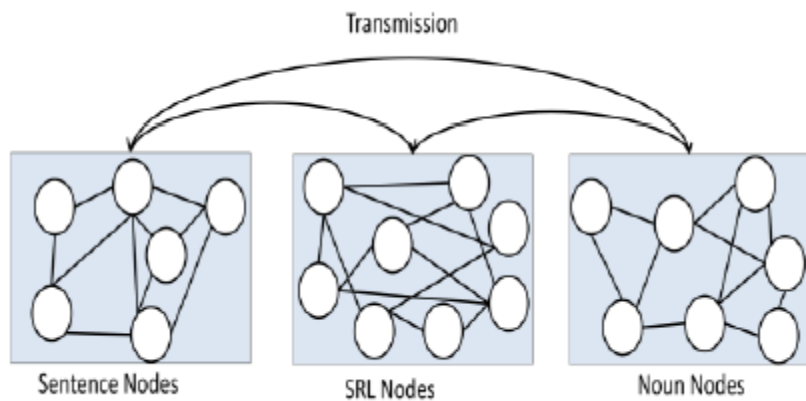
AGENT and MARKET_CHANGE_VERB => PRODUCT

PATIENT and (MONEY or PERCENT or QUANTITY) and MARKET_CHANGE_VERB =>
PRICE_CHANGE

ARG4 and NUMBER and MARKET_CHANGE_VERB=> CURRENT_PRICE

语义角色的应用

- 文档摘要
 - 从文档集中抽取若干重要句子形成摘要
 - 语义角色信息可以帮助进行句子重要性评价



SRL与句法线索

- 常見情況下，语义角色可以根据特定的句法位置所确定
 - Agent: subject
 - Patient: direct object
 - Instrument: object of “with” PP
 - Beneficiary: object of “for” PP
 - Source: object of “from” PP
 - Destination: object of “to” PP
- 然而，不一定如此
 - The boy broke the window.
 - The hammer broke the window.
 - John gave Mary the book.
 - The book was given to Mary by John.
 - John went to the movie with his car.
 - John went to the movie with his wife.
 - John bought the car for Mary.
 - John bought the car for \$21K.

SRL并不容易!

选择限制(Selectional Restrictions)

- **特定动词对于其语义角色的约束**
 - Agents should be animate (有生命的)
 - Beneficiaries should be animate
 - Instruments should be tools
 - Patients of “eat” should be edible (可食用的)
 - Sources and Destinations of “go” should be places.
 - Sources and Destinations of “give” should be animate.
- **可利用本体层次结构信息(如WordNet)来确定上述约束是否满足**
 - “John” is a “Human” which is a “Mammal” which is a “Vertebrate” which is an “Animate”

选择限制的应用

- 帮助保留或排除特定的语义角色结果
 - “John bought the car for \$21K”
 - Beneficiaries should be Animate (不满足)
 - Instrument of a “buy” should be Money (满足)
 - “John went to the movie with Mary”
 - Instrument should be Inanimate (不满足)
 - “John drove the van to work with Mary.”
 - Instrument of a “drive” should be a Vehicle (工具是the van, 而非Mary)

SRL经验性方法

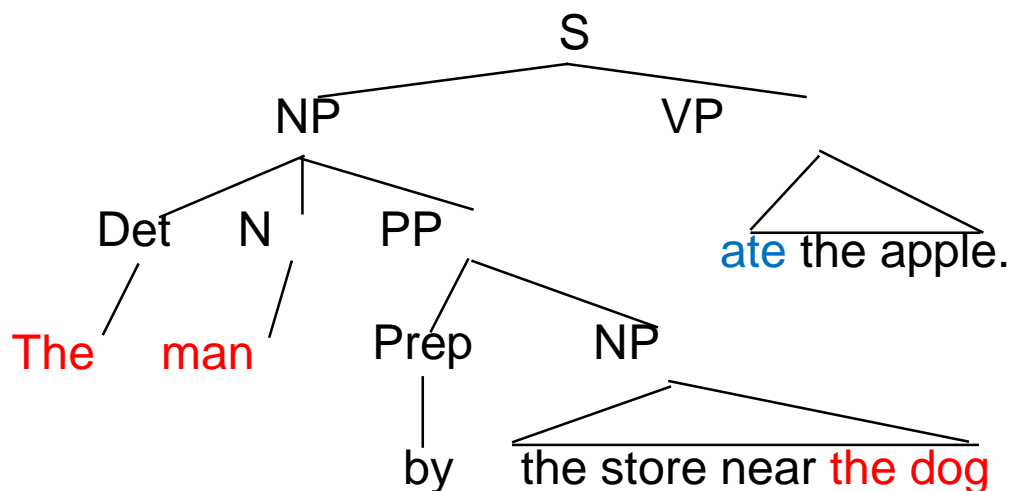
- 难以获取所有的选择限制与本体知识
- 经验性方法/统计方法能够自动获取与使用知识，进行更有效的SRL

基于序列标注的SRL

- **SRL可看做序列标注问题**
- **对于每个动词，标注该动词的可能的语义角色.**
- **可采用标准的序列标注模型**
 - Token classification
 - HMMs
 - CRFs

基于句法树的SRL

- 通过利用句法规则帮助识别语义角色
 - E.g. “the agent is usually the subject of the verb”.
- 需要句法树识别准确的主语



“The man by the store near the dog ate an apple.”

“The man” is the agent of “ate” not “the dog”.

