

# 基于图重写的语义分析方法实现

王希豪

指导老师：孙薇薇

EECS, Peking University

*victorwonder@pku.edu.cn*

2020 年 6 月 9 日

- 1 概述
- 2 基于连续词串的句法分析模型
- 3 基于图重写的语义分析模型
- 4 实验
- 5 总结

复现了陈宇非师兄<sup>1</sup>提出的基于同步超边替换文法 (Synchronous Hyperedge Replacement Grammar, 简称 SHRG) 的语义图分析方法, 主要分为以下三个部分:

- 基于连续词串的句法分析器 (Span Parser)
- SHRG 文法自动提取
- 基于 SHRG 的语义分析器

模型通过 Python 和 PyTorch 实现, 代码约 5.5K 行。

---

<sup>1</sup>Y. Chen 等. "Accurate SHRG-Based Semantic Parsing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018-07: 408-418. <https://www.aclweb.org/anthology/P18-1038>.

自然语言处理 (Natural Language Processing, 简称 NLP) 是利用计算机自动分析、处理自然语言数据的研究领域:

- 自然语言理解 (Natural Language Understanding, 简称 NLU)
- 自然语言生成 (Natural Language Generation, 简称 NLG)

# 研究背景

自然语言处理 (Natural Language Processing, 简称 NLP) 是利用计算机自动分析、处理自然语言数据的研究领域:

- 自然语言理解 (Natural Language Understanding, 简称 NLU)
- 自然语言生成 (Natural Language Generation, 简称 NLG)

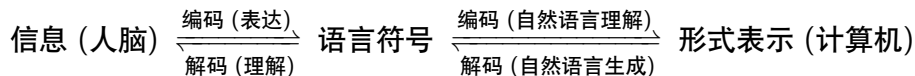


图: 信息、语言符号和形式表示的关系

# 语言的形式表示

- 词嵌入 (Word Embedding)
- 句法分析 (Syntactic Parsing)
- 语义分析 (Semantic Parsing)

- 词嵌入 (Word Embedding)
  - word2vec
  - GloVe
  - ELMo
  - BERT
- 句法分析 (Syntactic Parsing)
- 语义分析 (Semantic Parsing)

- 词嵌入 (Word Embedding)
  - word2vec
  - GloVe
  - ELMo
  - BERT
- 句法分析 (Syntactic Parsing)
  - 成分分析 - 短语结构
  - 依存分析 - 依存结构
- 语义分析 (Semantic Parsing)



# 语言的形式表示

- 词嵌入 (Word Embedding)
  - word2vec
  - GloVe
  - ELMo
  - BERT
- 句法分析 (Syntactic Parsing)
  - 成分分析 - 短语结构
  - 依存分析 - 依存结构
- 语义分析 (Semantic Parsing)
  - 逻辑表达式
  - 语义依存图
  - 概念语义图
    - 基础依存结构 (EDS)
    - 抽象语义表示 (AMR)

- 1 概述
- 2 基于连续词串的句法分析模型**
- 3 基于图重写的语义分析模型
- 4 实验
- 5 总结

# 句法结构 (Syntactic Structure)

*Syntax is the study of the principles and processes by which sentences are constructed in particular languages<sup>2</sup>.*

---

<sup>2</sup>N. Chomsky, D. W. Lightfoot. *Syntactic structures*. Walter de Gruyter, 2002. 

# 句法结构 (Syntactic Structure)

*Syntax is the study of the principles and processes by which sentences are constructed in particular languages<sup>2</sup>.*

两种句法结构:

- 上下文无关文法  $\xrightarrow{\text{成分分析 (Constituent Parsing)}}$  短语结构
- 依存文法  $\xrightarrow{\text{依存分析 (Dependency Parsing)}}$  依存结构

<sup>2</sup>N. Chomsky, D. W. Lightfoot. *Syntactic structures*. Walter de Gruyter, 2002. 

# 句法结构 (Syntactic Structure)

*Syntax is the study of the principles and processes by which sentences are constructed in particular languages<sup>2</sup>.*

两种句法结构:

- 上下文无关文法  $\xrightarrow{\text{成分分析 (Constituent Parsing)}}$  短语结构
- 依存文法  $\xrightarrow{\text{依存分析 (Dependency Parsing)}}$  依存结构

三种分析方法:

- 基于状态转移 (Transition-based) 的方法
- 基于翻译 (Translation-based) 的方法
- 基于因子分解 (Factorization-based) 的方法

<sup>2</sup>N. Chomsky, D. W. Lightfoot. *Syntactic structures*. Walter de Gruyter, 2002. 

