

# Языковое моделирование

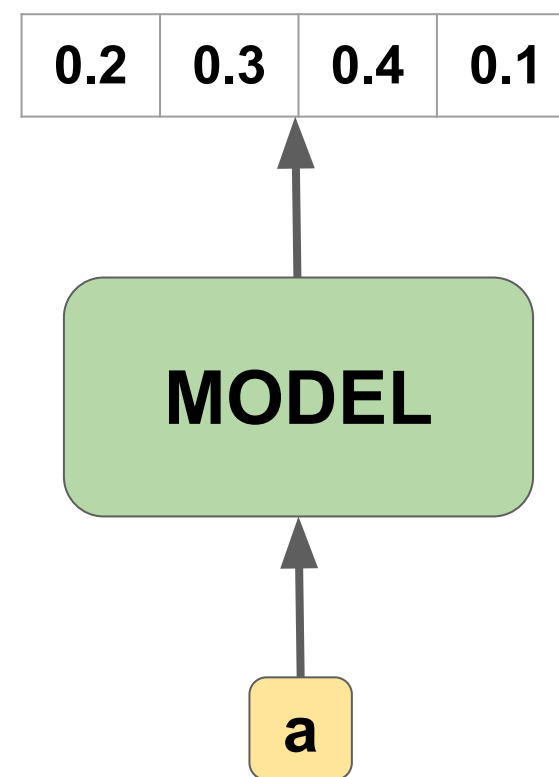
## Лекция 4

Методы генерации текста

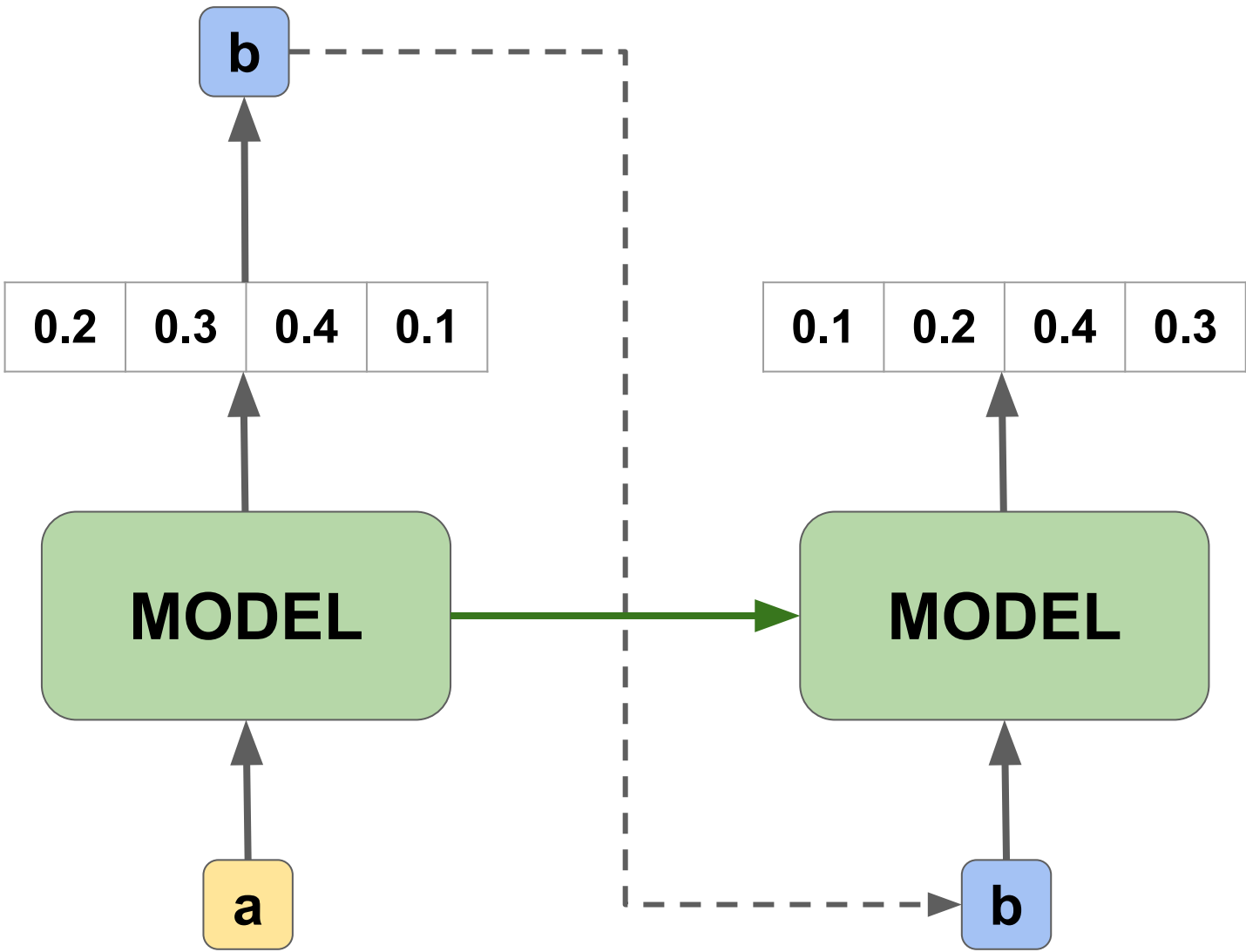
# План занятия

1. Жадный способ генерации
2. Beam Search
3. Top-k sampling
4. Top-p sampling
5. Сравнение подходов