基于句法树的SRL

- 假设句法树已知
- 对于每个谓语，将句法树中每个节点标注为非语义角色或某种可能的语义角色

Color Code:

not-a-role

agent

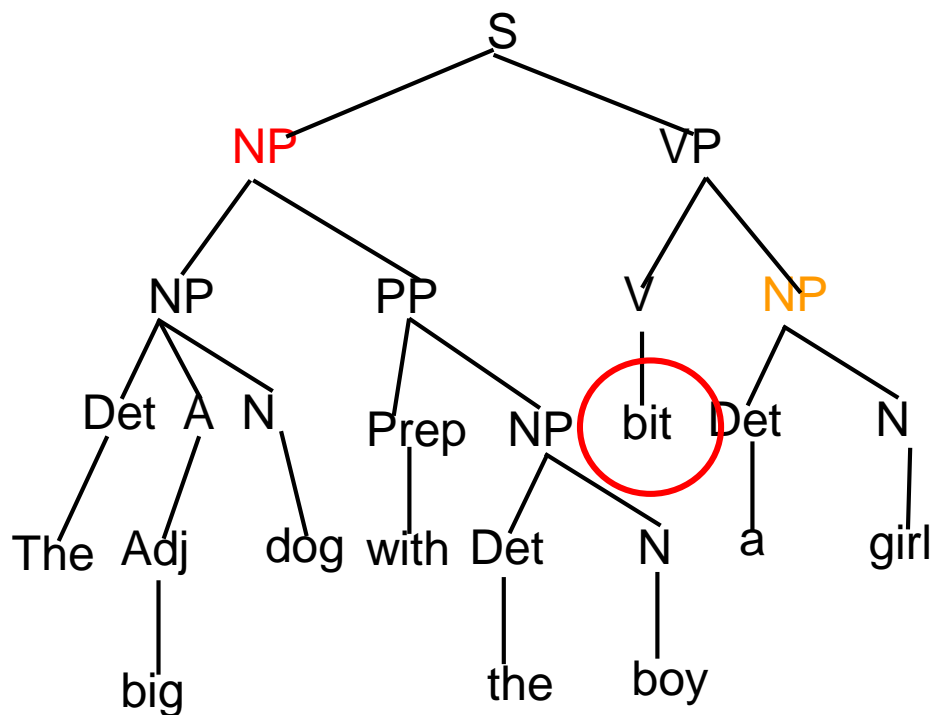
patient

source

destination

instrument

beneficiary



基于句法树的SRL

- 对句法节点进行分类
- 能够使用任一分类学习方法
- 特征是关键

基于句法树的SRL

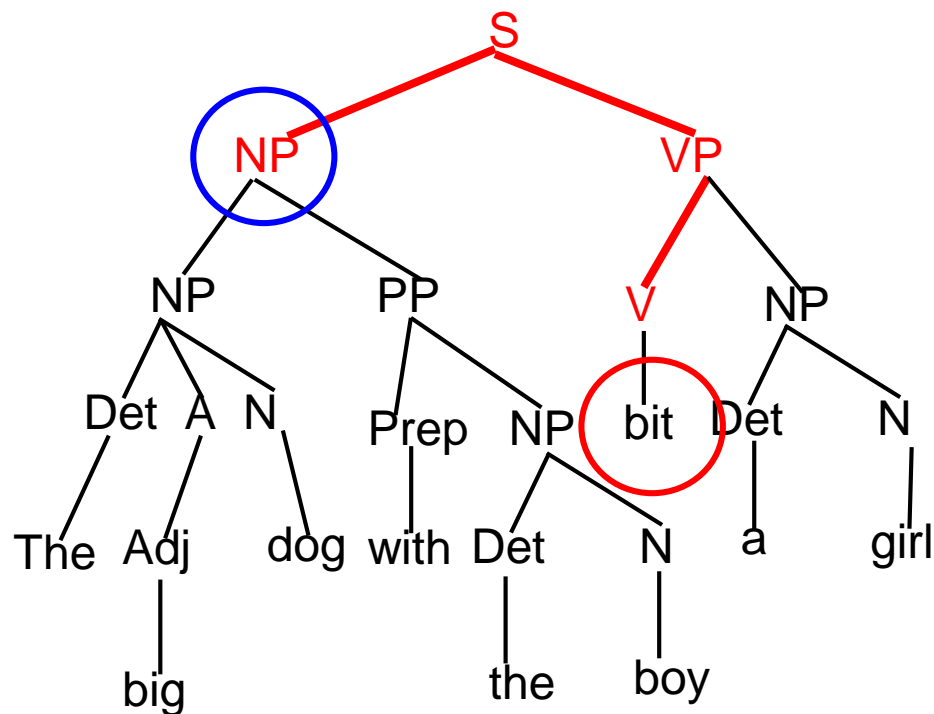
- **特征**

- **Phrase type**: 句法标记 (e.g. NP).
- **Parse tree path**: 候选节点和谓词之间的句法树路径.
- **Position**: 在句子中候选节点在谓词之前还是之后?
- **Voice**: 谓词是主动语态还是被动语态?
- **Head Word**: 候选节点的头词(head word).

