

Лекция №13

Исправление опечаток в запросах

Евгений Чернов



Ошибки поисковых запросов



Запрос “одноклассники”:

- однокла~~с~~ники
- *jlyjrkfscybrb*
- одн~~а~~классники
- однокласс
- одн~~а~~кла~~с~~ники
- *jlyjrkfscybrb*
- однокл~~с~~ники
- од~~к~~ла~~с~~ники
- одноклассни
- одно~~к~~ассники

Типы ошибок: орфография



- Ошиблись в букве:
 - *вк~~а~~такте*
 - *ж~~ы~~вые обои*
 - *к~~о~~талог орифлейм*
- Пропустили букву:
 - *однокла~~с~~ники*
 - *самые качествен~~н~~ые свечи зажигания*
 - *ск~~ч~~ать игру*
- Лишняя буква
 - *тан~~ы~~цы айренби видео*
- Поменяли местами буквы
 - *скач~~та~~ь медиагет*
 - *купить ба~~р~~слет*

Типы ошибок: пробелы



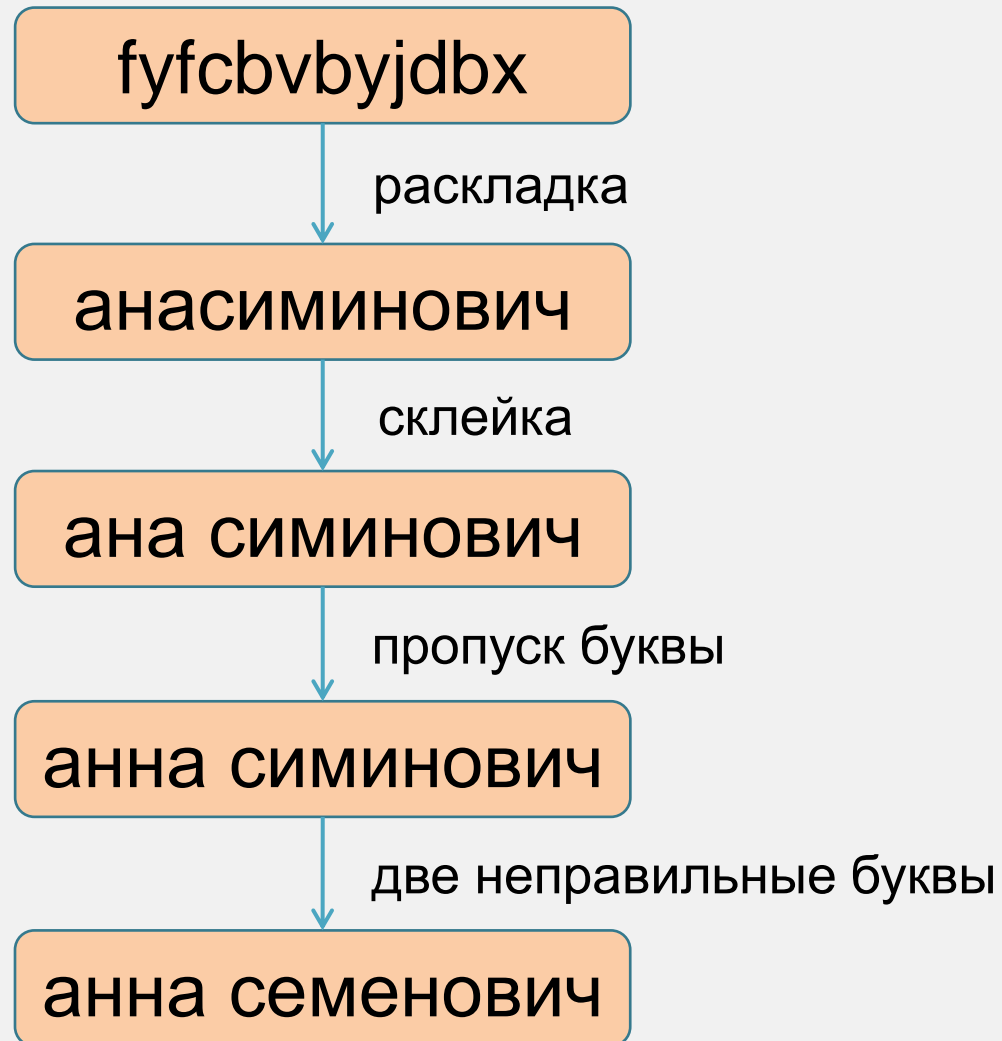
- Пропустили:
 - *голос 5 сезон **6**выпуск*
 - *замен**а**задней крестовины*
- Поставили лишний:
 - *что скачать что**о** **б**ы открывалась презентация*
 - *ма**к** **б**ук и вирусы*

