Автоматический синтаксический анализ

Екатерина Владимировна Еникеева

2021

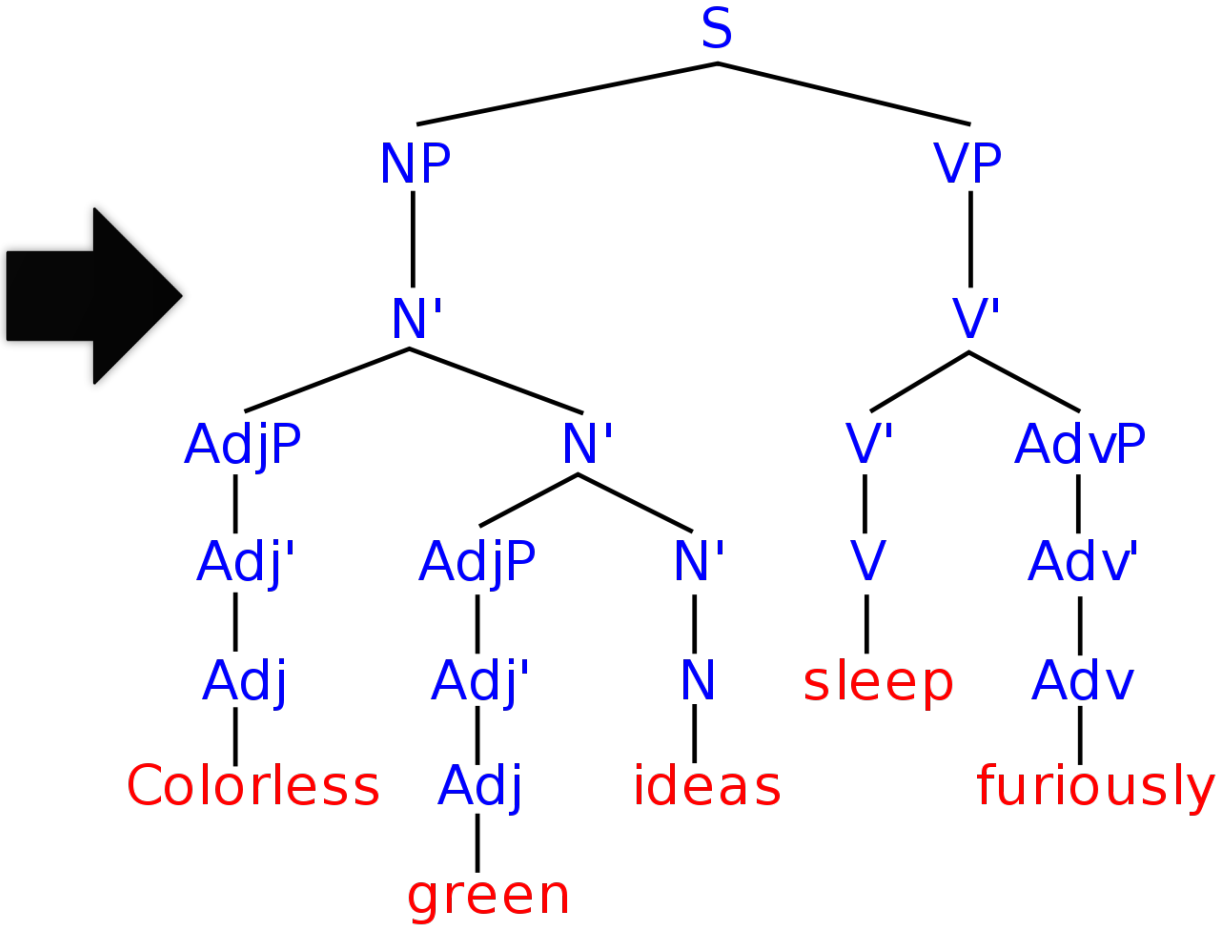
Автоматическая обработка естественного языка, лекция 5

План

1. Зачем нужен синтаксический анализ?
2. Терминология
3. Структура составляющих
4. Данные и их представление
5. КС-грамматики
6. Более сложные грамматики

Основная цель

Colorless green ideas sleep furiously.



