Лингвистика для математиков

Исправление опечаток 1

Удивительно, но начнём мы сегодня с задачи

В автоматической обработке естественного языка (например, при автоматической проверке орфографии) часто бывает нужно определить, **насколько различны два написанных слова**. Одна из количественных мер, используемых для этого, называется р**асстоянием Дамерау–Левенштейна** — в честь Владимира Левенштейна и Фредерика Дамерау. Левенштейн придумал способ измерения «расстояний» между словами, а Дамерау независимо от него выделил несколько классов, в которые попадает большинство опечаток.

1.	acre	car	2
2.	anteater	theatre	4
3.	banana	nanny	3
4.	cat	crate	2
5.	cocoon	cuckoo	3
6.	emporium	empower	4
7.	goer	ogre	2
8.	lyra	lay	2
9.	life	death	5
10.	point	sirloin	5
11.	stone	sonnet	3
12.	surge	ruse	3
13.	task	tusk	1
14.	peat	tape	4

15.	baba	arab
16.	contest	toner
17.	eel	lee
18.	martial	marital
19.	monarchy	democracy
20.	seatback	backseat
21.	warfare	farewell
22.	smoking	hospital
23.	ape	ea

Задание 1. Заполните пропуски.

Задание 2. Дайте определение расстоянию Дамерау–Левенштейна и предположите, какие классы опечаток выделил Дамерау.

Задание 3. Даны два слова с длинами m и n (m > n). Каково максимально возможное расстояние Дамерау–Левенштейна между этими словами? Минимально возможное? (Выразите ответы через m и n).

Изначальная идея

ALPHIBET ALPHABET	Words are same length and differ in only one position. They are the same word.
ALHPABET ALPHABET	Words are the same length and differ in two adjacent positions. If these are interchanged, the words match.
ALPHABET ALLPHABET ALPHABET ALPHABET	The entry word is one character longer. The first difference character is discarded. The characters following it are shifted left and the two words match.
ALPHABET ALPABET ALPABET	The entry word is one character shorter. The first difference character of the dictionary word is discarded and the characters following are shifted left. The two words now match.

Что такое spell checker

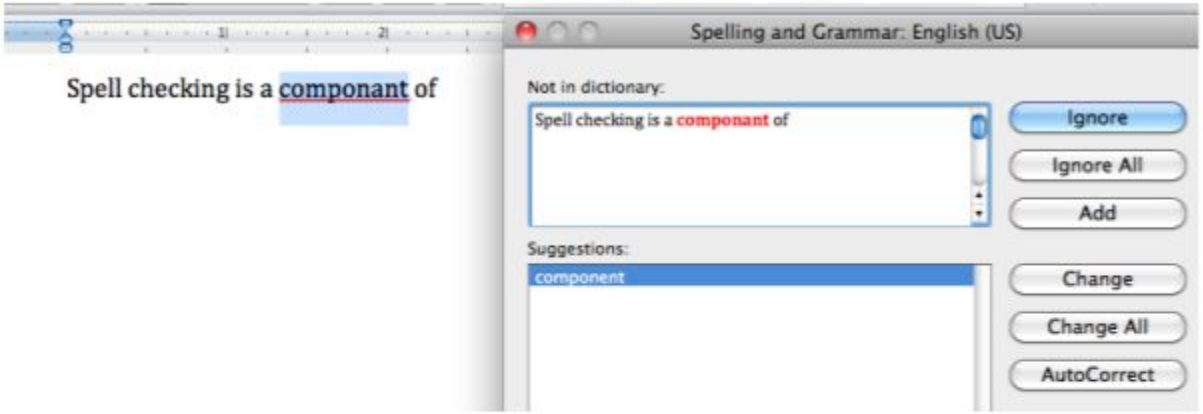
Софт/программа, которая проверяет текст на наличие опечаток

Задачи

- Поиск опечаток
- Исправление опечаток:
  - Автокоррекция (типа T9)
  - Предлагать исправление
  - Предлагать варианты исправлений

