自然语言处理

Natural Language Processing

权小军 教授

中山大学数据科学与计算机学院

quanxj3@mail.sysu.edu.cn

课程回顾

概述

- 句法分析是自然语言处理中的基础性工作，它分析**句子的句法结构**（主谓宾结构）和**词汇间的依存关系**（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

概述

- 句法分析是自然语言处理中的基础性工作，它分析**句子的句法结构**（主谓宾结构）和**词汇间的依存关系**（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

句法分析不是自然语言处理任务的最终目标，但它往往是实现最终目标的一个关键环节！

概述

- **任务类型:**
  - 短语结构分析(Phrase Parsing) ,也叫成分结构分析
    - 分析句子的主谓宾定状补的句法结构
      - 完全句法分析：以获取整个句子的句法结构为目的；
      - 局部句法分析：以获得局部成分为目的；
  - 依存句法分析(Dependency Parsing)
    - 通过分析语言单位内成分之间的依存关系揭示其句法结构，如并列、从属、比较、递进等。

短语结构分析

- 目标： 实现高正确率、高鲁棒性(robustness)、高速度的自动句法分析过程;
- 困难： 自然语言中存在大量的复杂的结构歧义(structural ambiguity)；

短语结构分析

- 基本规则和统计的句法分析器：
  - 基于CFG规则的分析方法
    - CFG: Context-Free Grammar (上下文无关文法)
    - 代表：线图分析法(chart parsing)
  - 基于 PCFG 的分析方法
    - PCFG: Probabilistic Context-Free Grammar (概率上下文无关文法)

1、概述

2、短语结构分析

a) 上下文无关文法

上下文无关文法（CFG）

- **CFG由一系列规则组成，每条规则给出了语言中的某些符号可以被组织或排列在一起的方式.**

<u>符号被分成两类</u>：

- 终结点(叶子节点)：就是指单词，例如 book；
- 非终结点(内部节点)：句法标签，例如 NP 或者 NN；

<u>规则是由一个“→”连接的表达式</u>：

- 左侧：只有一个 non-terminal；
- 右侧：是一个由符号组成的序列；

上下文无关文法（CFG）

CFG示例:

- □ 符号:
  - ■ 终结点: rat, the, ate, cheese;
  - ■ 非终结点: S, NP, VP, DT, VBD, NN;
- □ 规则:

$S \rightarrow NP\ VP$
$NP \rightarrow DT\ NN$
$VP \rightarrow VBD\ NP$
$DT \rightarrow the$
$NN \rightarrow rat$
$NN \rightarrow cheese$
$VBD \rightarrow ate$

基于上下文无关文法的句法分析

基于上下文无关文法（CFG）的句法分析是指基于预定义的语法，为输入语句生成恰当的句法树，要求该树：

- ✓ 符合给定语法；
- ✓ 叶子节点包含所有的词；

1、概述

2、短语结构分析

a) 上下文无关文法

b) 线图分析法

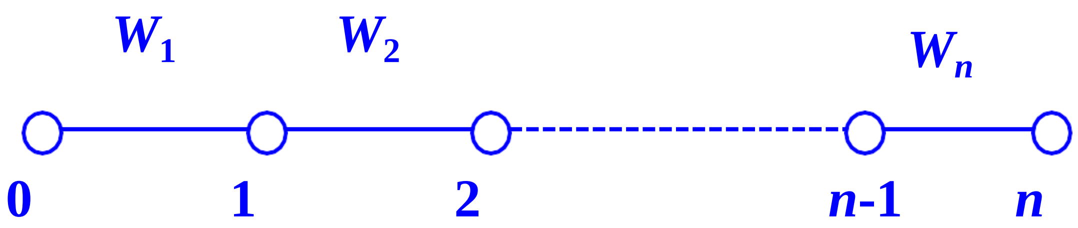
线图分析法

❑ 三种策略

- 自底向上 (Bottom-up)
- 从上到下 (Top-down)
- 从上到下和从下到上结合

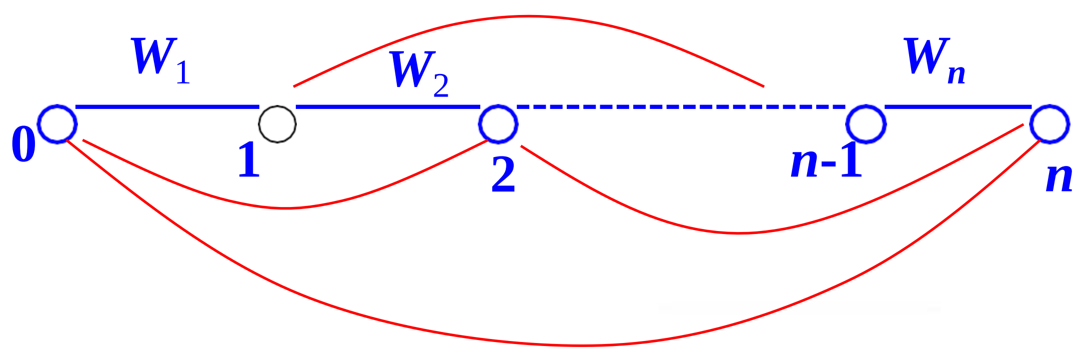
线图分析法

- 自底向上的线图分析算法
  - 给定一组 CFG 规则： $XP \rightarrow \alpha_1 \ldots \alpha_n \quad (n \geq 1)$
  - 给定一个句子的**词性序列**： $S = W_1 W_2 \cdots W_n$
  - 构造一个线图：一组结点和边的集合；



线图分析法

执行：查看任意相邻几条边上的词性串是否与某条规则的右部相同，如果相同，则增加一条新的边跨越原来相应的边，新增加边上的标记为这条规则的头(左部)。重复这个过程，直到没有新的边产生。



1、概述

2、短语结构分析

a) 上下文无关文法

b) 线图分析法

c) **概率上下文无关文法**

概率上下文无关文法

- 对于可能产生多种语法分析结果的问题，我们该如何应对呢?
- 引入概率上下文无关文法（PCFG， Probabilistic context-free grammar）： **给每棵树计算一个概率！**

概率上下文无关文法

❑ PCFG 规则

形式：$A \rightarrow \alpha \qquad [p]$

- $\mathrm{NP} \rightarrow \mathrm{DT\ NN} \qquad [p = 0.45]$
- $\mathrm{NN} \rightarrow \mathrm{leprechaun} \qquad [p = 0.0001]$

概率上下文无关文法

❑ PCFG 规则

形式：$A \rightarrow \alpha \qquad [p]$

约束：$\sum_{\alpha} p(A \rightarrow \alpha) = 1$

例如：
$$\left.\begin{array}{l} NP \rightarrow NN \;\; NN, \;\; 0.60 \\ NP \rightarrow NN \;\; CC \;\; NN, \;\; 0.40 \end{array}\right\} \sum p = 1$$

概率上下文无关文法

给定一个语法分析树，我们可以计算它的概率：

$$P(T) = \prod_{i=1}^{n} P(\mathrm{RHS}_i | \mathrm{LHS}_i)$$

Lecture 11: 句法分析（下）

提 纲

1、依存句法分析概述

2、短语结构与依存结构的关系

3、汉英句法结构特点对比

4、应用实例：情感分析

依存句法分析

❑依存句法理论

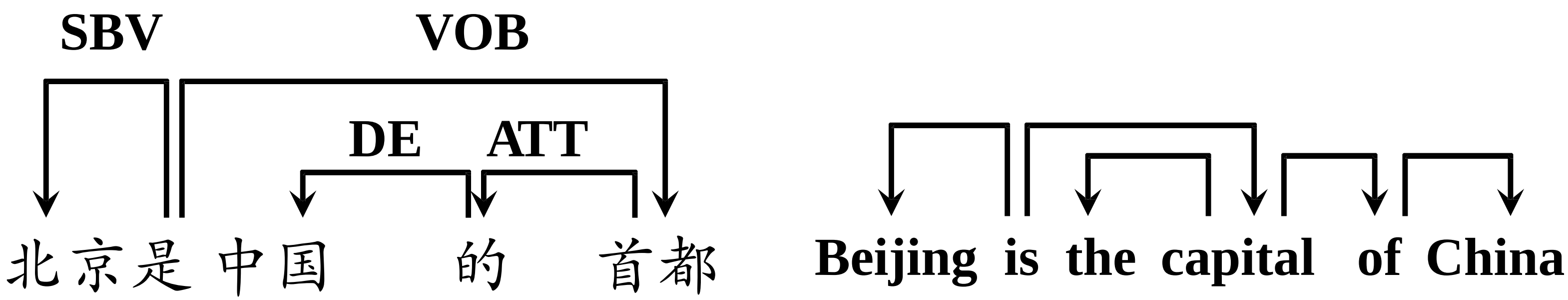
现代依存语法理论的创立者是法国语言学家吕西安·泰尼埃 (Lucien Tesnière , 1893-1954);

泰尼埃 认为：一切结构句法现象可以概括为关联(connexion)、组合(jonction)和转位(tanslation)这三大核心。句法关联建立起词与词之间的从属关系，这种从属关系是由支配词和从属词联结而成；动词是句子的中心，并支配其他成分，它本身不受其他任何成分的支配。

依存句法分析

在依存语法理论中，依存就是指词与词之间支配与被支配的关系，这种关系不是对等的，而是有方向的。处于支配地位的成分称为支配者(governor)，而处于被支配地位的成分称为从属者 (modifier)。