# 短语结构 (Phrase Structure)

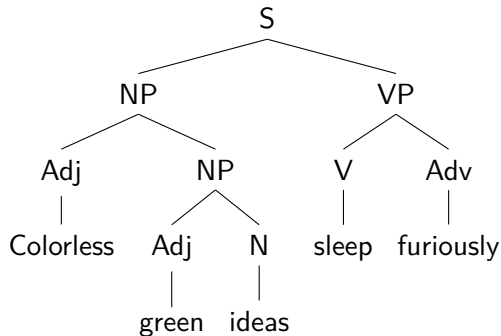


图: “Colorless green ideas sleep furiously” 的短语结构。

# 依存结构 (Dependency Structure)

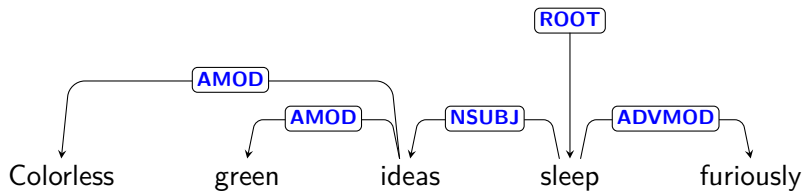


图: “Colorless green ideas sleep furiously” 的依存结构。

# 基于状态转移 (Transition-based) 的方法

步数	词语	栈	动作
1	Colorless		移进
2	green	Colorless	规约
3	green	Adj	移进
4	ideas	Adj green	规约
5	ideas	Adj Adj	移进
6	sleep	Adj Adj ideas	规约
7	sleep	Adj Adj N	规约
8	sleep	Adj NP	规约

表: “Colorless green ideas sleep furiously” 的移进-规约分析过程



# 基于状态转移 (Transition-based) 的方法

步数	词语	栈	动作
9	sleep	NP	移进
10	furiously	NP sleep	规约
11	furiously	NP V	移进
12		NP V furiously	规约
13		NP V Adv	规约
14		NP VP	规约
15		S	结束

表: “Colorless green ideas sleep furiously” 的移进-规约分析过程

# 基于翻译 (Translation-based) 的方法

机器翻译：源语言  $\xrightarrow{\text{翻译模型}}$  目标语言

# 基于翻译 (Translation-based) 的方法

机器翻译: 源语言  $\xrightarrow{\text{翻译模型}}$  目标语言

句法分析: 自然语言  $\xrightarrow{\text{翻译模型}}$  序列化句法树

# 基于翻译 (Translation-based) 的方法

机器翻译: 源语言  $\xrightarrow{\text{翻译模型}}$  目标语言

句法分析: 自然语言  $\xrightarrow{\text{翻译模型}}$  序列化句法树

```
( S ( NP ( Adj Colorless )  
      ( NP ( Adj green )  
            ( N ideas ) ) )  
      ( VP ( V sleep )  
            ( Adv furiously ) ) ) )
```

图: 序列化的 “Colorless green ideas sleep furiously” 短语句法树

# 基于因子分解 (Factorization-based) 的方法<sup>3</sup>

句法分析流程：

- ① 将句法树划分成多个子结构，子结构两两可以组合
- ② 找出句子所有可能的子结构，得到每个子结构的特征
- ③ 构建打分模型，根据每个子结构的特征打分
- ④ 选择分数最高的一些子结构，组合得到合法的句法树

---

<sup>3</sup>R. McDonald, F. Pereira. *Discriminative learning and spanning tree algorithms for dependency parsing*. University of Pennsylvania, 2006.

