Разметка последовательности

Языковые модели

В предыдущих занятиях:

- Мы познакомились с векторной моделью:
  - Документ – вектор
  - Слово – вектор

В предыдущих занятиях:

- Мы познакомились с векторной моделью:
  - Документ – вектор
  - Слово – вектор
- Векторная модель документа и слова **не учитывает порядок слов**

В предыдущих занятиях:

- Мы познакомились с векторной моделью:
  - Документ – вектор
  - Слово – вектор
- Векторная модель документа и слова **не учитывает порядок слов**
- **Языковая модель** – вероятностная модель, которые учитывают порядок слов

Языковая модель

- Предсказать слово в последовательности:

«на улице идет дождь, а у нас идет

Языковая модель

- Предсказать слово в последовательности:

«на улице идет дождь, а у нас идет концерт»

Языковая модель

- **Предсказать слово в последовательности:**

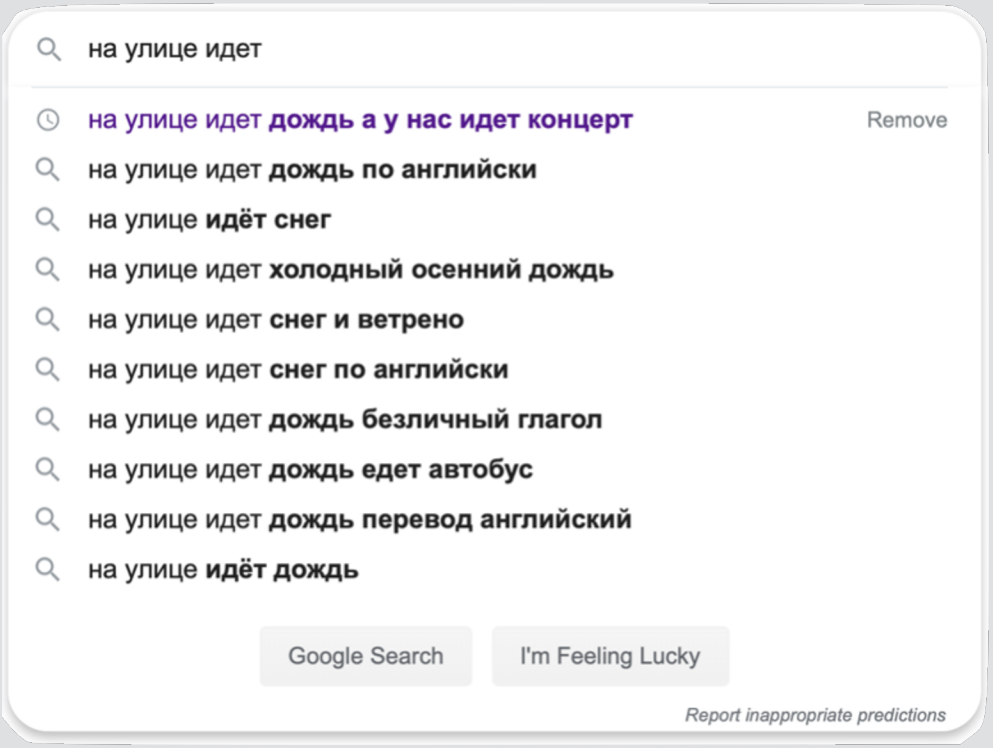
«на улице идет дождь, а у нас идет концерт»

- **Выбрать более вероятное предложение:**

P(«на улице идет дождь») $>$ P(«на улице идет концерт»)

Пример: поисковые подсказки

Google



Вероятностная языковая модель

- Формальные задачи:
  - Пусть $W = (w_1, w_2, \ldots, w_n)$ – предложение, w_i – слово, V – словарь

Вероятностная языковая модель

- Формальные задачи:
  - Пусть $W = (w_1, w_2, \ldots, w_n)$ – предложение, w_i – слово, V – словарь
- Вероятность предложения W:
  - $P(W) = P(w_1, w_2, \ldots, w_n)$

Вероятностная языковая модель

- Формальные задачи:
  - Пусть $W = (w_1, w_2, \ldots, w_n)$ – предложение, w_i – слово, V – словарь
- Вероятность предложения W:
  - $P(W) = P(w_1, w_2, \ldots, w_n)$
- Вероятность следующего слова в последовательности:
  - $P(w_n | w_{n-1}, w_{n-2}, \ldots, w_1)$

Как оценить вероятность предложения?

- Разворачиваем последовательность слов:

Как оценить вероятность предложения?

- Разворачиваем последовательность слов:

$P(W) =$

Как оценить вероятность предложения?

- Разворачиваем последовательность слов:

$P(W) =$

$= P(w_1, w_2, \ldots, w_n) =$

Как оценить вероятность предложения?

- Разворачиваем последовательность слов:

$$P(W) =$$

$$= P(w_1, w_2, \ldots, w_n) =$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)\ldots P(w_n|w_1, \ldots w_n - 1)$$

Как оценить вероятность предложения?

- Разворачиваем последовательность слов:

$$P(W) =$$

$$= P(w_1, w_2, \ldots, w_n) =$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)\ldots P(w_n|w_1, \ldots w_n - 1)$$

$P(\text{«на улице идет дождь»}) =$

Как оценить вероятность предложения?

- Разворачиваем последовательность слов:

$$P(W) =$$

$$= P(w_1, w_2, \ldots, w_n) =$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)\ldots P(w_n|w_1, \ldots w_n - 1)$$

$$P(\text{«на улице идет дождь»}) =$$

$$= P(\text{«на»}) \cdot P(\text{«улице»}|\text{«на»}) \cdot P(\text{«идет»}|\text{«на улице»}) \cdot$$

$$\cdot P(\text{«дождь»}|\text{«на улице идет»})$$

Как оценить вероятность предложения?

- Обозначим $w_{1:i-1}$ – предыстория слова w_i

Как оценить вероятность предложения?

- Обозначим $w_{1:i-1}$ – предыстория слова w_i
- Сворачиваем последовательность слов:

$$P(W) = P(w_1, w_2, \ldots, w_n) = \prod_i P(w_i | w_{1:i-1})$$

Как оценить вероятность предложения?

$$P(w_i|w_{1:w_i-1}) = \frac{\text{count}(w_{1:i-1}w_i)}{\text{count}(w_{1:i-1})}$$

Как оценить вероятность предложения?

$$P(w_i|w_{1:w_i-1}) = \frac{\text{count}(w_{1:i-1}w_i)}{\text{count}(w_{1:i-1})}$$

В Национальном корпусе русского языка:

$P(\text{«на улице идет дождь»}) =$

$$= \frac{\text{count}(\text{«на улице идет дождь»})}{\text{count}(\text{«на улице идет»})} = \frac{5}{15} = \frac{1}{3}$$

А. А. Марков

1856 – 1922

один из самых известных математиков российского происхождения

Марковская модель

- Дано слово w_i и его предыстория $w_{1:i-1}$

Марковская модель

- Дано слово w_i и его предыстория $w_{1:i-1}$
- Ограничим длину предыстории:

Марковская модель

- Дано слово w_i и его предыстория $w_{1:i-1}$
- Ограничим длину предыстории:
  - Модель униграмм: $P(w_i)$

$$P(\text{«на улице идет дождь»}) = P(\text{«на»})P(\text{«улице»})P(\text{«идет»})P(\text{«дождь»})$$

Марковская модель

- Дано слово w_i и его предыстория $w_{1:i-1}$
- Ограничим длину предыстории:
  - Модель униграмм: $P(w_i)$

$$P(\text{«на улице идет дождь»}) = P(\text{«на»})P(\text{«улице»})P(\text{«идет»})P(\text{«дождь»})$$

  - Модель биграмм : $P(w_i|w_{i-1})$

$$P(\text{«на улице идет дождь»}) =$$
$$= P(\text{«на»})P(\text{«улице»}|\text{«на»})P(\text{«идет»}|\text{«улице»})P(\text{«дождь»}|\text{«идет»})$$

Оценки максимального правдоподобия

- Оценка $P(w_i)$: $\frac{\text{count}(w_i)}{\sum_{w_j \in V} \text{count}(w_j)}$

  V – словарь, все уникальные слова

Оценки максимального правдоподобия

- Оценка $P(w_i)$: $\frac{\text{count}(w_i)}{\sum_{w_j \in V} \text{count}(w_j)}$

  V – словарь, все уникальные слова

- Оценка $P(w_i|w_{i-1})$: $\frac{\text{count}(w_{i-1}w_i)}{\text{count}(w_{i-1})}$

Проблемы

- $\text{count}(w_{i-1}w_i)$ может быть равно 0
- Тогда $\dfrac{\text{count}(w_{i-1}w_i)}{\text{count}(w_{i-1})} = 0$ и $P(W) = 0$

$$\text{count}(\text{«идет снег»}) = 0$$

$$\frac{\text{count}(\text{«идет снег»})}{\text{count}(\text{«снег»})} = 0$$

$$\text{count}(\text{«на улице идет снег»}) = 0$$

Проблемы

- $\text{count}(w_{i-1}w_i)$ может быть равно 0
- Тогда $\frac{\text{count}(w_{i-1}w_i)}{\text{count}(w_{i-1})} = 0$ и $P(W) = 0$

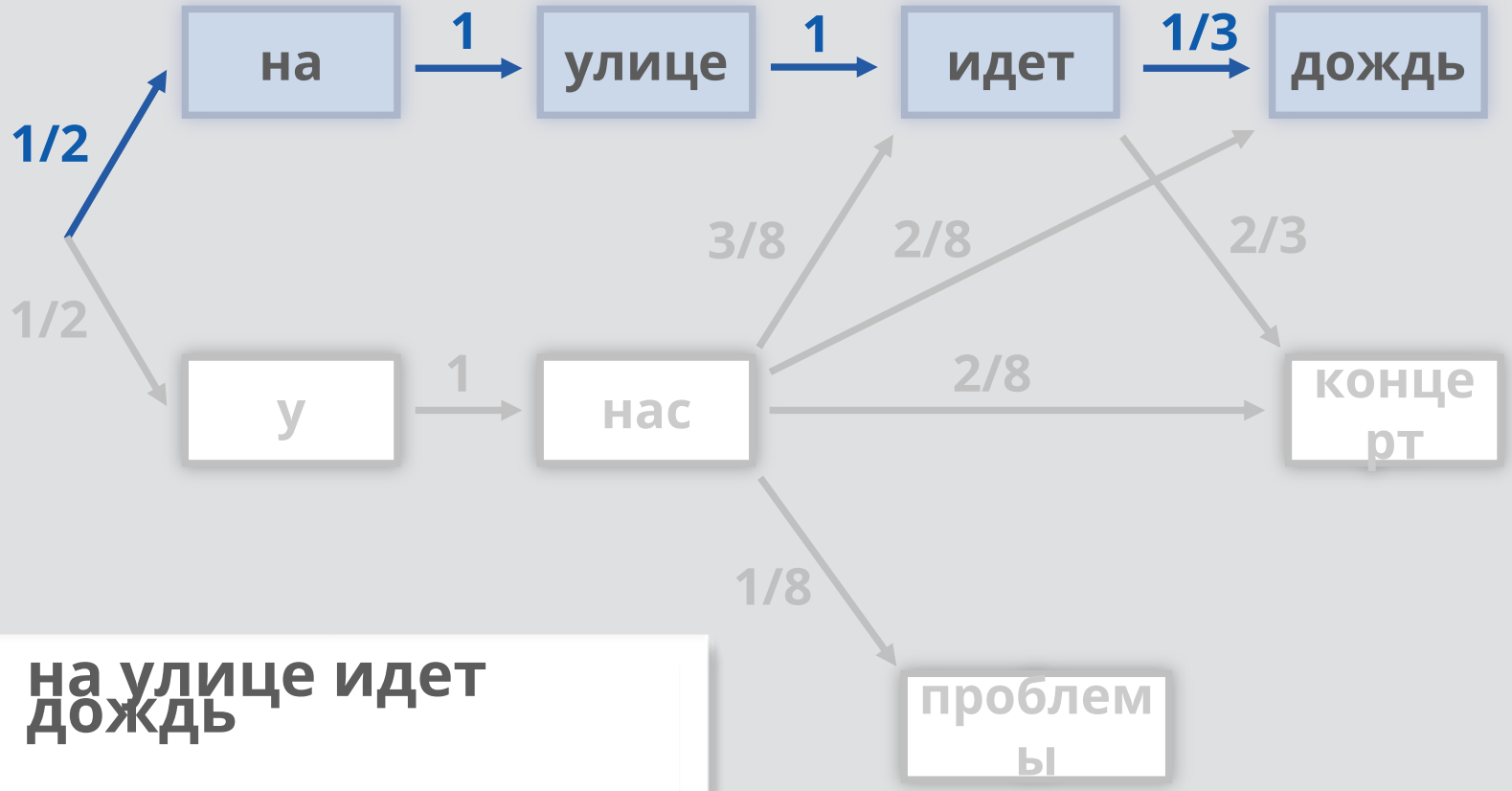
Решение: сглаживание

- Допустим, каждое слово встречается минимум α раз

$$\frac{\text{count}(w_{i-1}w_i) + \alpha}{\text{count}(w_{i-1}) + \alpha|V|}$$

Генерация текстов

Генерация текстов



- на улице идет дождь

Генерация текстов

- на улице идет дождь
- на улице идет концерт

Генерация текстов

- на улице идет дождь
- на улице идет концерт
- у нас идет концерт

Генерация текстов

- на улице идет дождь
- на улице идет концерт
- у нас идет концерт
- у нас проблемы

Проблемы счетных языковых моделей

- Генерируют несогласованный и неосмысленный текст

Проблемы счетных языковых моделей

- Генерируют несогласованный и

  - На улице заметили странную рыбу по горячим следам

  Летом прогнозируют похолодание может откусить хлеба

Проблемы счетных языковых моделей

- Генерируют несогласованный и неосмысленный текст
  - **На улице заметили странную рыбу по горячим следам**
  - **Летом прогнозируют похолодание может откусить хлеба**
- Требуют огромного количества параметров:
  - Модель биграмм насчитывает $|V|^2$ параметров

Качество языковых моделей

- **Внешняя** оценка качества:
  - Используем языковую модель в последующей задаче
  - Оцениванием качество этой задачи

Качество языковых моделей

- Внутренняя оценка качества: перплексия

$$PP(W) = P(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} =$$

$$= \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N)}}$$ – среднее геометрическое вероятностей

Качество языковых моделей

- Внутренняя оценка качества: перплексия

$$PP(W) = P(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} =$$

$$= \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N)}}$$ – среднее геометрическое вероятностей

- Чем меньше перплексия, тем лучше
- Чем лучше языковая модель, тем лучше она предсказывает текст

Качество языковых моделей

- Внутренняя оценка качества: перплексия

$$PP(W) = P(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N)}} -$$

среднее геометрическое вероятностей

- Чем меньше перплексия, тем лучше
- Чем лучше языковая модель, тем лучше она предсказывает текст
- PP в модели биграмм: $PP(W) = \sqrt[N]{\frac{1}{\prod_{i=1}(w_i|w_{i-1})}}$

Счетные языковые модели

- Языковые модели моделируют **вероятность последовательности слов**

Счетные языковые модели

- Языковые модели моделируют **вероятность последовательности слов**
- **Модель биграмм:** оцениваем вероятность слова после предыдущего

Счетные языковые модели

- Языковые модели моделируют **вероятность последовательности слов**
- **Модель биграмм:** оцениваем вероятность слова после предыдущего
- **Сглаживание** помогает избежать проблемы нулевых значений

Счетные языковые модели

- Языковые модели моделируют **вероятность последовательности слов**
- **Модель биграмм:** оцениваем вероятность слова после предыдущего
- **Сглаживание** помогает избежать проблемы нулевых значений
- Мера качества: **перплексия**; чем меньше перплексия, тем лучше

Разметка последовательности

Разметка последовательности

- Мы умеем генерировать последовательность слов – текст
- Как ввести дополнительные ограничения в модель генерации текстов?
- Как использовать Марковские модели для анализа последовательностей?

Морфологический анализ

люблю	грозу	в	начале	мая
ГЛАГ	СУЩ	ПРЕД	СУЩ	СУЩ

- Неоходимый шаг для разрешения неоднозначности
- Фильтрация по частям речи сокращает признаковое пространство при классификации текстов
- Части речи
  - **Открытые:** глаголы, существительные, прилагательные, наречия, имена собственные
  - **Закрытые:** союзы, предлоги, местоимения, числительные, частицы, артикли,

Морфологический анализ

Только по контексту можно определить, к какой части речи относится слово:

Морфологический анализ

Только по контексту можно определить, к какой части речи относится слово:

- Дети за лето **стали** сильнее.
- Рабочие выплавили много **стали** за смену.
- Эти типы **стали** есть в цехе.

Морфологический анализ

Только по контексту можно определить, к какой части речи относится слово:

- **Дети за лето стали сильнее.**
- **Рабочие выплавили много стали за смену.**
- **Эти типы стали есть в цехе.**

стали

стать, **ГЛАГ**

сталь, **СУЩ**

Методы морфологического анализа

Частотный морфологический анализ: нам понадобится большой размеченный корпус

```
# sent_id = 2003Nobelevskie_premii.xml_2
# text = Лауреатами стали не врачи и даже не биологи.
1   Лауреатами  лауреат NOUN    _   Animacy=Anim|Case=Ins|Gender=Masc|Number=Plur   2   obl 2:obl   _
2   стали   стать   VERB    _   Aspect=Perf|Mood=Ind|Number=Plur|Tense=Past|VerbForm=Fin|Voice=Mid  0   root    0:root  _
3   не  не  PART    _   _   4   advmod  4:advmod    _
4   врачи   врач    NOUN    _   Animacy=Anim|Case=Nom|Gender=Masc|Number=Plur   2   nsubj   2:nsubj _
5   и   и   CCONJ   _   _   8   cc  8:cc    _
6   даже    даже    PART    _   _   8   advmod  8:advmod    _
7   не  не  PART    _   _   8   advmod  8:advmod    _
8   биологи биолог  NOUN    _   Animacy=Anim|Case=Nom|Gender=Masc|Number=Plur   4   conj    4:conj  SpaceAfter=No
9   .   .   PUNCT   _   _   2   punct   2:punct _
```

Методы морфологического анализа

Частотный морфологический анализ: нам понадобится большой размеченный корпус

```
# sent_id = 2003Nobelevskie_premii.xml_2
# text = Лауреатами стали не врачи и даже не биологи.
1   Лауреатами  лауреат NOUN    _   Animacy=Anim|Case=Ins|Gender=Masc|Number=Plur   2   obl 2:obl   _
2   стали   стать   VERB    _   Aspect=Perf|Mood=Ind|Number=Plur|Tense=Past|VerbForm=Fin|Voice=Mid  0   root    0:root  _
3   не  не  PART    _   _   4   advmod  4:advmod    _
4   врачи   врач    NOUN    _   Animacy=Anim|Case=Nom|Gender=Masc|Number=Plur   2   nsubj   2:nsubj _
5   и   и   CCONJ   _   _   8   cc  8:cc    _
6   даже    даже    PART    _   _   8   advmod  8:advmod    _
7   не  не  PART    _   _   8   advmod  8:advmod    _
8   биологи биолог  NOUN    _   Animacy=Anim|Case=Nom|Gender=Masc|Number=Plur   4   conj    4:conj  SpaceAfter=No
9   .   .   PUNCT   _   _   2   punct   2:punct _
```

Пример: «стали»

- Оценим: $P(\text{стали} \mid \text{ГЛАГ})$ и $P(\text{стали} \mid \text{СУЩ})$
- Будем выбирать более вероятную часть речи

Скрытые марковские модели

Скрытые Марковские модели (Hidden Markov Models, HMM):

Скрытые марковские модели

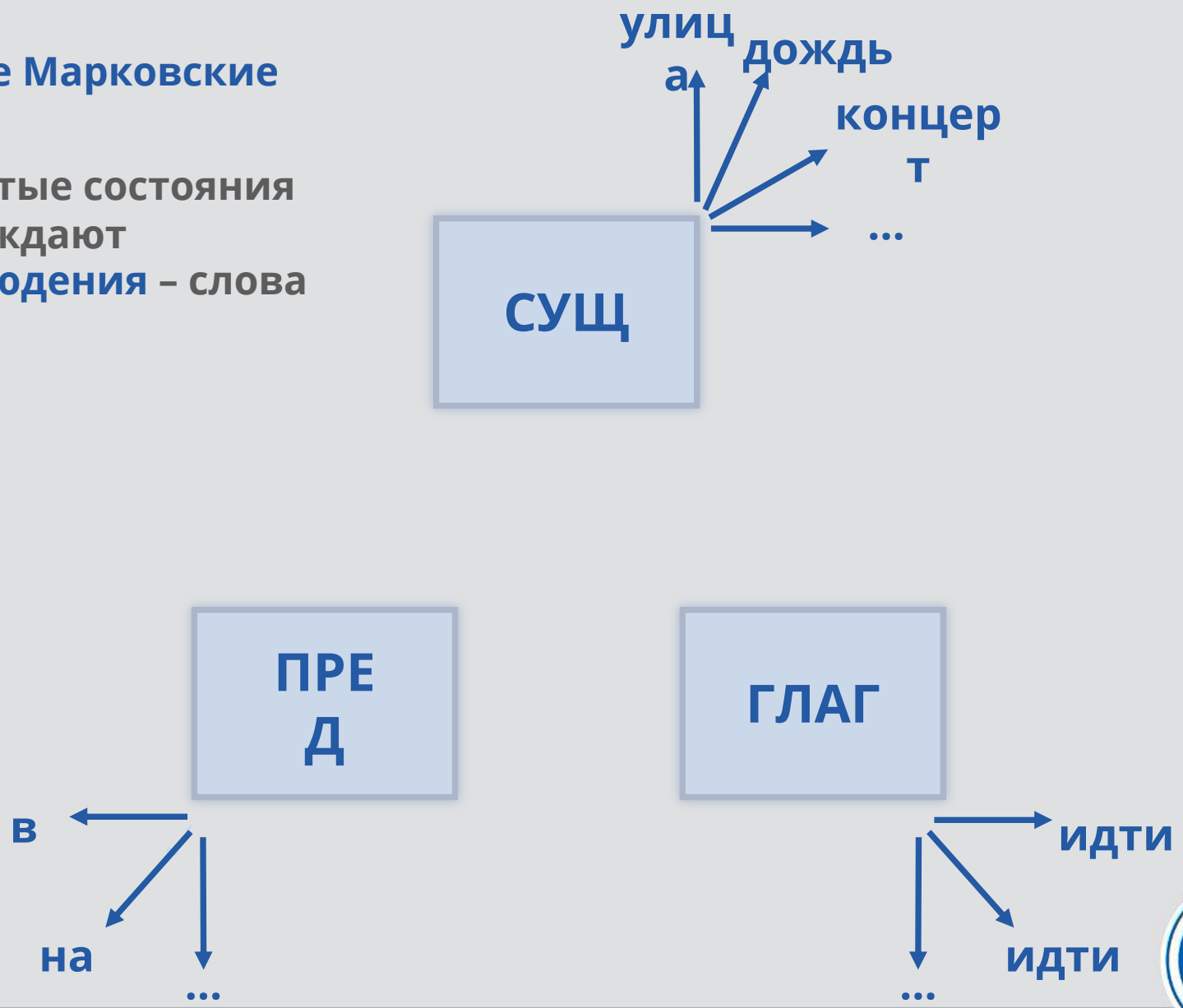
Скрытые Марковские модели:

- **Добавим в цепи Маркова скрытые состояния – части речи**

Скрытые марковские модели

Скрытые Марковские модели:

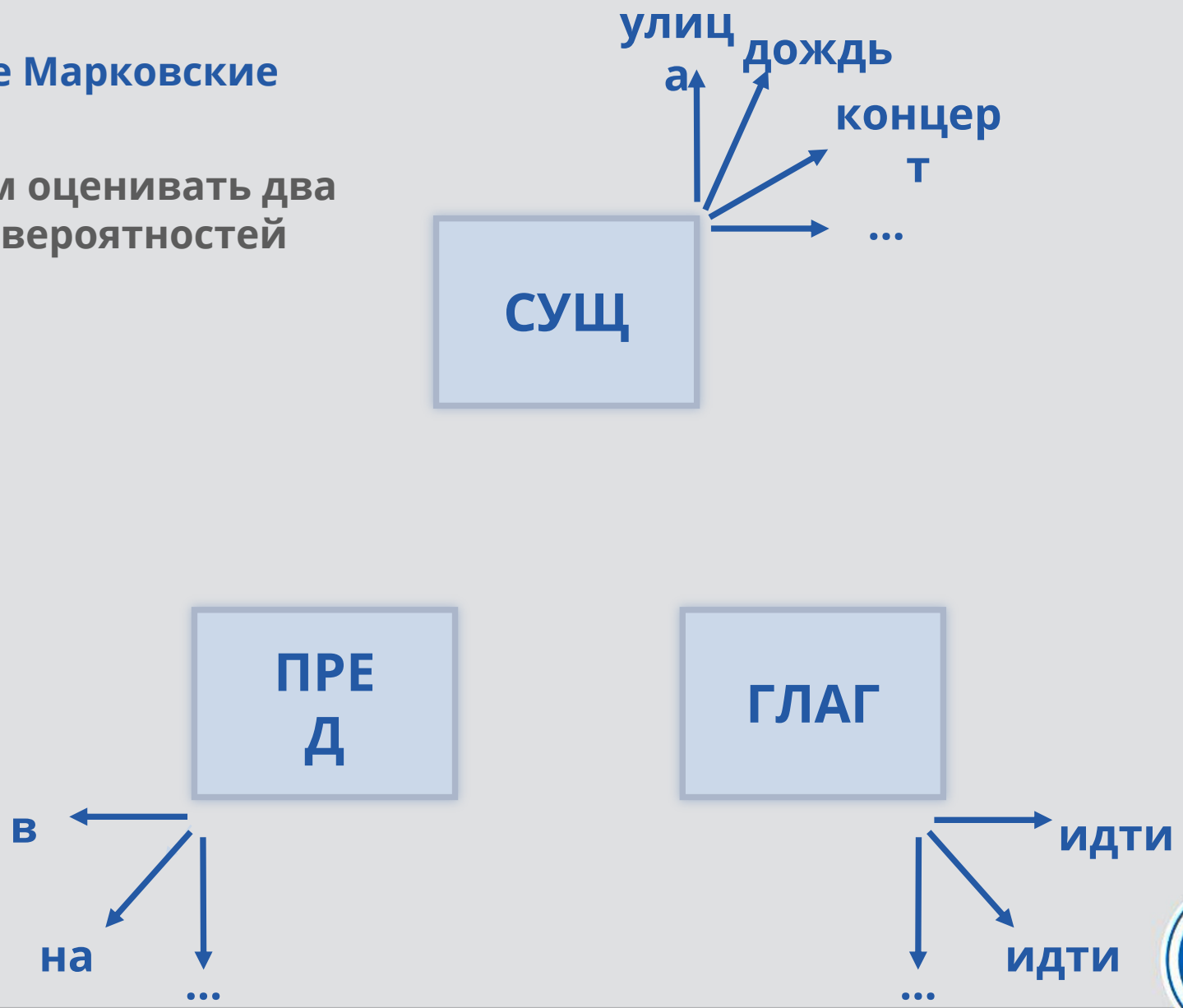
- Скрытые состояния порождают **наблюдения** – слова



Скрытые марковские модели

Скрытые Марковские модели:

- Будем оценивать два вида вероятностей



Скрытые марковские модели

Скрытые Марковские модели:

- Будем оценивать два вида вероятностей:
  - Вероятность того, что скрытое состояние порождает слово

Скрытые марковские модели

Скрытые Марковские модели:

- **Будем оценивать два вида вероятностей:**
  - **Вероятность перехода между скрытыми состояниями**