依存句法分析



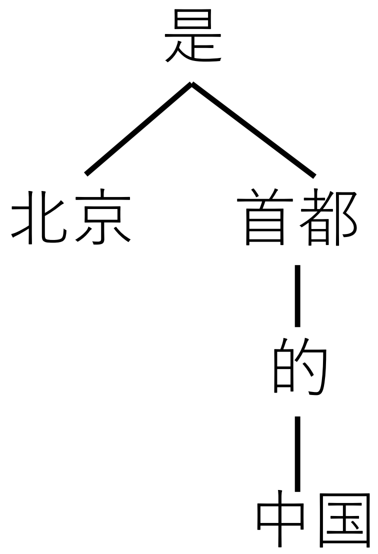
(a) 有向图-1

(b) 有向图-2

用带有方向的弧(或称边)来表示两个成分之间的依存关系，支配者在有向弧的发出端，被支配者在箭头端，我们通常说被支配者依存于支配者。

依存句法分析

(c) 依存树

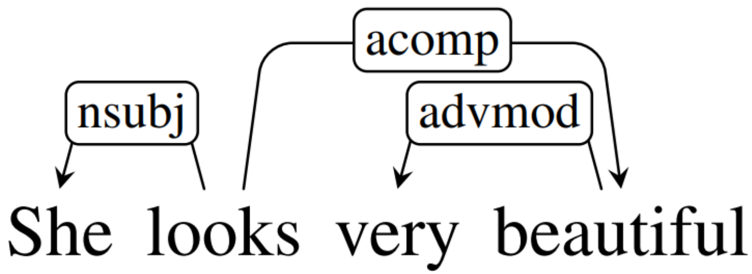


Usually, dependencies form a tree (connected, acyclic, single-head)

图(c)是用树表示的依存结构，树中子节点依存于该节点的父节点。

依存句法分析 Stanford Dependencies representation

- *acomp*: adjectival complement (形容词补语)
  - 动词的形容词补语是形容词短语，起补语的作用



Source: *https://nlp.stanford.edu/software/dependencies_manual.pdf*

依存句法分析 Stanford Dependencies representation

- *advmod*: adverb modifier (副词修饰语)
  - 一个词的副词修饰语是一个副词或副词短语，用来修饰词的意思。

"Genetically modified food"	*advmod*(modified, genetically)
"less often"	*advmod*(often, less)

Source: *https://nlp.stanford.edu/software/dependencies_manual.pdf*

依存句法分析 Stanford Dependencies representation

- *amod*: adjectival modifier (形容词修饰语)
  - 名词短语的形容词修饰语是用来修饰名词短语意义的形容词短语。

"Sam eats red meat"	*amod*(meat, red)
"Sam took out a 3 million dollar loan"	*amod*(loan, dollar)
"Sam took out a $ 3 million loan"	*amod*(loan, $)

Source: *https://nlp.stanford.edu/software/dependencies_manual.pdf*

依存句法分析 Stanford Dependencies representation

- *nsubj*: nominal subject (名词主语)
  - 名词主语是一个名词短语，是从句的句法主语。

"Clinton defeated Dole"	*nsubj*(defeated, Clinton)
"The baby is cute"	*nsubj*(cute, baby)

Source: *https://nlp.stanford.edu/software/dependencies_manual.pdf*

依存句法分析

1970年计算语言学家J. Robinson在论文《依存结构和转换规则》中提出了依存语法的4条公理：

(1) 一个句子只有一个独立的成分；

(2) 句子的其他成分都从属于某一成分；

(3) 任何一成分都不能依存于两个或多个成分；

(4) 如果成分A直接从属于成分B，而成分C在句子中位于A和B之间，那么，成分C或者从属于A，或者从属于B，或者从属于A和B之间的某一成分。

依存句法分析

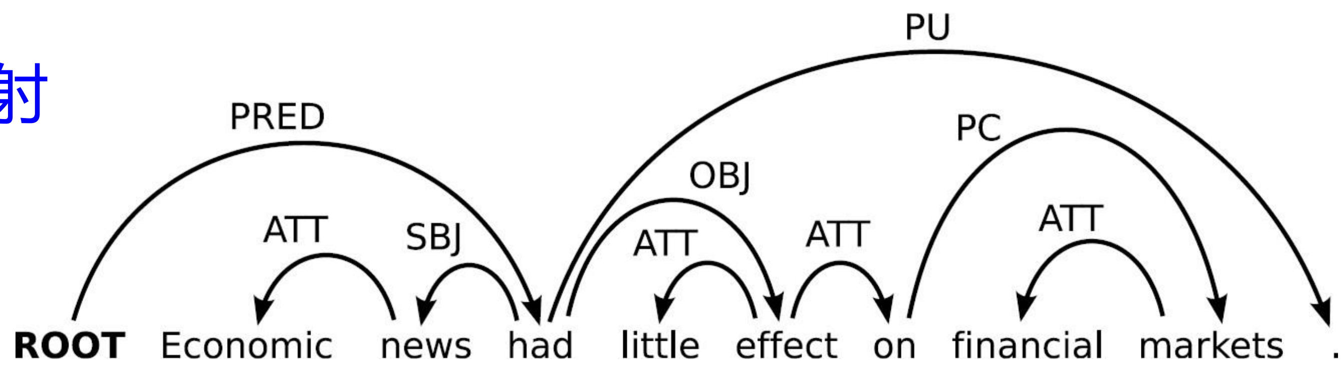
这4条公理相当于对依存图和依存树的形式约束为:

- 单一父结点(single headed)
- 连通(connective)
- 无环(acyclic)
- 可投射(projective)

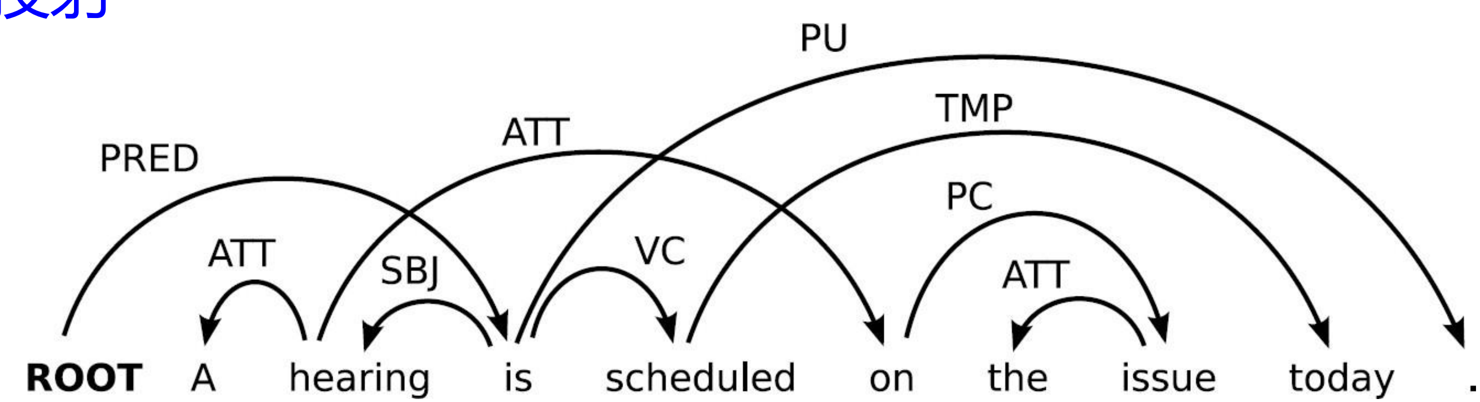
由此来保证句子的依存分析结果是一棵有根的树结构

依存句法分析

投射



非投射



依存句法分析

依存语法的优势

1) 依存关系和实际的语义关系比较接近，有助于对句子的语义方面的理解；
2) 定义相对比较简单，有助于高效率的句法分析；
3) 因为能够有效建模长距离依赖关系，依存句法更适合词序列比较自由、灵活的语言；

依存句法分析

- 依存句法分析方法

依存句法分(dependency parsing)的任务就是分析出句子中所有词汇之间的依存关系。

依存句法分析

传统句法分析算法可大致归为以下4类：

- 生成式的分析方法(generative parsing)
- 判别式的分析方法(discriminative parsing)
- 决策式的(确定性的)分析方法(deterministic parsing)
- 基于约束满足的分析方法(constraint satisfaction parsing)

