ТЕХНОСФЕРА

Лекция №13

Исправление опечаток в запросах

Евгений Чернов

Очепяточник

Ошибки поисковых запросов

Запрос “*одноклассники*”:

- *однокла**с**ники*
- ***jlyjrkfccybrb***
- *одн**а**классники*
- *одноклacc*
- *одн**а**кла**с**ники*
- ***jlyjrkfcybrb***
- *однок**лс**сники*
- *о**дк**ла**с**ники*
- *одноклассни*
- *одно**ка**ссники*

Типы ошибок: орфография

- Ошиблись в букве:
  - *вк**а**нтакте*
  - *ж**ы**вые обои*
  - *к**о**талог орифлейм*
- Пропустили букву:
  - *однокла**с**ники*
  - *самые качестве**н**ые свечи зажигания*
  - *с**кч**ать игру*
- Лишняя буква
  - *тан**ы**цы айренби видео*
- Поменяли местами буквы
  - *скач**та**ь медиагет*
  - *купить б**ар**слет*

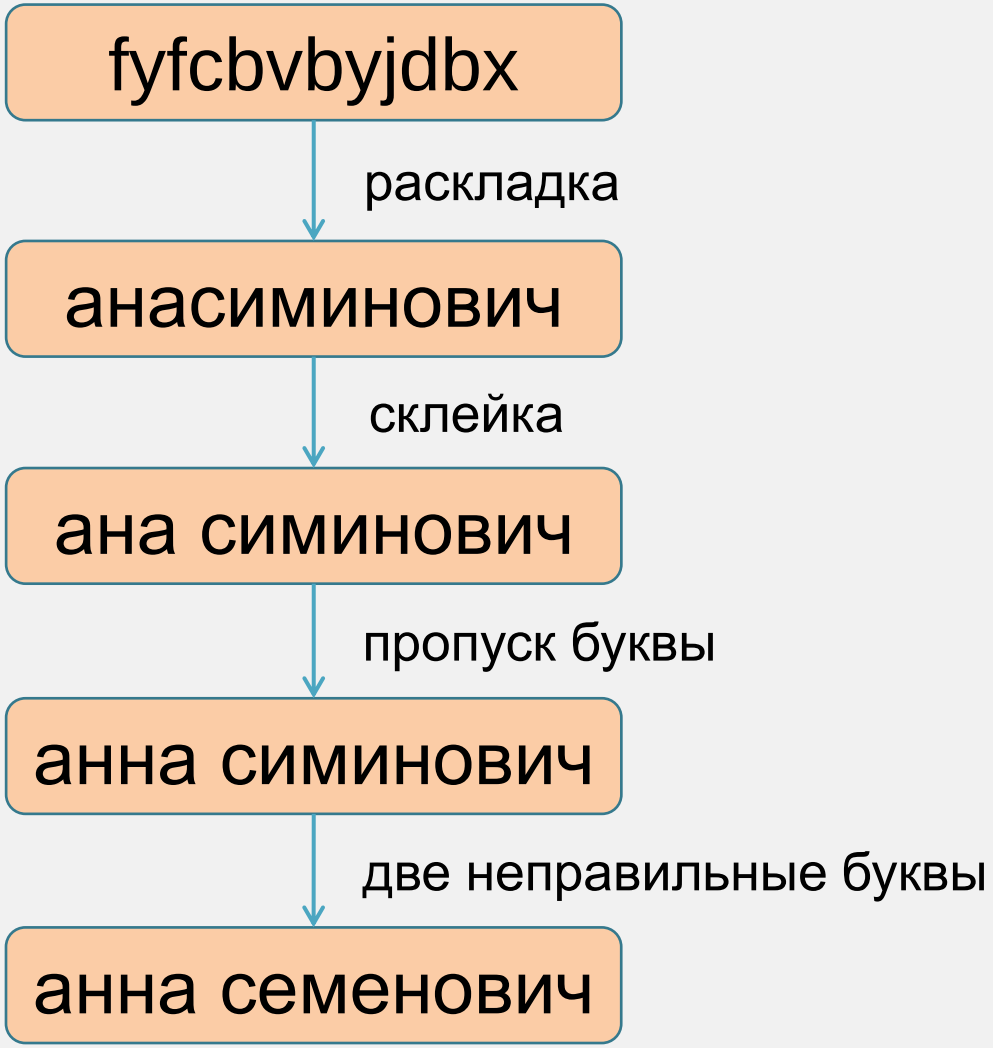
Типы ошибок: пробелы

- Пропустили:
  - *голос 5 сезон **6в**ыпуск*
  - *замен**аз**адней крестовины*
- Поставили лишний:
  - *что скачать чт**о б**ы открывалась презентация*
  - *ма**к б**ук и вирусы*

Типы ошибок: раскладка / транслит

- Раскладка:
  - jlyjrkfccybrb (одноклассники)
  - мл (vk)
- Транслитирация:
  - kupit televizor (купить телевизор)
  - мейл ру (mail ru)