# 基于因子分解 (Factorization-based) 的方法

句法分析流程：

- ① 将句法树划分成多个子结构，子结构两两可以组合
- ② 找出句子所有可能的子结构，得到每个子结构的特征
- ③ 构建打分模型，根据每个子结构的特征打分
- ④ 选择分数最高的一些子结构，组合得到合法的句法树

# 连续词串 (Span)

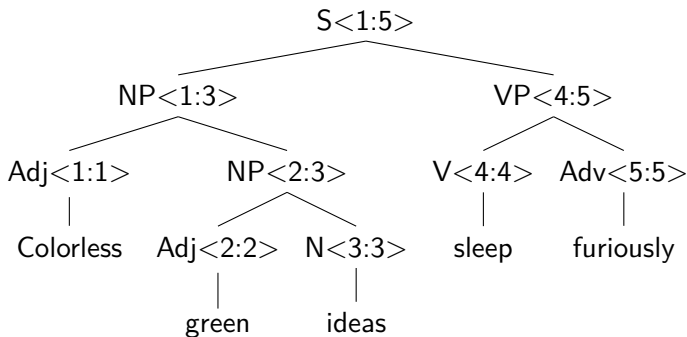
## 定义 (Span)

由句子中第  $i$  个位置到第  $j$  个位置的  $j-i+1$  个词语按照在原句中的顺序排列得到的子序列称为连续词串 (Span)，用符号  $\langle i:j \rangle$  表示。

# 连续词串 (Span)

## 定义 (Span)

由句子中第  $i$  个位置到第  $j$  个位置的  $j-i+1$  个词语按照在原句中的顺序排列得到的子序列称为连续词串 (Span)，用符号  $\langle i:j \rangle$  表示。



图：“Colorless green ideas sleep furiously”的短语结构与连续词串。



# 基于因子分解 (Factorization-based) 的方法

句法分析流程：

- ① 将句法树划分成多个子结构，子结构两两可以组合
- ② 找出句子所有可能的子结构，得到每个子结构的特征
- ③ 构建打分模型，根据每个子结构的特征打分
- ④ 选择分数最高的一些子结构，组合得到合法的句法树

# 循环神经网络 (RNN)

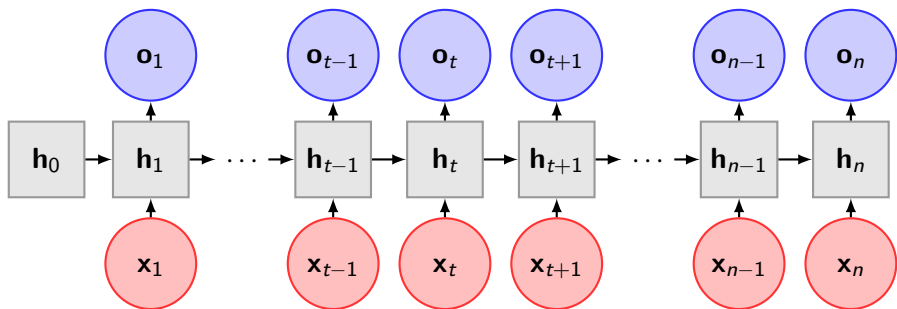


图: 循环神经网络模型图示

# 长短期记忆网络 (LSTM)

LSTM 隐状态向量计算过程<sup>4</sup>:

$$f_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f)$$

$$i_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i)$$

$$o_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c)$$

$$\mathbf{c}_t = f_t \circ \mathbf{c}_{t-1} + i_t \circ \tilde{\mathbf{c}}_t$$

$$\mathbf{h}_t = o_t \circ \tanh(\mathbf{c}_t)$$

---

<sup>4</sup>S. Hochreiter, J. Schmidhuber. "Long Short-Term Memory". *Neural Computation*, 1997, 9(8): 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.

<sup>5</sup>J. Cross, L. Huang. "Span-Based Constituency Parsing with a Structure-Label System and Provably Optimal Dynamic Oracles". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016-11: 1–11. <https://www.aclweb.org/anthology/D16-1001>.

# 长短期记忆网络 (LSTM)

LSTM 隐状态向量计算过程<sup>4</sup>:

$$f_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f)$$

$$i_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i)$$

$$o_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c)$$

$$\mathbf{c}_t = f_t \circ \mathbf{c}_{t-1} + i_t \circ \tilde{\mathbf{c}}_t$$

$$\mathbf{h}_t = o_t \circ \tanh(\mathbf{c}_t)$$

LSTM-Minus<sup>5</sup>:

- $\text{FEATURE}(i, j) = (\mathbf{f}_j - \mathbf{f}_{i-1}, \mathbf{b}_j - \mathbf{b}_{j+1})$

<sup>4</sup>S. Hochreiter, J. Schmidhuber. "Long Short-Term Memory". *Neural Computation*, 1997, 9(8): 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.