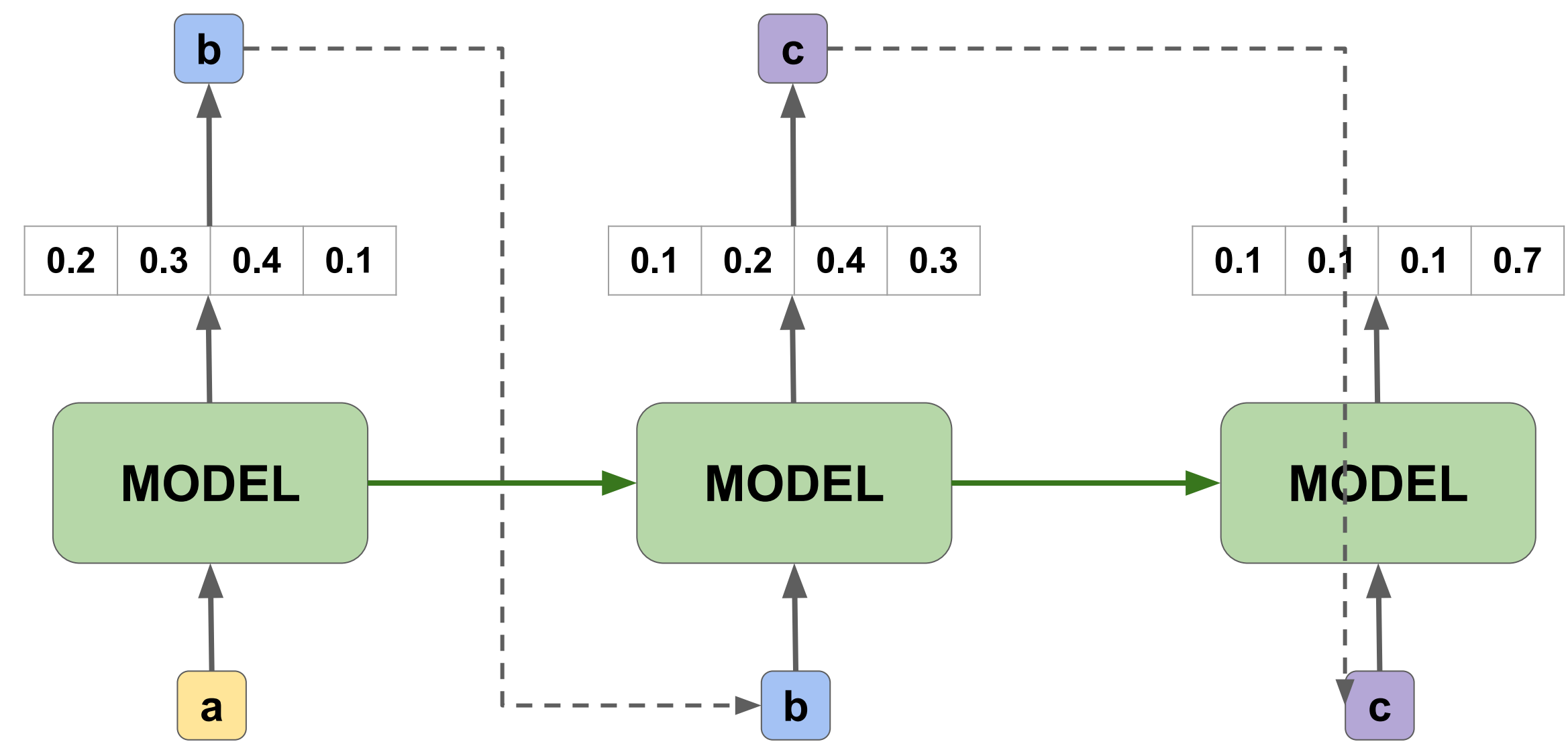
# Как генерировать текст с помощью языковой модели



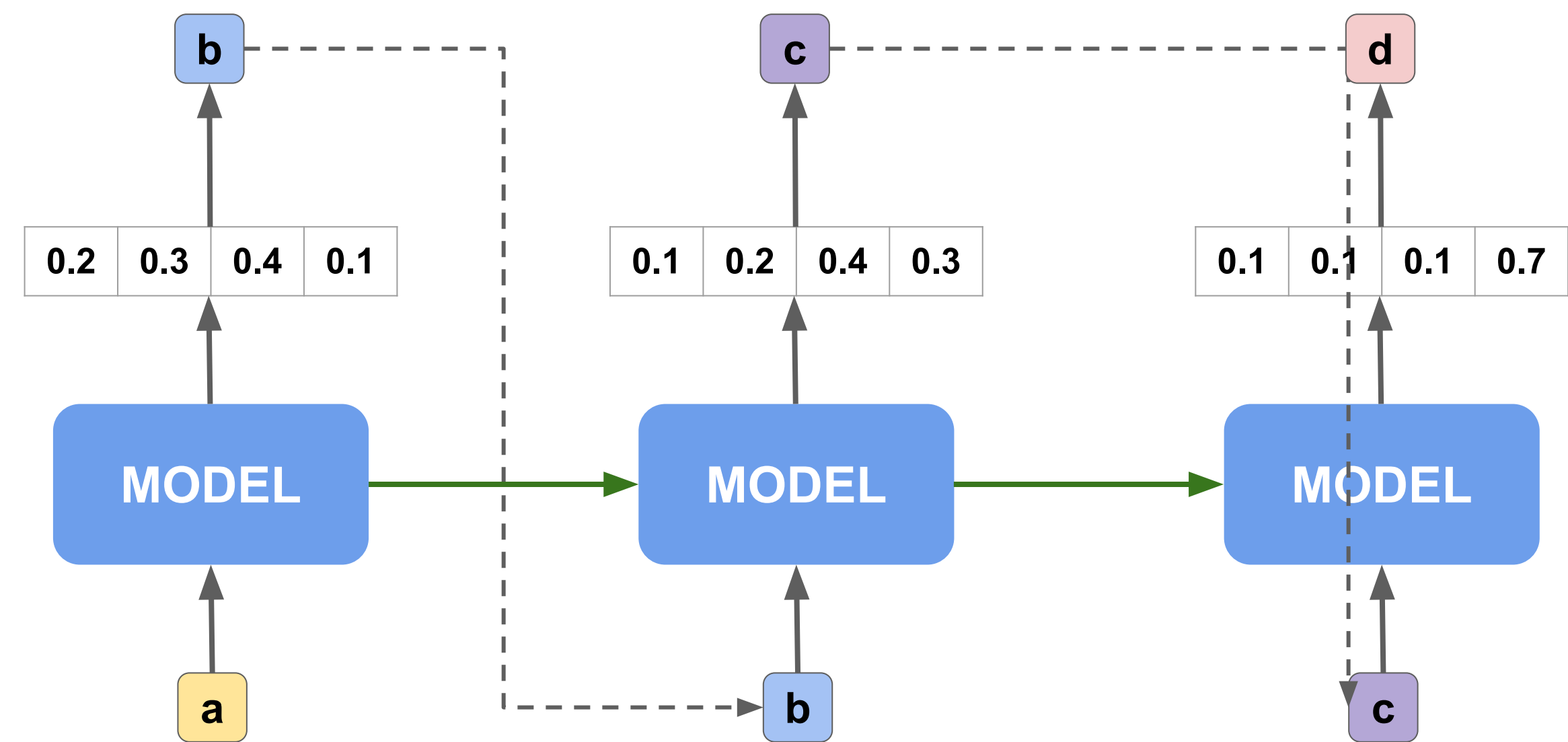
# Как генерировать текст с помощью языковой модели



