

Lecture 2

Strings, language modelling, MC

Natural Language Processing
Ivan Smetannikov

11.03.2020

Acknowledgments

Лекции основаны на материалах Антона Михайловича Алексеева:

- <https://alexeyev.github.io/nlp-itmo-spring-2019>
- <https://my.compscicenter.ru/courses/introduction-nlp/2019-autumn/classes/>

План лекции

- String distances
- LM intro + N-grams
- MC + IT

Мотивация

Из промышленной разработки: Если можно избежать использования ML, то так и нужно делать!

Важно: алгоритмы на строках активно используются для подготовки данных и создания handcrafted features

Мотивация

Примеры:

1. Есть список названий компаний автоматически извлеченный из текста. Требуется сложить различные варианты написания одной и той же компании в один кластер без использования дополнительных баз знаний.
2. Исправление орфографических ошибок и опечаток с помощью словарей и статистики ошибок очень простыми методами на основе расстояний между строками и математической статистики.

Метрики на строках

В предположении отсутствия сдвигов (shifts) строк:
Hamming distance = вычисляем замены

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

Был придуман для подсчета точечных ошибок в бинарных кодах, тут используется для символов.

R	i	c	h	a	r	d
r	i	c	h	e	r	d

H	a	m	m	i	n	g	
H	a	m	m	m	i	n	g

Метрики на строках

Jaro similarity (1989)

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases}$$

m – число совпадающих символов.
совпадающих = положение
отличается не более чем на

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1$$

t – половина числа всех
совпадающих символов, в
которых все буквы в
неправильном порядке

B	A	E	N	X	I	E
B	A	N	K	S	E	Y

m = 4
t = 0
d = 0.71

Метрики на строках

Возможно небольшое число сдвигов:

Levenshtein distance

Минимальное число операций,
необходимое для преобразования
одной строки в другую: **insertions**,
deletions, **substitutions**.

Решается с помощью динамического
программирования

p	o	n	e	j	e
o	l	e	j	e	k

poneje - DEL
oneje - INS
onejek - SUB
olejek



d = 3

Метрики на строках

Максимальная общая подпоследовательность **Longest Common Subsequence (LCS)**

O	O	O	—	A	R	G	O	—	—	—
A	—	R	—	G	—	O	—	L	L	C

LCS = 4

Метрики на строках

Все метрики, описанные выше, в общем случае именуются редакционным расстоянием (**edit distances**, хотя обычно в русском языке под ним имеют ввиду расстояние Левенштейна), которое включает: **insertion, substitution, transpositions and deletions**

Реализации

Python

[nltk.metrics.distance](#)

python-Levenshtein

[Jellyfish](#)! (+ has **soundex**!)

...

+ Lucene (Java) has NgramIndex

Extra topic: regular expressions

Дополнительная информация

Если бы о строках можно было бы задавать какую-то дополнительную информацию

Например, чтобы подстрока содержала данные специального формата: номер телефона, электронная почта и тд

Применение

Требует осторожности

An [arguably] elegant weapon
for an [arguably] more civilized age!

Как только вы научитесь ими пользоваться, вы захотите их
использовать везде, но

- подходят не ко всему
(*don't parse XML with regex*),
- требует дополнительной поддержки в проде

Применение

В некоторых простых задачах NLP они будут наиболее quick-win решением

- Извлечение именованных сущностей
- Классификация текстов
- ...

RegEx: примеры

.	Any character but \n
\d	Digit
\D	Not a digit
\w	Letter, digit, _
\W	Not a letter or digit or _
\s	Whitespace char
\S	Not a whitespace char
\b	Word bound
\B	Not a word bound
^ \$	The beginning and the end of the string

Each regex sets a **language**:

... - any 3-char strings

\d\d\d - any 3-digit 'number' (may start with 0)

921\s\s\d\d\d\s\s - phone numbers of certain format

But how do we use full stop as a full stop?

Escaping!