Типы ошибок: смешанные

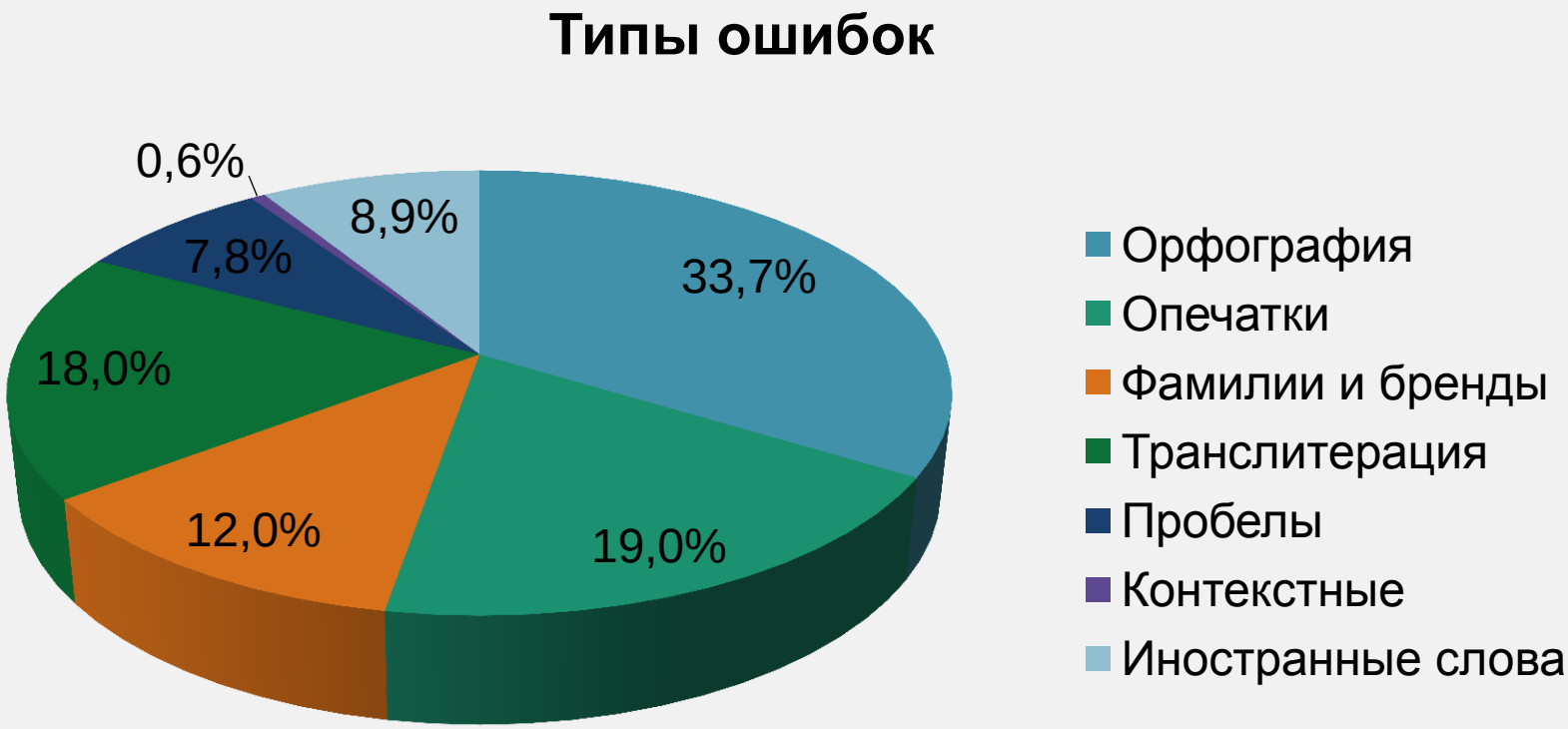


Типы ошибок: сложные случаи

- Ошибка или нет?
  - пагода -> погода (буддийское сооружение культового характера)
  - vnc -> мтс (Virtual Network Computing)
- Ошибка или название домена:
  - rfcnh.kz -> кастрюля (сайт в казахстане)
- Ошибка или модель изделия:
  - крюк для укладки труб RHSV (КРЫМ)
- Учет контекста:
  - клон (нет ошибки)
  - африканский **к**лон -> африканский слон