Обозначения

- $s_j \in V_s$ - скрытые состояния, части речи

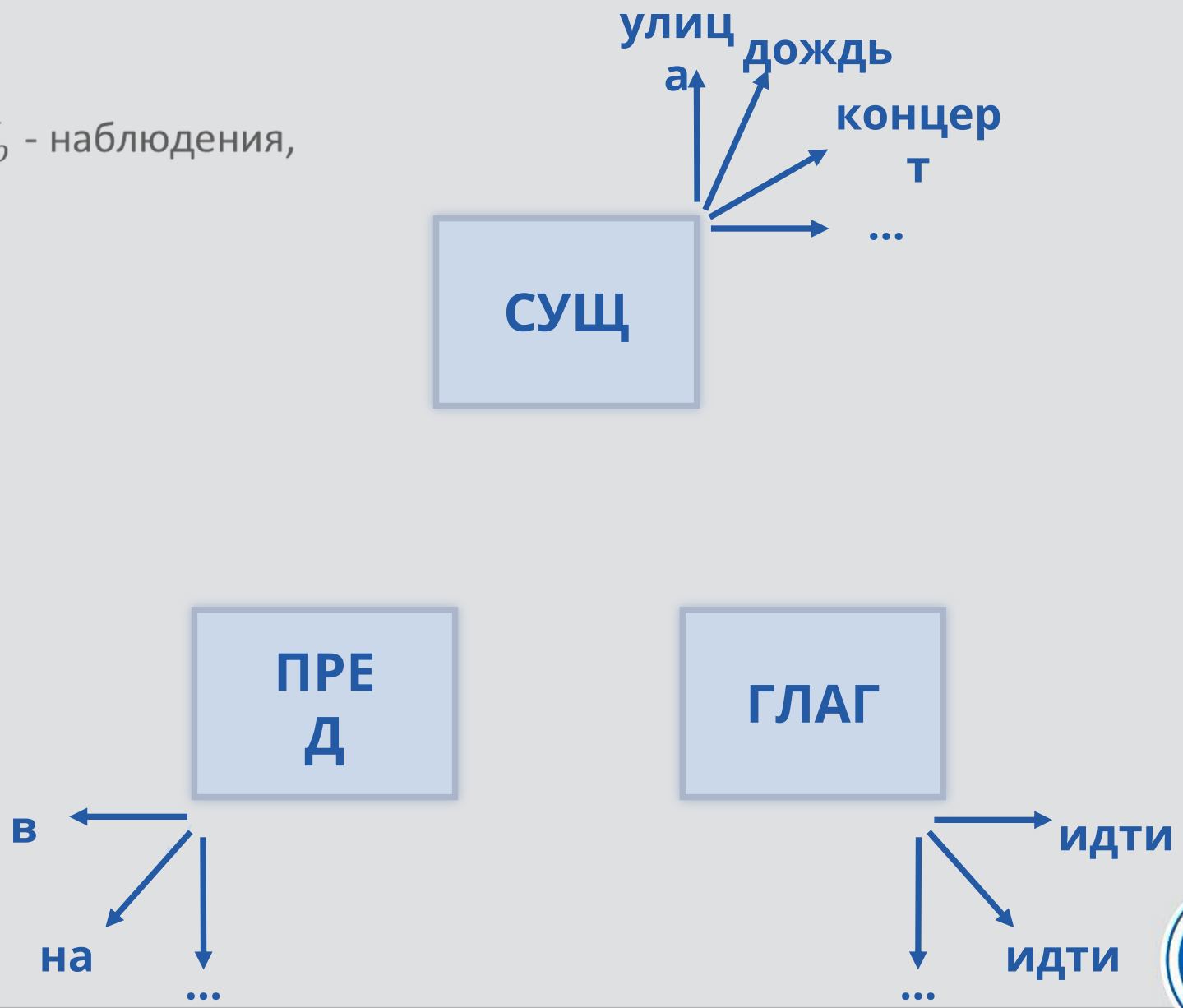
СУЩ

ПРЕД

ГЛАГ

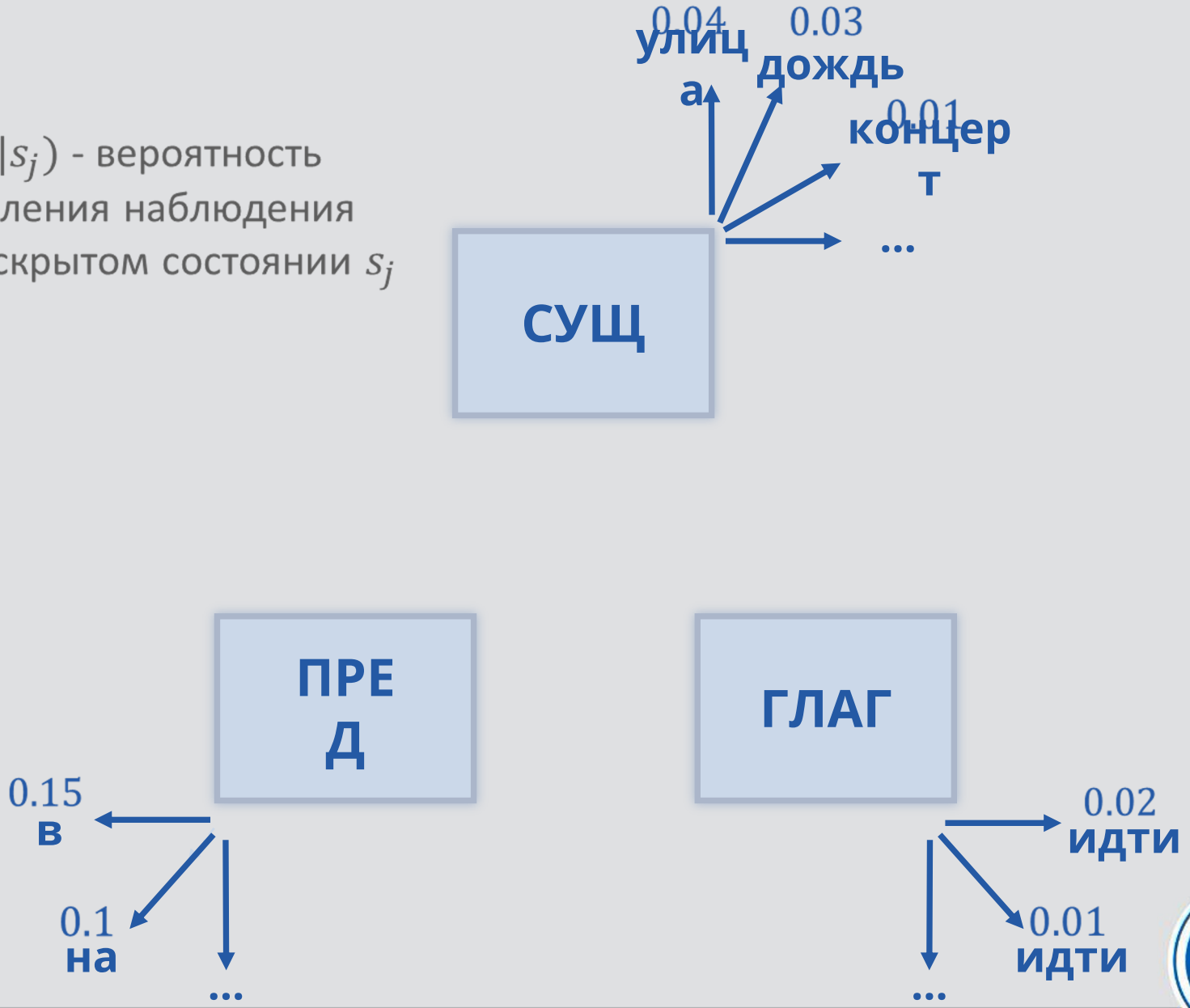
Обозначения

- $o_i \in V_o$ - наблюдения, слова



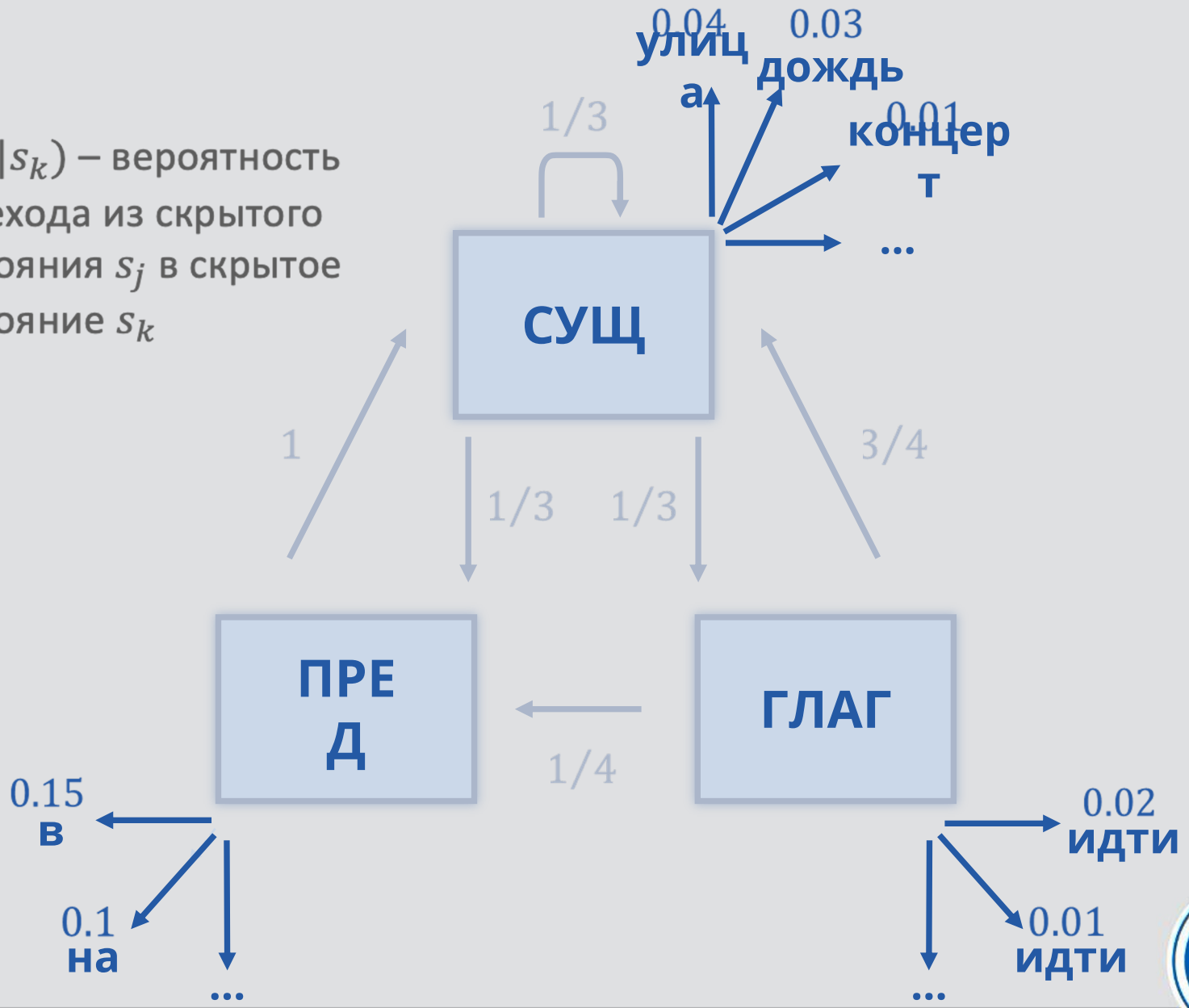
Обозначения

- $P(o_i|s_j)$ - вероятность появления наблюдения o_i в скрытом состоянии s_j



Обозначения

- $P(s_j|s_k)$ – вероятность перехода из скрытого состояния s_j в скрытое состояние s_k



Свойства вероятностей в скрытой марковской модели

- $P(o_i|s_j)$ – вероятность слова зависит только от текущего скрытого состояния
- $\sum_k P(s_j|s_k) = 1$ – из скрытого состояния s_j можно попасть в другие скрытые состояния
- π_j – начальное распределение вероятностей

Три задачи скрытых марковских моделей

Дана последовательность наблюдений:

$O = o_1, o_2, \ldots, o_T$

и построенная модель λ

Три задачи скрытых марковских моделей

Дана последовательность наблюдений:

$O = o_1, o_2, \ldots, o_T$

и построенная модель λ

Задача 1. Оценить вероятность порождения последовательности O моделью HMM

Алгоритм прямого обратного хода

Три задачи скрытых марковских моделей

Дана последовательность наблюдений:

$O = o_1, o_2, \ldots, o_T$

и построенная модель λ

Задача 2. Декодировать последовательность наблюдений: оценить, какая последовательность скрытых состояний могла породить наблюдения?

Алгоритм Витерби

Три задачи скрытых марковских моделей

Дана последовательность наблюдений:

$O = o_1, o_2, \ldots, o_T$

и построенная модель λ

Задача 3. Оценить параметры скрытой модели Маркова λ

Алгоритм Баума-Велша

Динамические алгоритмы

Жадные алгоритмы, которые на каждом шаге выбирают локально оптимальное решение с учетом предыстории

- **Алгоритм прямого обратного хода:** насколько вероятно текущее состояние
- **Алгоритм Витерби:** наиболее вероятное скрытое состояние на текущем шаге
- **Алгоритм Баума-Велша:** лучшая оценка при данных наблюдениях

Скрытые марковские модели

- Простой статистический метод разметки последовательности, в том числе, морфологического анализа
- Использует исключительно частоты совстречаемости слов и их частеречные теги
- Помогает разрешать морфологическую омонимию

Нейросетевые языковые модели

Первое поколение языковых моделей

- Счетные языковые модели имеют множество недостатков:
  - Проблема нулевых значений
  - Тяжело учитывать длинные контексты
  - Много параметров
  - Высокая перплексия
  - Генерируется несогласованный и неинтересный текст

Второе поколение языковых моделей

Нейросетевые языковые модели решают задачу языкового моделирования: по предыдущим словам (контексту) предсказать следующее слово

Нейросетевая языковая модель

- Модель n-грамм: по $n-1$ слову требуется предсказать n-ное слово
- Вход: $w_1, w_2, \ldots, w_{n-1}$
- Выход: w_n

Нейросетевая языковая модель

- Модель n-грамм: по $n-1$ слову требуется предсказать n-ное слово
- Вход: $w_1, w_2, \ldots, w_{n-1}$
- Выход: w_n

«все студенты открыли свои _____»

Вход: «все студенты открыли свои»

Выход: «_____»

Нейросетевая языковая модель

- Модель n-грамм: по $n-1$ слову требуется предсказать n-ное слово
- Вход: $w_1, w_2, \ldots, w_{n-1}$
- Выход: w_n

«все студенты открыли свои ноутбуки»

Вход: «все студенты открыли свои»

Выход: «ноутбуки»

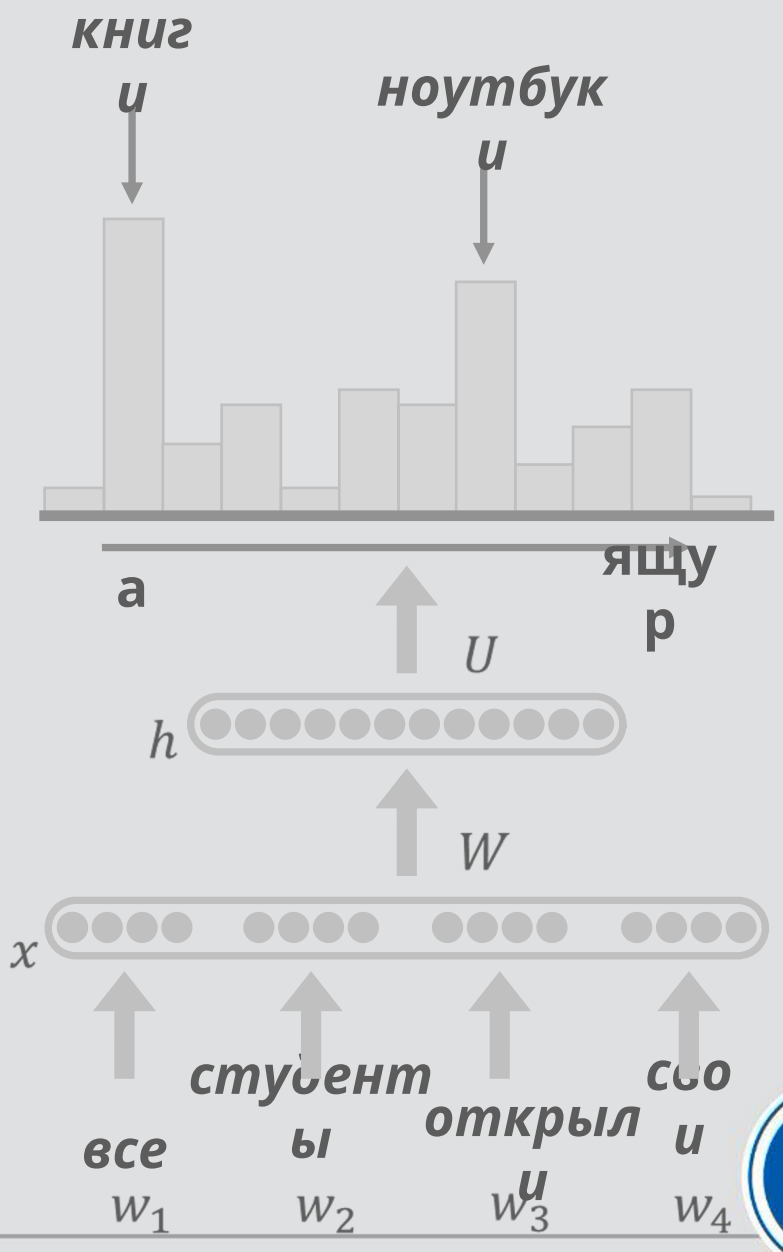
Нейросетевая языковая модель

- Модель n-грамм: по $n-1$ слову требуется предсказать n-ное слово
- Вход: $w_1, w_2, \ldots, w_{n-1}$
- Выход: w_n

«все студенты открыли свои **ноутбуки**»

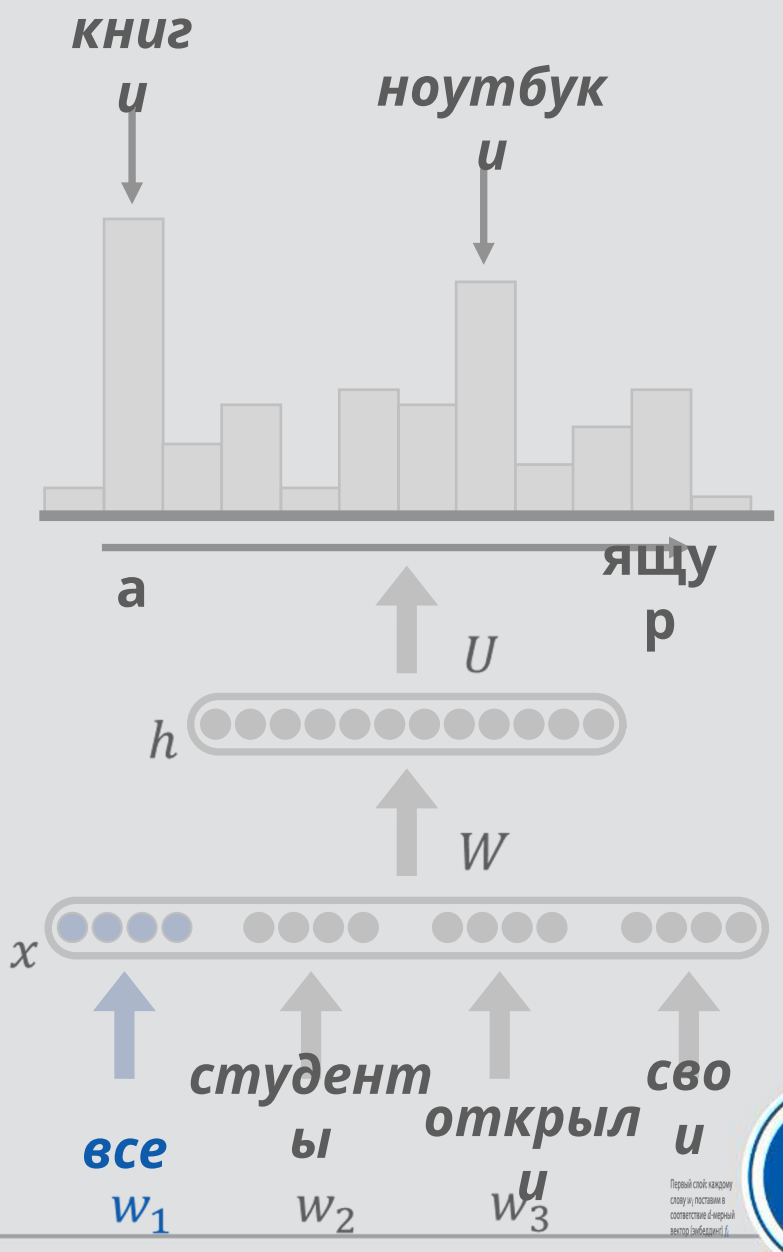
Вход: «все студенты открыли свои»

Выход: «ноутбуки»



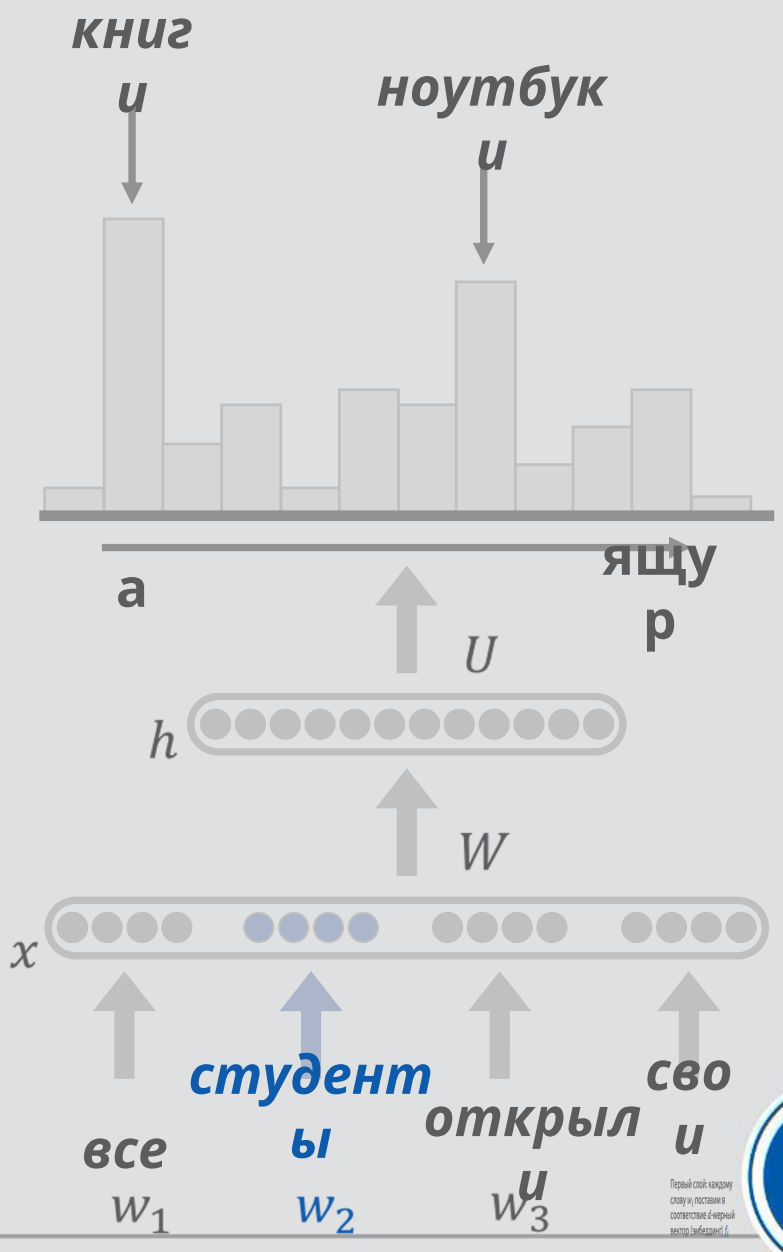
Нейросетевая языковая модель

- Первый слой: каждому слову w_i поставим в соответствие d-мерный вектор (эмбеддинг) f_i



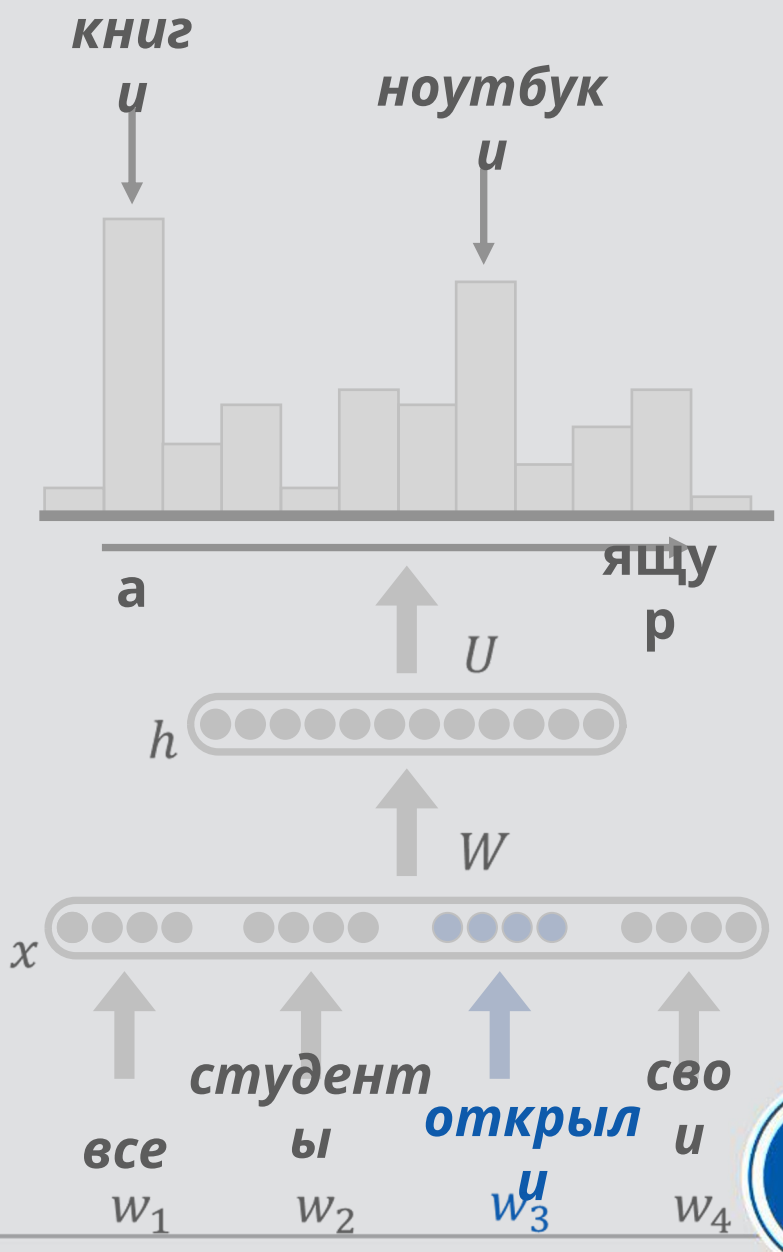
Нейросетевая языковая модель

- Первый слой: каждому слову w_i поставим в соответствие d-мерный вектор (эмбеддинг) f_i



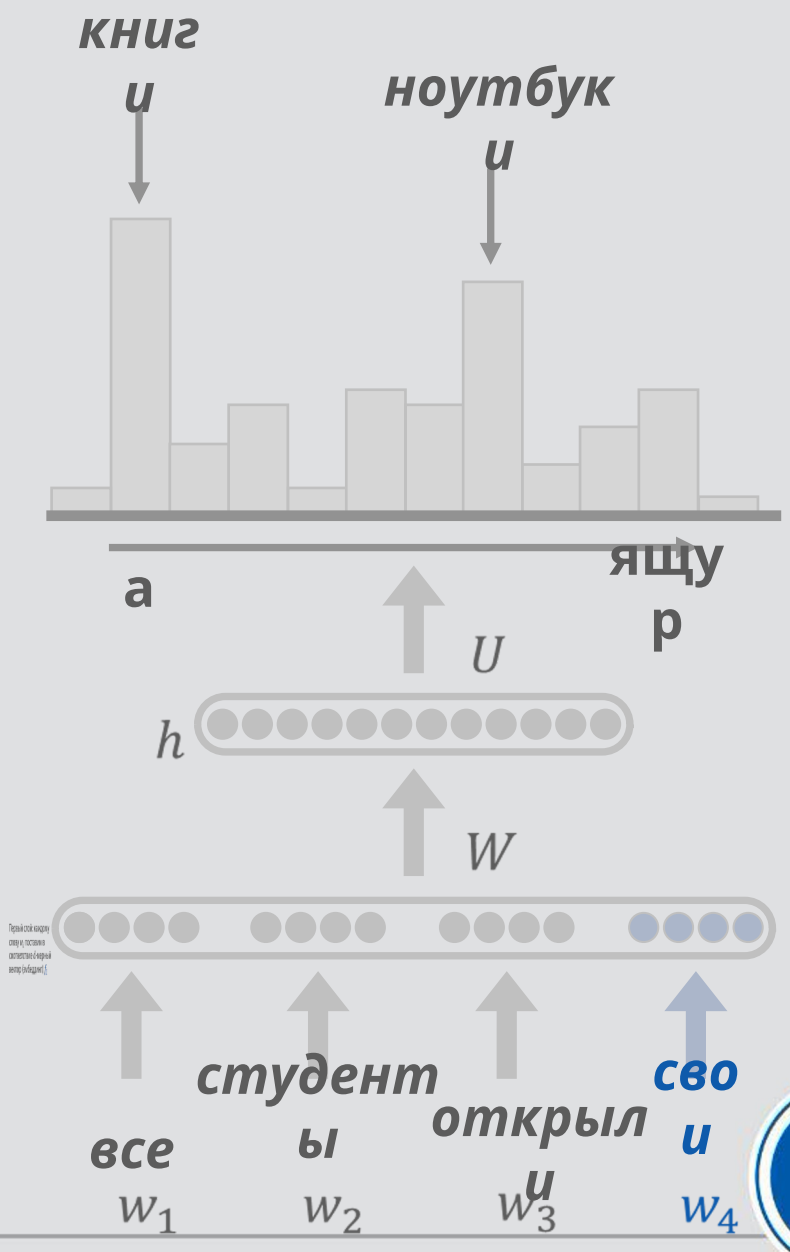
Нейросетевая языковая модель

- Первый слой: каждому слову w_i поставим в соответствие d-мерный вектор (эмбеддинг) f_i



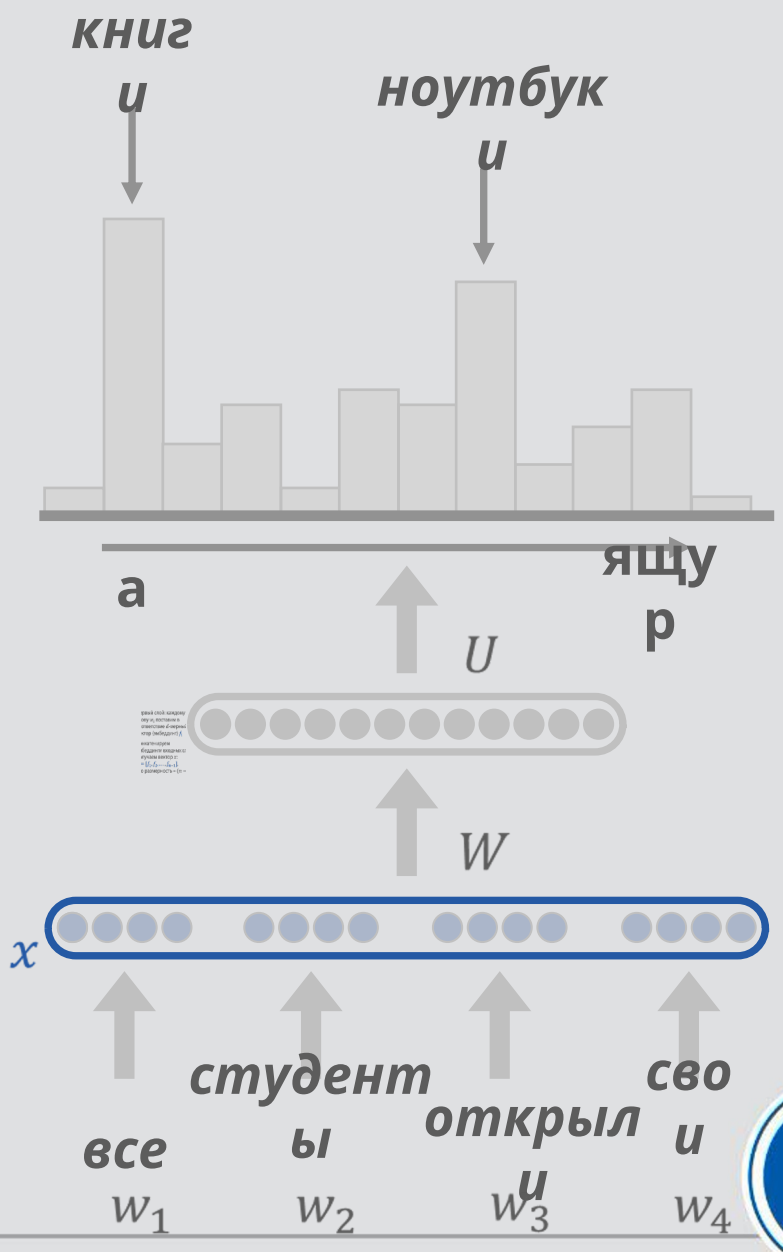
Нейросетевая языковая модель

- Первый слой: каждому слову w_i поставим в соответствие d-мерный вектор (эмбеддинг) f_i



Нейросетевая языковая модель

- Первый слой: каждому слову w_i поставим в соответствие d-мерный вектор (эмбеддинг) f_i
- Конкатенируем эмбеддинги входных слов, получаем вектор x:
  $x = [f_1, f_2, \ldots, f_{n-1}]$,
  его размерность – $(n-1)d$

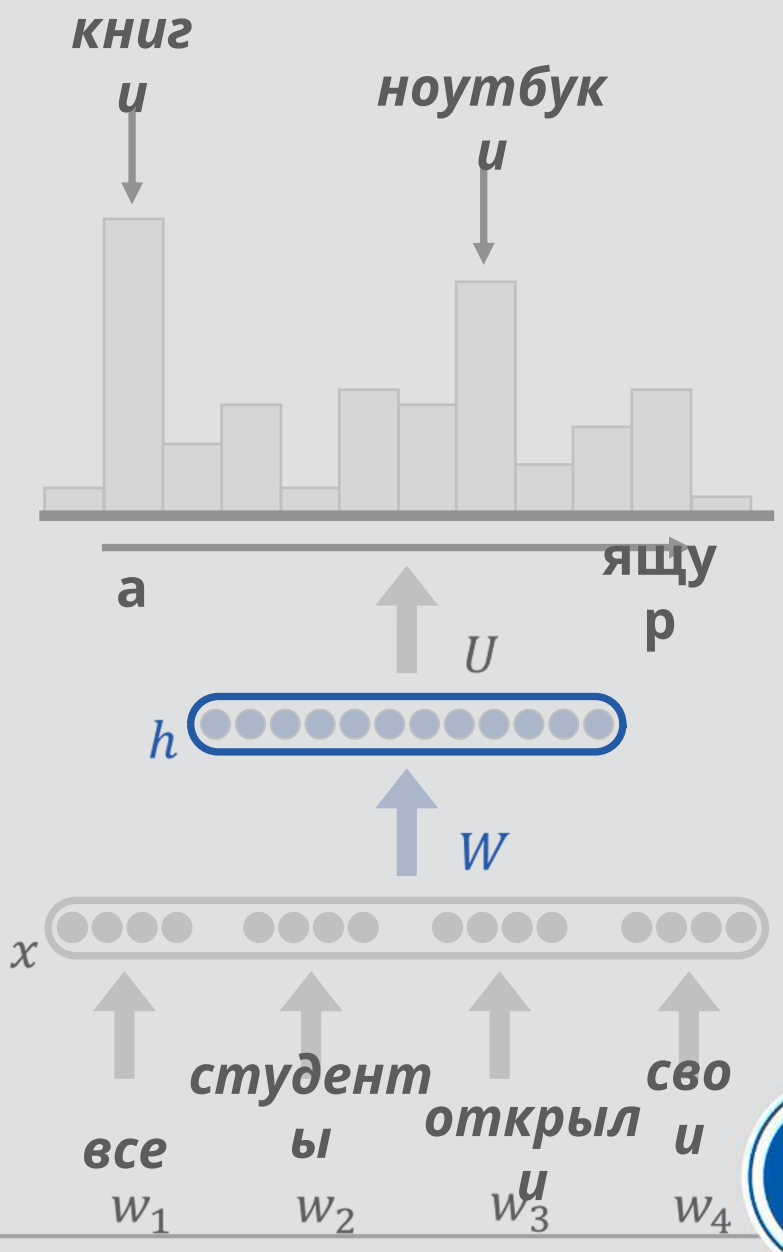


Нейросетевая языковая модель

- Скрытый слой:

$$h = \sigma(Wx + b)$$

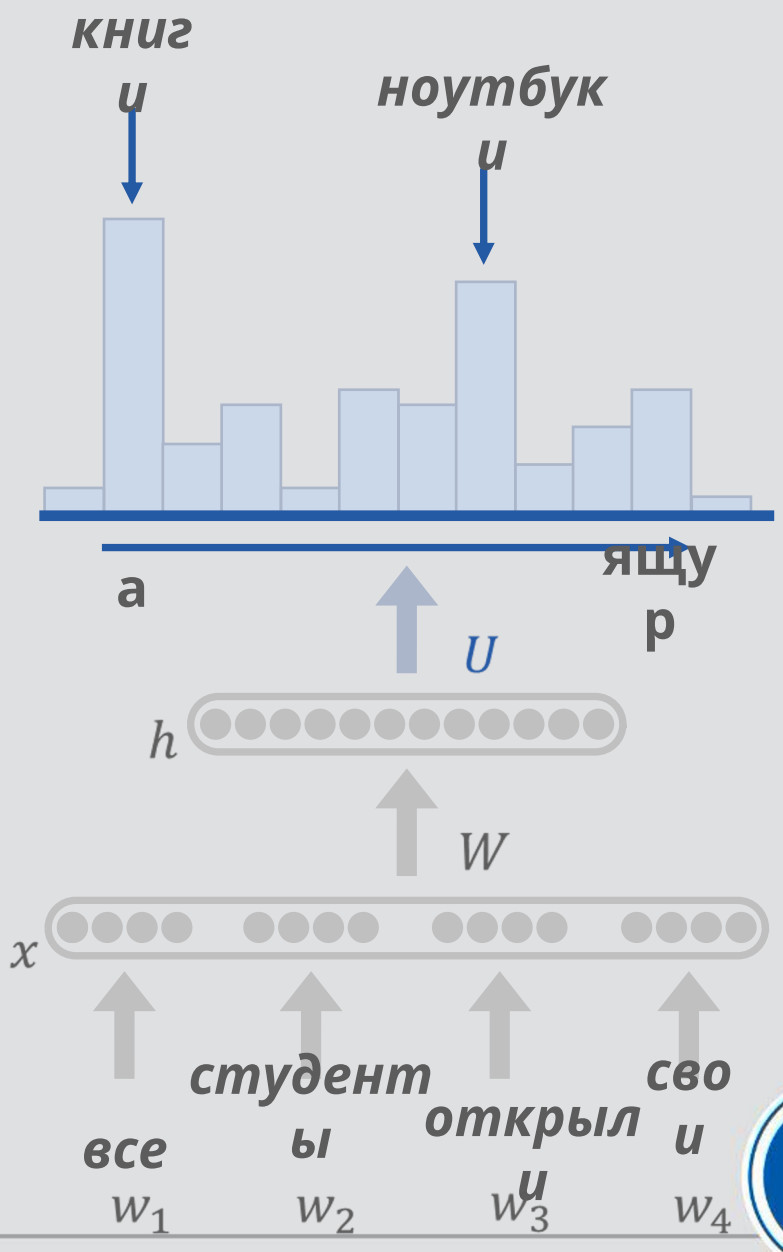
- W, b_1 – матрица весов и вектор сдвигов – обучаемые параметры



