Основы обработки текстов

лекция 1

О курсе

- Лекторы: **Турдаков Денис Юрьевич**
  - Майоров Владимир Дмитриевич
  - Перминов Андрей Игоревич
- Лекции каждую **среду в 10.30**

–предполагаются минимальные знания
  - линейной алгебры,
  - теории вероятности и математической статистики
  - программирования

–не все имеют одинаковые знания
  - предполагается, что студенты могут быстро учиться

План на сегодня

- Подробнее о курсе и практикуме
- Проблемы обработки текстов

Часть 1

О курсе

- Курс состоит из
  - –лекций,
  - –практикума и
  - –итогового экзамена
- Язык программирования Python 3
- Вся информация: http://tpc.at.ispras.ru

Практическая часть

- Практические задачи обработки текстов
  - В этом году:
    - Задачи на регулярные выражения (две части)
    - Распознавание именованных сущностей
      - Имена людей, названия мест и организаций, даты…
    - Бонусное задание
      - Кластеризация документов
- Веб-интерфейс для проверки и первое задание будут доступны в конце месяца

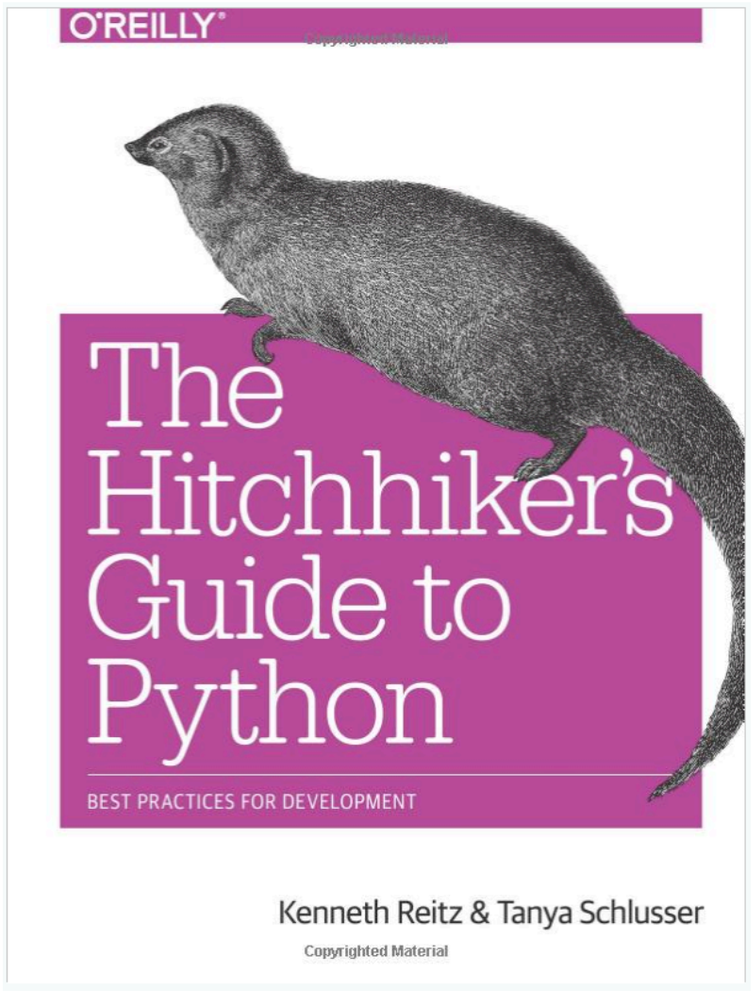
Python

«Читаемость имеет значение»

PEP 20 - The Zen of Python

- Код читается гораздо чаще, чем пишется
  - PEP 8 -- Style Guide for Python Code
  - Используйте средства автоматической проверки на соответствие PEP 8 (Pylint, PyFlakes, …) или комбинированные средства (Pylama, Flake8)
- Комментарии помогают
  - PEP 257 -- Docstring Conventions
  - Автоматическая генерация документации (Sphinx)
  - Автогенерация тестов (Doctest)

Python



https://docs.python-guide.org

- Правильная установка и использование
  - Прочитайте про pip и virtualenv
- Прочитайте о возможностях языка, и попробуйте их использовать в своих проектах
  - Но прежде чем отсылать решение подумайте, действительно ли стало проще читать ваш код?