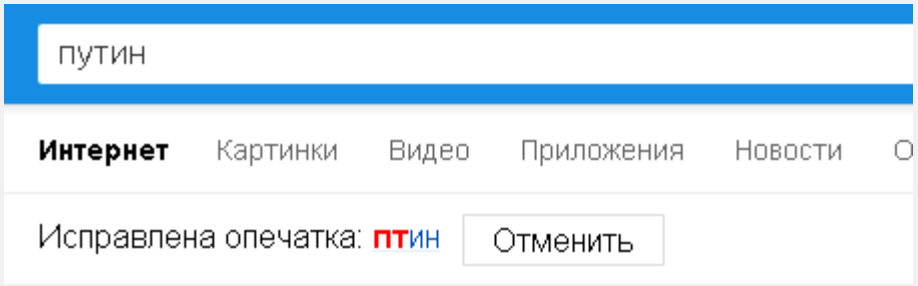
Статистика ошибок

~ 11% запросов в потоке имеют ошибку

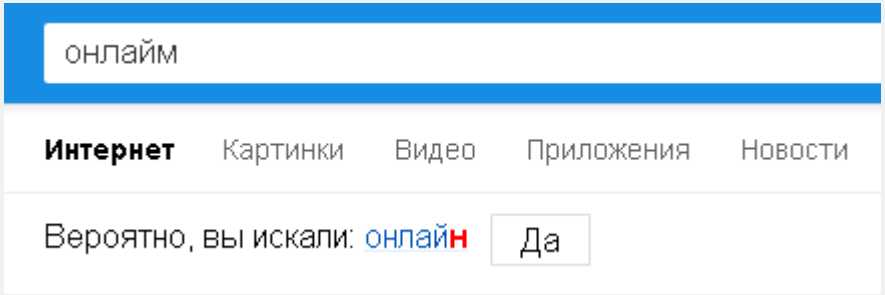


Типы исправлений

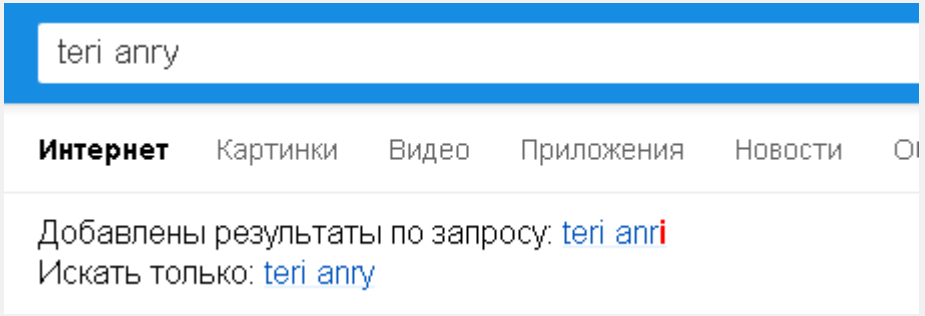
- Авто-исправление



- Подсказка



- Смешение



Простой поиск очепаток

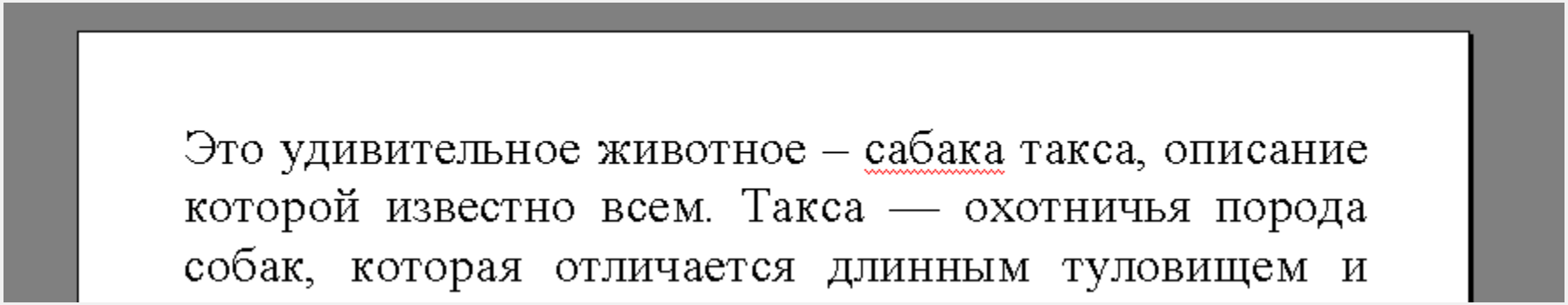
1. Ищем слово в словаре
2. Если его там нет, значит оно содержит ошибку:

Простой поиск очепаток

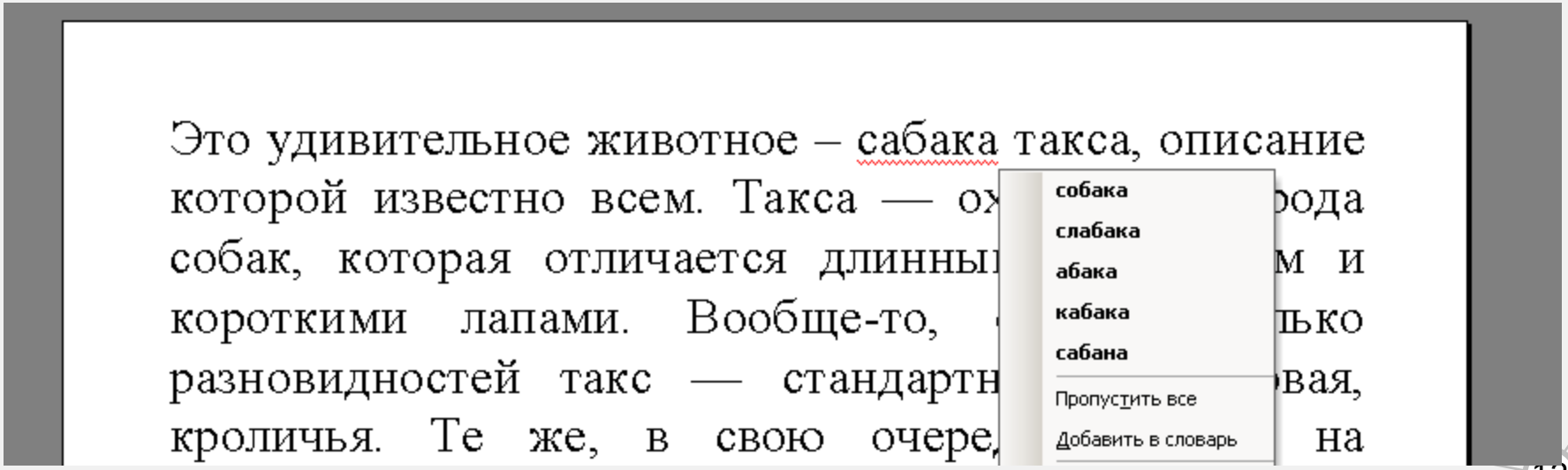
3. Ищем ближайшее слово в словаре, чтобы предложить правильный вариант слова

Простой опечаточник

Активно используется в текстовых редакторах:

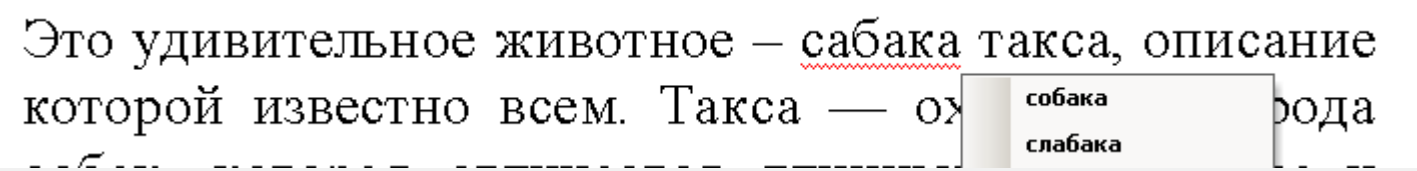


Варианты из словаря (можно пополнять):



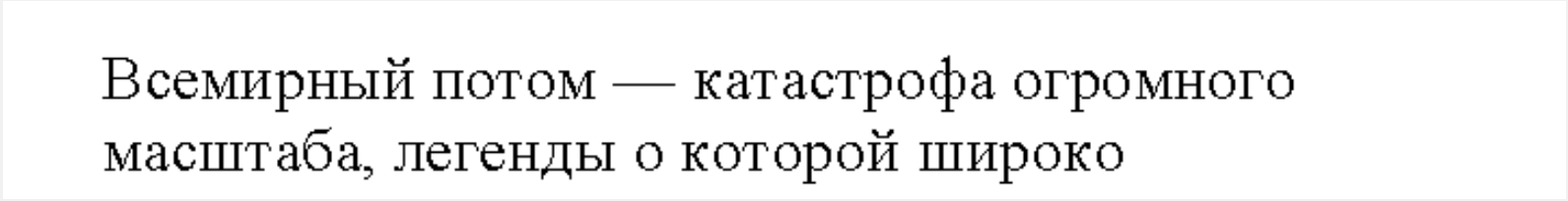
Простой опечаточник: недостатки

Не учитывается контекст:



Неполный словарь:

В слове есть ошибка, но оно есть в словаре:



Простой опечаточник: задачи

Формирование словаря

- Орфографический словарь
  - для разных языков свой
  - учет морфологии
- Слова из наборов текстов (литература, новостной контент, запросы)
  - недостаточная полнота
  - как избавиться от ошибок?