Типы ошибок: раскладка / транслит



- Раскладка:
 - jlyjrkfssybrb (онокласники)
 - мл (vk)
- Транслитерация:
 - kupit televizor (купить телевизор)
 - мейл ру (mail ru)

Типы ошибок: смешанные



Типы ошибок: сложные случаи



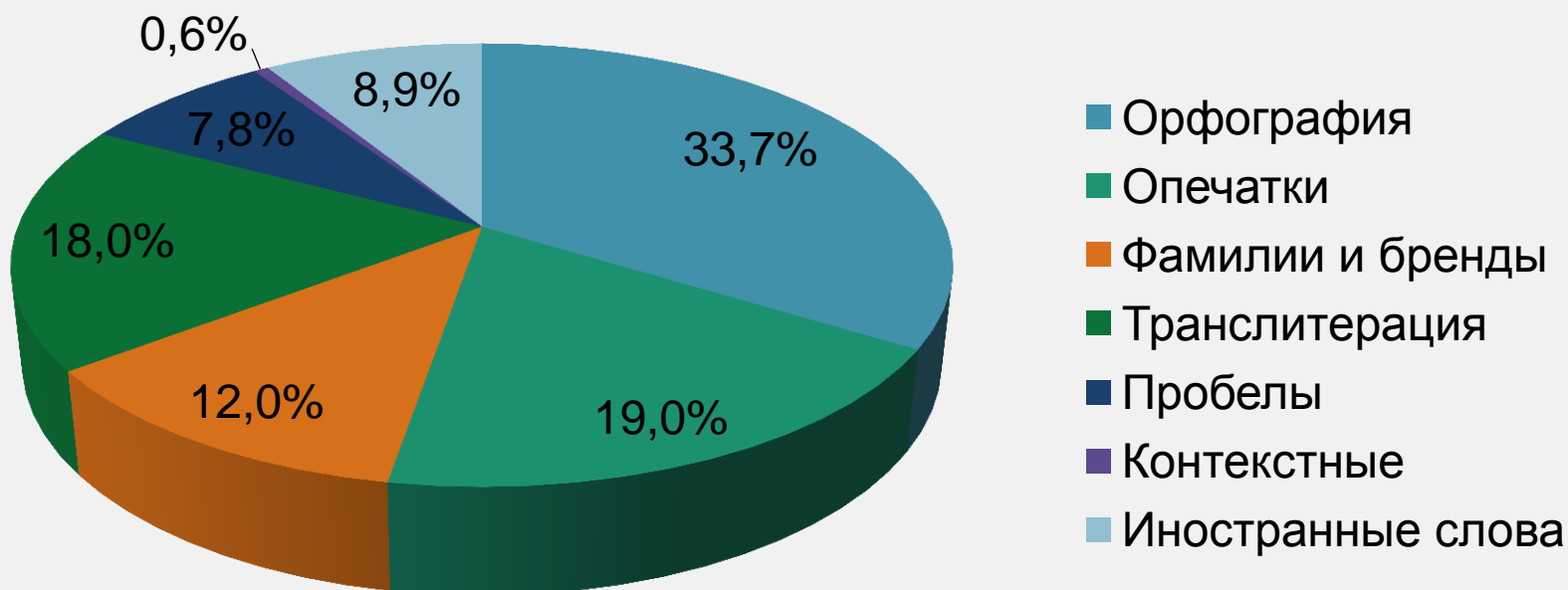
- Ошибка или нет?
 - пагода → погода (буддийское сооружение культового характера)
 - vnc → мтс (Virtual Network Computing)
- Ошибка или название домена:
 - rfcnh.kz → кастрюля (сайт в казахстане)
- Ошибка или модель изделия:
 - крюк для укладки труб RHSV (КРЫМ)
- Учет контекста:
 - клон (нет ошибки)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