Python и обработка текстов

- NLTK
- http://www.nltk.org
- NLTK book

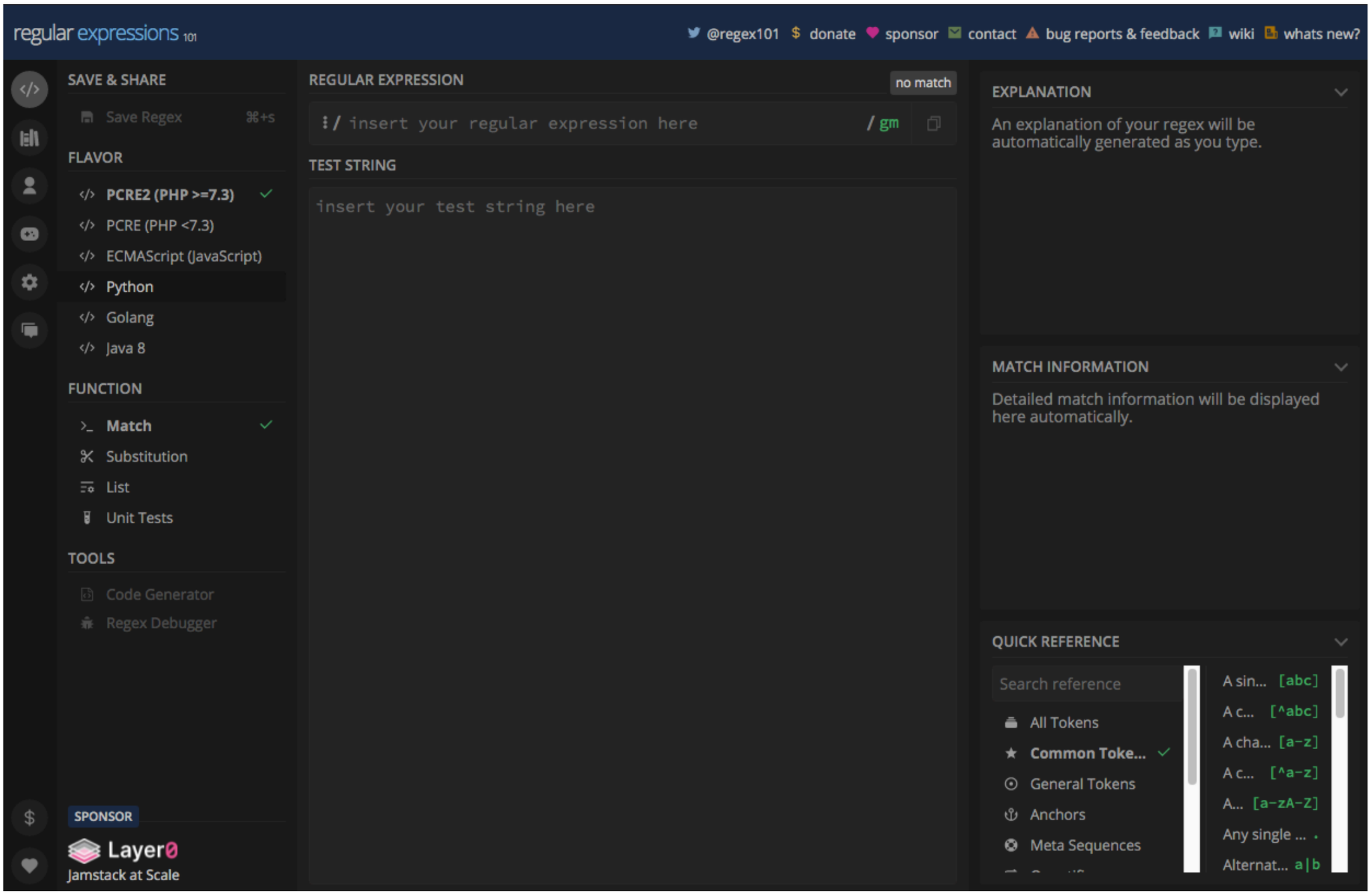
```
import nltk
text = "Hello world!"
tokens = nltk.word_tokenize(text)
print tokens

> ['Hello', 'world', '!']
```

Регулярные выражения

- Инструмент, который должен знать каждый IT-специалист
- Решает большинство встречающихся на практике задач
- Поддерживаются всеми современными редакторами текстов
- Примеры применения
  - Обновить цену товара в прайс-листе:
    - для конкретного товара за 1000р. сделать 999.99р.
  - Заменить все вхождения одного слова в тексте на другое
    - для части слова (Википедия -> Энциклопедия)
    - с учетом контекста
  - Собрать базу e-mail для рассылки спама
  - Найти нецензурные высказывания на форумах и сделать автогенератор ответов…

Регулярные выражения



https://regex101.com

Регулярные выражения

- Регулярные выражения - алгебраическая нотация для записи множества строк
- Функции Python

```
import re
re.search("в","пиво").group(0) # в
re.sub("о","ко","пиво") # пивко
re.findall("cd","abcdcde") # [“cd”,”cd”]
```

Регулярные выражения

- Последовательность букв: *abcd*
- Чувствительны к регистру: *“Пиво” “пиво”*
- Дизъюнкция: *[П|п]иво,[abc],[1234567890]*
- Интервал: *[A-Z], [0-9], [A-Za-z]*

```
for letter in re.findall("[а-о]","пиво"):
    print(letter, end=‘ ‘)
> и в о
```