Hello.\s - "Hello! ", "Hello. ", "Hello of "

Hello\.\s - just "Hello. "

RegEx: примеры

*	'Kleene star', repetition of the previous character 0+ times
?	Zero or one characters
+	Repetition, at least one time
{2}	Repetition, two times
{1,3}	Repetition from 1 to 3 times
{2,}	Repetition more that 1 time
[A-Za-z0-9шыж]	Any character listed inbraces
[^xyz]	Neither
ма(ма ть)	One of the groups separated with
[whatever]*?	? after repetition - "greedy" search

RegEx: советы

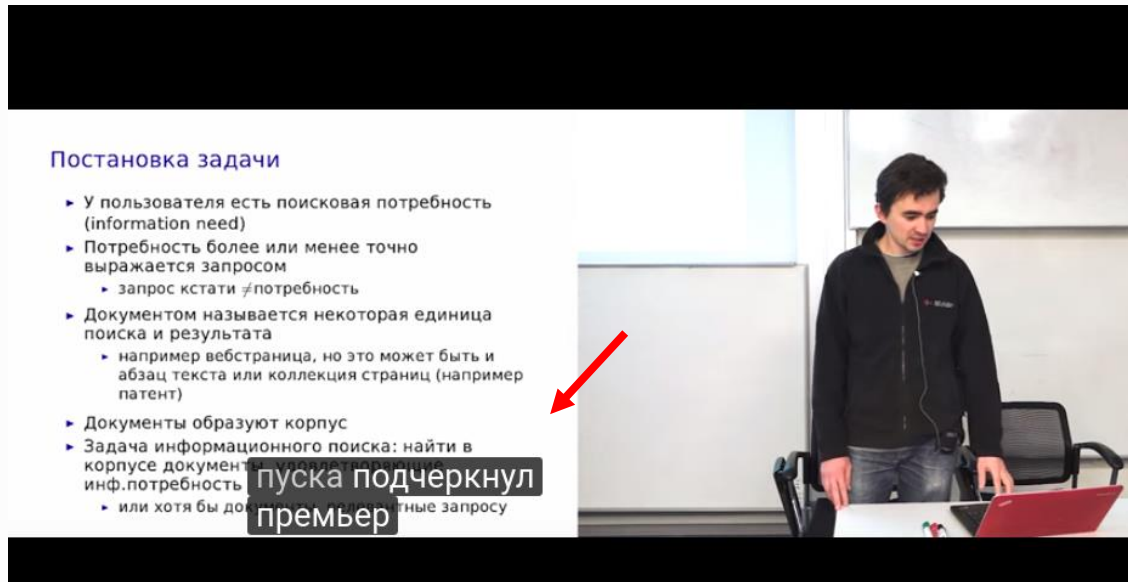
- По возможности переиспользовать
- Есть сомнения – поискать готовые решения и написать тесты
- Put some regex cheatsheets on the office's wall
- Диалекты Regex: POSIX, PCRE
- Если планируете использовать выражение несколько раз, например в цикле, то компилируйте их
- Практикуйтесь больше. Онлайн курс: <https://regexone.com/>

План лекции

- String distances
- LM intro + N-gram
- MC + IT

Мотивация

Во многих задачах требуется оценить является ли текст естественным. Иногда достаточно просто оценить вероятность последовательности слов.



Постановка задачи

- ▶ У пользователя есть поисковая потребность (information need)
- ▶ Потребность более или менее точно выражается запросом
 - ▶ запрос кстати \neq потребность
- ▶ Документом называется некоторая единица поиска и результата
 - ▶ например вебстраница, но это может быть и абзац текста или коллекция страниц (например патент)
- ▶ Документы образуют корпус
- ▶ Задача информационного поиска: найти в корпусе документ, удовлетворяющий инф. потребность
 - ▶ или хотя бы документ, релевантный запросу

пуска подчеркнул премьер

Дмитрий сказал на самом деле:

«поиск по патернам например»

<https://youtu.be/APcwsxUpGrQ?t=1m38s>

Мотивация

- **Распознавание речи / машинный перевод / исправление правописания / альтернативные коммуникации**
например: имеем несколько возможных вариантов декодирования фразы, выбираем наиболее вероятную с точки зрения языковой модели
- **Информационный поиск (IR)**
ранжирование: для каждого документа d мы строим его языковую модель и сортируем все документы по $P(q|d)$ (где q это запрос)
- **DN!** Генерация текстов, имитирующих стиль автора

Введение

- **Языковая модель** позволяет оценивать вероятность любой последовательности слов (или вероятность следующего слова)
- Как оценить вероятность той или иной фразы? Например с помощью условной вероятности.