<sup>5</sup>J. Cross, L. Huang. "Span-Based Constituency Parsing with a Structure-Label System and Provably Optimal Dynamic Oracles". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016-11: 1–11. <https://www.aclweb.org/anthology/D16-1001>.

# 基于因子分解 (Factorization-based) 的方法

句法分析流程：

- ① 将句法树划分成多个子结构，子结构两两可以组合
- ② 找出句子所有可能的子结构，得到每个子结构的特征
- ③ **构建打分模型，根据每个子结构的特征打分**
- ④ 选择分数最高的一些子结构，组合得到合法的句法树

# 多层感知机 (MLP)

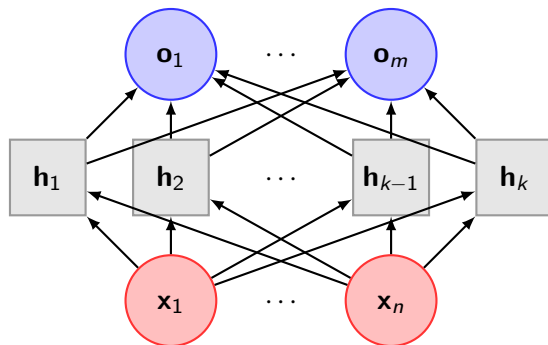


图: 多层感知机模型图示

# 基于因子分解 (Factorization-based) 的方法

句法分析流程：

- ① 将句法树划分成多个子结构，子结构两两可以组合
- ② 找出句子所有可能的子结构，得到每个子结构的特征
- ③ 构建打分模型，根据每个子结构的特征打分
- ④ **选择分数最高的一些子结构，组合得到合法的句法树**

---

## Algorithm 1 CYK 算法

---

**Require:** 二维数组 SCORE, 序列长度  $n$

**Ensure:** 最大分数  $C[1, n]$ , 区间划分方式 BACK

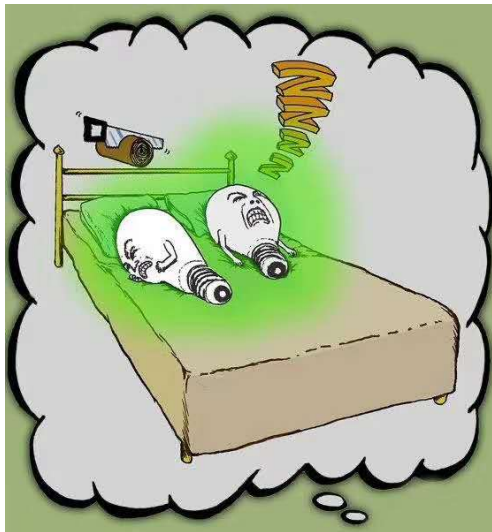
```
1: for  $i = 1 \rightarrow n$  do
2:    $C[i, j] = \text{SCORE}[i, j]$ 
3: for  $i = n \rightarrow 1$  do
4:   for  $j = i \rightarrow n$  do
5:     for  $k = i \rightarrow j - 1$  do
6:       if  $C[i, j] < C[i, k] + C[k + 1, j] + \text{SCORE}[i, j]$  then
7:          $C[i, j] = C[i, k] + C[k + 1, j] + \text{SCORE}[i, j]$ 
8:          $\text{BACK}[i, j] = k$ 
9: return  $C[1, n]$ , BACK
```

---



- 1 概述
- 2 基于连续词串的句法分析模型
- 3 基于图重写的语义分析模型**
- 4 实验
- 5 总结

Colorless green ideas sleep furiously.



图：漫画：“Colorless green ideas sleep furiously.”

## 定义 (语义图)

语义图是利用“图”的数学模型来表示句子语义的表示方法：

- 语义图的每个结点表示一个词语
- 语义图的每条表表示一对词之间的二元关系

# 语义图与概念语义图

## 定义 (语义图)

语义图是利用“图”的数学模型来表示句子语义的表示方法:

- 语义图的每个结点表示一个词语
- 语义图的每条表表示一对词之间的二元关系

## 定义 (概念语义图)

概念语义图的结点表示一个概念 (Concept):