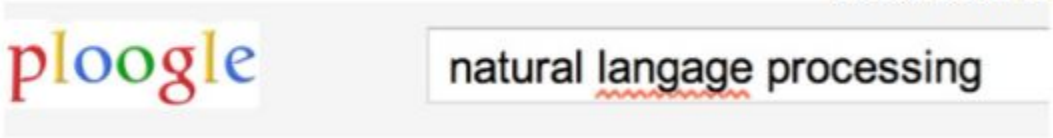
Применение

Word processing



Phones

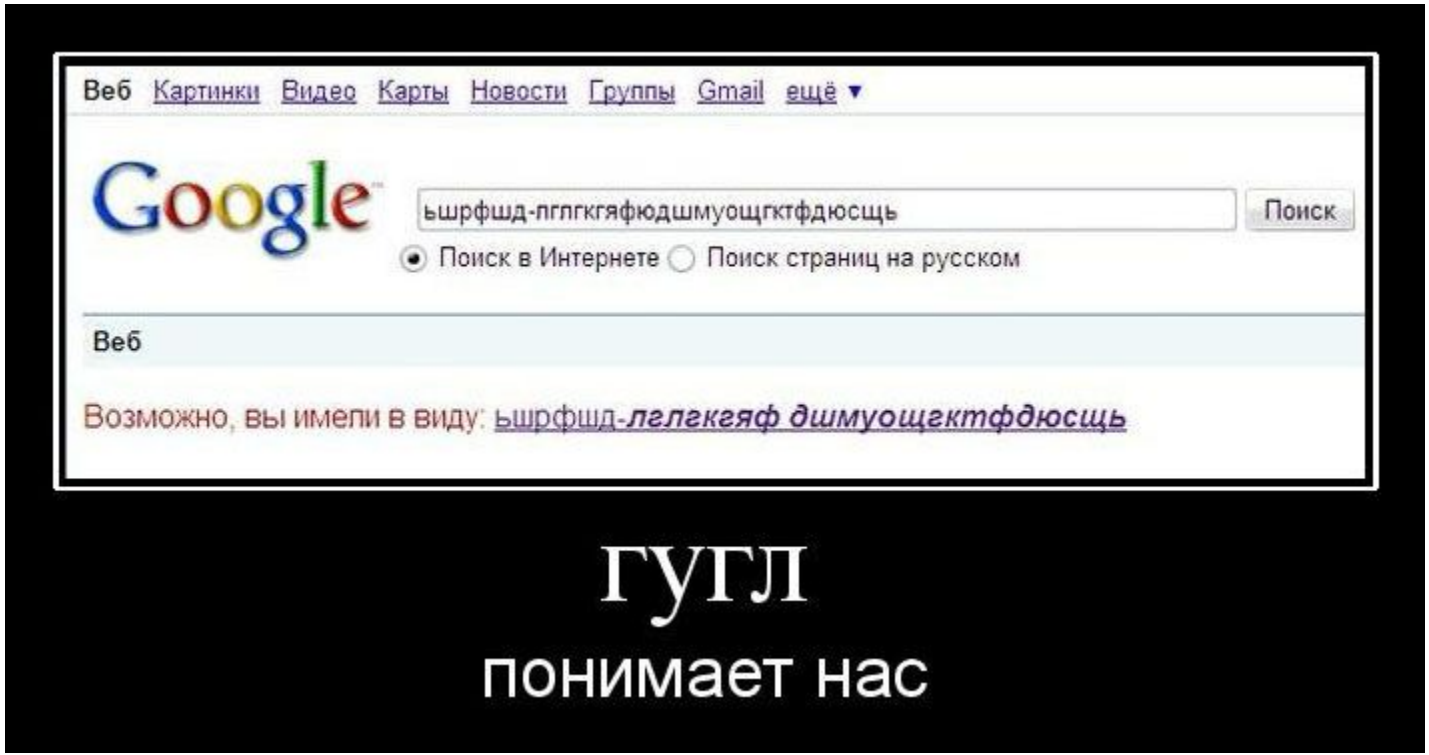
Web search



Showing results for natural ***language*** processing

Search instead for natural langage processing

Just a really old meme



Виды опечаток

- **Non-word errors**
- **Real world errors**
- Когнитивные ошибки
- Ошибки при записи речи “на слух”
- Ошибки при транслитерации
- Сокращения, слэнг
- Намеренные опечатки

Задание. 1913 год - не тот мир

ВОЙНА и МІРЪ.

ЧАСТЬ ПЕРВАЯ.

I.

— Eh bien, mon prince. Gênes et Lucques ne sont plus que des apanages, des поместья, de la famille Buonaparte. Non, je vous préviens que si vous ne me dites pas que nous avons la guerre, si vous vous permettez encore de pallier toutes les infamies, toutes les atrocités de cet Antichrist (ma parole, j'y crois) — je ne vous connais plus, vous n'êtes plus mon ami, vous n'êtes plus мой вѣрный рабъ, comme vous dites¹). Ну, здравствуйте, здравствуйте. Je vois que je vous fais peur²), садитесь и разсказывайте.

Такъ говорила въ іюлѣ 1805 года извѣстная Анна Павловна Шереръ, фрейлина и приближенная императрицы Маріи Ѳеодоровны, встрѣчая важнаго и чиновнаго князя Василія, перваго

Работа с real word опечатками

Для каждого слова w генерируем список кандидатов:

• Ищем кандидатов с похожим *произношением*

• Ищем кандидатов с похожим *написанием*

Выбираем лучшего кандидата. Как?