Условная вероятность

- Условная вероятность

$$P(Y|X) = \frac{P(X, Y)}{P(X)} \Rightarrow P(X, Y) = P(Y|X)P(X)$$

- Цепное правило (chain rule) для большого числа переменных

$$P(x_1 x_2 \dots x_n) = P(x_n | x_1 \dots x_{n-1}) \dots p(x_2 | x_1) p(x_1)$$

- Как его вычислить?

$$P(x_i | x_1 \dots x_{i-1}) = \frac{\text{Count}(x_1 \dots x_{i-1} x_i)}{\text{Count}(x_1 \dots x_{i-1})}$$

* Тут и дальше обозначаем Count(...) таким же образом как C(...) и c(...)

Условная вероятность

- Условная вероятность

$$P(Y|X) = \frac{P(X, Y)}{P(X)} \Rightarrow P(X, Y) = P(Y|X)P(X)$$

- Цепное правило (chain rule) для большого числа переменных

$$P(x_1 x_2 \dots x_n) = P(x_n | x_1 \dots x_{n-1}) \dots p(x_2 | x_1) p(x_1)$$

- Как его вычислить?

$$P(x_i | x_1 \dots x_{i-1}) = \frac{\text{Count}(x_1 \dots x_{i-1} x_i)}{\text{Count}(x_1 \dots x_{i-1})}$$

$$P(\text{happy families are all}) = P(\text{all} | \text{happy families are}) \times \\ \times P(\text{are} | \text{happy families}) \times P(\text{families} | \text{happy}) \times P(\text{happy})$$

Условная вероятность

- Условная вероятность

$$P(Y|X) = \frac{P(X, Y)}{P(X)} \Rightarrow P(X, Y) = P(Y|X)P(X)$$

- Цепное правило (chain rule) для большого числа переменных

$$P(x_1 x_2 \dots x_n) = P(x_n | x_1 \dots x_{n-1}) \dots p(x_2 | x_1) p(x_1)$$

- Как его вычислить?

$$P(x_i | x_1 \dots x_{i-1}) = \frac{\text{Count}(x_1 \dots x_{i-1} x_i)}{\text{Count}(x_1 \dots x_{i-1})}$$

(нет, слишком длинная цепочка событий)

Что же делать?

- Предположение: текст удовлетворяет Марковскому свойству

$$P(x_i | x_1 \dots x_{i-1}) = P(x_i | x_{i-K} \dots x_{i-1})$$

Это значит, что текущее событие зависит не более чем от K предыдущих

- Примеры: $K = 0$ (unigram model)

$$P(\text{happy families are all}) =$$

$$P(\text{all}) \times P(\text{are}) \times P(\text{families}) \times P(\text{happy})$$

$K = 1$ (bigram model)

$$P(\text{happy families are all}) = P(\text{all} | \text{are}) \times$$

$$\times P(\text{are} | \text{families}) \times P(\text{families} | \text{happy}) \times P(\text{happy})$$

N-граммная модель

- Можем оценивать так:

$$P(x_i | x_{i-N+1} \dots x_{i-1}) = \frac{\text{Count}(x_{i-N+1} \dots x_{i-1} x_i)}{\text{Count}(x_{i-N+1} \dots x_{i-1})}$$

$$P(x_i | x_{i-1}) = \frac{\text{Count}(x_i, x_{i-1})}{\text{Count}(x_{i-1})}$$

- Пример для биграмм:

$$\begin{aligned} P(\text{hello}, i, \text{love}, \text{you}) &= \\ &= P(\text{hello} | \wedge) P(i | \text{hello}) P(\text{love} | i) P(\text{you} | \text{love}) P(\$ | \text{you}) \end{aligned}$$

Оценка качества моделей

- **Внешняя**

Проверяем модель на некоторой большой внешней задаче (перевод, исправление ошибок и проч.) Если целевая метрика (translators work time, editor's time, clicks count, earned money, etc.) растёт, модель стала лучше.