- 如果概念和句子的某个词语对应, 则称为 surface concept
- 如果概念表示句子中提取的语义, 则称为 abstract concept

# 语义图与概念语义图

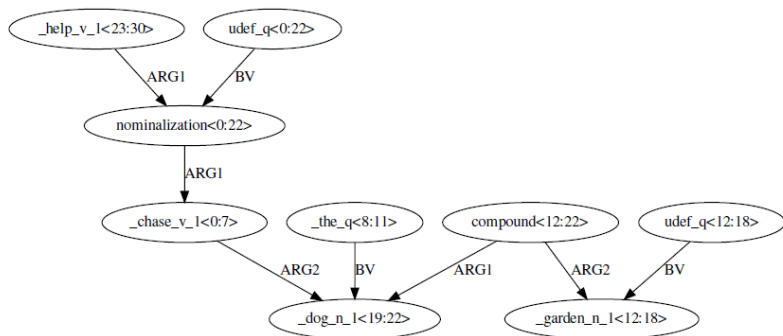
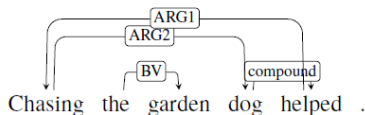


图: 英语句子 “Chasing the garden dog helped” 的语义图和概念语义图。

# 上下文无关文法 (CFG)

## 定义 (Context-free Grammar)

一个上下文无关文法  $G$  是四元组  $\langle N, T, S, P \rangle$ , 其中:

- $N$  是非终结符集合
- $T$  是终结符集合
- $S$  起始符号
- $P$  是规则集合, 每条规则都是  $A \rightarrow \alpha, A \in N, \alpha \in (N \cup T)^*$

# 上下文无关文法 (CFG)

## 定义 (Context-free Grammar)

一个上下文无关文法  $G$  是四元组  $\langle N, T, S, P \rangle$ , 其中:

- $N$  是非终结符集合
- $T$  是终结符集合
- $S$  起始符号
- $P$  是规则集合, 每条规则都是  $A \rightarrow \alpha, A \in N, \alpha \in (N \cup T)^*$

## 定义 (Chomsky 范式)

如果 CFG  $G$  的每条规则属于下面两种类型:

- $A \rightarrow BC, A, B, C \in N$
- $A \rightarrow \alpha, A \in N, \alpha \in T$

则称该 CFG 是 Chomsky 范式

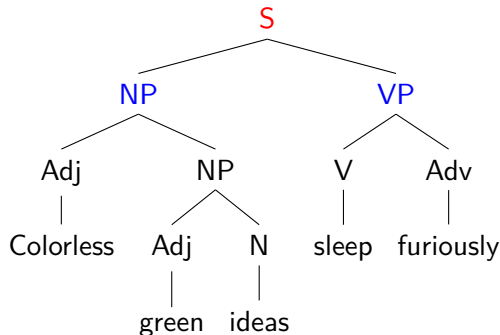
# CFG 与短语结构

- 短语分析是基于“句子由特定 CFG 生成”的假设进行的
- 短语句法树每个内部结点对应一条 CFG 规则



# CFG 与短语结构

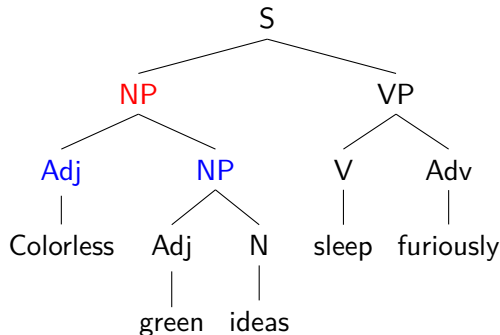
- 短语分析是基于“句子由特定 CFG 生成”的假设进行的
- 短语句法树每个内部结点对应一条 CFG 规则



图：“Colorless green ideas sleep furiously”的短语结构。

# CFG 与短语结构

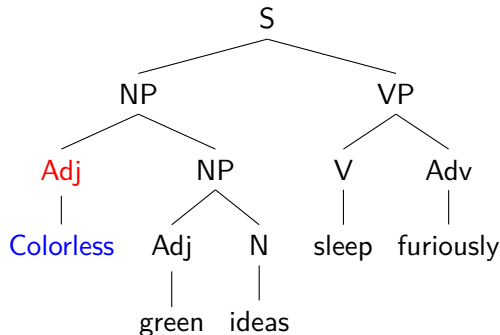
- 短语分析是基于“句子由特定 CFG 生成”的假设进行的
- 短语句法树每个内部结点对应一条 CFG 规则