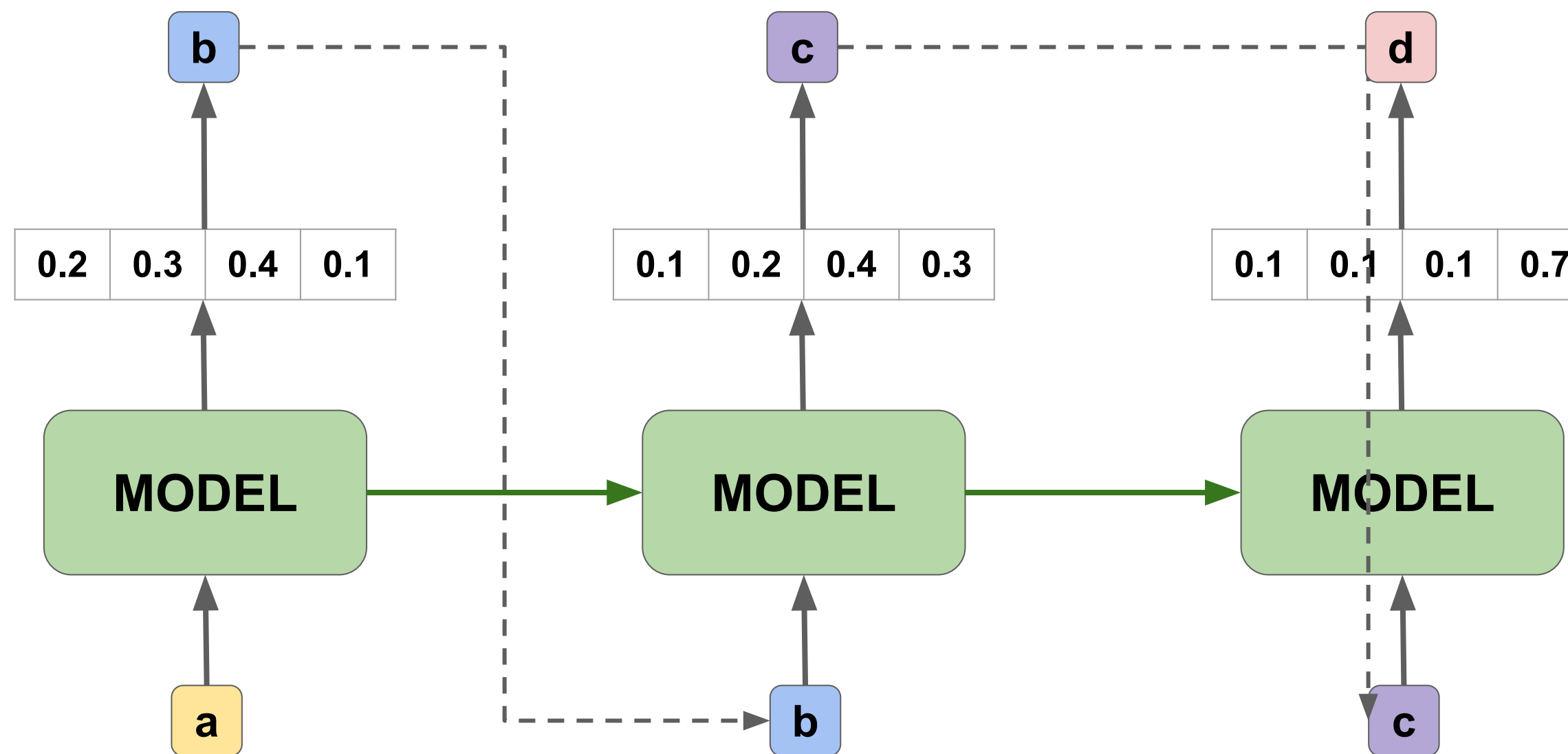
# Как генерировать текст с помощью языковой модели



# Как генерировать текст с помощью языковой модели



# Как генерировать текст с помощью языковой модели



Проблема:

Расчет вероятностей всех возможных вариантов генерации  $O(V^n)$  слишком долгий.

$V$  — размер словаря,  $n$  — длина генерируемой последовательности

# **Greedy — простейшая стратегия выбора следующего слова**

Алгоритм: на каждом шаге выбираем самый вероятный токен.

Сложность:  $O(n)$



# Greedy — простейшая стратегия выбора следующего слова

Алгоритм: на каждом шаге выбираем самый вероятный токен.

Сложность:  $O(n)$


A	0.3
B	0.4
C	0.2
D	0.1

# Greedy — простейшая стратегия выбора следующего слова

Алгоритм: на каждом шаге выбираем самый вероятный токен.

Сложность:  $O(n)$

A	0.3
B	0.4
C	0.2
D	0.1

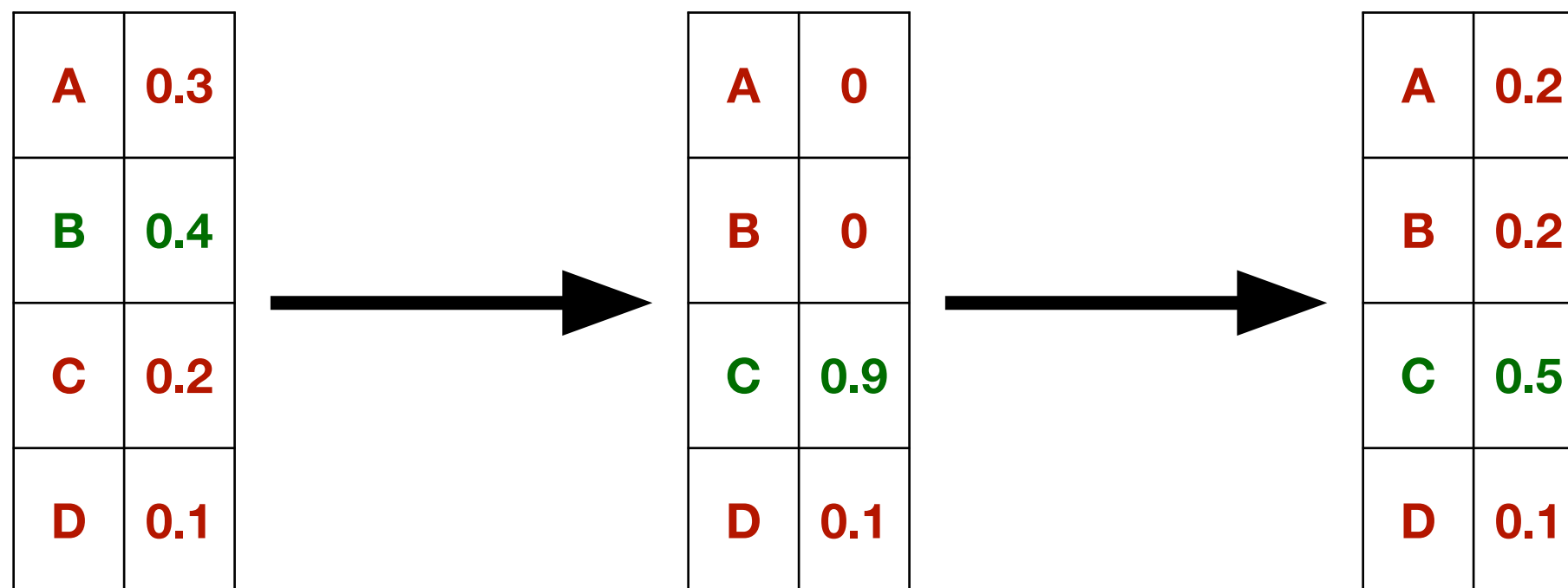


A	0
B	0
C	0.9
D	0.1

# Greedy – простейшая стратегия выбора следующего слова

Алгоритм: на каждом шаге выбираем самый вероятный токен.

