Языковые модели (Language Models)

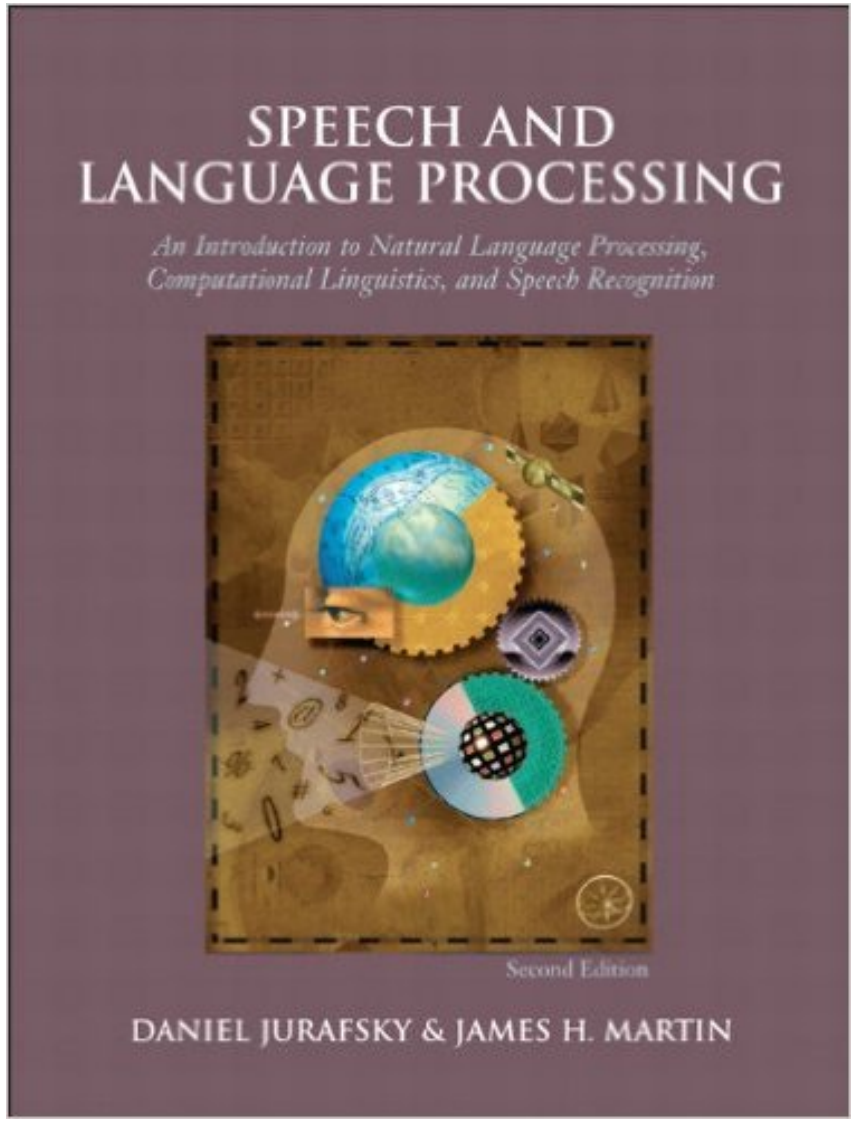
Елена Тутубалина

Kazan Federal University, Russia

22 февраля 2018

Литература

Speech and Language Processing, by Daniel Jurafsky and James H. Martin, 2008 `https://web.stanford.edu/~jurafsky/slp3/`



Лекция #3 курса “Введение в обработку естественного языка” П. Браславского – `https://stepik.org/course/1233/`

Введение

- “предсказание слов”:
  - задача — определить вероятность последовательности слов
- модель N-грамм — вероятностная модель предсказания последнего слова в цепочке из n слов
- связанная задача — подсчет вероятности цепочек слов
- юниграммы, биграммы, триграммы и т.д.
- модели N-грамм также называются языковыми моделями.

Пример

Во многих задачах бывает нужно проверить “естественность”.