基于句法树的SRL

Path Feature Value:

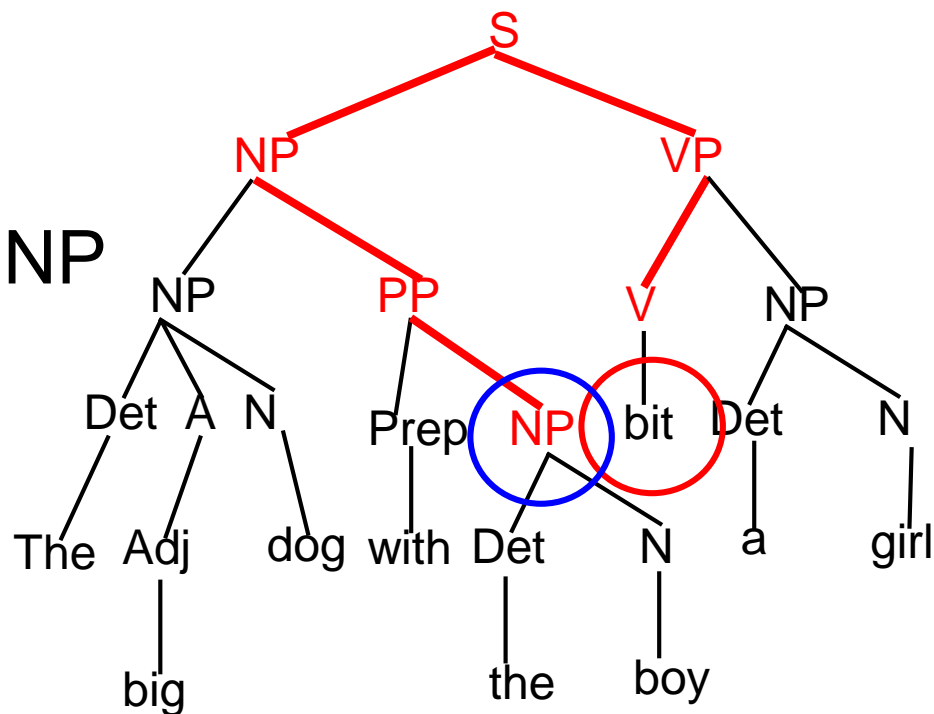
V ↑ VP ↑ S ↓ NP



基于句法树的SRL

Path Feature Value:

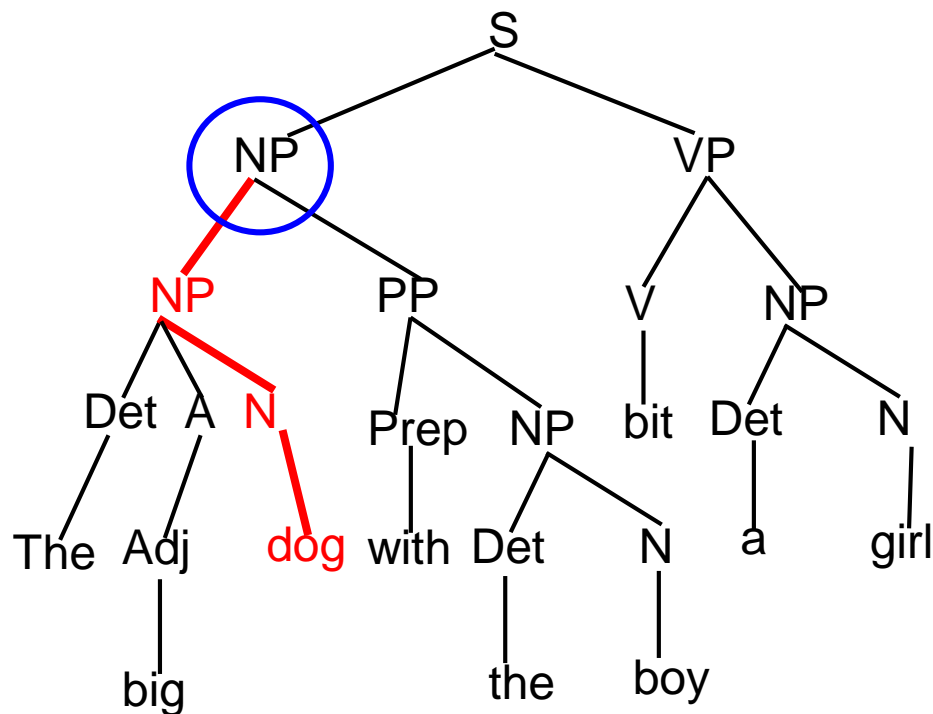
V ↑ VP ↑ S ↓ NP ↓ PP ↓ NP



基于句法树的SRL

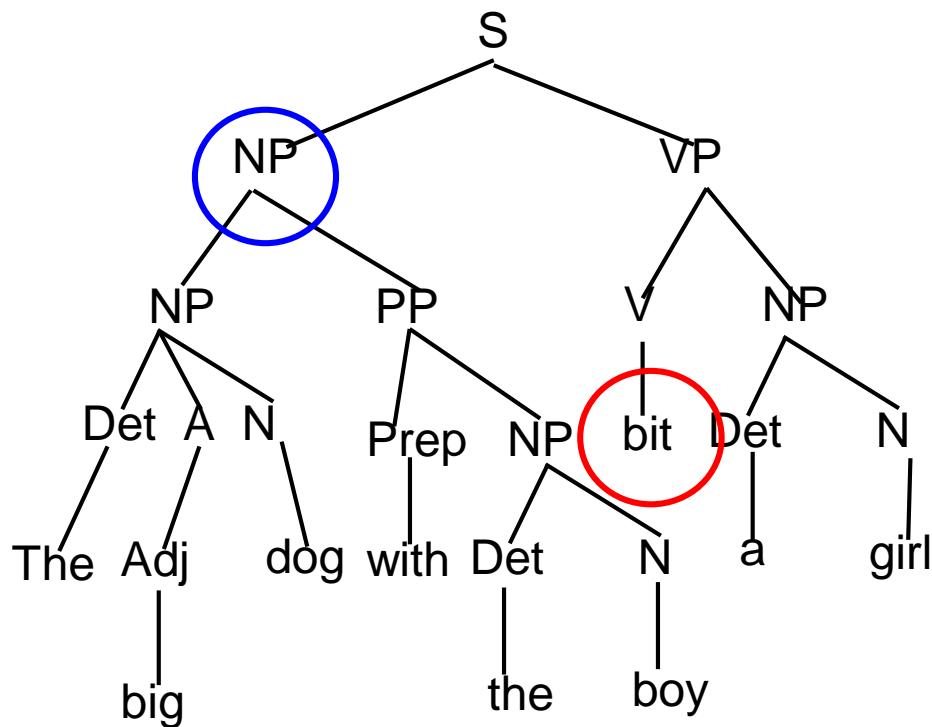
- Head Word: 可通过规则确定

Head Word:
dog



基于句法树的SRL

Phrase type	Parse Path	Position	Voice	Head word
NP	$V \uparrow VP \uparrow S \downarrow NP$	precede	active	dog



基于句法树的SRL

- 业界提出很多其他的有用特征
- 分类结果可能会违反约束
 - 例如: an action has at most one agent?
 - 可使用一些方法(比如ILP等)加强约束, 进行最终的选择
- 句法分析的错误会导致句法树不准确
 - 可使用多个可能的句法树, 然后对结果进行融合
 - 对句法分析与SRL进行联合建模
- 还可采用两步走策略
 - 首先确定节点是否为角色(argument).
 - 如果节点是角色, 那么确定其角色类型(type).

SRL数据集

- FrameNet
 - Developed at Univ. of California at Berkeley
 - Based on notion of Frames
- PropBank
 - Developed at Univ. of Pennsylvania
 - Based on elaborating their Treebank
- Chinese PropBank

句子级语义分析效果

- 语义角色标注的总体水平(F值)在80%左右
- 深层语义分析很困难，目前并没有成熟的技术和系统能够生成逻辑表达式

篇章分析

篇章(Discourse)

- 之前通常单独地从语法或语义上分析一个句子
- 自然语言通常由一系列句子组成
- 通常, 一个句子很难被单独理解

Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a kite. "Don't do that," said Penny. "Jack has a kite. **He will make you take *it* back.**"

篇章(Discourse)

- 篇章是一组连贯且有结构的句子
(*Discourse* is a coherent structured group of sentences)
 - 例如：独白(monologues), 对话(dialogues)
- 篇章分析的主要任务
 - 篇章分割(Discourse Segmentation)
 - 句间关系识别(Determining Coherence Relations)
 - 指代消解(Reference Resolution)
- 理想情况下需要深层文本理解技术来应对以上任务, 但目前为止主要采用浅层分析方法

篇章分割(Discourse Segmentation)

- **篇章分割: 将文档分割成子话题的线性序列**
 - 每个子话题可由多个自然段组成
 - 例如: 科技论文可分割为: Abstract, Introduction, Methods, Results, Conclusions
- **篇章分割的应用:**
 - 文档摘要: 对每个段落单独摘要
 - 信息检索与信息抽取: 在合适的段落上进行
- **相关任务: 对于语音识别文本的段落分割**

无监督的篇章分割

- 无训练数据
- 基于凝聚性的方法(Cohesion-based approach)
 - 将文档分割成子话题，每个子话题中的段落/句子之间相互凝聚，子话题边界处的凝聚性较差

凝聚性(Cohesion)

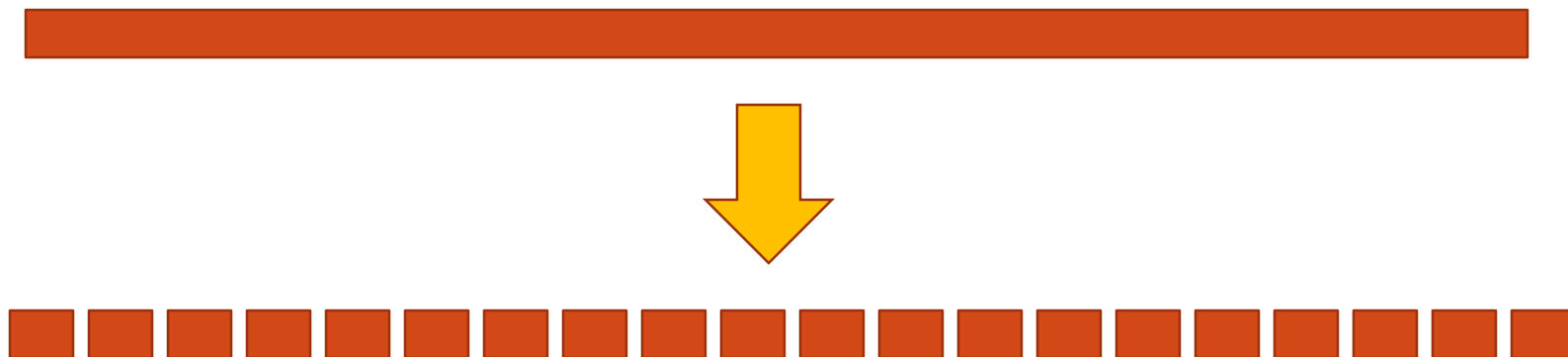
- 文本单元之间的语言关联性
- 词汇凝聚性(Lexical Cohesion): 使用相同或相似的词语关联文本单元
 - Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a kite. "Don't do that," said Penny. "Jack has a kite. He will make you take it back."
- 非词汇凝聚性(Non-lexical Cohesion): 例如, 回指