- **Внутренняя**

Когда внешняя оценка слишком сложна либо мы не хотим привязываться к конкретной предметной области.

Оценка качества моделей

- Не существует идеального корпуса данных в котором встречаются все возможные n -граммы!
- Описанная модель возвращает $P(x, \dots) = 0$ когда текст содержит хотя бы один n -грамм, который не встречался в обучающей выборке

Инструменты

nltk (nltk.models)

Moses (open source SMT engine)

Language Models in Moses

The language model should be trained on a corpus that is suitable to the domain. If the although using additional training data is often beneficial.

Our decoder works with the following language models:

- the [SRI language modeling toolkit](#), which is freely available.
- the [IRST language modeling toolkit](#), which is freely available and open source.
- the [RandLM language modeling toolkit](#), which is freely available and open source.
- the [KenLM language modeling toolkit](#), which is included in Moses by default.
- the [DALM language modeling toolkit](#), which is freely available and open source.
- the [OxLM language modeling toolkit](#), which is freely available and open source.
- the [NPLM language modeling toolkit](#), which is freely available and open source.

Наборы данных

- *Сырые наборы данных, специфичные для вашей задачи
- WMT
- Google NGrams
- National corpora (e.g. НКРЯ), OpenCorpora

План лекции

- String distances
- LM intro + N-gram
- MC + IT

Марковские модели

Обсуждаемые ранее N-граммные модели это Марковские модели

Марковское свойство: условное распределение следующего состояния стохастического процесса зависит только от текущего состояния

$$\mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = \mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n)$$

Процесс с дискретным временем (или последовательность случайных событий) называется Марковской цепью

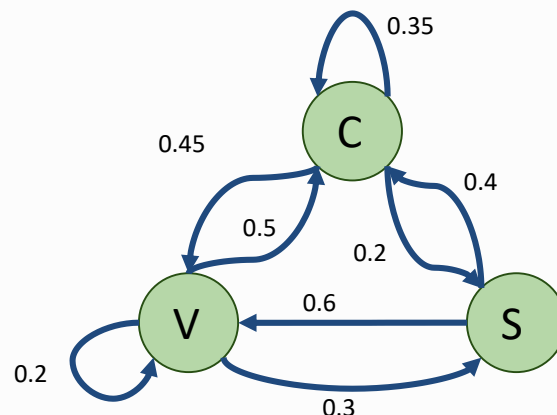
Марковская цепь

Модель задается стохастической матрицей = матрицей вероятностей переходов

Пример. События: vowel (v), consonant (c), whitespace/punctuation (s)

$P_{\text{trans}} =$

	v	c	s
v	0.2	0.5	0.3
c	0.45	0.35	0.2
s	0.6	0.4	0.0



DEMO: <http://antonalexeev.hop.ru/markov/index.html>

Марковская цепь

Марковская цепь задается матрицей вероятностей переходов и вероятностями изначальных состояний

$$\pi = (p_1^{(0)}, \dots, p_n^{(0)})^T$$
$$P_{trans} = \{p_{i \rightarrow j}, i, j \in 1 : n, \sum_{j=1}^n p_{i \rightarrow j} = 1 \forall i\}$$

Вероятность траектории длины один x_i

$$p = p_i$$

Вероятность траектории длины два $x_i \rightarrow x_j$

$$p = p(x_i)p(x_j|x_i) = \pi_i P_{i,j}$$

Вероятность траектории длины три $x_i \rightarrow x_j \rightarrow x_k$

$$p = p(x_i)p(x_j|x_i)p(x_k|x_i, x_j) = p(x_i)p(x_j|x_i)p(x_k|x_j) = \pi_i P_{i,j} P_{j,k}$$