Поиск ближайших слов

- Расстояние Левенштейна

Расстояние Левенштейна

- Дистанция редактирования – минимальное количество операций вставки (**I**nsert), удаления (**D**elete) или замены одного символа на другой (**R**eplace), необходимых для превращения одной строки в другую
- CONNECT -> CONEHEAD:

M	M	M	R	R	R	R	I
C	O	N	N	E	C	T	
C	O	N	E	H	E	A	D

- Расстояние: 5

Расстояние Левенштейна

Saturday -> Sunday:

1. Delete **a**
2. Delete **t**
3. Replace **r** to **n**

		S	**a**	**t**	**u**	**r**	**d**	**a**	**y**
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	3

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Расчет расстояния Левенштейна

		Б	О	Е	Ц
	0	1	2	3	4
Б	1	0	1	2	3
Е	2	1	1	1	2
Р	3	2	2	2	2
Ц	4	3	3	3	2
Ы	5	4	4	4	3

1. Замена Е на О
2. Замена Р на Е
3. Удаление Ы

1. Добавление О после Б
2. Удаление Р
3. Удаление Ы

Расстояние Левенштейна: вариации

Учет перестановки букв (расстояние Дамерау-Левенштейна):

- расст**яо**ние -> расст**оя**ние:
  - 2 операции замены букв -> 1 операция транспозиции

Учет неправильной раскладки:

- hfccnjzybt -> расстояние:
  - 10 операций замены букв -> 1 операция смены раскладки

Разный веса операций:

- удаление: 0,8
- вставка: 1,2

Разные веса для разных символов:

- w(п, р) = 0,2
- w(п, ъ) = 0,9

Расстояние Левенштейна: взвешенное

Диктант: *На террасе Агриппина Саввишна исподтишка потчевала моллюсками и винегретом коллежского асессора.*

Результат: *На террассе Агреппина Саввишна исподтшика потчивала молюсками и венегретом коллежского ассессора.*

$$P_{\text{замены}}(\text{и}, \text{е}) = \frac{2}{8} = 0{,}25$$

$$P_{\text{удаления}}(\text{л}, \text{л}) = \frac{1}{2} = 0{,}5$$

$$P_{\text{перестановки}}(\text{и}, \text{ш}) = \frac{1}{2} = 0{,}5$$

$$P_{\text{вставки}}(\text{с}, \text{с}) = \frac{2}{8} = 0{,}25$$

Расстояние Левенштейна: взвешенное

Диктант:

- сложно
- дорого
- малая полнота

Логи запросов:

- меньшая точность
- большая полнота

смортеть безплатно филмы -> смотреть бесплатно фильмы

Простой опечаточник: переход к вероятностям

Как выбрать из нескольких вариантов с одинаковым расстоянием Левенштейна:

расстояние 1 от “поток”:

{пото**м**, пото**п**, поро**к**, п**я**ток, **л**оток, **м**оток, **р**оток}

Как учесть частотность запроса:

- африканский крон -> африканский к**л**он (1 лев.)
- африканский крон -> африканский **сл**он (2 лев.)

Переходим к вероятностям!

Опечаточник: формальная постановка задачи

orig – запрос, который ввел пользователь (возможно содержит опечатку)

fix – исправленный запрос пользователя

D – словарь (множество) всех возможных запросов

$$fix^* = \underset{fix \in D}{\operatorname{argmax}}\big(P(fix|orig)\big)$$

Вычисление вероятности

$$P(fix|orig) = \frac{P(orig|fix)P(fix)}{P(orig)}$$

Модель ошибок (Error Model)

- $P(orig|fix)$ – вероятность того, что пользователь напишет запрос orig, когда хотел написать запрос fix
- Вероятность должна быть связана с близостью запросов (например, с расстоянием Левенштейна)
- Вероятность от 0 до 1
- $P(orig|fix) = \alpha^{-lev(orig,fix)}$
- α – коэффициент, который подбирается оптимизацией

Модель ошибок (Error Model)

Saturday -> Sunday:

S – S : 1

a - _: 1

t - _: 1

u – u: 1

r – n: 1

d – d: 1

a – a: 1

y – y: 1

		S	**a**	**t**	**u**	**r**	**d**	**a**	**y**
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	3

Модель ошибок: биграмная статистика

президент ->
перзидеед

^п - ^п : 1

п_ - пе: 1

_р – ер: 1

ре – р_ : 1

ез - _з: 1

зи – зи: 1

ид – ид: 1

де – де: 1

ен – ее: 1

нт – ед: 1

Модель языка (Language Model)

- Некоторая статистика языка
- В нашем случае построенная по запросам
- Язык меняется:
  - ip**a**d -> ip**o**d
  - последний из м**а**гик**я**н -> последний из м**о**гик**а**н
- Статистика собирается за определенный период
- Должна периодически обновляться

Модель языка (Language Model)
Как посчитать P(query)

1. $\text{частотность запроса} = \frac{\text{сколько раз вводили запрос}}{\text{общее количество запросов}}$
   - работает для достаточно частотных запросов
   - запросы с ошибками как правило низкочастотные
2. разбиваем на слова: $query = w_1 w_2 w_3 \ldots w_n$
   - Униграммная модель (слова не зависят друг от друга):

$$P(query) = P(w_1 w_2 w_3 \ldots w_n) = P(w_1)P(w_2)\ P(w_3)\ldots\ P(w_n)$$

   - Биграммная модель: $P(w_1 w_2) = P(w_1|w_2)P(w_2)$

$$P(query) = P(w_1 w_2 w_3 \ldots w_n) = P(w_1|w_2)P(w_2|w_3)\ldots\ P(w_n)$$

3. Разбиение запроса на буквы/слоги

Поиск исправления

Какая осталась проблема?

$$fix^* = \underset{fix \in D}{\operatorname{argmax}}\big(P(fix|orig)\big) = \underset{fix \in D}{\operatorname{argmax}}\left(\frac{P(orig|fix)P(fix)}{P(orig)}\right)$$

Как перебрать все варианты исправлений?

Генерация кандидатов исправлений

1. Разбиваем запрос на части
2. Для каждой части составляем список замен
3. Оцениваем вес каждой замены
4. Составляем граф слов
5. Находим оптимальный путь в графе

Разбиение запроса на части

По пробелам и знакам препинания

Достоинство:

- Однозначный способ разбиения
- Простота реализации

Недостатки:

- Невозможно склеить две части слова:

  нижний нов город –> нижний новгород

- Лишнее разбиение в перераскладке:

  yb;ybq –> yb ybq –> ни ний

Генерация вариантов замены: Soundex

Soundex - алгоритм нахождения фонетических альтернатив

Например: *chebyshev / tchebyscheff*

Алгоритм:

- Превратить каждый токен в 4-х символьную сокращённую форму
- То же самое сделать для терминов запроса
- Построить и использовать отдельный индекс сокращённых форм

Алгоритм Soundex

1. Оставим первый символ термина
2. Следующие символы заменяются на ’0’ : A, E, I, O, U, H, W, Y
3. Заменить символы на цифры:
   - B, F, P, V на 1
   - C, G, J, K, Q, S, X, Z на 2
   - D,T на 3
   - L на 4
   - M, N на 5
   - R на 6
4. Повторно удалять по цифре из последовательных повторов
5. Удалить все нули
6. Добавить в конец нули до 4 символов

Soundex для HERMAN

Оставим H

ERMAN → 0RM0N

0RM0N → 06505

06505 → 06505

06505 → 655

Результат: H655

Для HERMANN будет сгенерирован тот же код.

