Автоматический морфологический анализ

(часть 2)

Еникеева Екатерина Владимировна

2021

Автоматическая обработка естественного языка, лекция 4

План

1. Морфологическая неоднозначность
2. Проблемы rule-based методов
3. Brill tagger
4. HMM tagger
5. Другие статистические методы
6. Оценка качества
7. * RNN tagger

Уровни морф. неоднозначности

- неоднозначность лемм
  - леммы имеют разный POS-тег
    `стали` → `сталь` NOUN / `стать` VERB
  - леммы имеют один POS-тег и совпадающие формы, но разную начальную
    графине → `графин` NOUN masc / `графиня` NOUN femn
  - супплетивные формы
    `дети` → `дитя` / `ребёнок`
- неоднозначность форм одной лексемы
  `красного` → `красный` ADJ masc / neut gent
- транспозиция (conversion / zero derivation)
  В палату привезли `больного`.

Вспомним про rule-based

Плюсы

- Не нужны обучающие данные, но нужен хорошо размеченный корпус
- Результаты не ухудшаются из-за расширения множества тегов
- Используются независимые друг от друга правила (или группы правил)

Минусы

- Жёсткая система правил
- Низкая полнота
- Много ручной работы
- Набор правил нельзя/сложно адаптировать к другим языка

Brill tagger

Автоматическое построение правил по корпусу

> обучение на размеченном корпусе — Brill 1992-1994

> обучение без учителя (unsupervised) — Brill 1995

Требования:

- Словарь / обучающий корпус
- Шаблоны правил

Идея Brill tagger

- Каждому слову в обучающей выборке присваиваем самый частотный тег для этого слова
- Сравниваем с эталонной разметкой и формулируем правила **изменения** приписанного тега (transformation)
- Повторяем несколько итераций, пока не будет достигнут запланированный эффект:
  - полное отсутствие улучшений
  - заданный уровень точности
  - заданное максимальное число правил

Unsupervised Brill tagger

(Brill 1995)

- Корпус текстов без предварительной разметки и словарь
- Предварительная разметка текста по словарю с указанием всех вариантов

The	can	will	rust
DT	MD **NN** VB	**MD** NN VB	NN **VB**

Правила в Brill tagger

Общий вид правил:

«Заменить тег X на тег Y в контексте C, где X является последовательностью из двух или более тегов, а Y – один тег, такой что $Y \in X$».

Пример построения правила

Строим частотную модель для шаблонов правил:

После слова *the* среди однозначной разметки чаще всего встречаются слова с тегом NN. Можем сформулировать следующее правило:

- Заменять тег MD_NN_VB на NN после слова *the*

The	can	will	rust
DT	MD **NN** VB	**MD** NN VB	NN **VB**

Вероятностные методы

- Скрытые марковские модели (Hidden Markov Model, HMM)

  вычисление параметров:

  - Алгоритм Витерби (Viterbi)
  - Алгоритм Баума-Уэлча (Baum – Welch)

- Нейросетевые модели

Sequence labelling task

- Задача разметки последовательности

1 токен → 1 тег

Предложение длины N → последовательность тегов длины N

- Probabilistic sequence model: строим распределение вероятностей на возможных последовательностях тегов, выбираем наиболее вероятную

Как использовать частоты?

- Простейший вариант – присваивать каждой словоформе наиболее вероятную морфологическую интерпретацию – вспомним 1-gram LM
- За вероятности принимаются нормализованные частоты присвоения тега определенной форме в размеченном корпусе:

$$P(t|w) = \frac{count(w,t)}{|C|}$$

Марковская цепь

- Множество возможных состояний / states: $Q = q_1 \dots q_N$
- Матрица вероятностей переходов из состояния i в j / transition probability matrix: $A = a_{11} \dots a_{ij} \dots a_{NN}$
- Исходное распределение вероятностей состояний: $\pi = \pi_1 \dots \pi_N$