Нейросетевая языковая модель

- Получим вероятности следующего слова из y:

$$\hat{y} = \mathrm{softmax}(Uh + b_2)$$

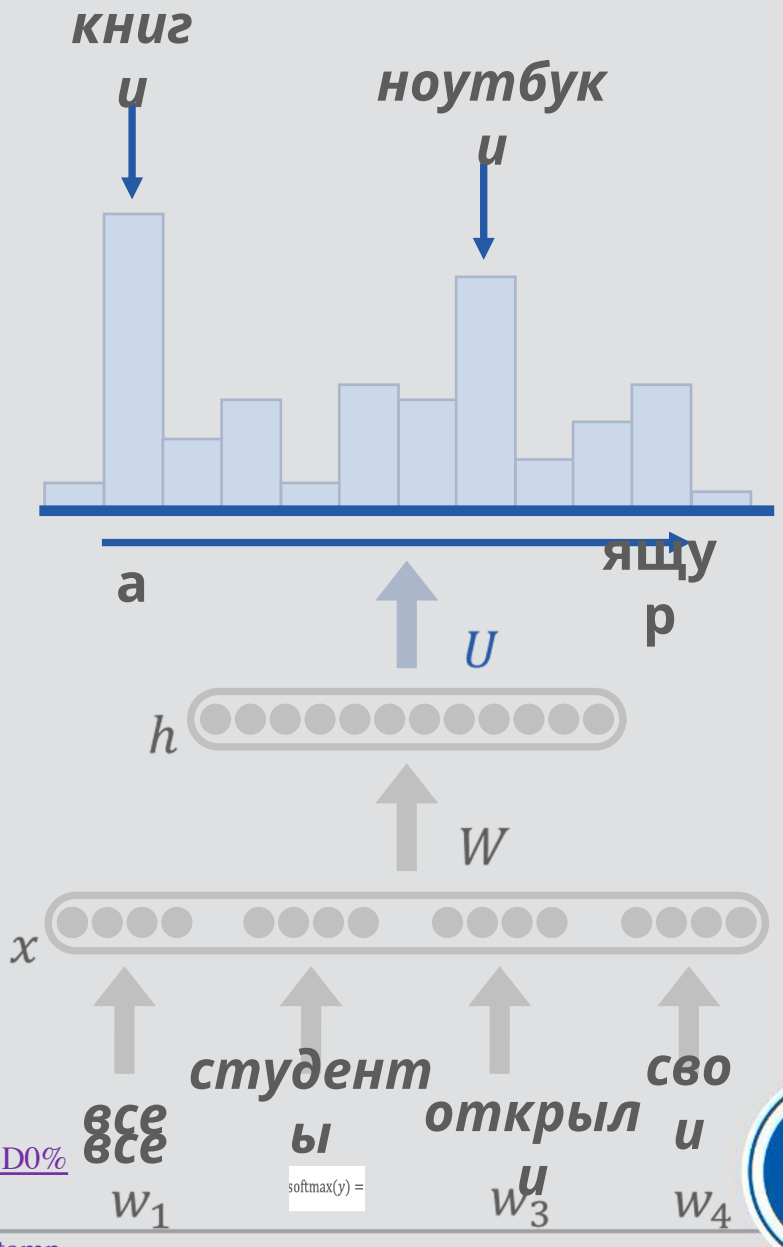


Нейросетевая языковая модель

- Получим вероятности следующего слова из y:

$$\hat{y} = \mathrm{softmax}(Uh + b_2)$$

$$\mathrm{softmax}(y) = \frac{e^{y_k}}{\sum_i e^{y_i}}$$



https://www.istockphoto.com/ru/%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F/reminder-blue-square-grunge-stamp-

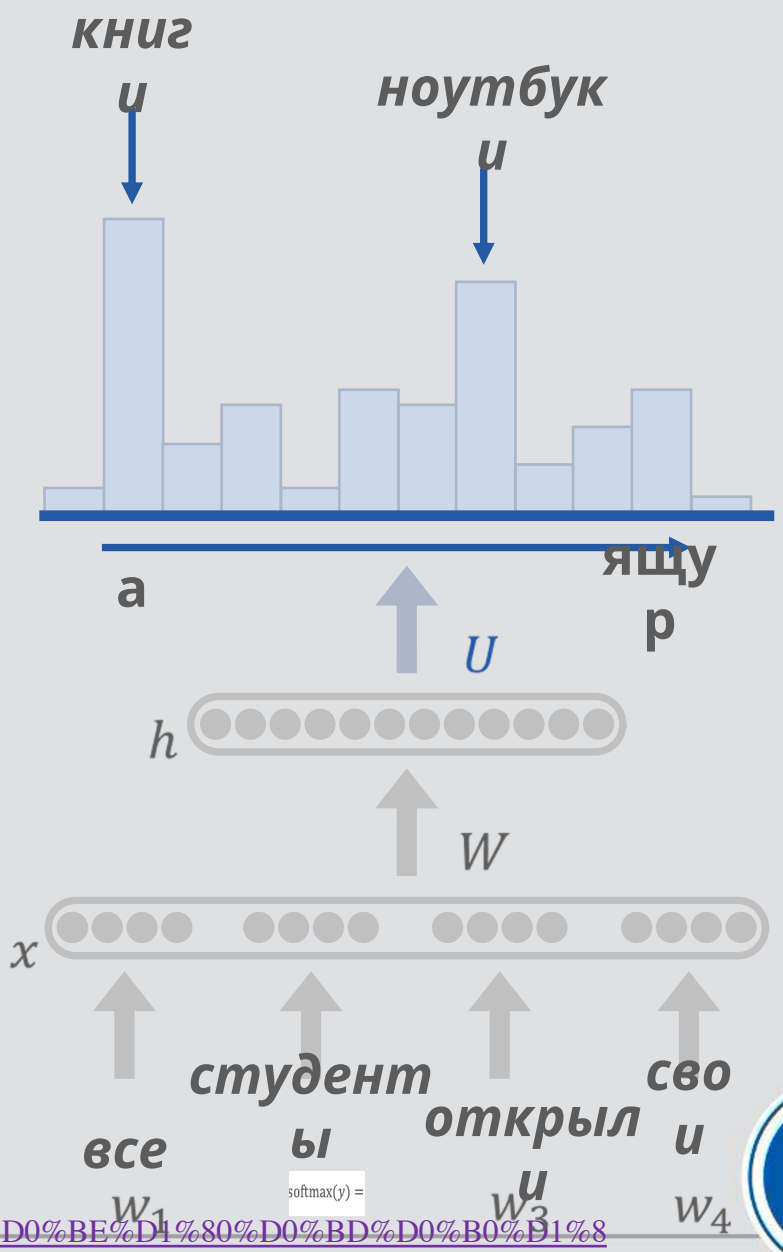
Нейросетевая языковая модель

- Получим вероятности следующего слова из y:

$$\hat{y} = \mathrm{softmax}(Uh + b_2)$$

$$\mathrm{softmax}(y) = \frac{e^{y_k}}{\sum_i e^{y_i}}$$

- $\hat{y} \in \mathbb{R}^{|V|}$
- U, b_1 – матрица весов и вектор сдвигов – обучаемые параметры

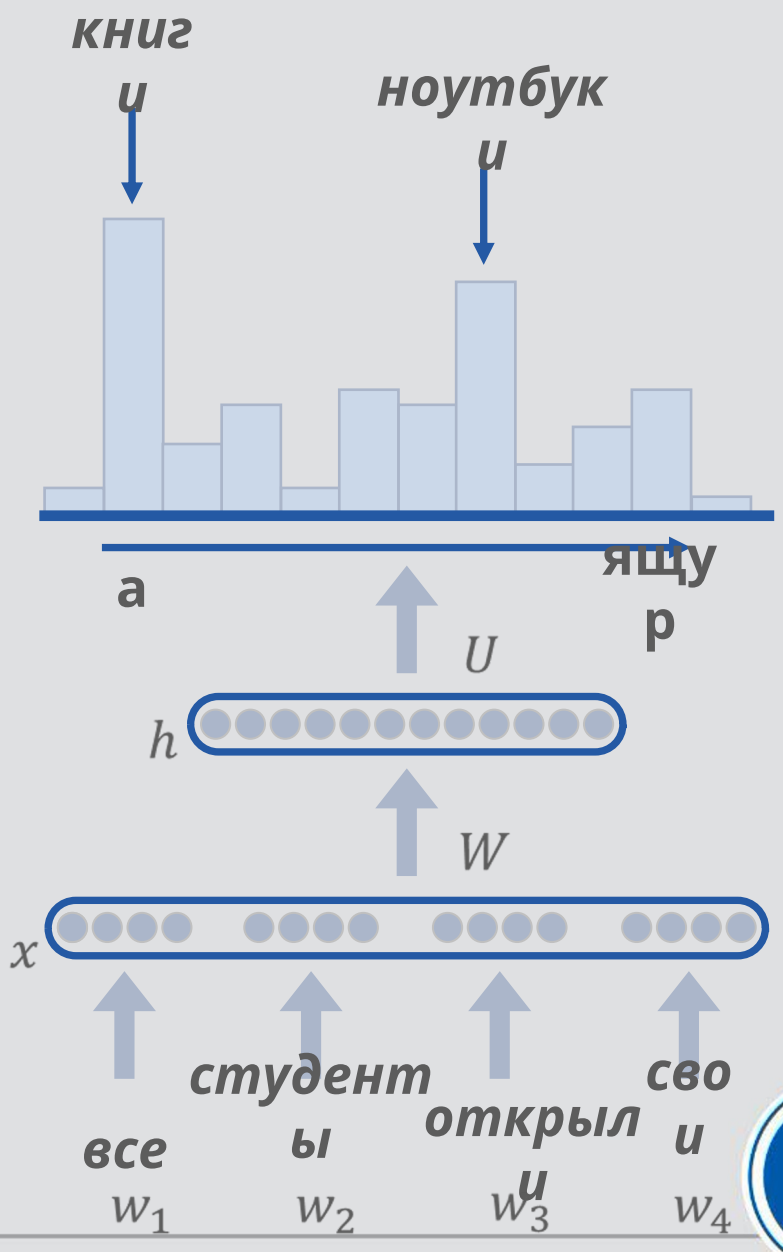


https://www.istockphoto.com/ru/%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F/reminder-blue-square-grunge-stamp-on-white-gm843610854-137961575

Нейросетевая языковая модель

Обучение:

- разбиваем корпус на n-граммы, число n-грамм — T
- $n-1$ слов подаем на вход сети, предсказываем n-ное слово



Нейросетевая языковая модель

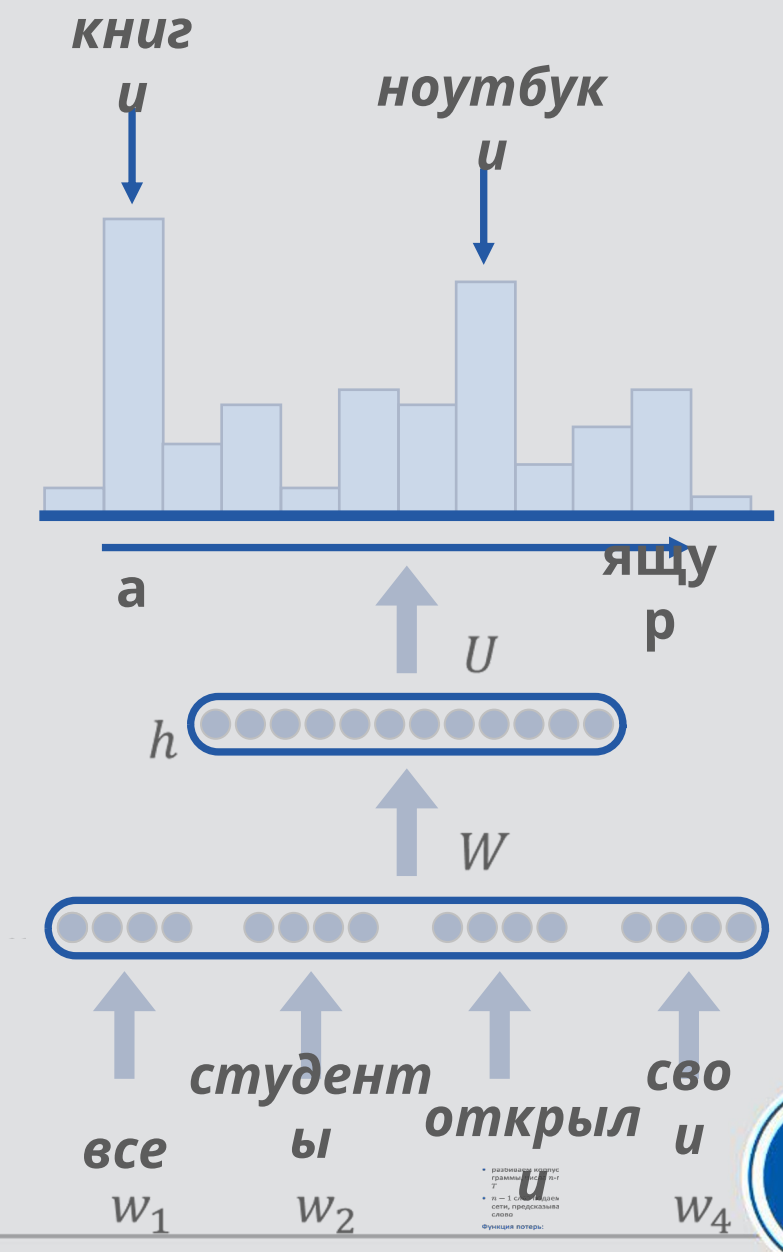
Обучение:

- разбиваем корпус на n-граммы, число n-грамм — T
- $n-1$ слов подаем на вход сети, предсказываем n-ное слово

Функция потерь:

- отрицательный логарифм правдоподобия

$$L = -\frac{1}{T} y \log(P(\hat{y}))$$



Языковые модели

Счетные языковые модели

Нейросетевые языковые модели

Языковые модели

Счетные языковые модели

- Много параметров
- Высокая перплексия
- Быстрые вычисления
- Фиксированная длина контекста

Нейросетевые языковые модели

Языковые модели

Счетные языковые модели

- Много параметров
- Высокая перплексия
- Быстрые вычисления
- Фиксированная длина контекста

Нейросетевые языковые модели

- Нет необходимости хранить все n-граммы
- Медленные вычисления
- Фиксированная длина контекста

Языковые модели

Счетные языковые модели

- Много параметров
- Высокая перплексия
- Быстрые вычисления
- Фиксированная длина контекста

Нейросетевые языковые модели

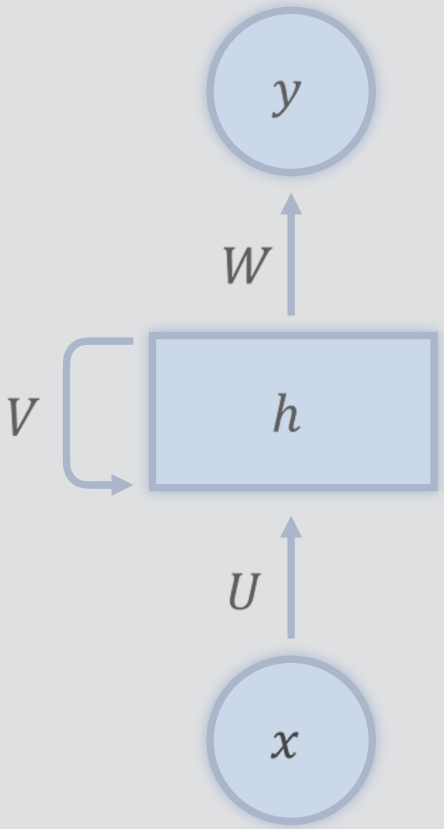
- Нет необходимости хранить все n-граммы
- Медленные вычисления
- Фиксированная длина контекста

Как сделать вычисления быстрее и избавиться от ограничений на длину контекста?

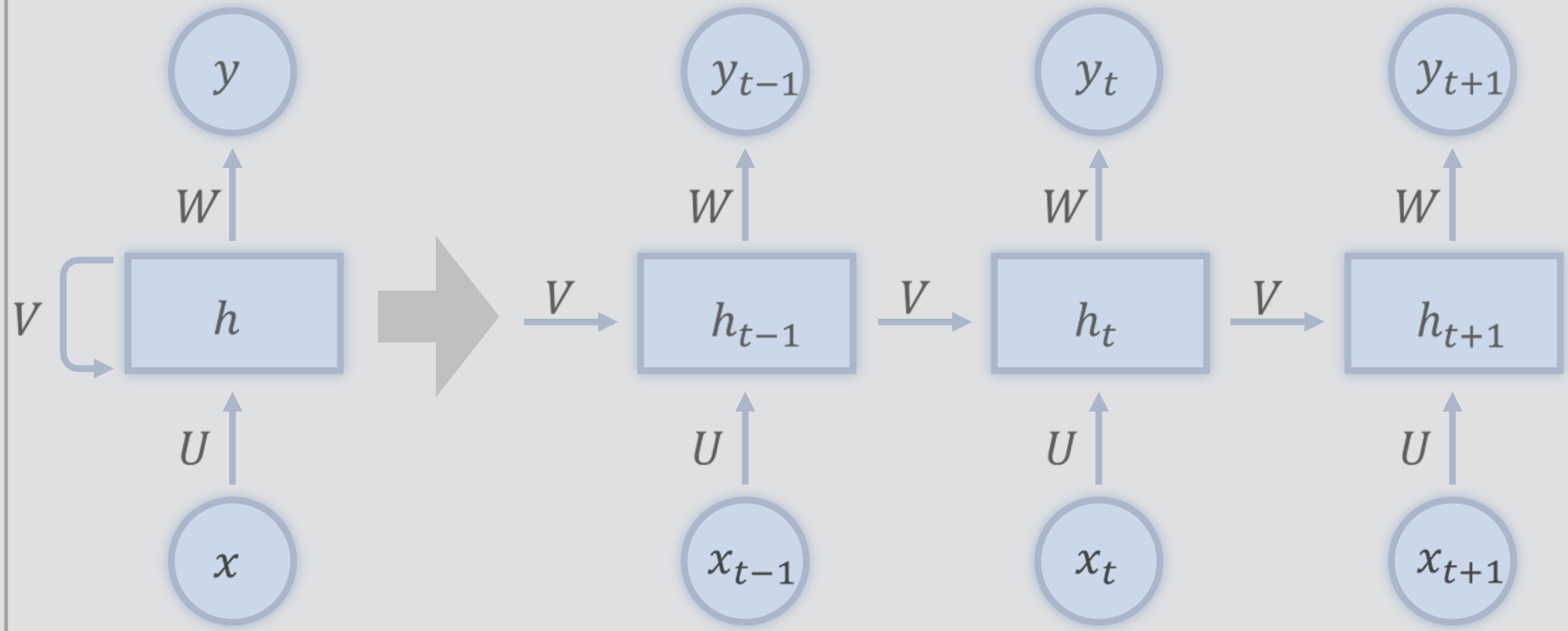
Рекуррентные нейронные сети

Рекуррентная нейронная сеть

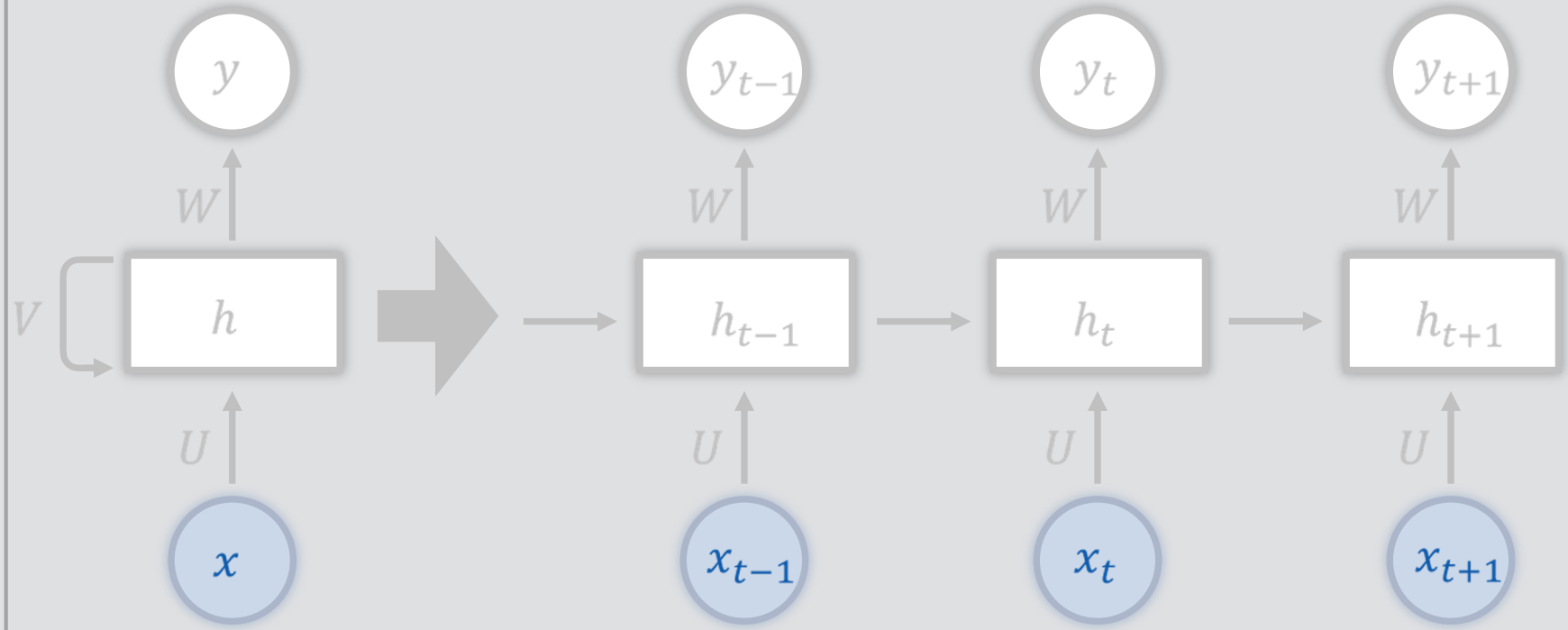
- Позволяет обрабатывать входные последовательности произвольной длины
- Имеет вектор состояния, в который сохраняет обработанные входы



Рекуррентная нейронная сеть

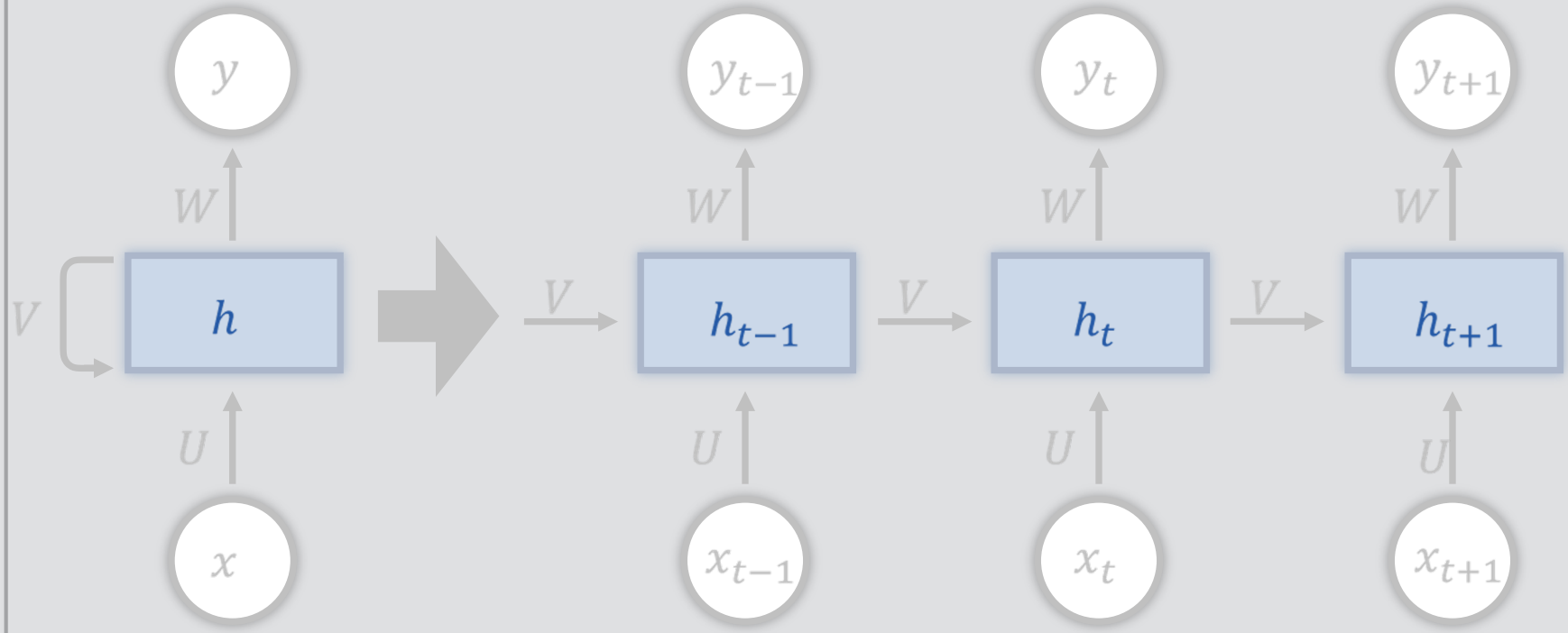


Рекуррентная нейронная сеть



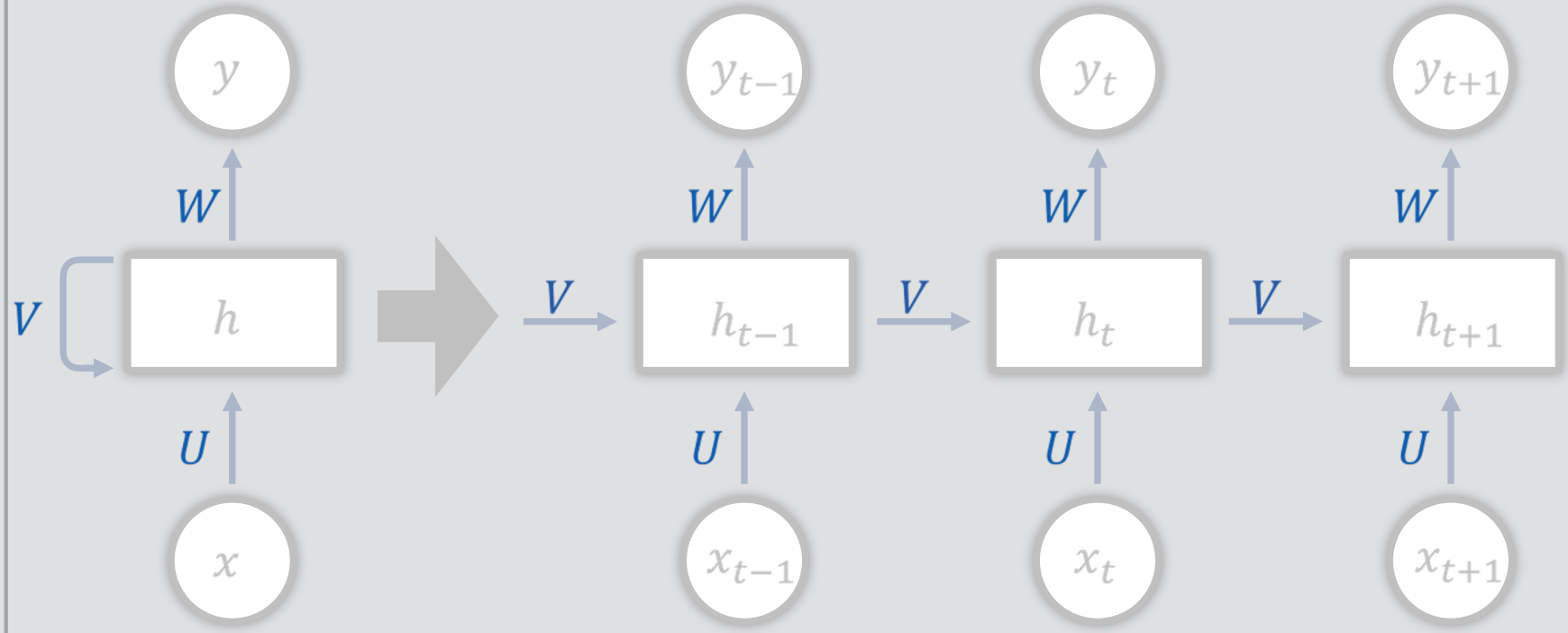
- x – входные слова (эмбеддинги)