图：“Colorless green ideas sleep furiously”的短语结构。

# CFG 与短语结构

- 短语分析是基于“句子由特定 CFG 生成”的假设进行的
- 短语句法树每个内部结点对应一条 CFG 规则



图：“Colorless green ideas sleep furiously”的短语结构。

# 超边替换文法 (HRG)

## 定义 (超边)

超边 (Hyperedge) 是边概念的扩展, 可以只有一个顶点或有三个以上顶点。指定了顶点顺序的超边称为有向超边。

## 定义 (超图)

含有超边的图就是超图 (Hyper-graph)。本文模型只考虑边标记的、点无标记的有向超图。

# 超边替换文法 (HRG)

## 定义 (Hyperedge Replacement Grammar)

一个超边替换文法  $G$  是四元组  $\langle N, T, S, P \rangle$ , 其中:

- $N$  是非终结符 (边) 的集合
- $T$  是终结符 (边) 的集合
- $S$  是起始符号
- $P$  是规则集合, 每条规则都是  $A \rightarrow \alpha$ ,  $A \in N$  是一条超边,  $\alpha$  是一张由  $N$  和  $T$  中超边组成的超图

# 超边替换文法 (HRG)

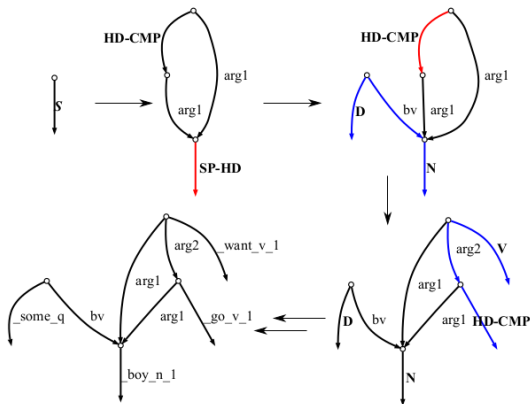


图: 超边替换文法生成 “Some boys want to go.” 语义图的过程图。图中红色的超边表示在下一步中要被替换为蓝色超边构成的子图。

四种方法：

- 基于状态转移的方法
- 基于翻译的方法
- 基于因子分解的方法
- **基于语义组合的方法**
  - 句子的语义图看做是短语语义图的组合

## 定义 (推导树)

用于指导超边替换文法生成语义图的、每个内部节点度数为 2 的二叉树称为推导树 (Derivation Tree):

- 自顶向下看, 推导树说明了特定 CFG 生成句子的过程。
- 自底向上看, 推导树展示了短语语义组合的顺序。



# 同步超边替换文法 (SHRG)

## 定义 (推导树)

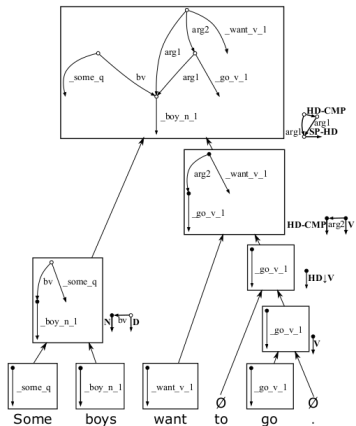
用于指导超边替换文法生成语义图的、每个内部节点度数为 2 的二叉树称为推导树 (Derivation Tree):

- 自顶向下看, 推导树说明了特定 CFG 生成句子的过程。
- 自底向上看, 推导树展示了短语语义组合的顺序。

## 定义 (Synchronous Hyperedge Replacement Grammar)

根据推导树, 自底向上进行语义组合得到语义图的方法。

# 同步超边替换文法 (SHRG)



图：“Some boys want to go.” 语义图生成过程

## 定义 (内点与外点)

对于超图  $H$  的一张子图  $G$ , 所有顶点可以分为内点与外点:

- 只与子图  $G$  中的边相连的顶点称为内点 (internal node)
- 与图  $H - G$  中的边相连的顶点称为外点 (external node)