Задачи

Для естественных языков:

- предварительный этап для семантического анализа
- извлечение фактов / диалоговые фреймы
- разрешение анафоры, кореференции

Другое:

- языки программирования

Пример узкой задачи

Привести фразу к начальной форме:

Примеры использования

все | глобальное потепление | прочие переводы

Global warming!
Глобальное потепление!

They're gonna stop global warming.
Они собираются остановить глобальное потепление.

It is already hit by global warming.
Глобальное потепление уже ударило по ней.

Maybe it's that global warming stuff, she thought.
Может, все дело в глобальном потеплении? подумала Сэмми.

The Polar Orbiting Density Scanner (PODS) was designed to help measure the effects of global warming.
Полярный орбитальный сканер плотности был создан для того, чтобы тщательнее контролировать последствия глобального потепления.

Oh, damn that global warming!
— Ох, чертово глобальное потепление!

In turn, they exacerbate global warming.
В свою очередь, они обостряют процесс глобального потепления.

Нормализация фразы

- генерация форм по данной форме / лемме
- оценка n-граммной LM
- оптимизация поиска: beam search

Когда нужен синтаксис?

хороший день vs *хорошего дня*
два котенка vs *двух котят*

Чанкинг

- не всегда нужно строить целое дерево

Например: выделить только NP

Adj + Noun

(Adj) + Noun*

Noun + Noun gent

…

Терминология

- парсинг / parsing – парсер / parser
  - ЕЯ: преобразование в синтаксическое представление
  - языки разметки
  - ЯП: преобразование кода в представление, которое затем обрабатывает компилятор
- parse tree / синтаксический разбор — результат парсинга

Терминология 2

Parse tree может выглядеть как

- Структура составляющих – constituent structure
  - объект phrase structure grammars
- Структура / дерево зависимостей – dependency tree / structure / parse
  - объект dependency grammars

Демо: Stanford Parser

Stanford Parser

http://nlp.stanford.edu:8080/parser/index.jsp

Please enter a sentence to be parsed:

My dog also likes eating sausage.

Language: English Sample Sentence Parse

Your query

My dog also likes eating sausage.

Tagging

```
My/PRP$  dog/NN  also/RB  likes/VBZ  eating/VBG  sausage/NN  ./.
```

Parse

```
(ROOT
  (S
    (NP (PRP$ My) (NN dog))
    (ADVP (RB also))
    (VP (VBZ likes)
      (S
        (VP (VBG eating)
          (NP (NN sausage)))))
    (. .)))
```

Universal dependencies

```
nmod:poss(dog-2, My-1)
nsubj(likes-4, dog-2)
advmod(likes-4, also-3)
root(ROOT-0, likes-4)
xcomp(likes-4, eating-5)
obj(eating-5, sausage-6)
```

Universal dependencies, enhanced

```
nmod:poss(dog-2, My-1)
nsubj(likes-4, dog-2)
advmod(likes-4, also-3)
root(ROOT-0, likes-4)
xcomp(likes-4, eating-5)
obj(eating-5, sausage-6)
```

Структура составляющих

Составляющая— независимая синтаксическая единица:

- можно перемещать в пределах предложения:
  - *John talked [to the children] [about rules].*
  - *John talked [about rules] [to the children].*
  - **John talked rules to the children about.*
- можно заменять на грамматически похожие:
  - *I sat [on the box / on top of the box / in front of you].*

Структура составляющих

NP – Noun Phrase – именная группа
VP – Verb Phrase – глагольная группа
PP – Prepositional Phrase – предложная группа
и т.д.

- расположены линейно
- вкладываются друг в друга
- скобочный формат (bracketed notation) / дерево

[[[Colorless [green ideas]] [sleep furiously]]

Разметка составляющих

Penn Treebank Constituent Tags

Penn Treebank:

