

*Лекция 4*

# **Графические модели для работы с последовательностями слов**

*Потапенко Анна Александровна*

*12 сентября 2018*

# **1. Задачи разметки последовательностей (Sequence labeling/tagging)**

# Примеры задач

*Распознавание частей речи:*

PRON

I

VERB

saw

DET

a

PROPN

Heffalump

NOUN

today.

*Распознавание именованных сущностей:*

Once upon a time, a very long time ago now, about

**[last Friday]**, **[Winnie-the-Pooh]** lived in a forest all by

himself under the name of **[Sanders]**.

# Разметка последовательностей

**Дано:** последовательность слов (токенов)

**Найти:** последовательность меток (тэгов)

## Примеры задач:

- распознавание частей речи (part of speech tagging, POS)
- распознавание именованных сущностей (named entity recognition, NER)
- выделение семантических ролей (semantic role labeling)
- снятие омонимии слов (word sense disambiguation, WSD)
- неглубокий синтаксический разбор (chunking, shallow syntax parsing)

# POS-тэги из Universal Dependencies

Open class words	
ADJ	adjective
ADV	adverb
INTJ	interjection
NOUN	noun
PROPN	proper noun
VERB	verb

Other	
PUNCT	punctuation
SYM	symbol
X	other

Closed class words	
ADP	adposition
AUX	auxiliary verb
CCONJ	coordinating conjunction
DET	determiner
NUM	numeral
PART	particle
PRON	pronoun
SCONJ	subordinating conjunction

<http://universaldependencies.org/>

# ВІО-нотация (beginning - inside - outside)

*Что может быть именованной сущностью:*

люди, организации, места, дата, время, количество, ...

*Набор NER тэгов в CoNLL-2003: B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC, O*

*Пример определения семантических ролей:*

<b>B_ACT</b>	<b>I_ACT</b>	<b>I_ACT</b>	<b>O</b>	<b>B_NUM_PER</b>	<b>O</b>	<b>B_LOC</b>	<b>I_LOC</b>
<b>Book</b>	<b>a</b>	<b>table</b>	<b>for</b>	<b>3</b>	<b>in</b>	<b>Domino's</b>	<b>pizza</b>

# Возможные подходы

1. Правилковые модели (example: EngCG tagger)
2. Одельные классификаторы для каждого токена
- 3. Графически модели (HMM, MEMM, CRF)**
4. Нейронные сети (*следующая лекция*)

## **2. Скрытая марковская модель (Hidden Markov Model)**



# Скрытая марковская модель

## Обозначения:

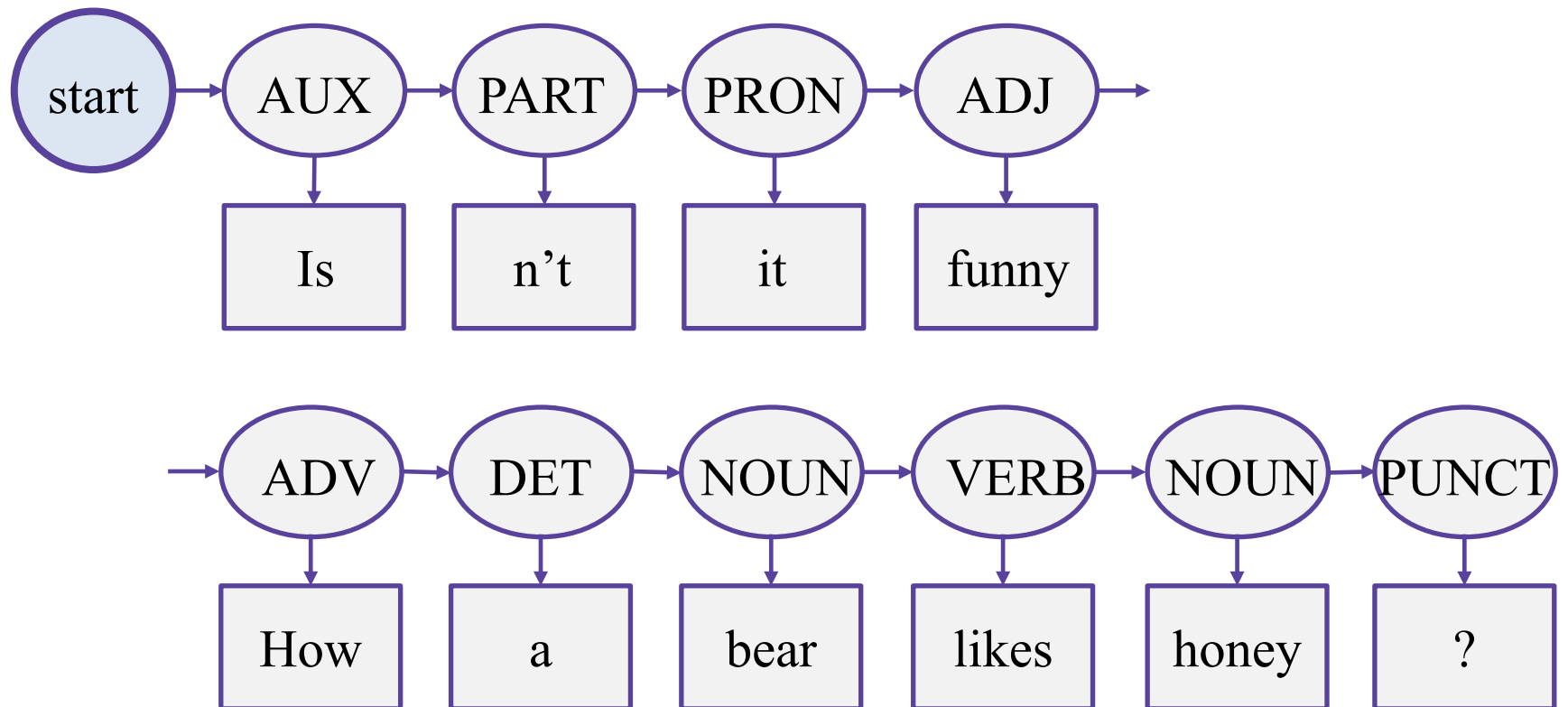
$\mathbf{X} = x_1, \dots, x_T$     последовательность слов

$\mathbf{Y} = y_1, \dots, y_T$     последовательность меток

## Генерация текста:

1. Генерируется следующая метка при условии предыдущей
2. Генерируется слово при условии метки

# Генерация текста: пример



Каждая стрелочка – это условная вероятность.

# Скрытая марковская модель

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y}) p(\mathbf{y}) = \prod_{t=1}^T p(x_t|y_t) p(y_t|y_{t-1})$$



наблюдаемые

скрытые

Предположение Маркова:

$$p(\mathbf{y}) \approx \prod_{t=1}^T p(y_t|y_{t-1})$$

Предположение о независимости:

$$p(\mathbf{x}|\mathbf{y}) \approx \prod_{t=1}^T p(x_t|y_t)$$

# Формальное определение модели

1. Множество  $S = s_1, s_2, \dots, s_N$  скрытых состояний
2. Начальное состояние  $s_0$
3. Матрица  $A$  вероятностей переходов  $a_{ij} = p(s_j | s_i)$
4. Множество  $O$  выходных токенов
5. Матрица  $B$  выходных вероятностей  $b_{kj} = p(o_k | s_j)$

В нашем случае, состояния – это метки, токены – это слова:

$$x_t \in O, \quad y_t \in S$$

# Задачи и алгоритмы

## 1. Обучение модели:

$$A = ? \quad B = ?$$

*(по размеченной или по неразмеченной выборке)*

## 2. Оценивание апостериорных вероятностей:

$$p(\mathbf{y}|\mathbf{x}) = ?$$

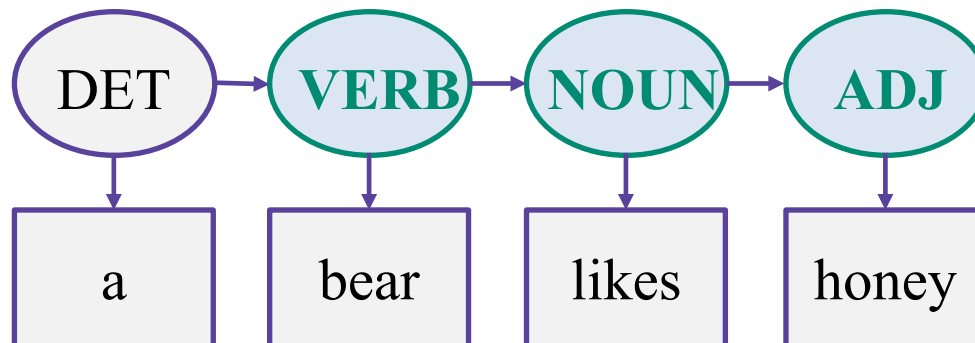
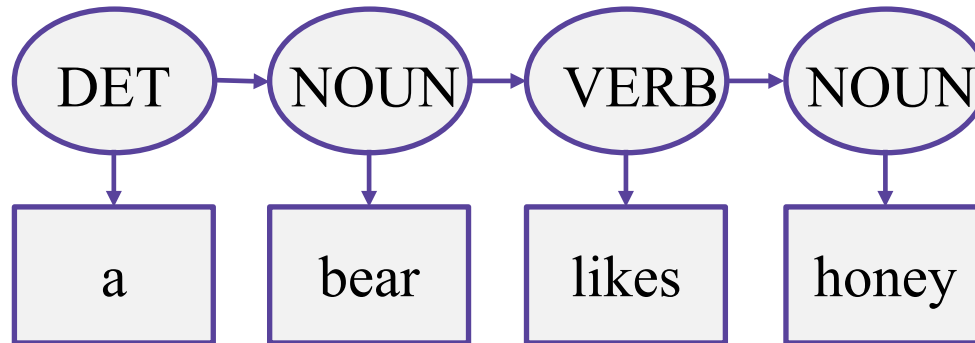
## 3. Поиск наиболее вероятных меток:

$$\mathbf{y} = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$$

### **3. Алгоритм Витерби: поиск наиболее вероятных меток**


# Мотивация

Одина и та же последовательность слов может быть получена из нескольких последовательностей (скрытых) меток:



# Задача декодирования

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^T p(y_t | y_{t-1}) p(x_t | y_t)$$

  
Вероятности перехода      Выходные вероятности

**Найти:**

Наиболее вероятную последовательность скрытых меток

$$\mathbf{y} = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y} | \mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$$

Решается динамическим программированием.

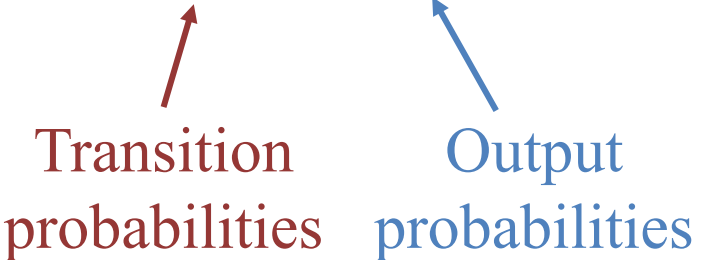


# Алгоритм Витерби

Пусть  $Q_{t,s}$  - самая вероятная последовательность скрытых состояний длины  $t$  с окончанием в состоянии  $S$   
а  $q_{t,s}$  - вероятность этой последовательности.