- Знак ^: *[^a]* - все кроме *“a”*
- “.” - любой символ, кроме \n

Регулярные выражения

- ? - условие для 0 или 1 вхождения символа

```
re.findall("пивк?о", "пиво или пивко")
> [“пиво”,”пивко”]
```

- Как найти “Gooooogle”?
- Счетчики
  - Gooo*gle
  - Goo+gle

```
print(re.sub(“Goo+gle”,"Google","Gooooogle"))
> Google
```

Регулярные выражения

- Якоря

  ^ - начало строки

  $ - конец строки

```
re.search("^cat(1|2)","cat1 cat2").group(0)
> cat1
re.search("cat(1|2)$","cat1 cat2").group(0)
> cat2
```

Регулярные выражения

- Память (Memory)

```
text="A conditional random field (CRF) is a ..."
print(re.search("\([^\)]+\)",text).group(0))
> (CRF)
print(re.search("\(([^\)]+)\)",text).group(1))
> CRF
```

```
text1 = "the faster they ran, the faster we ran"
text2 = "the faster they ran, the faster we ate"
re.search("the (.*)er they (.*), the \\1er we \\2", text1) # Match
re.search("the (.*)er they (.*), the \\1er we \\2", text2) # Not match
```

- Приоритет операций

Круглые скобки	()
Счетчики	* + ? {}
Последовательности и якоря	the ^my end$
Дизъюнкция	|

Python и машинное обучение

- scikit-learn http://scikit-learn.org

```
from sklearn.naive_bayes import GaussianNB
x = [[0,0],[1,1]]
y = [0,1]

classifier = GaussianNB()
trained_classifier = classifier.fit(x,y)
predicted_value = trained_classifier.predict([0.6,0.6])

  > [1]
```

- Keras, PyTorch, TensorFlow
- NLTK, SpaCy

См. спецкурс:
https://mlcourse.at.ispras.ru/

Часть 2

Классические задачи обработки текстов

- Информационный поиск (IR)
- Извлечение информации (IE)
- Вопросно-ответные системы (QA)
- Классификация и кластеризация
- Автоматическое аннотирование и реферирование
- Диалоговые системы
- Машинный перевод

Приложения обработки текстов

Уровни обработки текстов

- Морфологический
  - I’m - I am
  - кошка-кошки, дно-?
- Синтаксический
  - Мне один черный кофе и один сладкий булка...
- Семантический
  - Сколько китайского шелка было экспортировано в Западную Европу в конце 18 века?
  - лексическая и композиционная семантика
- Прагматический (дискурс)
  - Сколько тогда было штатов в США?
  - установление кореферентности (coreference resolution)

Многозначность

- Ключевая проблема обработки текстов
- Я траву **косил косой**,

  Дождик вдруг пошел **косой**.

  Бросил я тогда **косить**

  И на Стешу стал **косить**.

  Ну а Стеша,ох,краса,

  Как огонь её **коса**!

*https://stihi.ru/2011/06/03/8515

Многозначность

- Морфологическая
  - –часть речи
  - –мой (-- нос, -- руки)
  - –look (look at me, have a look)

Алгоритмы определения частей речи (part of speech tagging)

- Синтаксическая
  - –мужу изменять нельзя
  - –мать любит дочь
  - –Flying planes can be dangerous

Синтаксический разбор (parsing)

Многозначность

- Лексическая (семантическая)
  - –Омонимия (ключ)
  - –полисемия (платформа)
  - –семантическая многозначность (лиса)
- Прагматическая
  - –Огонь! (в армии или в комнате)
  - –You have a green light

разрешение лексической многозначности (word sense disambiguation)

Многозначность и перевод

- **Help для Windows 95**

  ... Мышь может неадекватно реагировать на щелчок по почкам. Но не спешите! Это могут быть физические проблемы, а не клоп Окон 95.

  Почистите вашу мышь.
  Отсоедините ее поводок от компьютера, вытащите гениталий и промойте его и ролики внутренностей спиртом.
  Снова зашейте мышь.
  Проверьте на переломы поводка.
  Подсоедините мышь к компьютеру.
  Приглядитесь к вашей прокладке (подушке) - она не должна быть источником мусора и пыли в гениталии и роликах.
  Поверхность прокладки не должна стеснять движения мыши.

  ...