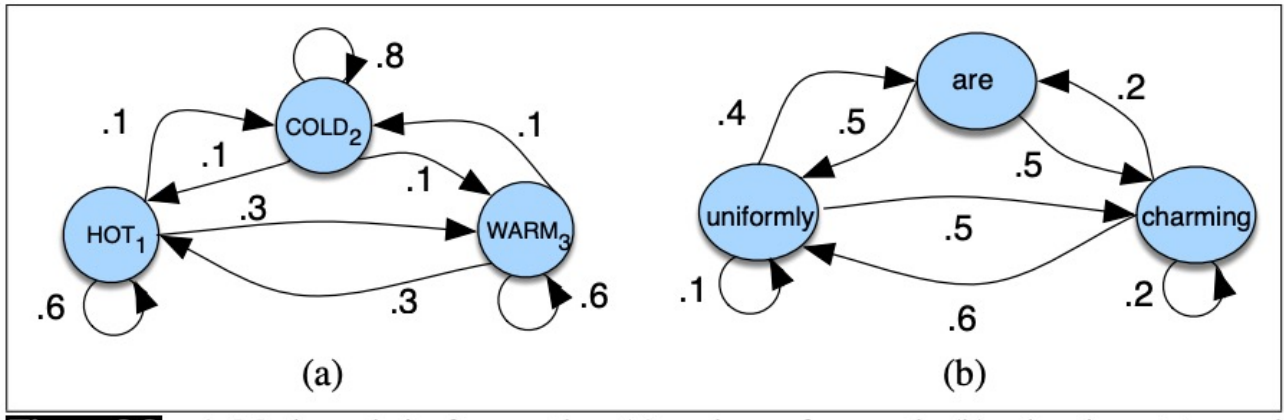


Figure 8.8 A Markov chain for weather (a) and one for words (b), showing states and transitions. A start distribution π is required; setting $\pi = [0.1, 0.7, 0.2]$ for (a) would mean a probability 0.7 of starting in state 2 (cold), probability 0.1 of starting in state 1 (hot), etc.

Скрытая Марковская модель

- Hidden Markov Model (HMM)
- Множество возможных состояний / states: $Q = q_1 \dots q_N$
- Матрица вероятностей переходов из состояния i в j / transition probability matrix: $A = a_{11} \dots a_{ij} \dots a_{NN}$
- Последовательность наблюдений / observations: $O = o_1 \dots o_T$
- Последовательность вероятностей наблюдений / emission probabilities: $B = b_i(o_t)$
- Исходное распределение вероятностей состояний: $\pi = \pi_1 \dots \pi_N$

Markov assumption

Предполагаем *марковское свойство / Markov assumption* (как в n-граммных языковых моделях):

- встречаемость каждого тега в определенном месте цепочки зависит только от предыдущего тега
  $P(q_i|q_1 \dots q_{i-1}) = P(q_i|q_{i-1})$
- то, какое слово находится в том или ином месте цепочки, полностью определяется тегом (а не, допустим, соседними словами)
  $P(o_i|q_1 \dots q_T, o_1 \dots o_T) = P(o_i|q_i)$

> *марковская модель 1-го порядка*

HMM tagger

Вероятности перехода A:

$$a_{i,i-1} = P(t_i|t_{i-1}) = \frac{count(t_{i-1}, t_i)}{count(t_{i-1})}$$

Вероятности наблюдений B:

$$b_i(w_i) = P(w_i|t_i) = \frac{count(t_i, w_i)}{count(t_i)}$$

- Как здесь можно использовать готовый морфологический словарь?