基于凝聚性的无监督篇章分割

- **TextTiling 算法(Hearst, 1997)**
 - 比较相邻的文本块
 - 寻找词汇上的转换

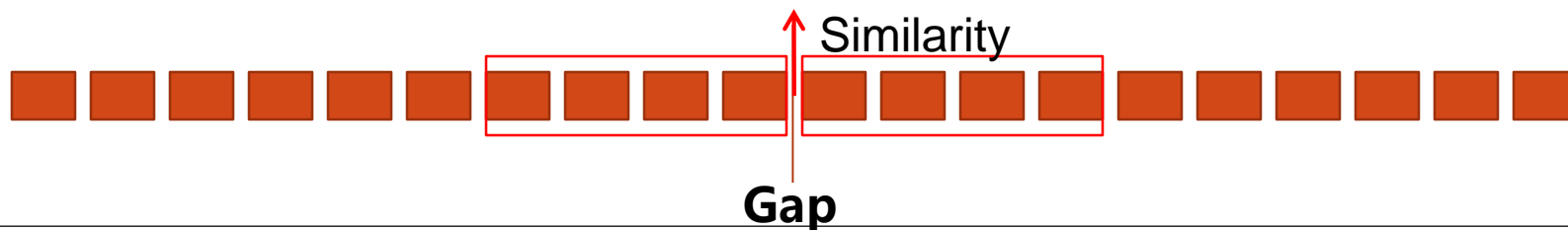
TextTiling算法

- 预处理: Tokenization, remove stop words, stemming
- 将文本切割成同样长度的文本块 (例如 20个词)



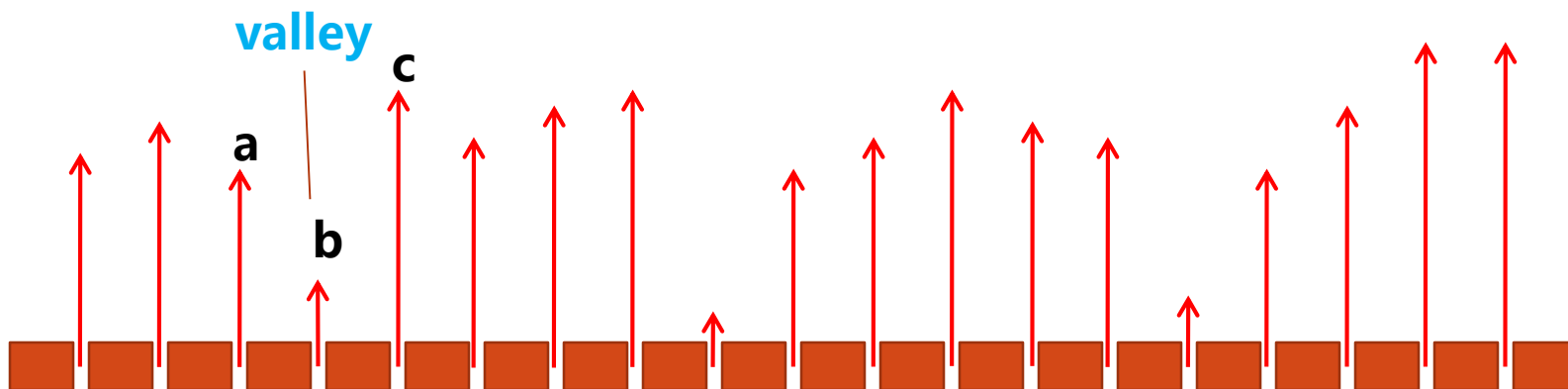
TextTiling算法

- 计算文本块间隙(gap)的词汇凝聚值
- 词汇凝聚值: 在间隙之前与之后的词语之间的相似度
 - 可采用窗口控制
 - 余弦相似度



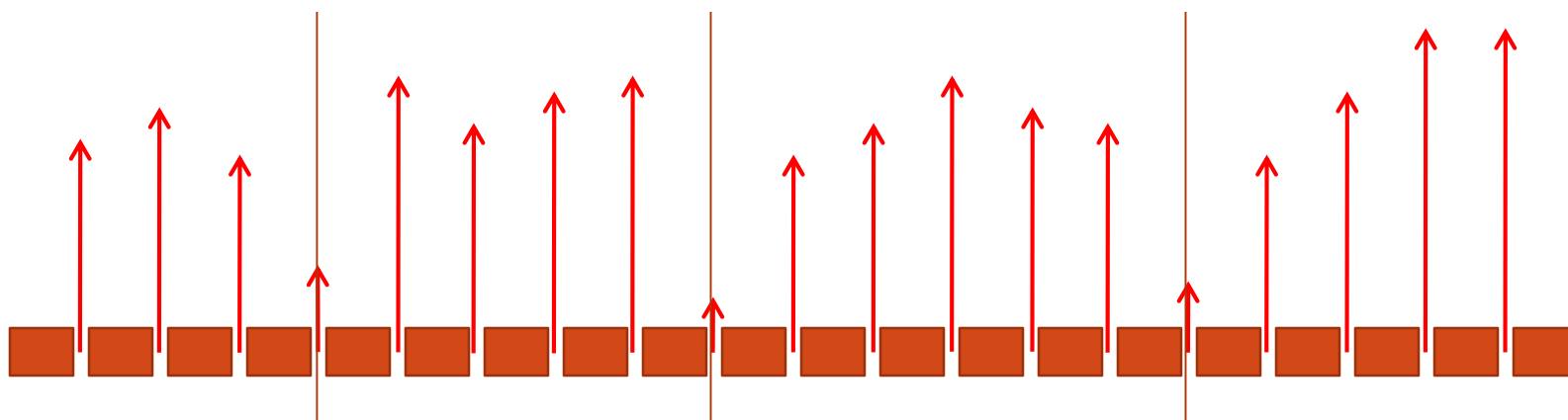
TextTiling算法

- 绘制相似度值，计算相似度较低的间隙的深度值 (the depth scores of the “similarity valleys”): $(a-b) + (c-b)$
- 如果深度值大于设定的阈值，那么该间隙为分割边界