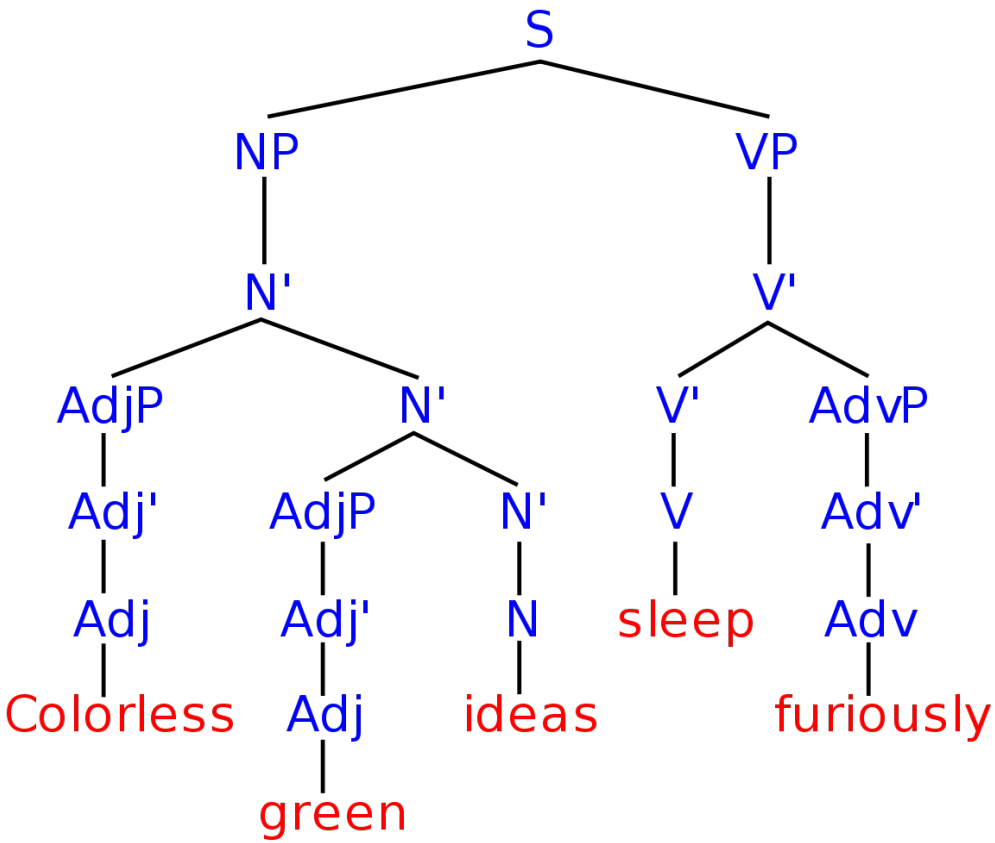
- Brown Corpus
- 1M words from WSJ
- ATIS - Air Traffic Information System

Формальная грамматика

- Словарь V
  - Конечное множество нетерминалов V_N включает *start symbol (S)*
  - Конечное множество терминалов V_T (+ пустой символ ε)
- Грамматика
  - Конечное множество правил (продукций) P

➢ Описывает, какие последовательности допустимы в данном языке

(Не)терминалы



Пример 1

$$V_N = \{S\}$$

$$V_T = \{a, b, \varepsilon\}$$

$$S \to aSb$$

$$S \to \varepsilon$$

Какие последовательности допустимы в данном языке?

Пример 1

$$V_N = \{S\}$$

$$V_T = \{a, b, \varepsilon\}$$

$$S \to aSb$$

$$S \to \varepsilon$$

Какие последовательности допустимы в данном языке?

$ab, aabb, aaabbb$ ….

$a^n b^n$

Иерархия Хомского

Пусть V^* — множество всех строк; V^+ — непустых строк.

- тип 0 – неограниченные грамматики

  $\alpha \to \beta$, где $\boldsymbol{\alpha}$ – любая последовательность, содержащая нетерминал, $\boldsymbol{\beta}$ – любая последовательность
- тип 1 – контекстно-зависимые грамматики

  $\alpha A\beta \to \alpha\gamma\beta : \alpha, \beta \in V^*, A \in V_N \qquad \alpha \to \beta : \ 1 \le |\alpha| \le |\beta|$
- тип 2 – **контекстно-свободные грамматики**

  $A \to \beta : \beta \in V^+ (\beta \in V^*)$
- тип 3 – регулярные грамматики

  $A \to B\gamma \ / \ A \to \gamma : \gamma \in V_T^*, A, B \in V_N$

Пример 2

$$S \to \langle NP \rangle \langle VP \rangle$$
$$\langle NP \rangle \to A \langle NP \rangle$$
$$\langle NP \rangle \to N$$
$$\langle VP \rangle \to V \langle NP \rangle$$
$$\langle VP \rangle \to V$$
$$N \to ideas \mid linguists$$
$$V \to generate \mid hate \mid eat$$
$$A \to great \mid green$$

Great linguists generate green ideas.