Тогда  $q_{t,s}$  можно подсчитать динамически:

$$q_{t,s} = \max_{s'} q_{t-1,s'} \cdot p(s|s') \cdot p(o_t|s)$$

  
Transition probabilities      Output probabilities

$Q_{t,s}$  можно восстановить по argmax-ам.

# Алгоритм Витерби

**Input:** observations of length  $T$ , state-graph of length  $N$

**Output:** best-path

for each state  $s$  from 1 to  $N$  do

$$q[1, s] \leftarrow p(s|s_0) \cdot p(o_1|s)$$

$$\text{backpointers}[1, s] \leftarrow 0$$

for each time step  $t$  from 2 to  $T$  do

for each state  $s$  from 1 to  $N$  do

$$q[t, s] \leftarrow \max_{s'=1}^N q[t-1, s'] \cdot p(s|s') \cdot p(o_t|s)$$

$$\text{backpointers}[t, s] \leftarrow \operatorname{argmax}_{s'=1}^N q[t-1, s'] \cdot p(s|s')$$

$$s \leftarrow \operatorname{argmax}_{s'=1}^N q[T, s']$$

return the backtrace path from backpointers  $[T, s]$

# Пример: вероятности переходов

Рассматриваются следующие POS-теги:  
ADJ, NOUN, VERB.

Пусть в начальный момент вероятности всех тегов равны  $1/3$ . Затем:

from \ to	<b>ADJ</b>	<b>NOUN</b>	<b>VERB</b>
<b>ADJ</b>	0.4	0.4	0.2
<b>NOUN</b>	0.2	0.4	0.4
<b>VERB</b>	0.1	0.6	0.3

Сумма вероятностей в каждой строке равна 1.

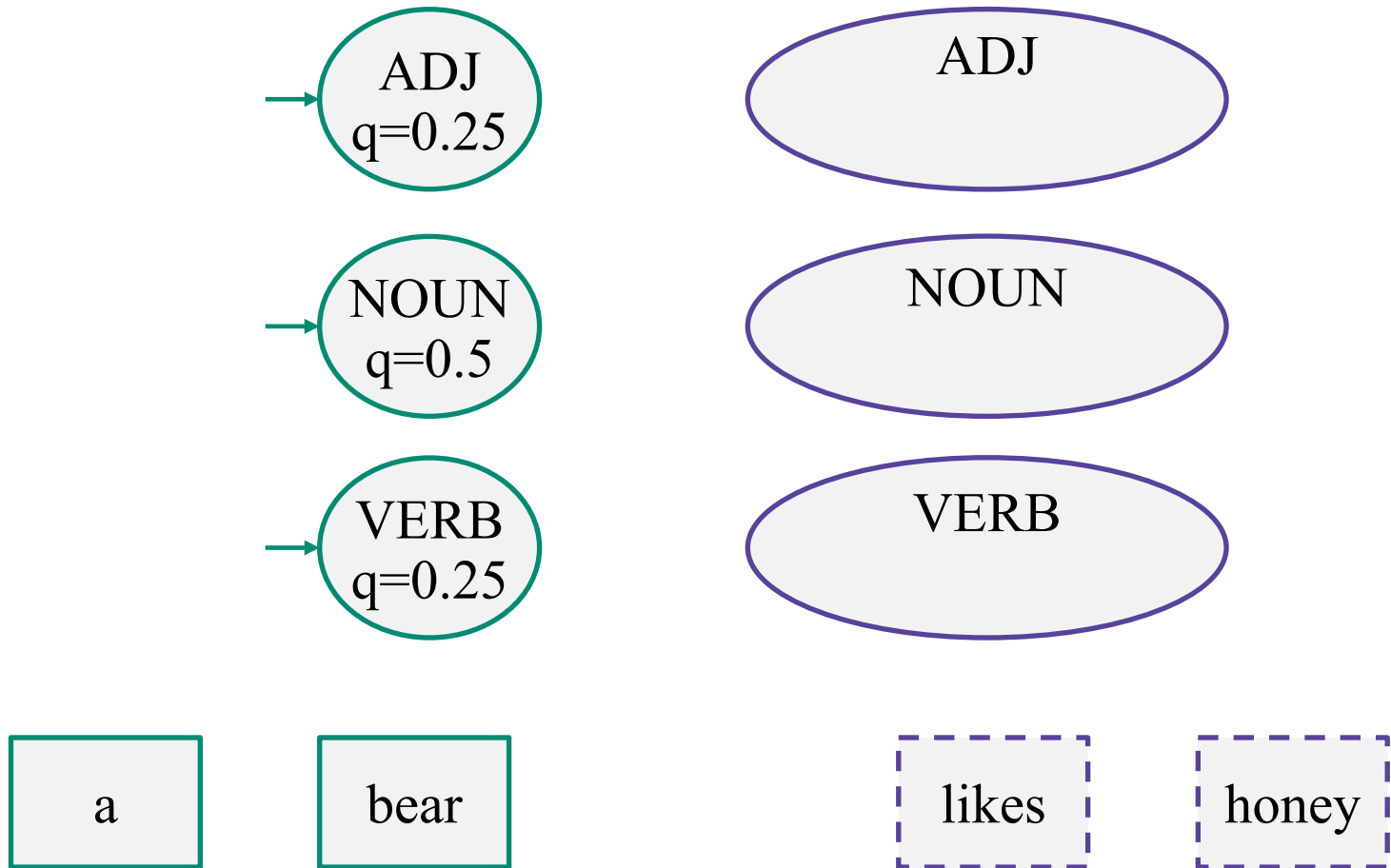
# Пример: выходные вероятности

Рассмотрим такие вероятности:

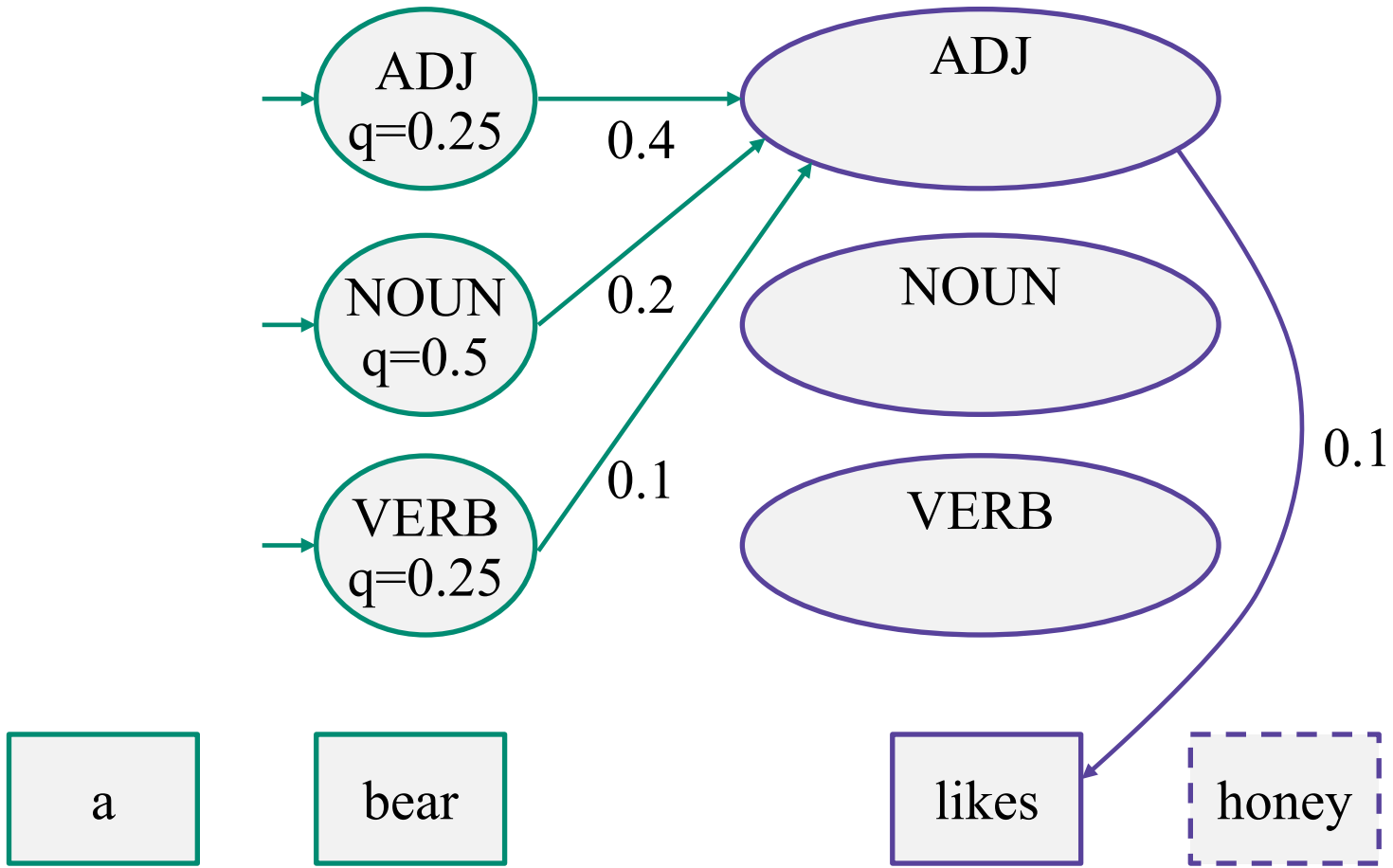
tag\word	<b>a</b>	<b>bear</b>	<b>fly</b>	<b>honey</b>	<b>likes</b>	<b>sweet</b>
<b>ADJ</b>	0.2	0.1	0.1	0.1	0.1	0.4
<b>NOUN</b>	0.1	0.2	0.2	0.2	0.2	0.1
<b>VERB</b>	0.1	0.2	0.2	0.1	0.3	0.1

Вероятности в каждой строке суммируются в 1.