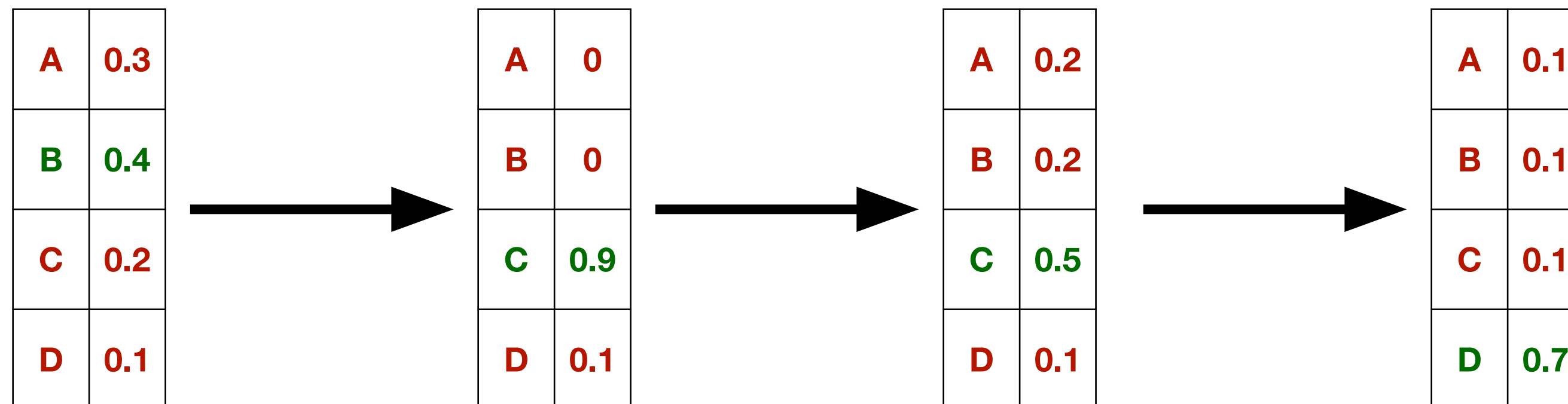
Сложность:  $O(n)$



# Greedy – простейшая стратегия выбора следующего слова

Алгоритм: на каждом шаге выбираем самый вероятный токен.

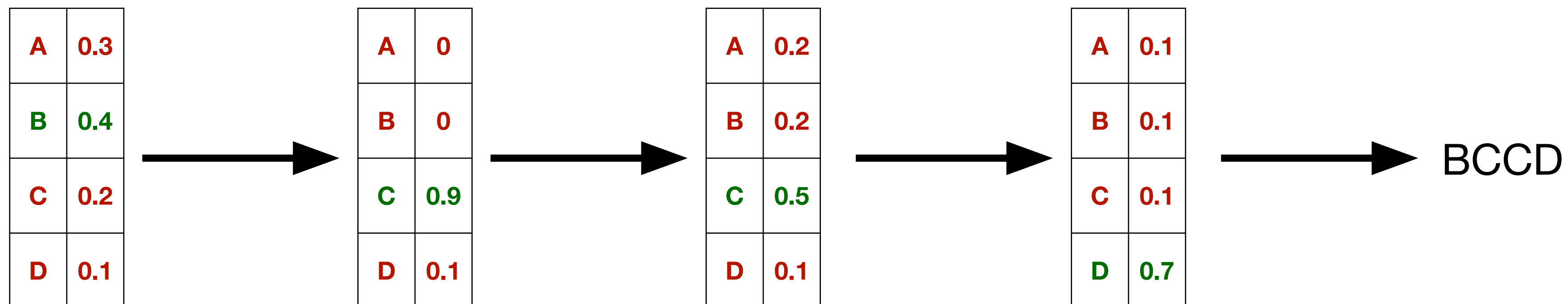
Сложность:  $O(n)$



# Greedy – простейшая стратегия выбора следующего слова

Алгоритм: на каждом шаге выбираем самый вероятный токен.

Сложность:  $O(n)$



# Beam search — ещё один способ генерации

Основная идея: на каждом шаге поддерживать несколько самых вероятных гипотез.

Сложность:  $O(nk)$ , где  $n$  — длина последовательности,  $k$  — количество гипотез.

На практике параметр  $k$  не превышает шести.