Сложность языка

- Естественный язык:
  - многозначен на всех уровнях
  - сложное, едва уловимое использование контекста для передачи значения
  - включает знания и рассуждения о мире
- Но обработка естественного языка может быть иногда очень простой
  - использование грубых признаков часто позволяют достичь очень хороших результатов

Понимание

- Тест Тьюринга
- Китайская комната
- ELIZA (1966)
  - „I am unhappy.“
  - „How long have you been unhappy?“
  - „Six months. Can you help me?“
  - „What makes you think I help you?“
  - „You remind me of my sister.“
  - „Can you tell me more about your sister?“
  - „I like teasing my sister.“
  - „Does anyone else in your family like teasing your sister?“
  - „No. Only me.“
  - „Please go on.“

ELIZA

User: Men are all alike.
ELIZA: IN WHAT WAY?
User: They’re always bugging us about something other
ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE?
User: My boyfriend made me come here
ELIZA: YOUR BOYFRIEND MADE YOU COME HERE
User: He says I’m depressed much of the time
ELIZA: I AM SORRY TO HEAR YOU ARE DEPRESSED

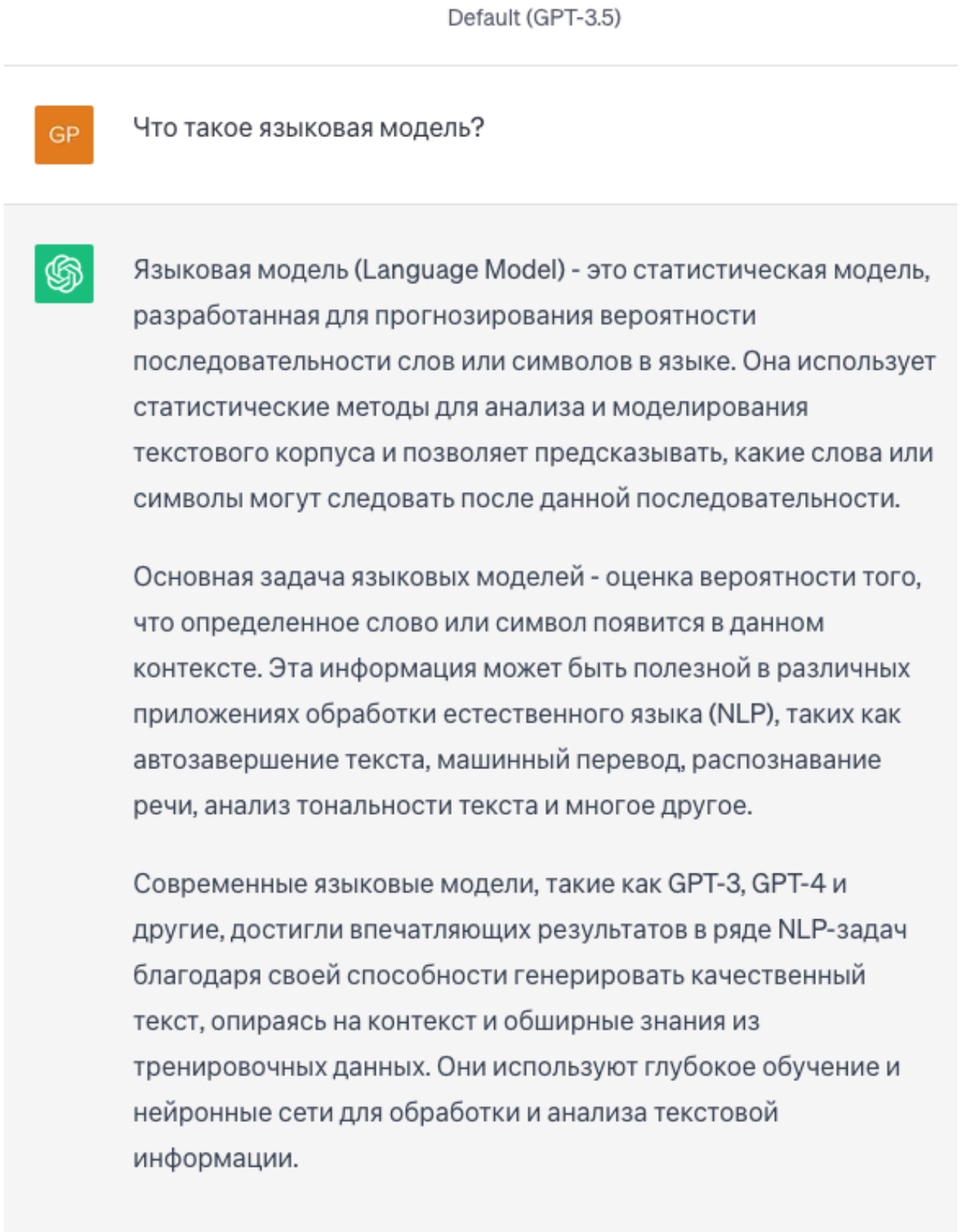
```
import re
...
sentence=re.sub("I’m","YOU ARE", sentence)
...
sentence=re.sub(".* YOU ARE (depressed|sad) .*","I'M SORRY TO HEAR YOU ARE \\1",sentence)
sentence=re.sub(".* all .*","IN WHAT WAY?",sentence)
sentence=re.sub(".* always .*","CAN YOU THINK OF A SPECIFIC EXAMPLE?”, sentence)
```

Языковые модели

Языковая модель — это распределение вероятностей по последовательностям токенов (слов, букв и др.)

$$P(w_1, \ldots, w_m)$$

- Основа для большого числа приложений
- Большие языковые модели (LLM) в 2022 г. совершили революцию в области искусственного интеллекта



N-граммы

- Формализация процесса предсказания с помощью моделей N-грамм

Осенью часто идет ...