Soundex для русских слов

Нахождение сигнатуры слова режиссер:

- убрать повторяющиеся буквы: режисер
- оставить только согласные: ржср
- оглушить звонкие согласные: ршср

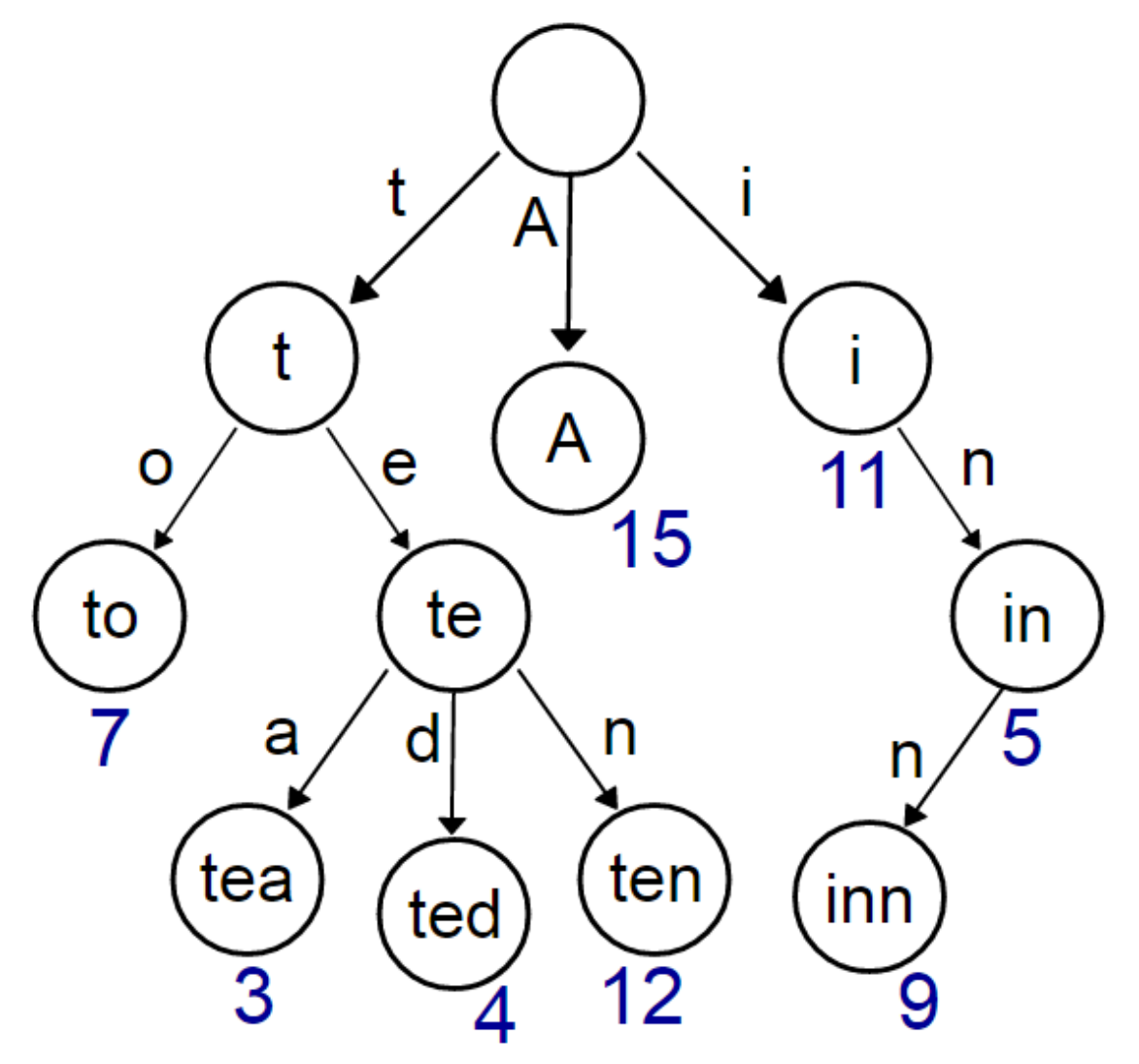
Sign(режесер) = ршср

Words(ршср) = [режиссер, рыжая серии, оружие зорро]

