Предобработка текстовых данных

Препроцессинг = предобработка

Нормализация (разное понимание)

Сегментация

Токенизация

Приведение к нижнему регистру (?)

Лемматизация/стемминг

Удаление пунктуации (?)

Удаление стоп-слов (?)

Нормализация

Нормализация

Зависит от задачи

Удаление лишней информации: html разметка, код, хэштеги, url-ы…

Унификация тире/дефисов, кавычек (если нужна пунктуация)

Удаление мусорной пунктуации

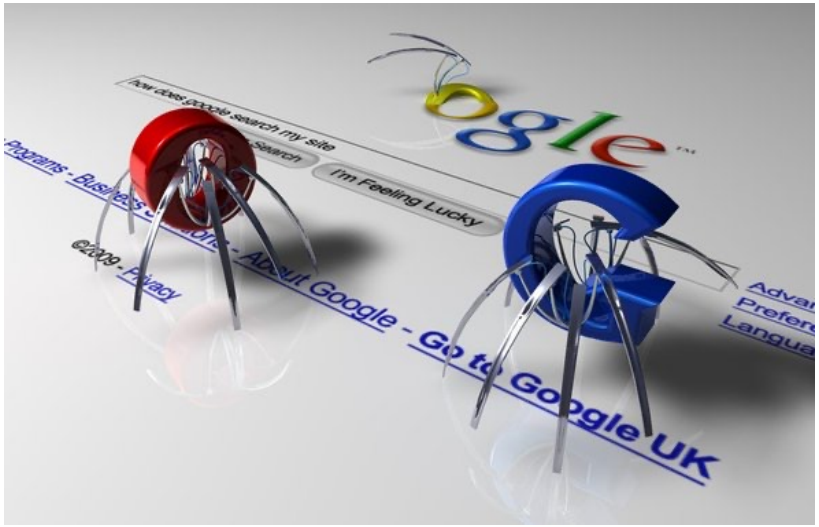
Нормализация

Парсинг и краулинг

Краулер («паук») – ходит по сайтам (сайту) и собирает html разметку с текстовыми данными

Парсинг – извлечение конкретной информации с сайта (напр., данные по товарам).

В КЛ данные собирают краулером (нам интересен весь текст пользователя, журнала или газеты)



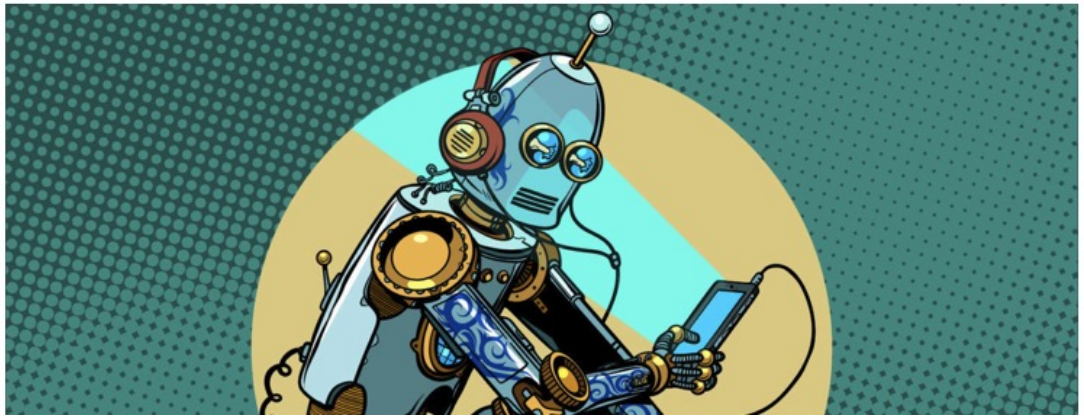
Пример текста из краулера

```
<?xml version="1.0" encoding="UTF-8"?>
<articles>
 <article>
   <url>https://m.livejournal.com/read/user/vd_juliya/38949/comments/p1</url>
   <rule>jjpost</rule>
   <crawldatetime>2020/03/11 15:12:15</crawldatetime>

   <comment>
<text><o>
Какая у тебя красивая и уже по настоящему осенняя дорога на
работу)))...эх,нам бы сейчас вашего дождика,хоть немножко...жара замучила
уже...
</o></text>
<author>cvetohnicaanuta</author>
<datetime>September 22 2015, 17:54:24 UTC</datetime>
   </comment>

   <comment>
<text><o>
Ага, дорога, красивая. Кругом деревья. Это я еще поздно спохватилась. На
выезде с нашего района вообще лес вдоль дороги))))<br />
</o></text>
```

Изучаем синтаксические парсеры для русского языка

Блог компании Сбер , Программирование *, Машинное обучение *, Искусственный интеллект

Привет! Меня зовут Денис Кирьянов, я работаю в Сбербанке и занимаюсь проблемами обработки естественного языка (NLP). Однажды нам понадобилось выбрать синтаксический парсер для работы с русским языком. Для этого мы углубились в дебри морфологии и токенизации, протестировали разные варианты и оценили их применение. Делимся опытом в этом посте.