## 定义 (内点与外点)

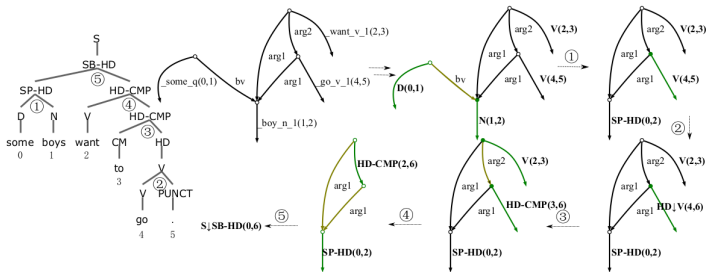
对于超图  $H$  的一张子图  $G$ , 所有顶点可以分为内点与外点:

- 只与子图  $G$  中的边相连的顶点称为内点 (internal node)
- 与图  $H - G$  中的边相连的顶点称为外点 (external node)

自底向上文法规则自动抽取流程:

- ① 后序遍历推导树的每个结点
- ② 根据句法树结点对应的 span, 在语义图中寻找对应的子图
- ③ 将找到的子图缩成一条超边, 子图的外点就是超边的顶点

# 文法规则自动抽取



	①	②	③	④	⑤
Shared LHS	SP-HD	HD↓V	HD-CMP	HD-CMP	S↓SP-HD
RHS (syntax)	D + N	V + PUNCT	CM + HD↓V	V + HD-CMP	SP-HD + HD-CMP
RHS (semantics)	$N \downarrow \text{bv} \downarrow D$	$V \downarrow$	$HD \downarrow V$	$HD-CMP \downarrow \text{arg2} \downarrow V$	$HD-CMP \downarrow \text{arg1} \downarrow SP-HD$

图: 文法规则自动抽取过程

- ① 后序遍历推导树的每个结点
- ② 为每个结点确定对应的语义组合规则
- ③ 自底向上组合得到语义图

- ① 后序遍历推导树的每个结点
- ② 为每个结点确定对应的语义组合规则
- ③ 自底向上组合得到语义图

问题一：结点无对应语义规则

- ① 后序遍历推导树的每个结点
- ② 为每个结点确定对应的语义组合规则
- ③ 自底向上组合得到语义图

## 问题一：结点无对应语义规则

- 结点为叶子结点，构造由一条超边组成的子图，超边标签为词语对应的 surface concept
- 结点为内部结点，对应一条  $n$  个结点的超边，构造  $n+1$  个点组成的子图，从中间点往每个外点连边



- ① 后序遍历推导树的每个结点
- ② 为每个结点确定对应的语义组合规则
- ③ 自底向上组合得到语义图

问题一：结点无对应语义规则

问题二：结点对应多条语义规则

- ① 后序遍历推导树的每个结点
- ② 为每个结点确定对应的语义组合规则
- ③ 自底向上组合得到语义图

**问题一：** 结点无对应语义规则

**问题二：** 结点对应多条语义规则

- 利用神经网络对每条规则进行打分
- 搜索得到分数和最高的语义图

- 1 概述
- 2 基于连续词串的句法分析模型
- 3 基于图重写的语义分析模型
- 4 实验**
- 5 总结

数据集为 DeepBank 1.1 版<sup>6</sup>  
wsj20 作为开发集, wsj21 作为测试集

---

<sup>6</sup>M. E. Peters 等. “Deep contextualized word representations”. *arXiv preprint arXiv:1802.05365*, 2018.

<sup>7</sup>[www.coli.uni-saarland.de/~yzhang/files/jigsaw.jar](http://www.coli.uni-saarland.de/~yzhang/files/jigsaw.jar)

数据集为 DeepBank 1.1 版<sup>6</sup>  
wsj20 作为开发集, wsj21 作为测试集

数据预处理:

- ① 利用 jigsaw<sup>7</sup> 工具分离标点符号
- ② 简化 DeepBank 句法树标签, 如 HD-CMP\_U\_C 只保留 HD-CMP
- ③ 压缩 DeepBank 句法树所有的单链, 以适应 SHRG 算法
- ④ 抽取语义规则, 在结点上添加语义信息 (对应语义图的外点数目)

<sup>6</sup>M. E. Peters 等. "Deep contextualized word representations". *arXiv preprint arXiv:1802.05365*, 2018.