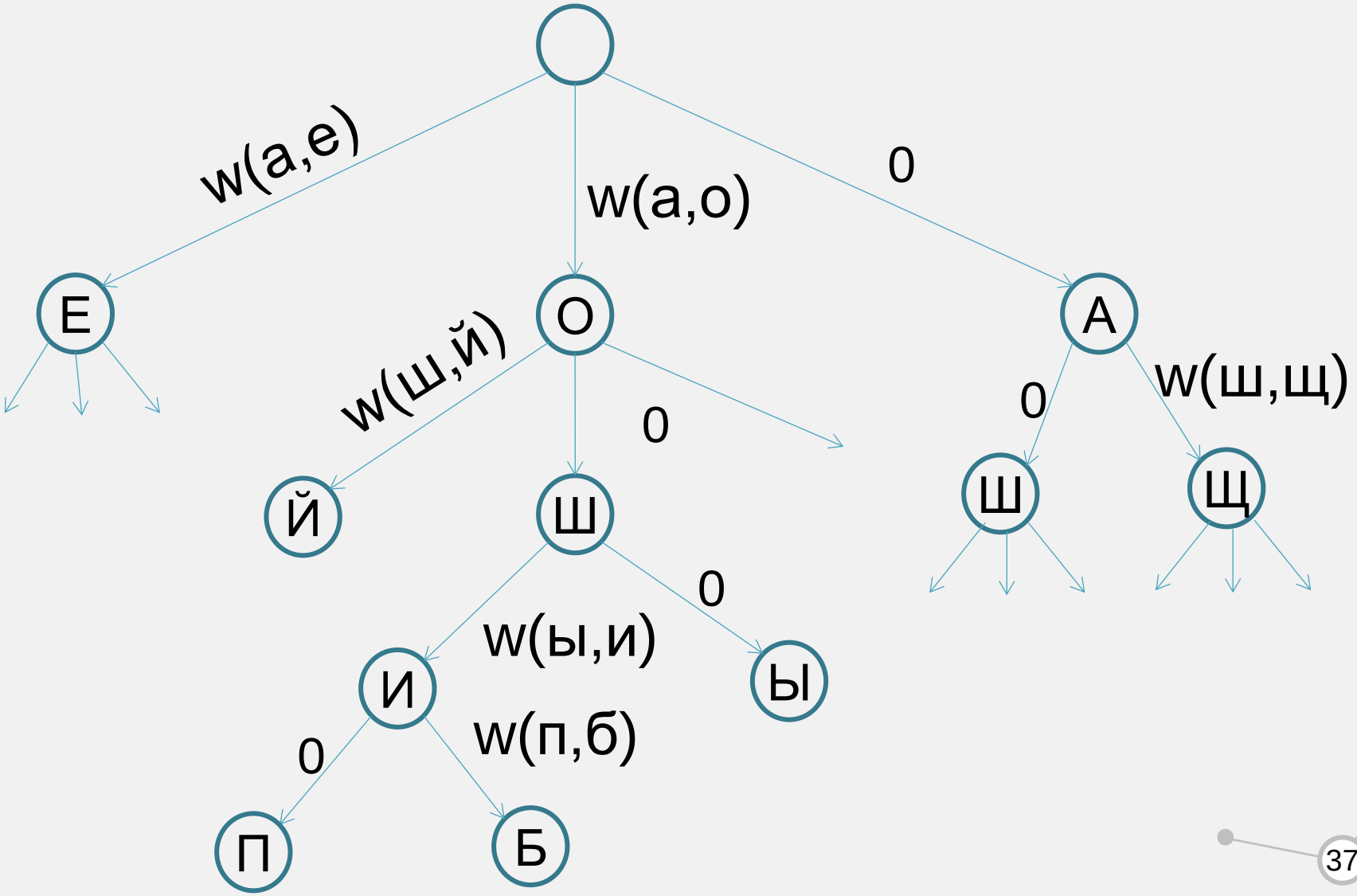
Sign(сонце) = снц

солнце ∉ Words(снц)

Генерация вариантов замены: нечеткий поиск в боре



Нечеткий поиск в Боре: АШЫПКА



Нечеткий поиск в Боре: вес кандидатов

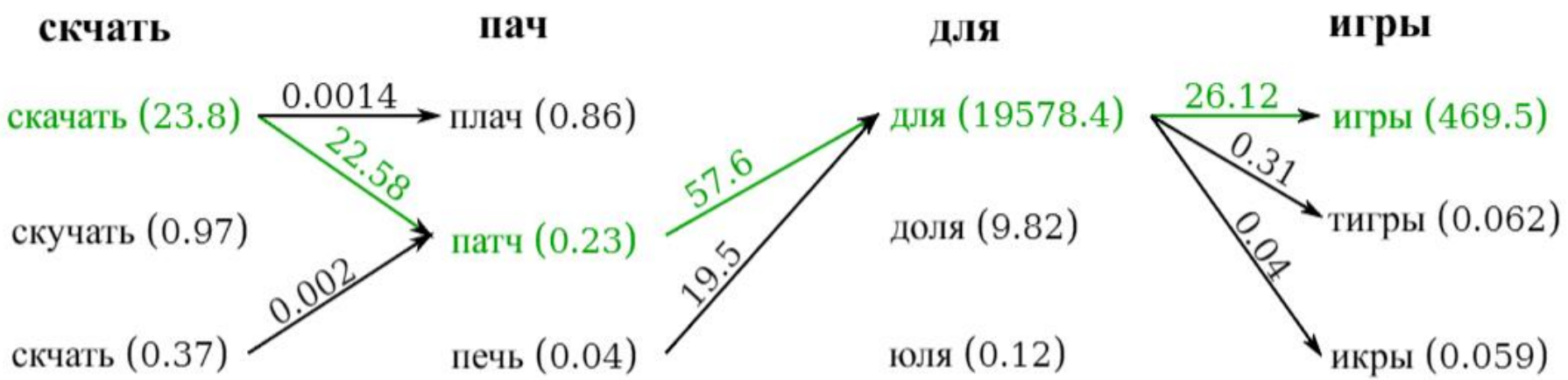
$$\alpha \log_2\big(Frequency(C)\big) + \log_2(P(W|C))$$

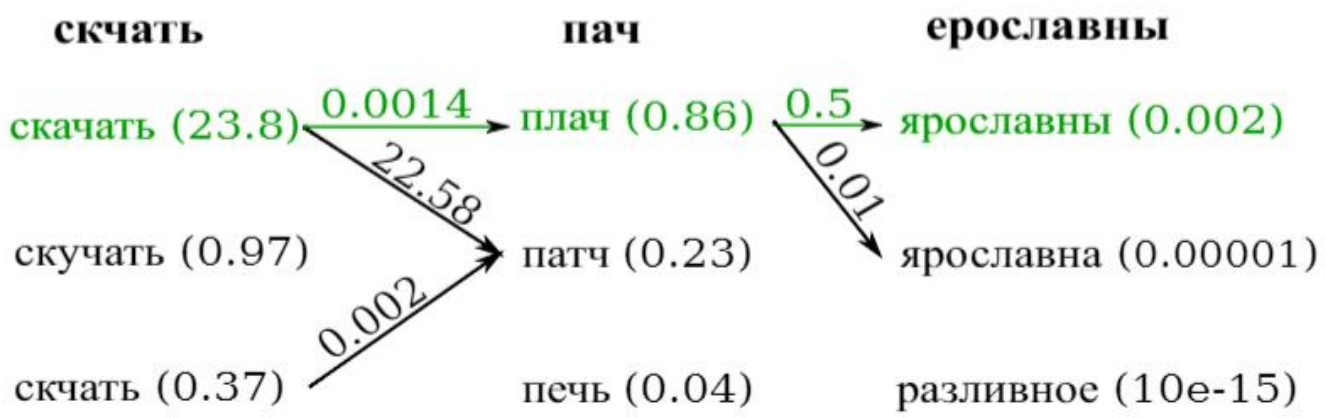
Вес вычисляется в процессе поиска:

- Частоту можно хранить в каждом узле
- Вероятность P(W|C) считается для текущего префикса
- Если вес становится очень маленьким, то глубже не ищем
- Всегда храним N лучших кандидатов

Как учесть вставоку и удление символа?