提 纲

1、依存句法分析概述

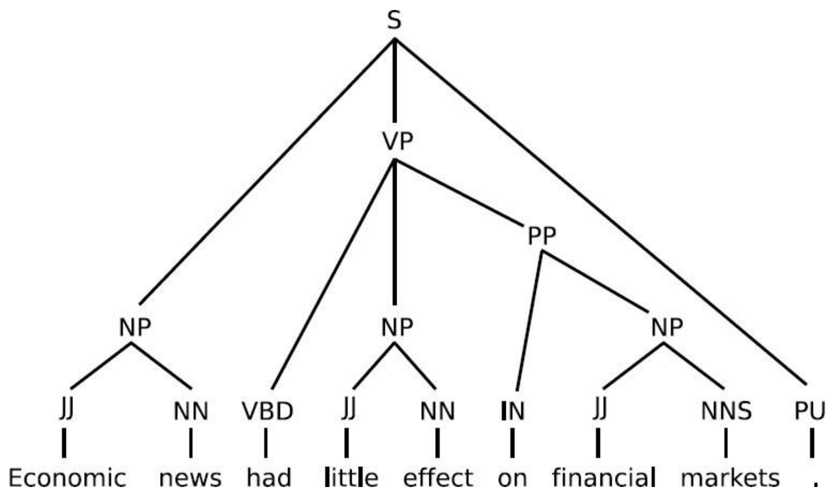
2、短语结构与依存结构的关系

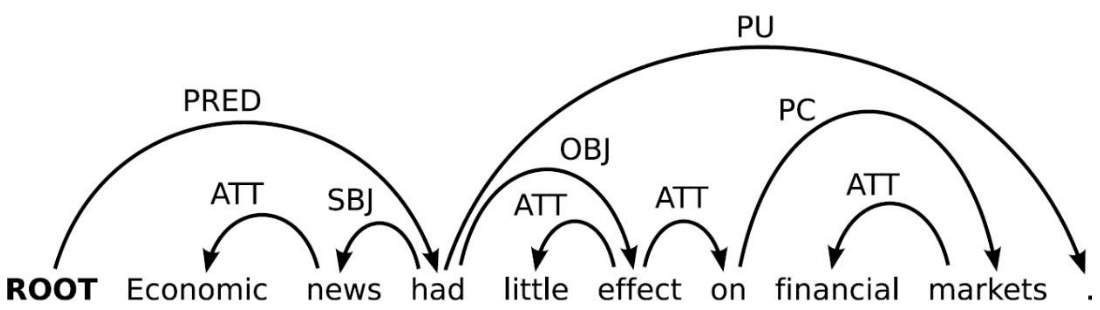
3、汉英句法结构特点对比

4、应用实例：情感分析

短语结构与依存结构

依存结构表达的信息和短语结构句法树不一样，可以表达**更长距离**的信息依存关系





短语结构与依存结构的关系

- 短语结构可转换为依存结构
- 实现方法：

(1) 定义中心词抽取规则，产生中心词表；

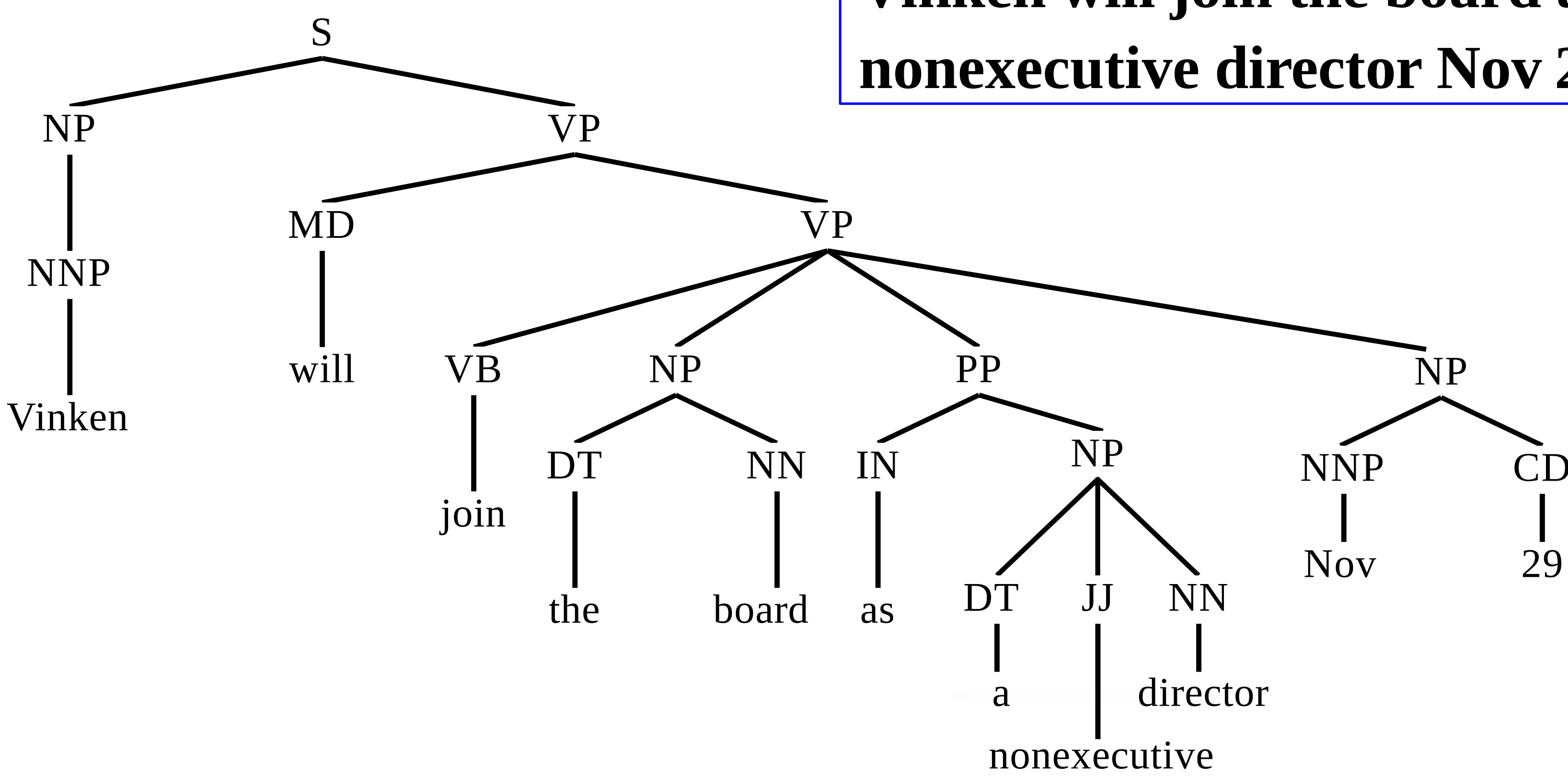
(2) 根据中心词表，为每个节点选择中心子节点；

(3) 将非中心子节点的中心词依存到中心子节点的中心词上，得到相应的依存结构。

短语结构与依存结构的关系

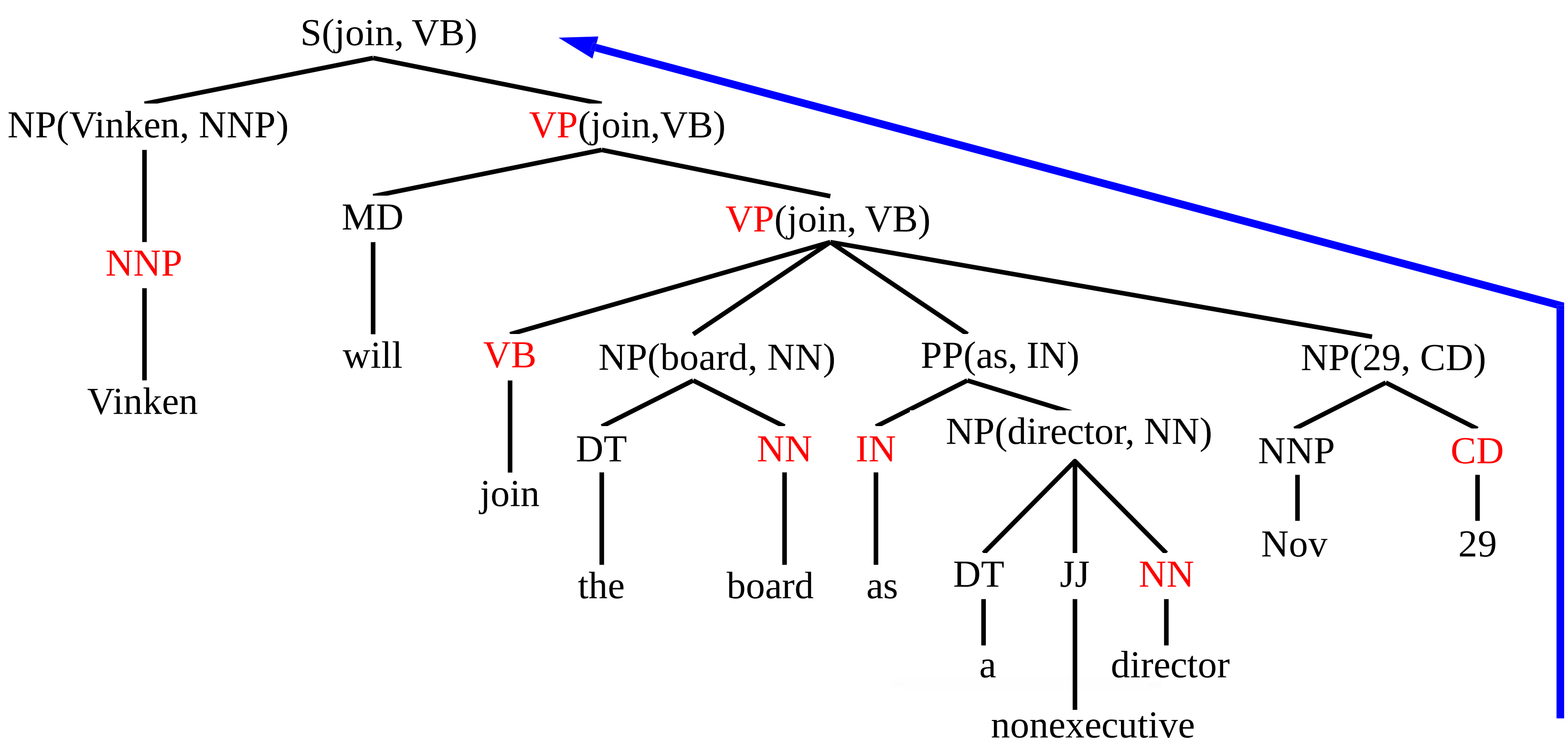
例如：给定如下短语结构树

Vinken will join the board as a nonexecutive director Nov 29



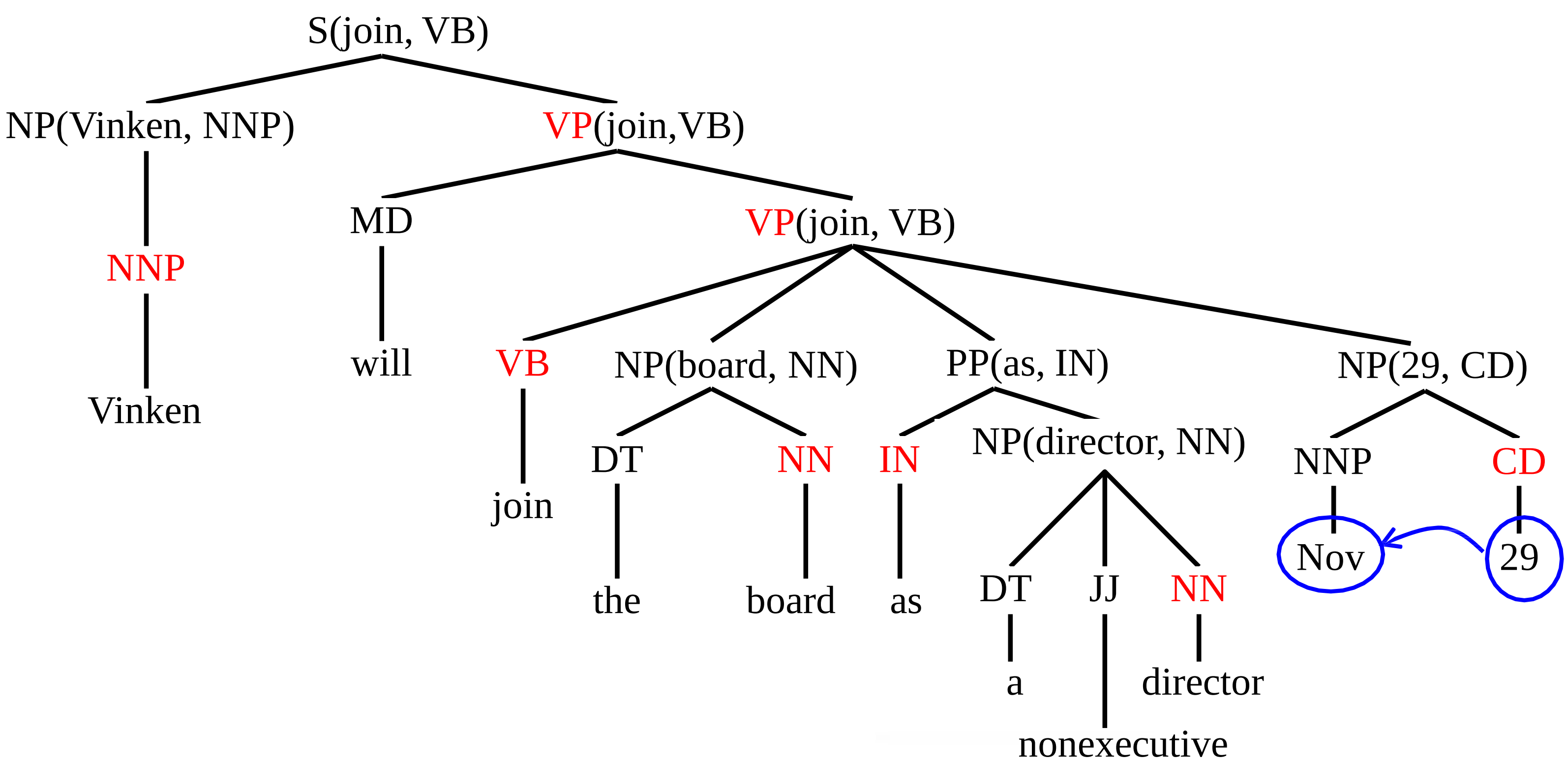
短语结构与依存结构的关系

- 根据中心词表为每个节点选择中心子节点(中心词通过自底向上传递得到)