Рекуррентная нейронная сеть



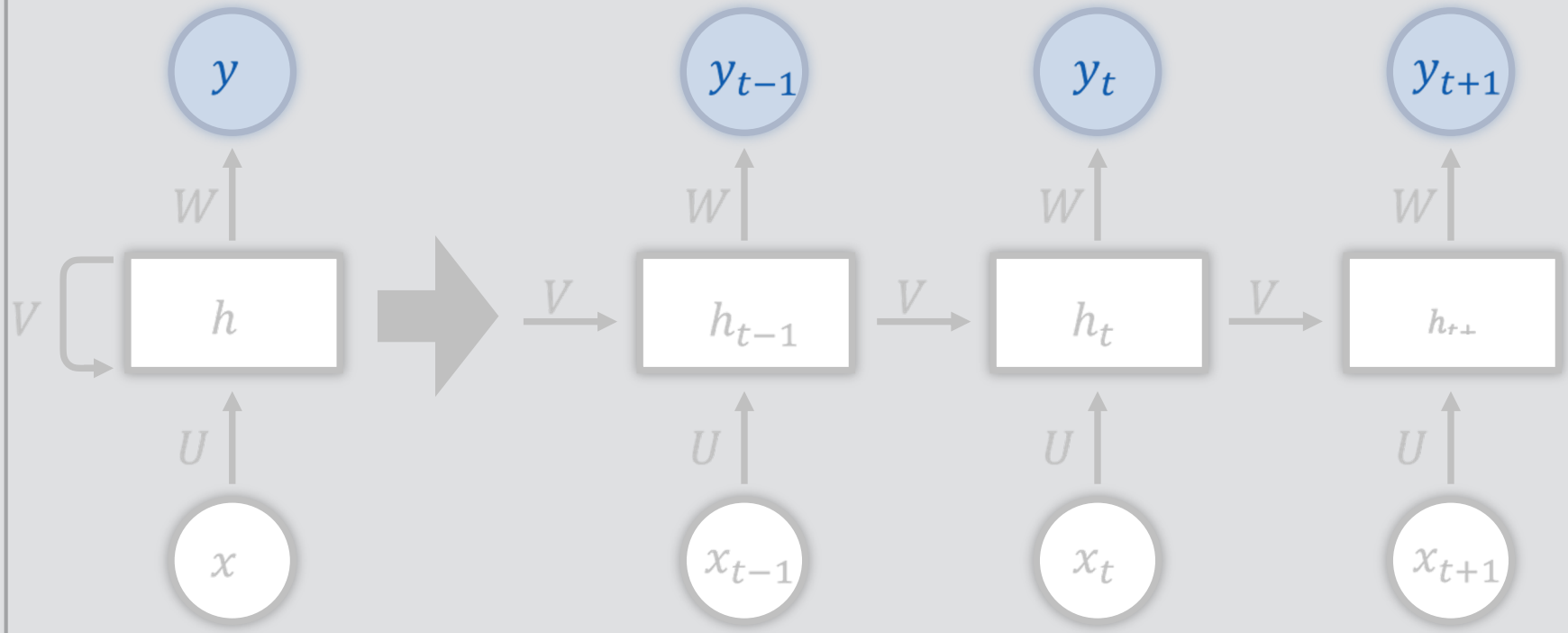
- x – входные слова (эмбеддинги)
- h – вектор памяти

Рекуррентная нейронная сеть



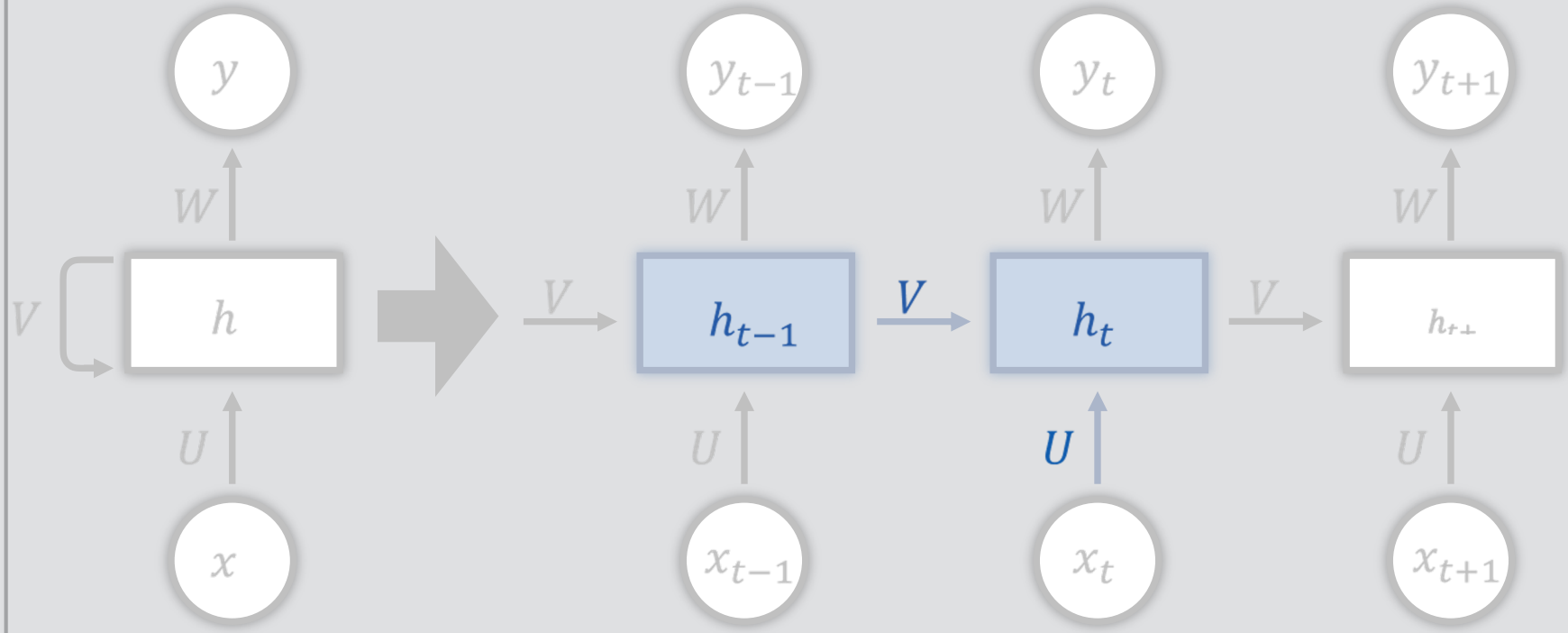
- x – входные слова (эмбеддинги)
- h – вектор памяти
- U, W, V – обучаемые матрицы весов

Рекуррентная нейронная сеть



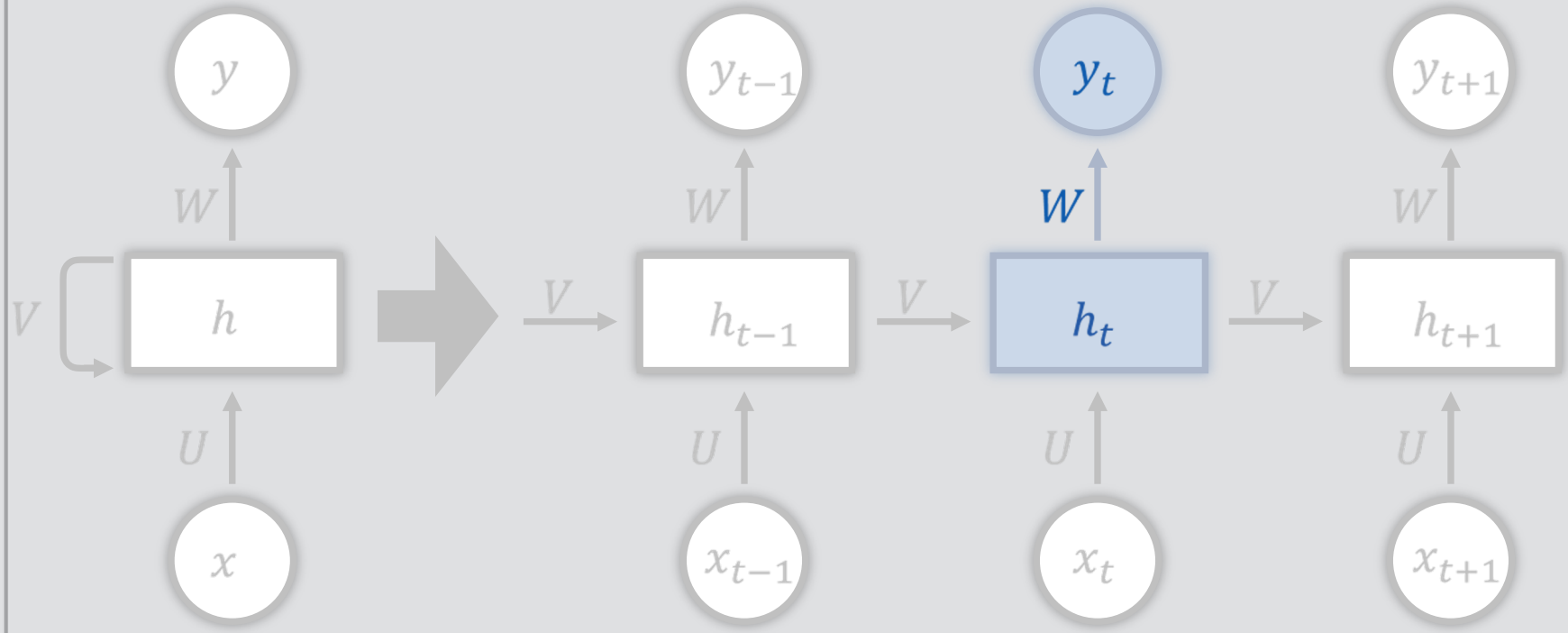
- x – входные слова (эмбеддинги)
- h – вектор памяти
- U, W, V – обучаемые матрицы весов
- y – выход

Рекуррентная нейронная сеть



- $h_t = \sigma(Vh_{t-1} + Ux_t + b_1)$ – обновление вектора памяти

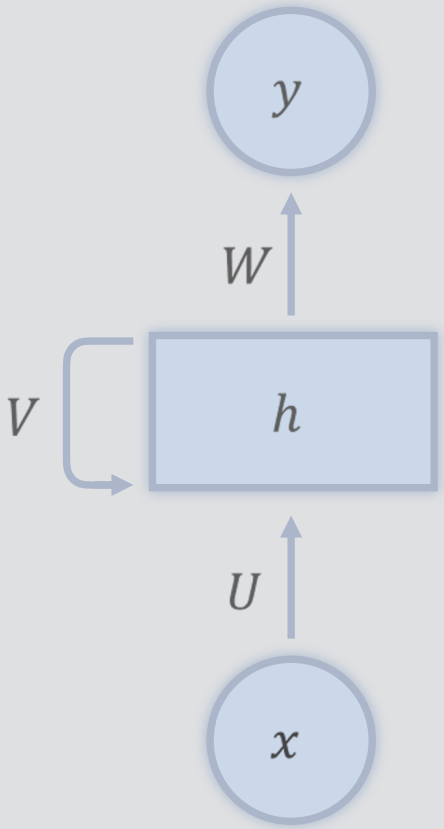
Рекуррентная нейронная сеть



- $h_t = \sigma(Vh_{t-1} + Ux_t + b_1)$ – обновление вектора памяти
- $\widehat{y_t} = softmax(Wh_t + b_2) \in \mathbb{R}^{|V|}$ – предсказание

Рекуррентная нейронная сеть

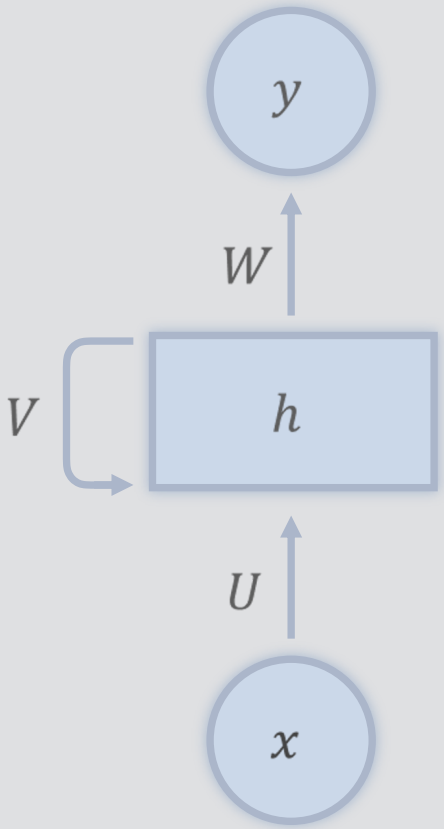
- Позволяет обрабатывать входные последовательности произвольной длины
- Имеет вектор состояния, в который сохраняет обработанные входы



Обучение рекуррентной нейронной сети

- Разбиваем корпус на слова:

$$w_1, w_2, \ldots, w_n$$



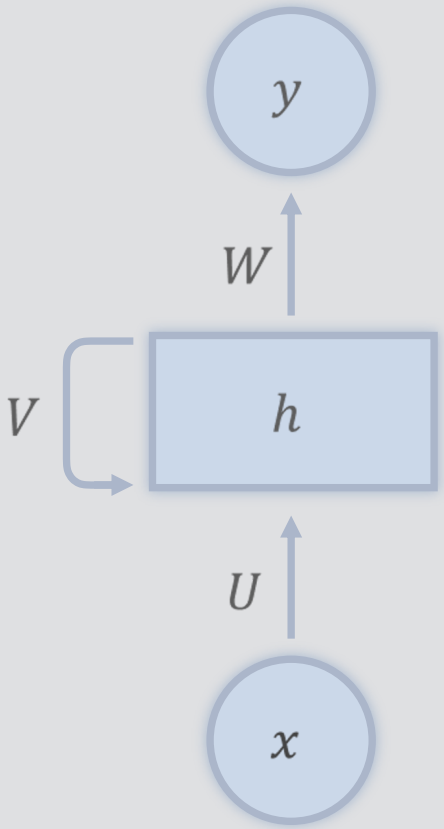
Обучение рекуррентной нейронной сети

- Разбиваем корпус на слова:

$$w_1, w_2, \ldots, w_n$$

- Эмбеддинги слов:

$$x_i = E w_i$$



Обучение рекуррентной нейронной сети

- Разбиваем корпус на слова:

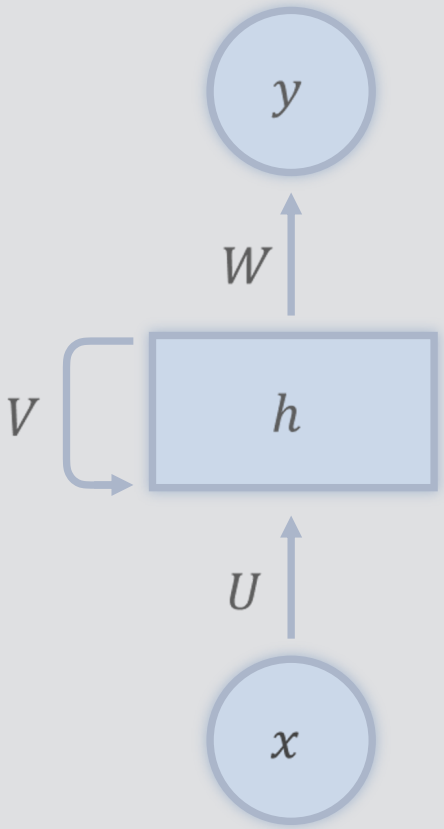
$$w_1, w_2, \ldots, w_n$$

- Эмбеддинги слов:

$$x_i = E w_i$$

- Сдвинем на одно слово направо предложение:

$$y_1 = w_2,\ y_{n-1} = w_n$$



Обучение рекуррентной нейронной сети

- Разбиваем корпус на слова:

$$w_1, w_2, \ldots, w_n$$

- Эмбеддинги слов:

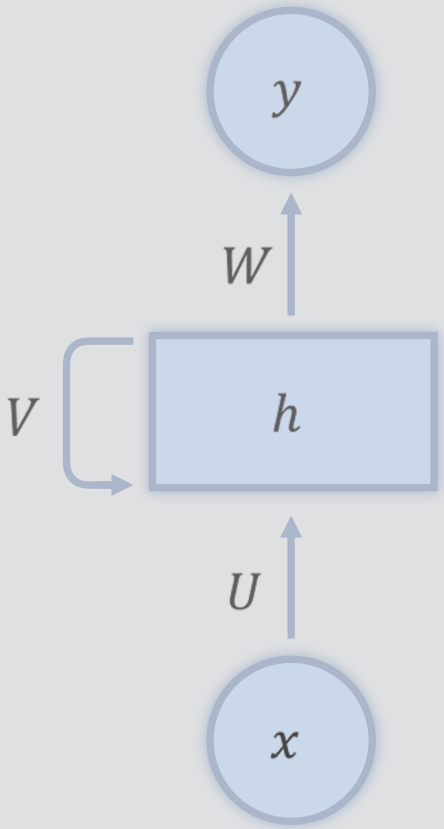
$$x_i = E w_i$$

- Сдвинем на одно слово направо предложение:

$$y_1 = w_2,\ y_{n-1} = w_n$$

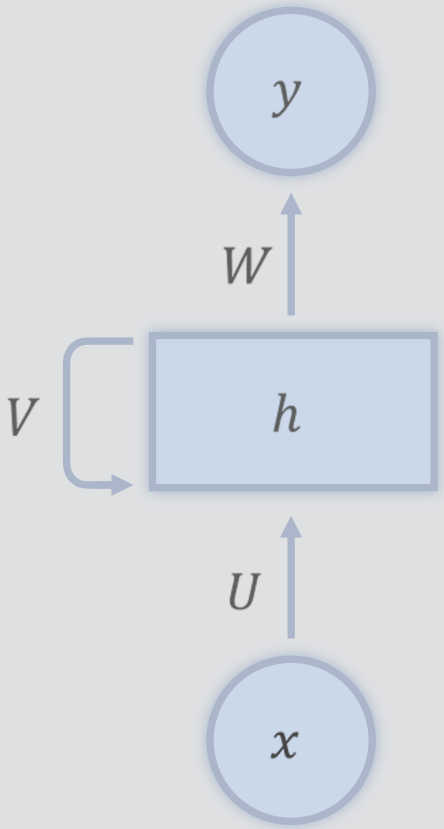
- Предсказываем:

$$(x_i, y_i)$$



Обучение рекуррентной нейронной сети

- **Вспомогательные символы начала и конца предложения: <s>, </s>**

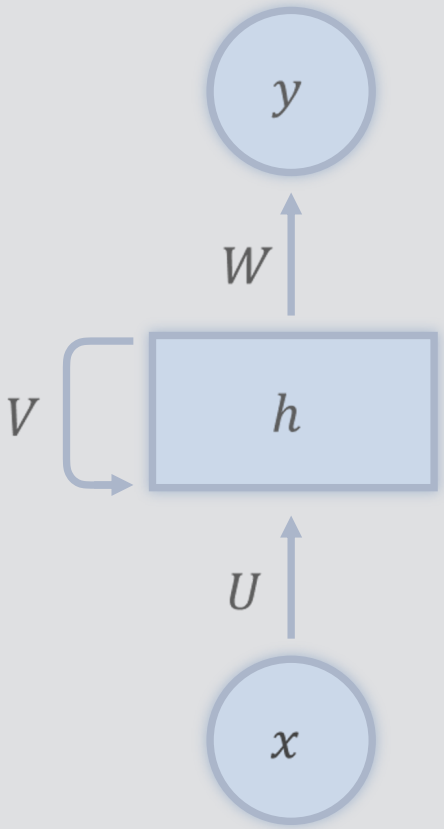


Обучение рекуррентной нейронной сети

- **Вспомогательные символы начала и конца предложения: <s>, </s>**

W = «на улице идет дождь»

- (<s> → на)
- (на → улице)
- (улице → идет)
- (идет → дождь)
- (дождь → </s>)

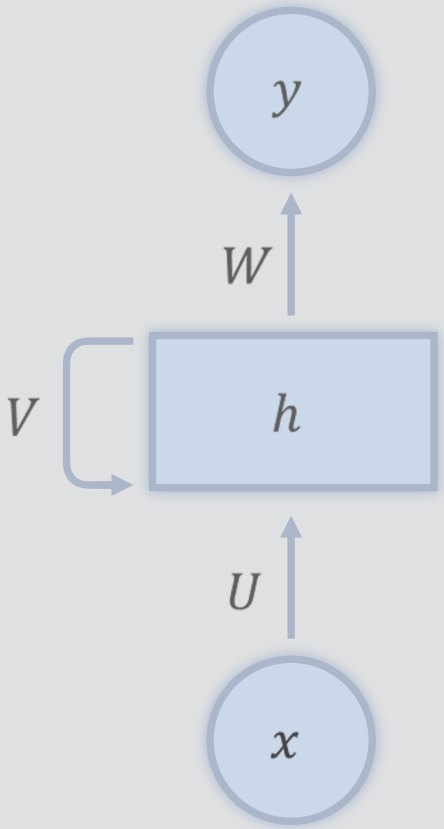


Обучение рекуррентной нейронной сети

- На каждом шаге считаем функцию потерь, например, кросс-энтропию:

$$\mathrm{loss}_{\mathrm{local}} = CE(y_t, \widehat{y_t})$$

$$= -\sum_{w \in V} y_t^w \log \widehat{y_t^w}$$

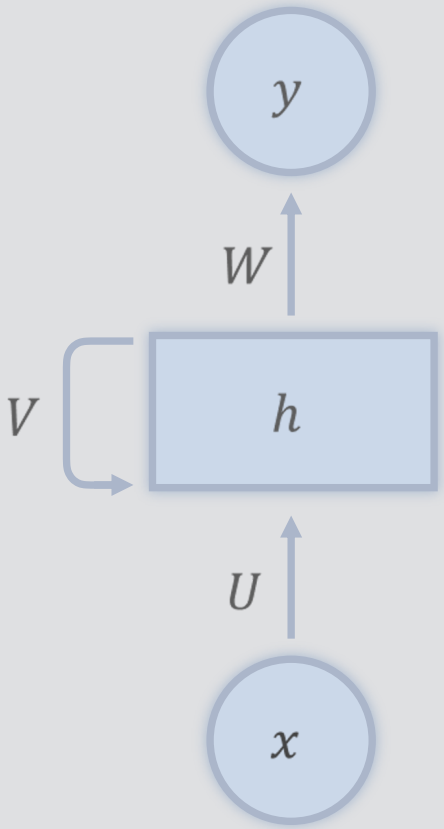


Обучение рекуррентной нейронной сети

- На каждом шаге считаем функцию потерь, например, кросс-энтропию:

$$\text{loss}_{\text{local}} = CE(y_t, \widehat{y_t})$$

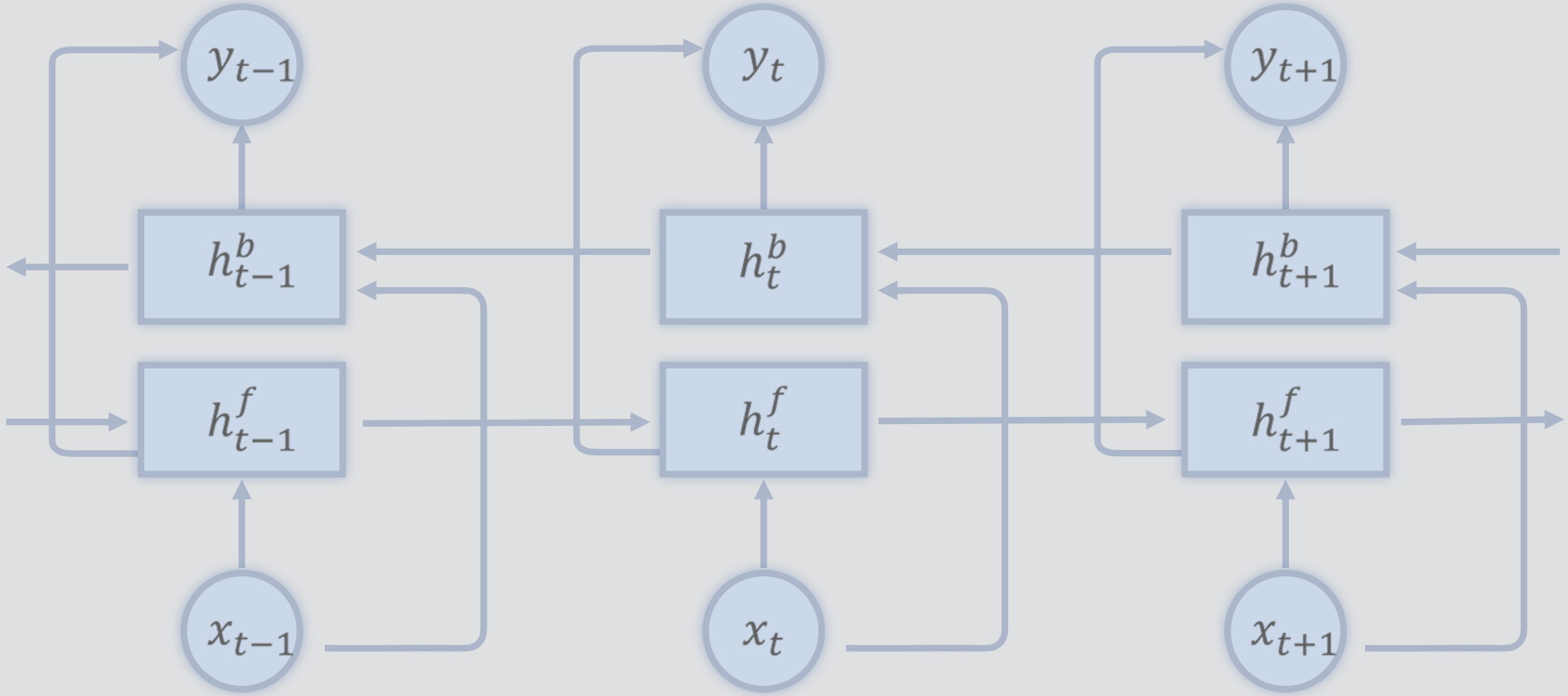
$$= -\sum_{w \in V} y_t^w \log \widehat{y_t^w}$$



- Общая функция потерь:

$$\text{loss}_{\text{global}} = \sum \text{loss}_{\text{local}}$$

Двунаправленная рекуррентная нейронная сети



Рекуррентная нейронная сеть

Сильные стороны

- Обрабатывают входы переменной длины
- Имеют механизм памяти
- Количество параметров не зависит от размера корпуса

Рекуррентная нейронная сеть

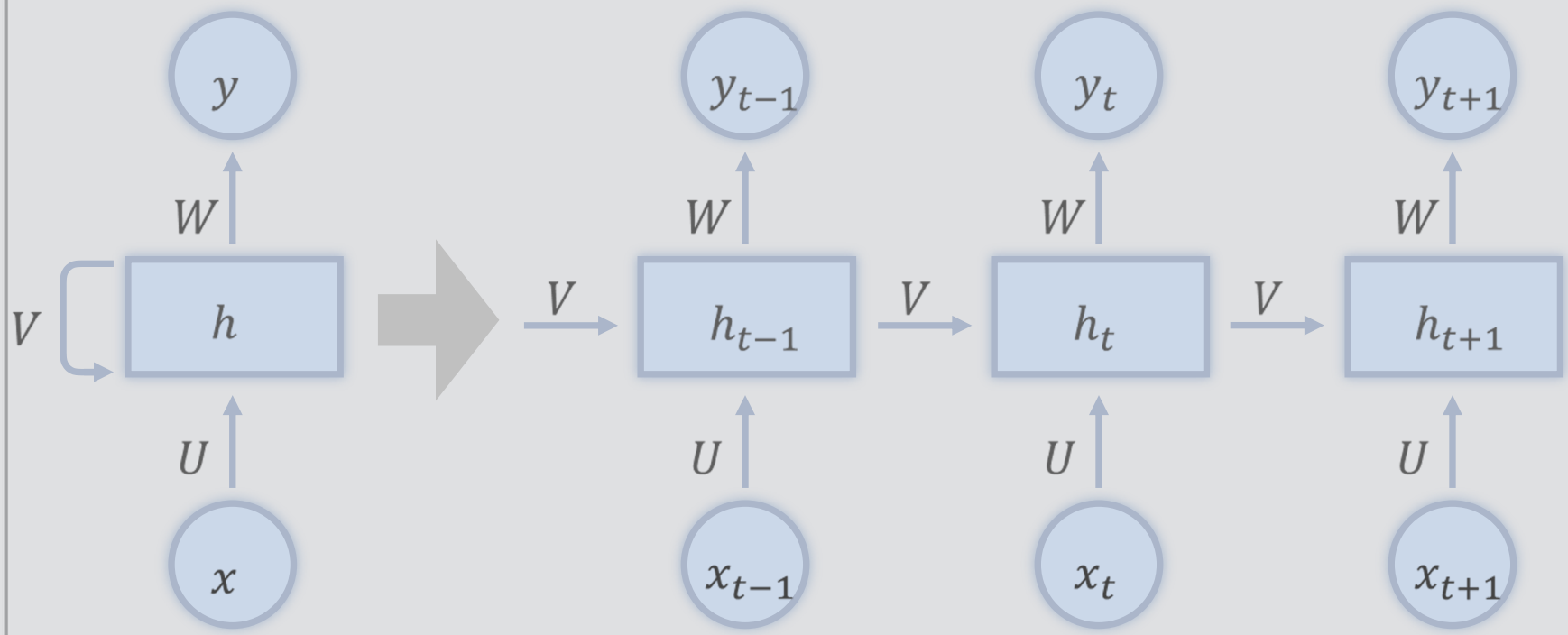
Сильные стороны

- Обрабатывают входы переменной длины
- Имеют механизм памяти
- Количество параметров не зависит от размера корпуса

Слабые стороны

- Медленные вычисления
- Механизм памяти практически не работает

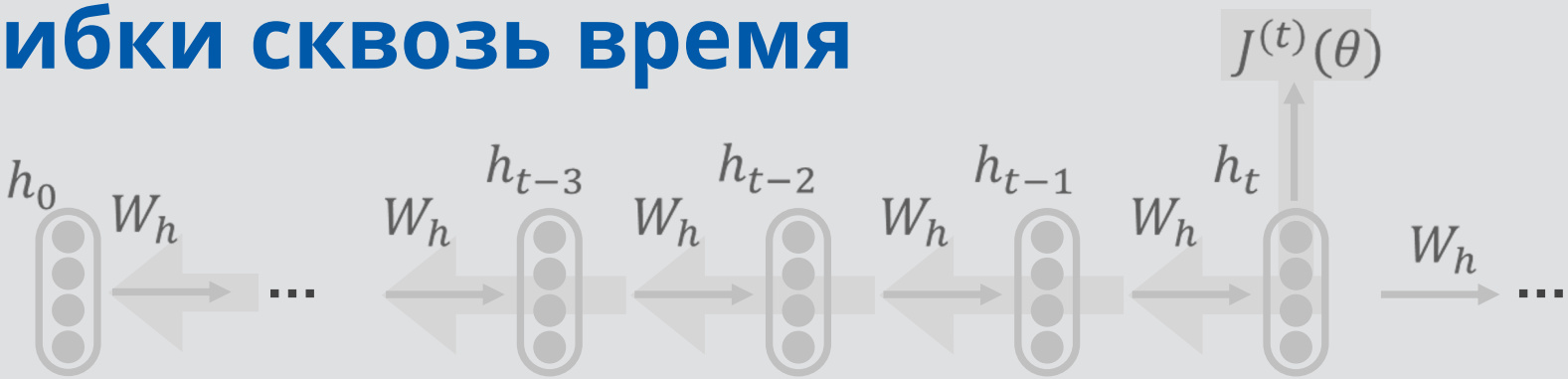
Механизм памяти



$$h_n = \sigma(h_{n-1}, x_n) = \sigma\left(\overbrace{\sigma(h_{n-2}, x_{n-1})}^{h_{n-1}}, x_n\right) = \sigma\left(\sigma\left(\overbrace{\sigma(h_{n-3}, x_{n-3})}^{h_{n-2}} x_{n-1}\right), x_n\right)$$

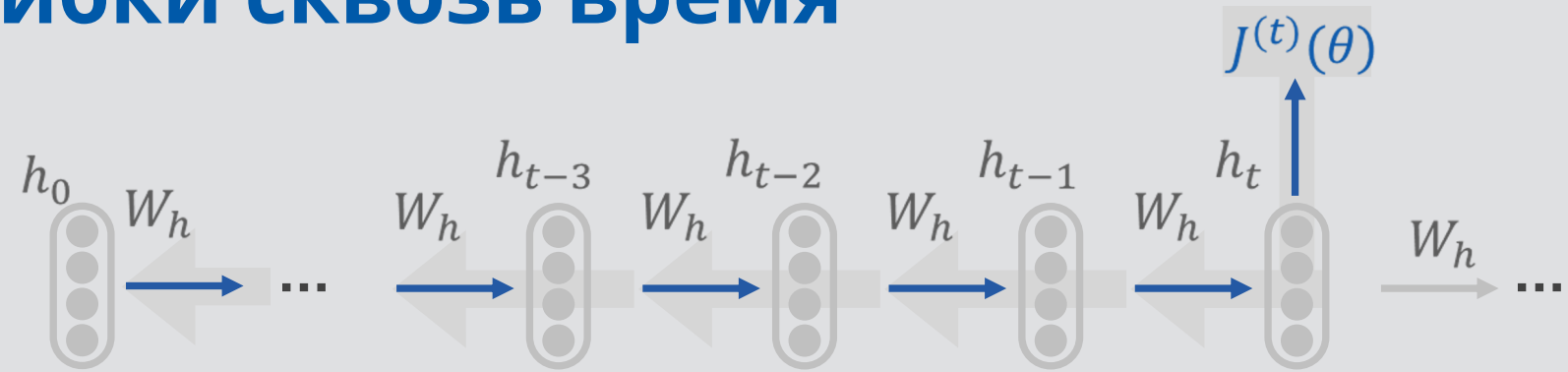
Получается, что на каждом шаге мы полностью обновляем вектор памяти!

