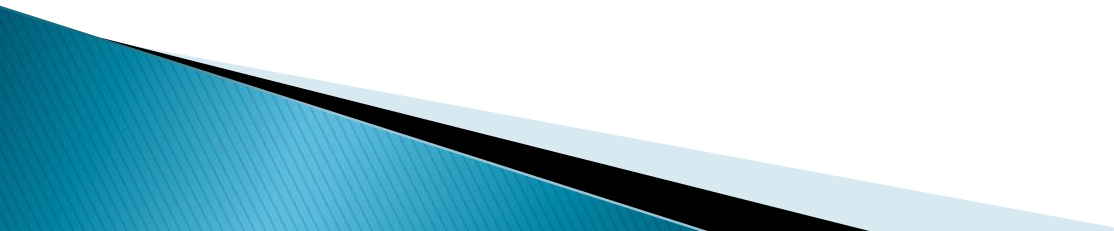


# Извлечение именованных сущностей



# Задача

- ▶ Задача извлечения именованных сущностей из текста (NER) заключается в идентификации заданных сущностей в тексте
  - ▶ Вход: неструктурированный текст
  - ▶ Результат: объекты
- 

# Задача извлечения именованных сущностей

## News NER example

Kofi Atta Annan is a Ghanaian diplomat who served as the seventh Secretary General of the United Nations from January 1, 1997, to January 1, 2007, serving two five-year terms. Annan was the co-recipient of the Nobel Peace Prize in October 2001.

Kofi Annan was born on April 8, 1938, to Victoria and Henry Reginald Annan in Kumasi, Ghana. He is a twin, an occurrence that is regarded as special in Ghanaian culture. Efua Atta, his twin sister, shares the same middle name, which means 'twin'. As with most Akan names, his first name indicates the day of the week he was born: 'Kofi' denotes a boy born on a Friday. The name Annan can indicate that a child was the fourth in the family, but in his family it was simply a name which Annan inherited from his parents.

In 1962, Annan started working as a Budget Officer for the World Health Organization, an agency of the United Nations. From 1974 to 1976, he was the Director of Tourism in Ghana. Annan then returned to work for the United Nations as an Assistant Secretary General in three consecutive positions.

Person
Location
Organization
Date
Nationality
Title

В Москве на вечеринке «Крыши мира» в «Бессонице» за диджейский пульт встанет канадский музыкант Art Department. Об этом «Ленте.ру» сообщили организаторы. Мероприятие пройдет в пятницу, 16 ноября. Art Department — проект канадского музыканта Джонни Уайта. В прошлом году Уайт выступил на вечеринках Circoloco и Elrow, отыграл в берлинском Watergate и Hi на Ибике, а также был заявлен в качестве одного из хедлайнеров фестиваля BPM в Португалии. Art Department — постоянный участник вечеринок Paradise, знаковых шоукейсов Джейми Джонса. Помимо диджеинга, Уайт ведет свой лейбл No.19 Music, объединивший таких музыкантов, как Мэтью Джонсон, Джейми Джонс, Martinez Brothers и Дэннис Феррер. Заказать билеты можно по ссылке.

# BIOES-схема

PER – персона

ORG – организация

LOC – место

B – первый токен в спане сущности, который состоит из больше чем 1 слова (beginning)

I – то, что находится в середине (inside)

E – последний токен сущности, которая состоит больше чем из 1 элемента (ending)

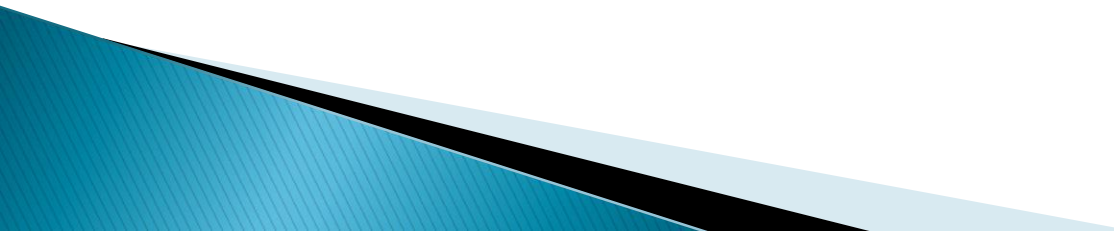
S – сущность состоит из одного слова (single)

O – не относится ни к какой сущности (out)

Карл Фридрих Иероним фон Мюнхгаузен родился в Боденвердере

B-PER I-PER I-PER I-PER E-PER OUT OUT S-LOC

# Контекстно–свободная грамматика

- ▶ Грамматика – способ описания формального языка
  - ▶ Порождающая грамматика задает правила, с помощью которых можно построить любое слово языка
  - ▶ В контекстно–свободной грамматике левые части всех продукций являются одиночными нетерминалами
- 

# Построение грамматики

- ▶ Терминал — объект, непосредственно присутствующий в словаре языка и имеющий конкретное, неизменяемое значение
- ▶ Нетерминал — объект, обозначающий какую-либо сущность языка и не имеющий конкретного символического значения
- ▶ Основу правил составляют предикаты
- ▶ Чтобы задать грамматику, требуется задать:
  - словарь терминалов и нетерминалов
  - набор правил вывода
  - выделить начальный нетерминал

# Правила

## ▶ Пример 1

- Терминал «100»
- Терминал «г.»
- Нетерминал Вес = 100 г.