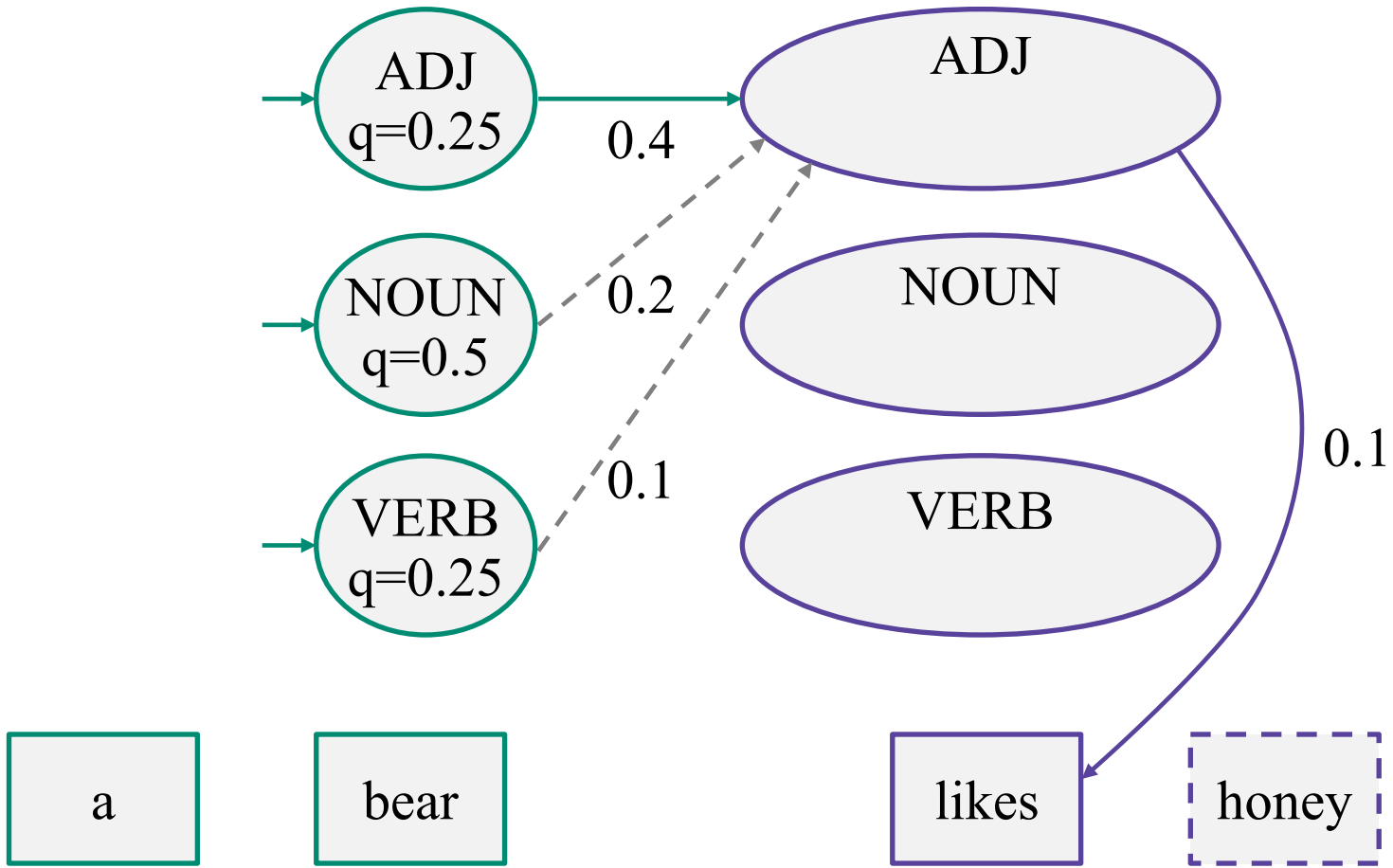
# Вероятности скрытых состояний до likes