Граф слов


скчать
пач
для
игры
скачать (23.8)
0.0014
плач (0.86)
22.58
скучать (0.97)
патч (0.23)
0.002
скчать (0.37)
печь (0.04)
57.6
19.5
для (19578.4)
доля (9.82)
юля (0.12)
26.12
игры (469.5)
0.31
тигры (0.062)
0.04
икры (0.059)


скчать
пач
ерославны
скачать (23.8)
0.0014
плач (0.86)
0.5
ярославны (0.002)
22.58
0.01
скучать (0.97)
патч (0.23)
ярославна (0.00001)
0.002
скчать (0.37)
печь (0.04)
разливное (10e-15)

Архитектура опечаточника

Классификатор fix/none

Для каждого варианта исправления запроса должен решить возможно оно или нет:

- нечеткий п**ио**ск строки -> нечеткий п**ои**ск строки OK
- нечеткий п**ио**ск строки -> нечеткий п**и**ск строки NO

Факторы для классификации:

- Позапросные
  - длина (в символах, в словах)
  - частотность
  - языковой вес
- Про исправление
  - вес исправления
  - вероятность контекста
- Поведенческие
  - процент согласий / отказов от исправлений

Ранжирование

- Нужно выбрать из возможных исправлений лучшее
- Простой вариант:
  - Сортируем все по весу ошибки
- Сложный вариант:
  - Берем множество факторов
  - Строим классификатор “лучше / хуже”
  - Количество вариантов ограничено -> можем провести сравнение каждого с каждым
  - По сумме побед выбираем победителя

Итерации

Что делать, если несколько ошибок в запросе?

*методы государственно**е** по**д**е**рк**и лит**и**рат**к**р**а***

Исправляем несколькими итерациями:

1. *методы государственно**е** по**дерк**и лит**е**рат**у**р**ы***
2. *методы государственно**е** по**дд**ер**ж**ки литературы*
3. *методы государственно**й** поддержки литературы*

Классификатор auto/sugg

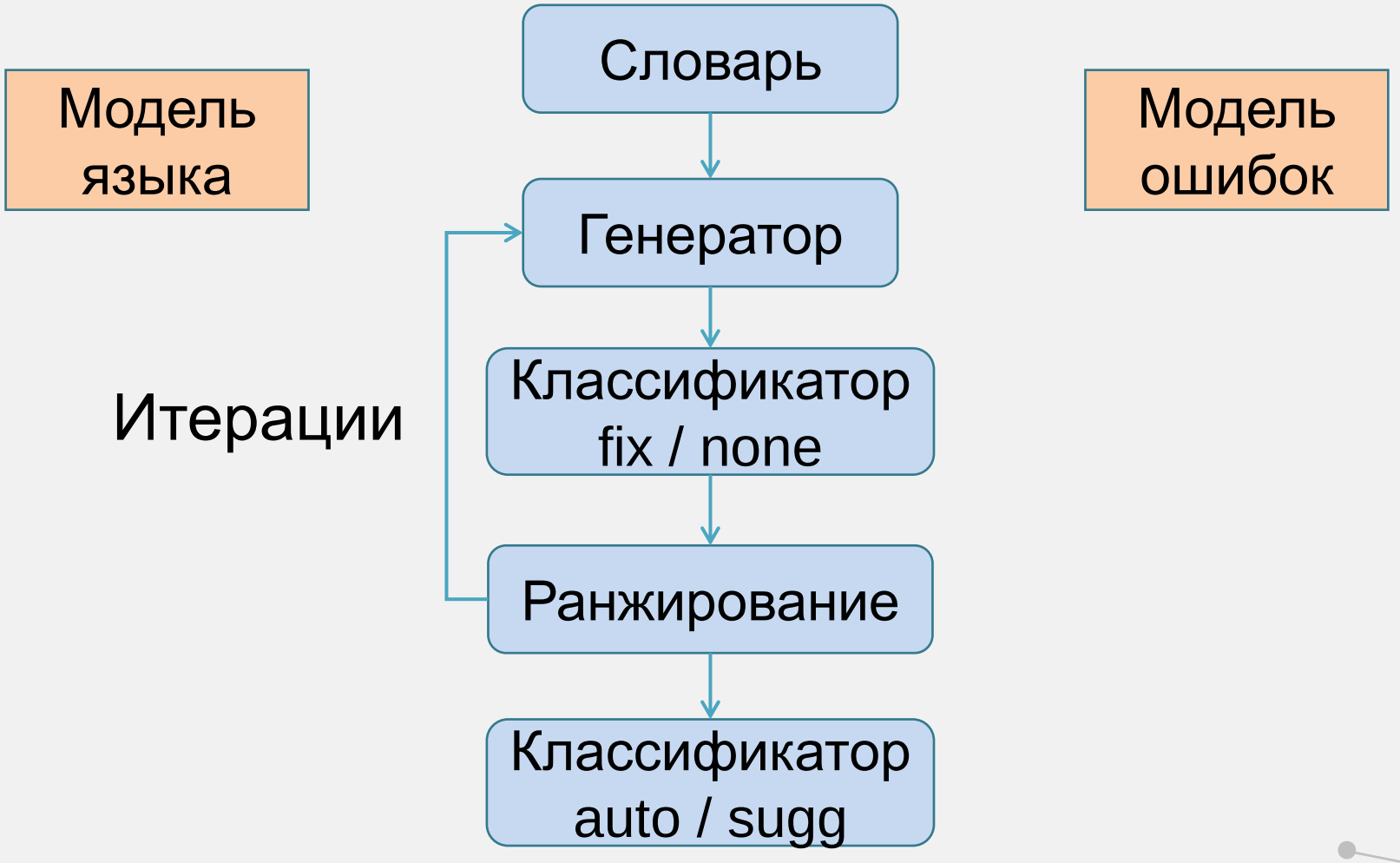
Простой классификатор:

- Используем уверенность классификатора fix / none
- Есть запросы, которые нужно исправлять, но нет уверенности во что исправлять:
  - поск - > {поиск, писк}
  - атташа - > {атташе, наташа}
- Вероятность исправления высокая, но не факт, что есть ошибка:
  - vnc -> мтс
  - пагода -> погода

Классификатор ML:

- Фичи из классификатора fix / none

Итерации


Модель языка
Словарь
Генератор
Классификатор fix / none
Ранжирование
Классификатор auto / sugg
Итерации
Модель ошибок

Разные типы исправлений

Если одновременно есть несколько типов исправлений?

