



ИСПОЛЬЗОВАНИЕ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ ИСПРАВЛЕНИЯ ОШИБОК

Ксения Варегина
tg @Varvar_Ks

Какие ошибки будем исправлять?

How Do You Spell PARAGRAM?



Common Misspellings

- paragrakm, parageam, paragfram, paragdram, parabgram, paragramsm, pa4ragram, p-aragram, paraqgram, Paracram, parqgram, parzgram, paeagram, laragram, pararam, pareagram, parfagram, paragramm, paraganm, paragramk, paragrram, patagram, pparagram, paragrsm.

Везде—Errare humanum est

Ошибки в словах [[]]:

- Пропуск буквы (кросовки → кроссовки)
- Вставка буквы (фломастекр → фломастер)
- Замена буквы (эксперемент → эксперимент)
- Перестановка букв (пространтсво → пространство)

Склейка и разрезание:

- Пропуск пробела (купитьдиван → купить диван)
- Вставка пробела (пол года → полгода)

Раскладка клавиатуры (rfr cltkfnn cfqn → как сделать сайт)

Транслитерация (май нейм из саша → my name is sasha)



Подробнее о статистике в поисковых запросах [здесь](#)

Про способы исправления слов ...



Можете
ознакомиться:

- [Здесь](#)
- [Спел-чекер Питера Норвига](#)
- [Здесь](#)
- [Здесь](#)
- [Здесь](#)

19.12.2022

МОДЕЛИ ОШИБОК

Стоит хорошенько проработать все виды ошибок, чтобы получить модели ошибок. Этот шаг мы пропускаем сейчас. И переходим к...

19.12.2022

МОДЕЛЬ ЯЗЫКА

на основе N-грамм

Вспомним, о чем идет речь

N-грамма — последовательность из n элементов. Например, последовательность звуков, слогов, слов или букв.

This is Big Data AI Book

Uni-Gram	This	Is	Big	Data	AI	Book
Bi-Gram	This is	Is Big	Big Data	Data AI	AI Book	
Tri-Gram	This is Big	Is Big Data	Big Data AI	Data AI Book		

Как работает N-граммная модель [\[2\]](#)

По тексту (который используется для обучения) проходим окном размером в N слов ([в Joom остановились на триграммах](#)).

Подсчитываем количество раз которые встретилось каждое сочетание (n-грамма).

Традиционная модель языка на основе n-грамм выглядит так. Для фразы $w_1 w_2 \dots w_k$ её вероятность вычисляется по формуле

$$P(w_1 w_2 \dots w_k) = P(w_1) \cdot P(w_2 | w_1) \cdot P(w_3 | w_1 w_2) P(w_k | w_1 w_2 w_{k-1}),$$

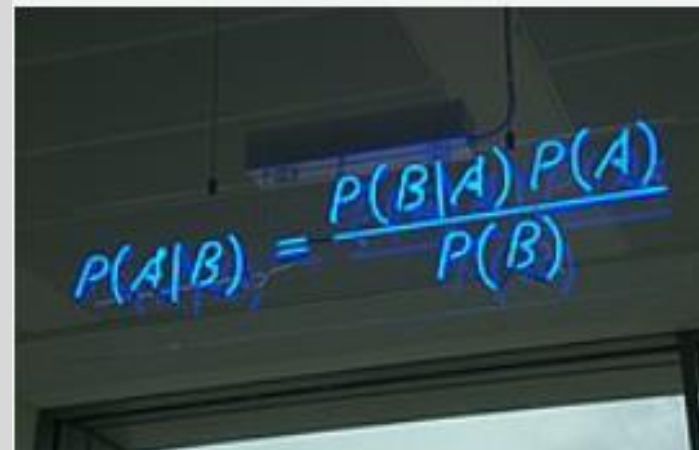
где $P(w_1)$ — непосредственно частота слова, а $P(w_3 | w_1 w_2)$ — вероятность слова w_3 при условии, что перед ним идут $w_1 w_2$ — не что иное, как отношение частоты триграммы $w_1 w_2 w_3$ к частоте биграммы $w_1 w_2$. (Заметьте, что эта формула — просто результат многократного применения формулы Байеса.)

Формула Байеса

Если коротко:

*По **формуле Байеса** можно более точно пересчитать вероятность, беря в расчет как ранее известную информацию, так и данные новых наблюдений. Формула Байеса позволяет «переставить причину и следствие»: по известному факту события вычислить вероятность того, что оно было вызвано данной причиной.*

Если не коротко, тогда [сюда](#) и [сюда](#).



A photograph of a chalkboard with the formula for Bayes' theorem written in blue chalk. The formula is
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Иными словами, если мы захотим вычислить $P(\text{мама мыла раму})$, обозначив частоту произвольной n-граммы за f , мы получим формулу

$$P(\text{мама мыла раму}) = f(\text{мама}) \cdot \frac{f(\text{мама мыла})}{f(\text{мама})} \cdot \frac{f(\text{мама мыла раму})}{f(\text{мама мыла})} = f(\text{мама мыла раму}).$$

Вероятность встретить n-грамму

оцениваем по количеству таких n-грамм в обучающем тексте

Вероятность $P(w_1, \dots, w_m)$ встретить предложение (w_1, \dots, w_m) из m слов примерно равна произведению всех n-грамм размера n , из которых состоит это предложение:

$$P(w_1, \dots, w_m) = \prod_{i=1}^m P(w_i \mid w_1, \dots, w_{i-1}) \approx \prod_{i=1}^m P(w_i \mid w_{i-(n-1)}, \dots, w_{i-1})$$

И ЧТО С ЭТИМ ДЕЛАТЬ?

Вероятность каждой из n-граммы определяется через количество раз, которое встретилась эта n-грамма по отношению к количеству раз, которое встретилась такая же n-грамма но без последнего слова:

$$P(w_i \mid w_{i-(n-1)}, \dots, w_{i-1}) = \frac{\text{count}(w_{i-(n-1)}, \dots, w_{i-1}, w_i)}{\text{count}(w_{i-(n-1)}, \dots, w_{i-1})}$$

19.12.2022

А ЧТО ДАЛЬШЕ?

По сути у нас уже есть языковая модель

На практике в чистом виде такую модель не используют, так как у неё есть следующая проблема. Если какая-то n -грамма не встречалась в обучающем тексте — всё предложение сразу же получит нулевую вероятность.

Остался один тонкий момент - если пользователь ввёл совсем странную фразу и соответствующих n -грамм у нас в статистике и нет вовсе? Было бы легко для незнакомых n -грамм положить $\mathbf{f} = \mathbf{0}$, если бы на эту величину не надо было делить. Здесь на помощь приходит сглаживание (smoothing).

[Самый популярный](#)