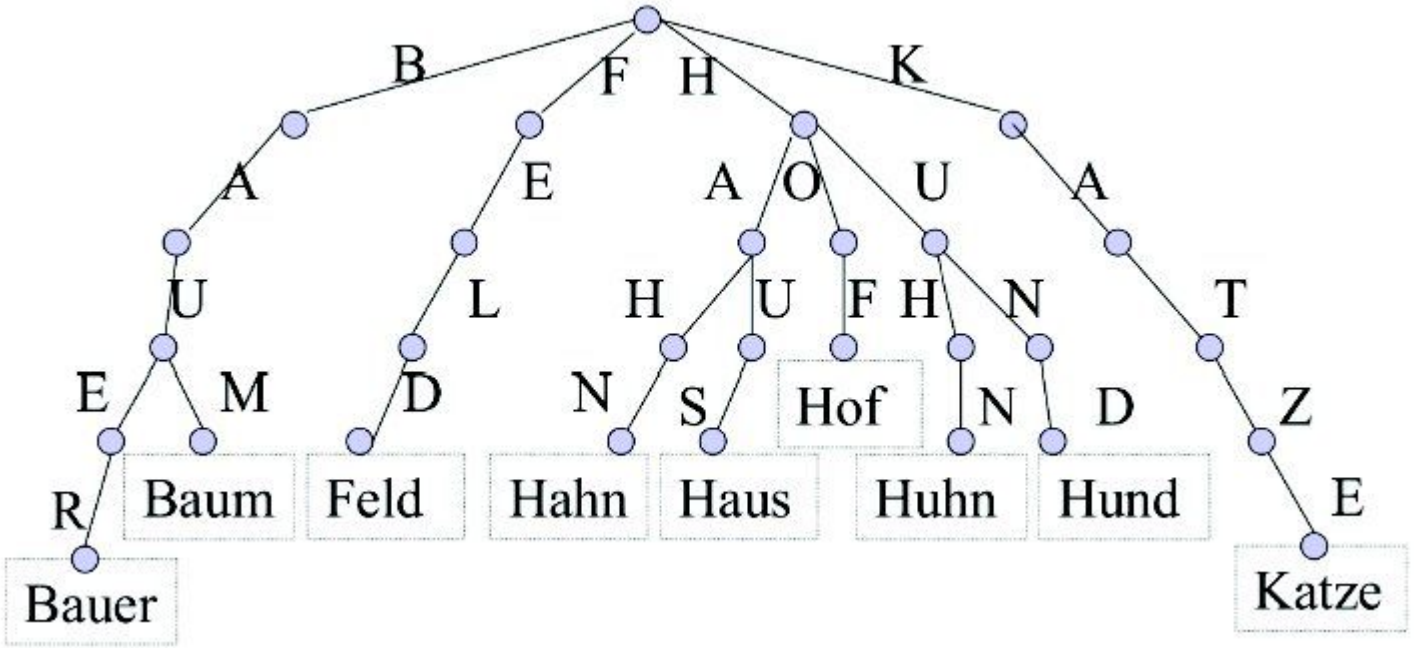
• алгоритм Noisy Channel

• классификатор

Работа с non-word опечатками

- Поиск non-word опечаток:
  - Составляем словарь.
  - Если слово не в словаре → это опечатка.
  - Чем больше словарь, тем лучше
- Исправление non-word опечаток:
  - Сгенерировать кандидатов реальных слов (из словаря) близких по буквам к опечатке.
  - Выбрать лучшего кандидата

