

# Занятие 3

## Предобработка. Часть II

Катя Герасименко  
Летняя олимпиадная школа МФТИ  
01.08.2018

# Регистр

Зависит от задачи.

Чаще всего – приводить все к нижнему.

Но, например, для именованных сущностей информацию о регистре надо обязательно сохранить.

# Спеллингчек

Степень необходимости зависит от задачи.

В чат-боте и поиске лучше иметь (и вообще возможность нечеткого поиска – fuzzy search) (но в вашем проекте не нужно прикручивать спеллингчек)

В основе хорошего спеллингчека:

- модель для подбора кандидатов (расстояние Дамерау-Левенштейна, например)
- языковая модель для выбора среди кандидатов

# Расстояние Левенштейна

**Расстояние Левенштейна** (одна из метрик редакционного расстояния (edit distance)) – минимальное количество операций **вставки**, **удаления** и **замены**, требующееся для того чтобы превратить одну строку в другую.

М	М	М	Р	І	М	Р	Р
С	О	Н	Н		Е	С	Т
С	О	Н	Е	Н	Е	А	Д

# Расстояние Левенштейна

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

		А	Р	Е	С	Т	А	Н	Т
	0	1	2	3	4	5	6	7	8
Д	1	1	2	3	4	5	6	7	8
А	2	1	2	3	4	5	5	6	7
Г	3	2	2	3	4	5	6	6	7
Е	4	3	3	2	3	4	5	6	7
С	5	4	4	3	2	3	4	5	6
Т	6	5	5	4	3	2	3	4	5
А	7	6	6	5	4	3	2	3	4
Н	8	7	7	6	5	4	3	2	3

# Расстояние Левенштейна

- Можно давать больший вес замене
- Можно учитывать еще транспозицию (80% всех опечаток) – расстояние Дамерау-Левенштейна

# Лемматизация

Русский язык – язык с богатой морфологией.

*(с) каждая лекция о предобработке*

- Лемматизация для английского – обязательно
- Лемматизация для русского – а must (если у вас не очень большой корпус)

Связанная с этим проблема – снятие омонимии  
*(стали, сорока)*

# Какие инструменты есть

## py morphology2:

- словарь + правила
- не смотрит на контекст, у слова одинаковые разборы и их порядок тоже одинаковый
- «скор» разбора считается по парадигме, не по корпусу
- много разных функций и хорошо выстроенная архитектура
- быстрый



# Какие инструменты есть

**mystem:**

- префиксные деревья (*trie*) + статистика
- учитывает контекст при снятии омонимии
- есть обертка для питона, но не всегда удобно пользоваться + дольше по сравнению с запуском из командной строки
- mystem в целом чудовищно долго работает на Windows

# Стоп-слова

Служебные и околослужебные слова не несут семантики и часто только создают шум.

Решение: сделать список таких слов, выкидывать их на этапе предобработки

Список может быть разным в зависимости от задачи