Статистика ошибок



~ 11% запросов в потоке имеют ошибку

Типы ошибок



Типы исправлений



- Авто-исправление

путин

Интернет Картинки Видео Приложения Новости О

Исправлена опечатка: **птин**

- Подсказка

онлайн

Интернет Картинки Видео Приложения Новости

Вероятно, вы искали: **онлайн**

- Смещение

teri anry

Интернет Картинки Видео Приложения Новости О

Добавлены результаты по запросу: **teri anri**
Искать только: **teri anry**

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 поиск очепаток

очепаток
Пропустить все
Добавить в словарь

Простой печаточник



Активно используется в текстовых редакторах:

Это удивительное животное – сабака такса, описание которой известно всем. Такса — охотничья порода собак, которая отличается длинным туловищем и

Варианты из словаря (можно пополнять):

Это удивительное животное – сабака такса, описание которой известно всем. Такса — охотничья порода собак, которая отличается длинным туловищем и короткими лапами. Вообще-то, такса — это разновидность кроличья. Те же, в свою очередь,

собака
слабака
абака
кабака
сабана

Пропустить все
Добавить в словарь

Простой опечаточник: недостатки



Не учитывается контекст:

Это удивительное животное — сабака такса, описание которой известно всем. Такса — охотничья собака рода

Неполный словарь:

Простой опечаточник: недо

В слове есть ошибка, но оно есть в словаре:

Всемирный потоп — катастрофа огромного масштаба, легенды о которой широко

Простой печаточник: 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

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

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**я**ние \rightarrow расст**о**яние:
 - 2 операции замены букв \rightarrow 1 операция транспозиции

Учет неправильной раскладки:

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

Разный веса операций:

- удаление: 0,8
- вставка: 1,2

Разные веса для разных символов:

- $w(\text{п}, \text{р}) = 0,2$
- $w(\text{п}, \text{ъ}) = 0,9$

Расстояние Левенштейна: взвешенное



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

Результат: *На терра~~с~~е Агр~~е~~ппина Саввишна исподт~~и~~шка
потч~~и~~вала мол~~л~~юсками и в~~е~~негретом коллежского а~~с~~сессора.*

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

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

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

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

Расстояние Левенштейна: взвешенное



Диктант:

- сложно
- дорого
- малая полнота

Логи запросов:

- меньшая точность
- большая полнота

смор^теть без^платно фил^мы → смотреть бесплатно фильмы

Простой опечаточник: переход к вероятностям



Как выбрать из нескольких вариантов с одинаковым расстоянием Левенштейна:

расстояние 1 от “поток”:

{потом, потоп, порок, пяток, лоток, моток, роток}

Как учесть частотность запроса:

- африканский крон → африканский клон (1 лев.)
- африканский крон → африканский слон (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

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

Вычисление вероятности



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

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 \rightarrow 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: биграммная статистика



президент -> перзидеед

$\wedge_{\text{п}} - \wedge_{\text{п}} : 1$

$\text{п_} - \text{пе} : 1$

$\text{_р} - \text{ер} : 1$

$\text{ре} - \text{р_} : 1$

$\text{ез} - \text{_з} : 1$

$\text{зи} - \text{зи} : 1$

$\text{ид} - \text{ид} : 1$

$\text{де} - \text{де} : 1$

$\text{ен} - \text{ее} : 1$

$\text{нт} - \text{ед} : 1$

Модель языка (Language Model)



- Некоторая статистика языка
- В нашем случае построенная по запросам
- Язык меняется:
 - `ipad` -> `ipod`
 - последний из магикян -> последний из могикан
- Статистика собирается за определенный период
- Должна периодически обновляться

Модель языка (Language Model)

Как посчитать $P(query)$



1. 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: $query = w_1 w_2 w_3 \dots w_n$

- Униграммная модель (слова не зависят друг от друга):

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

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

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

3. Разбиение запроса на буквы/слоги

Поиск исправления



Какая осталась проблема?