Методом составления словаря



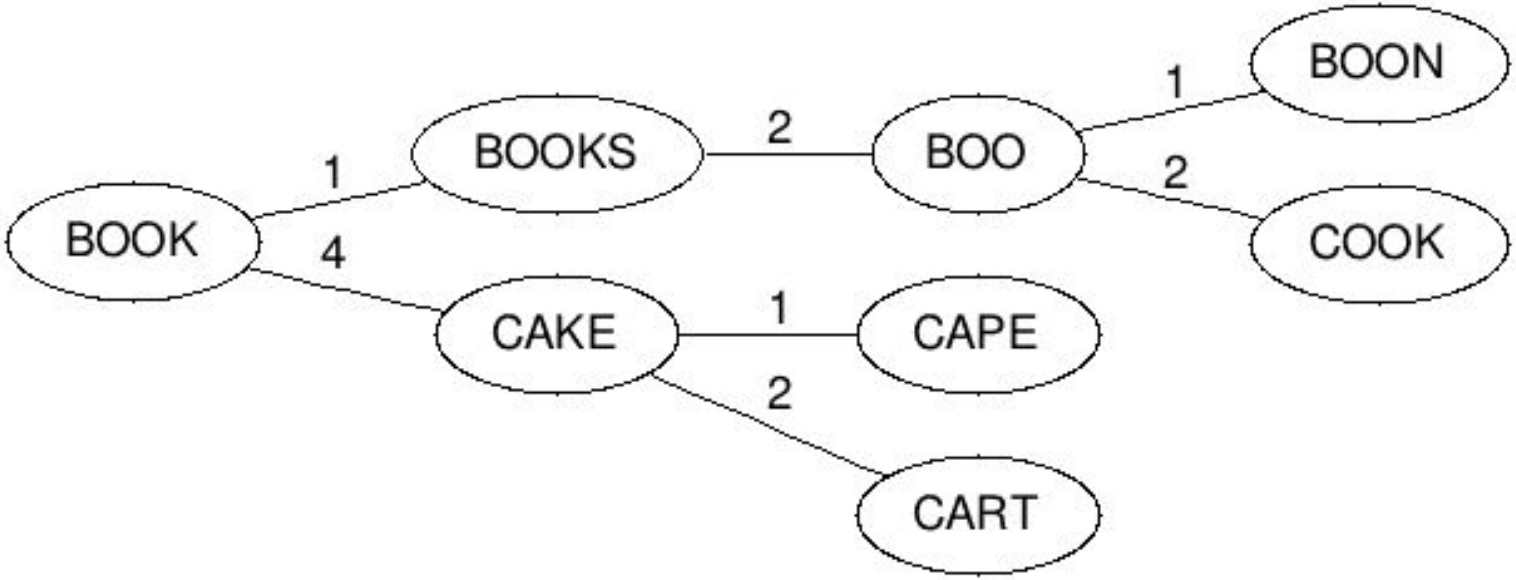
Работа с non-word опечатками

- Поиск non-word опечаток:
  - Составляем словарь.
  - Если слово не в словаре → это опечатка.
  - Чем больше словарь, тем лучше
- Исправление non-word опечаток:
  - Сгенерировать кандидатов реальных слов (из словаря) близких по буквам к опечатке.
  - Выбрать лучшего кандидата

Методом Bk-tress

- Преобразуем словарь в дерево
- Корень - случайное слово из словаря
- Слова из словаря связываются с корнем в дерево, на связях указано
- расстояние между словами по какой-нибудь метрике
- Approximate string matching
- Не надо сравнивать искомое слово с каждым словом в словаре → быстрее наивного метода

Bk-tress



Что такое “близкие слова”

- Можно искать близкие слова в словаре
- Для этого нужно задать функцию расстояния на множестве слов

Функции расстояния между строками

- Hamming расстояние = количество необходимых замен в строке

Арина Vs. Алина = 1
Karolin Vs. Kerstin = 3
01101 Vs. 00100 = 2

NB: Расстояния работают для любых строк/последовательностей, не только для слов

Функции расстояния между строками

- Hamming расстояние = количество необходимых замен в строке

Арина Vs. Алина = 1
Karolin Vs. Kerstin = 3

- расстояние Левенштейна и Дамерау–Левенштейна = количество необходимых замен, удалений вставок

kitten Vs. sitting = 3

kitten → sitten (substitution of "s" for "k")
sitten → sittin (substitution of "i" for "e")
sittin → sitting (insertion of "g" at the end)

Функции расстояния между строками

- Hamming расстояние = количество необходимых замен в строке

Арина Vs. Алина = 1
Karolin Vs. Kerstin = 3

- расстояние Левенштейна и Дамерау–Левенштейна = количество необходимых замен, удалений вставок

kitten Vs. sitting = 3

kitten → sitten (substitution of "s" for "k")
sitten → sittin (substitution of "i" for "e")
sittin → sitting (insertion of "g" at the end)

В Дамерау +транспозиция

лавка Vs. савок = 3
лавка → лавак
лавак → савак
савак → савок

Модель близости слов

Формальное определение:

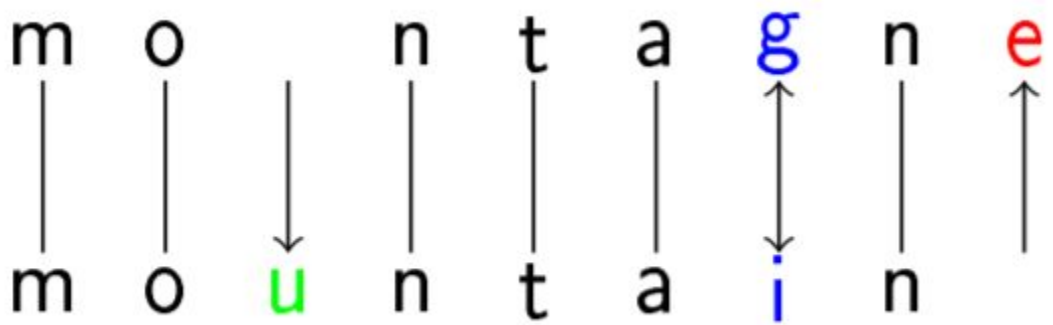
Расстояние Левенштейна p(u, v) между словами u и v -- минимальное число замен, вставок и удалений, необходимых, чтобы получить v из u.