Linguists hate green ideas.

Great ideas eat linguists.

...

Нормальная форма Хомского

грамматика в НФ Хомского (CNF) содержит правила вида:

$$A \to BC$$

$$A \to \alpha$$

$$\left(S \to \varepsilon\right)$$

- является КС-грамматикой
- любую КС-грамматику можно привести к НФ Хомского
- binary branching – строим бинарные деревья

Упражнение 1

Попробуем привести к CNF следующую КС-грамматику:

$S \rightarrow AB$
$A \rightarrow 0A1 \mid \varepsilon$
$B \rightarrow B1 \mid \varepsilon$

Упражнение 2

Попробуем построить КС-грамматику, порождающую следующие предложения:

Я ем грушу.
Я ем красивую грушу.
Я ем большую красивую грушу.
Я ем большую красивую грушу и яблоко.
Я ем большую красивую грушу и зелёное яблоко.

Сложные явления 1

- Модель управления (subcategorization frame)

Verb-with-NP-complement → find | leave | repeat | ...
Verb-with-S-complement → think | believe | say | ...
Verb-with-Inf-VP-complement → want | try | need | ...
VP → Verb-with-NP-comp NP
VP → Verb-with-S-comp S s
…

Frame	Verb	Example
∅	eat, sleep	I ate
NP	prefer, find, leave	Find [$_{NP}$ the flight from Pittsburgh to Boston]
NP NP	show, give	Show [$_{NP}$ me] [$_{NP}$ airlines with flights from Pittsburgh]
PP_{from} PP_{to}	fly, travel	I would like to fly [$_{PP}$ from Boston] [$_{PP}$ to Philadelphia]
NP PP_{with}	help, load	Can you help [$_{NP}$ me] [$_{PP}$ with a flight]
VPto	prefer, want, need	I would prefer [$_{VPto}$ to go by United Airlines]
S	mean	Does this mean [$_{S}$ AA has a hub in Boston]

Figure 12.6 Subcategorization frames for a set of example verbs.

Сложные явления 2

- Сочинение (coordination)

NP → NP and NP
VP → VP and VP
…

X → X - metarule

А также: согласование, long-distance dependencies …

Лексикализованные грамматики

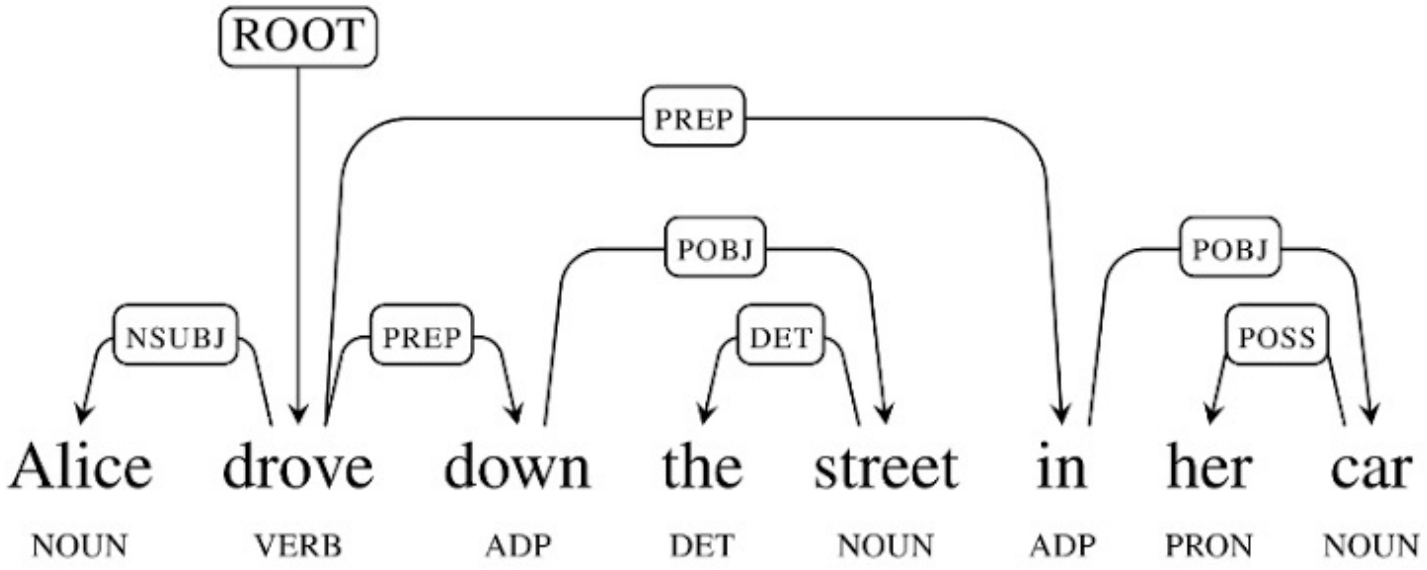
- Combinatory Categorial Grammar (CCG)
- Lexical-Functional Grammar (LFG)
- Head-Driven Phrase Structure Grammar (HPSG)
- Tree-Adjoining Grammar (TAG)
- …

