И. В. Гоглев

КОНТЕКСТНО-СВОБОДНЫЕ ГРАММАТИКИ ДЛЯ АНАЛИЗА ТЕКСТОВОЙ ИНФОРМАЦИИ

Рассматриваются вариант использования контекстно-свободных грамматик для анализа текстовой информации.

Введение

Сегодня объёмы текстовых данных экспоненциально возрастают. В это же время технологии анализа и обработки текстовой информации стремительно меняются и модернизируются под влиянием новых тенденций и новых технологий. Анализ различной текстовой информации активно интегрируется в программные решения и программные продукты. Один из способов анализа текстовых данных – использование контекстно-свободных грамматик.

I. Контекстно-свободная грамматика

Контекстно-свободная грамматика – это набор правил объединения синтаксических компонентов в осмысленные строки. Например, именное словосочетание «the castlec (замок) включает определяющее слово (обозначенное тегом DT) и существительное (N).[1] Ниже представлена таблица синтаксических категорий для контекстно-свободных грамматик.

Таблица 1 – Синтаксические категории и их обозначения

Символ	Синтаксическая категория
S	Предложение
NP	Именное словосочетание
VP	Глагольное словосочетание
PP	Предложное словосочетание
DT	Определяющее слово
N	Существительное
V	Глагол
ADJ	Прилагательное
P	Предлог
TV	Переходный глагол
IV	Непереходный глагол

II. Правила контекстно-свободных грамматик

Построение правил контекстно-свободных грамматик разберём на предложении «John looks in the shopc (Джон заглянул в магазин). Именное словосочетание «the shopc включает в себя определяющее слово (DT) и существительное(N). Предложное словосочетание(PP) «in the shopc включает предлог (P) и именное словосочетание (NP). Глагольное словосочетание (VP) «looks in the shopc включает глагол (V) и предложное словосочетание (PP). Предложение (S) «John looks in the shopc включает всебя имя собственное (NNP) и глагольное словосочетание (VP). На основании данной информации строим правила для контекстно-свободной грамматики: S -> NNP VP; VP -> P NP; PP -> P PN; NP -> DT N; NNP-> 'John'; V -> 'looks'; P -> 'in'; DT -> 'the'; N -> 'shop'.

III. Анализ текстовых данных

Для работы с контекстно-свободными грамматиками существует библиотека NLTK (Natural Language Toolkit). Изменим правила грамматики для возможности анализа предложений с похожей структурой: «Mary saw Bobc, «Mary ate an applec. Ниже представлен листинг кода и результат выполнения:

```
from nltk import CFG, RecursiveDescentParser
grammar_rules = CFG.fromstring("""
S -> NP VP | NNP VP
VP -> V NP | V NP PP | V NNP | V NNP PP | V PP
V -> "saw" | "ate" | "looks"
NNP -> "John" | "Mary" | "Bob"
NP ->  Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "dog" | "cat" | "cookie" | "park" | "shop" | "apple"
PP -> P NP | P NNP
P -> "in" | "on" | "by" | "with"
""")

sent = "John looks in the shop".split()
rd_parser = RecursiveDescentParser(grammar_rules)
for p in rd_parser.parse(sent):
      print(p)
sent = "Mary ate an apple".split()
rd_parser = RecursiveDescentParser(grammar_rules)
for p in rd_parser.parse(sent):
      print(p)

sent = "Mary saw Bob".split()
rd_parser = RecursiveDescentParser(grammar_rules)
for p in rd_parser.parse(sent):
      print(p)
```

Рис. 1 – Пример использования контекстно-свободных грамматик

```
(S (NNP John) (VP (V looks) (PP (P in) (NP (Det the) (N shop)))))
(S (NNP Mary) (VP (V ate) (NP (Det an) (N apple))))
(S (NNP Mary) (VP (V saw) (NNP Bob)))
```

Рис. 2 – Результат выполнения программы

Также в NLTK существует возможность изобразить результат анализа в виде графа.

```
a=[]
for tree in rd_parser.parse(sent):
    a.append(tree)
a[0].draw()
```

Рис. 3 – Листинг кода отображения графа

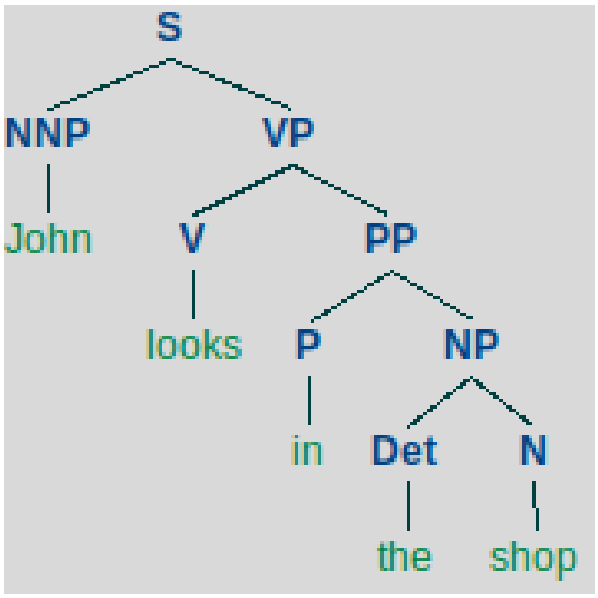

Рис. 4 – Граф предложения

Как можно заметить, левый элемент графа(«Johnc) – это субъект предложения, а крайний левый элемент графа («shopc) – объект.

IV. Выводы

Использование контекстно-свободных грамматик для анализа различной текстовой информации является преспективным направлением в обработке текстовых данных. Одно из главных преимуществ использования контекстно-свободных грамматик – их универсальность. Одна грамматика может использоваться для анализа нескольких, схожих по структуре предложений. При помощи контекстно-свободных грамматик возможно представить предложение в виде графа синтаксических категорий.

1. Бенгфорт, Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт, Р. Билбро, Т. Охеда. – СПб // Питер. – 2019. – С.368

Гоглев Иван Валентинович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, ivangoglev1998@gmail.com.

Научный руководитель: Герман Олег Витольдович, доцент кафедры информационных технологий автоматизированных систем БГУИР, к.т.н.