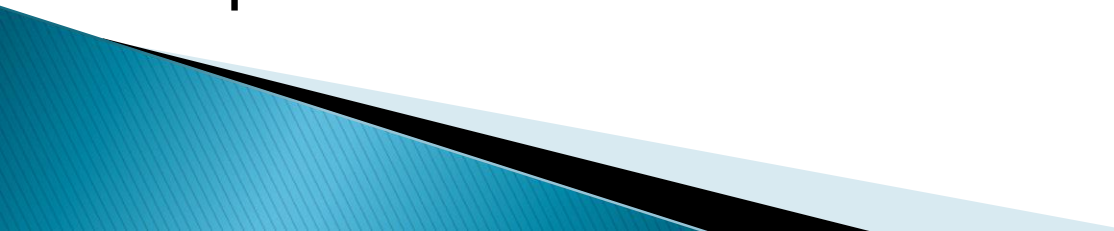
## ▶ Пример 2

- Число = 1 | 3 | 100 | 200 | 300 | 400 | 500
- Единица = г. | т. | кг.
- Вес = Число Единица

▶ Вес → Число Единица

▶ Вес → Числительное Существительное

# Парсер

- ▶ Парсер просматривает предложения (цепочки) и ищет в них подцепочки слов, соответствующие правилам грамматики
  - ▶ Подцепочки интерпретируются в разбитые по полям факты
  - ▶ В качестве терминалов выступают слова словаря, части речи, некоторые символы
  - ▶ Нетерминал строится из терминалов и нетерминалов
  - ▶ Если нетерминал встречается только в левой части и никогда в правой, значит, это вершина грамматики
- 



# Интерпретация

- ▶ Грамматика
  - PP = Prep Noun
  - S = Verb PP
- ▶ Объект
  - Fact = {Field1 : string; Field2 : int}
- ▶ Интерпретация
  - S = Verb interp (Fact.Field1) PP

# Парсеры для русского языка

Томи́та-парсер	Yargy
Разрабатывался много лет внутри Яндекса	Open source, разрабатывается сообществом
10 000+ строк кода на C++	1000+ на Python
CLI	Python-библиотека
Protobuf + конфигурационные файлы	Python DSL
Нет готовых правил	<u>Natasha</u> — готовые правила для извлечения имён, дат, адресов и других сущностей
Медленный	Очень медленный

# Пример 1

```
from yargy import rule, and_, or_, Parser
from yargy.predicates import caseless, normalized,
dictionary, gte, lte
```

```
DAY = and_( gte(1), lte(31) )
MONTH = and_( gte(1), lte(12) )
YEAR = and_( gte(1900), lte(2021) )
DATE = rule(DAY , '.', MONTH, '.', YEAR)
parser = Parser(DATE)
```

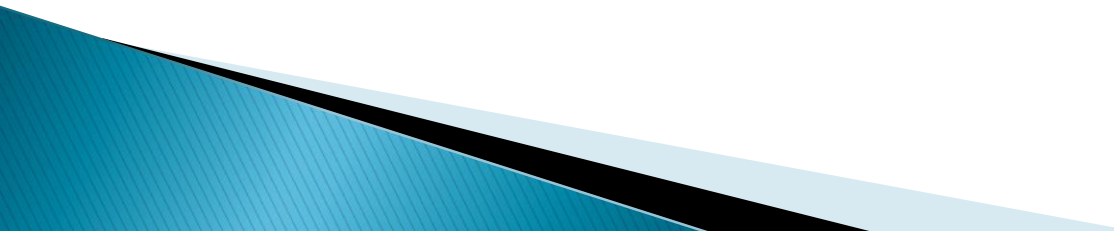
# Пример 1

```
text = 'Первый искусственный спутник Земли был  
запущен 04.10.1957'
```

```
for match in parser.findall(text):  
    print(match.span, [_value for _ in match.tokens])
```

Результат:

```
[47, 57) ['04', '.', '10', '.', '1957']
```



# Пример 2

```
Date = fact('Date', ['d', 'm', 'y'])
```

```
DAY = and_(gte(1), lte(31)).interpretation(Date.d)
```

```
MONTH = and_(gte(1), lte(12)).interpretation(Date.m)
```