[В.Левенштейн 1965]

Модель близости слов

d(montagne, mountain) = 3

Посчитали количество необходимых операций



Вычисление расстояния Левенштейна

Введём обозначения:

- w = w[0] ... w[n-1] -- слово, где |w| = n -- длина слова
- w[i] -- i-ый символ слова, где w[i,j] -- подслово с i-ой по j-ую позицию (не включая j)
- w[,j] -- префикс по j-ую позицию (не включая j)
- w[i,] -- суффикс с i-ой позиции (включая i)

Вычисление расстояния Левенштейна

Введём обозначения:

- w = w[0] ... w[n-1] -- слово, где |w| = n -- длина слова
- w[i] -- i-ый символ слова, где w[i,j] -- подслово с i-ой по j-ую позицию (не включая j)
- w[,j] -- префикс по j-ую позицию (не включая j)
- w[i,] -- суффикс с i-ой позиции (включая i)

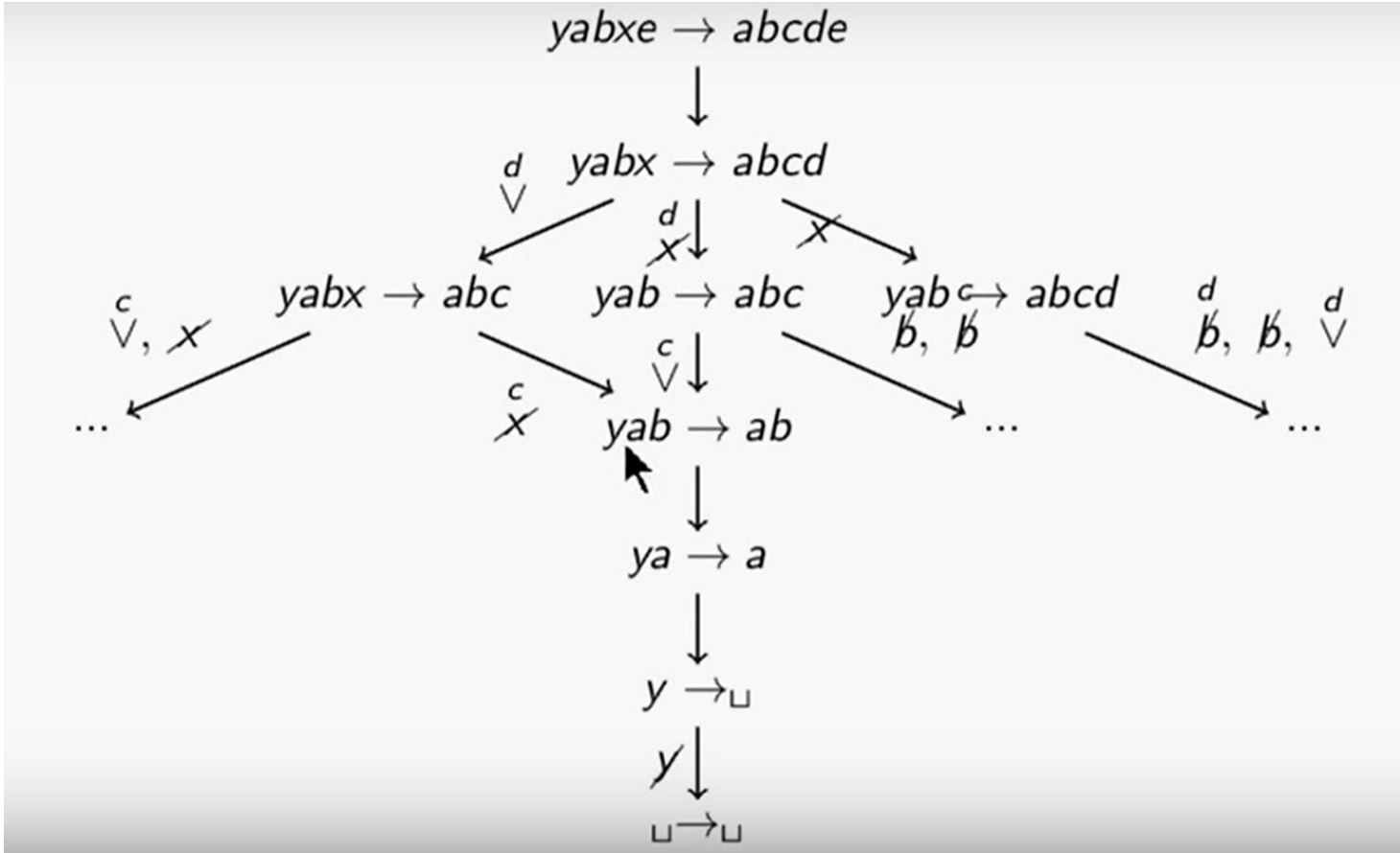
Идея:

Будем вычислять расстояние d[ij] = p(u[,i], v[,j]) рекурсивно через значения для меньших i, j. Если |u| = m, |v| = n, то ответом будет d[mn]. То есть последняя возможная операция.