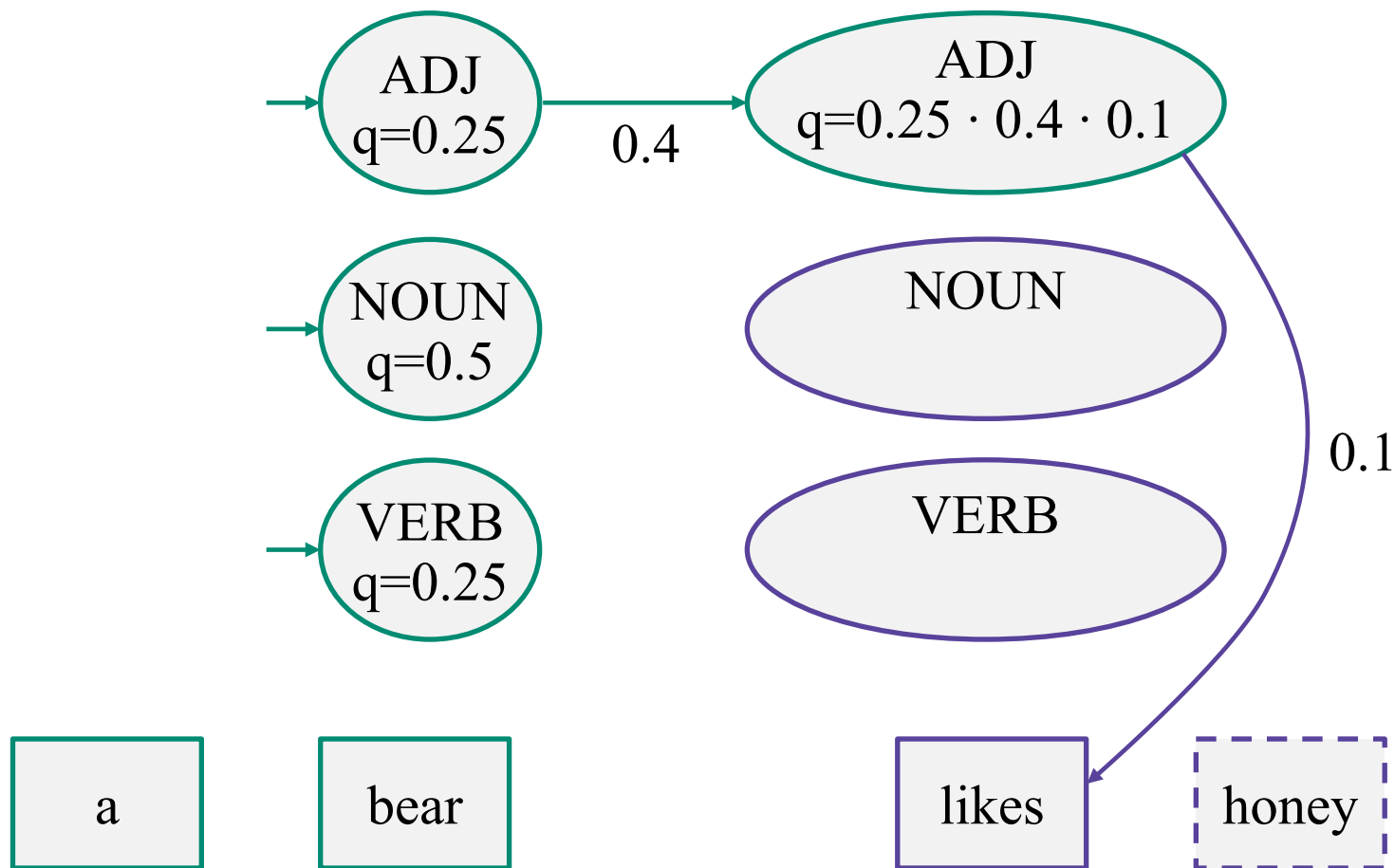
# Возможные переходы в состояние ADJ



# Лучший переход в состоянии ADJ

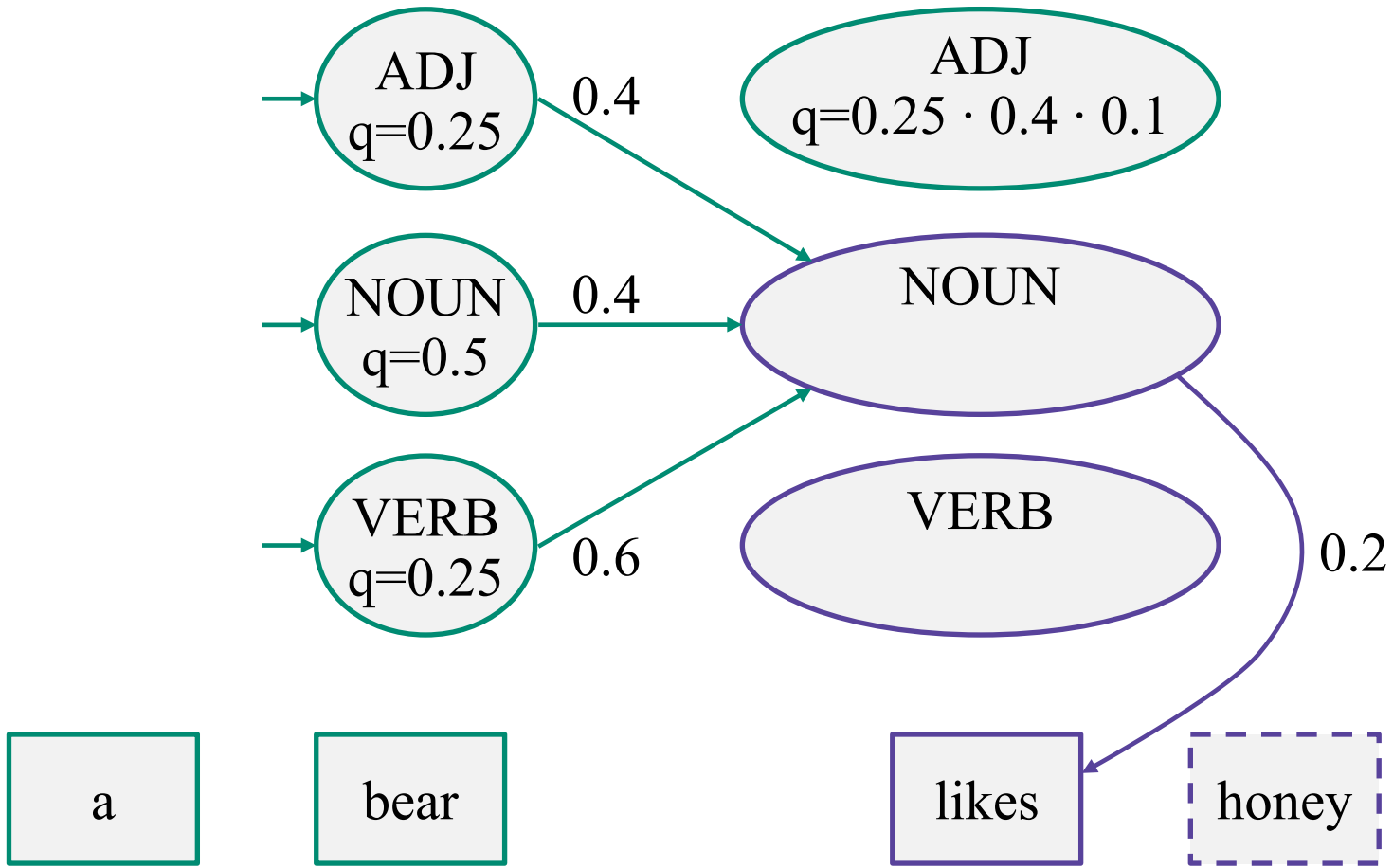


# Вероятность ADJ при условии likes

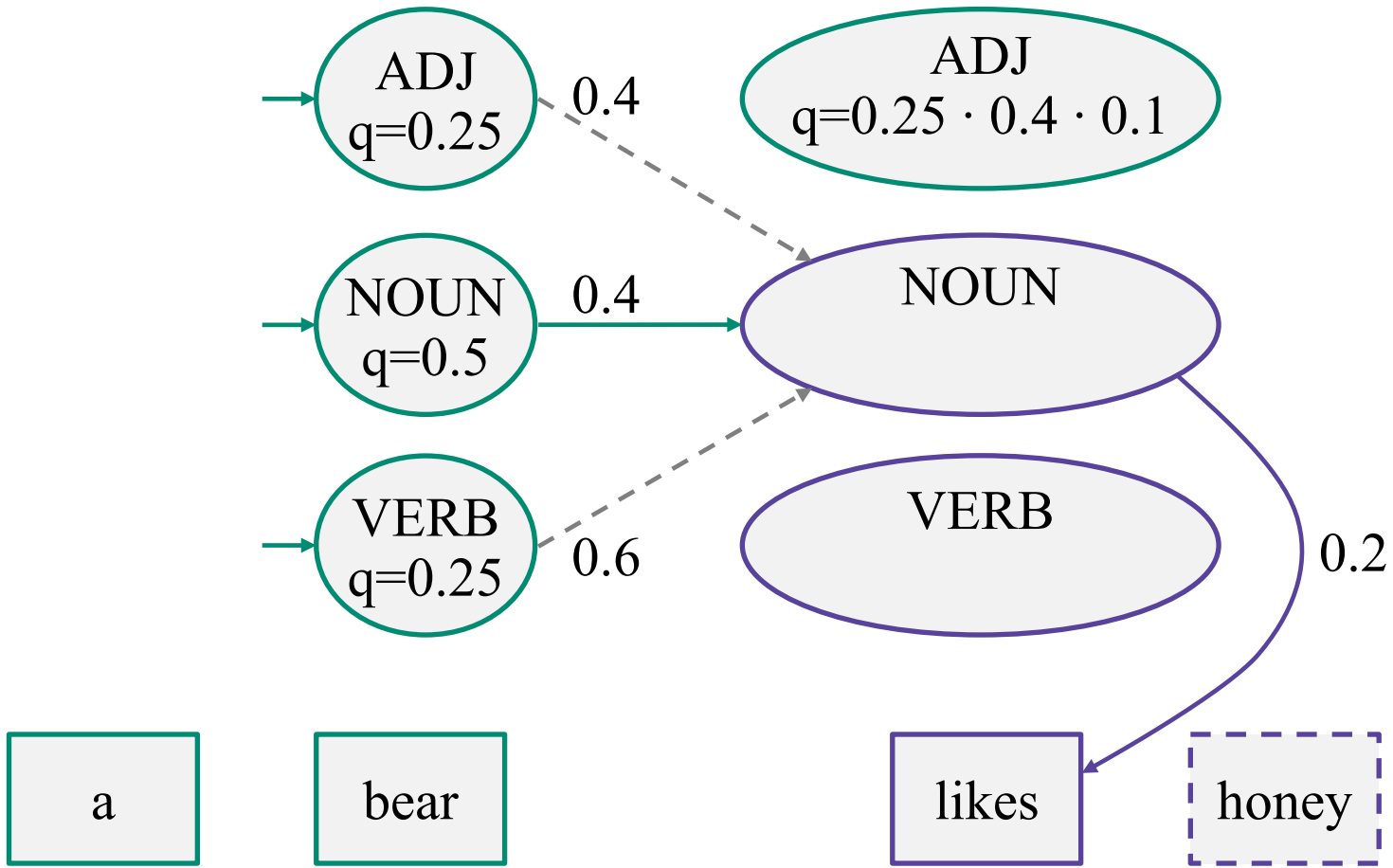




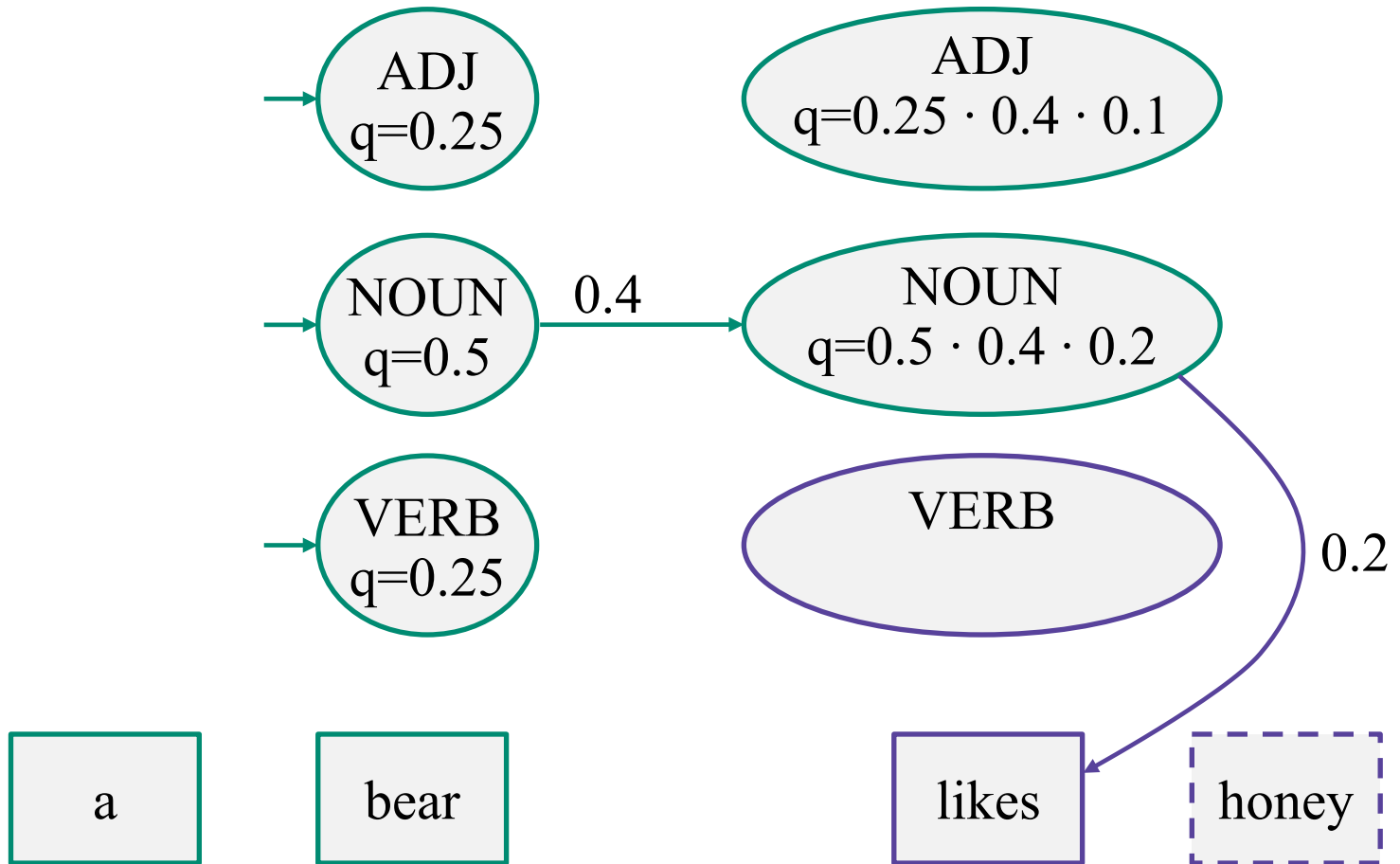
# Возможные переходы в состояние NOUN



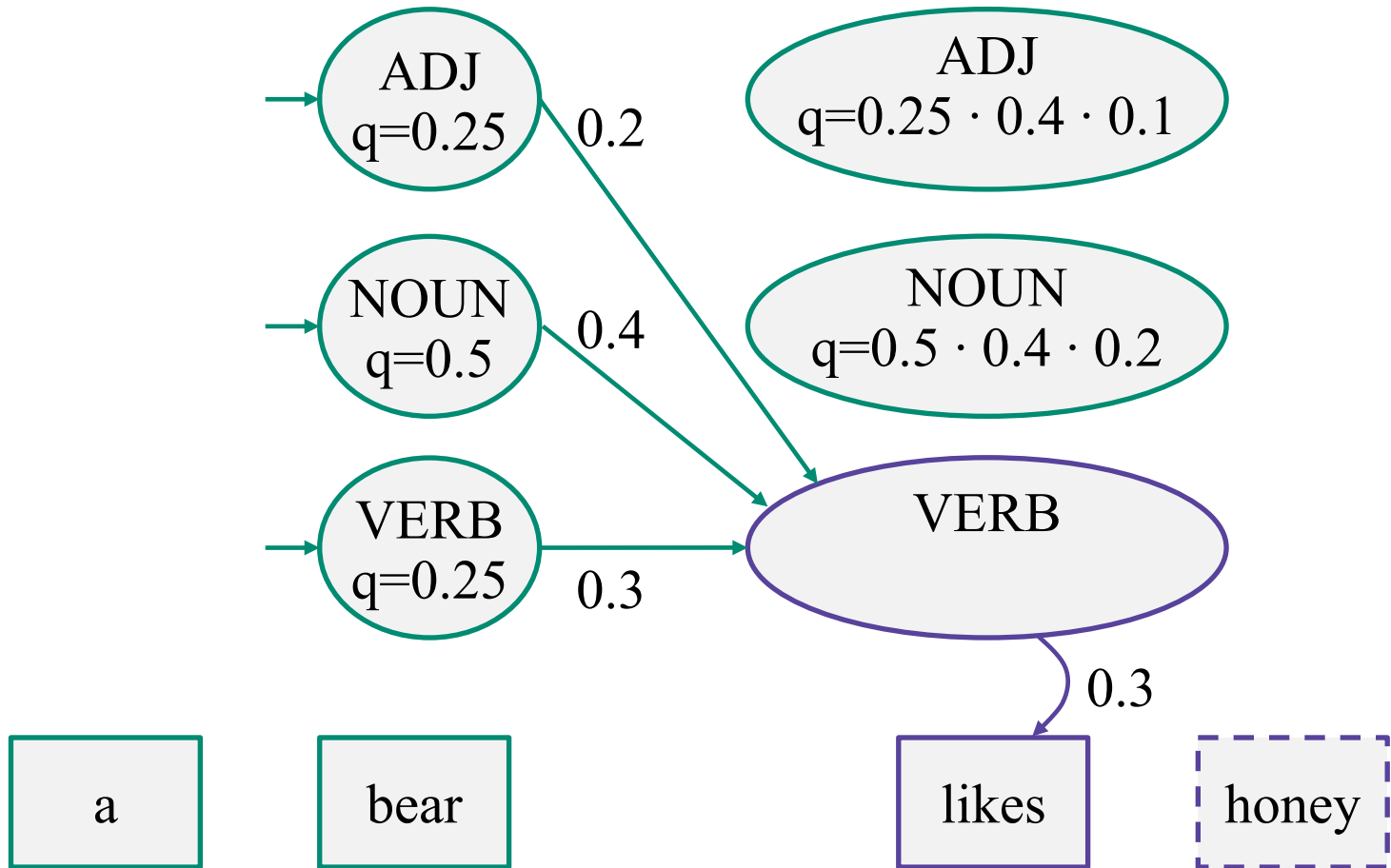
# Лучший переход в состояние NOUN



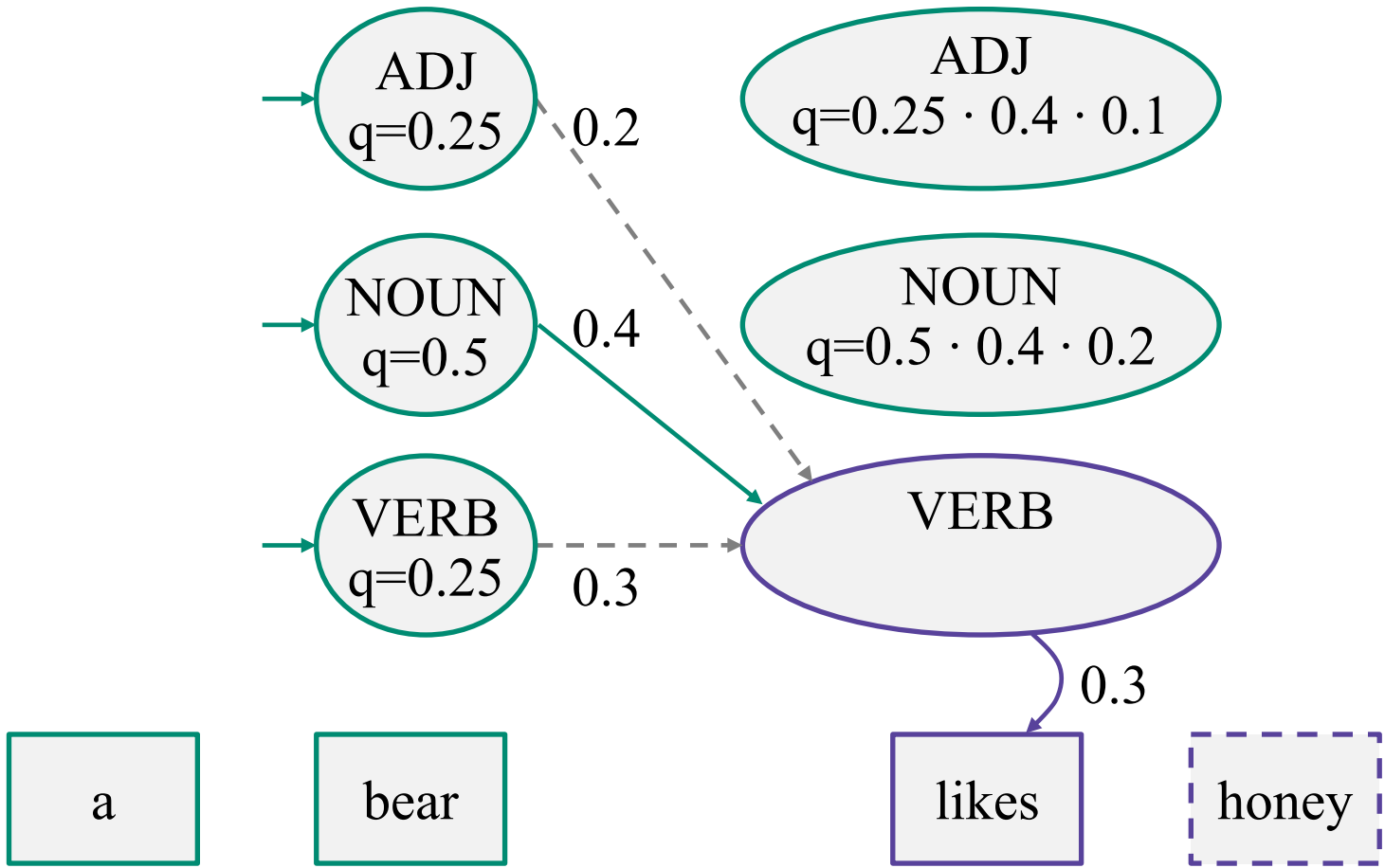
# Вероятность NOUN при условии likes



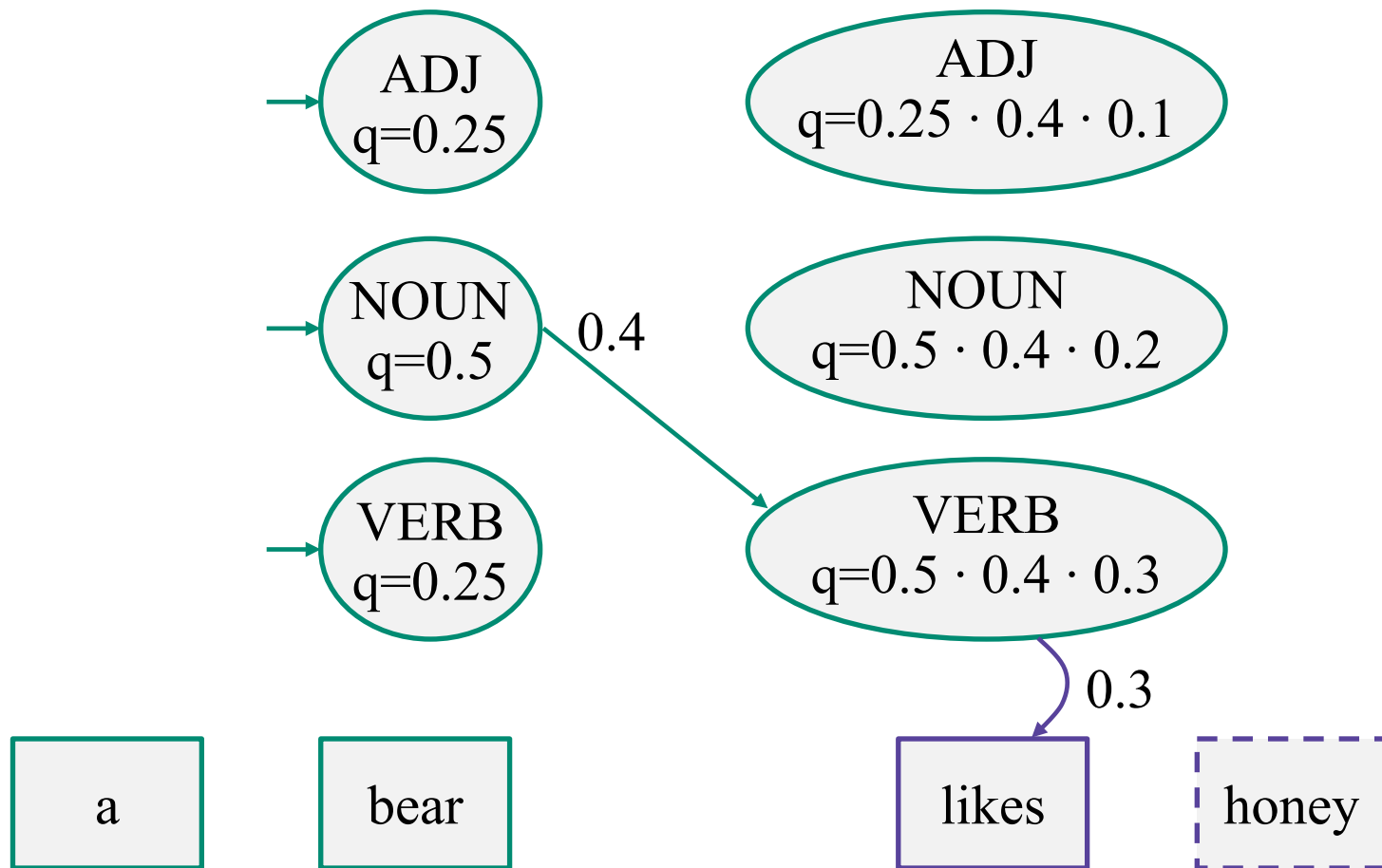