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

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 **в** город → нижний новгород*

- Лишнее разбиение в перераскладке:

*у**б**;у**б**q → у**б** у**б**q → ни **и** 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 Н

ERMAN \rightarrow ORMON

ORMON \rightarrow 06505

06505 \rightarrow 06505

06505 \rightarrow 655

Результат: Н655

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**

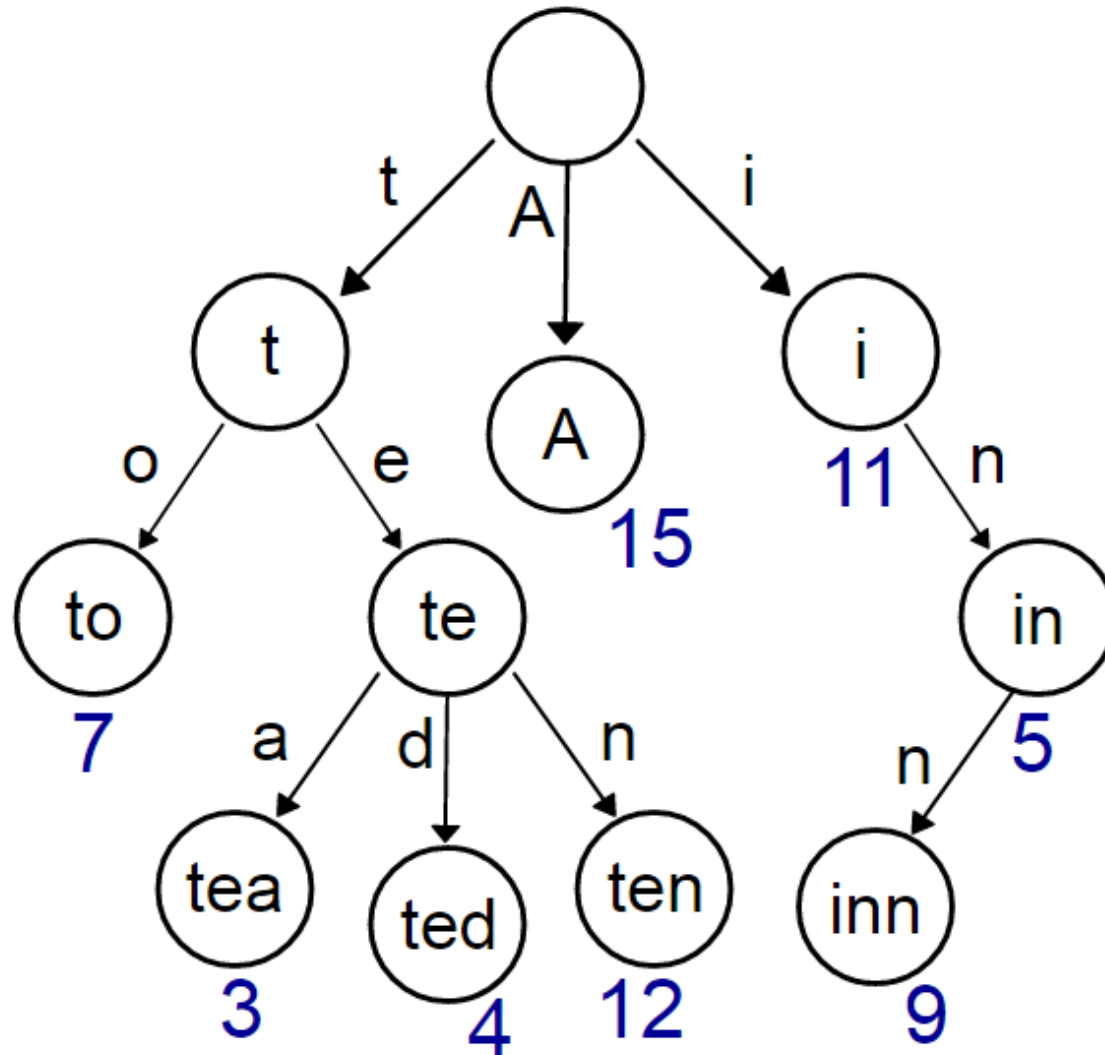
$Sign(режесер) = ршср$

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

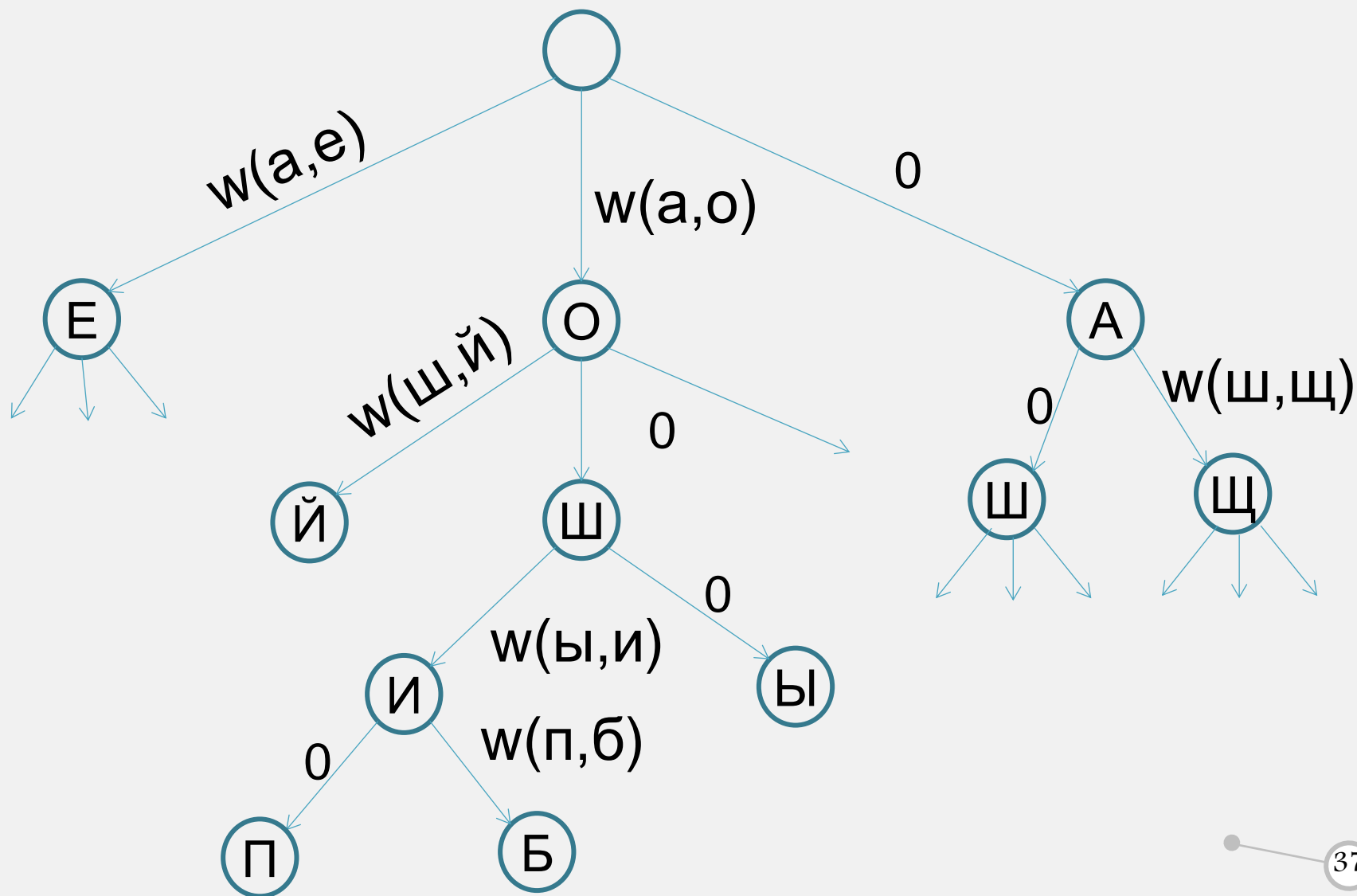
$Sign(солнце) = снц$

$солнце \notin Words(снц)$

Генерация вариантов замены: нечеткий поиск в боре



Нечеткий поиск в Боре: АШЫПКА



Нечеткий поиск в Боре: вес кандидатов



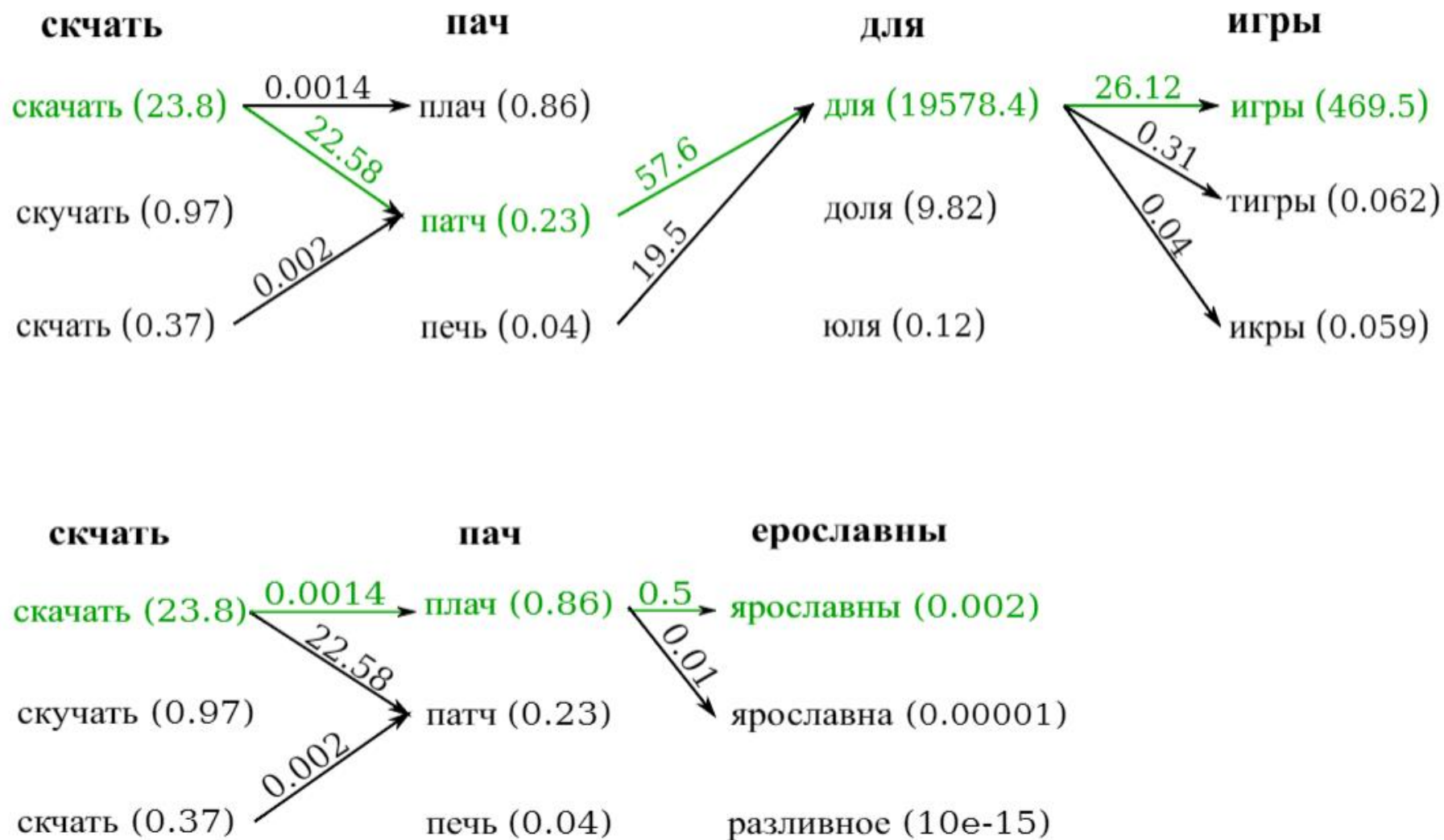
$$\alpha \log_2(\text{Frequency}(C)) + \log_2(P(W|C))$$

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 вставооку и удление символа?

Граф слов



Архитектура печаточника



Классификатор fix/none



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

- нечеткий п*и*иск строки → нечеткий п*о*иск строки OK
- нечеткий п*и*иск 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



- 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



Что делать, если несколько ошибок в запросе?