Вычисление расстояние Левенштейна

Разделяй и властвуй

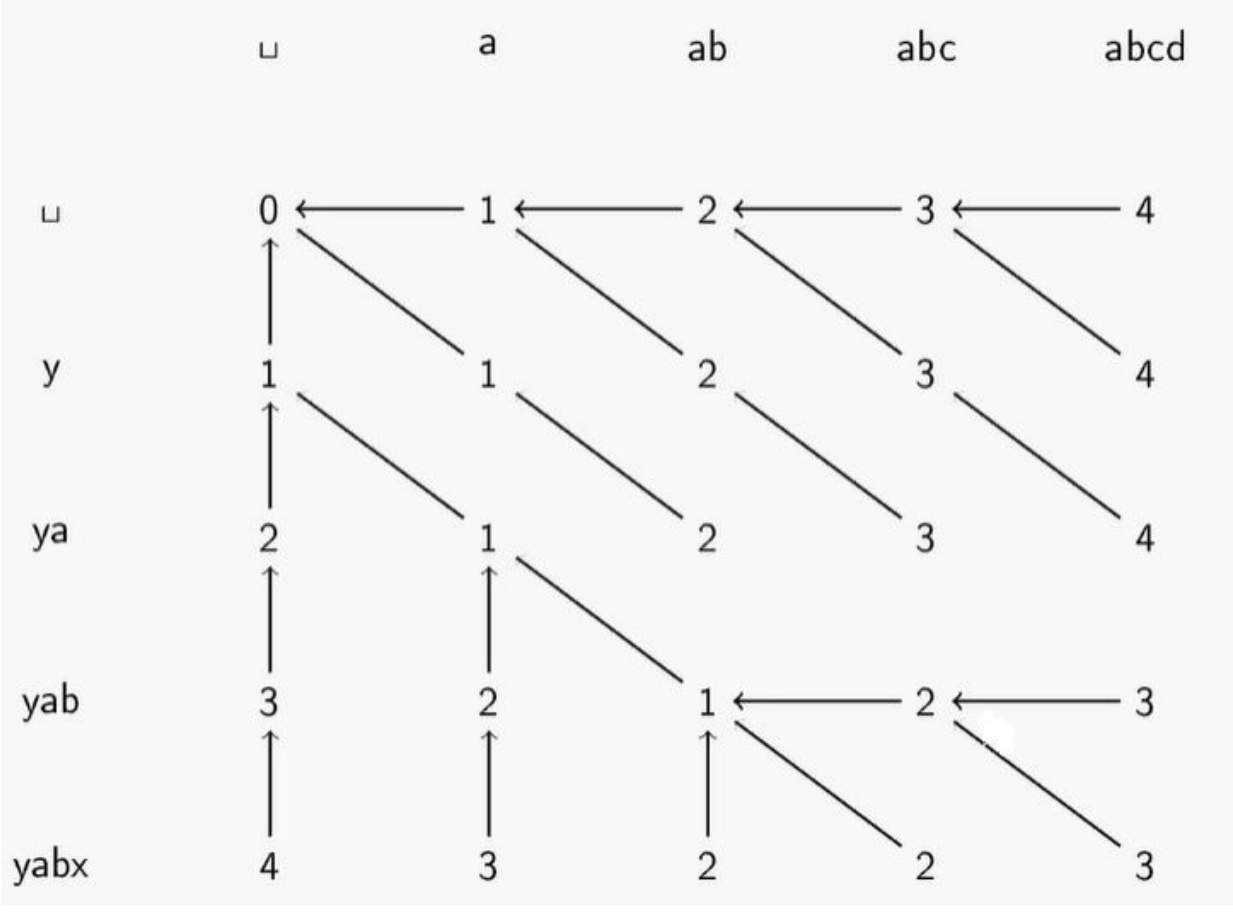
Какие есть подзадачи



Вычисление расстояние Левенштейна

То же самое в виде таблицы

yabxe → abcde



Формула расстояния Левенштейна

$$D(i,j) = \begin{cases} 0, & i = 0,\ j = 0 \\ i, & j = 0,\ i > 0 \\ j, & i = 0,\ j > 0 \\ \min\{ & \\ \quad D(i, j-1) + 1, & \\ \quad D(i-1, j) + 1, & j > 0,\ i > 0 \\ \quad D(i-1, j-1) + \mathrm{m}(S_1[i], S_2[j]) & \\ \} & \end{cases}$$

Вычисление расстояние Левенштейна

1) Посчитайте расстояние между **соль Vs. волос** с помощью таблицы
2) Какое расстояние будет между **kitten Vs. mitten** и между **kitten Vs. kiten**. Устно.
3) А если **seatback Vs. backseat**?
4) В чем проблема?

Оптимальное выравнивание

Это путь по таблице, который приводит к преобразованию одной строки в другую с минимальным значением расстояния Левенштейна, то есть это оптимальный порядок операций замена, удаление, добавление.

Его можно найти **не** для всех пар строк

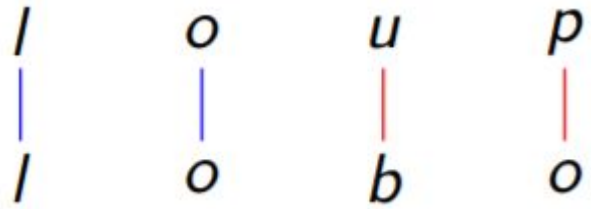
→ Взвешенное расстояние Левенштейна

Взвешенное расстояние Левенштейна

Какое расстояние между этими словами **d(loup, lobo)** из здравого смысла?