# Варианты переходов в состояние VERB



# Лучший переход в состояние VERB



# Вероятность VERB после likes



# Вероятности скрытых состояний после likes

→ ADJ  
 $q=0.25$

ADJ  
 $q=0.25 \cdot 0.4 \cdot 0.1$

→ NOUN  
 $q=0.5$

NOUN  
 $q=0.5 \cdot 0.4 \cdot 0.2$

→ VERB  
 $q=0.25$

VERB  
 $q=0.5 \cdot 0.4 \cdot 0.3$

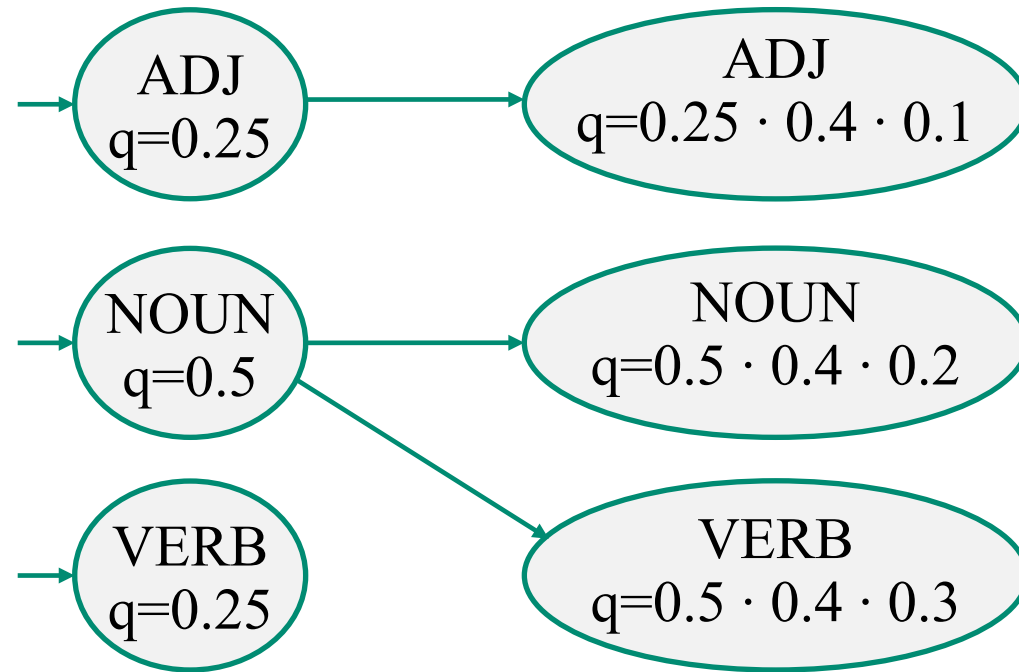
a

bear

likes

honey

# Запоминаем лучшие переходы!



a

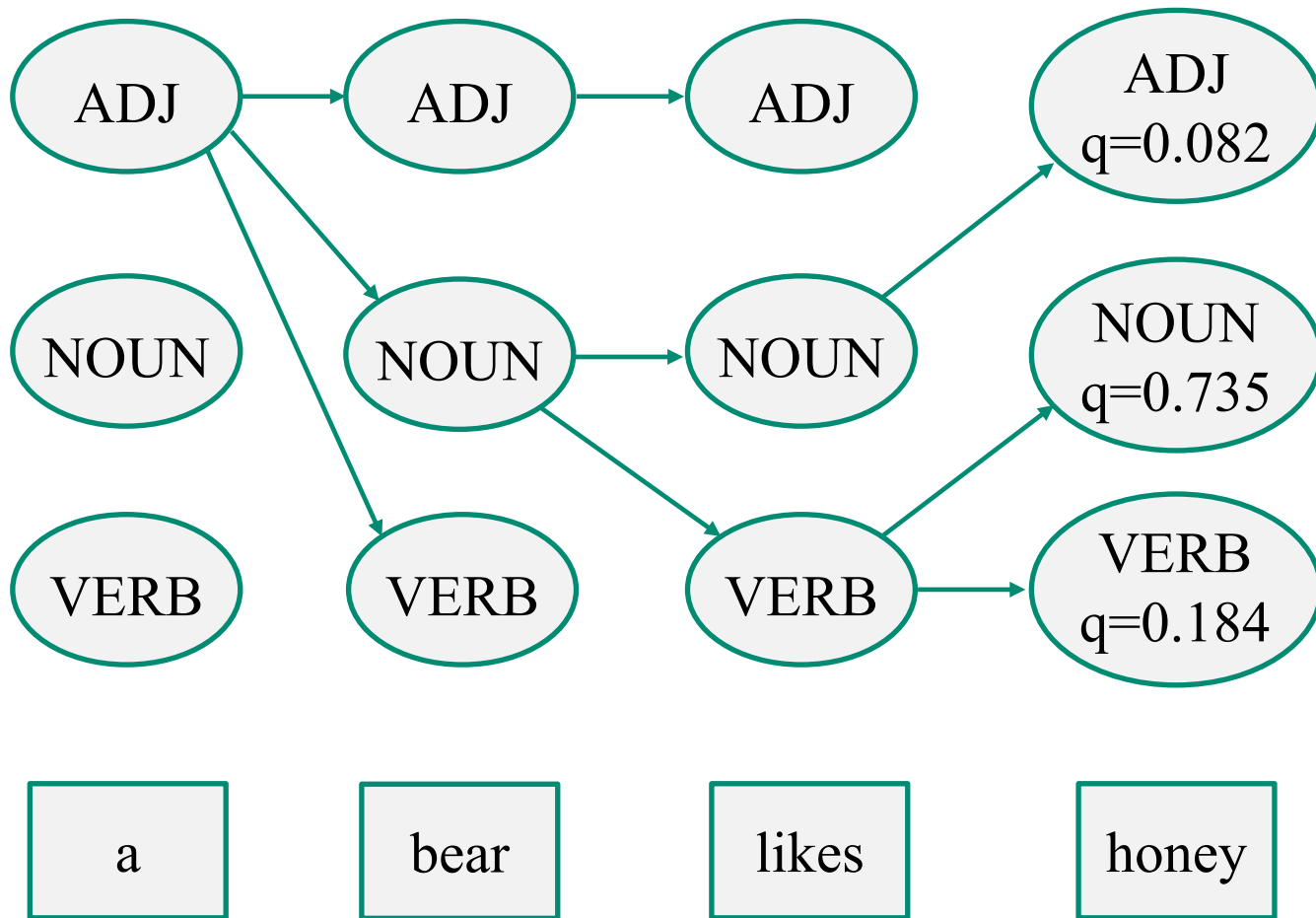
bear

likes