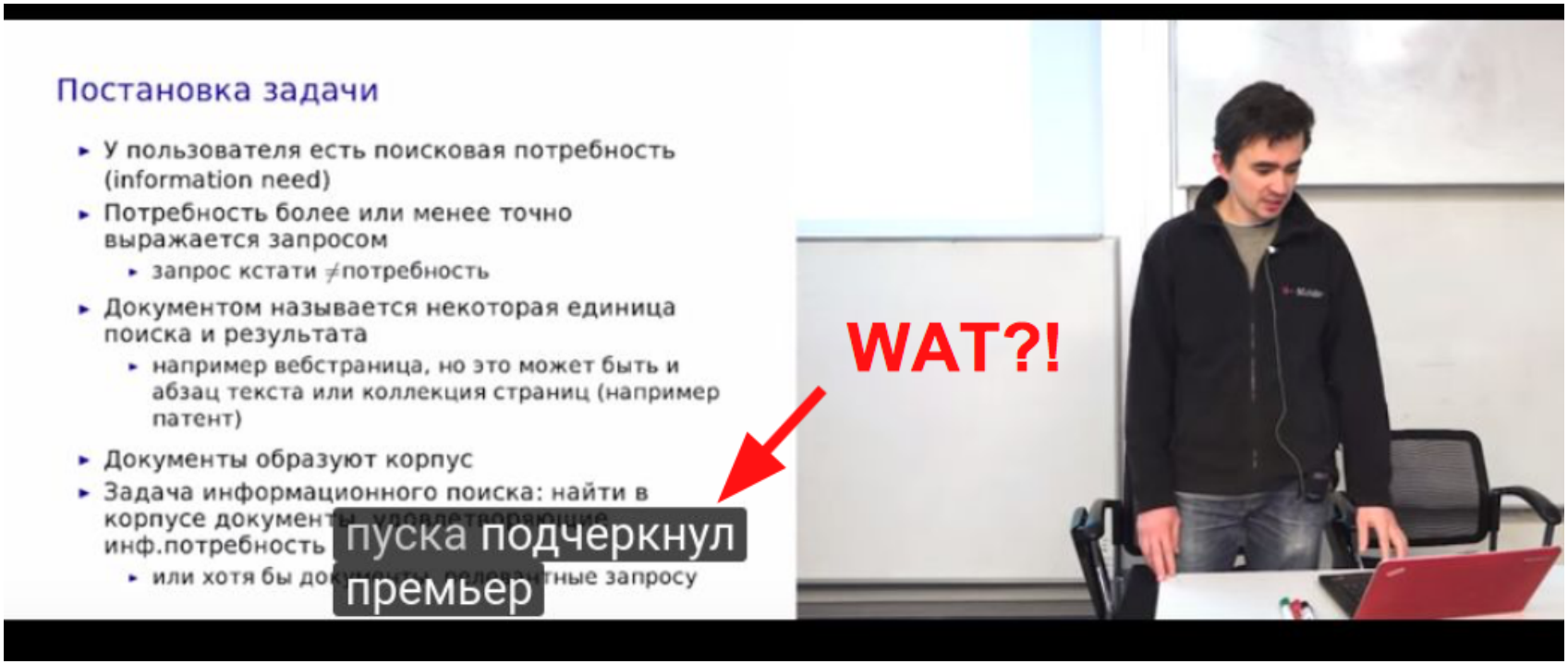
“Ожидаемая” вероятность

На острове Уайт неизвестные вынесли макет трицератопса из тематического парка при сувенирном магазине и бросили его посередине одной из местных дорог.

Цепочка с меньшей вероятностью

тематического неизвестные бросили На макет вынесли Уайт трицератопса дорог парка посередине одной из из сувенирном местных его при магазине и острове

Пример



На деле Дмитрий говорит: *...поиск по патентам, например.*
https://youtu.be/APcwsx UpGrQ?t=1m38s

Применения языковых моделей

- распознавание речи
  - *скрипка лиса* VS *скрип колеса*
  - *I saw a van* VS *eyes awe of an*
- OCR, т.е., распознавание рукописного или печатного текста
- машинный перевод
  - *сильный чай* VS *крепкий чай*
- корректировка правописания (spelling correction)
  - *курсовая работа* VS *курсовая робота*
- augmentative communication
- predictive text input (например, виртуальные клавиатуры)
- идентификация авторства
- и т.д.