Вероятности на примере

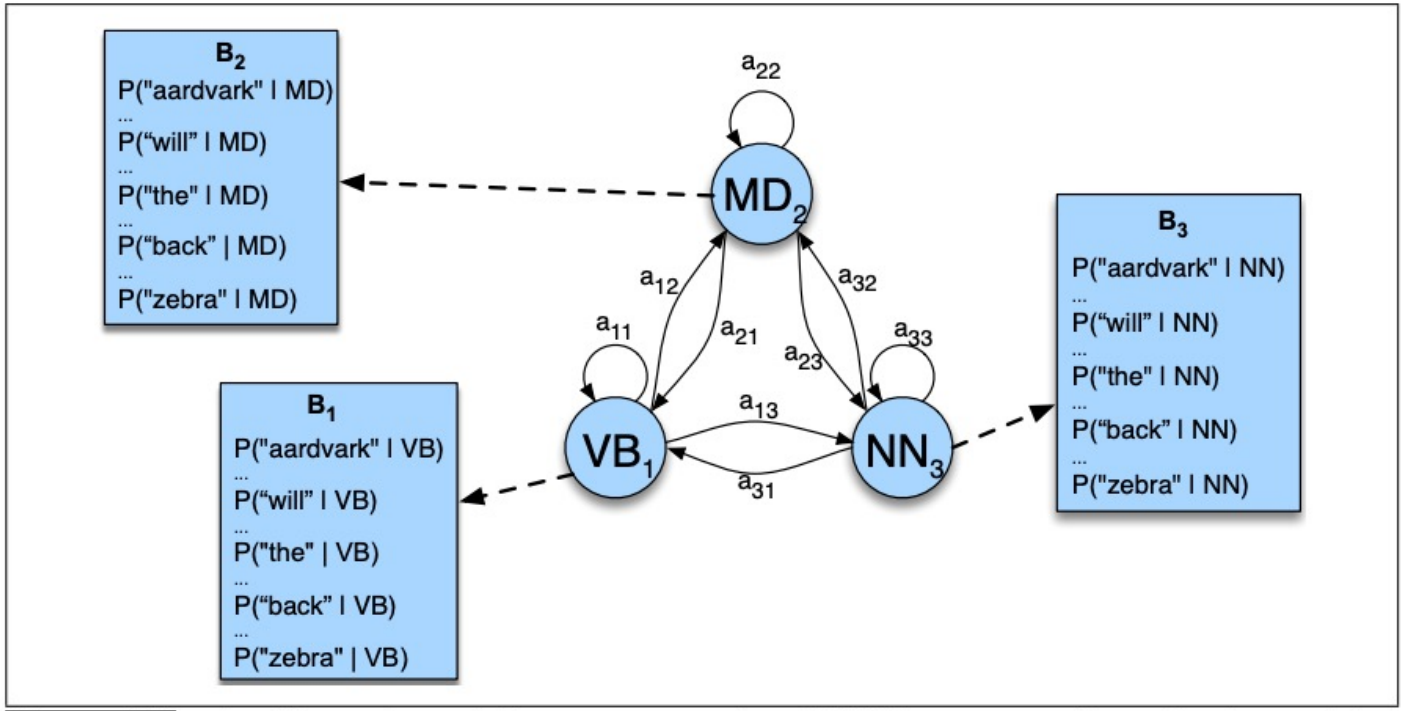


Figure 8.9 An illustration of the two parts of an HMM representation: the *A* transition probabilities used to compute the prior probability, and the *B* observation likelihoods that are associated with each state, one likelihood for each possible observation word.

HMM decoding

Decoding – определение последовательности скрытых состояний, соответствующее наблюдениям:

$$\hat{t}_{1:n} = arg \max_{t_1 \dots t_n} P(t_1 \dots t_n | w_1 \dots w_n)$$

$$= arg \max_{t_1 \dots t_n} \frac{P(w_1 \dots w_n | t_1 \dots t_n)\, P(t_1 \dots t_n)}{P(w_1 \dots w_n)}$$

$$= arg \max_{t_1 \dots t_n} P(w_1 \dots w_n | t_1 \dots t_n) P(t_1 \dots t_n)$$

$$= \arg \max_{t_1 \dots t_n} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

Алгоритм Витерби

Позволяет сократить количество вычислений:

1. Инициализация:

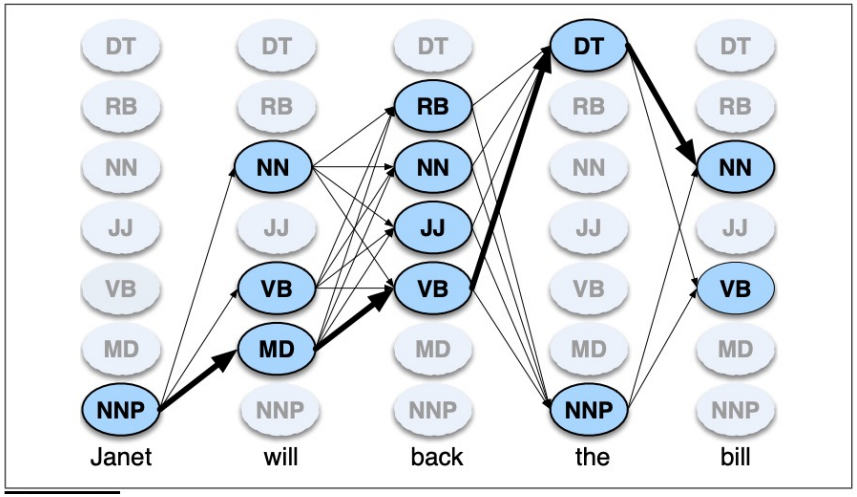
   для первого токена используем только вероятность наблюдения и π