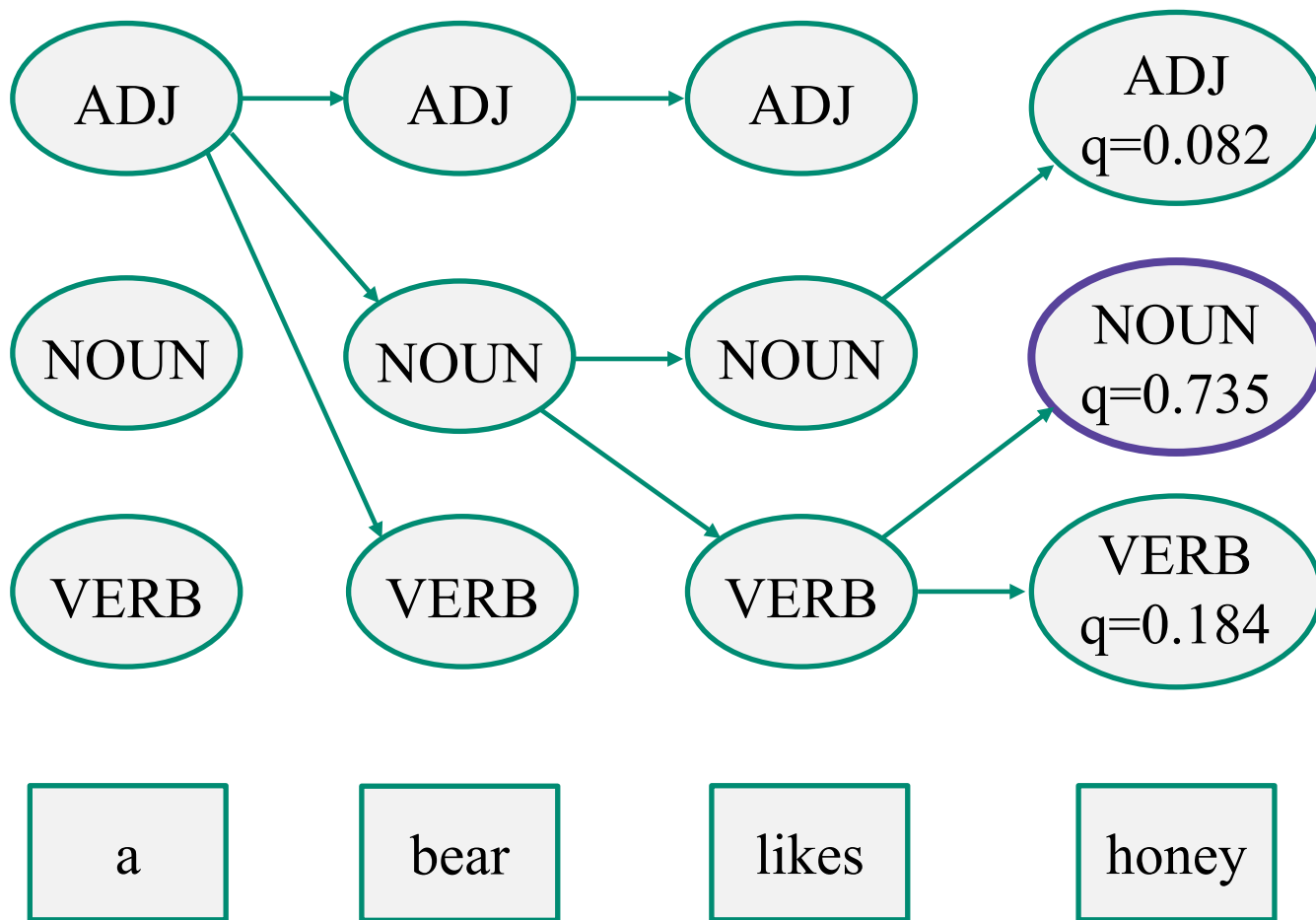
honey



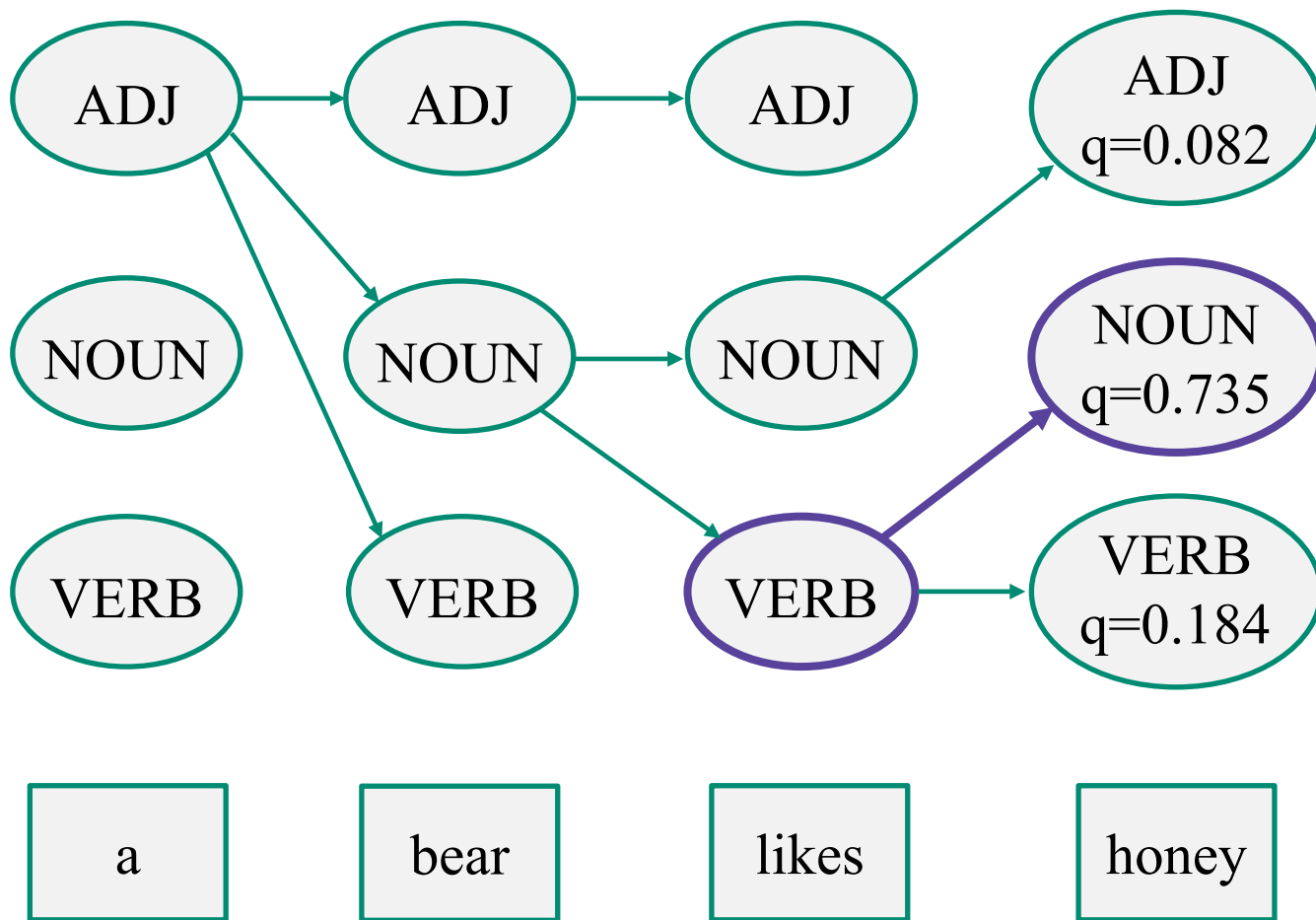
# Восстановление пути



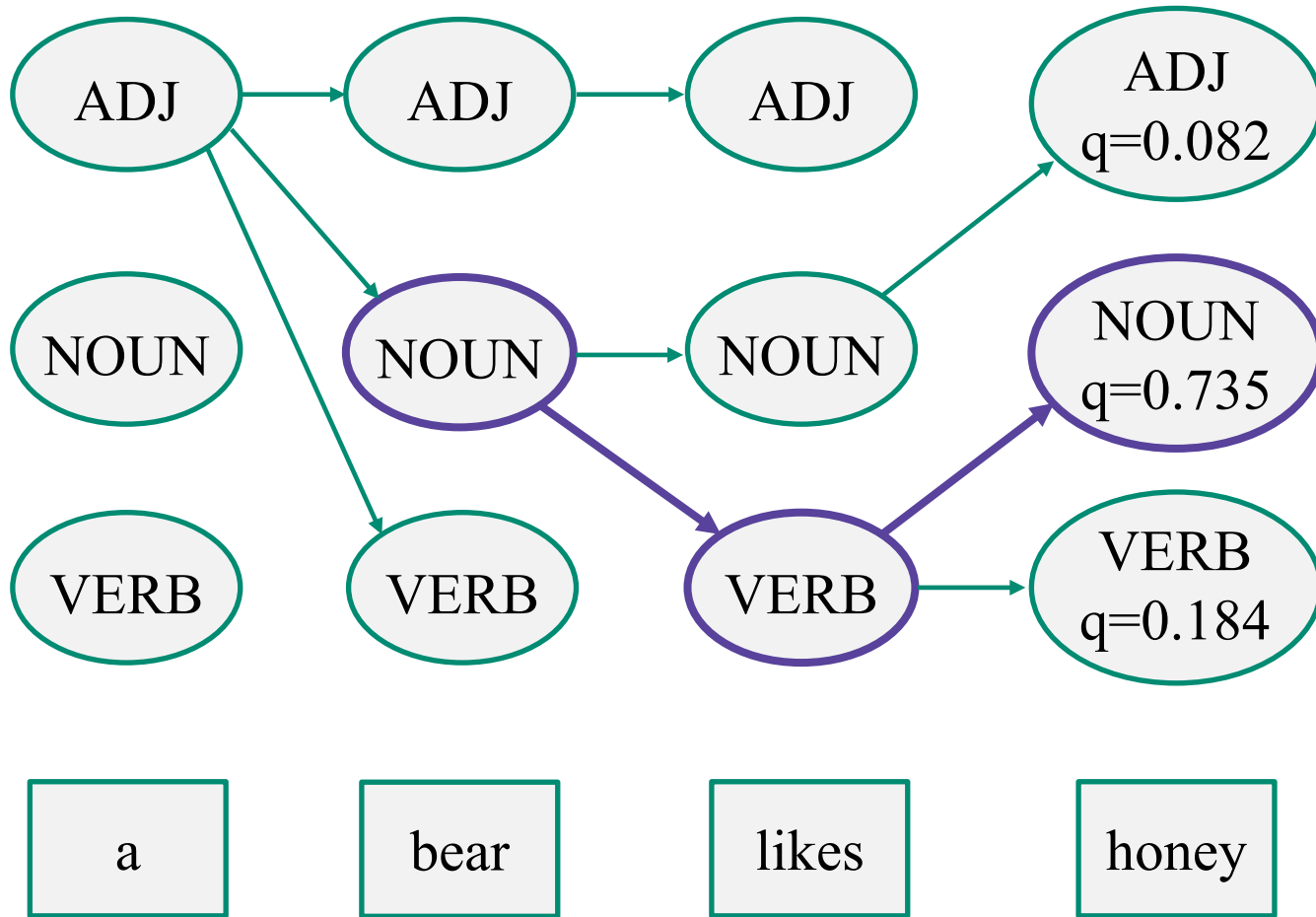
# Восстановление пути



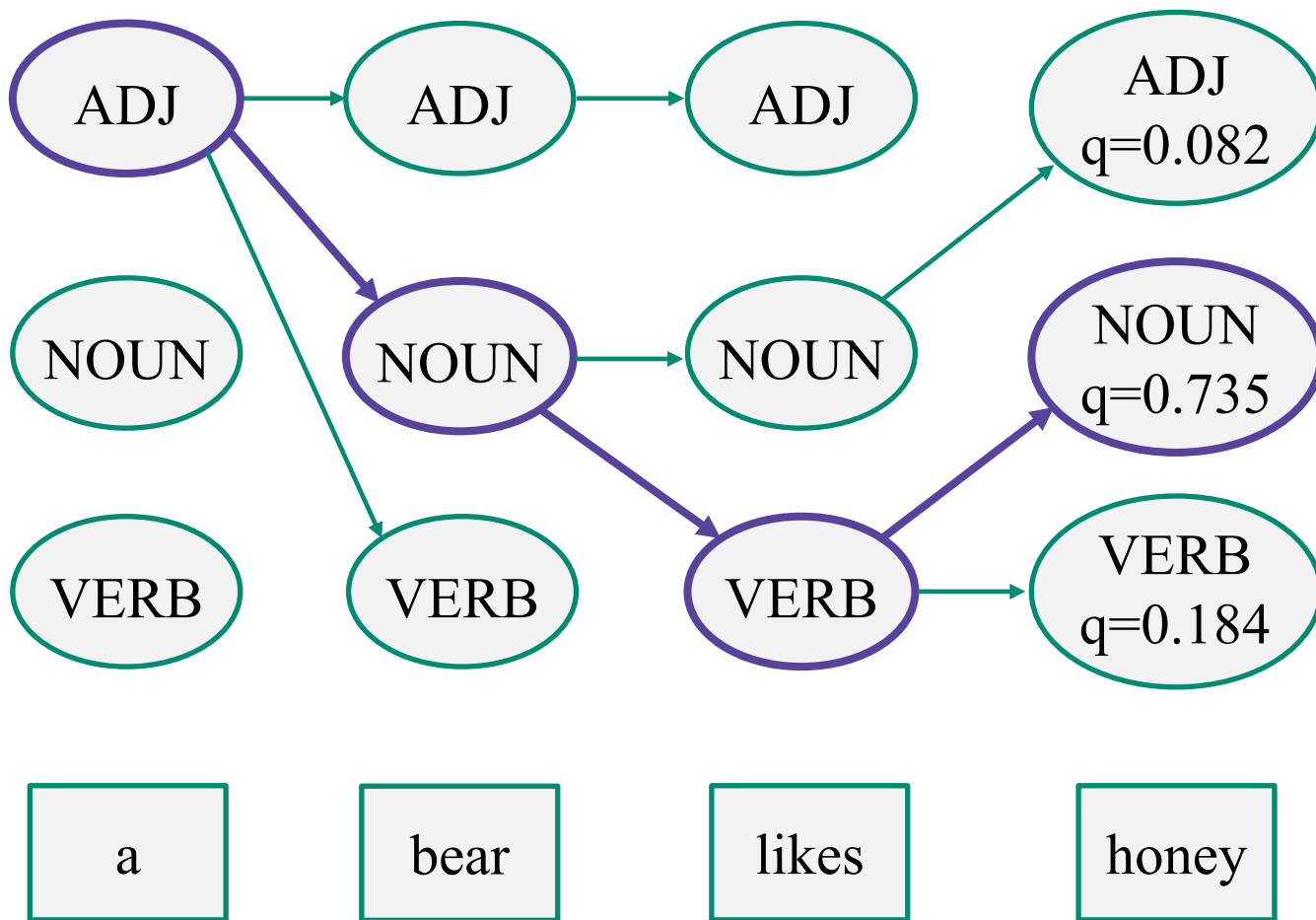
# Восстановление пути



# Восстановление пути

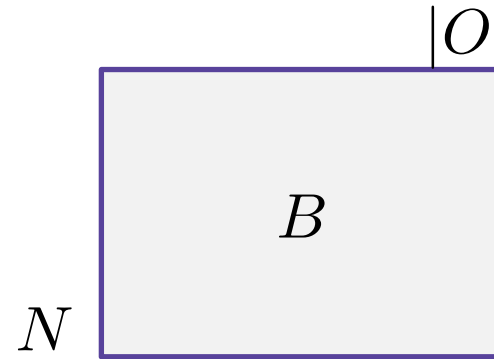
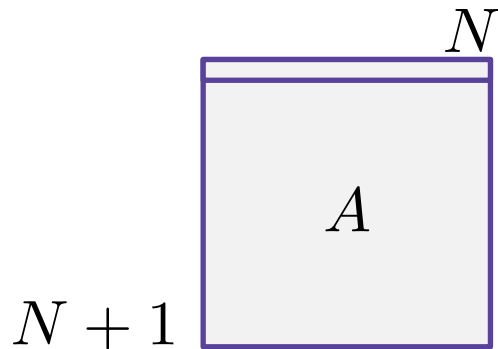


# Восстановление пути



### **3. Обучение НММ по размеченным и неразмеченным данным**

# Как обучить модель?



Метод максимума правдоподобия приводит к частотным оценкам вероятностей:

$$a_{ij} = p(s_j | s_i) = \frac{c(s_i \rightarrow s_j)}{c(s_i)}$$

$$b_{ik} = p(o_k | s_i) = \frac{c(s_i \rightarrow o_k)}{c(s_i)}$$

# Метод максимума правдоподобия

То же самое немного в других обозначениях:

$$a_{ij} = p(s_j | s_i) = \frac{\sum_{t=1}^T [y_{t-1} = s_i, y_t = s_j]}{\sum_{t=1}^T [y_t = s_i]}$$

Техническая деталь: предложения склеиваются спец. токенами в единый корпус длины  $T$ .

А можно ли как-то обучать НММ по неразмеченной выборке (только тексты)?