<sup>7</sup>[www.coli.uni-saarland.de/~yzhang/files/jigsaw.jar](http://www.coli.uni-saarland.de/~yzhang/files/jigsaw.jar)

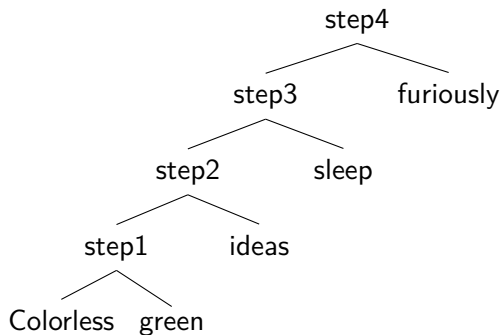
- 使用 evalb<sup>8</sup>工具评价句法分析的结果
- 使用 edm<sup>9</sup>指标评价语义分析的结果
- 比较不同结构二叉树作为推导树对语义分析的影响
  - DeepBank 句法树
  - 顺序组合句法树

---

<sup>8</sup><https://nlp.cs.nyu.edu/evalb/>

<sup>9</sup>R. Dridan, S. Oepen. "Parser Evaluation Using Elementary Dependency Matching". In: *Proceedings of the 12th International Conference on Parsing Technologies*. Dublin, Ireland: Association for Computational Linguistics, 2011-10: 225-230. <https://www.aclweb.org/anthology/W11-2927>.

# 顺序组合句法树



图：顺序组合句法树

# 实验结果

	无语义信息		有语义信息	
	括号准确率	标签准确率	括号准确率	标签准确率
DeepBank	85.73	92.99	85.67	91.95
顺序结构	89.10	95.29	45.46	93.94

表: 句法分析结果



# 实验结果

	无语义信息		有语义信息	
	括号准确率	标签准确率	括号准确率	标签准确率
DeepBank	85.73	92.99	85.67	91.95
顺序结构	89.10	95.29	45.46	93.94

表: 句法分析结果

		Precision	Recall	F-score
DeepBank	顶点	83.68	83.13	83.41
	边	67.75	66.90	67.32
	综合	75.71	74.98	75.34
顺序结构	顶点	61.48	58.64	60.01
	边	14.52	14.68	14.60
	综合	37.21	36.55	36.88

表: 语义分析结果

- 1 概述
- 2 基于连续词串的句法分析模型
- 3 基于图重写的语义分析模型
- 4 实验
- 5 总结**





# 存在的问题

- ① 特征提取时只使用了较为简单的 LSTM-Minus 模型，可以改为 Transformer 等复杂模型提高特征提取的准确度。
- ② 句法分析器对于标签的预测略差，可以训练更好的分类器进行分类。
- ③ 语义规则抽取时没有考虑到外部结点的顺序，部分结构相同的语义规则被判断成了不同的规则。影响到了实验的效果。
- ④ 语义分析器对于语义图的边预测效果较差，需要继续训练改进。

- ① 继续调参训练 Span Parser 和 HRG Parser 至论文水准
- ② 语义规则的正则化

# 感谢聆听！

# 参考文献 I

-  Y. Chen, W. Sun and X. Wan. “*Accurate SHRГ-Based Semantic Parsing*”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018-07: 408–418.  
<https://www.aclweb.org/anthology/P18-1038>.
-  N. Chomsky and D. W. Lightfoot. *Syntactic structures*. Walter de Gruyter, **2002**.
-  R. McDonald and F. Pereira. *Discriminative learning and spanning tree algorithms for dependency parsing*. University of Pennsylvania, **2006**.
-  S. Hochreiter and J. Schmidhuber. “*Long Short-Term Memory*”. *Neural Computation*, **1997**, 9(8): 1735–1780.  
<https://doi.org/10.1162/neco.1997.9.8.1735>.

## 参考文献 II

-  J. Cross and L. Huang. “*Span-Based Constituency Parsing with a Structure-Label System and Provably Optimal Dynamic Oracles*”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016-11: 1–11.  
<https://www.aclweb.org/anthology/D16-1001>.
-  M. E. Peters, M. Neumann, M. Iyyer et al. “*Deep contextualized word representations*”. *arXiv preprint arXiv:1802.05365*, **2018**.
-  R. Drìdan and S. Oepen. “*Parser Evaluation Using Elementary Dependency Matching*”. In: *Proceedings of the 12th International Conference on Parsing Technologies*. Dublin, Ireland: Association for Computational Linguistics, 2011-10: 225–230.  
<https://www.aclweb.org/anthology/W11-2927>.