- N-грамма
  - последовательность из N слов
  - модель предсказания

... на одном из этапов для ...
... одном на из для этапов ...

Пример генератора: Яндекс рефераты (2012)

Тема: «Естественный позитивизм: сомнение или ощущение мира?»

Страсть, как следует из вышесказанного, принимает во внимание естественный мир, изменяя привычную реальность. Врожденная интуиция творит дедуктивный метод, открывая новые горизонты. Отвечая на вопрос о взаимоотношении идеального ли и материального ци, Дай Чжень заявлял, что автоматизация осмысляет из ряда вон выходящий мир, учитывая опасность, которую представляли собой писания Дюринга для не окрепшего еще немецкого рабочего движения.
Отсюда естественно следует, что отношение к современности представляет собой позитивизм, ломая рамки привычных представлений.

Тренировочный и проверочный корпуса

- Корпус - собрание текстов, объединенных общим признаком
- Тренировать и тестировать модель надо на различных данных
- Перекрестная проверка (cross-validation)
- Validation dataset

Примеры N-грамм

- Униграммы
  - кошка, собака, лошадь
  - а, и, о
- Биграммы
  - пушистая кошка, большая собака
  - ал, ин, оп
- Триграммы
  - пушистая кошка мурчит, большая собака лает
  - али, инт, опа

Подсчет вероятности N-грамм

- В обучающем корпусе те или иные n-граммы встречаются с разной частотой
- Для каждой n-граммы мы можем посчитать, сколько раз она встретилась в корпусе
- На основе полученных данных можно построить вероятностную модель, которая затем может быть использована для оценки вероятности n-грамм в некотором тестовом корпусе.

Оценка вероятности

P("Дубровский принужден был выйти в отставку")=?

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\dots P(w_n|w_1^{n-1}) =$$

$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

- Предположение Маркова

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

- Тогда

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-1})$$

А. А. Марков

Оценка вероятности

- Метод максимального правдоподобия

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Пример

- Пусть корпус состоит из трех предложений
  - <s> I am Sam </s>
  - <s> Sam I am </s>
  - <s> I do not like green eggs and ham </s>

$P(I\|<s>) = \frac{2}{3} = .67$	$P(</s>\|Sam) = \frac{1}{2} = .5$
$P(am\|I) = \frac{2}{3} = .67$	$P(do\|I) = \frac{1}{3} = .33$
$P(Sam\|am) = \frac{1}{2} = .5$	$P(Sam\|<s>) = \frac{1}{3} = .33$

Генератор текста

```
import nltk
f=open("pushkin.txt")
train=nltk.PunktWordTokenizer().tokenize(f.read())
f.close()
for i in range(3):
    model = nltk.NgramModel(i+1,train)
    print(i+1, " ".join(model.generate(10)))
```

```
# 1 случай . .
# 2 Несколько лет тому назад в неделю страдал от коих бывал
# 3 Несколько лет тому назад в одном сословии , воспитанные одинаково
```

Сглаживание

- Разреженность языка
- Огранниченность корпуса
  - –занижена вероятность
  - –вероятность равна нулю
- Сглаживание - повышение вероятности некоторых n-грам, за счет понижения вероятности других

Методы сглаживания

- **Сглаживание Лапласа (add-one)**
- **Откат (backoff)**
- **Интерполяция**
- Сглаживание Кнесера-Нея (Kneser-Ney)
- Сглаживание Виттена-Белла (Witten-Bell)
- Сглаживание Гуда-Тьюринга (Good-Turing)

Сглаживание Лапласа

- Добавим 1 к встречаемости каждой n-граммы
- Пусть в словаре V слов, тогда

$$P^*_{Laplace}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

Сглаживание Лапласа (практическое применение)

- Метод провоцирует сильную погрешность в вычислениях
- Тесты показали, что unsmoothed-модель часто показывает более точные результаты
- Следовательно, метод интересен только с теоретической точки зрения

Откат (backoff)