Алгоритм обратного распространения ошибки сквозь время



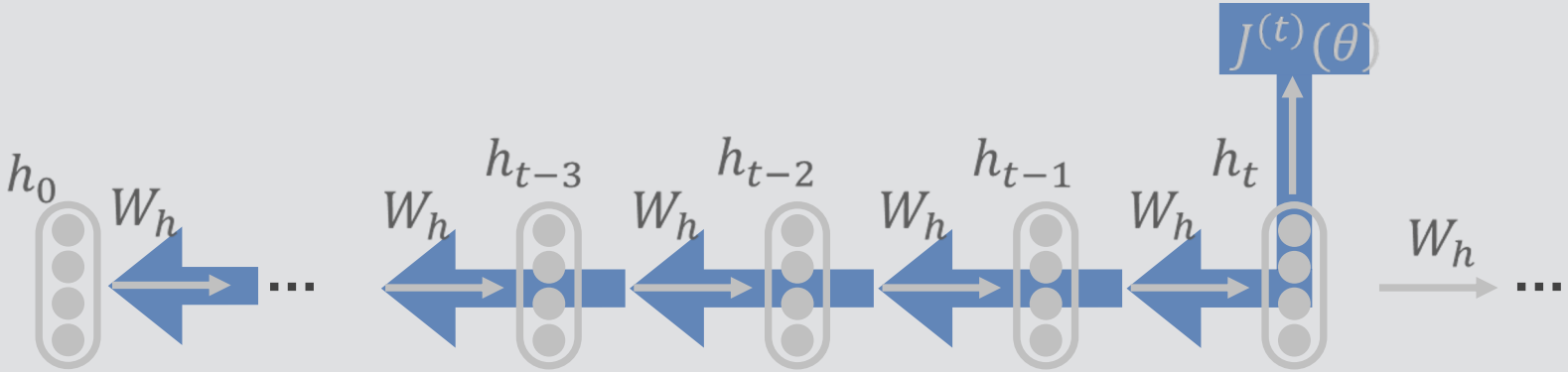
- Разворачиваем РНС во времени

Алгоритм обратного распространения ошибки сквозь время



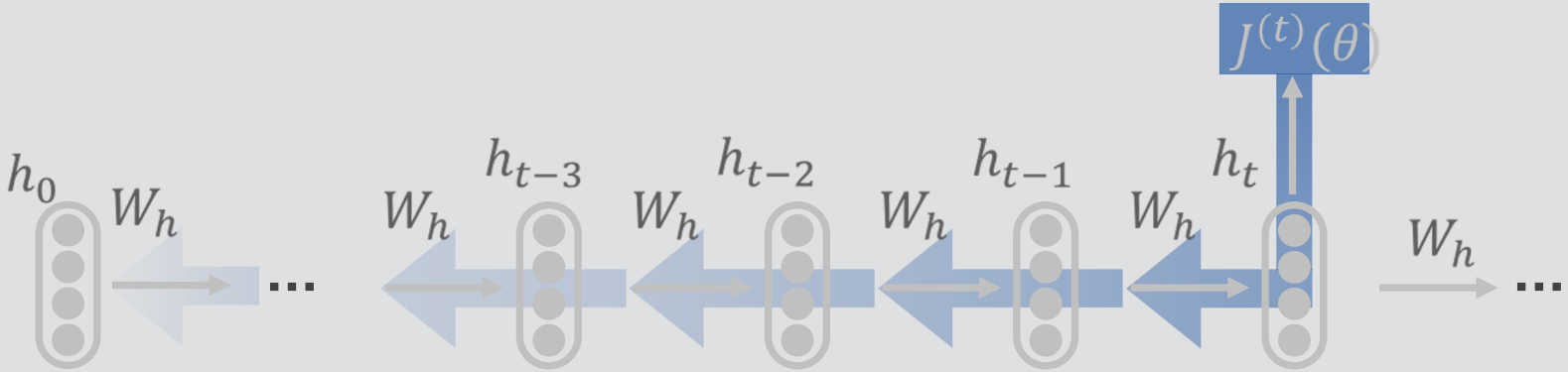
- Разворачиваем РНС во времени
- Вычисляем все локальные функции потерь

Алгоритм обратного распространения ошибки сквозь время



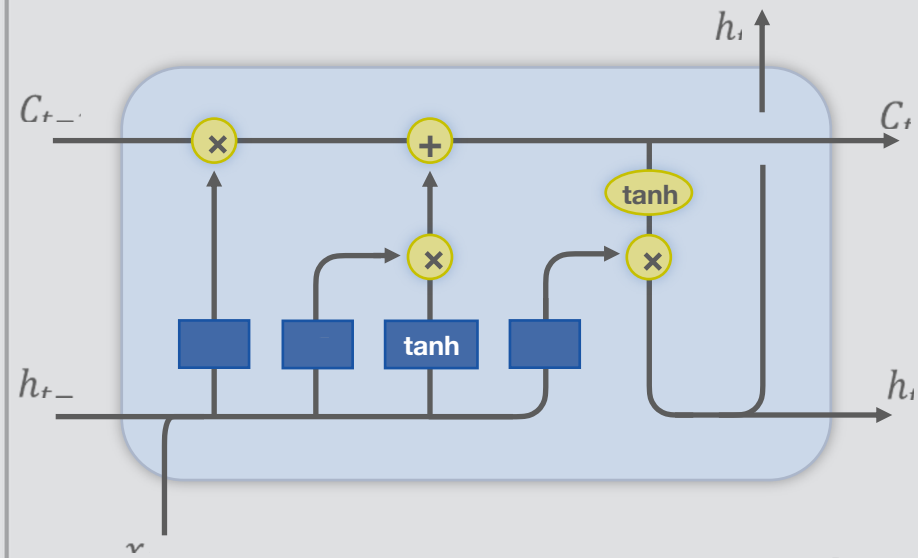
- Разворачиваем РНС во времени
- Вычисляем все локальные функции потерь
- Пробрасываем градиент от последнего слова к первому

Алгоритм обратного распространения ошибки сквозь время

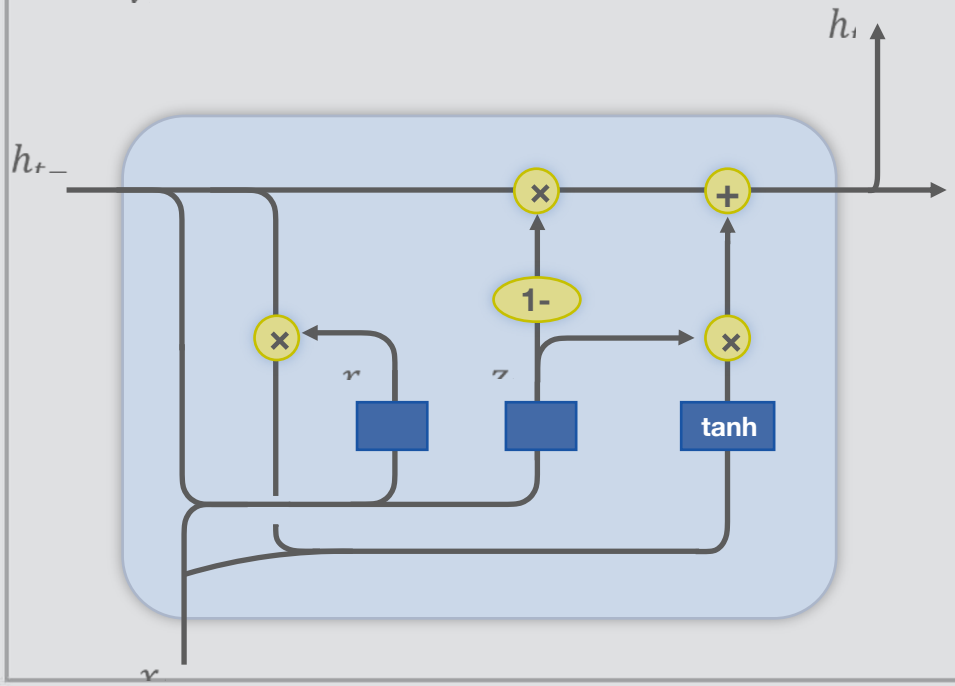


- Разворачиваем РНС во времени
- Вычисляем все локальные функции потерь
- Пробрасываем градиент от последнего слова к первому
- Градиент затухает: по цепному правилу дифференцирования возникает слишком много частных производных

Продвинутые архитектуры РНС

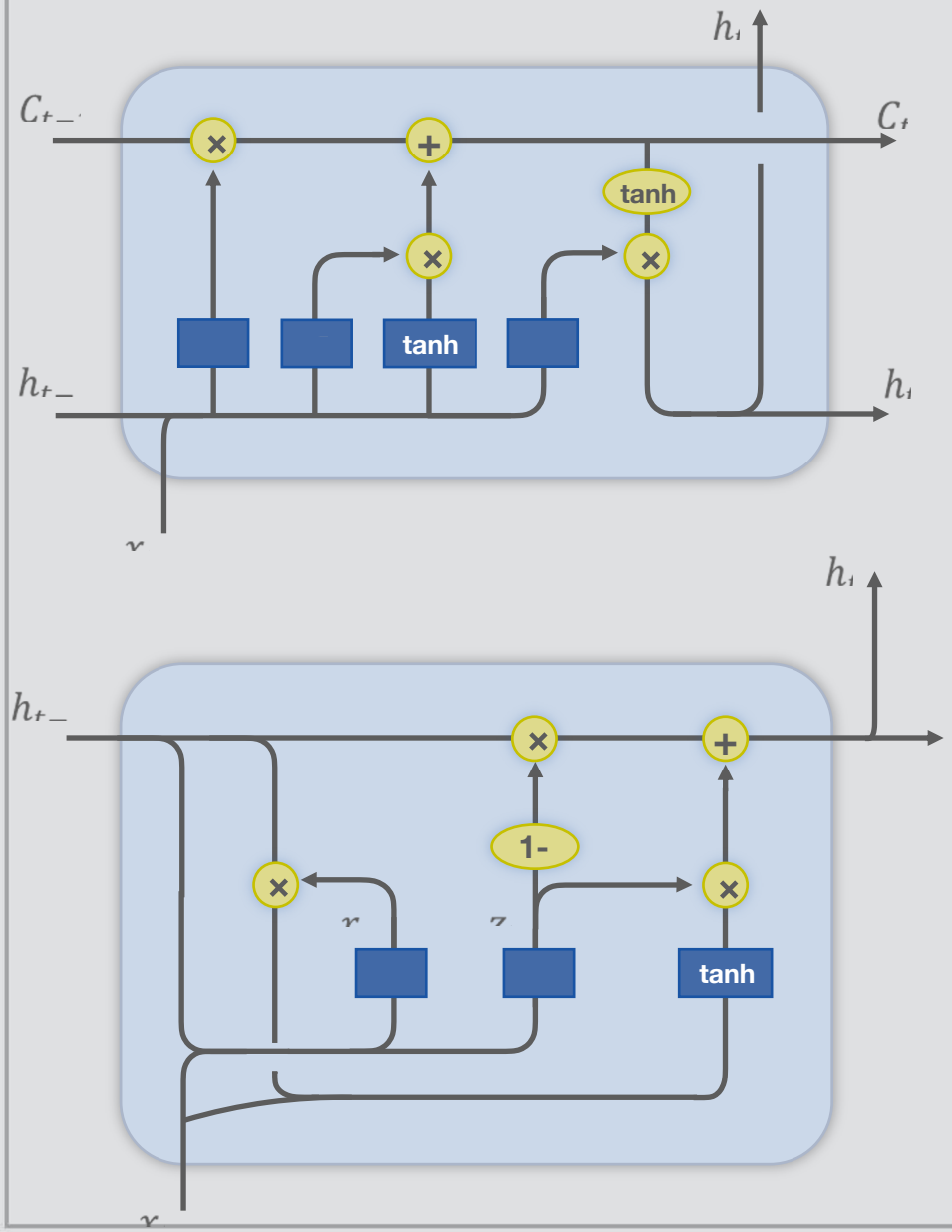


Long short term memory (LSTM)



Gated Recurrent Unit (GRU)

Продвинутые архитектуры РНС



Варианты РНС используют дополнительные маски преобразования для контроля за вектором памяти

Рекуррентные нейронные сети

- Семейство архитектур, которые обрабатывают последовательности производной длины
- Могут быть использованы в разных задачах обработки текстов, в том числе, для классификации текстов, генерации текстов и извлечения именованных сущностей

Генерация текстов

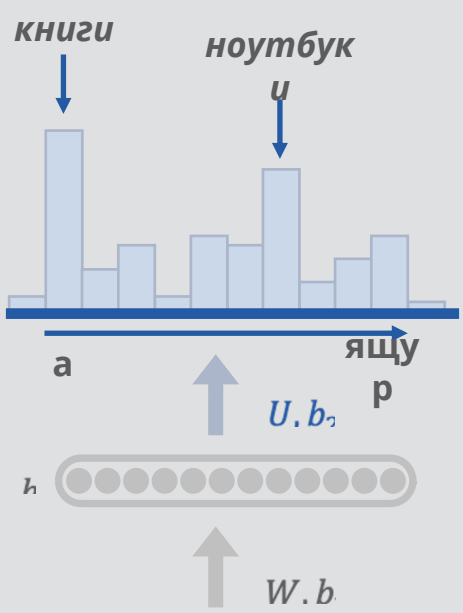
Постановка задачи

- Контекст из m слов: $x_1, \dots, x_m$
- Требуется его продолжить и сгенерировать n следующих слов: $x_{m+1}, \dots, x_{m+n}$
- Стратегия генерации P определяет, какое слово будет сгенерировано на каждом шаге: $P(x_{1:m+n}) = \prod_{i=1}^{m+n} P(x_i | x_1, \dots, x_{i-1})$

Постановка задачи

- Стратегия генерации P определяет, какое слово будет сгенерировано на каждом шаге: $P(x_{1:m+n}) = \prod_{i=1}^{m+n} P(x_i | x_1, \ldots, x_{i-1})$

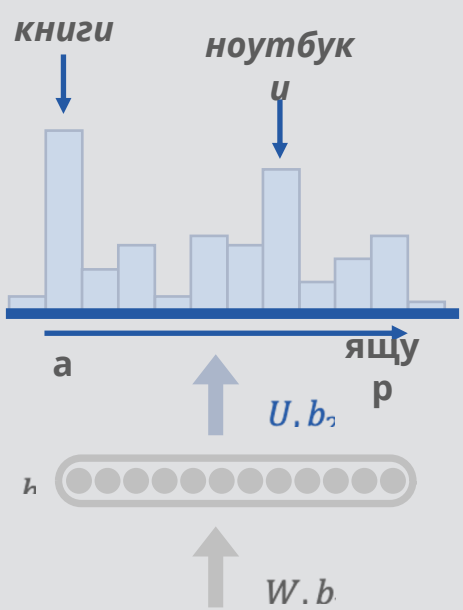
- Оценить все возможные комбинации n слов – слишком трудная задача



softmax на последнем слое задает распределение вероятностей

$$P(x_i = word_l | x_{1:i-1}) = u_l$$

и служит основой для разных стратегий декодирования



softmax на последнем слое задает распределение вероятностей

$$P(x_i = word_l | x_{1:i-1}) = u_l$$

и служит основой для разных стратегий декодирования

- Жадный поиск
  [greedy sampling]
- Лучевой поиск
  [beam search]
- Жадная генерация с температурой
  [greedy sampling with temperature]
- Top-k генерация
  [top-k sampling]
- Ядерный генерация
  [nucleus sampling]

Сравнение алгоритмов декодирования

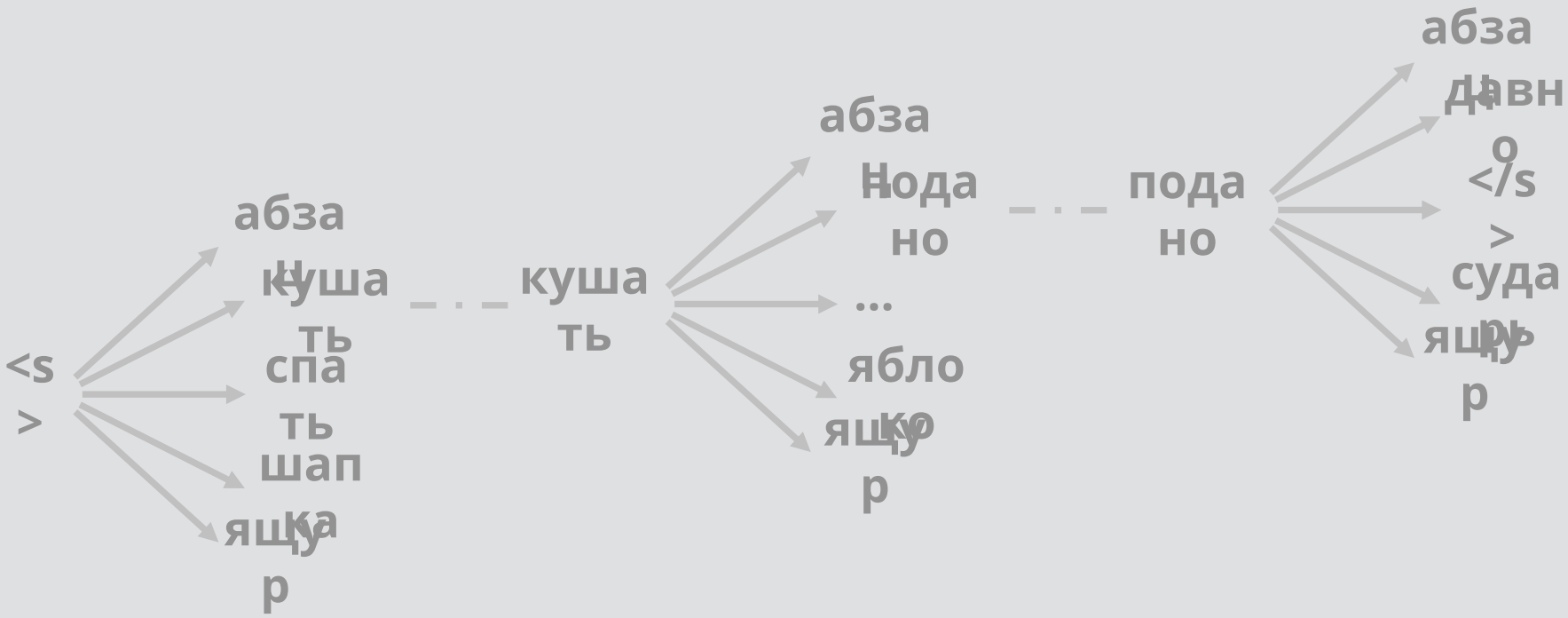
Поиск

- Детерминированные алгоритм
- Выбирают наиболее вероятные слова на каждом шаге

Генерация

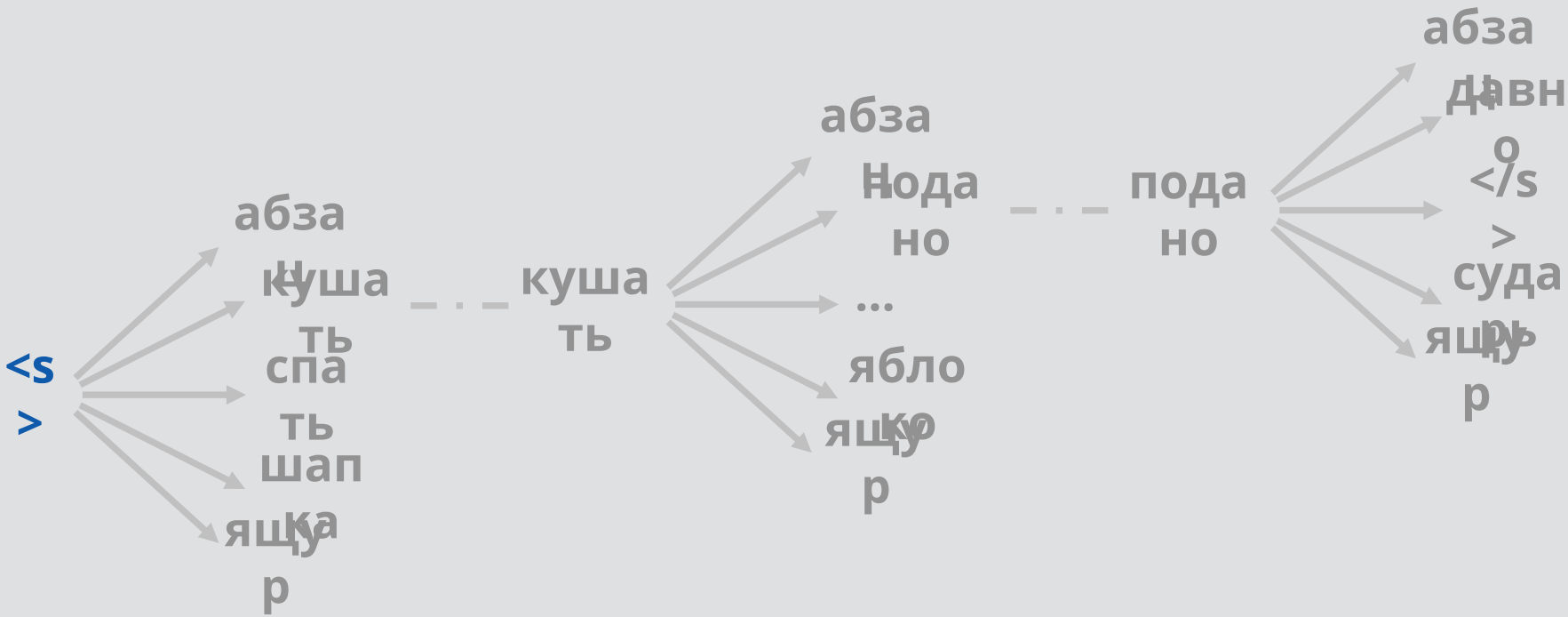
- Недетерминированные алгоритмы
- Генерируют из распределения, заданном на последнем слое