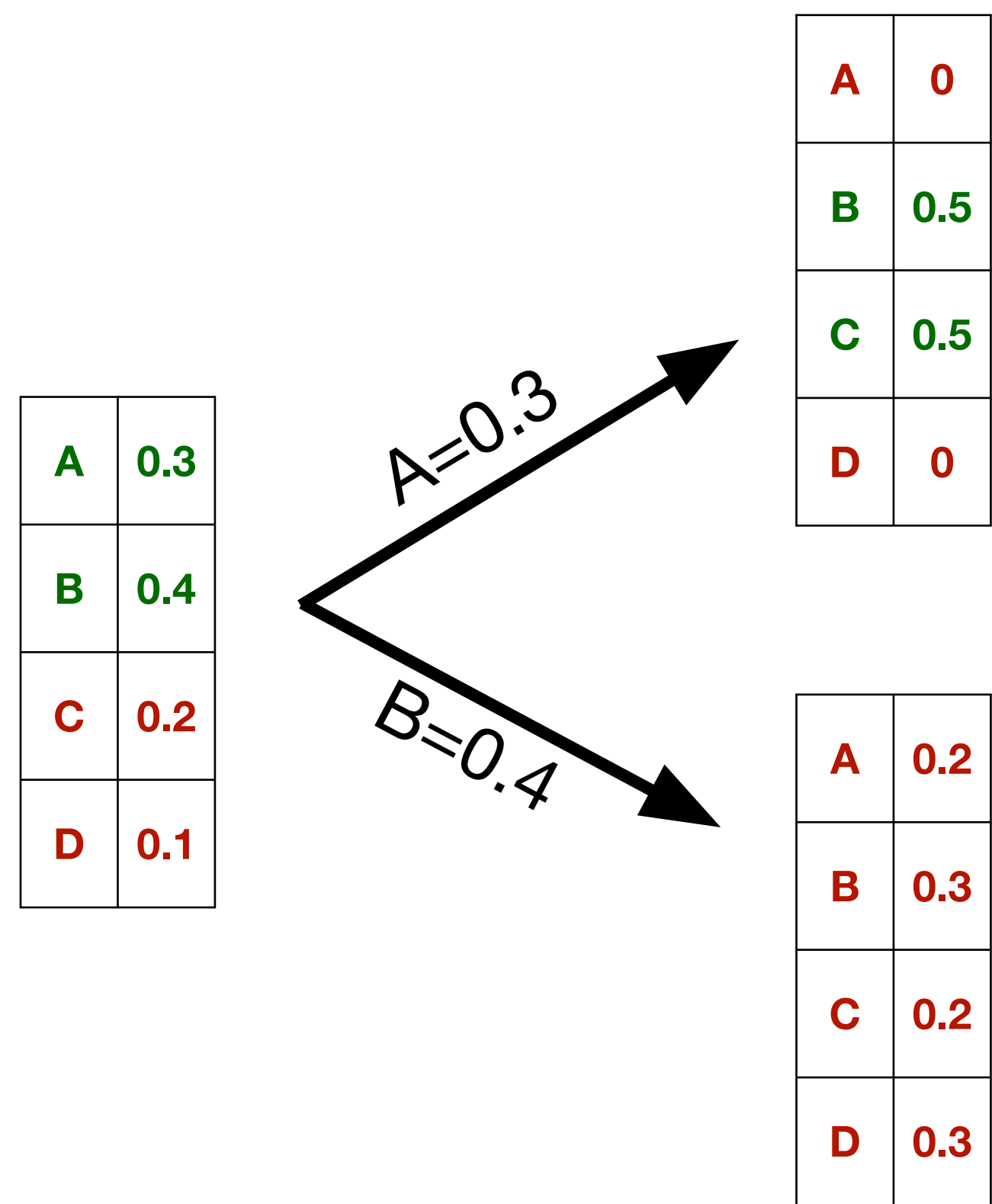
# Пример работы beam search

Параметр  $k=2$

A	0.3
B	0.4
C	0.2
D	0.1

# Пример работы beam search

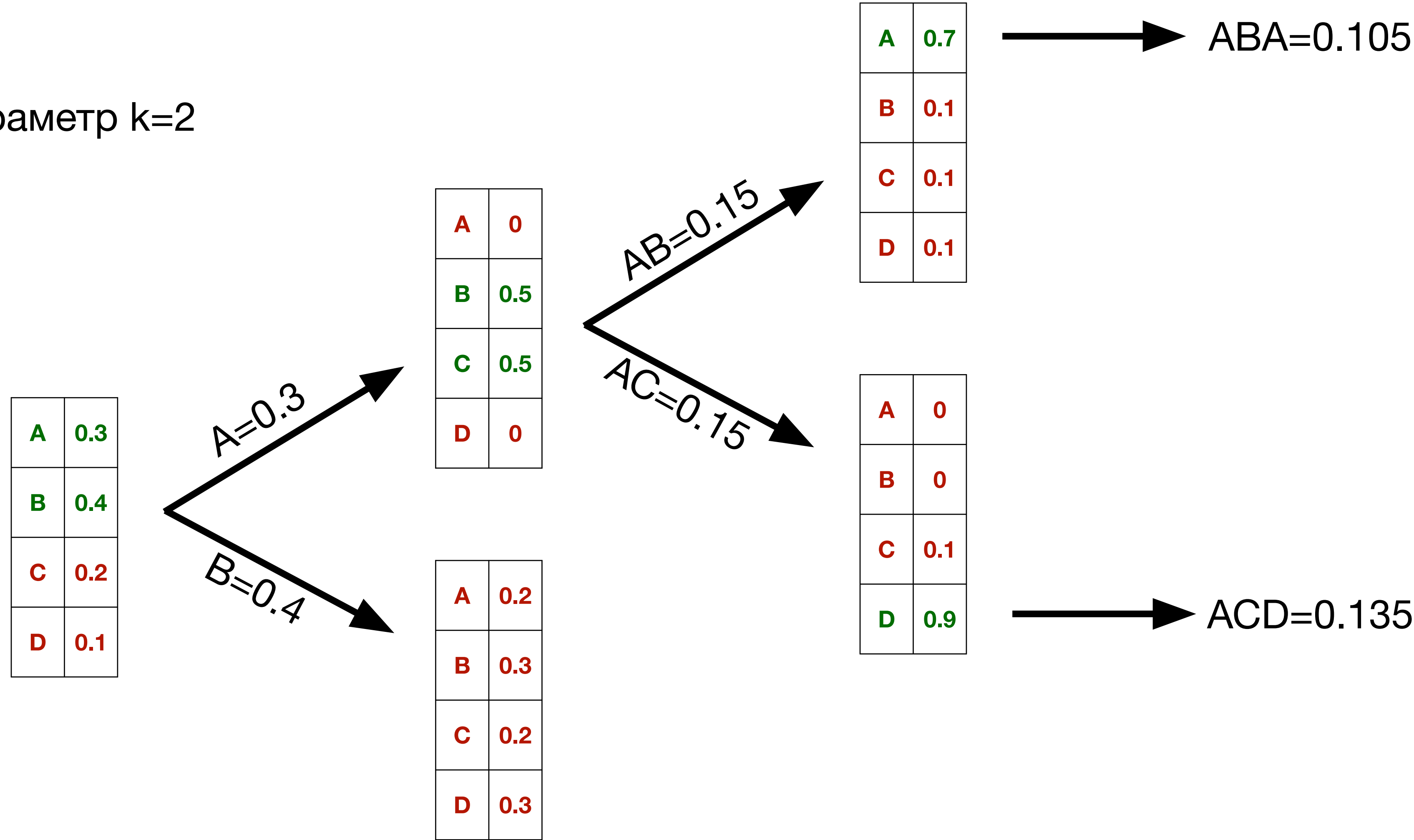
Параметр k=2



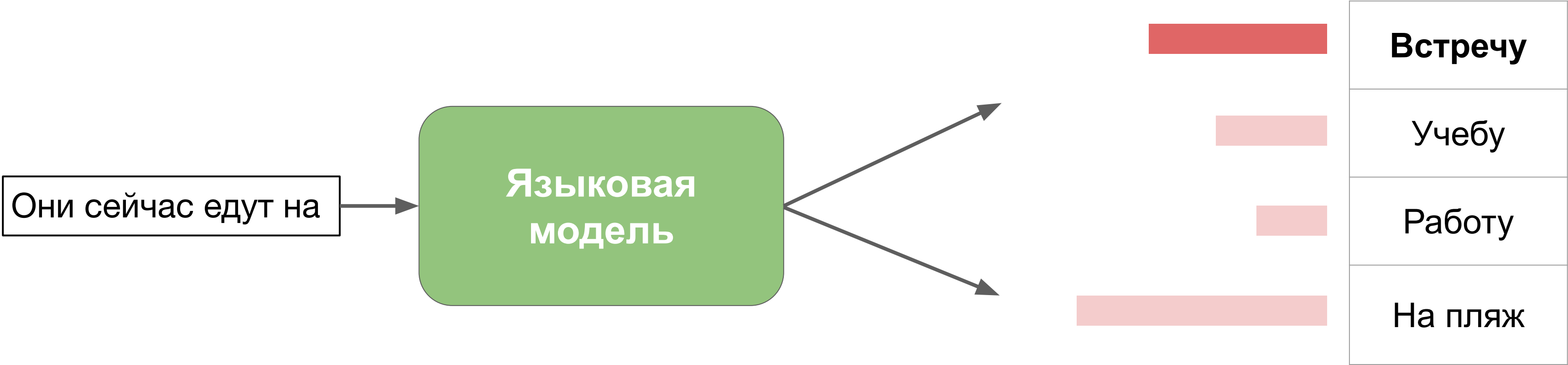


# Пример работы beam search

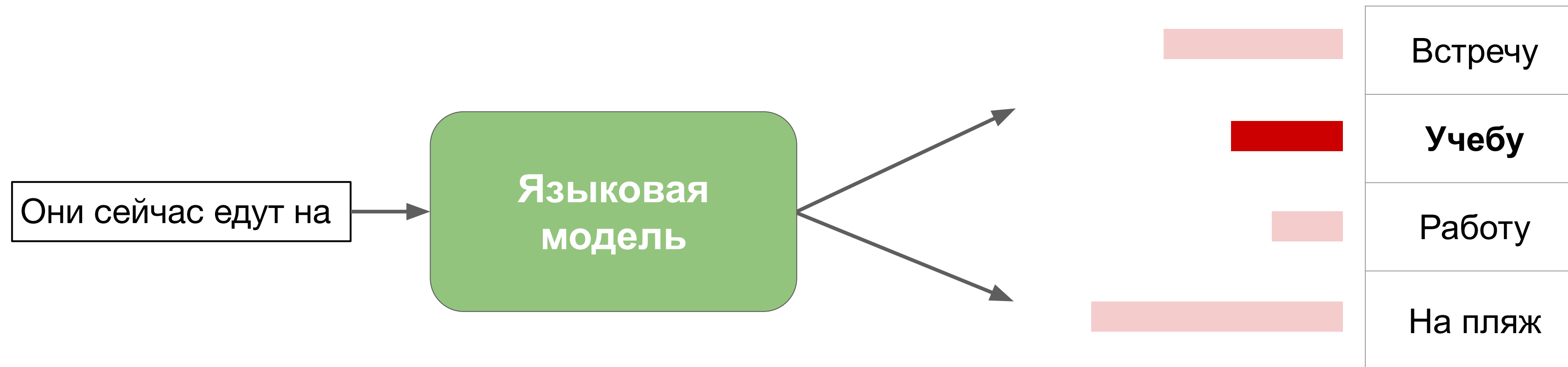
Параметр k=2