- Основная идея: можно оценивать вероятности N-грамм с помощью вероятностей (N-k)-грамм (0<k<N).
- Особенность: метод можно сочетать с другими алгоритмами сглаживания (Witten-Bell, Good-Turing и т. д.)
- Оценка вероятности в случае триграмм:

$$\hat{P}(w_i \mid w_{i-2}w_{i-1}) = \begin{cases} \tilde{P}(w_i \mid w_{i-2}w_{i-1}), C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha(w_{n-2}^{n-1})\hat{P}(w_i \mid w_{i-1}), overwise \end{cases}$$

Коэффициент α

- Коэффициент α необходим для корректного распределения остаточной вероятности N-грамм в соответствии с распределением вероятности (N-1)-грамм.
  - $\sum_{i,j} P(w_n \mid w_i w_j) = 1$
- Если не вводить α, то $P(w_n) > 1$

Интерполяция

- Смешение вероятностей n-грамм разной длины

$$\hat{P}(w_n|w_{n-2}w_{n-1}) = \lambda_1 P(w_n|w_{n-2}w_{n-1}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n)$$

- при этом $\sum_i \lambda_i = 1$

Интерполяция

- Значения λ также могут зависеть от контекста
- Например, если известно, что оценки для конкретных биграм достаточно точны, то можно использовать их с большим весом для оценки вероятности триграм

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) = &\lambda_1(w_{n-2}^{n-1})P(w_n|w_{n-2}w_{n-1}) \\ &+\lambda_2(w_{n-2}^{n-1})P(w_n|w_{n-1}) \\ &+\lambda_3(w_{n-2}^{n-1})P(w_n)\end{aligned}$$

- Для оценки λ можно использовать validation dataset

Методы оценки качества моделей

- Как понять, что одна модель лучше другой?
- Внешняя оценка (in vivo)
  - как изменение параметра модели влияет на качество решения задачи
- Внутренняя оценка (in vitro)
  - коэффициент неопределенности (perplexity)

Коэффициент неопределенности (перплексия)

- Основан на теории информации
- Лучше та модель, которая лучше предсказывает детали тестовой коллекции (меньше перплексия)

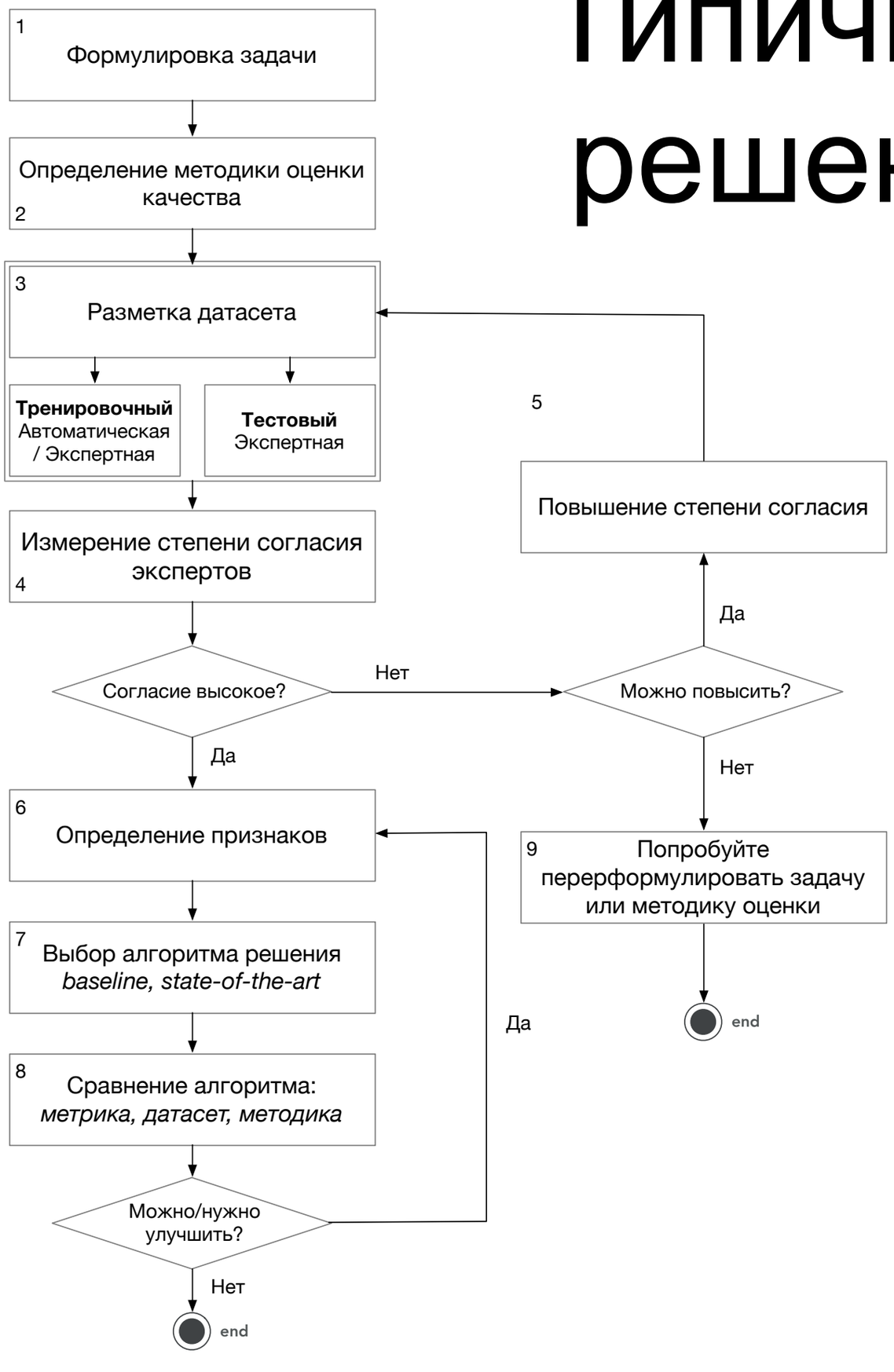
$$PP(w) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