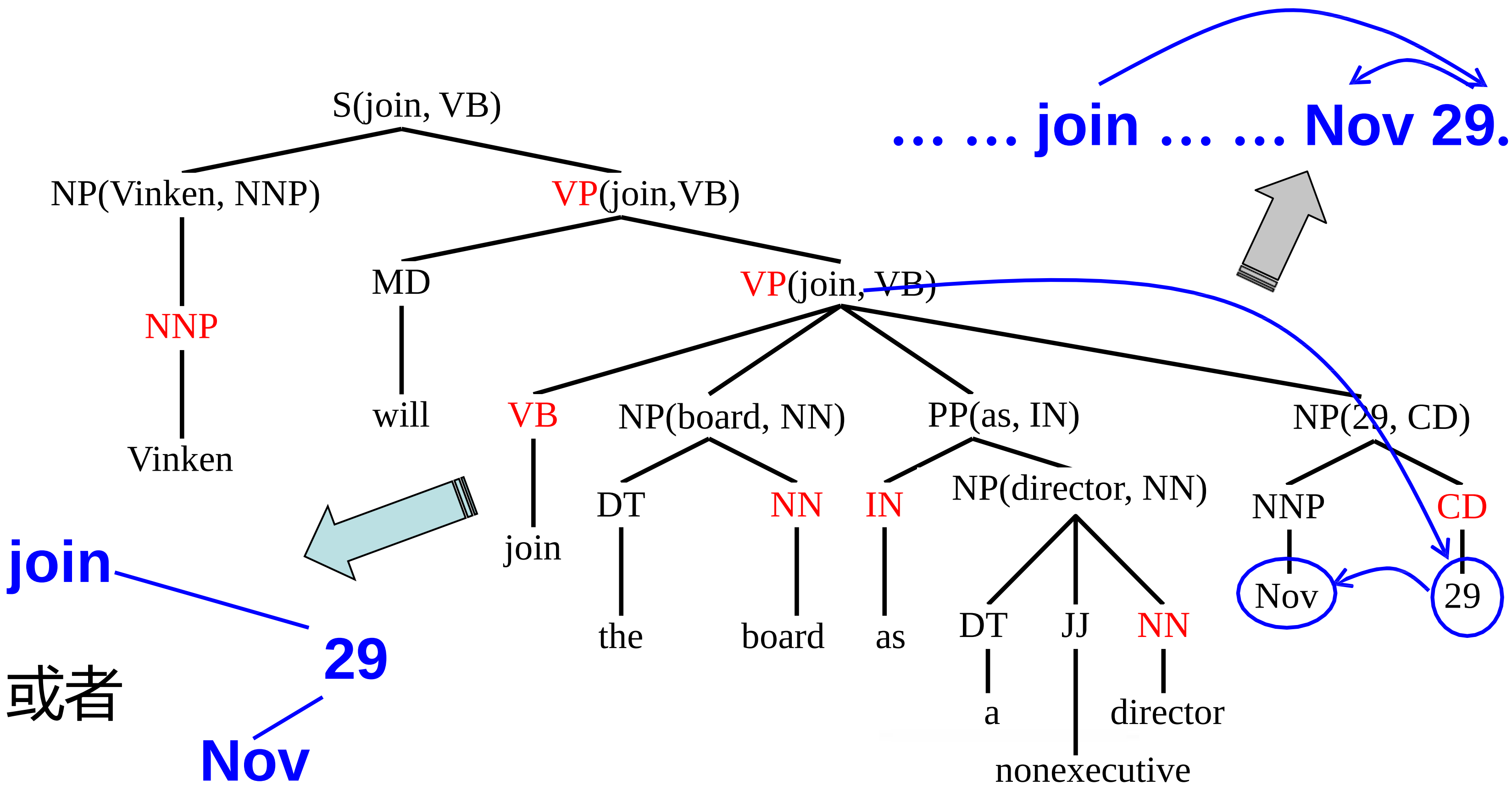
短语结构与依存结构的关系

- 将非中心子节点的中心词依存到中心子节点的中心词上



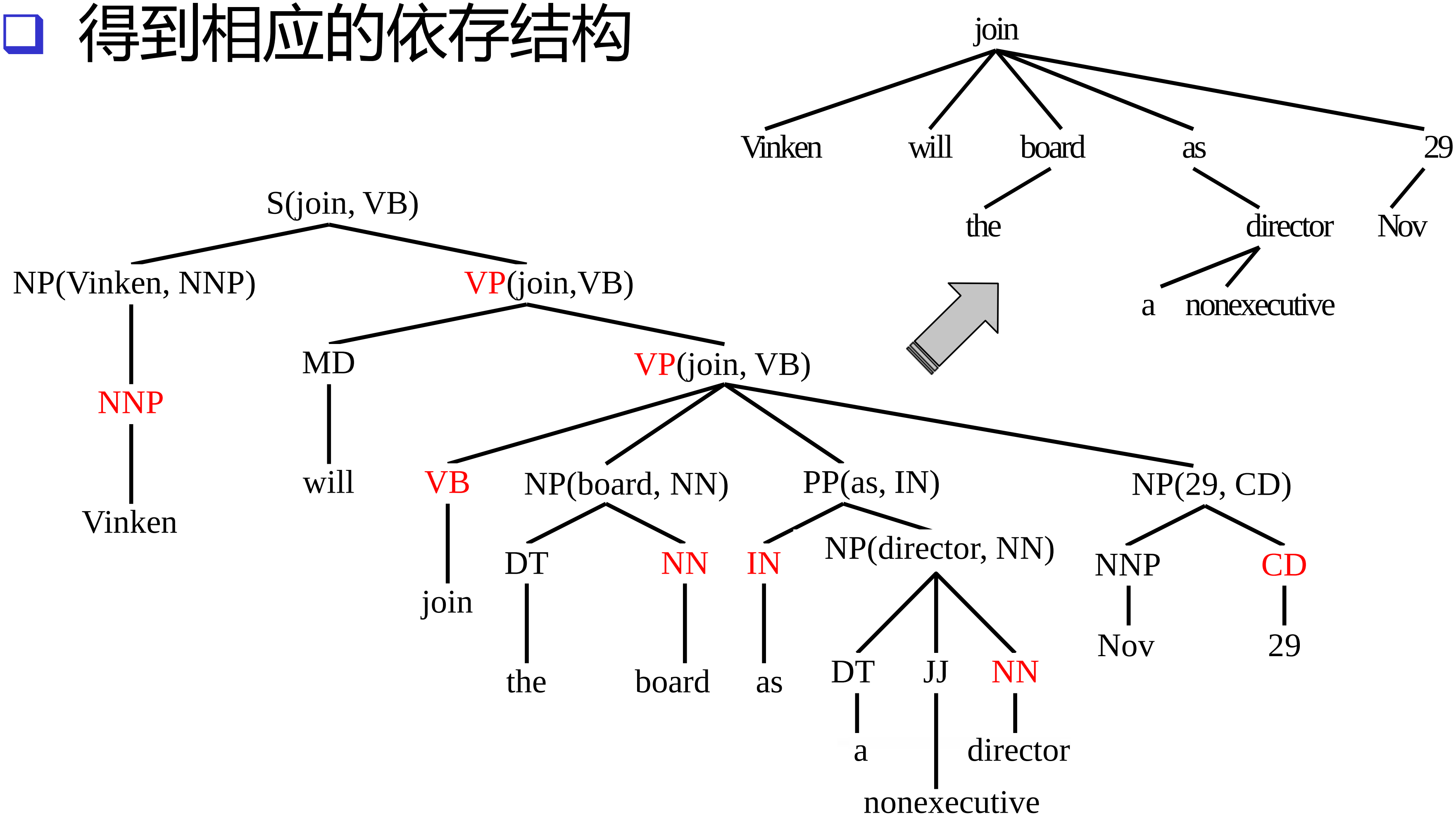
短语结构与依存结构的关系

- 将非中心子节点的中心词依存到中心子节点的中心词上



短语结构与依存结构的关系

- 得到相应的依存结构



提 纲

1、依存句法分析概述

2、短语结构与依存结构的关系

3、汉英句法结构特点对比

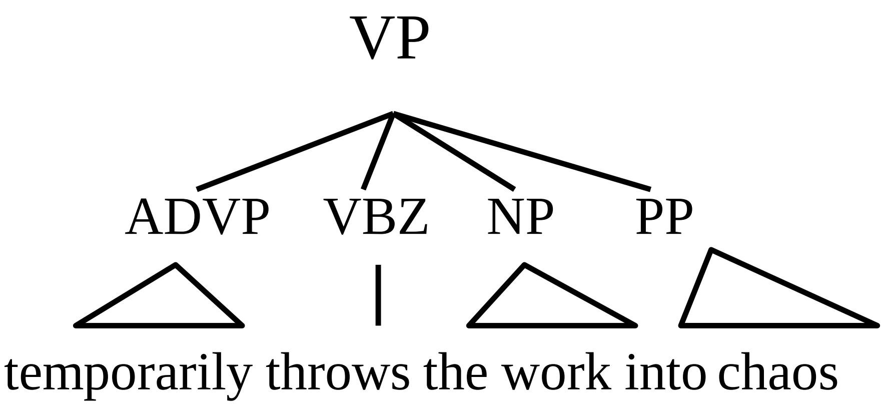
4、应用实例：情感分析

汉英句法结构特点对比

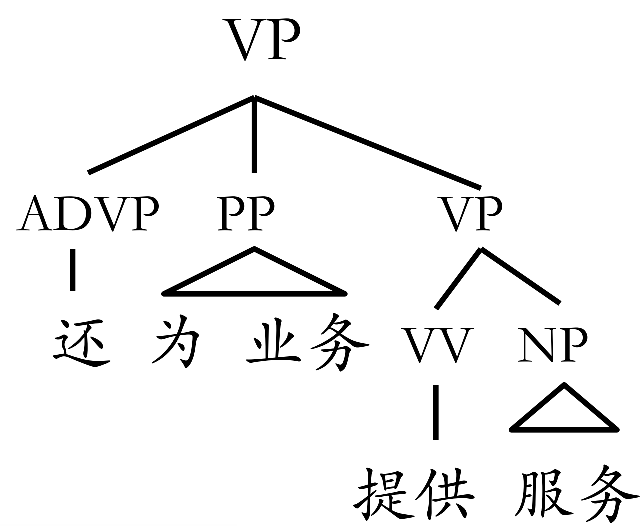
说明：撇开汉语的分词问题和词性消歧错误可能对句法分析器带来的影响，即保证句法分析器的输入为完全正确的词性序列，仅仅考虑句子结构本身的问题；

汉英句法结构特点对比

(1)英语短语绝大多数以左部为中心，而汉语短语比较复杂，大多数短语类是以右部为短语中心，除了动词和介词的补语在它们的中心词之后。如：



(a)



(b)

汉英句法结构特点对比

(2) 在汉语句子中没有做主语的先行代词的情况普遍存在，但在英语中这种情况很少出现。这样就使得汉语句法分析器很难判断一个输入到底是没有主语的子句结构还是仅仅是一个动词短语VP，如：

He thinks <u>it</u> is true. / 他认为□是对的。

汉英句法结构特点对比

英语中当多个单句连接起来构成复句的时候，单句与单句之间需要有显式的连接词或者短语。汉语则不同，一个句子是表达一个完整意义的语言单元，这种特点在长句中表现得特别明显。

汉英句法结构特点对比

这些长句内部的各个简单句是为了表意的需要而连接在一起的，它们彼此的句法结构完全是独立的，表示彼此之间逻辑关系的连接词不是必需的，这类长句在汉语中称之为“流水复句”，例如：

“我现已步入中年，每天挤车，搞得我精疲力尽，这种状况，直接影响我的工作，家里的孩子也没人照顾。”

汉英句法结构特点对比

❑ 汉语长句的层次化句法分析方法

(1) 对包含“分割”标点的长句进行分割;

(2) 对分割后的各个子句分别进行句法分析(第一级分析)，分析得到的子树根节点的词类或者短语类别标记作为第二级句法分析的输入;

(3) 通过第二遍分析找到各子句或短语之间的结构关系，从而获得最终整句的最大概率分析树。

提 纲

1、依存句法分析概述

2、短语结构与依存结构的关系

3、汉英句法结构特点对比

4、应用实例：情感分析

应用实例：情感分析

Relational Graph Attention Network for Aspect-based Sentiment Analysis

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, Rui Wang

Abstract

Aspect-based sentiment analysis aims to determine the sentiment polarity towards a specific aspect in online reviews. Most recent efforts adopt attention-based neural network models to implicitly connect aspects with opinion words. However, due to the complexity of language and the existence of multiple aspects in a single sentence, these models often confuse the connections. In this paper, we address this problem by means of effective encoding of syntax information. Firstly, we define a unified aspect-oriented dependency tree structure rooted at a target