*методы государственное **е** под**е**рки лит**и**рат**к**ра*

Исправляем несколькими итерациями:

1. *методы государственное под**е**рки лит**е**рату**р**ы*
2. *методы государственное под**д**ерж**ж**ки литературы*
3. *методы государственной**й** поддержки литературы*



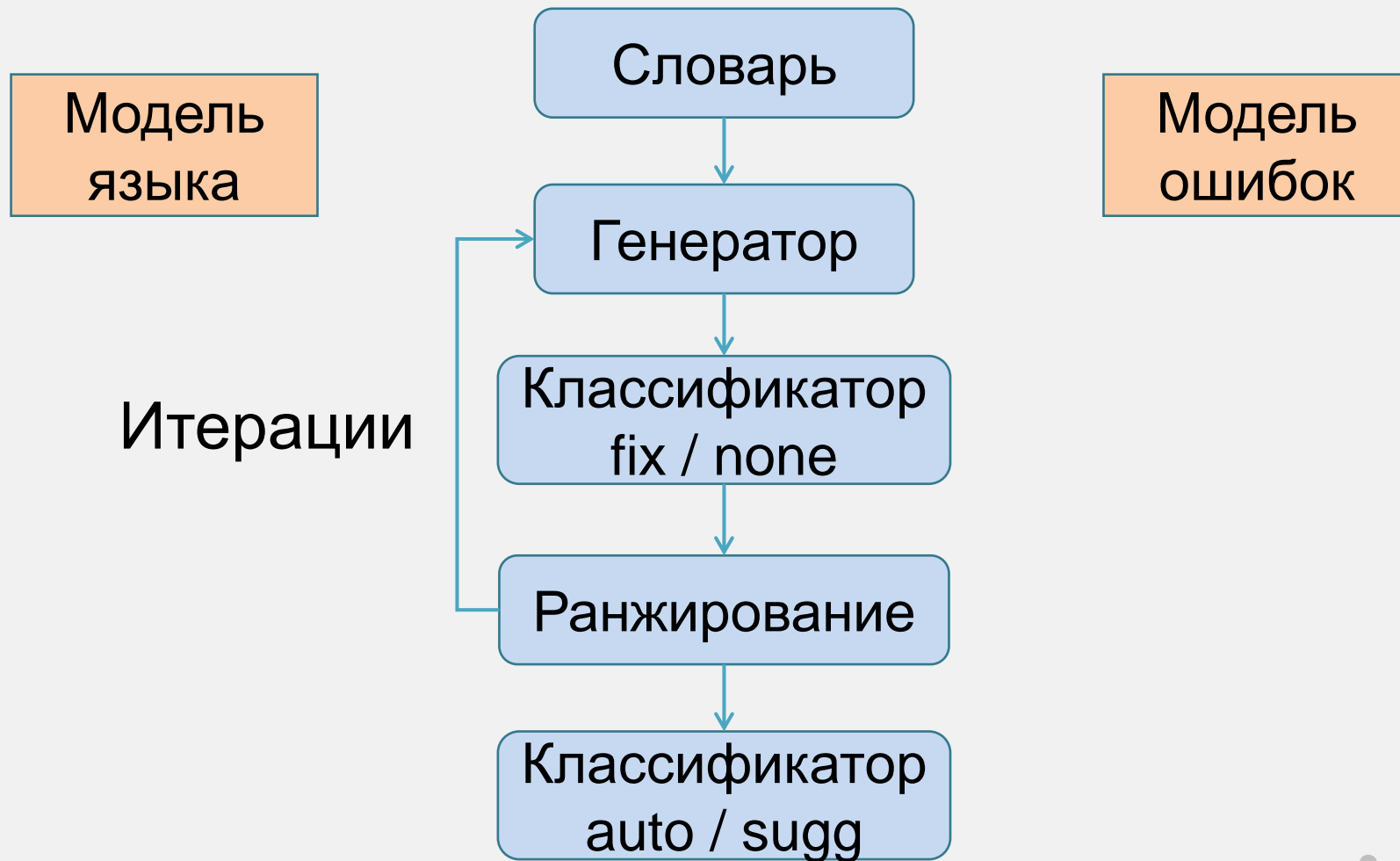
Простой классификатор:

- 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



Разные типы исправлений



Если одновременно есть несколько типов исправлений?