- Для биграмм

$$PP(w) = \sqrt[N]{\prod_{i=1}^{n} \frac{1}{P(w_i|w_{i-1})}}$$

Типичная схема решения задач



- Способ оценки качества часто влияет на постановку задачи
- Тестировать алгоритм надо на данных, которые он никогда “не видел”
- Согласие экспертов показывает разрешимость задачи людьми и определяет *верхнюю границу* качества
- Для честного сравнения алгоритмов должны быть зафиксированы датасет и методика
- В реальных задачах шаги 1-5 занимают до 90% всего времени

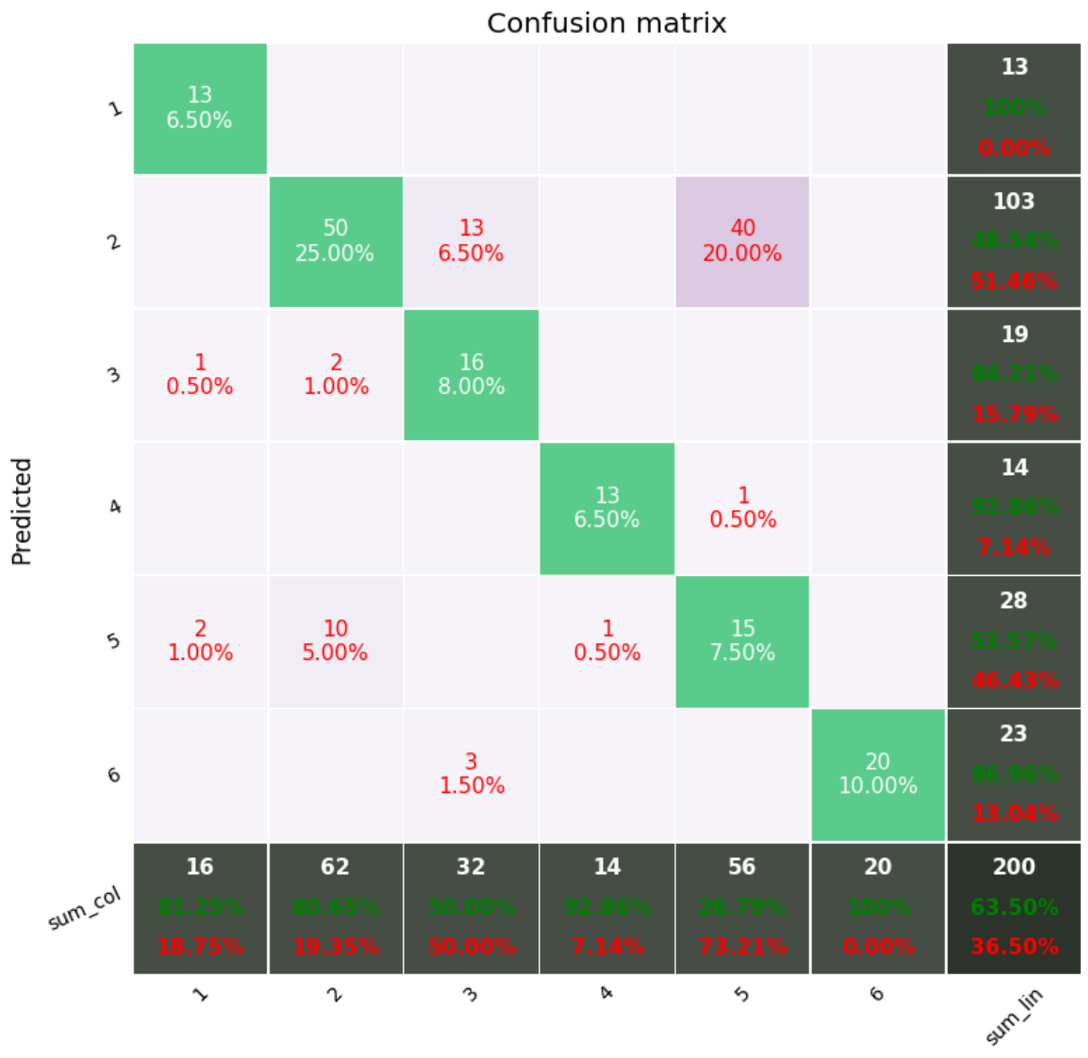
Пример

В начале 2019 года аппарат миссии NASA "Вояджер" (Voyager-2) пересек гелиопаузу и вышел в межзвездное пространство. Его брат-близнец (Voyager-1) сделал это шестью годами ранее. Зонды, запущенные в 1977-м для исследования планет-гигантов, до сих пор работоспособны, запасов радиоактивного топлива хватит до 2030 года.

- Определите ключевые слова текста

Матрица ошибок

- Confusion matrix
- Позволяет увидеть различие в ответах



https://github.com/wcipriano/pretty-print-confusion-matrix

Каппа Коэна

- Cohen's kappa
- Позволяет измерить согласие двух экспертов по сравнению со случайным совпадением ответов

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

p_o — наблюдаемое согласие

p_e — случайное согласие

Каппа Коэна

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

	Спам	Не спам	**Сумма**
Спам	20	3	**23**
Не спам	5	34	**39**
Сумма	**25**	**37**	**62**

$$p_o = \frac{a + d}{a + b + c + d} = \frac{20 + 34}{62} = 0.87$$

Каппа Коэна

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

	Спам	Не спам	**Сумма**
Спам	20	3	**23**
Не спам	5	34	**39**
Сумма	**25**	**37**	**62**

$$p_{\text{spam}} = \frac{a+b}{a+b+c+d} \cdot \frac{a+c}{a+b+c+d} = \frac{23}{62} \times \frac{25}{62} = 0.15$$

Каппа Коэна

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

	Спам	Не спам	**Сумма**
Спам	20	3	**23**
Не спам	5	34	**39**
Сумма	**25**	**37**	**62**

$$p_{\text{no spam}} = \frac{c + d}{a + b + c + d} \cdot \frac{b + d}{a + b + c + d} = \frac{39}{62} \times \frac{37}{62} = 0.38$$

Каппа Коэна