aspect by reshaping and pruning an ordinary dependency parse tree. Then, we propose a relational graph attention network (R-GAT) to encode the new tree structure for sentiment prediction. Extensive experiments are conducted on the SemEval 2014 and Twitter datasets, and the experimental results confirm that the connections between aspects and opinion words can be better established with our approach, and the performance of the graph attention network (GAT) is significantly improved as a consequence.

PDF

BibTeX

Search

Video

Anthology ID: 2020.acl-main.295

Volume: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics

应用实例：情感分析

- Aspect-based Sentiment Analysis

Sentence: The appetizers are OK but the service is slow!	
Aspect #1: appetizers	Label: positive
Aspect #2: service	Label: negative

Table 1: An example of Aspect-Based Sentiment Analysis

应用实例：情感分析

- Aspect-based Sentiment Analysis

Sentence: The appetizers are OK but the service is slow!	
Aspect #1: $\boxed{\text{appetizers}}$	Label: $\boxed{\text{positive}}$
Aspect #2: service	Label: negative

Table 1: An example of Aspect-Based Sentiment Analysis

应用实例：情感分析

- Aspect-based Sentiment Analysis

Sentence: The appetizers are OK but the service is slow!	
Aspect #1: **appetizers**	Label: **positive**
Aspect #2: **service**	Label: **negative**

Table 1: An example of Aspect-Based Sentiment Analysis

应用实例：情感分析

- Aspect-based Sentiment Analysis

Sentence: The appetizers are OK but the service is slow!	
Aspect #1: appetizers	Label: positive
Aspect #2: service	Label: negative

Table 1: An example of Aspect-Based Sentiment Analysis

应用实例：情感分析

Opinion words are not always available!

应用实例：情感分析

Opinion words are not always available!

For example:

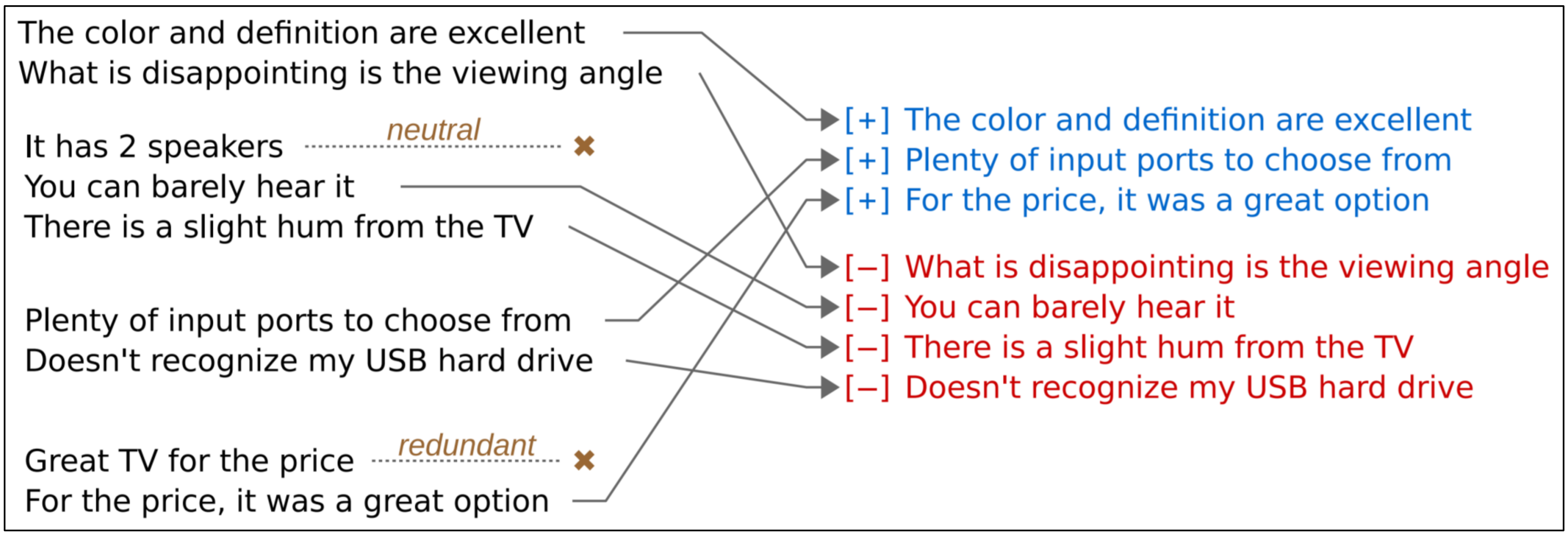
有种活在**诗**里的感**觉**：烟**笼**寒水月**笼**沙，夜泊秦淮近酒家

(褒义隐式情感)

应用实例：情感分析

Main issues:

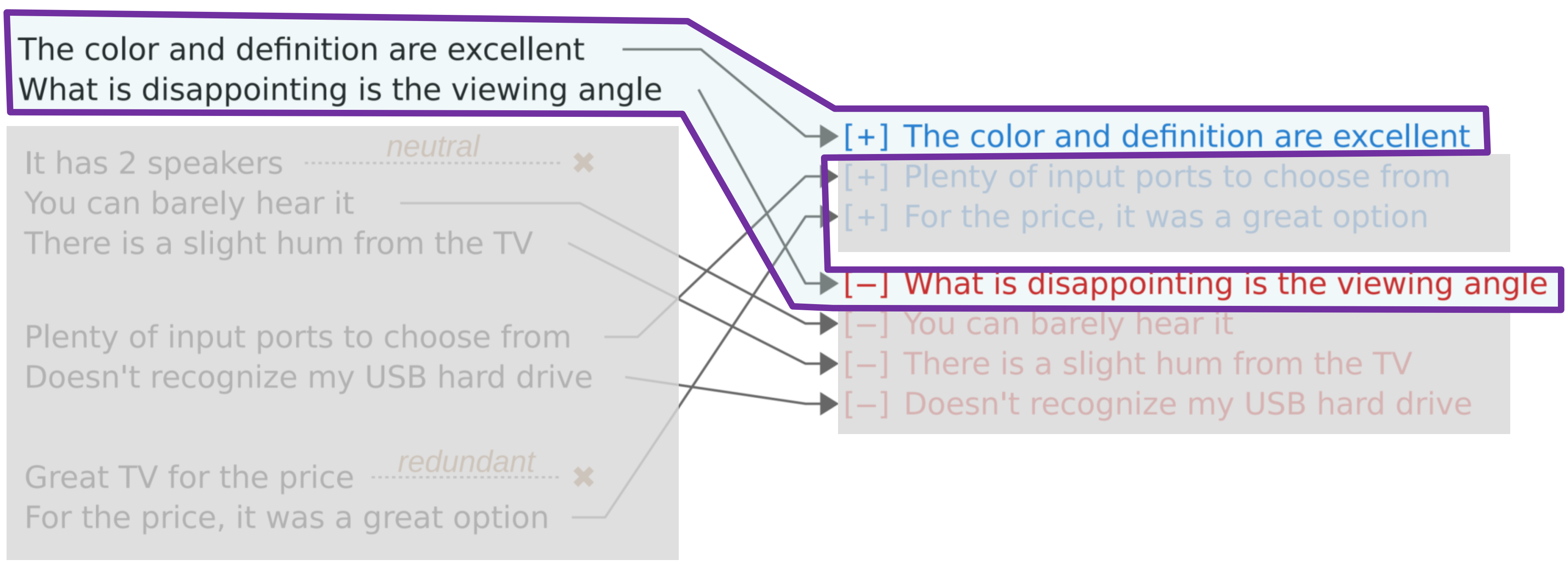
- One sentence contains multiple aspects (1~13) with different sentiments, which may confuse the models；



应用实例：情感分析

Main issues:

- One sentence contains multiple aspects (1~13) with different sentiments, which may confuse the models；



应用实例：情感分析

Main issues:

- Due to complexity of language morphology and syntax, attention mechanisms sometimes cannot distinguish real opinion words

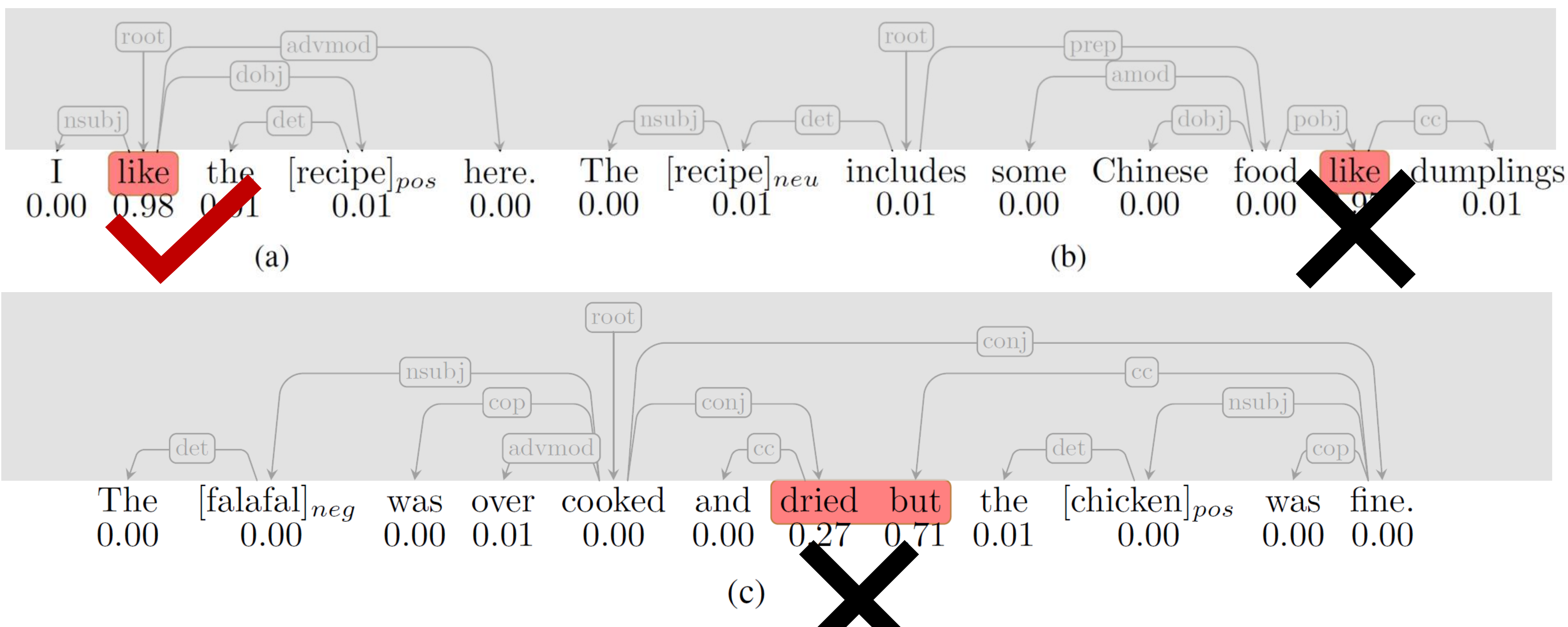


Figure: Three examples of restaurant reviews

应用实例：情感分析

- We propose to exploit sentence syntax information explicitly to improve the effect of the connections.

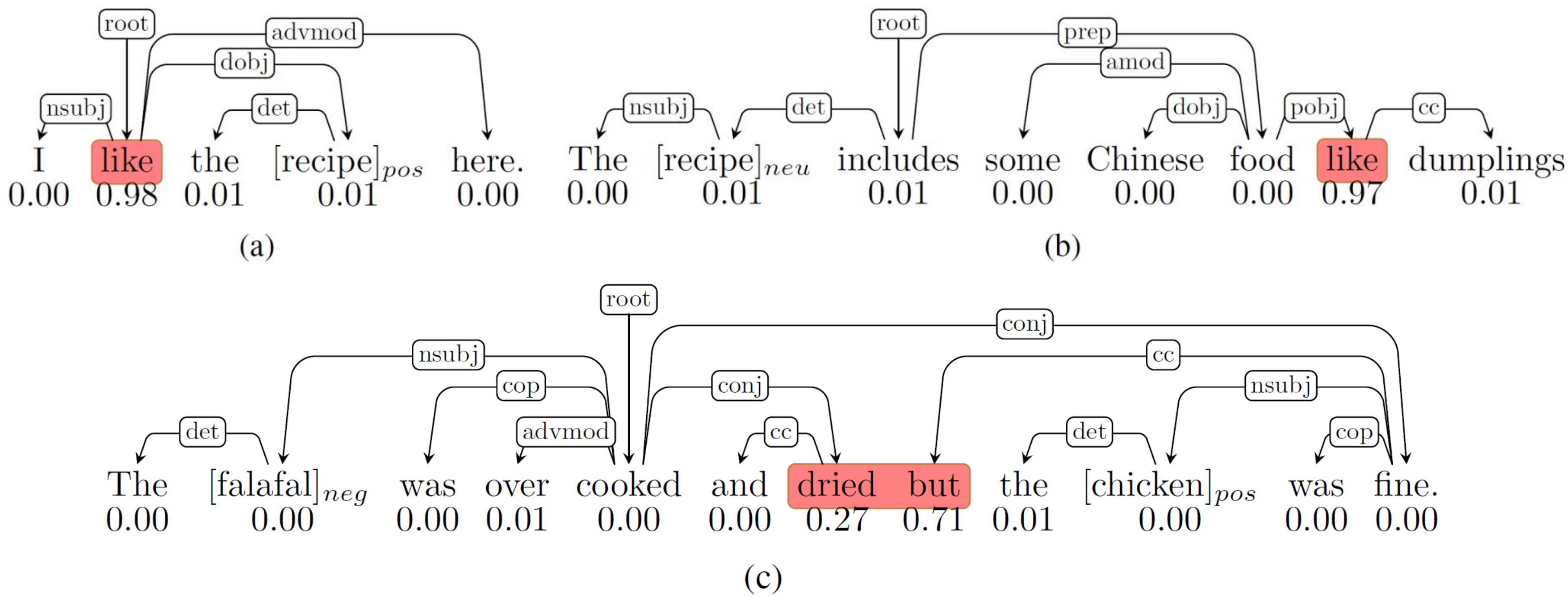


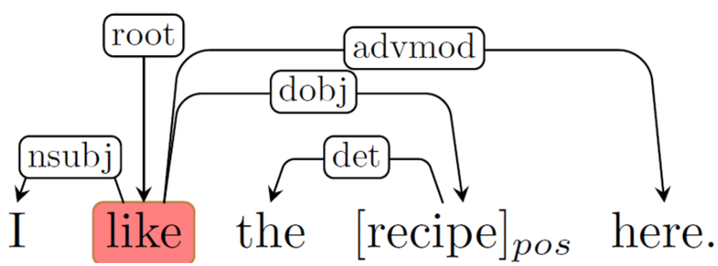
Figure: Three examples of restaurant reviews

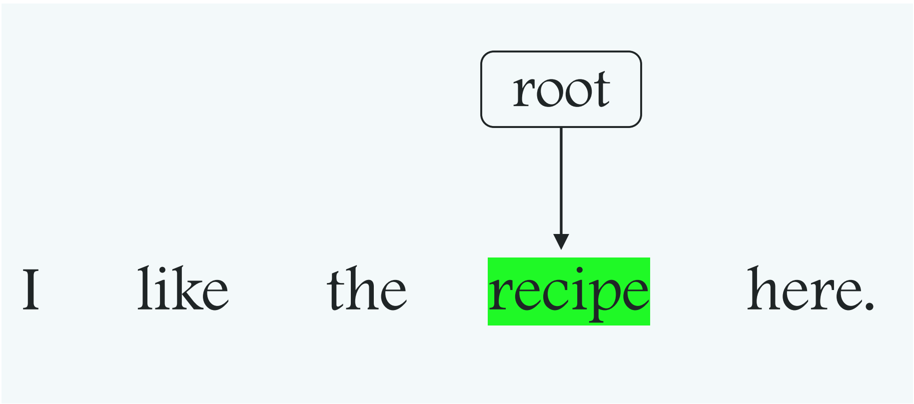
应用实例：情感分析

- We define an aspect-oriented dependency tree structure, reshaped and pruned from an original dependency-based parse tree, to express useful syntax information.