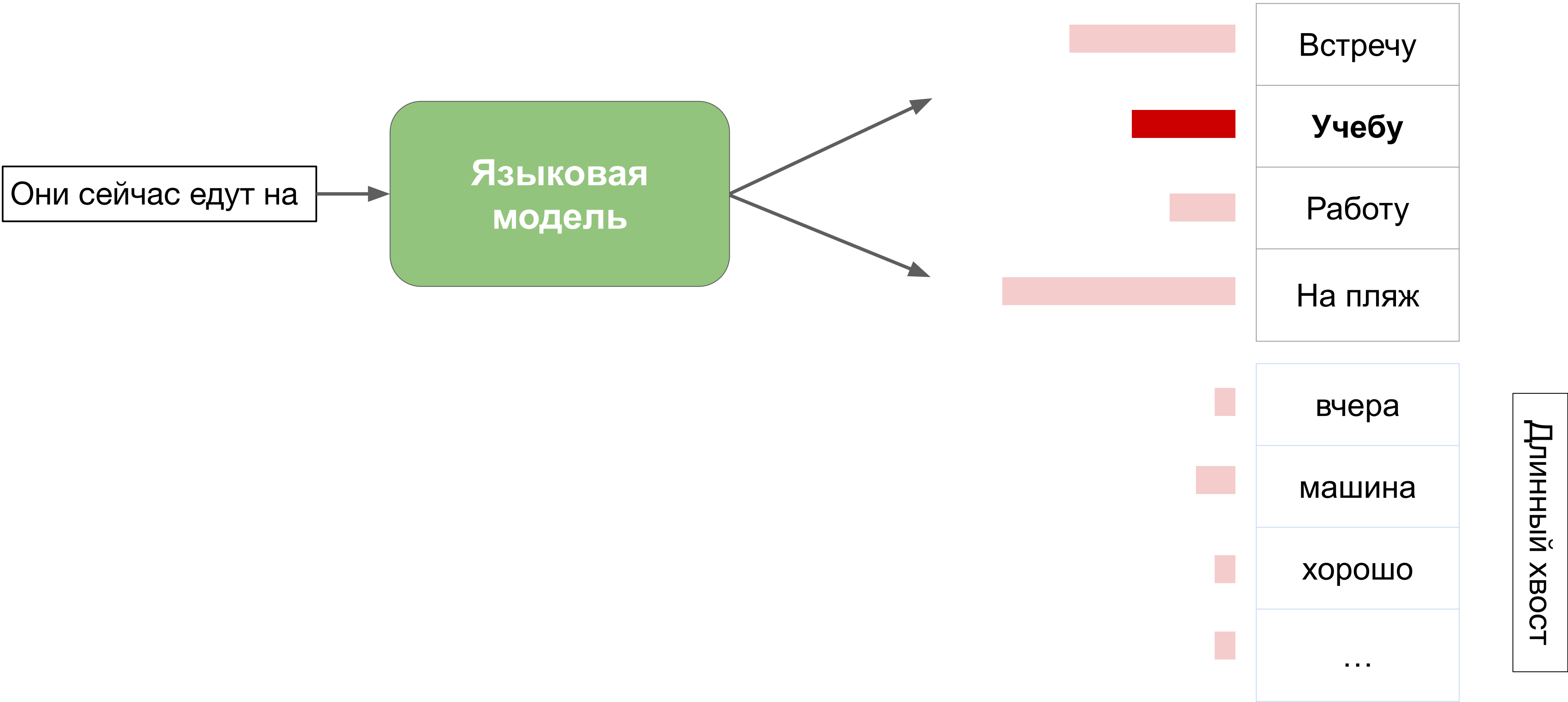
# Наивное сэмплирование



# Наивное сэмплирование

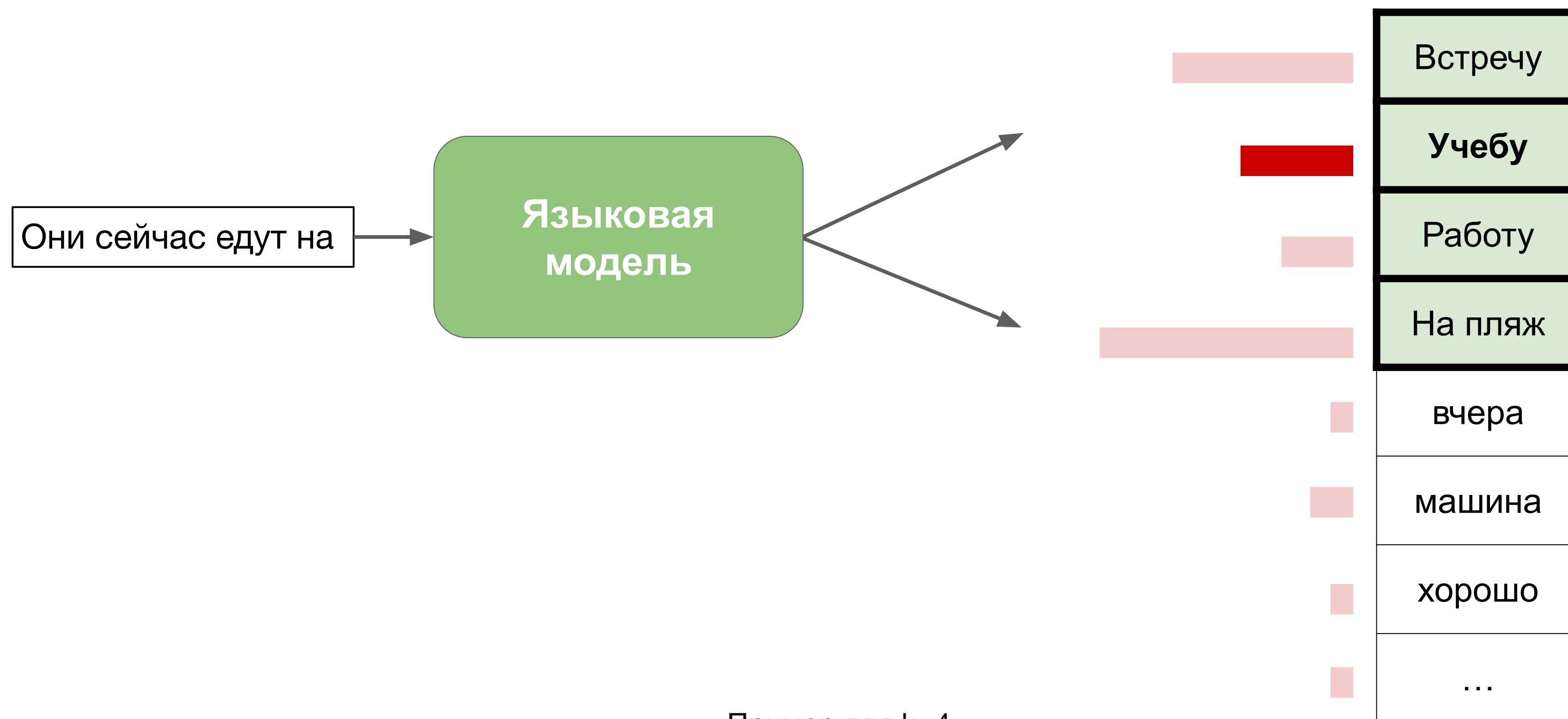


# Проблема наивного сэмплирования



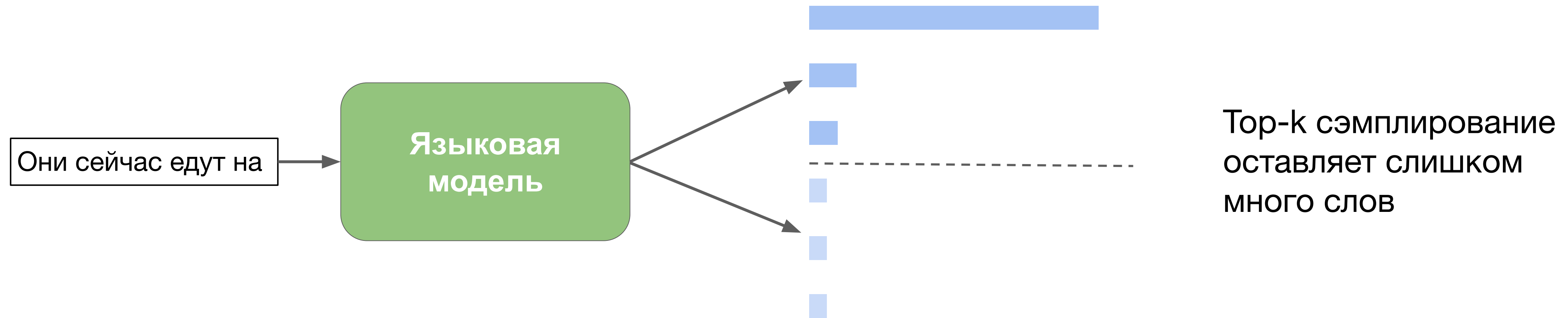
# Тор-к сэмплирование

Возьмем  $K$  наиболее вероятных токенов, где  $K$  — фиксированный гиперпараметр

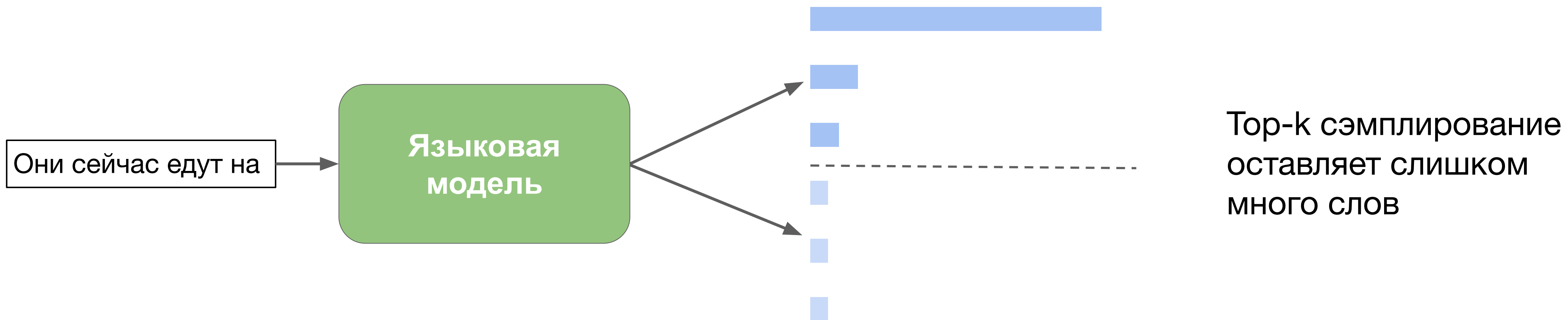