TextTiling算法

- 绘制相似度值，计算相似度较低的间隙的深度值 (the depth scores of the “similarity valleys”): $(a-b) + (c-b)$
- 如果深度值大于设定的阈值，那么该间隙为分割边界



TextTiling算法

人工分
割结果

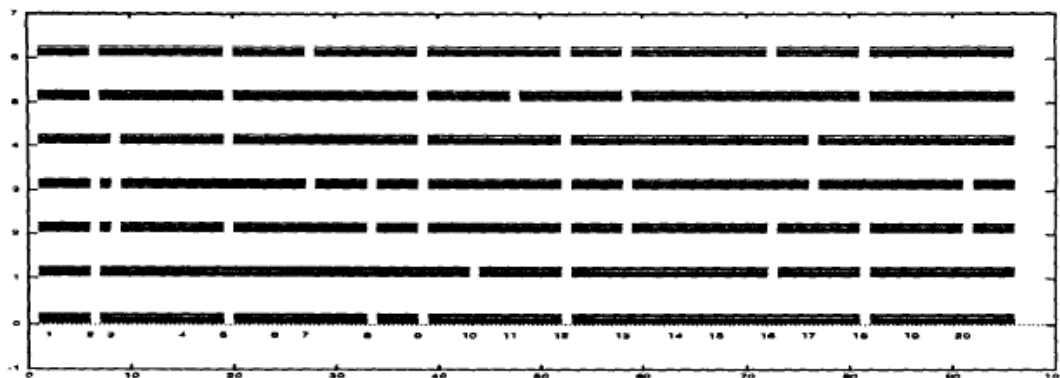


Figure 3: Judgments of seven readers on the *Stargazer* text. Internal numbers indicate location of gaps between paragraphs; x-axis indicates token-sequence gap number, y-axis indicates judge number, a break in a horizontal line indicates a judge-specified segment break.

算法分
割结果

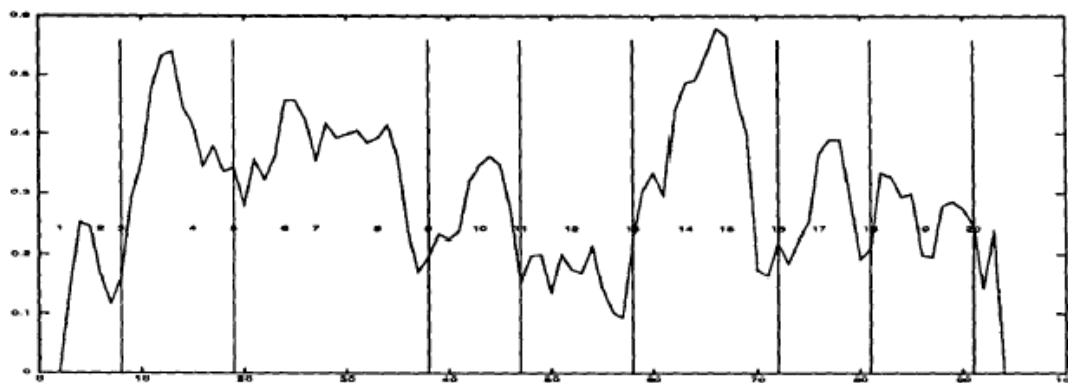


Figure 4: Results of the block similarity algorithm on the *Stargazer* text. Internal numbers indicate paragraph numbers, x-axis indicates token-sequence gap number, y-axis indicates similarity between blocks centered at the corresponding token-sequence gap. Vertical lines indicate boundaries chosen by the algorithm; for example, the leftmost vertical line represents a boundary after paragraph 3. Note how these align with the boundary gaps of Figure 3 above.

有监督的篇章分割

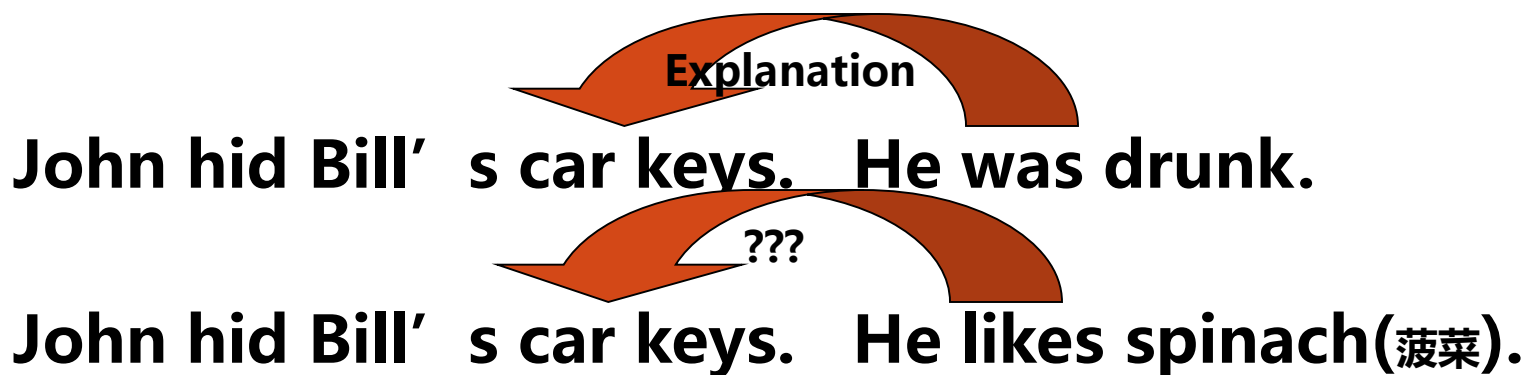
- **对于某些分割任务容易获取标注数据**
 - 例如：自然段落划分
 - 用于在语音识别结果中划分段落
- **建模为分类任务：判断句子边界是否为段落边界**
 - 使用任一分类器：SVM, Naïve Bayes, Maximum Entropy等
- **或者，建模为序列标注任务：将每个句子边界标注为段落边界标记或者非段落边界标记**