Скраулили вот этот пост с Хабра

https://habr.com/ru/company/sberbank/blog/418701/

Текст поста из краулера

Привет! Меня зовут Денис Кирьянов, я работаю в Сбербанке и занимаюсь проблемами обработки естественного языка (NLP). Однажды нам понадобилось выбрать синтаксический парсер для работы с русским языком. Для этого мы углубились в дебри морфологии и токенизации, протестировали разные варианты и оценили их применение. Делимся опытом в этом посте.

 <h2>Подготовка к отбору </h2>
 Начнём с основ: как все работает? Мы берем текст, проводим токенизацию и получаем некоторый массив псевдослов-токенов. Этапы дальнейшего анализа укладываются в пирамиду:

Нормализация

Как удалить все теги, оставшиеся после выкачки краулера?

Basic Text Processing

Regular Expressions

Regular expressions

A formal language for specifying text strings

How can we search for any of these?

- woodchuck
- woodchucks
- Woodchuck
- Woodchucks

Regular Expressions: Disjunctions

Letters inside square brackets []

Pattern	Matches
`[wW]oodchuck`	Woodchuck, woodchuck
`[1234567890]`	Any digit

Ranges `[A-Z]`

Pattern	Matches	
`[A-Z]`	An upper case letter	`Drenched Blossoms`
`[a-z]`	A lower case letter	`my beans were impatient`
`[0-9]`	A single digit	`Chapter 1: Down the Rabbit Hole`

Regular Expressions: Negation in Disjunction

Negations [^Ss]

- Carat means negation only when first in []

Pattern	Matches	
[^A-Z]	Not an upper case letter	Oyfn pripetchik
[^Ss]	Neither 'S' nor 's'	I have no exquisite reason"
[^e^]	Neither e nor ^	Look here
a^b	The pattern a carat b	Look up a^b now

Regular Expressions: More Disjunction

Woodchuck is another name for groundhog!

The pipe | for disjunction

Pattern	Matches
`groundhog|woodchuck`	woodchuck
`yours|mine`	yours
`a|b|c`	= [abc]
`[gG]roundhog|[Ww]oodchuck`	Woodchuck

Regular Expressions: ? *+ .

Pattern	Matches	
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n

Stephen C Kleene

Kleene *, Kleene +

Regular Expressions: Anchors ^ $

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 "Hello"
\.$	The end.
.$	The end? The end!

Example

Find me all instances of the word “the” in a text.

`the`

Misses capitalized examples

`[tT]he`

Incorrectly returns `other` or `theology`

`[^a-zA-Z][tT]he[^a-zA-Z]`

Errors

The process we just went through was based on fixing two kinds of errors:

1. Matching strings that we should not have matched (there, then, other)

   False positives (Type I errors)

2. Not matching things that we should have matched (The)

   False negatives (Type II errors)

Регулярные выражения

To be continued на парах по Питону

Сегментация

Сегментация (на предложения)

Простая задача, какие могут быть сложности?

Сегментация (на предложения)

Точка – не всегда конец предложения (инициалы, сокращения)

Пунктуационные кластеры (???!!!) – делить только после целого кластера

Не всегда заглавная буква после .?! (тексты соцсетей)

Прямая речь – как делить?

Как отделять названия?

Пункты списков – разные предложения?

Многоточие – отдельная проблема…

Сегментация (на предложения)

- И т. д. и т. п. В общем, вся газета
- Встречаемся у м. Китай-город.
- Эта шоколадка за 400р. ничего из себя не представляла.
- А у нас жара...даже не знаю радоваться или огорчаться этому.
- в своей игре только что чуть вассермана не обыграли о.о
- - Куда, - говорю, - едем, батя?
- А.С.Пушкин
- А. Я не знала. Прости

Сегментация (на предложения)

Разные готовые решения, например, sent_tokenize из NLTK

NLTK

NLTK (Natural Language Toolkit) – ведущая платформа для создания NLP-программ на Python.

NLTK включает в себя большой набор библиотек для обработки текста, а именно: сегментации, токенизации, стемминга и много другого.