# Алгоритм Баума-Велша (Baum-Welch)

**E-step:** оценка апостериорных вероятностей на скрытые переменные:

$$p(y_{t-1} = s_i, y_t = s_j)$$

*Решается динамическим программированием: алгоритм вперед-назад (forward-backward)*

**M-step:** оценки максимума правдоподобия на параметры:

$$a_{ij} = p(s_j | s_i) = \frac{\sum_{t=1}^T p(y_{t-1} = s_i, y_t = s_j)}{\sum_{t=1}^T p(y_t = s_i)}$$

# Резюме по алгоритмам в НММ

Обучение:

- ✓ по размеченным данным: метод максимума правдоподобия (частотные оценки вероятностей)
- по неразмеченным данным: алгоритм Баума-Велша (EM-алгоритм):
  - E-шаг: алгоритм вперед-назад для оценивания апостериорных вероятностей на параметры

Применение:

- ✓ алгоритм Витерби для поиска максимально вероятной последовательности меток

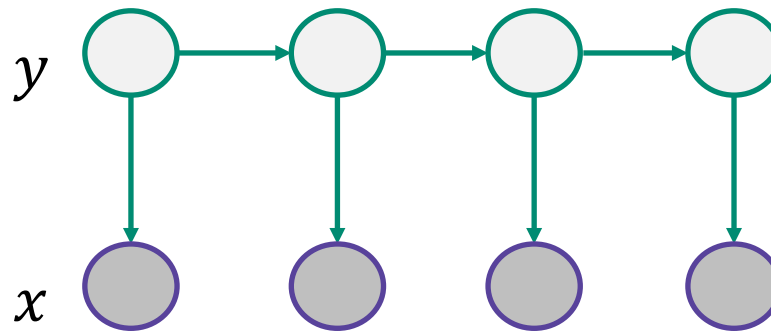
## **4. Другие графические модели: MEMM, CRF**

# Hidden Markov Model (HMM)


$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^T p(y_t | y_{t-1}) p(x_t | y_t)$$



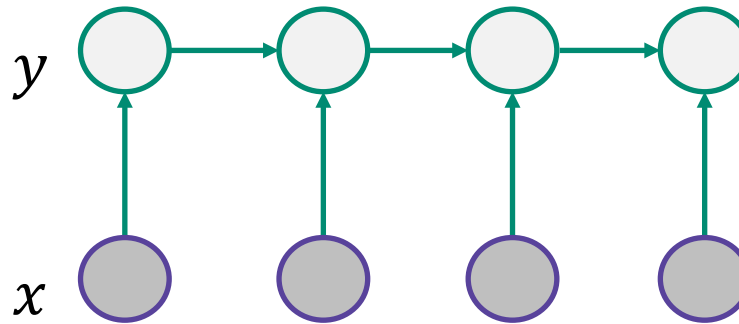
Генеративная  
модель



# Maximum Entropy Markov Model (MEMM)

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^T p(y_t|y_{t-1}, x_t)$$


Дискриминативная  
модель



# Maximum Entropy Markov Model (MEMM)

$$p(y_t | y_{t-1}, x_t) = \frac{1}{Z_t(y_{t-1}, x_t)} \exp \left( \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right)$$

↑  
Нормализация

↑  
вес

↑  
признак

# Maximum Entropy Markov Model (MEMM)

$$p(y_t | y_{t-1}, x_t) = \frac{1}{Z_t(y_{t-1}, x_t)} \exp \left( \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right)$$

↑  
Нормализация

↑  
вес

↑  
признак

## Conditional Random Field (линейный случай)

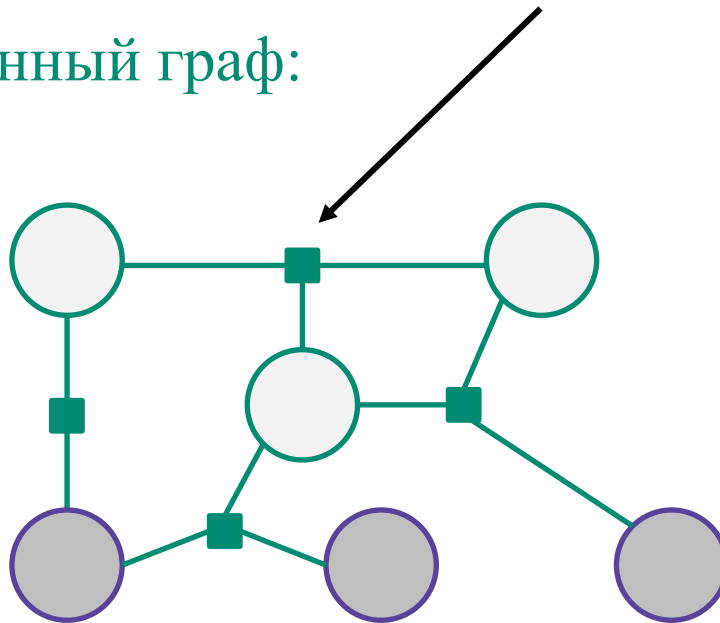
$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^T \exp \left( \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right)$$

# Conditional Random Field (общий случай)

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{a=1}^A \Psi_a(y_a, x_a)$$

↑  
Потенциалы

Неориентированный граф:





# Генерация признаков

Удобно использовать следующие типы признаков:

## Label-observation:

$$1. f(y_t, y_{t-1}, x_t) = [y_t = y] g_m(x_t)$$

$$2. f(y_t, y_{t-1}, x_t) = [y_t = y][y_{t-1} = y']$$

$$3. f(y_t, y_{t-1}, x_t) = [y_t = y][y_{t-1} = y'] g_m(x_t)$$

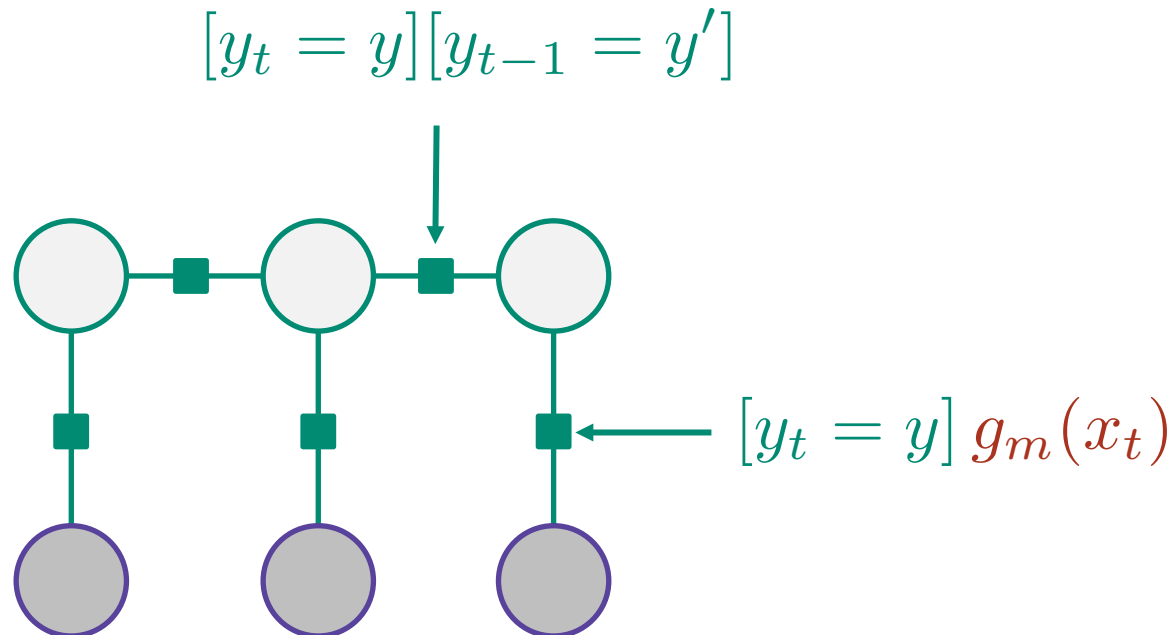
Откуда брать функции  $g_m(x_t)$  ?

# Примеры таких функций

	$w_t = v$	$\forall v \in$
	part-of-speech tag for $w_t$ is $j$	$\forall$ tags $j$
	$w_t$ is in a phrase of syntactic type $j$	$\forall$ tags $j$
Capitalized	$w_t$ matches $[A-Z][a-z]^+$	
AllCaps	$w_t$ matches $[A-Z]^+$	
EndsInDot	$w_t$ matches $[\^\.]^+.*\.$	
	$w_t$ matches a dash	
	$w_t$ appears in a list of stop words	
	$w_t$ appears in list of capitals	

# Часто достаточно парных взаимодействий

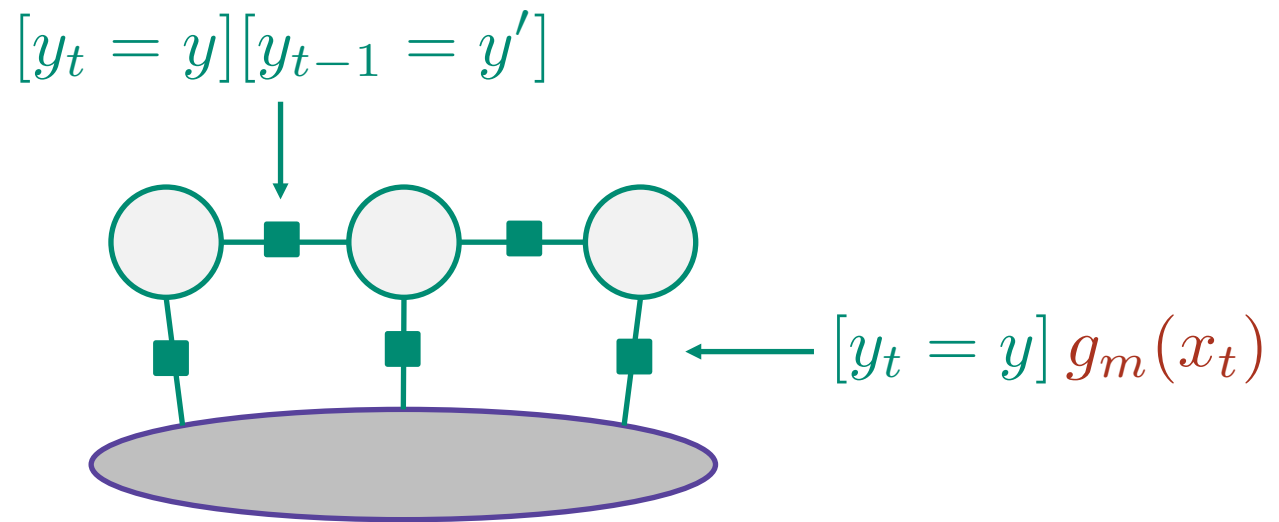
1.  $f(y_t, y_{t-1}, x_t) = [y_t = y] g_m(x_t)$
2.  $f(y_t, y_{t-1}, x_t) = [y_t = y][y_{t-1} = y']$



# Зависимость признаков от всех слов

Будем считать, что текущий вход  $x_t$  содержит не только текущее слово  $w_t$ , но еще и соседние:  $w_{t-1}$  и  $w_{t+1}$ .

Или вообще все слова последовательности:



*Такой трюк возможен только в дискриминативной модели.*

# Готовые реализации CRF

CRF++	<a href="https://sourceforge.net/projects/crfpp/">https://sourceforge.net/projects/crfpp/</a>
MALLET	<a href="http://mallet.cs.umass.edu/">http://mallet.cs.umass.edu/</a>
GRMM	<a href="http://mallet.cs.umass.edu/grmm/">http://mallet.cs.umass.edu/grmm/</a>
CRFSuite	<a href="http://www.chokkan.org/software/crfsuite/">http://www.chokkan.org/software/crfsuite/</a>
FACTORIE	<a href="http://www.factorie.cc">http://www.factorie.cc</a>

# Резюме

## Вероятностные графические модели:

- Hidden Markov Models (генеративная, направленная)
- Maximum Entropy Markov Models (дискриминативная, направленная)
- Conditional Random Field (дискриминативная, ненаправленная)

## Практика:

- POS: обучение HMM, Витерби (*первая домашка*)
- NER: CRF из готового пакета + генерация фичей...

Или: bi-LSTM (*следующая лекция и вторая домашка*)