有监督的篇章分割

- **特征:**
 - **凝聚性特征:** 词重叠, 词余弦相似度, 回指等
 - **其他特征:** 篇章标识或线索词(Discourse markers or cue word)
- **篇章标识或线索词:** 能够预示篇章结构的词或短语
 - 例如, 广播新闻中 “good evening” , “joining us now”
 - “Coming up next” 作为子话题的结束, 等.
 - 可以人工撰写、也可基于特征选择自动确定

文本连贯性(Text Coherence)

- 一系列独立的句子并不能构成一个篇章，因为缺乏连贯性
- 连贯性：两个文本单元之间的意义关系，可解释不同文本单元的意义是如何结合起来构建更大文本单元的意义



连贯性关系(Coherence Relations)

- Hobbs (1979)定义了12种关系, 例如:

Result

The Tin Woodman was caught in the rain. His joints rusted.

Parallel

The scarecrow wanted some brains. The Tin Woodman wanted a heart.

Elaboration

Dorothy was from Kansas. She lived in the midst of the great Kansas prairies.

Occasion

Dorothy picked up the oil-can. She oiled the Tin Woodman's joints.

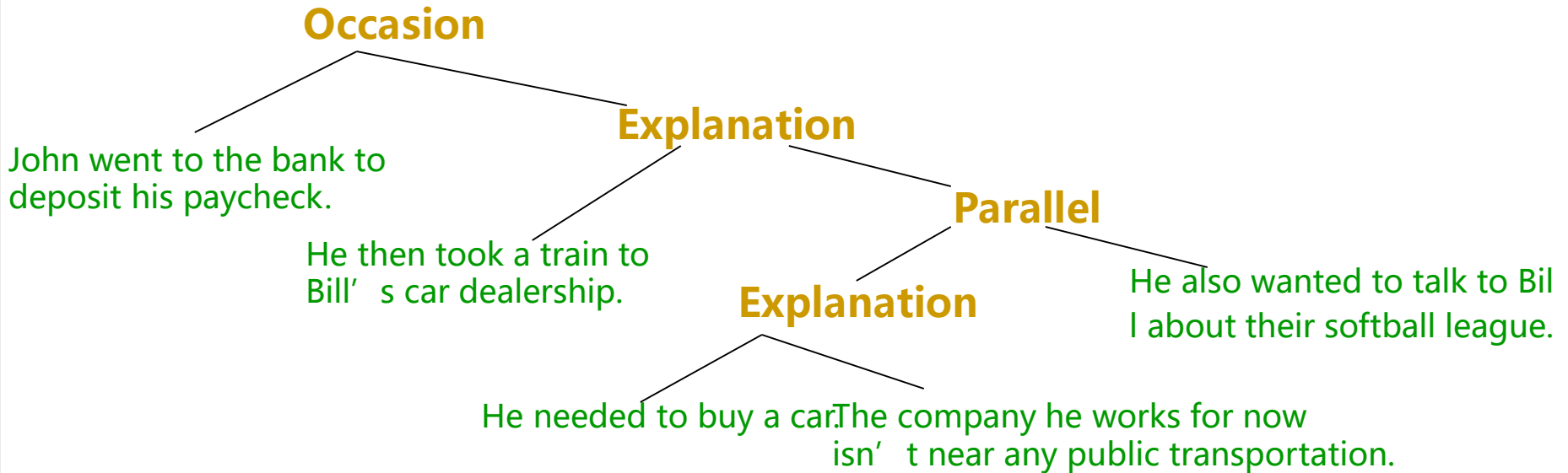
- RST(Mann and Thompson 1987)定义了25种

篇章结构(Discourse Structure)

- 篇章结构: 基于连贯关系的篇章层次式结构
 - 类似句法树结构
 - 树节点表示句子之间的连贯关系: discourse segment (not linear)

John went to the bank to deposit his paycheck. He then took a train to Bill' s car dealership. He needed to buy a car. The company he works for now isn' t near any public transportation. He also wanted to talk to Bill about their softball league.

篇章结构(Discourse Structure)



篇章结构(Discourse Structure)

- 篇章结构的应用

- 文摘系统：可以忽略或合并被Elaboration 关系连接的单元
- 问答系统：利用Explanation关系进行回答
- 信息抽取系统：不需要对从没有连贯关系的单元上抽取的信息融合

篇章解析(Discourse Parsing)

- **连贯关系识别(Coherence Relation Assignment):** 自动识别篇章单元之间的连贯关系
- **篇章解析(Discourse Parsing):** 自动获取整个篇章的篇章结构
- **以上两个问题都是难以解决的问题, 但一些浅层的方法能够达到一定的效果, 例如, 基于篇章标识或线索词**