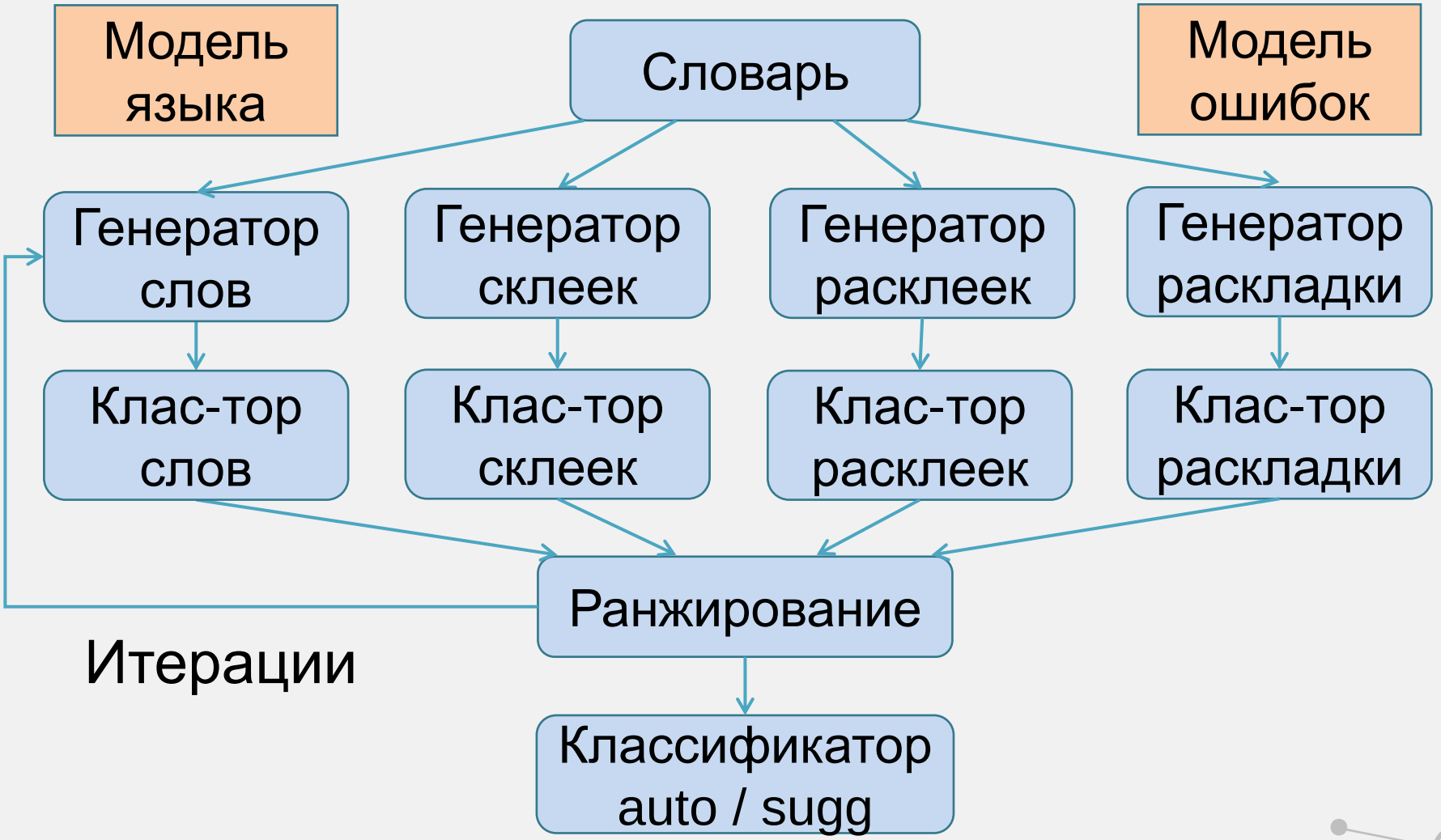
fyyfcbvbyjdbx -> анна семенович

Итерации:

1. fyyfcbvbyjdbx -> аннасименович (раскладка)
2. аннасименович -> анна сименович (склейка)
3. анна с**и**менович -> анна с**е**менович (слово)

Нужны разные генераторы и классификаторы

Разные типы исправлений



Оставшиеся проблемы

Названия доменов – необычные слова

- go.mal.кг -> go.mail.ru

Учет пунктуации

- e.chernov@corp.mail.ru
- S.M.A.R.T.

Учет регистра букв:

- ВкАНтакТе -> ВкОНтакТе

ТЕХНОСФЕРА

Отмечайтесь и оставляйте отзыв

Спасибо за внимание!

Евгений Чернов

e.chernov@corp.mail.ru

Практическое задание

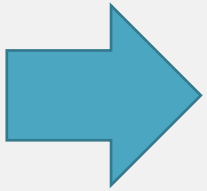
Простой спелчекер:

1. Строим словарь из текстов на lenta.ru https://cloud.mail.ru/public/857B/uheFJxvHF
2. Реализуем функцию подсчета расстояния Левенштейна
3. Делаем спелчекер с помощью поиска ближайших по Левенштейну слов
4. Делаем спелчекер путем генерации ближайших слов

Практическое задание

Простой спелчекер:

1. На вход подается набор слов
2. На выходе набор исправленных слов

путн		**путин**
оцинил		**оценил**
роботу		**работу**
новвых		**новых**
самалетав		**самолетов**
и		**и**
виртолтов		**вертолетов**
в		**в**
сирийи		**сирии**

Домашнее задание: Сделать спелчекер

Требования:

1. На вход принимать строки с запросами (из stdin)
2. Для каждого запроса выводить его правильную форму
3. Система должна содержать:
   - Модель языка
   - Модель ошибок
   - Генератор исправлений с помощью нечеткого поиска в боре
   - Классификатор
   - Итерации
   - Кроме словарных исправлений еще: split, join и layout

Домашнее задание: Сделать спелчекер

Исходные данные:

- Файл с запросами и исправлениями: https://cloud.mail.ru/public/E61x/ExC6EVSx6

Формат:

- запрос<TAB>исправление (если запрос с ошибкой)
- запрос (если запрос без ошибки)