Сегментация (на предложения)

А также множество других готовых решений:
Razdel (https://github.com/natasha/razdel)

Ru_sent_tokenize от DeepPavlov
(https://github.com/deepmipt/ru_sentence_tokenizer)

Делить регулярками

Токенизация

Токенизация

Проблемы с дефисами

Нью-Йорк

14-летний

14-тилетний

31-ое

Все-таки vs он-таки

Очень-очень

Токенизация

Email-ы, url-ы, хештеги (если мы их оставили).

(Почему нам интересно в морфо- и синтаксическом разборе оставлять урлы и хештеги?)

marimitchurina@gmail.com

Olga_1990@gmail.com

70-mail-mail@ya.ru

Vk.ru

Вумен.ру

#Новыйгод2022

Токенизация

0.5

0,5

.5

1.0 vs 1.

29/09/2021

21.09.22

18:00

url-ы

What’s

crocs’ы

звуки удивления

**па

б/у vs он/она

Токенизация

Готовые решения

NLTK

Razdel

Mystem

регулярка

Приведение к нижнему регистру

Приведение к нижнему регистру

Где плохо, а где хорошо?

Приведение к нижнему регистру

Может быть плохо

NER и др. задачи Information Retrieval

Морфосинтаксис

Стемминг и лемматизация

Стемминг и лемматизация

Зачем это нужно?

Позволяет собрать по корпусу статистику использования **именно слова, а не его формы.**

По лемме и грамматическим значениям можно синтезировать словоформу (если нужно).

Стемминг

Стемминг

Приведение к «псевдооснове» слова

Грубое отрезание «лишнего»

```
dog, dogs, dog's, dogs' => dog
```

Стемминг. Идея

Как работают стеммеры

- Исчислим возможные суффиксы и окончания, объединив их в группы (например, окончания деепричастий, «ейш»/«ейше» «ост»/»ость» и т. п.)
- Будем последовательно удалять группы окончаний в «правильном порядке»

Output: быстрый, быстрее => быстр; побыстрее => побыстр

Стемминг

Плюсы

Минусы

Стемминг

Плюсы

Просто (проще лемматизации)

Можно запилить на правилах

Быстро работают

Минусы

Сложно для флективных языков

Может отрезать словообразовательные суффиксы

Супплетивные формы

Омонимия аффиксов (player и smarter)

Омонимы?

Разные слова к одной стемме (курить и куры -> кур)

Стемминг

```
the boy’s dogs are different sizes => the boy dog be differ size
```

Лемматизация

Лемматизация

Приведение к начальной форме слова (лемма)

Лемматизация

Плюсы

Минусы

Лемматизация

Плюсы

Учитывает омонимию

Супплетивизм

Больше смысловой нагрузки

Минусы

Сложно в реализации

Дольше работает

Даже хорошие SOTA парсеры, учитывающие морфу и синтаксис работают не идеально

Лемматизация

```
Stemmer: seen
Lemmatizer: see

Stemmer: drove
Lemmatizer: drive
```

Даже для английского лемматизация кажется лучше, но всегда нужно оценивать **скорость** и **качество**

Лемматизация

```
Stemmer: seen
Lemmatizer: see

Stemmer: drove
Lemmatizer: drive
```

Даже для английского лемматизация кажется лучше, но всегда нужно оценивать **скорость** и **качество**

Лемматизация. Алгоритмы

пальто-	плакать	рук-а
	плач-у	рук-и
	плач-ешь	рук-е
	плач-ет	рук-у
	плач-ем	рук-ой
	плач-ете	о рук-е
	плач-ут	

Грамматический словарь:

Словоизменительные парадигмы

Леммы

ЧР и грамматическая информация

Удаление пунктуации

Удаление пунктуации

Москва - это столица VS кто-то

Синий, зеленый VS 0,5

Удаление стоп-слов

Удаление стоп-слов

Союзы, предлоги…

Несут мало лексического смысла, только грамматическое

Создают ненужный шум

Не во всех задачах удаляем

Стоп-слова

Слово	Количество	Частота, %
и	14	3.71
в	13	3.45
быть	6	1.59
вы	6	1.59
но	6	1.59
не	5	1.33
их	4	1.06
как	4	1.06
а	3	0.80
где	3	0.80
если	3	0.80
который	3	0.80
на	3	0.80
с	3	0.80
такой	3	0.80

Удаление стоп-слов

Список стоп-слов в NLTK

Другой список: https://github.com/stopwords-iso/stopwords-ru/blob/master/stopwords-ru.txt