自动连贯关系识别

浅层的基于线索词的方法：

1. 识别文本中的线索词
2. 将文本分割成篇章单元
3. 对相邻单元连贯关系进行判别

自动连贯关系识别

- 识别线索词

- 预示篇章结构的短语, e.g. “joining us now” , “coming up next” etc.
- 连接词: “because” , “although” , “with” , “and”
- 然而, 这些词的出现并不总是指示篇章关系, 而是具有歧义性
 - With its distant orbit, Mars exhibits frigid weather conditions
 - We can see Mars with an ordinary telescope (非线索词)

自动连贯关系识别

- 篇章单元分割

- 通常采用子句(clauses)



Explanation

- With its distant orbit, Mars exhibits frigid weather conditions

- 使用人工规则或句法解析器获取篇章单元

自动连贯关系识别

相邻单元连贯关系判别

- 使用基于线索词或连词的规则
 - 例如，一个以 “because” 为开始的句子预示着其与后一个单元的Explanation关系
- 基于有效特征进行分类

基于线索词方法的不足

- 有时候连贯关系并不是由cue phrases所指示，而是隐含于句法，词汇、否定语义等



- I don' t want a truck. I' d prefer a convertible.
- 难以人工制定规则或获取训练数据
- 一种解决办法: 通过线索词自动找到简单的样例，然后删除其中的线索词从而产生难度大的训练样例
 - I don' t want a truck although I' d prefer a convertible. (简单样例)
 - I don' t want a truck. I' d prefer a convertible. (难度样例)
- 基于词、词对、词性等特征进行训练

Penn Discourse Treebank

- **大规模篇章语料资源**
 - 标注篇章关系、连接词意义等
 - 跟Penn Treebank关联

<http://www.seas.upenn.edu/~pdtb/>

指代消解(Reference Resolution)

- 确定语言表达所意指(参照)的实体

Mr. Obama visited the city. The president talked about Milwaukee's economy. He mentioned new jobs.

- “Mr.Obama” , “The president” and “He” 是引用表达式(*referring expressions*) , “Barack Obama” 是它们的指称对象(*referent*) , 它们共指 (*corefer*)
- *Anaphora* (回指) :引用表达式指向之前提到的实体 (antecedent先行词), 则称为anaphoric, e.g. “The president” , “He”
- *Cataphora* (后指) :引用表达式指向之后提到的实体 , 则称为cataphoric, e.g. “the city”

指称表达式

- **Indefinite noun phrases (NPs):** e.g. “a cat”
 - 在篇章上下文中介绍新事物
- **Definite NPs:** e.g. “the cat”
 - 指向听众熟悉的事物
- **Pronouns:** e.g. “he” , “ she” , “it”
- **Demonstratives:** e.g. “this” , “that”
- **One-anaphora:** “one”

两类指代消解任务

- *Coreference Resolution* (共指消解): 发现指向相同实体的指称表达式, 也就是寻找共指链
 - 前页例子中共指链为: {Mr. Obama, The president, he}, {the city, Milwaukee' s}
- *Pronominal Anaphora Resolution* (人称代词消解): 找到人称代词所指向的先行词
 - 前页例子中, “he” 指向 “Mr. Obama”
 - 是共指消解的子任务

人称代词消解的特征

- 约束

- 数量一致性

- Singular pronouns (it/he/she/his/her/him) refer to singular entities and plural pronouns (we/they/us/them) refer to plural entities

- 人称一致性

- He/she/they etc. must refer to a third person entity

- 性别一致性

- He -> John; she -> Mary; it -> car

- 句法约束

- John bought himself a new car. [himself -> John]
 - John bought him a new car. [him can not be John]

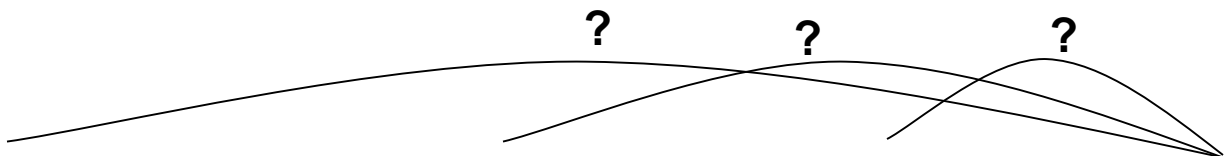
人称代词消解的特征

- 优先性:
 - 就近原则(Recency): 最近被提到的实体更可能被指代
 - John went to a movie. Jack went as well. He was not busy.
 - 语法角色(Grammatical Role): 主语位置的实体比宾语位置的实体更可能被指代
 - John went to a movie with Jack. He was not busy.
- 平行(Parallelism):
 - John went with Jack to a movie. Joe went with him to a bar.

人称代词消解的特征

- 优先性:
 - 动词语义(Verb Semantics): 某些动词对后面的代词指向其主语或宾语有所偏好
 - John telephoned Bill. He lost the laptop.
 - John criticized Bill. He lost the laptop.
 - 选择限制(Selectional Restrictions): 语义上的限制
 - John parked his car in the garage after driving it around for hours.
- 可使用上述特征进行分类

Mr. Obama visited the city. The president talked about Milwaukee's economy. He mentioned new jobs.



共指消解

- 能够用类似人称代词消解的方法解决：基于分类
- 除了二类分类，还可以对指称表达式进行聚类
- 扩展：跨文档共指消解
 - 问题：“John Smith” in 文档A = “John Smith” in 文档B?
 - 方法：
 - 综合考虑：
 - Within-document co-reference
 - Vector Space Model similarity

- **Some slides were taken or adapted from related slides written by Lucas Champollion, Rohit Kate, Raymond Mooney, Scott Yih, Kristina Toutanova, Stina Ericsson, etc. Thank them for sharing their slides.**