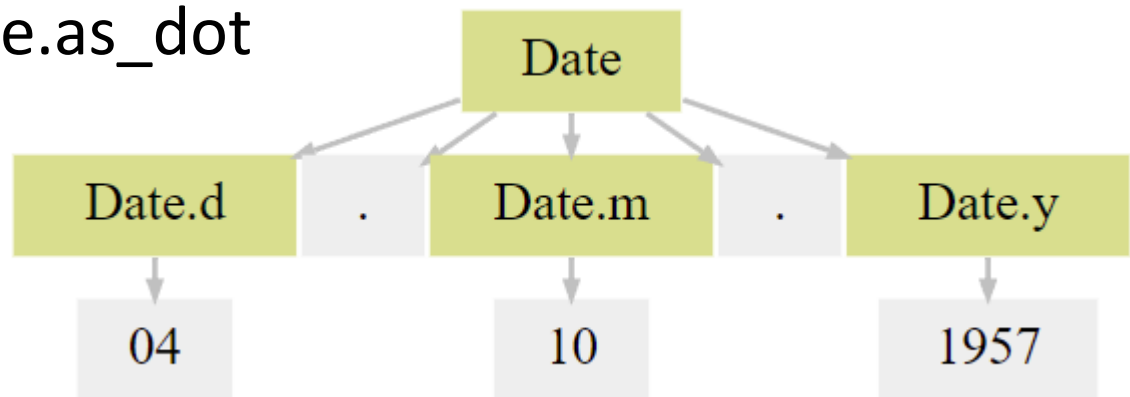
```
YEAR = and_(gte(1900), lte(2021)).interpretation(Date.y)
```

```
DATE = rule(DAY, '.', MONTH, '.', YEAR)
```

```
.interpretation(Date)
```

```
parser2 = Parser(DATE)
```

```
parser2.find(text).tree.as_dot
```



# Пример 2

```
text2 = 'Первый искусственный спутник Земли был  
запущен 4 октября 1957 года'
```

# Пример 2

```
text2 = 'Первый искусственный спутник Земли был  
запущен 4 октября 1957 года'  
Date = fact('Date', ['d', 'm', 'y'])  
DAY = and_(gte(1), lte(31)).interpretation(Date.d)  
MONTH = or_(rule(normalized('октябрь')), rule(normalized('ноябрь'))).interpretation(Date.m)  
YEAR = and_(gte(1900), lte(2021))  
YEARG = rule(YEAR, normalized('год').optional()).  
interpretation(Date.y)  
DATE = rule(DAY, MONTH, YEARG).interpretation(Date)  
parser3 = Parser(DATE)
```

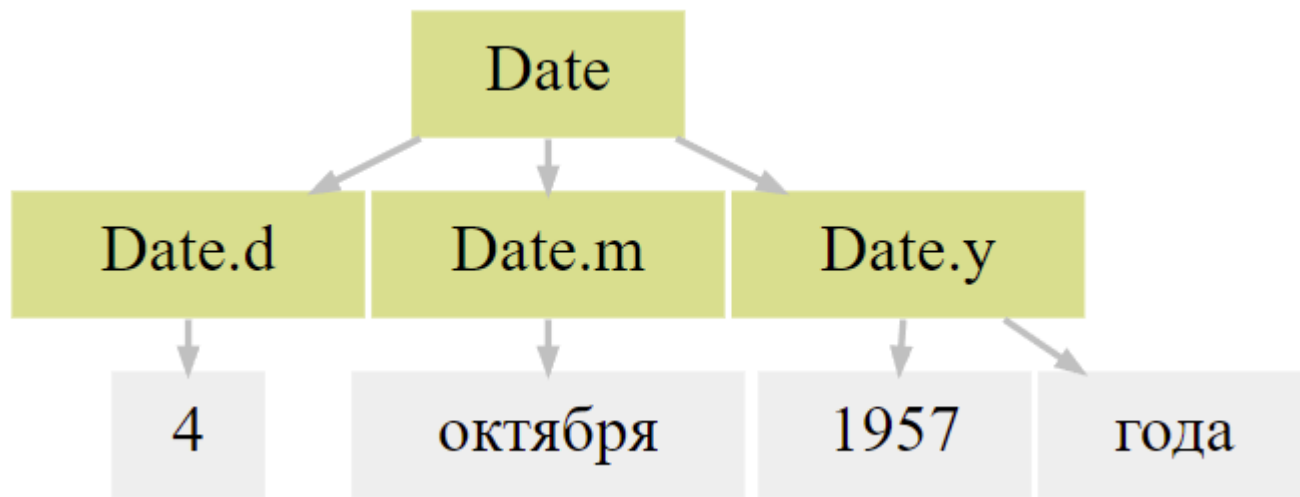
# Пример 2

```
text2 = 'Первый искусственный спутник Земли был  
запущен 4 октября 1957 года'
```

```
for match in parser3.findall(text2):  
    print(match.fact)
```

Результат: Date(d='4', m='октября', y='1957 года')

```
parser3.find(text2).tree.as_dot
```





# Пример 3

```
NN = rule(gram('NOUN'),gram('NOUN'))  
parser4 = Parser(NN)  
for match in parser4.findall(text):  
    print(match.span, [_value for _ in match.tokens])
```

Результат:

```
[21, 34) ['спутник', 'Земли']
```