Aspect-Oriented Dependency Tree

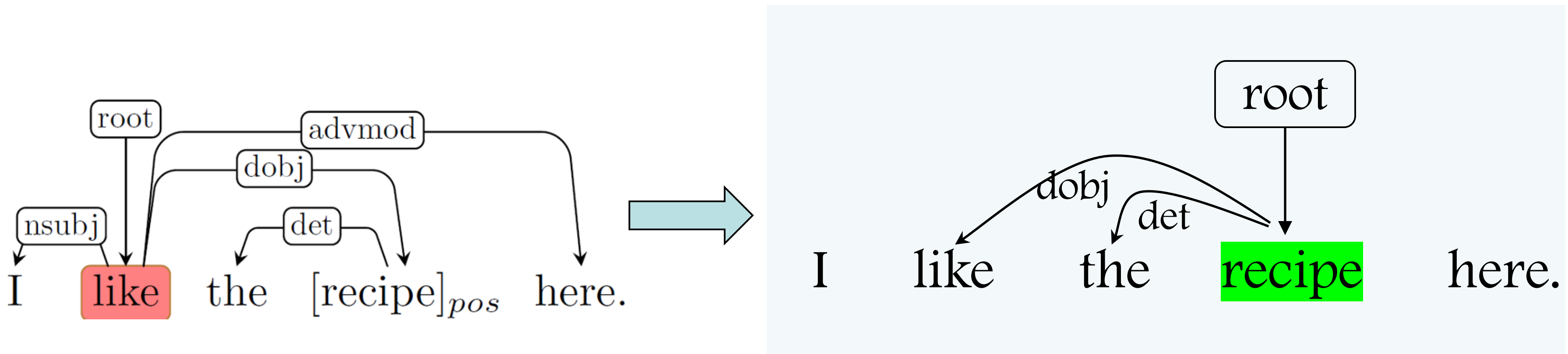
- **Step 1**: First, we place the target aspect at the root.





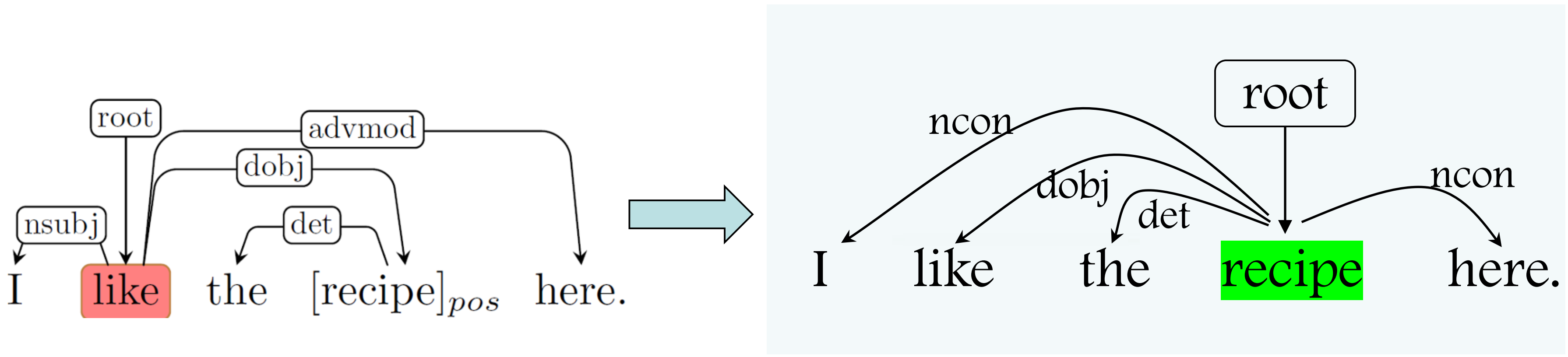
Aspect-Oriented Dependency Tree

- **Step 1**: we place the target aspect at the root.
- **Step 2**: we set the nodes with direct connections to the aspect as the children, for which the original dependency relations remain unchanged.

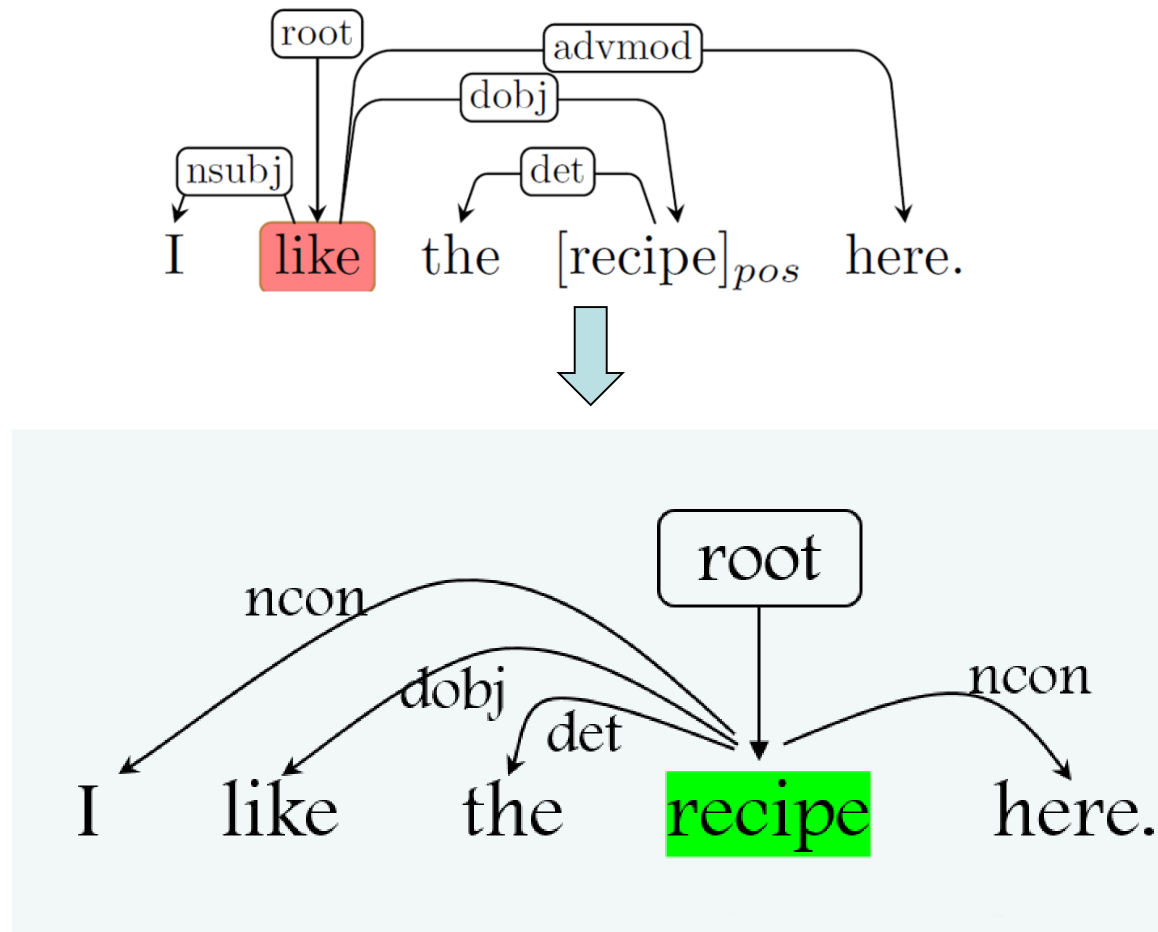


Aspect-Oriented Dependency Tree

- **Step 1**: we place the target aspect at the root.
- **Step 2**: we set the nodes with direct connections to the aspect as the children, for which the original dependency relations remain unchanged.
- **Step 3**: other original dependency relations are discarded, and instead, we put a virtual relation "ncon" (not connected) from each node to the aspect.