Пример для  $k=4$

# Проблемы top-k сэмплирования



# Проблемы top-k сэмплирования



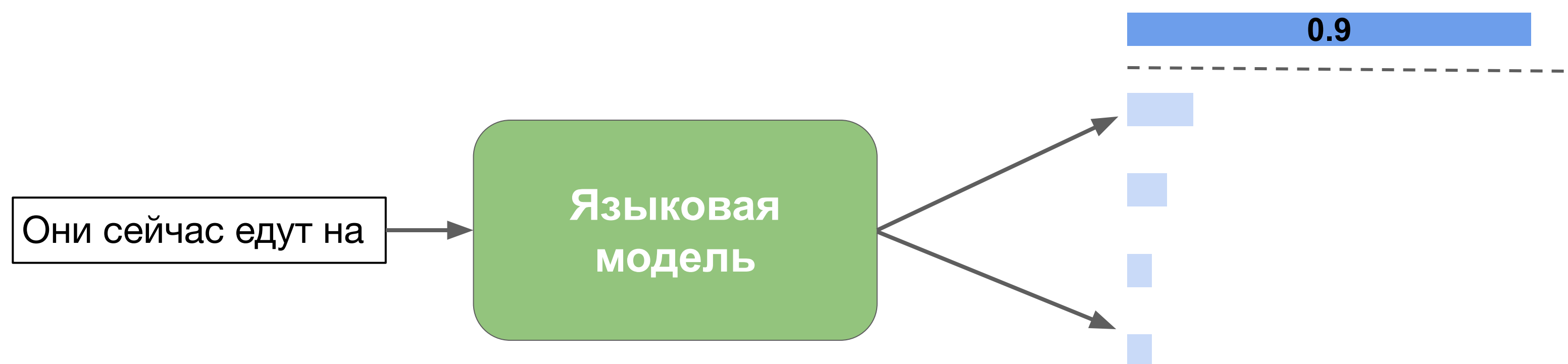
# Top-p (nucleus) сэмплирование

Параметр  $k$  подбирается динамически так, чтобы суммарная вероятность  $k$  наиболее вероятных токенов была выше некоторого порога  $p$ .  $p$  является фиксированным гиперпараметром.