Подсчёт слов: более формально

Цель

вычислить вероятность предложения или последовательности слов: $P(W) = P(w_1, w_2, w_3, w_4, w_5, \cdots, w_n)$

Связанная задача

вычислить вероятность следующего слова: $P(w_5 | w_1, w_2, w_3, w_4)$

Корпус

машиночитаемая коллекция текстов

Что считать “словом” в цепочках?

- “поверхностную форму” ?
- окончания? лемму (нормальную форму слова)?
- включать или не включать пунктуацию?
- как быть с сокращениями?

Примеры корпусов

- Национальный корпус русского языка
- Opencorpora.org
- Russian Internet Corpus
- Google Books N-Grams (не корпус)
- Google Web 1T 5-gram (не корпус, English)
- сотни прочих, см. `https://habrahabr.ru/post/152799/`
- сотни прочих, см. `https://github.com/niderhoff/nlp-datasets`

Через небольшие усилия корпус можно получить из:

- Wikipedia
- Common Crawl

Базовая модель: мотивация

$$P(\text{прокат}|\text{фильм выйдет в российский}) =$$

Базовая модель: мотивация

$$P(\text{прокат}|\text{фильм выйдет в российский}) =$$

$$= \frac{Count(\text{фильм выйдет в российский прокат})}{Count(\text{фильм выйдет в российский})}$$

Базовая модель: мотивация

$$P(\text{прокат}|\text{фильм выйдет в российский}) =$$

$$= \frac{Count(\text{фильм выйдет в российский прокат})}{Count(\text{фильм выйдет в российский})}$$

- “слабый” подход, т.к. язык – созидательный
- слишком много разных предложений
- данных может быть и не достаточно для подсчета всех значений

Идея

Используем цепное правило для подсчета P(фильм, выйдет, в, российский, прокат)

Нотация

- упростим $P(X_i = \text{“фильм”})$ до $P(\text{“фильм”})$
- последовательность слов $w_1 w_2 \dots w_n$ или w_1^n
- совместная вероятность $P(X = w_1, Y = w_2, \dots, Z = w_n)$ или $P(w_1, w_2, \dots, w_n)$

Условная вероятность

$P(B|A) = \frac{P(A,B)}{P(A)}$ -> $P(A,B) = P(A)P(B|A)$

$P(A,B,C,D) = P(A)\ P(B|A)\ P(C|A,B)\ P(D|A,B,C)$

Цепное правило (The Chain Rule)

$$P(X_1 \dots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1^2) \dots P(X_n|X_1^{n-1})$$

$$= \prod_{k=1}^{n} P(X_k|X_1^{k-1})$$

Применяем к цепочке слов:

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

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

P(фильм выйдет в российский прокат) = P(фильм) P(выйдет|фильм) P(в|фильм выйдет) P(российский|фильм выйдет в) P(прокат|фильм выйдет в российский)
Пока легче не стало!

Марковское свойство (предположение Маркова)

Условное распределение вероятностей будущих состояний зависит только от нынешнего состояния, а не от последовательности событий, которые предшествовали этому.

На примере *биграмм* (т.е., предположение первого порядка)

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

то есть:

$$P(\text{прокат}|\text{фильм выйдет в российский}) \approx$$

Марковское свойство (предположение Маркова)

Условное распределение вероятностей будущих состояний зависит только от нынешнего состояния, а не от последовательности событий, которые предшествовали этому.

На примере *биграмм* (т.е., предположение первого порядка)

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

то есть:

$$P(\text{прокат}|\text{фильм выйдет в российский}) \approx$$

$$\approx P(\text{прокат}|\text{российский})$$

Марковское свойство (предположение Маркова)

Условное распределение вероятностей будущих состояний зависит только от нынешнего состояния, а не от последовательности событий, которые предшествовали этому.

На примере *биграмм* (т.е., предположение первого порядка)

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

то есть:

$$P(\text{прокат}|\text{фильм выйдет в российский}) \approx$$

$$\approx P(\text{прокат}|\text{российский})$$

Для N-го порядка:

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

Упрощение с помощью марковского свойства

Для биграмм:

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

$P(y|x)$ для любых двух слов x и y являются параметрами модели. Внедряя марковское свойство мы значительно уменьшаем число параметров модели, которые необходимо посчитать.