Пример применения

N-граммная модель

- Можем оценивать так:

$$P(x_i | x_{i-N+1} \dots x_{i-1}) = \frac{\text{Count}(x_{i-N+1} \dots x_{i-1} x_i)}{\text{Count}(x_{i-N+1} \dots x_{i-1})}$$

$$P(x_i | x_{i-1}) = \frac{\text{Count}(x_i, x_{i-1})}{\text{Count}(x_{i-1})}$$

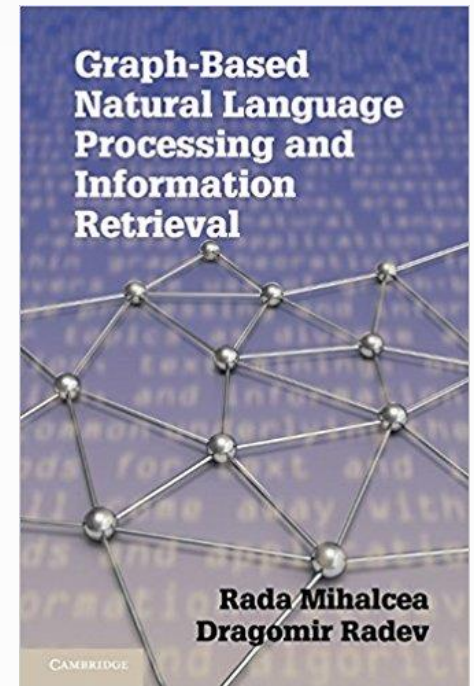
- Пример для биграмм:

$$\begin{aligned} P(\text{hello}, i, \text{love}, \text{you}) &= \\ &= P(\text{hello} | \wedge) P(i | \text{hello}) P(\text{love} | i) P(\text{you} | \text{love}) P(\$ | \text{you}) \end{aligned}$$

Альтернативный подход

Также можно решать задачу извлечения информации из текстов с помощью Graph-based NLP, книга 2011 года:

- graph theory
- probability theory
- linear algebra
- social networks analysis methods
- natural language processing



Примеры применения

- PageRank
- Language detection
- Named-entity recognition
- POS-tagging
- Speech recognition
- ...почти всегда может быть применен когда мы имеем дело с последовательностями

Теория информации

1948 - A Mathematical Theory of Communication,
Claude Shannon; основы теории информации

Имеет ряд приложений в алгоритмах сжатий,
криптографии, обработке сигналов и т.д.

Теория информации

Сколько информации содержит объект? Чем реже встречается событие, тем больше:

$$I(X) = -\log_2 p(x)$$

Например, если $p(x) = 1$

то $I(x) = 0$

Взаимная информация

Мера «объема общей информации» между X и Y

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right),$$

- Когда X и Y независимы – равна нулю
- Когда существует функциональная зависимость, превращается в энтропию X или Y

Имеет ряд приложений (feature selection)

Точечная взаимная информация

PMI

$$\text{pmi}(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}.$$

- PMI показывает объем дополнительной информации получаемой словом когда мы видим предыдущее слово
- Может применяться к словам не в последовательностях
- Дает больший вес редким фразам
- Имеет смысл использовать в качестве меры независимости (или как меры неслучайности совместного появления)

Взаимная информация

Подход с другой стороны

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right),$$



$$I(X; Y) = D_{\text{KL}}(p(x, y) \| p(x)p(y)).$$

Пример точечной взаимной информации

Извлечение: если слова появляются совместно немного реже, чем по отдельности, то они являются словосочетаниями; вероятности оцениваются частотой

Wikipedia, Oct. 2015

<i>word 1</i>	<i>word 2</i>	<i>count word 1</i>	<i>count word 2</i>	<i>count of co-occurrences</i>	PMI
puerto	rico	1938	1311	1159	10.0349081703
hong	kong	2438	2694	2205	9.72831972408
los	angeles	3501	2808	2791	9.56067615065
carbon	dioxide	4265	1353	1032	9.09852946116
prize	laureate	5131	1676	1210	8.85870710982
san	francisco	5237	2477	1779	8.83305176711
to	and	1025659	1375396	1286	-3.08825363041
to	in	1025659	1187652	1066	-3.12911348956
of	and	1761436	1375396	1190	-3.70663100173

https://en.wikipedia.org/wiki/Pointwise_mutual_information