Взвешенное расстояние Левенштейна

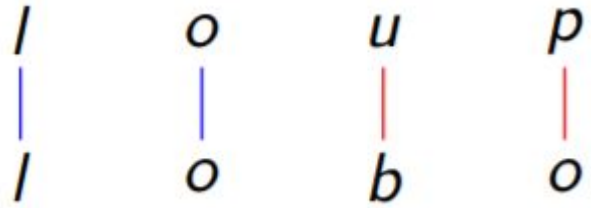
Какое расстояние между этими словами d(loup, lobo) из здравого смысла?



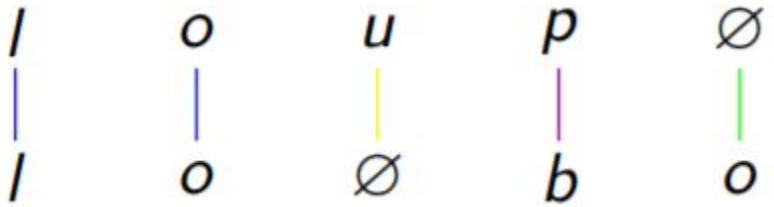
Теперь попробуйте посчитать расстояние Левенштейна

Взвешенное расстояние Левенштейна

Какое расстояние между этими словами d(loup, lobo) из здравого смысла?

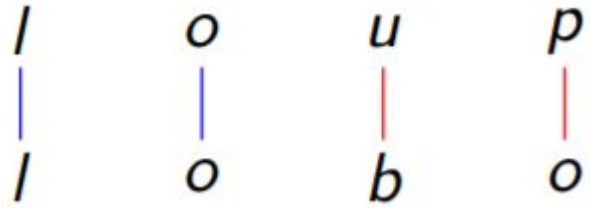


Теперь попробуйте посчитать расстояние Левенштейна

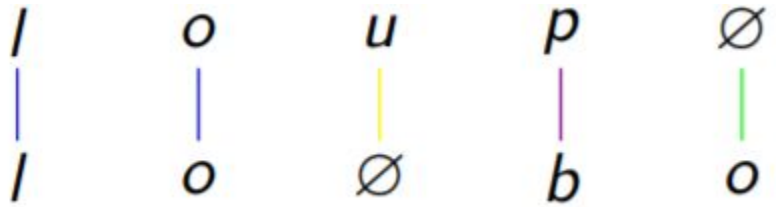


Взвешенное расстояние Левенштейна

Какое расстояние между этими словами d(loup, lobo) из здравого смысла?



Теперь попробуйте посчитать расстояние Левенштейна



Выход: присвоим нашим операциям какие-то “веса”

Модель близости слов

Еще раз как же выглядит модель по поиску ошибок и их исправлению этой моделью?

- Наивный подход: пройти по словарю и посчитать расстояние до каждого слова, выбрать ближайшее
- Но, так нельзя: словари большие (агглютинативные языки -- сотни тысяч слов), а расстояние считается медленно

Модель близости слов

Еще раз как же выглядит модель по поиску ошибок и их исправлению этой моделью?

- Наивный подход: пройти по словарю и посчитать расстояние до каждого слова, выбрать ближайшее
- Но, так нельзя: словари большие (агглютинативные языки -- сотни тысяч слов), а расстояние считается медленно
- Менее наивный подход: породить все слова, расстояние до которых меньше порога, найти их в словаре

Взвешенное расстояние Левенштейна

Как же определять веса? Можем условно считать, что вес - это вероятность опечатки на какой-то комбинации букв