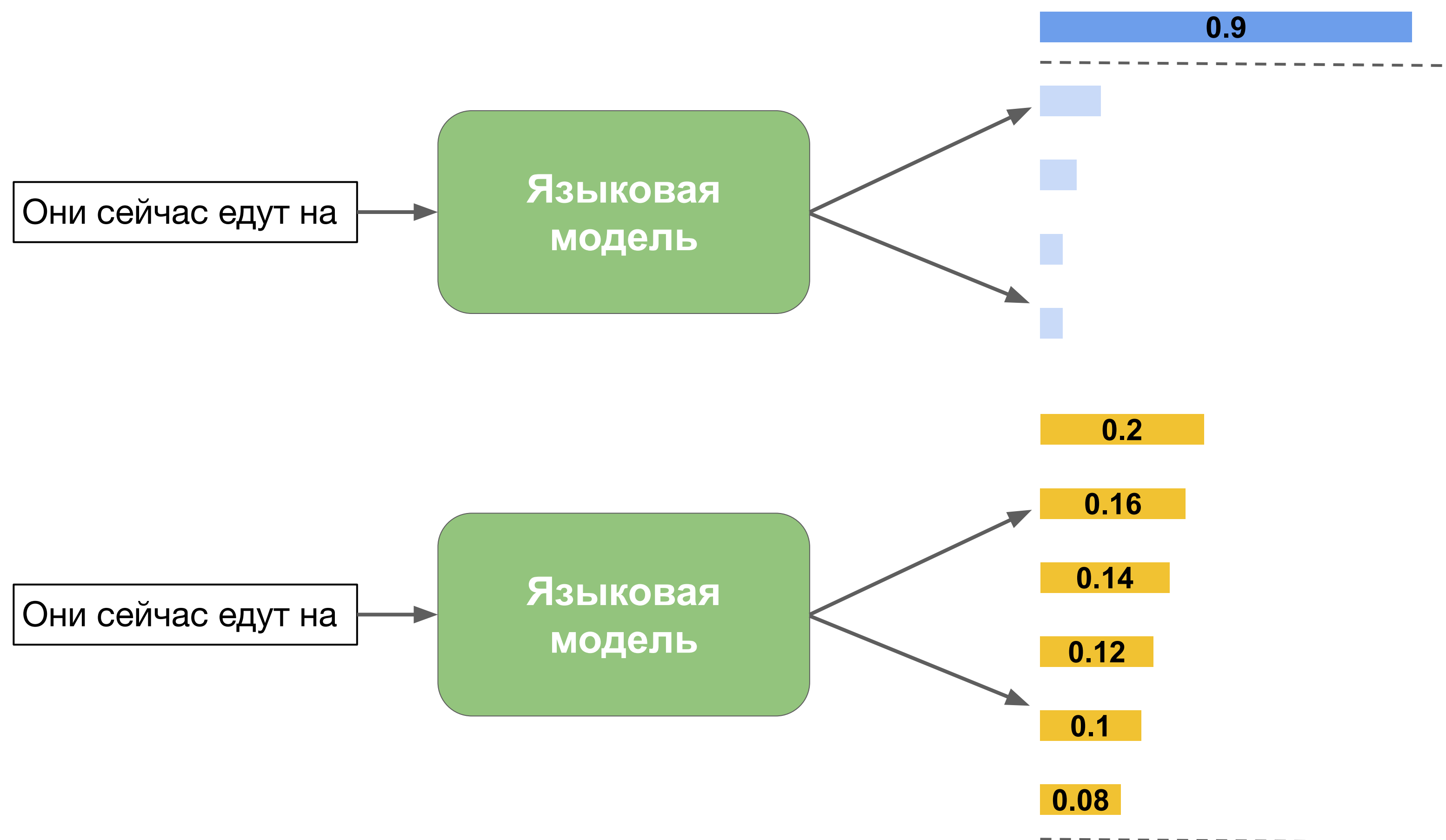
# Top-p (nucleus) сэмплирование

Параметр  $k$  подбирается динамически так, чтобы суммарная вероятность  $k$  наиболее вероятных токенов была выше некоторого порога  $p$ .  $P$  является фиксированным гиперпараметром.

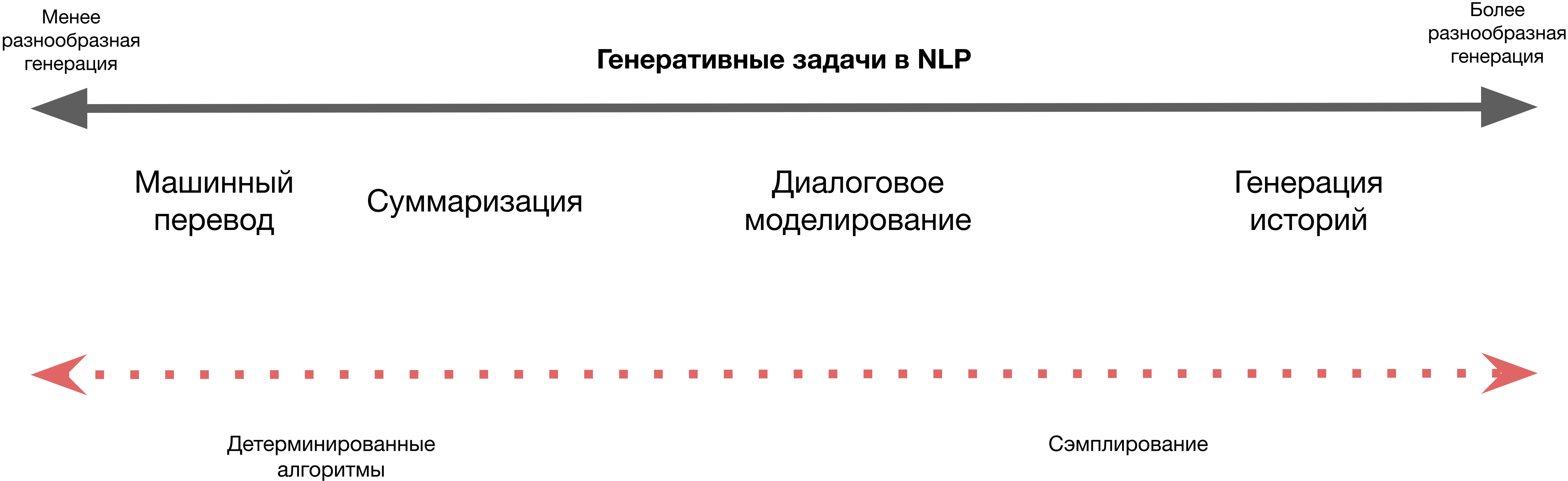


# Top-p (nucleus) сэмплирование

Параметр  $k$  подбирается динамически так, чтобы суммарная вероятность  $k$  наиболее вероятных токенов была выше некоторого порога  $p$ .  $P$  является фиксированным гиперпараметром.



# Зачем нужны различные методы генерации



# Итоги занятия

1. Узнали про сильные и слабые стороны детерминированных алгоритмов генерации: greedy и beam-search
2. Познакомились с разными видами сэмплирования и поняли различия top-p и top-k сэмплирования
3. Поняли зачем существуют столько разных способов генерации текста