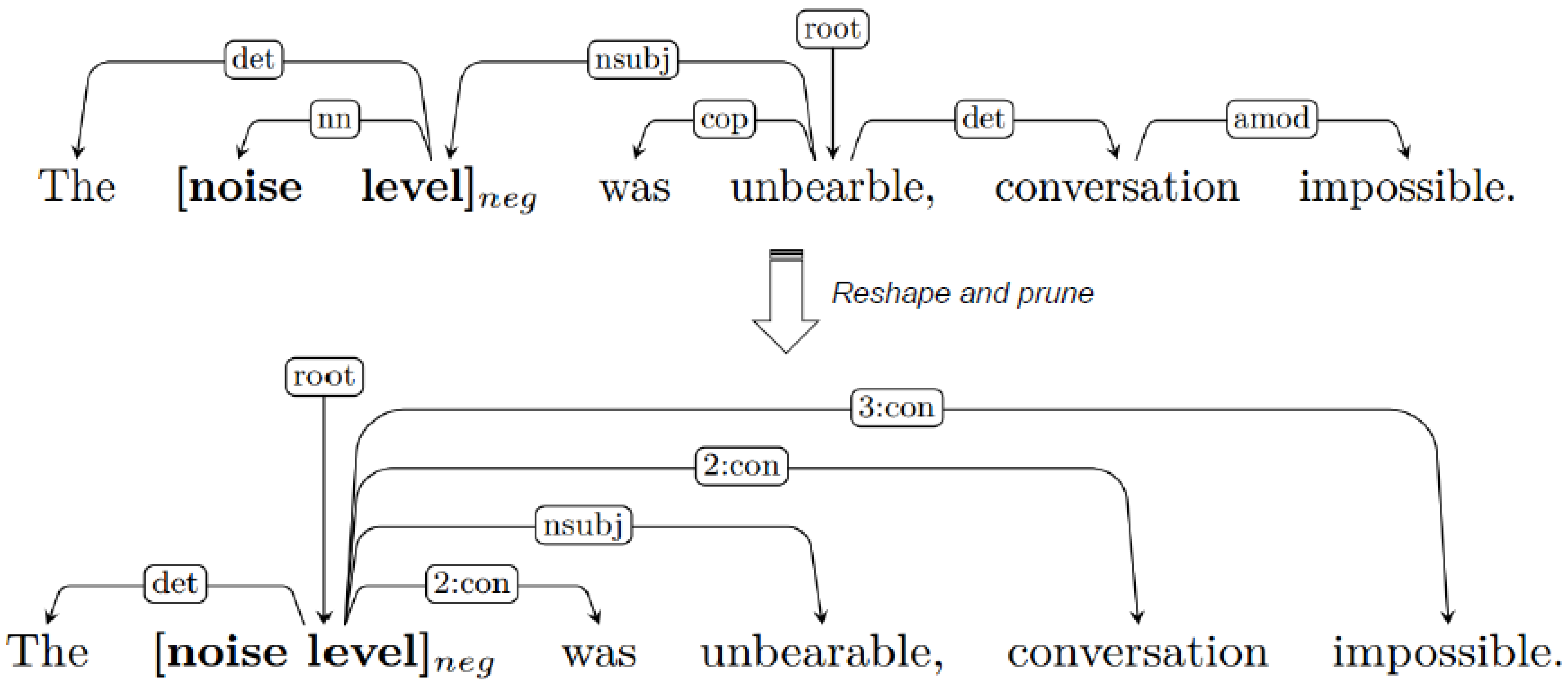


Aspect-Oriented Dependency Tree



aspect-oriented dependency tree

Aspect-Oriented Dependency Tree



aspect-oriented dependency tree

Aspect-Oriented Dependency Tree

- This is inspired by the previous finding that focusing on a small subset of contextual words syntactically close to the aspect is already sufficient.

- All the dependency trees are centered on the target aspect in a unified tree structure, which is convenient for both batch and parallelization operations.

Relational Graph Attention Network

- To encode the new dependency trees for sentiment analysis, we propose a Relational Graph Attention Network (R-GAT) by extending the Graph Attention Network (GAT) [1] to encode graphs with labeled edges.

[1]Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

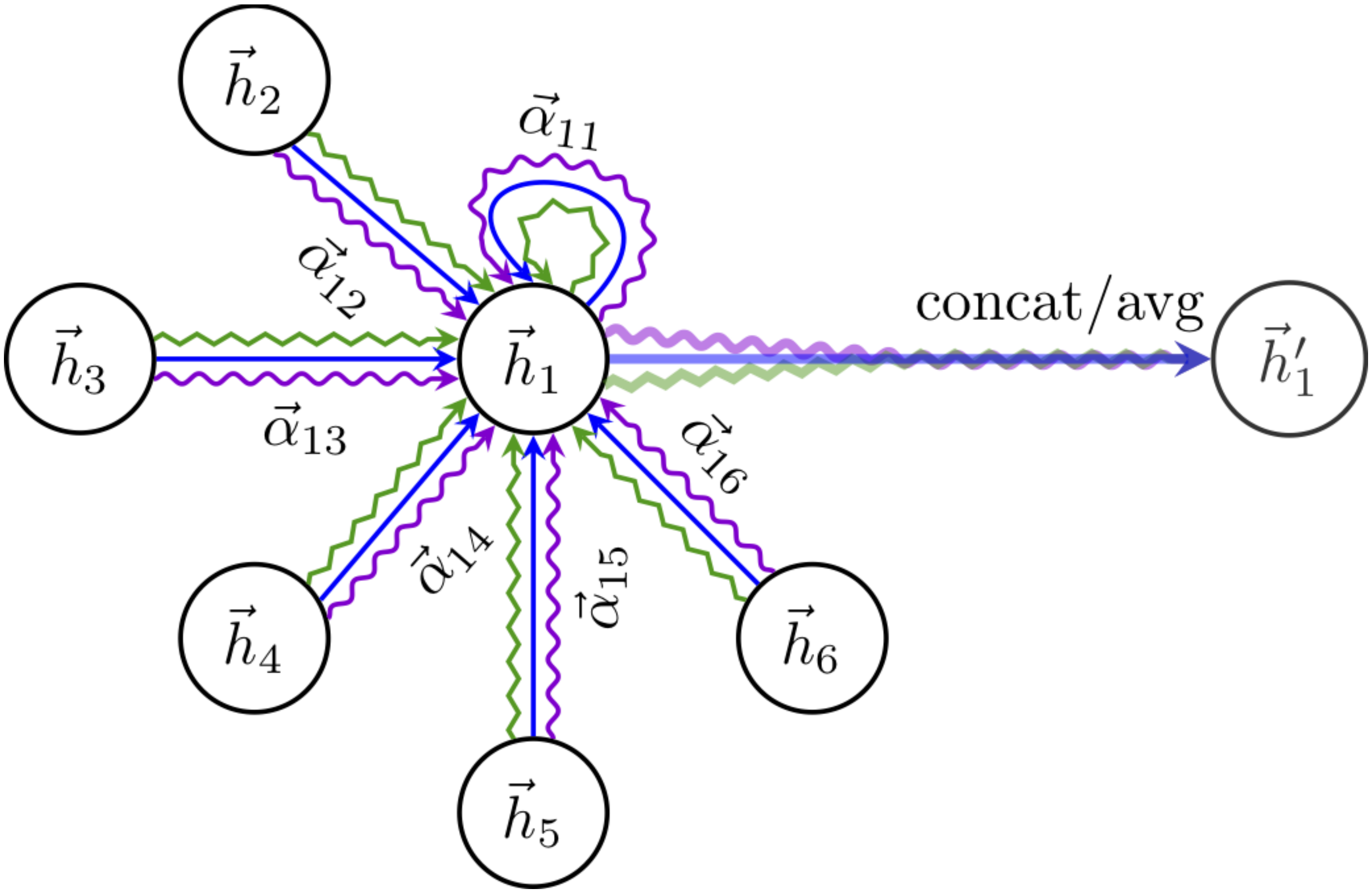
Graph Attention Network

- Dependency tree can be represented by a graph G with n nodes, where each represents a word in the sentence. The edges of G denote the dependency between words.
- The neighborhood nodes of node i can be represented by $\mathcal{N}_i$.
- GAT iteratively updates each node representation (e.g., word embeddings) by aggregating neighborhood node representations using multi-head attention:

$$h_{att_i}^{l+1} = ||_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{lk} W_k^l h_j^l \qquad (1)$$

$$\alpha_{ij}^{lk} = attention(i, j) \qquad (2)$$

Graph Attention Network



Relational Graph Attention Network

- GAT aggregates the representations of neighborhood nodes following the dependency paths. However, this process fails to take dependency relations into consideration, which may lose some important dependency information.
- Motivated by this, we propose to extend the original GAT with additional relational heads. We use these relational heads as relation-wise gates to control information flow from neighborhood nodes.

Relational Graph Attention Network

- Specifically, we first map the dependency relations into vector representations, and then compute a relational head as:

$$h_{rel_i}^{(l+1)} = ||_{m=1}^{M} \sum_{j \in \mathcal{N}_i} \beta_{ij}^{lm} W_m^l h_j^l \qquad (3)$$

$$g_{ij}^{lm} = \sigma(relu(r_{ij} W_{m1} + b_{m1}) W_{m2} + b_{m2}) \qquad (4)$$

$$\beta_{ij} = \frac{exp(g_{ij})}{\sum_{j=1}^{\mathcal{N}_i} exp(g_{ij})} \qquad (5)$$

Relational Graph Attention Network

- Specifically, we first map the dependency relations into vector representations, and then compute a relational head as:

$$h_{rel_i}^{(l+1)} = ||_{m=1}^{M} \sum_{j \in \mathcal{N}_i} \beta_{ij}^{lm} W_m^l h_j^l \tag{3}$$

$$g_{ij}^{lm} = \sigma(relu(r_{ij} W_{m1} + b_{m1}) W_{m2} + b_{m2}) \tag{4}$$

$$\beta_{ij} = \frac{exp(g_{ij})}{\sum_{j=1}^{\mathcal{N}_i} exp(g_{ij})} \tag{5}$$

where r_{ij} represents the embedding for relation between nodes i and j.

Relational Graph Attention Network

- The final Relational Graph Attention Network contains *K* attentional heads and *M* relational heads. The final representation of each node is computed by:

$$x_i^{l+1} = h_{att_i}^{l+1} \,||\, h_{rel_i}^{(l+1)}$$

$$h_i^{l+1} = relu(W_{l+1} x_i^{l+1} + b_{l+1})$$

Relational Graph Attention Network

Category	Method	Restaurant		Laptop		Twitter	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Syntax-aware	LSTM+SynATT	80.45	71.26	72.57	69.13	-	-
	AdaRNN	-	-	-	-	66.30	65.90
	PhraseRNN	66.20	59.32	-	-	-	-
	ASGCN	80.77	72.02	75.55	71.05	72.15	70.40
	CDT	82.30	74.02	77.19	72.99	74.66	73.66
	GAT	78.21	67.17	73.04	68.11	71.67	70.13
	TD-GAT	80.35	76.13	74.13	72.01	72.68	71.15
Attentional	ATAE-LSTM	77.20	-	68.70	-	-	-
	IAN	78.60	-	72.10	-	-	-
	RAM	80.23	70.80	74.49	71.35	69.36	67.30
	MGAN	81.25	71.94	75.39	72.47	72.54	70.81
	LSTM	79.10	69.00	71.22	65.75	69.51	67.98
	BERT	85.62	78.28	77.58	72.38	75.28	74.11
Others	GCAE	77.28	-	69.14	-	-	-
	JCI	-	68.84	-	67.23	-	-
	TNET	80.69	71.27	76.54	71.75	74.90	73.60
Ours	R-GAT	83.30	76.08	77.42	73.76	75.57	73.82
Ours	R-GAT+BERT	**86.60**	**81.35**	**78.21**	**74.07**	**76.15**	**74.88**

Table 2: The overall performance of different methods on the three datasets.

思考题

1. 什么是CFG/PCFG?
2. 简述依存句法与CFG/PCFG的区别?
3. 何为依存句法树的投射性?

Thank you!

权小军　中山大学数据科学与计算机学院