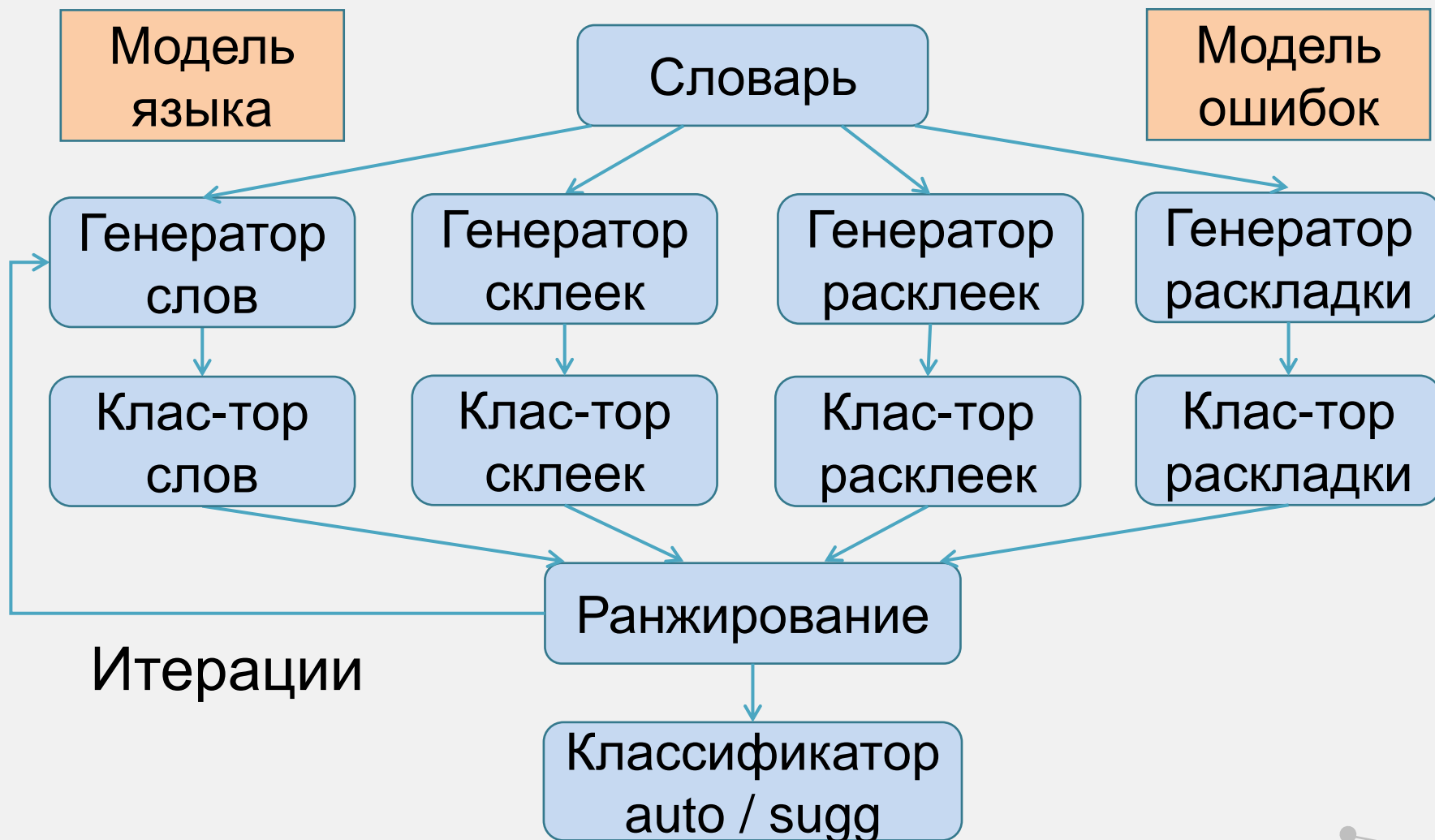
fyufcbvbyjdbx → анна семенович

Итерации:

1. *fyufcbvbyjdbx → аннасименович* (раскладка)
2. *аннасименович → анна сименович* (склейка)
3. *анна с**и**менович → анна се**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

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

оцнил

роботу

новбых

самалетав

и

виртолтов

в

сирийи



путин

оценил

работу

новых

самолетов

и

вертолетов

в

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