Модели на основе n-грамм:

- униграммная $P(w_k)$
- биграммная $P(w_k|w_{k-1})$
- триграммная $P(w_k|w_{k-1}, w_{k-2})$

Оценивание методом максимального правдоподобия

Maximum Likelihood Estimation (MLE)

Для биграмм:

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

Для общего случая:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

Пример

<s> роман «Война и мир» бьет рекорды продаж в Великобритании </s>
<s> сериал «Война и мир» стал бестселлером в Великобритании сразу после показа </s>
<s> «Война и мир» стала бестселлером в Великобритании </s>
<s> «Война и мир» стала бестселлером в Великобритании после экранизации </s>
<s> Экранизация «Би-би-си» сделала книгу «Война и мир» бестселлером в Великобритании </s>

$P(\text{роман}|\text{<s>}) = ?$
$P(\text{Экранизация}|\text{<s>}) = ?$
$P(\text{бестселлером}|\text{стала}) = ?$
$P(\text{Великобритании}|\text{в}) = ?$

Пример: Berkeley Restaurant Project corpus

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Частоты биграмм

9222 предложения

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Частоты биграмм

Нормализация с помощью юниграмм:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

Результат:

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

О чем говорят вероятности?

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat|to) = .28
- P(food|to) = 0
- P(want|spend) = 0
- P (i|<s>) = .25

P(<s> I want english food </s>) = P(I|<s>) * P(want|I) *
P(english|want) * P(food|english) * P(</s>|food) = .000031

Практические соображения

лучше оперировать логарифмами вероятностей:

- избежать переполнения
- сложение быстрее, чем умножение

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

Оценивание: обучающий и тестовый корпус

Обучающее - для вычисления значений параметров модели

Тестовое - для вычисления оценок качества

Отладочное (development) - для вычисления оценок качества, во избежание “перенастройки” приводящей к неправдоподобно хорошему результу на тестовом

Также (как альтернатива фиксированному разбиению или дополнение) может применяться метод *перекрёстной проверки (cross-validation)*.

Проблема “”неизвестных” слов

Вариации постановки задачи:

С закрытым лексиконом – предполагаем, что все слова заранее известны

С открытым лексиконом – обучающий корпус не содержит всех слов.

”Неизвестные“ слова
(Out-of-vocabulary, OOV), слова, которые не встретились в обучающем множестве. При обучении модели или её работе заменяются на <UNK>.

Учет неизвестных слов при обучении модели

1. зафиксировать лексикон V (по существующему словарю, или оставить слова с частотой выше порого, и др.),
2. заменить все слова вне V в обучающем корпусе на <UNK>,
3. далее – обходиться с <UNK> как с обычным словом.

Альтернатива – заменить первое вхождение каждого слова в обучающем корпусе на <UNK>.

Оценивание

Два подхода к оцениванию качества:

Внешнее – оценивается улучшение решения конечной прикладной задачи

Внутреннее – оценка, не зависящая от предполагаемого применения.

Perplexity (”показатель связности“):

$$PP(W) = \sqrt[n]{\frac{1}{P(w_1 w_2 \dots w_n)}}$$

Для биграммной модели:

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

Заключение

- термины, введённые на лекции: n-граммы, языковая модель, корпус, тестовое и обучающее множества, марковское свойство, метод максимального правдоподобия, perplexity
- следующая лекция будет про методы сглаживания

Методы сглаживания: Введение

- языковая модель должна *обобщать* (а не повторять) данные, на которых она обучалась
- проблема *“разреженных данных”* (sparse data)
  - всегда будут новые последовательности слов, которые не встречались в корпусе для обучения
- при MLE => большое число “нормальных” n-грамм получает нулевую вероятность
- при MLE => большое число слишком малых вероятностей
- нулевая вероятность ведёт к нулевой perplexity и “ломает” модель.

Сглаживание по Лапласу

Идея сглаживания

“зарезервировать” часть вероятностной массы для событий, которые еще не встречались

- суть – добавить по 1 частоты каждой n-грамме перед нормализацией
- “учебный” базовый метод, не применяется на практике

Для юниграмм

$$P_{MLE}(w_i) = \frac{c_i}{N}$$

$$P_{Laplace}(w_i) = \frac{c_i + 1}{N + V}$$

где V – количество типов слов.