Синтаксический анализ

- соотнесение входной строки (предложения) в заданной (или обученной по корпусу) грамматикой:

- распознавание (recognition) : да/нет – однозначный ответ
- собственно анализ (parsing) : дерево разбора / структура составляющих / последовательность продукций – возможны разные варианты

Структура зависимостей



Структура зависимостей

Элементы:

- зависимости (ребро, edge) : типы зависимостей
- узлы / вершины (node / vertex)
  - вершины / главные слова / хозяева
  - зависимые / подчинённые / слуги

> дерево – граф, у которого есть корневой узел (корень)

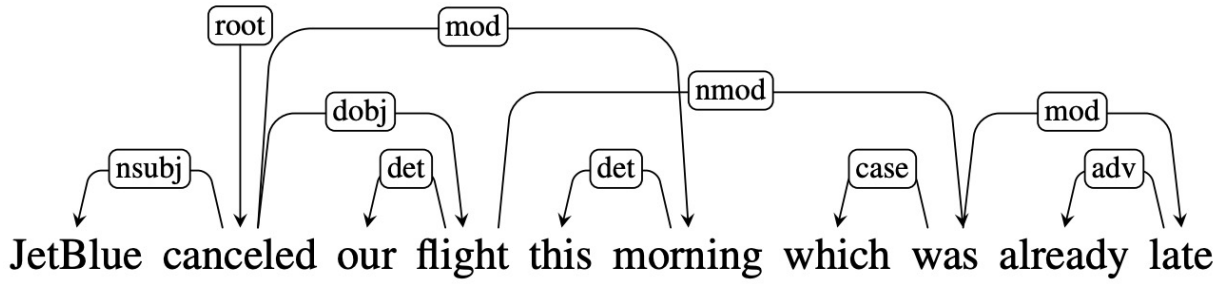
Требования

- Существует единственный корень
- В каждую вершину входит только одна стрелка
- Для каждой вершины существует единственный путь до неё от корня

Структура зависимостей

Важное свойство деревьев зависимостей – **проективность**

- никакая пара стрелок не пересекается (принцип непересечения стрелок)
- никакая стрелка не накрывает корневой узел (принцип обрамления стрелок)



Структура зависимостей

Проблемы:

- принцип единственности вершины
  - *Мы оставили комнату закрытой.*
- сочинительные конструкции
  - *Прошли день и ночь.*
  - *необходимые условия и результаты*
- иерархия синтаксических единиц

Описание зависимостей

- Могут быть получены из структуры составляющих с использованием алгоритмов head-finding:
  - задаются в КС-грамматике или по правилам
- Упорядоченные пары (head, dep)
  (flight, morning) (<root>, book)
- Пары с указанием направления стрелки:
  rightarc, leftarc
- Тройки (head, dep, relation)

Алгоритмы парсинга

- **top-down parsing** – нисходящие алгоритмы разбора (например, *Earley parser*)
- **bottom-up parsing** – восходящие алгоритмы разбора (например, *CYK parser*)

$S \longrightarrow \langle NP \rangle \langle VP \rangle$

$NP \longrightarrow N'$

$VP \longrightarrow V'$

$N' \longrightarrow \langle AdjP \rangle \langle N' \rangle$

…