Виды токенизаторов

1) Поделить по пробелам "мама мыла раму"

↓

"мама", "мыла" "раму"

2) BPE
byte-pair Encoding

"абаб", "аааа", "бвгд" ...
корпус.

1) ["а", "б", "в", "г", "д"] ← словарь
2) "аа"
3) "аб"
4) "аа" + "а" = "ааа"

Находим пару токенов, которая встречается чаще всего. Добавляем её в словарь.

3) WordPiece

1) ["а", "б", "в", "г", "д"] ← словарь
2) Score = $\frac{\text{freq of pair}}{[\text{freq } w_1] \times [\text{freq } w_2]}$

["под", "сказал"]
["подс", "казал"]

3) добавляем пару наибольшим score.

4) Unigram

1) словарь — все возможные токены. Капцинер, BPE с макс. словарем.

["бам", "бум", "бах"] - корпус

"а", "б", "м", "у", "х", "ба", "ам", "бу", "ум", "ах",
sum(2 3 2 1 1 2 1 1 1 1)
S

частоты

$$P(\text{"ба"}, \text{"м"}) = \frac{2}{S} \cdot \frac{2}{S}$$

$$P(\text{"б"}, \text{"а"}, \text{"м"}) = \frac{3}{S} \cdot \frac{2}{S} \cdot \frac{2}{S}$$

$$P(\text{"б"}, \text{"ам"}) = \frac{3}{S} \cdot \frac{1}{S}$$

2) Для каждого токена считаем насколько уменьшится вероятность при его удалении. Удаляем те, у которых вер-ть уменьшается меньше всего.

Языковые модели

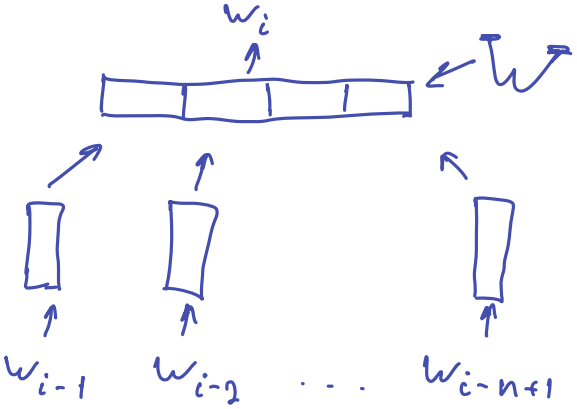
$$P(w_1, \dots, w_T) = p(w_1) \prod_{i=2}^{T} P(w_i \mid w_1, \dots, w_{i-1})$$

N-gram language model

$$p(w_i \mid w_1, \dots, w_{i-1}) \approx p(w_i \mid w_{i-1}, \dots, w_{i-n+1})$$

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

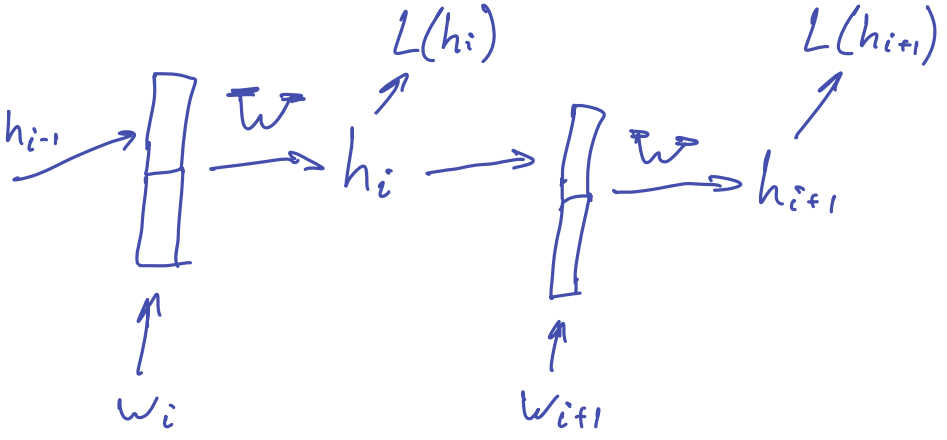
Нейросетевые языковые модели.

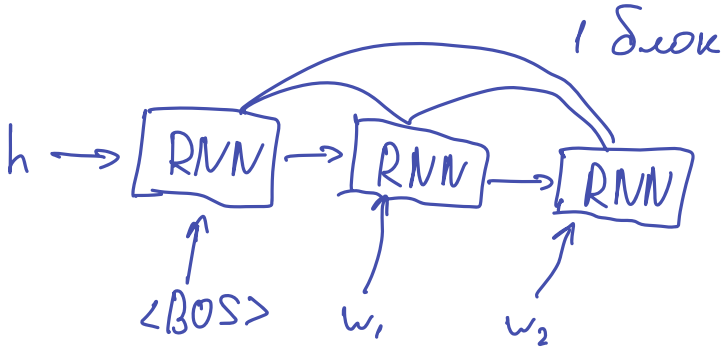


\+ не надо хранить n-граммы

\- увеличение окна увеличивает W

Рекуррентная языковая модель (RNN)





Teacher forcing

$$\nabla_w L(h_i) = \frac{\partial L(h_i)}{\partial h_i} \cdot \frac{\partial h_i}{\partial h_{i-1}} \cdots \frac{\partial h_1}{\partial w} = \frac{\partial L(h_i)}{\partial h_i} \prod_{j=i}^{1} \frac{\partial h_j}{\partial h_{j-1}} \frac{\partial h_1}{\partial w}$$