Как бы выглядела “исправленная” частота:

$$c_i^* = (c_i + 1)\frac{N}{N + V}$$

Пример

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Вероятности

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Для биграмм

$$P_{MLE}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$$

$$P_{Laplace}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) + 1}{C(w_{i-1}) + V}$$

Откат и интерполяция

- Sometimes it helps to use less context
  - Condition on less context for contexts you haven't learned much about
- Backoff:
  - use trigram if you have good evidence, otherwise bigram, otherwise unigram
- Interpolation:
  - mix unigram, bigram, trigram
- Interpolation works better

Интерполяция

- Simple interpolation:

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

- Lambdas зависят от контекста:

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

Дисконтирование

Альтернативное видение сглаживания – перераспределение частот (соответственно, и вероятности) за счет её уменьшения у “известных” n-грамм.

$$d = \frac{c^*}{c}$$

Оценивание популяции особей

Частота по MLE: c

		MLE	GT
	басс	0	?
	карп	10	?
	сом	0	?
	угорь	1	?
	окунь	3	?
	лосось	1	?
	форель	1	?
	треска	2	?

- Какова вероятность, что следующая рыба – форель? 1/18
- Какова вероятность поймать новую рыбу (сома или щуку)?
  - Предположим, что вероятность поймать новую равна вероятности поймать рыбу, которая до сих пор попадалась только один раз: 3/18
  - Тогда какова вероятность поймать форель? $< 1/18$

Сглаживание Гуд-Тьюринга

(Good-Turing)
Суть – использовать частоту n-грамм, встретившихся один раз, для того, чтобы оценить частоту n-грамм, не встретившихся ни разу.
N_c – количество типов n-грамм, встречающихся c раз
Частота по MLE: c
Частота по GT: $c^* = (c+1)\frac{N_{c+1}}{N_c}$

$$P^*_{GT}(\text{всех n-грамм с нулевой частотой}) = \frac{N_1}{N}$$

Посчитать оценки

- не видели (сом или щука)
- Видели однажды (форель)

Пример

Частота по MLE: c
Частота по GT: $c^* = (c+1)\frac{N_{c+1}}{N_c}$

- не видели (сом или щука): c = 0
  - MLE p = 0/18 = 0
  - P_{GT}(не видели) = $\frac{N1}{N}$ = 3/18
- видели однажды (форель): c = 1
  - MLE p = 1/18
  - C^*(форель) = 2 * $\frac{N2}{N1}$ = 2 * 1/3 = 2/3
  - P_{GT}(форель) = $\frac{2/3}{18}$ = 1/27

Гуд-Тьюринга – продолжение

- предполагает что N_0 известно
- допустим V - количество слов, чему равно N_0 в биграммной модели? V^2 – количество “видимых” n-грамм
- проблема при встрече $N_{c+1} = 0$. Один из методом решения – Simple Good-Turing:
  1. посчитать N_c
  2. пересчитать N_c методом линейной регрессии для:

$$\log(N_c) = a + b \log(c)$$

- на практике можно остановить перерасчет c^* после $c >$ некоего порога k

Задание 3

- Написать код оценивания методом максимального правдоподобия на примере юниграм и биграм (без сглаживания).
- Запустить программу на 2 корпусах разной тематики. Сравнить статистику.
- Полезный notebook: https://github.com/krishnamrith12/NotebooksNLP/blob/master/4.LanguageModels.ipynb
- Еще один полезный notebook: https://github.com/kjmazidi/NLP_class/tree/master/NLP_demosp

Задание 3 – Корпуса

- https://github.com/niderhoff/nlp-datasets
- https://github.com/awesomedata/awesome-public-datasets (NaturalLanguage part)
- Новости (1000 шт) (Ru) http://labinform.ru/pub/named_entities/descr_ne.htm
- Hotel-Review Datasets (Eng) http://www.cs.cmu.edu/ jiweil/html/hotel-review.html
- Отзывы о лекарствах (Eng) https://cimm.kpfu.ru/seafile/f/4ce6752436bd4a46904c/?dl=1
- Отзывы о больницах (Ru) https://cimm.kpfu.ru/seafile/d/ae6283ce2c80489496b6/