Жадный поиск



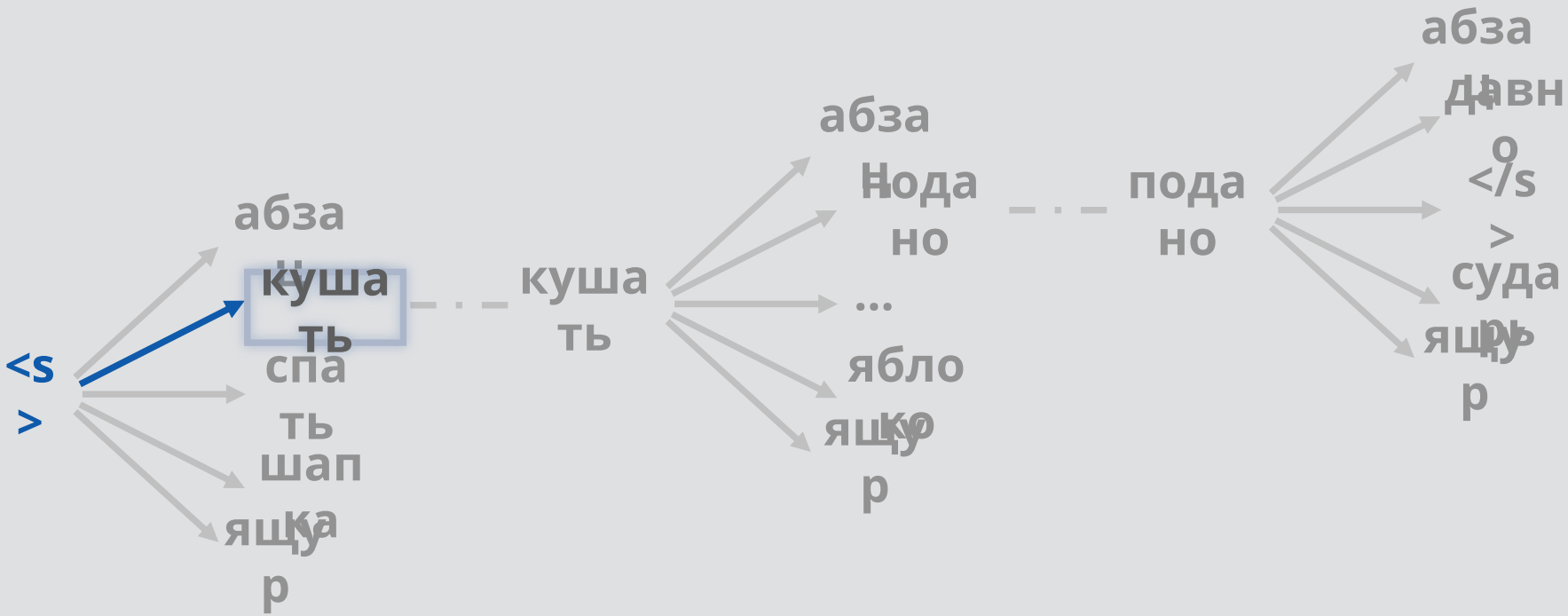
Жадный поиск

- Начинаем с символа начала предложения <s>



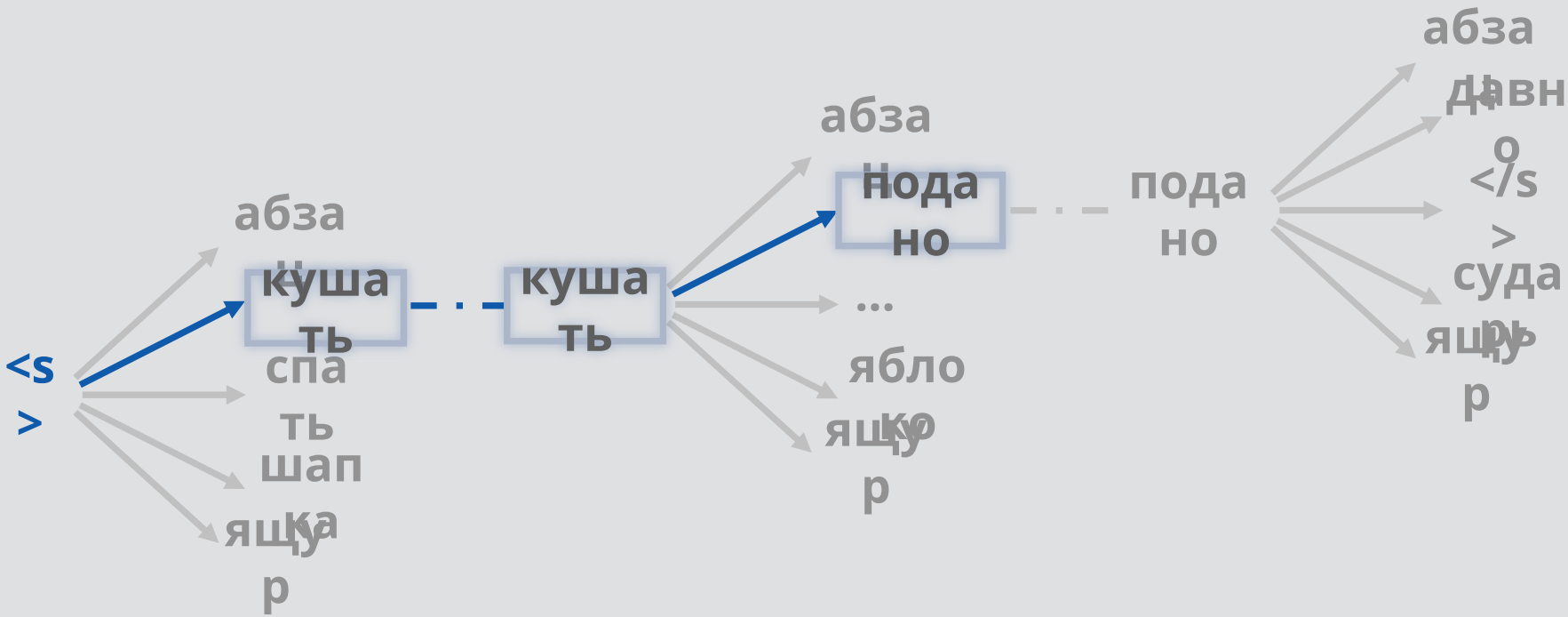
Жадный поиск

- **Выбираем наиболее вероятное первое слово**



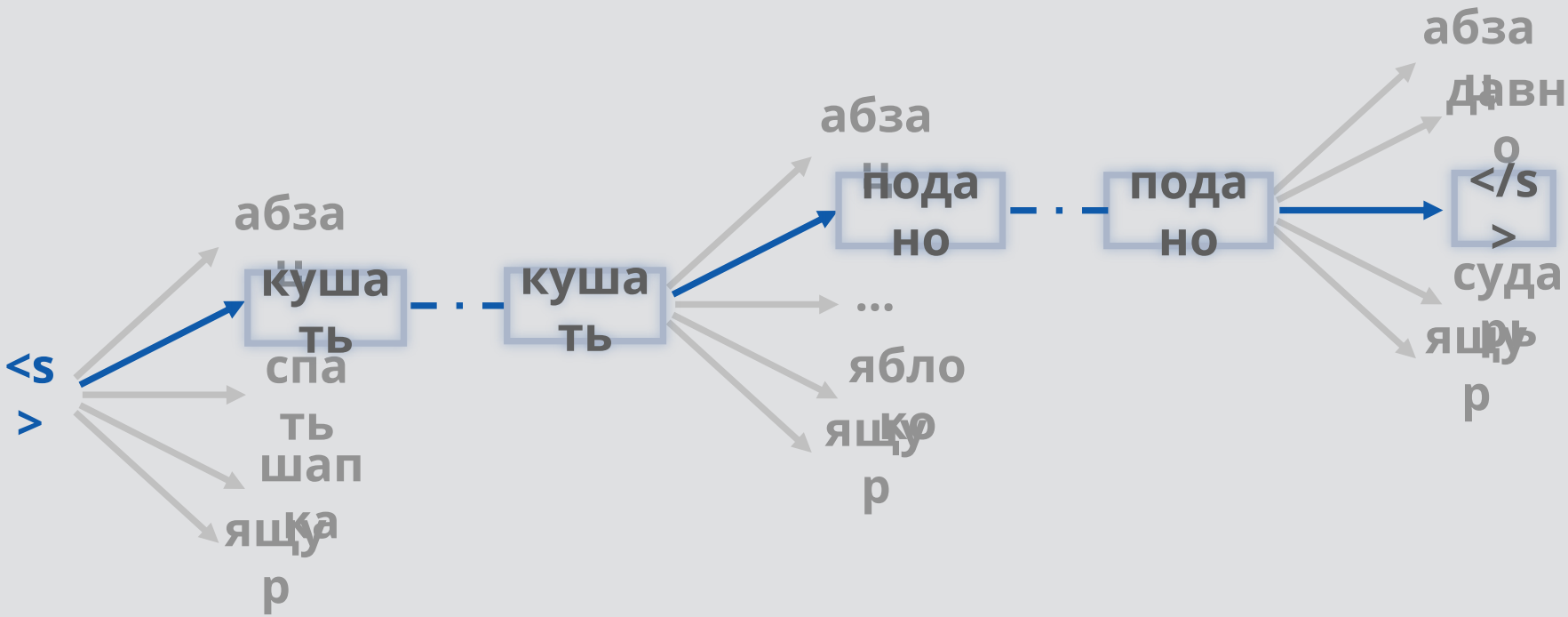
Жадный поиск

- Показываем РНС первое слово и выбираем второе

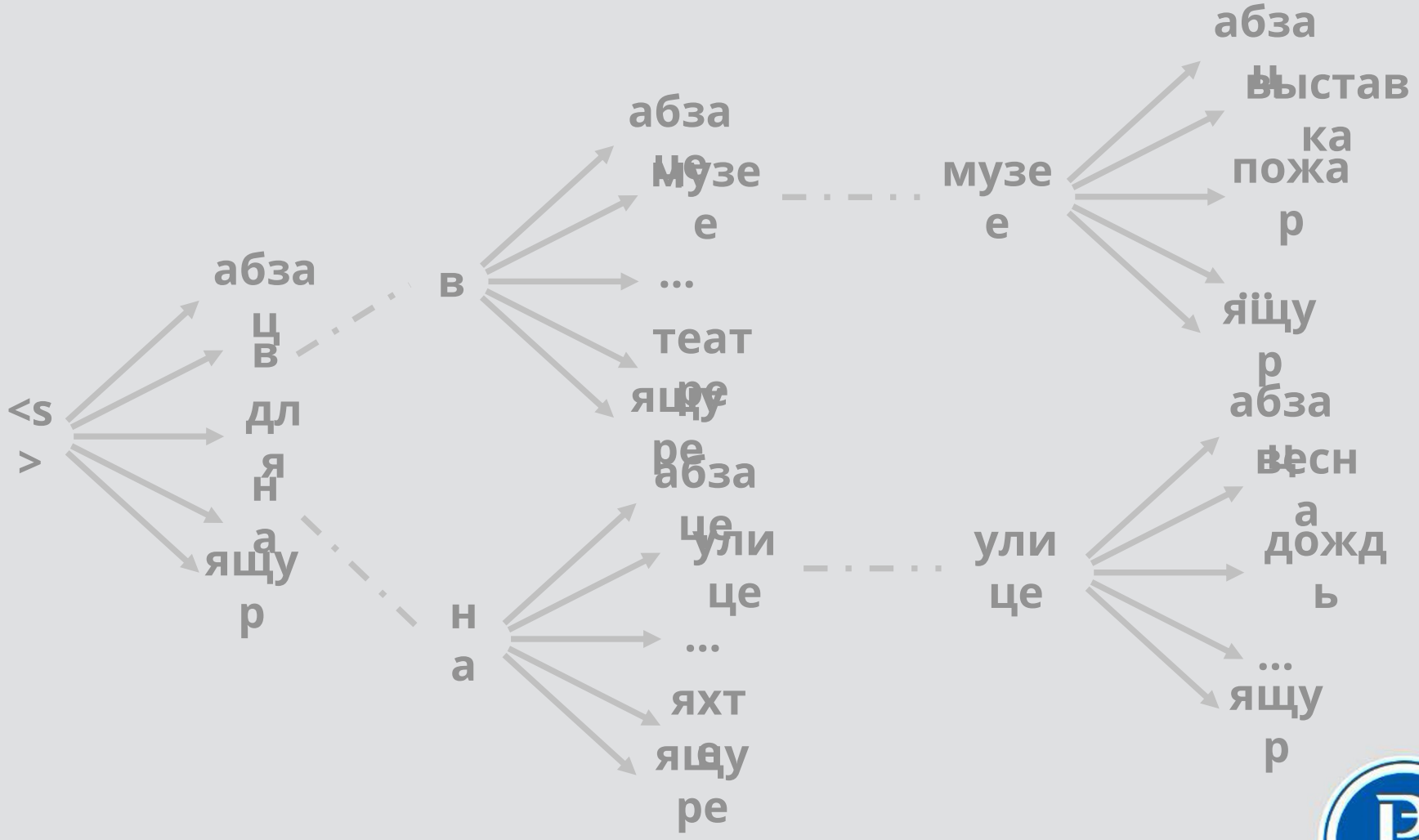


Жадный поиск

- Продолжаем, пока не выберем символ конца предложения </s>

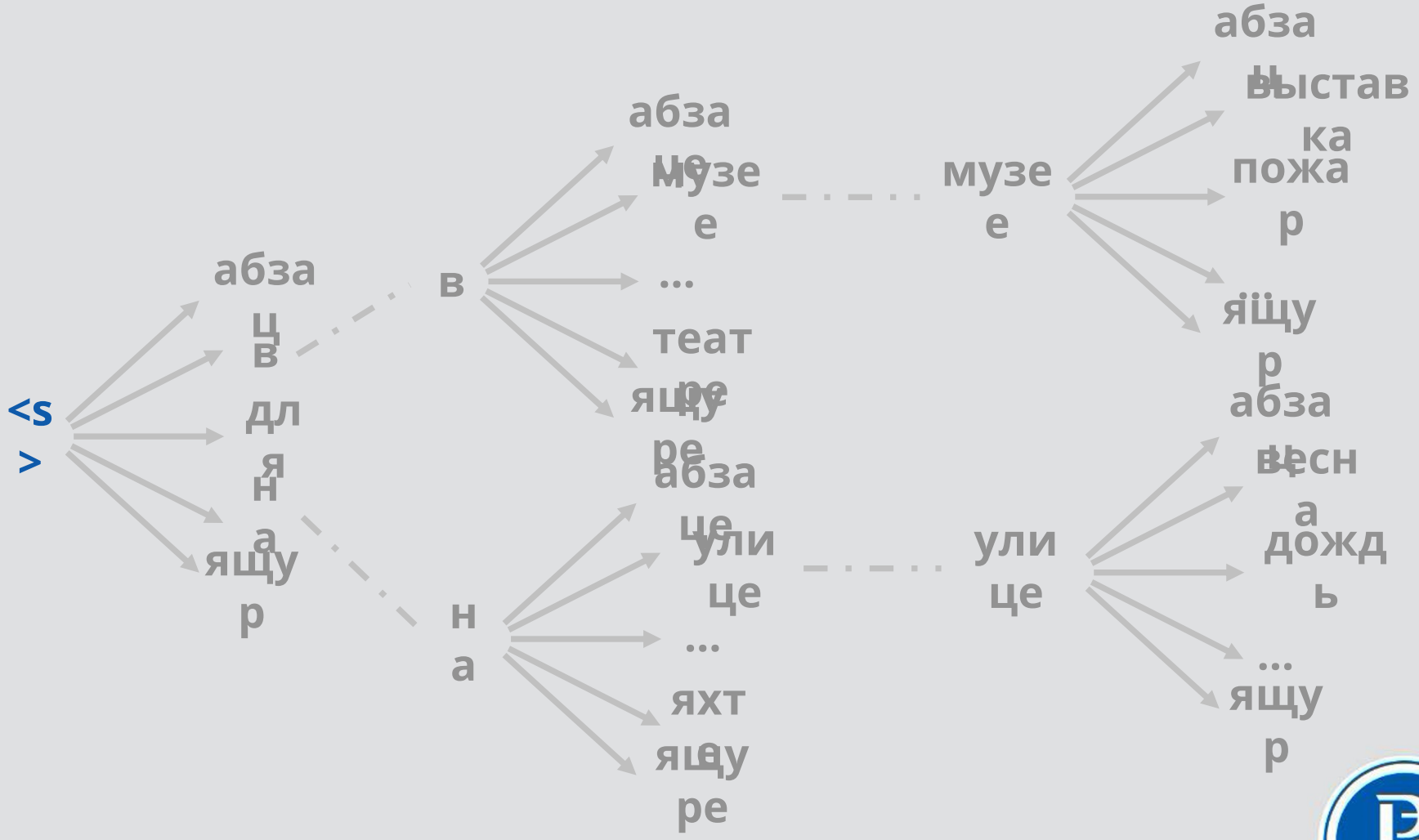


Лучевой поиск [beam search]



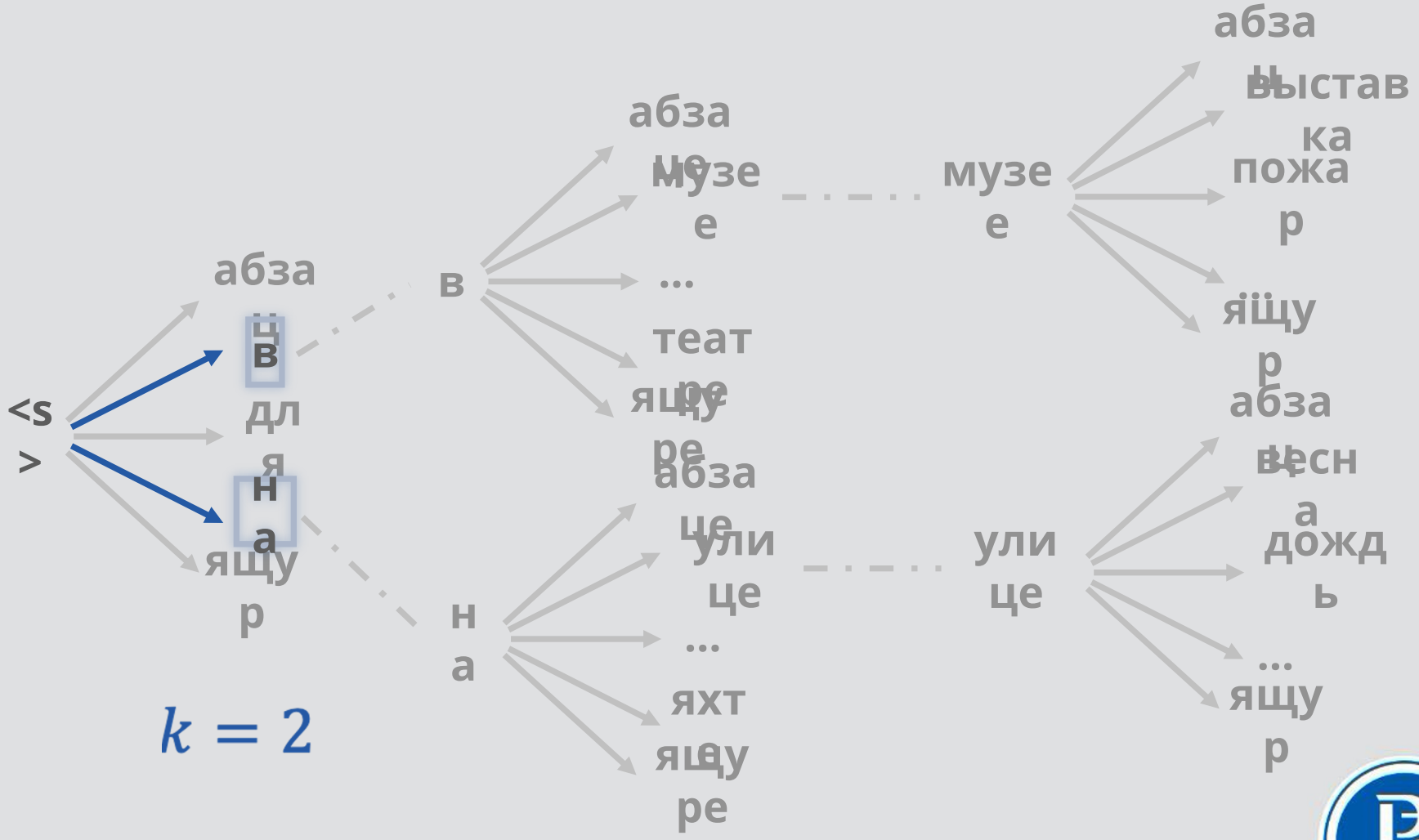
Лучевой поиск [beam search]

- Начинаем с символа начала предложения <s>



Лучевой поиск [beam search]

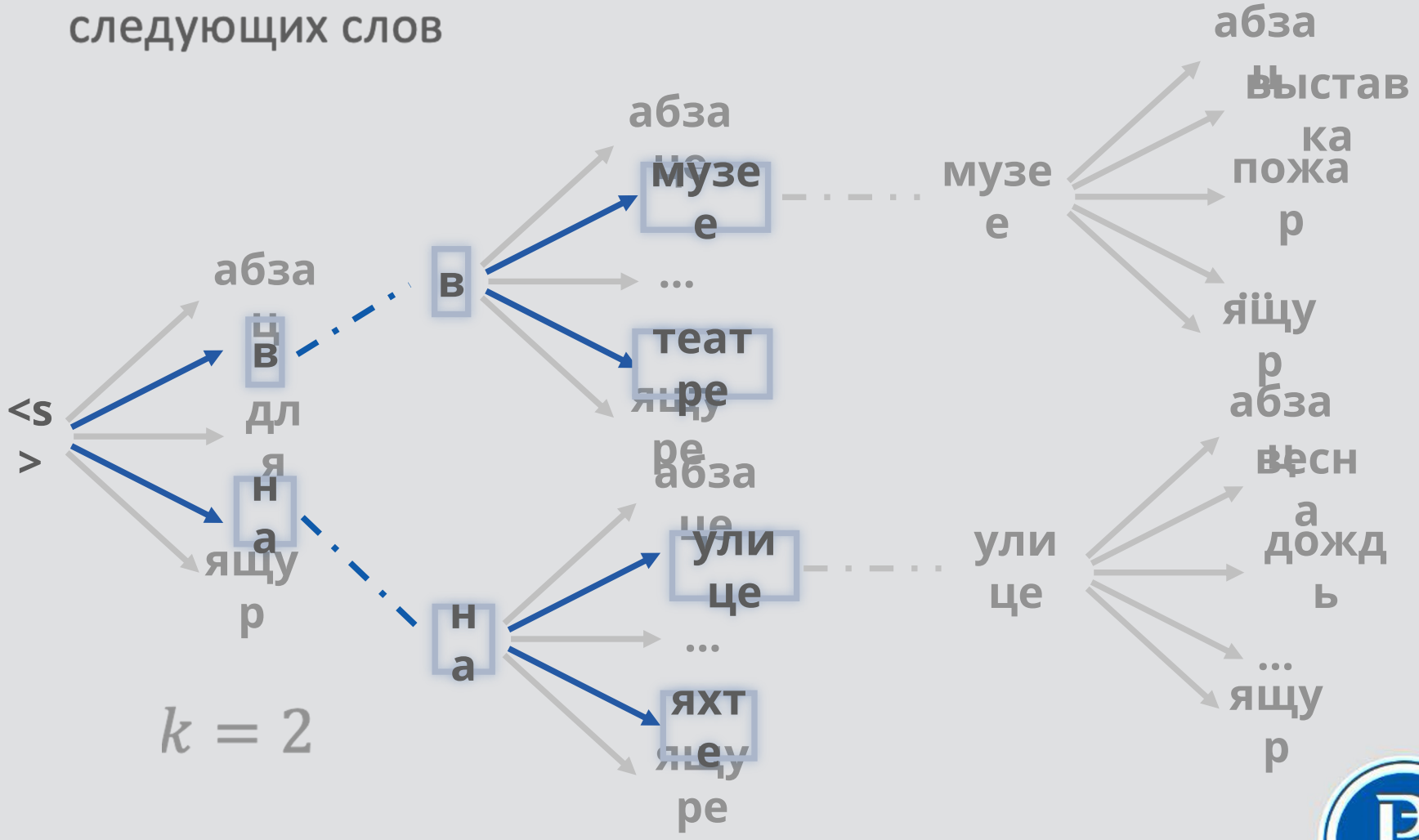
- Выбираем k наиболее вероятных слов



$k = 2$

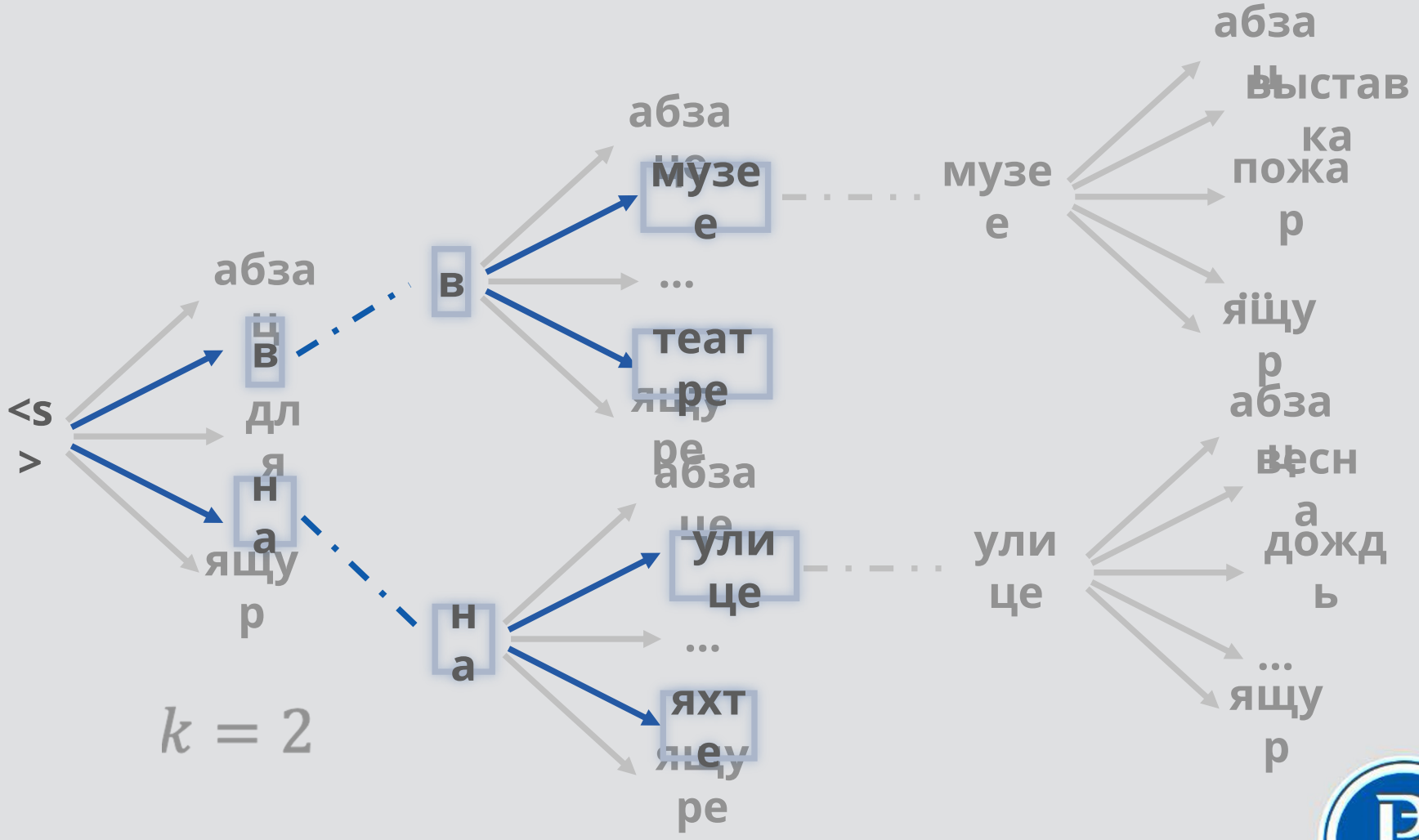
Лучевой поиск [beam search]

- Для каждого из k слов выбираем k наиболее вероятных следующих слов



Лучевой поиск [beam search]

- Получаем k^2 возможных продолжений

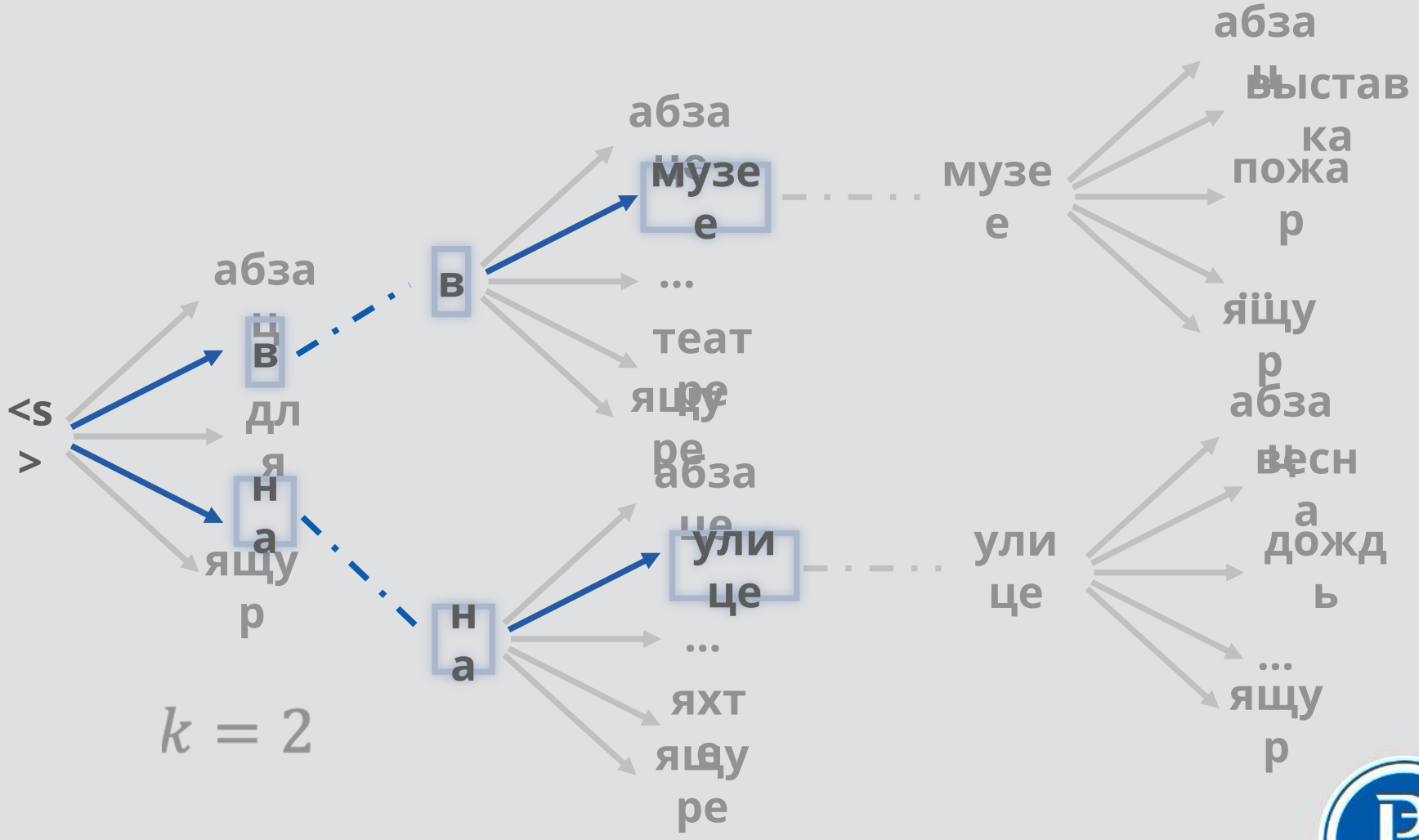


Лучевой поиск [beam search]

- Отбираем k наиболее вероятных продолжений, $k^2 - k$ продолжений отбрасываем

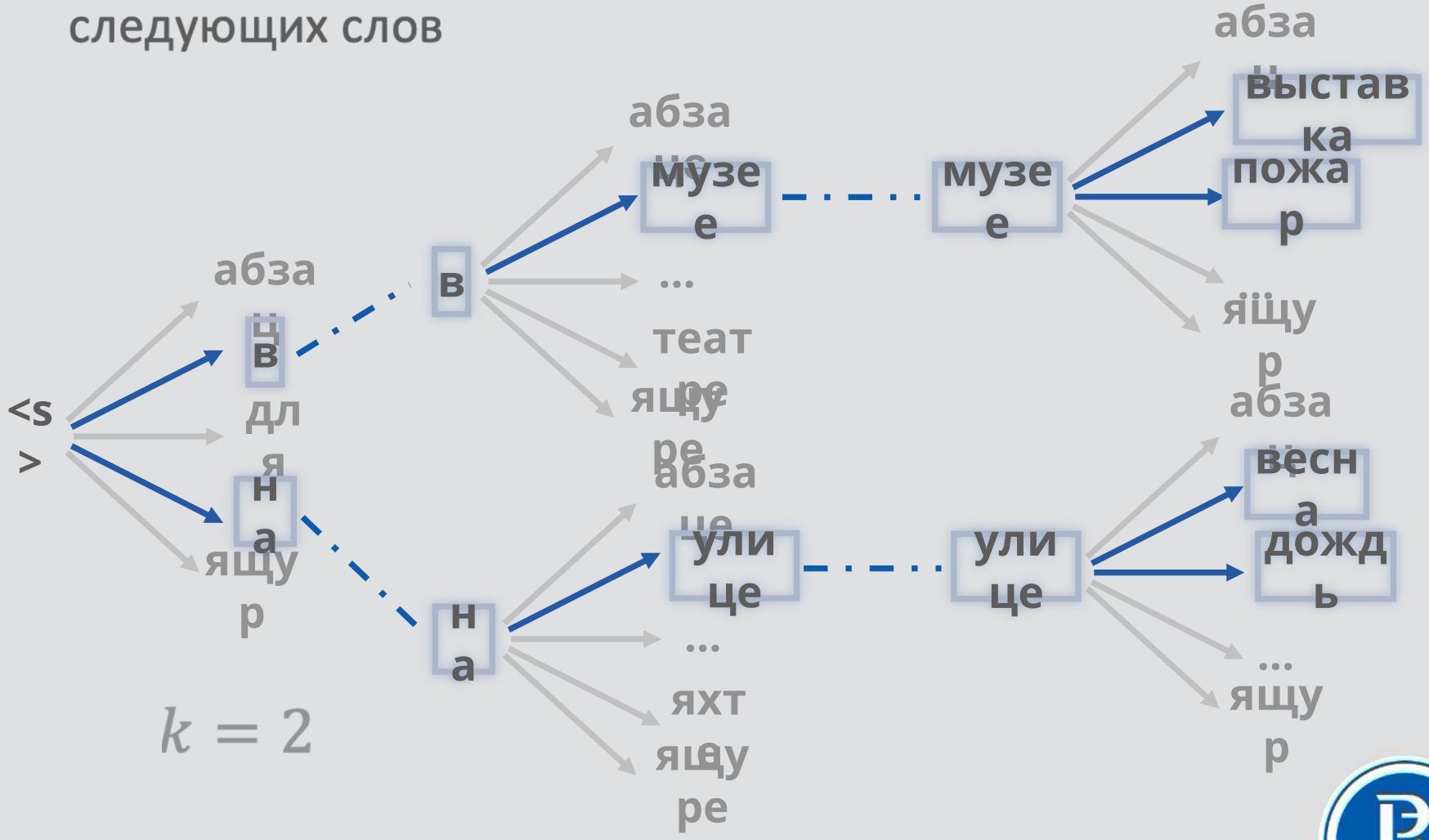
Лучевой поиск [beam search]

- **Повторим предыдущие шаги:**



Лучевой поиск [beam search]

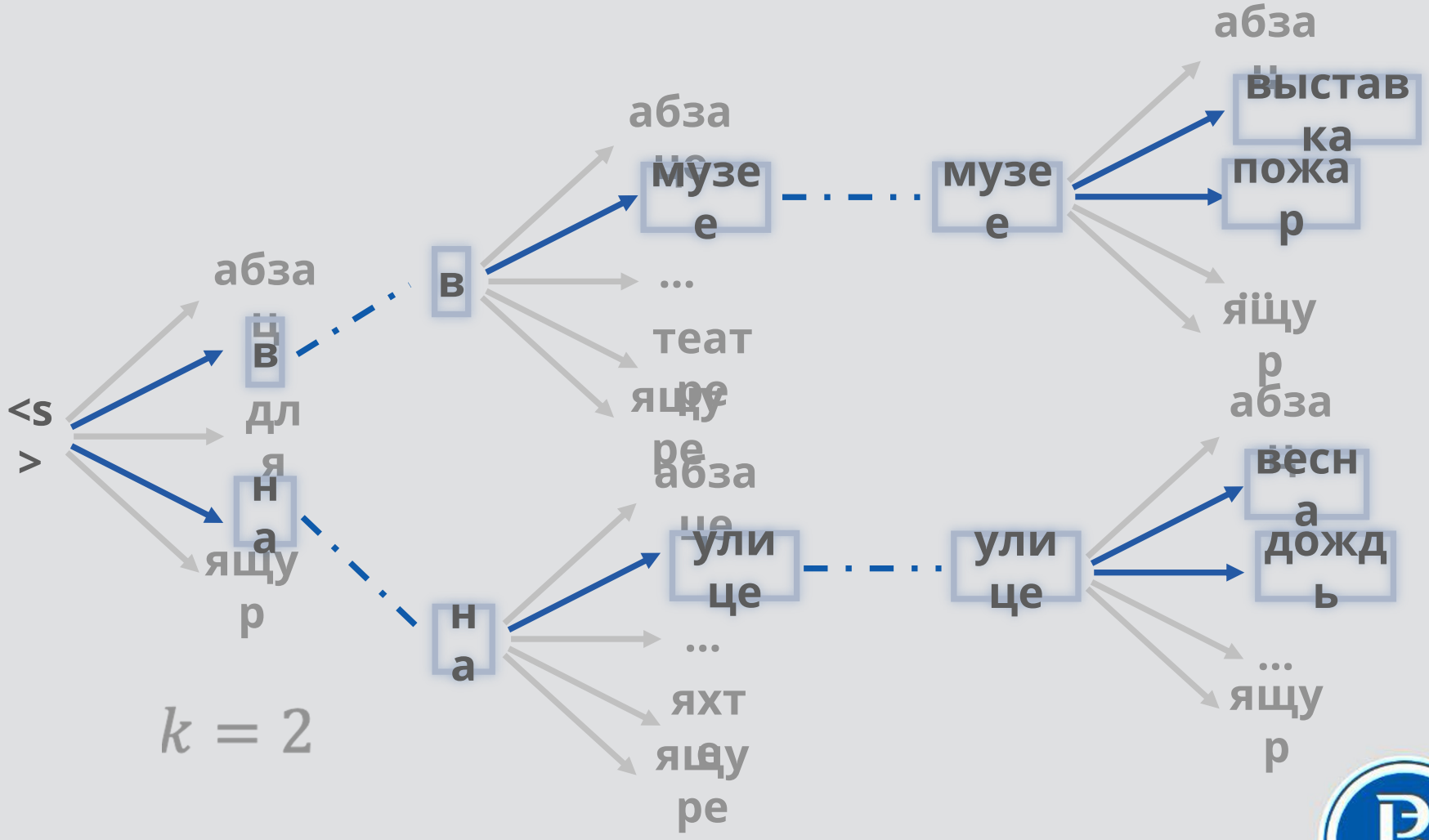
- Для каждого из k слов выбираем k наиболее вероятных следующих слов



$k = 2$

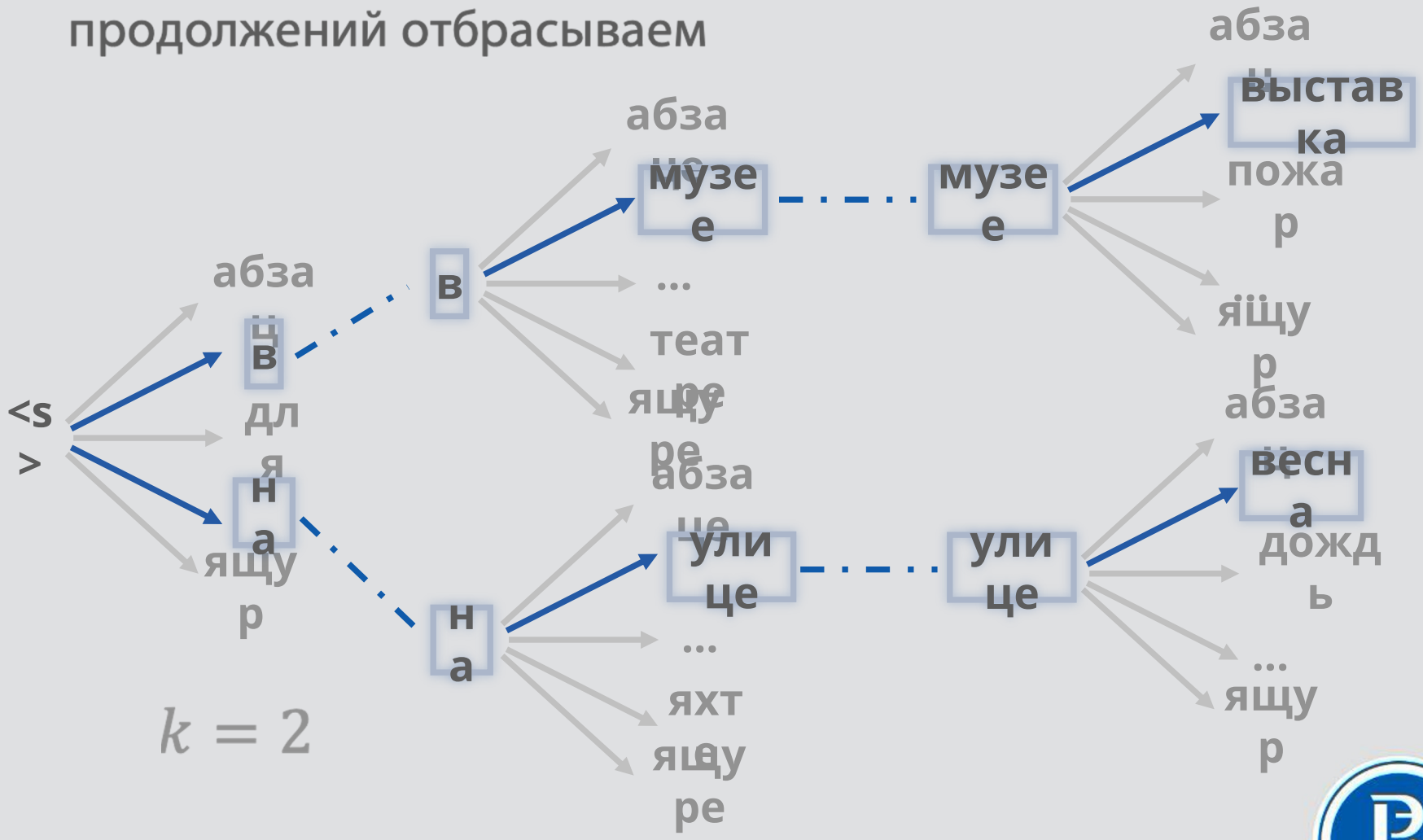
Лучевой поиск [beam search]

- Получаем k^2 возможных продолжений



Лучевой поиск [beam search]

- Отбираем k наиболее вероятных продолжений, $k^2 - k$ продолжений отбрасываем



Сравнение алгоритмов поиска

Жадный поиск

Лучевой поиск

Сравнение алгоритмов поиска

Жадный поиск

- Детерминированный алгоритм
- Быстрый алгоритм
- Склонен выбирать лучшее решение на каждом шаге
- Не гарантирует разнообразия реплик

Лучевой поиск

Сравнение алгоритмов поиска

Жадный поиск

- **Детерминированный алгоритм**
- **Быстрый алгоритм**
- **Склонен выбирать лучшее решение на каждом шаге**
- **Не гарантирует разнообразия реплик**

Лучевой поиск

- Детерминированный алгоритм
- Эквивалентен жадному поиску при $k = 1$
- Вычислительно сложный
- Позволяет генерировать более разнообразные реплики
- Сгенерированные реплики часто бывают слишком общими

Недетерменированные стратегии генерации

- Генерация с температурой: взвесим выходы softmax слоя

$$P(x_i = \mathrm{word}_l | x_{1:i-1}) = \frac{exp(u_l/t)}{\sum_{l'} exp(u'_l/t)}, \text{вес } t \in [0,1)$$

назовем температурой

- Температура придает больший вес более вероятным словам
- Качество генерации повышается, но генерируемый текст становится более однообразным

Top-k генерация

- На каждом шаге отбираем топ-k наиболее вероятных слов

$$\mathrm{Vocab}^k \in \mathrm{Vocab}$$

- Введем нормировочный коэффициент:

$$p' = \sum_{x \in \mathrm{Vocab}^k} p(x_i | x_{1:i-1})$$

- *Fan, Angela, Mike Lewis, and Yann Dauphin. "Hierarchical Neural Story Generation." ACL 2018*

Top-k генерация

- Шкалируем выходы softmax слоя:

$$P'(x_i|x_{1:i-1}) = \begin{cases} P(x_i|x_{1:i-1})/p', \text{ если } x_i \in \text{Vocab}^k \\ 0, \text{иначе} \end{cases}$$

- Проблема: как выбирать значение k?