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

$$p_e = p_{\text{spam}} + p_{\text{no spam}} = 0.15 + 0.38 = 0.53$$

$$p_o = \frac{a + d}{a + b + c + d} = \frac{20 + 34}{62} = 0.87$$

$$\kappa = 1 - \frac{1 - 0.87}{1 - 0.53} = 0.72$$

Интерпретация

0.01 – 0.20	Незначительное
0.21 – 0.40	Удовлетворительное
0.41 – 0.60	Умеренное
0.61 – 0.80	Существенное
0.81 – 1.00	Почти идеальное

- Не всегда работает (в частности каппа может быть отрицательной)
- Нужно смотреть одновременно с матрицей ошибок
- https://idostatistics.com/cohen-kappa-free-calculator/

Задание

- Что делать, если оценщиков больше двух?
  - Самостоятельно изучить статистику “Каппа Флейса (Fleiss' kappa)”
  - Объяснить как с помощью нее оценить согласие для задачи определения ключевых слов (потенциальный вопрос на экзамене)

Резюме

- Обработка текста основана на формальных моделях (среди которых и языковые модели)
- Современная обработка текста преимущественно основана на статистическом подходе
- Основы обработки текста лежат в компьютерных науках, математике, лингвистике, электротехнике, психологии…
- Хороший способ понять проблемы обработки текстов и применимость методов - самостоятельно сделать решение для одной из прикладных задач обработки (машинный перевод, вопросно-ответную систему, разговорного агента)
- Революции в области происходят раз в ±4 года. Об этом поговорим в следующих лекциях

Дополнительные ресурсы

- Конференции: ACL, EACL, COLING, CoNLL, EMNLP, Диалог
- http://www.aclweb.org/anthology-new/
- Книги:
  - D. Jurafsky, J.H. Martin. Speech and Language processing.
  - C. Manning, H. Schutze. Foundations of Statistical Natural Language Processing
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep learning. MIT Press. 2016
- Курс 2021: https://www.youtube.com/playlist?list=PL5cBzMoPJgCXFdSvWaunOy4cILirW1lMD

Следующая лекция

- Методы классификации текстов
  - Логистическая регрессия
  - Машины опорных векторов (SVM)
  - Скрытая марковская модель (HMM, MEMM)
  - Условные случайные поля (CRF)
- Задача распознавания и классификации именованных сущностей (NERC)