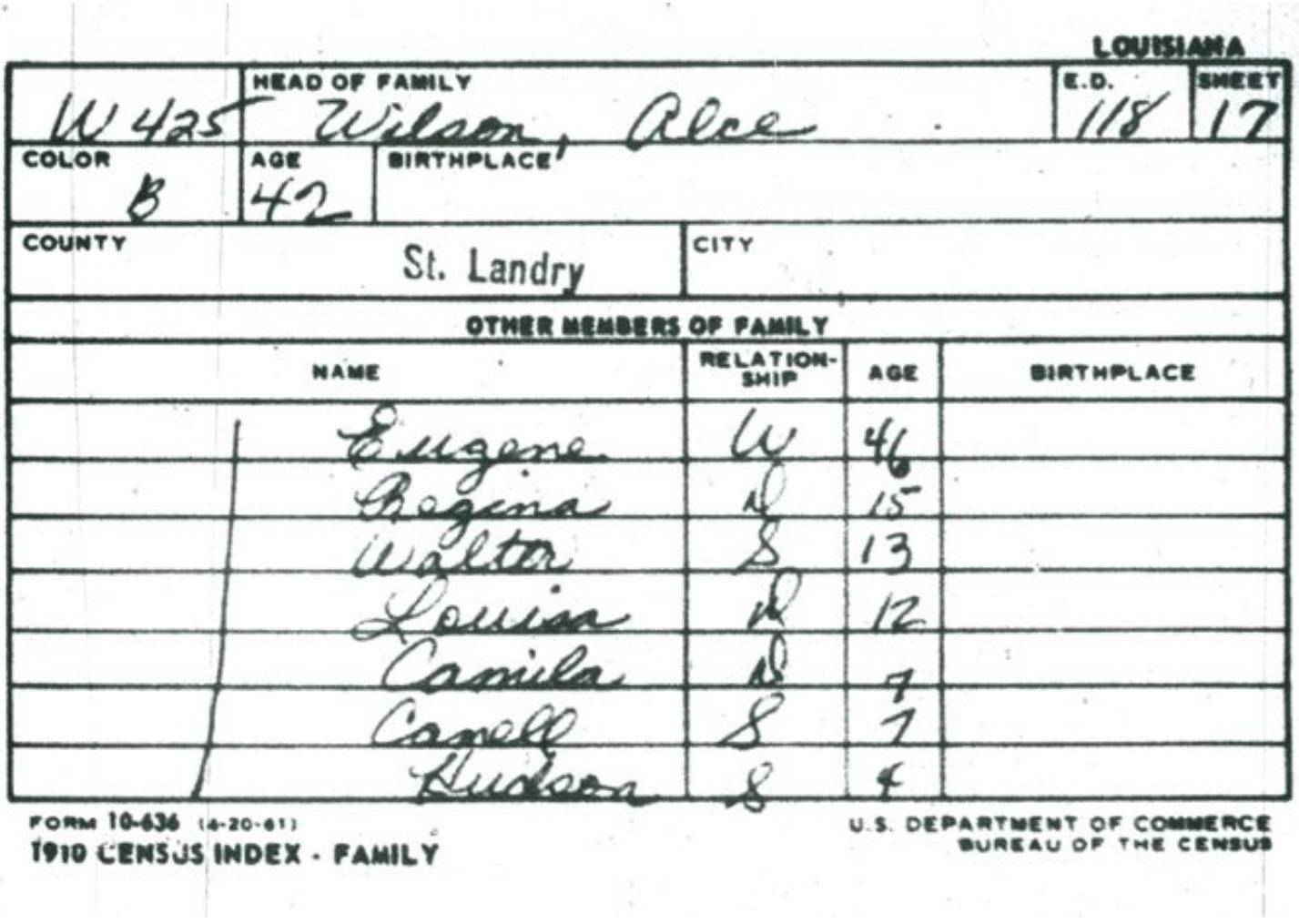
Как вычислить эту вероятность?

- корпус опечаток
- эвристики:
  - расположение символов на клавиатуре
  - фонетическая близость
  - графическая близость

Кстати, про фонетическую близость

- обычно в алгоритмах с метриками расстояния кандидатами в итоге считаются слова, которые отличаются от исходного на 1-2 буквы
- Больше отличающихся букв: *riiiiigtht –> right, donut –> doughnut, ave–> avenue*
- Тогда нужно использовать другие метрики расстояния, ориентированные на фонетическую близость и аббревиатуры/сокращения
- Первым алгоритмом работающим с фонетической схожестью был Soundex

Soundex



LOUISIANA

	HEAD OF FAMILY	E.D.	SHEET
W 425	Wilson, Alce	118	17

COLOR	AGE	BIRTHPLACE
B	42	

COUNTY	CITY
St. Landry	

OTHER MEMBERS OF FAMILY

NAME	RELATION-SHIP	AGE	BIRTHPLACE
Eugene	W	46	
Regina	d	15	
Walter	S	13	
Louisa	d	12	
Camila	d	7	
Carell	S	7	
Hudson	S	4	

FORM 10-636 (4-20-61)
1910 CENSUS INDEX - FAMILY

U.S. DEPARTMENT OF COMMERCE
BUREAU OF THE CENSUS

Дан список фамилий и соответствующих им кодов Soundex в перепутанном порядке. Некоторые символы пропущены:

Allaway*, *Anderson*, *Ashcombe*, *Buckingham*, *Chapman*, *Colquhoun*, *Evans*, *Fairwright*, *Kingscott*, *Lewis*, *Littlejohns*, *Stanmore*, *Stubbs*, *Tocher*, *Tonks*, *Whytehead

S312, T␣6␣, ␣5␣3, C42␣, T520, L␣42, A536, C155, ␣623, S356, ␣252, ␣152, ␣330, A251, A400, L2␣0

Soundex

Задание 1. Опишите пошагово, как генерируется код Soundex.

Задание 2. Установите соответствия между фамилиями и кодами Soundex и вставьте пропущенные символы.

Задание 3. Постройте коды Soundex для следующих фамилий:

Ferguson, *Fitzgerald*, *Hamnett*, *Keefe*, *Maxwell*, *Razey*, *Shaw*, *Upfield*.

Всем спасибо

Литература

1. А. Пиперски (2015). Математические модели в лингвистике. Курс в МГУ
2. Задача про Soundex. А.Пиперски (https://elementy.ru/problems/1608/Soundex)
3. А. Бердичевский. Задача про расстояние Дамерау-Левенштейна (https://elementy.ru/problems/1068/Rasstoyanie_DamerauLevenshteyna)