- *Fan, Angela, Mike Lewis, and Yann Dauphin. "Hierarchical Neural Story Generation." ACL 2018*

Ядерная генерация

- На каждом шаге отбираем по порогу минимальное множество вероятны слов

$$\mathrm{Vocab}^p \in \mathrm{Vocab} : \sum_{x \in \mathrm{Vocab}^p} P(x_i | x_{1:i-1}) \geq p$$

- Введем нормировочный коэффициент:

$$p' = \sum_{x \in \mathrm{Vocab}^p} p(x_i | x_{1:i-1})$$

- *Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. "The curious case of neural text degeneration." ICLR 2020*

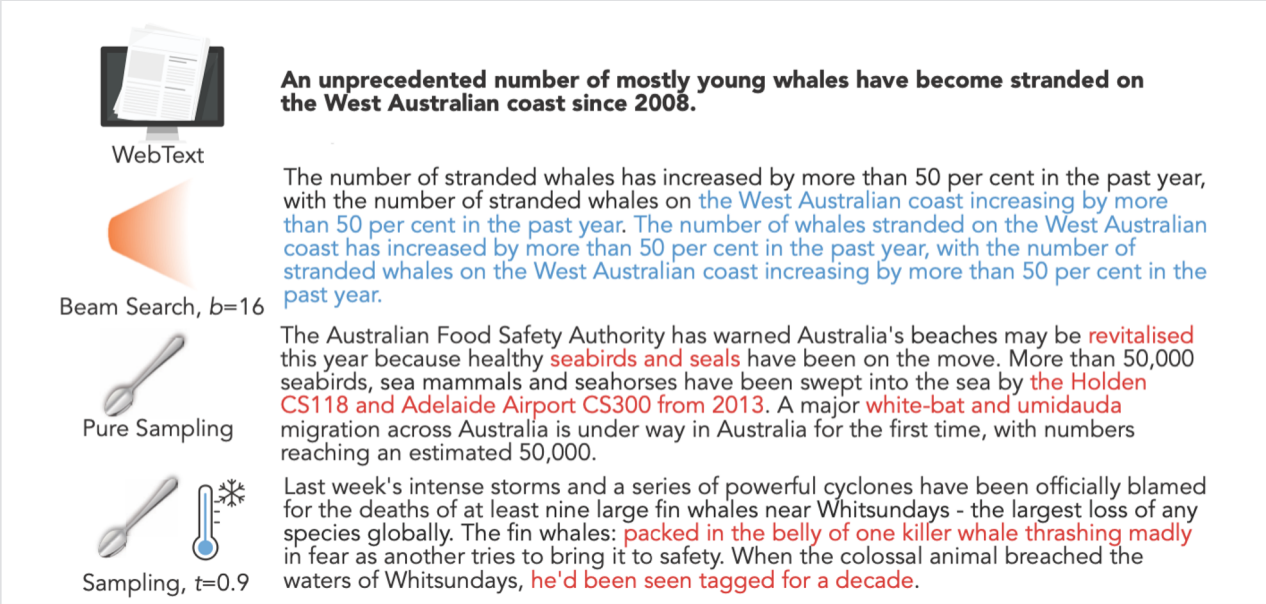
Ядерный поиск

- Шкалируем выходы softmax слоя:

$$P'(x_i|x_{1:i-1}) = \begin{cases} P'(x_i|x_{1:i-1})/p', \text{ если } x_i \in \text{Vocab}^k \\ 0, \text{иначе} \end{cases}$$

- Количество кандидатов в Vocab^p определяется на каждом шаге

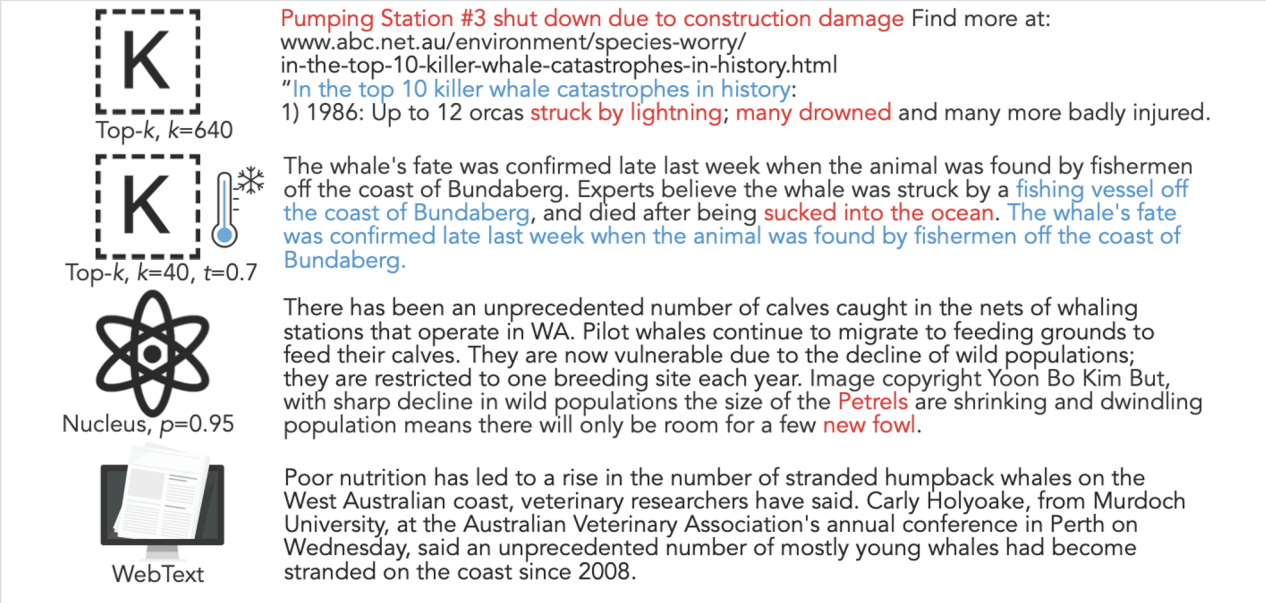
- ***Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. "The curious case of neural text degeneration." ICLR 2020***



Копирование

Рассогласование

- ***Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. "The curious case of neural text degeneration." ICLR 2020***



Копирование

Рассогласование

- ***Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. "The curious case of neural text degeneration." ICLR 2020***

Проблемы генерации текстов

- Повторение нескольких слов
- Отсутствие явной связи по смыслу
- Однообразие генерируемых текстов

Извлечение именованных сущностей

Пример

Непосредственным предшественником «Всемирного фонда дикой природы» был «Фонд охраны природы», созданный в Нью-Йорке в 1947 году Генри Фэйрфильдом Осборном

Пример

Непосредственным предшественником «Всемирного фонда дикой природы» был «Фонд охраны природы», созданный в Нью-Йорке в 1947 году Генри Фэйрфильдом Осборном

Organization

Organization

- **Organization** – компании, общественные организации, образовательные и медицинские учреждения и т.п.

Пример

Непосредственным предшественником «Всемирного фонда дикой природы» был «Фонд охраны природы», созданный в Нью-Йорке в 1947 году Генри Фэйрфильдом Осборном

Organization

Organization

Location

- **Organization** – компании, общественные организации, образовательные и медицинские учреждения и т.п.
- **Location** – города, страны

Пример

Organization

Непосредственным предшественником «Всемирного фонда дикой природы» был «Фонд охраны природы», созданный в Нью-Йорке в 1947 году Генри Фэйрфильдом Осборном

Organization

Location

Date

- **Organization** – компании, общественные организации, образовательные и медицинские учреждения и т.п.
- **Location** – города, страны
- **Date** – даты и время

Пример

Organization

Непосредственным предшественником «Всемирного фонда дикой природы» был «Фонд охраны природы», созданный в Нью-Йорке в 1947 году Генри Фэйрфильдом Осборном

Organization

Location Date Person

- **Organization** – компании, общественные организации, образовательные и медицинские учреждения и т.п.
- **Location** – города, страны
- **Date** – даты и время
- **Person** – имена и фамилии

IOB разметка

Фонд	→	B-Organization
охраны	→	I-Organization
природы	→	I-Organization
создан	→	O
в	→	O
Нью-Йорке	→	B-Location
в	→	O
1947	→	B-Date
году	→	I-Date
Генри	→	B-Person
Фейрфильдом		I-Person
Осборном		I-Person

Корпусы

- Persons 1000
  - Упоминания личных имен в новостных статьях
- FactRuEval
  - Новостные статьи и фрагменты Википедии
  - Стандартные типы сущностей
- Balto-Slavic NLP 2019
  - Новости о предвыборной компании в США и ЕС
  - 7 славянских языков
  - Стандартные типы сущностей

Корпусы

- **RuREBus**
  - **Документы стратегического планирования РФ**
  - **Задачи и цели планирования, организации, социальные институты, качественные и количественные показатели**

https://github.com/dialogue-evaluation/RuREBus

Формальная постановка задачи

Дана последовательность слов
$w^n = (w_1, \ldots, w_n), x \in V$

и метки (теги) слов
$y^n = (y_1, \ldots, y_n), y \in L$

Теги могут быть разными: например, частями речи

Формальная постановка задачи

Дана последовательность слов
$w^n = (w_1, \ldots, w_n), x \in V$

и метки (теги) слов
$y^n = (y_1, \ldots, y_n), y \in L$

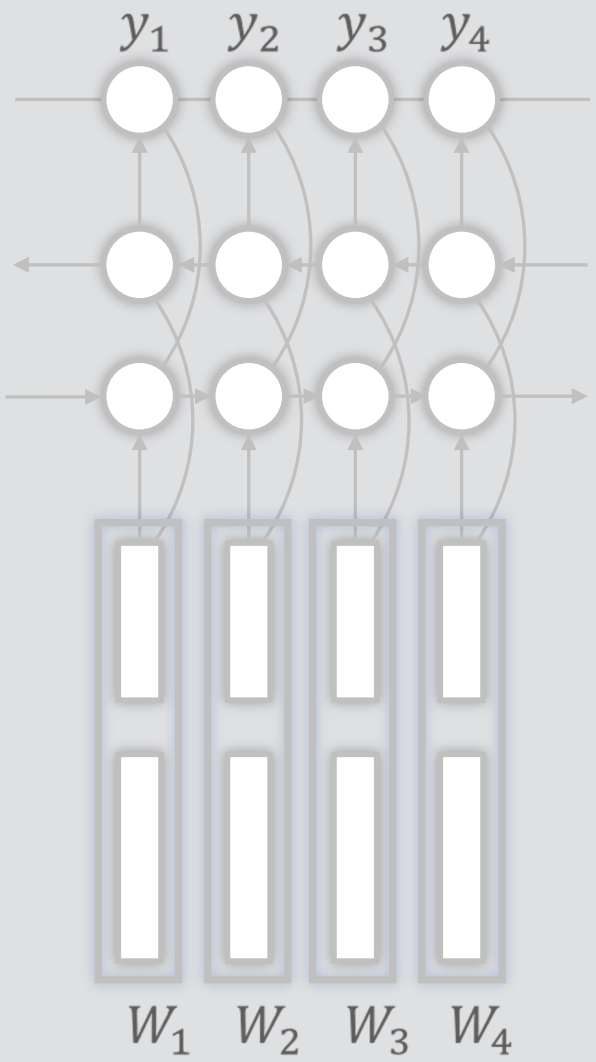
Теги могут быть разными: например, частями речи

Требуется обучить классификатор, который будет произвольной входной последовательности слов ставить в соответствие последовательность меток

$\gamma: w^n \to y^n$

CNN-biLSTM-CRF

Архитектура состоит из трех частей:



CRF

Обратная LSTM

Прямая LSTM

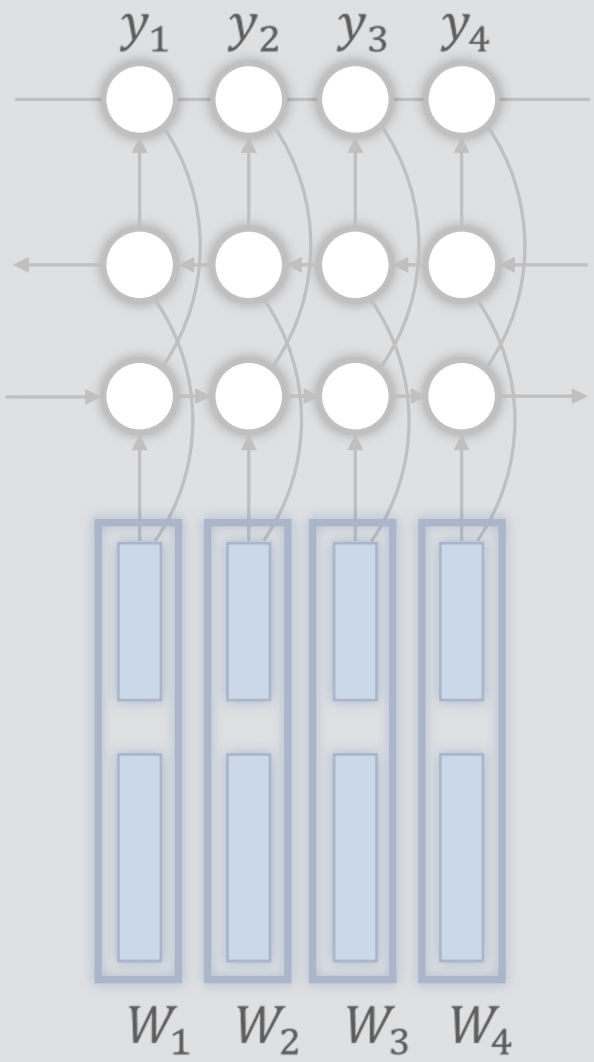
Символьное представлен

Вектора слов

CNN-biLSTM-CRF

Архитектура состоит из трех частей:

- Два вида представления слов



CRF

Обратная LSTM

Прямая LSTM

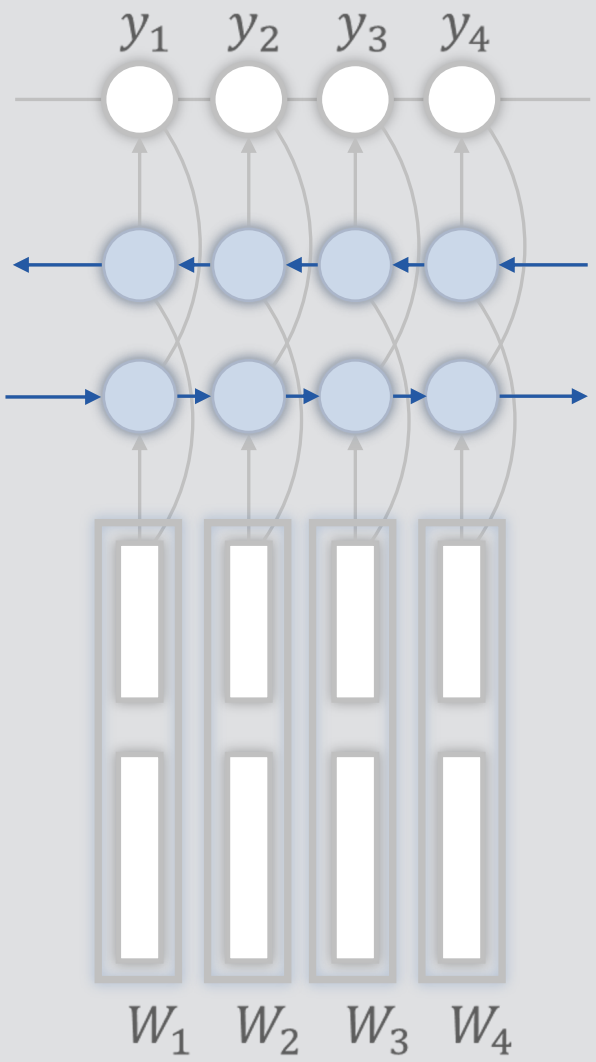
Символьное представлен е

Вектора слов

CNN-biLSTM-CRF

Архитектура состоит из трех частей:

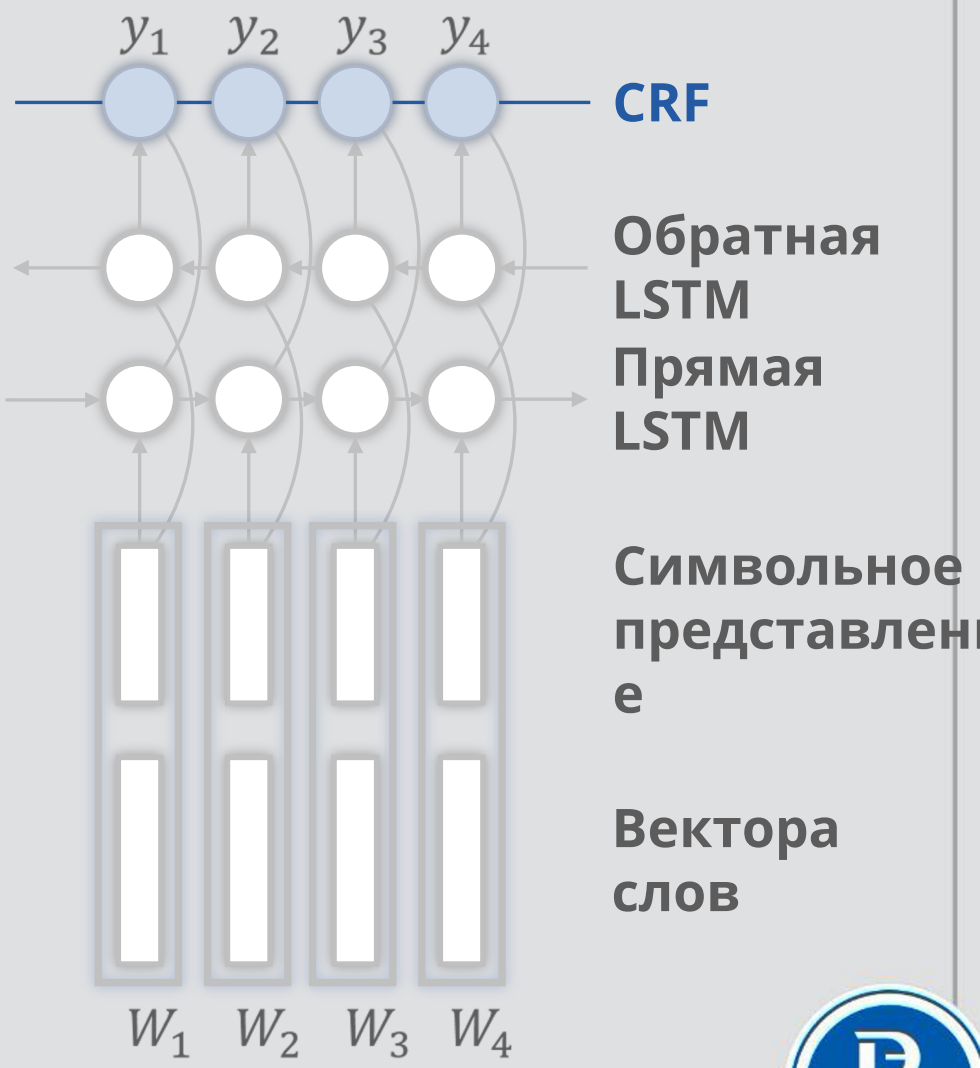
- Два вида представления слов
- LSTM для анализа контекста



CNN-biLSTM-CRF

Архитектура состоит из трех частей:

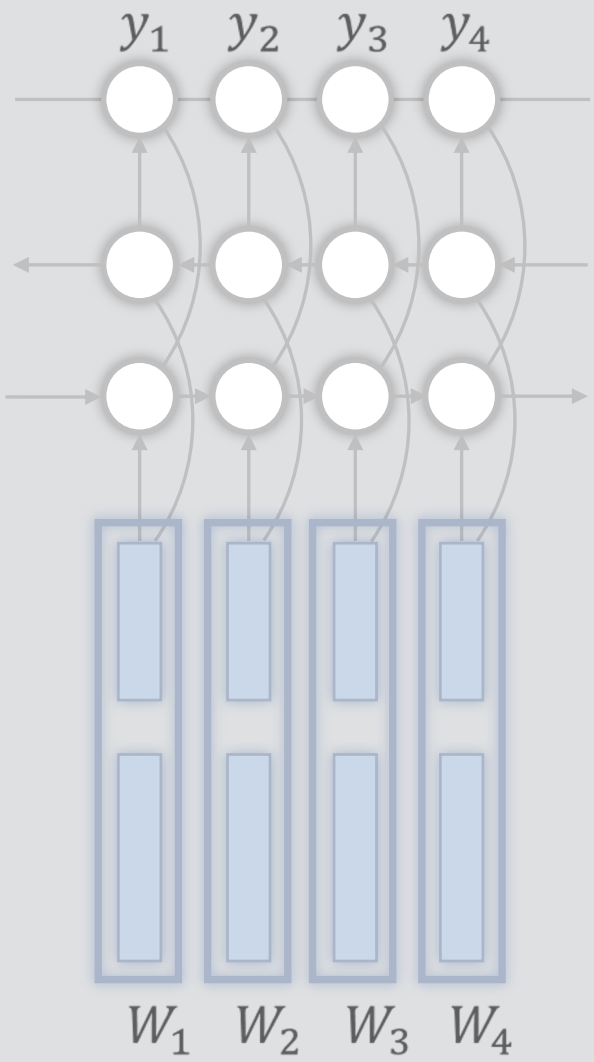
- Два вида представления слов
- LSTM для анализа контекста
- Условное случайное поле (CRF) для определения меток



CNN-biLSTM-CRF

Два вида представления слов

$x_i = [x_i^{\text{word}}, x_i^{\text{char}}]$



CRF

Обратная LSTM

Прямая LSTM

Символьное представление

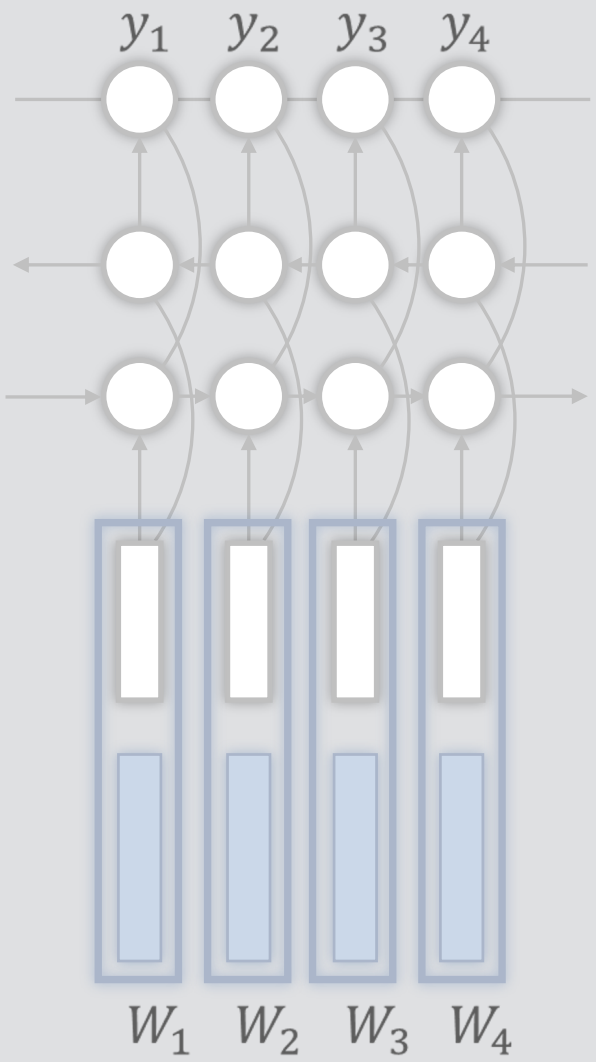
Вектора слов

CNN-biLSTM-CRF

Два вида представления слов

$x_i = [x_i^{\text{word}}, x_i^{\text{char}}]$

- Предобученные вектора слов $x_i^{\text{word}} = Ew_i$
- Свертка на уровне символов



CRF

Обратная LSTM

Прямая LSTM

Символьное представлен е

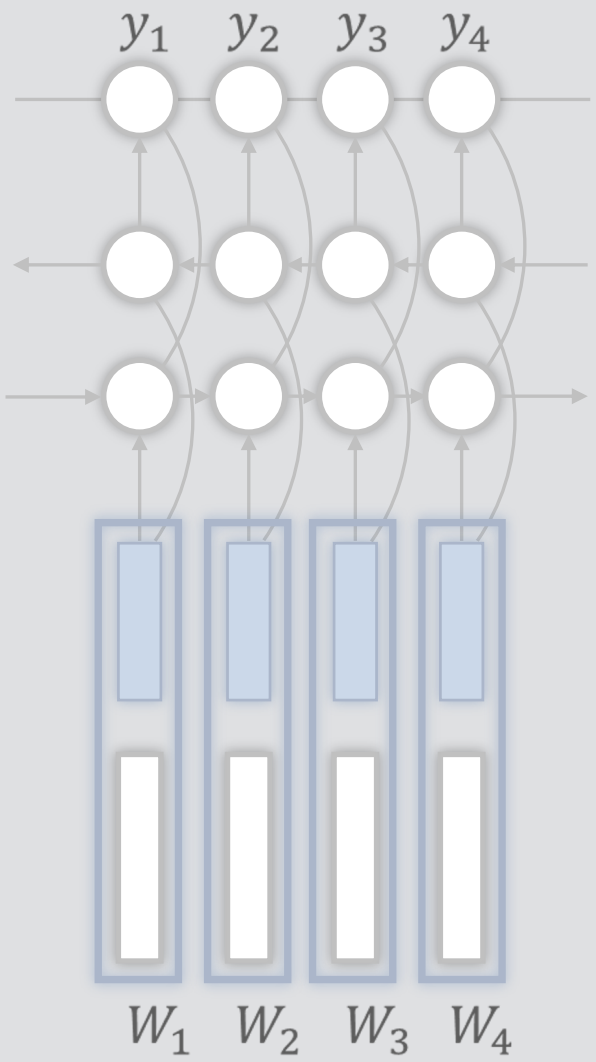
Вектора слов

CNN-biLSTM-CRF

Два вида представления слов

$x_i = [x_i^{\text{word}}, x_i^{\text{char}}]$

- Предобученные вектора слов $x_i^{\text{word}} = Ew_i$
- Свертка на уровне символов



CRF

Обратная LSTM

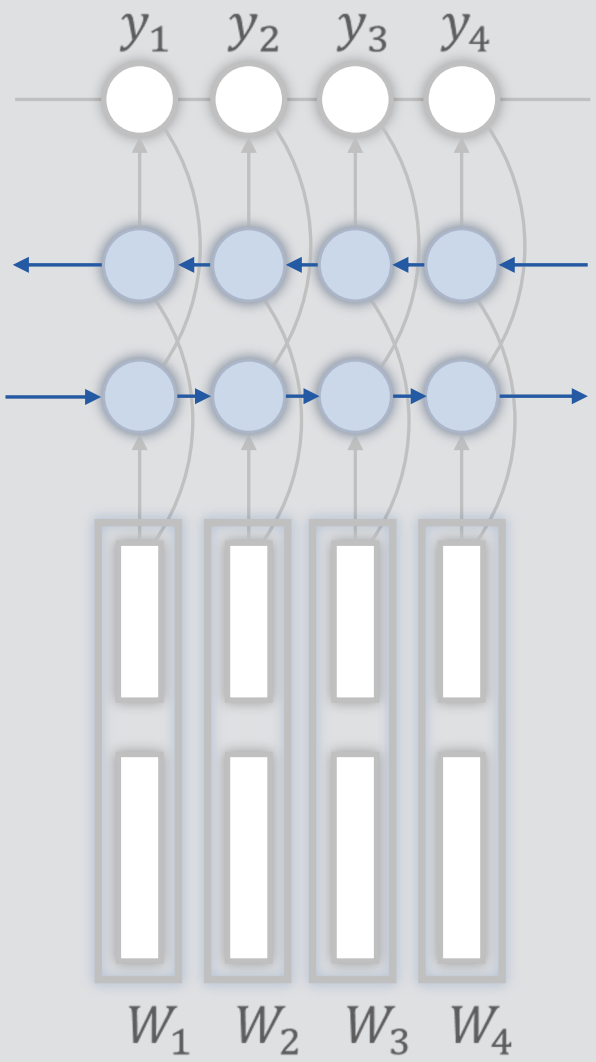
Прямая LSTM

Символьное представление

Вектора слов

CNN-biLSTM-CRF

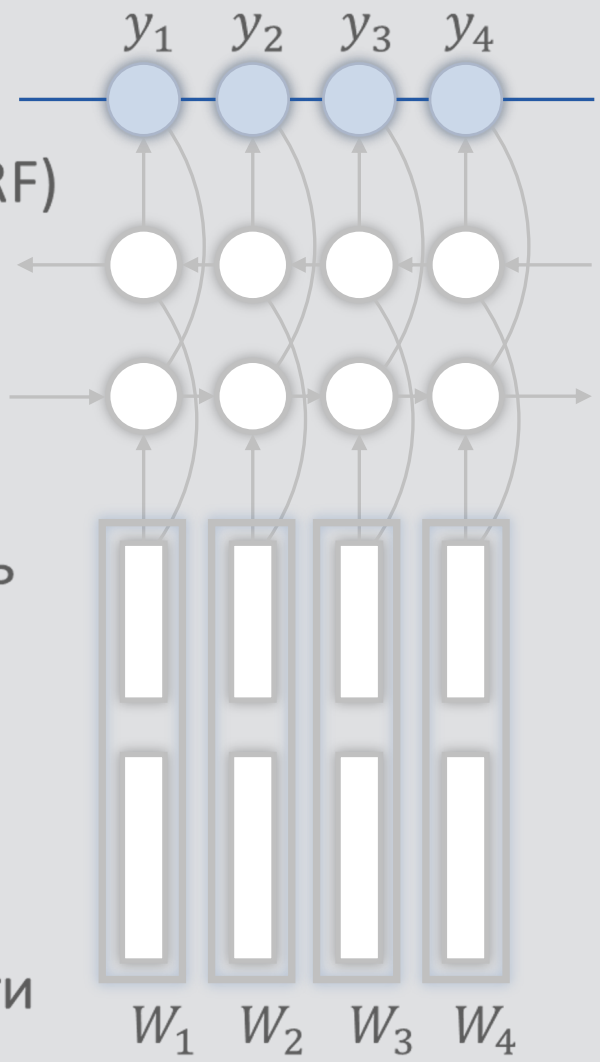
Двунаправленная LSTM (biLSTM) читает вектора слов и выдает контекстно-зависимые представления слов



CNN-biLSTM-CRF

Условное случайное поле (conditional random field, CRF)

- Получает на вход контекстно-зависимые вектора слов
- Определяет вероятность **последовательности** меток $\widehat{y^n}, \widehat{y_i} \in L$
- Помогает оценивать вероятность меток в контексте и в окрестности других меток



Меры качества

Стандартные меры качества (точность, полнота, f-мера) на уровне:

Меры качества

Стандартные меры качества (точность, полнота, f-мера) на уровне:

- Токенов – правильно ли определена метка этого токена

Меры качества

Стандартные меры качества (точность, полнота, f-мера) на уровне:

- Токенов – правильно ли определена метка этого токена
- Сущностей – правильно ли определена метка для всей сущности

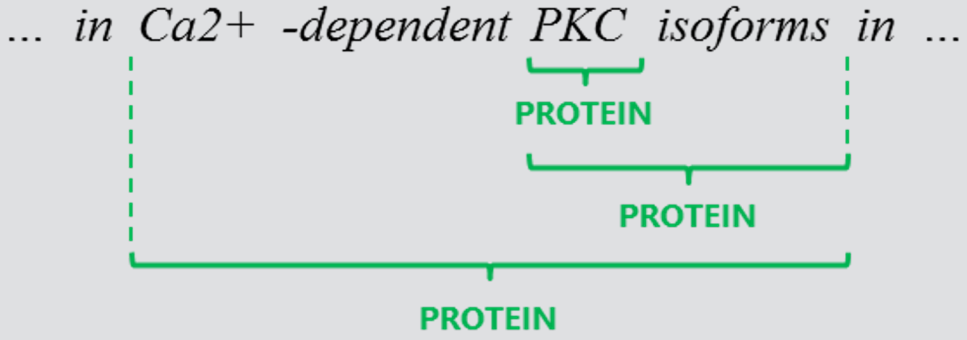
Приложения: извлекаем…

- Упоминание побочных действий лекарственных препаратов из пользовательских текстов
- Анатомические термины из истории болезней
- Юридические термины из исков и договоров
- Опыт и навыки из резюме

Вложенные именованные сущности

Вложенные именованные сущности

- На практике в сложных документах (юридических текстах, научных статьях, историях болезни) могут возникать вложенные сущности
- Существующие методы легко адаптировать к извлечению таких сущностей



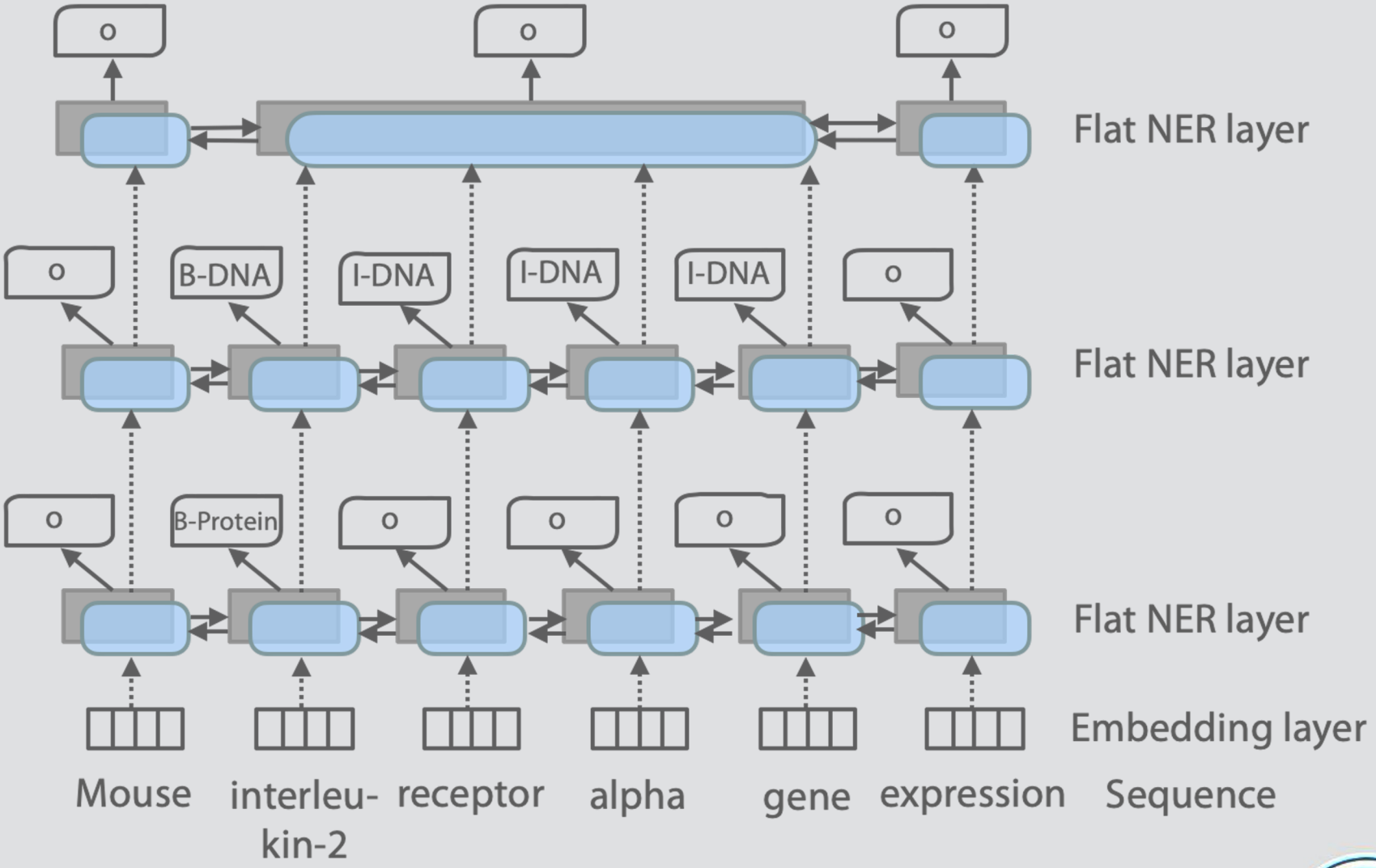
- *Shibuya, Takashi, and Eduard Hovy. "Nested Named Entity Recognition via Second-best Sequence Learning and Decoding." arXiv preprint arXiv:1909.02250 (2019).*

Вложенные именованные сущности

- Стек слоев: первый слой определяет первый уровень именованных сущностей, второй – следующий и т.д.
- Усредняем вектора слов, попавших в именованную сущность:

$$z_i' = \frac{1}{end - start + 1} \sum_{start}^{end} z_i$$

Вложенные именованные сущности



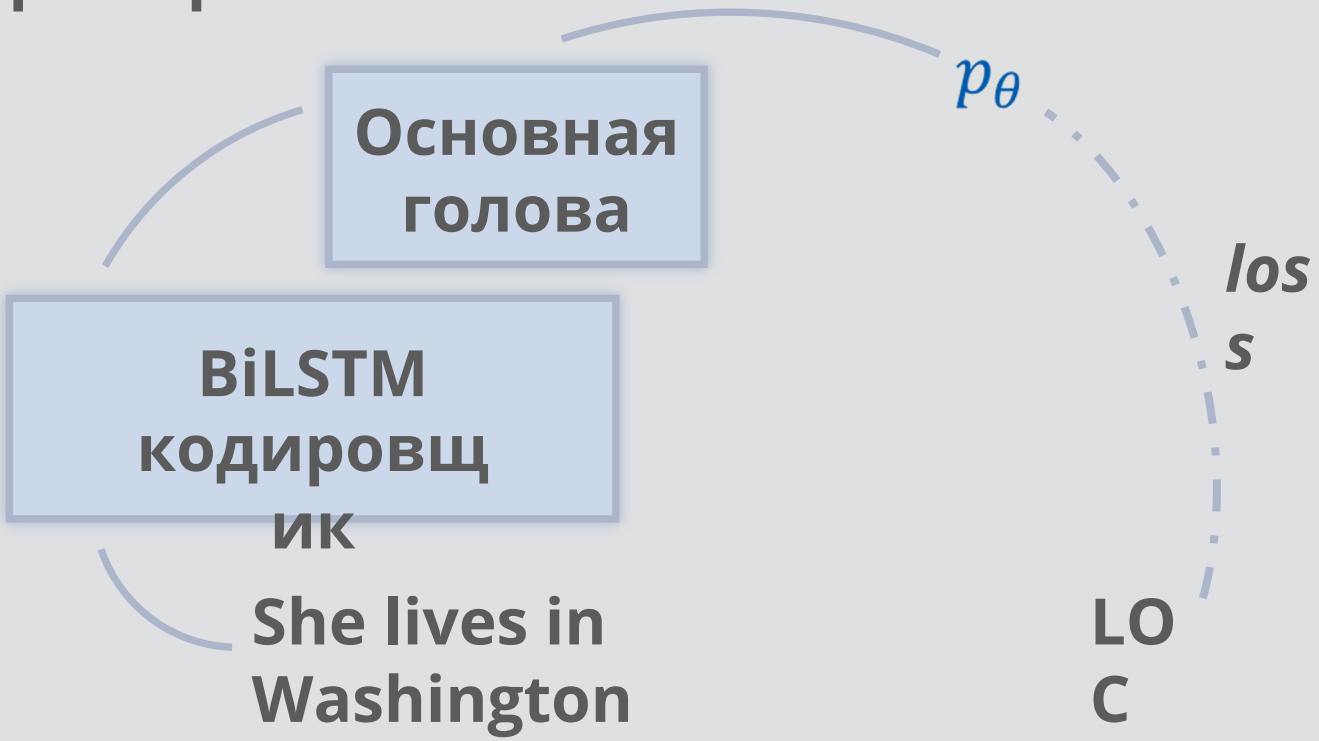
- *Ju, Meizhi, Makoto Miwa, and Sophia Ananiadou. "A neural layered model for nested named entity recognition." NAACL 2018*

Перекрестное обучение

- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

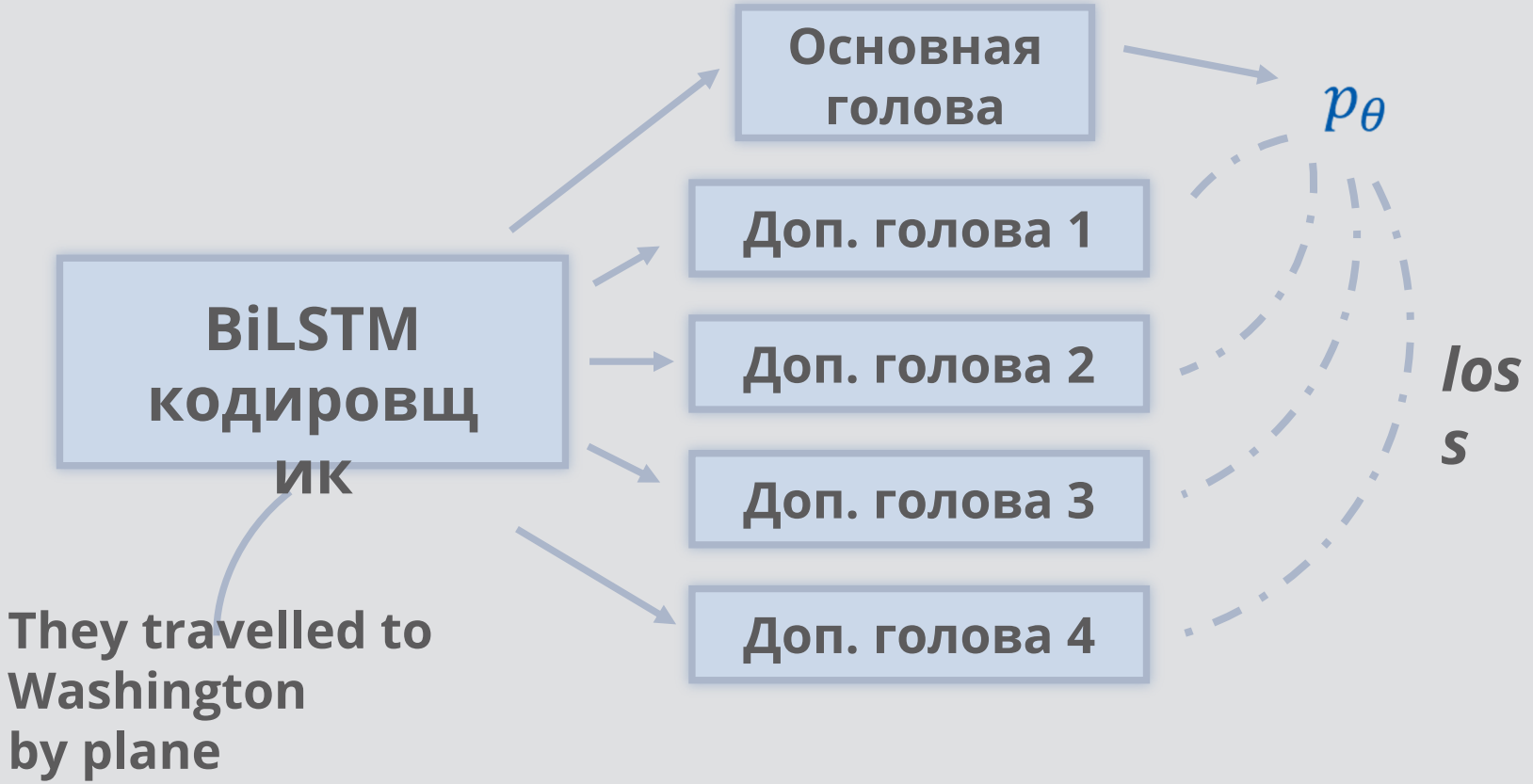
- Используем размеченные и неразмеченные данные одновременно
- Основная голова обучается по размеченным примерам



- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Используем размеченные и неразмеченные данные одновременно



- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

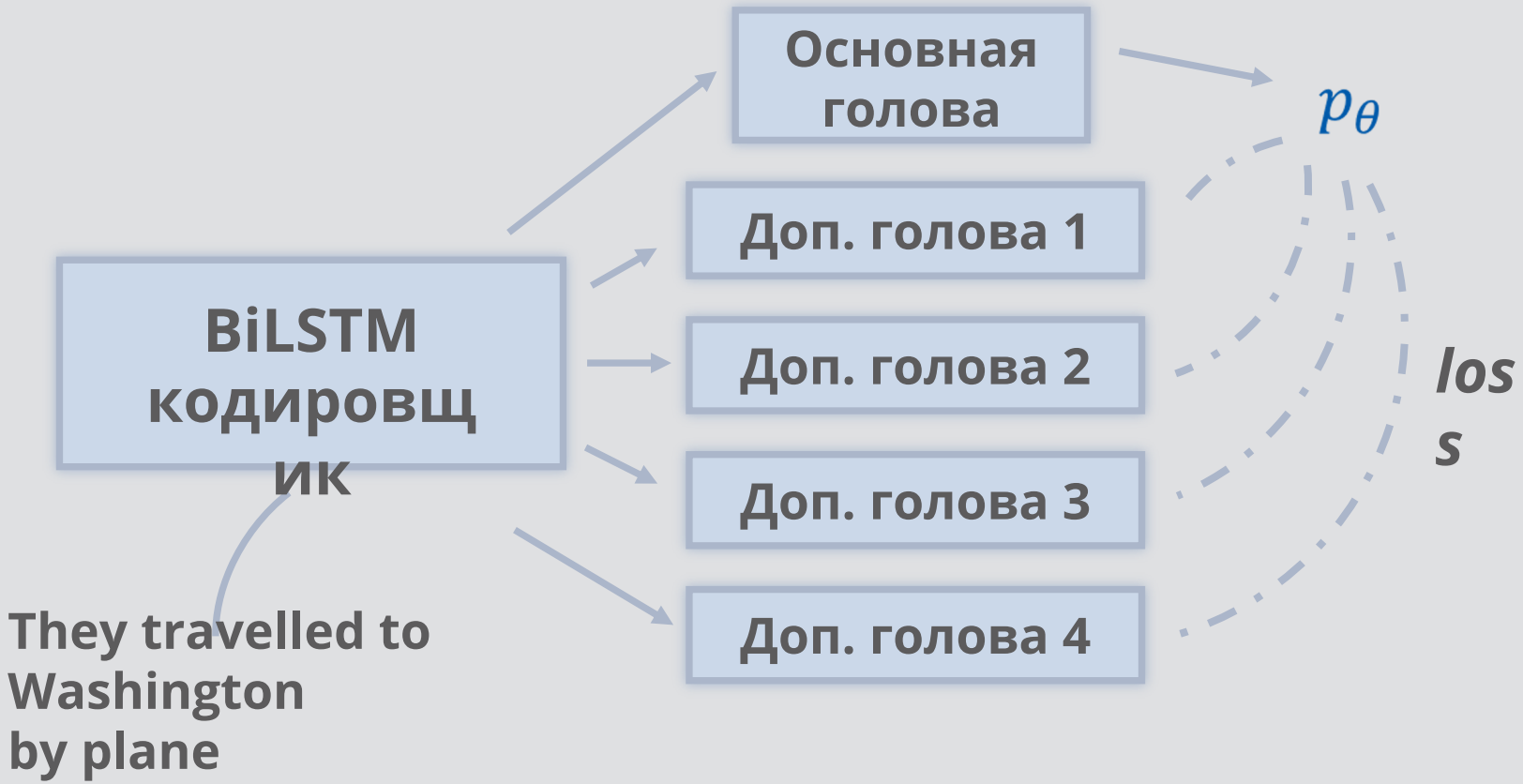
Перекрестное обучение

- Дополнительные головы обучаются по неразмеченным примерам воспроизводить предсказания основной головы

- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Основная голова учит дополнительные



- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Дополнительные головы видят только части примера

They travelled to Washington by plane

[future] They travelled to _____

[past] _____ by plane

[fwd] They travelled to Washington _____

[bwd] _____ Washington by plane

- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Дополнительные головы видят только части примера

Почти языковая модель!

They travelled to Washington by plane

[future]	They travelled to _____
[past]	_____ by plane
[fwd]	They travelled to Washington _____
[bwd]	_____ Washington by plane

- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Дополнительные головы видят только части примера

Предсказываем класс следующего слова, а не само слово

They travelled to Washington by plane

[future]	They travelled to _____
[past]	_____ by plane
[fwd]	They travelled to Washington _____
[bwd]	_____ Washington by plane

- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

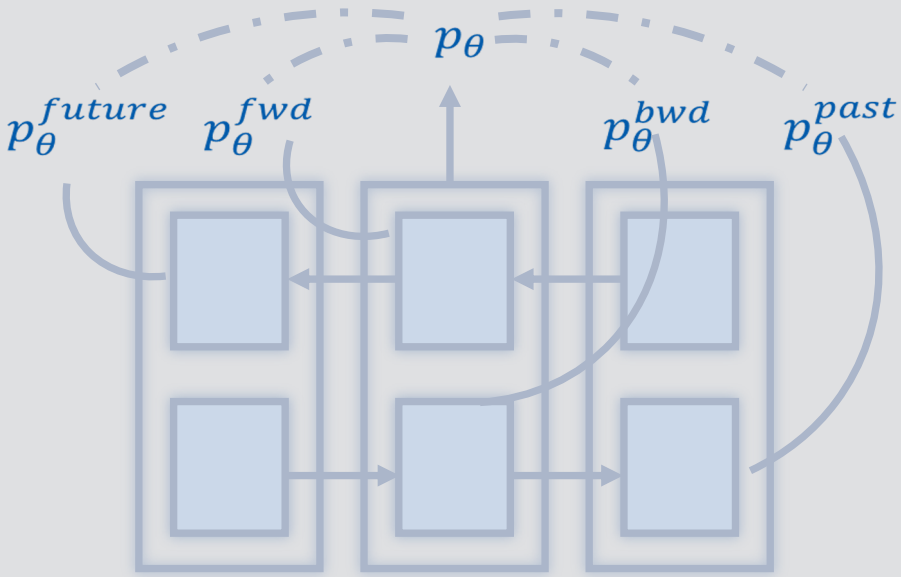
Перекрестное обучение

- Кодировщик состоит из двух BiLSTM слоев
- Основная голова имеет доступ к обоим BiLSTM слоям

- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Дополнительные головы видят только первый BiLSTM слой
- Второй слой знает контекст, поэтому дополнительные головы ограничены в доступе к нему

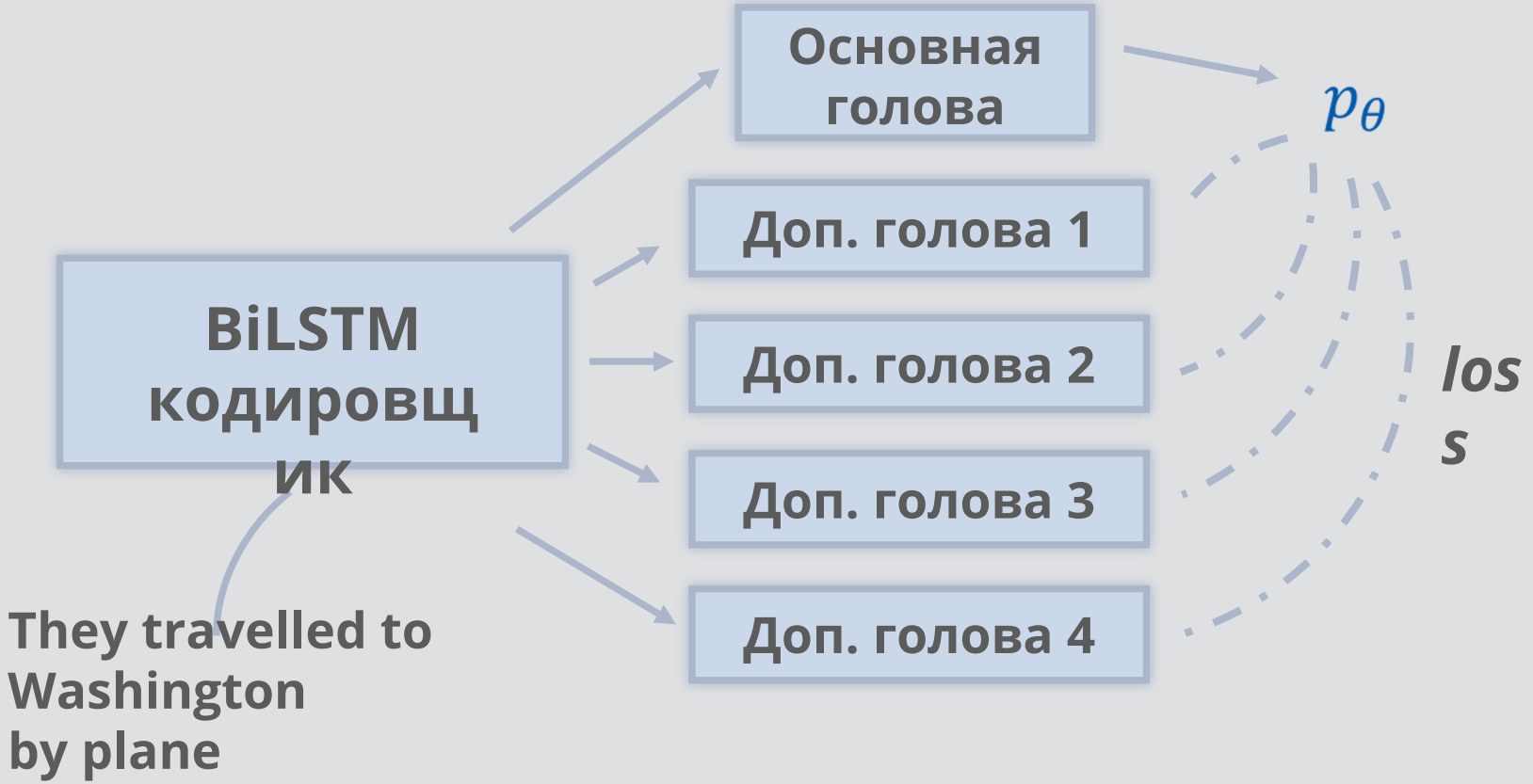


- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Комбинация обучения с учителем и самообучения:

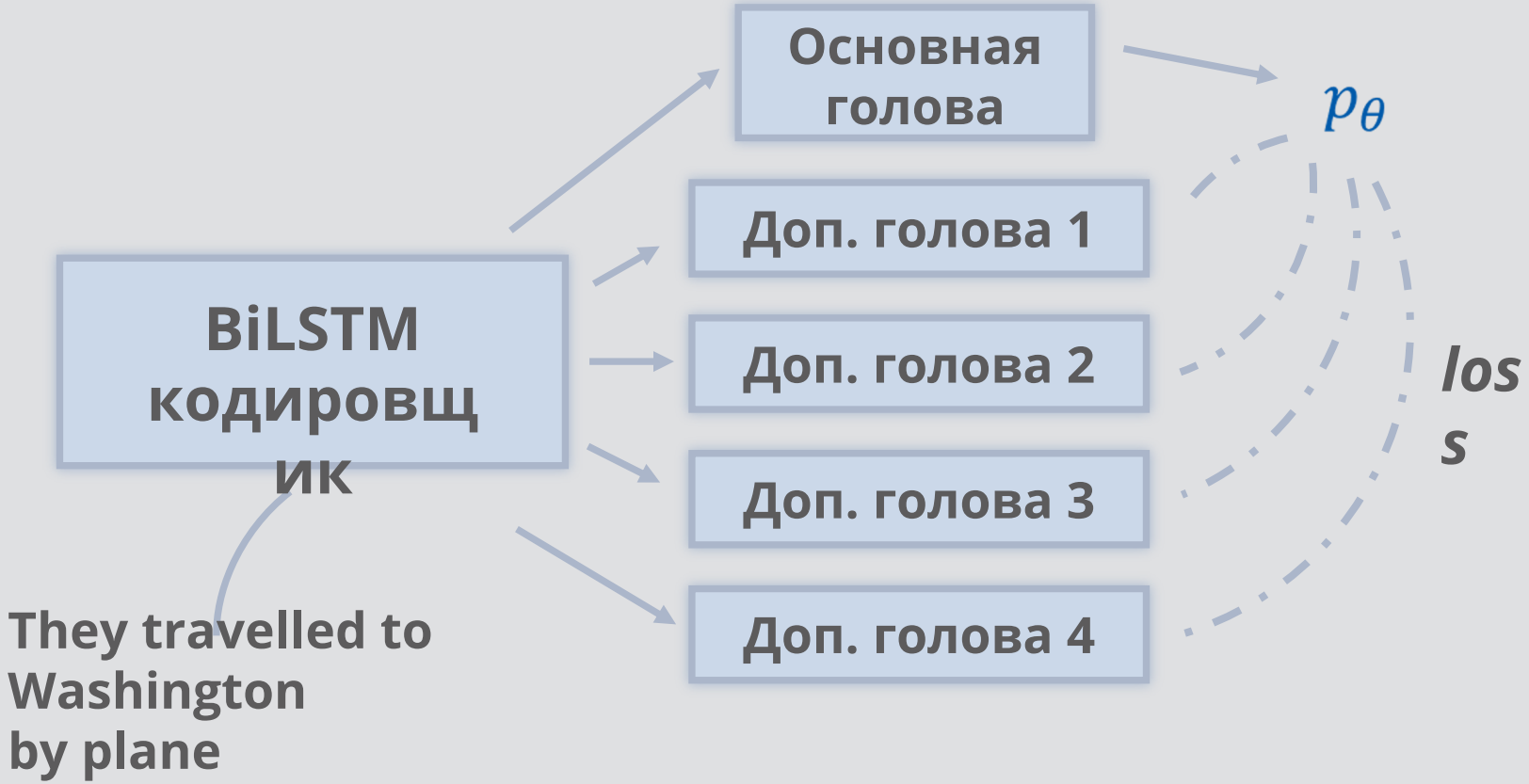
$$\mathcal{L} = \mathcal{L}_{sup} + \mathcal{L}_{CVT}$$



- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Хорошо работает на маленьких наборах данных

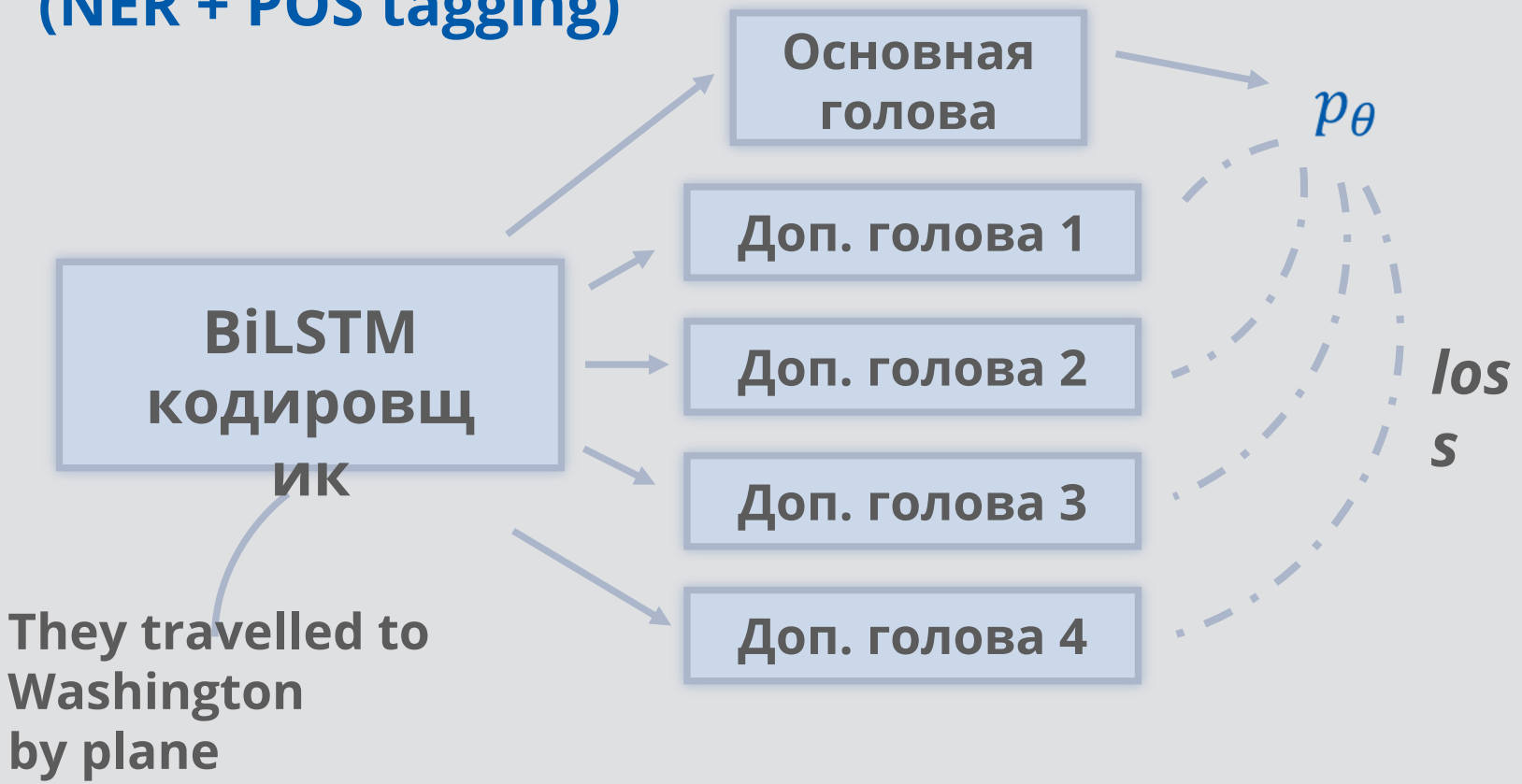


- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Перекрестное обучение

- Может решать одновременно несколько задач

(NER + POS tagging)



- *Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018*

Итоги недели

- Узнали, что такое языковая модель

Итоги недели

- Узнали, что такое языковая модель
- Познакомились с двумя новыми задачами:

Итоги недели

- Узнали, что такое языковая модель
- Познакомились с двумя новыми задачами:
  - Морфологический анализ

Итоги недели

- Узнали, что такое языковая модель
- Познакомились с двумя новыми задачами:
  - Морфологический анализ
  - Извлечение именованных сущностей

Итоги недели

- Узнали, что такое языковая модель
- Познакомились с двумя новыми задачами:
  - Морфологический анализ
  - Извлечение именованных сущностей
- Обсудили, как использовать рекуррентные нейронные сети для моделирования последовательностей