2. Рекурсия:

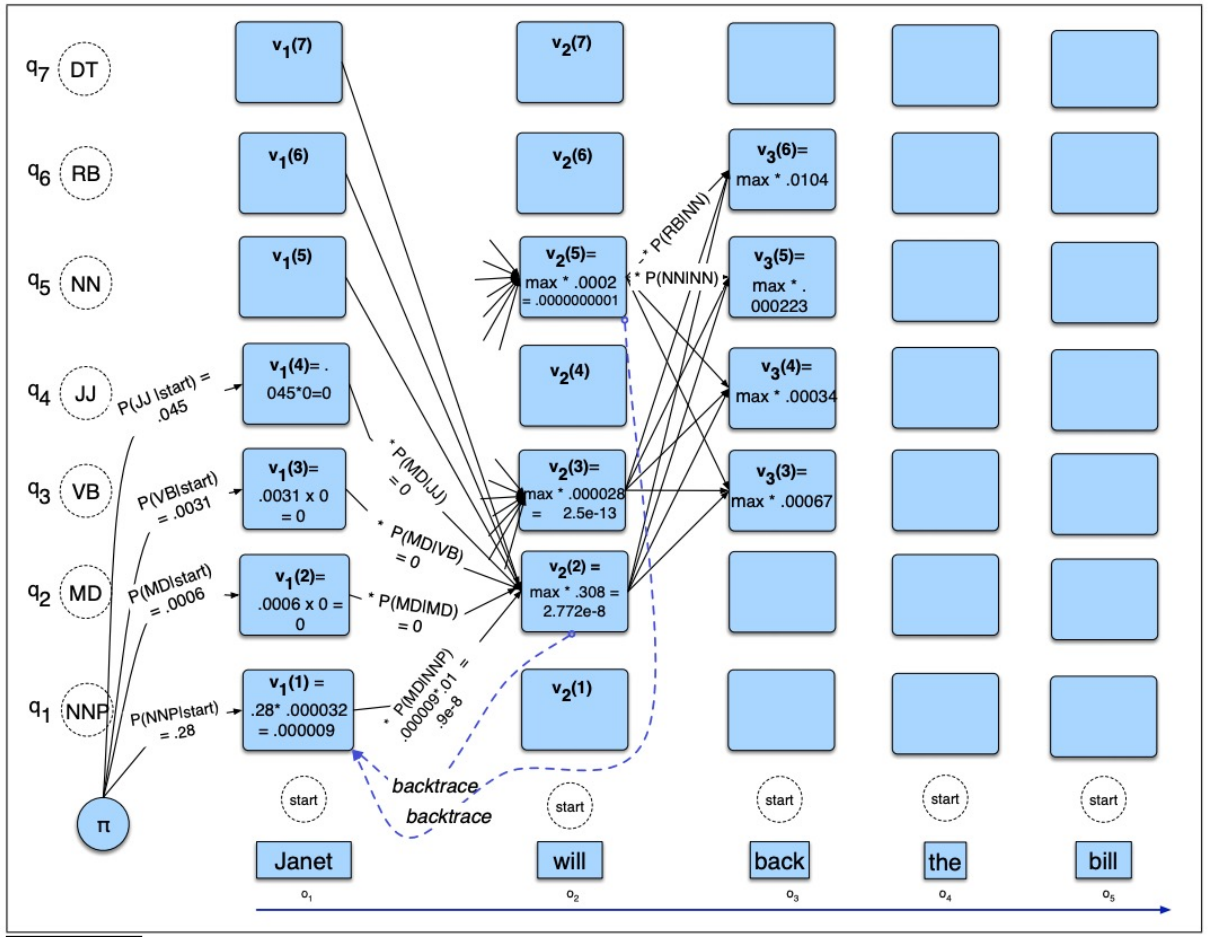
   в момент t выбираем наиболее вероятный путь к текущему состоянию

   $$v_t(j) = \max_{i=1\ldots N} v_{t-1}(i) a_{ij} b_j(o_t)$$

   сохраняем лучший тег (backpointer)



Алгоритм Витерби



Другие модели теггинга

- 3gram HMM (2-order assumption)
- maximum entropy Markov model – MEMM tagger
- Conditional Random Fields – CRF taggers
- Recurrent neural network – RNN taggers
- BiLSTM taggers
- …

Оценка качества теггинга

С учётом разрешения неоднозначности:

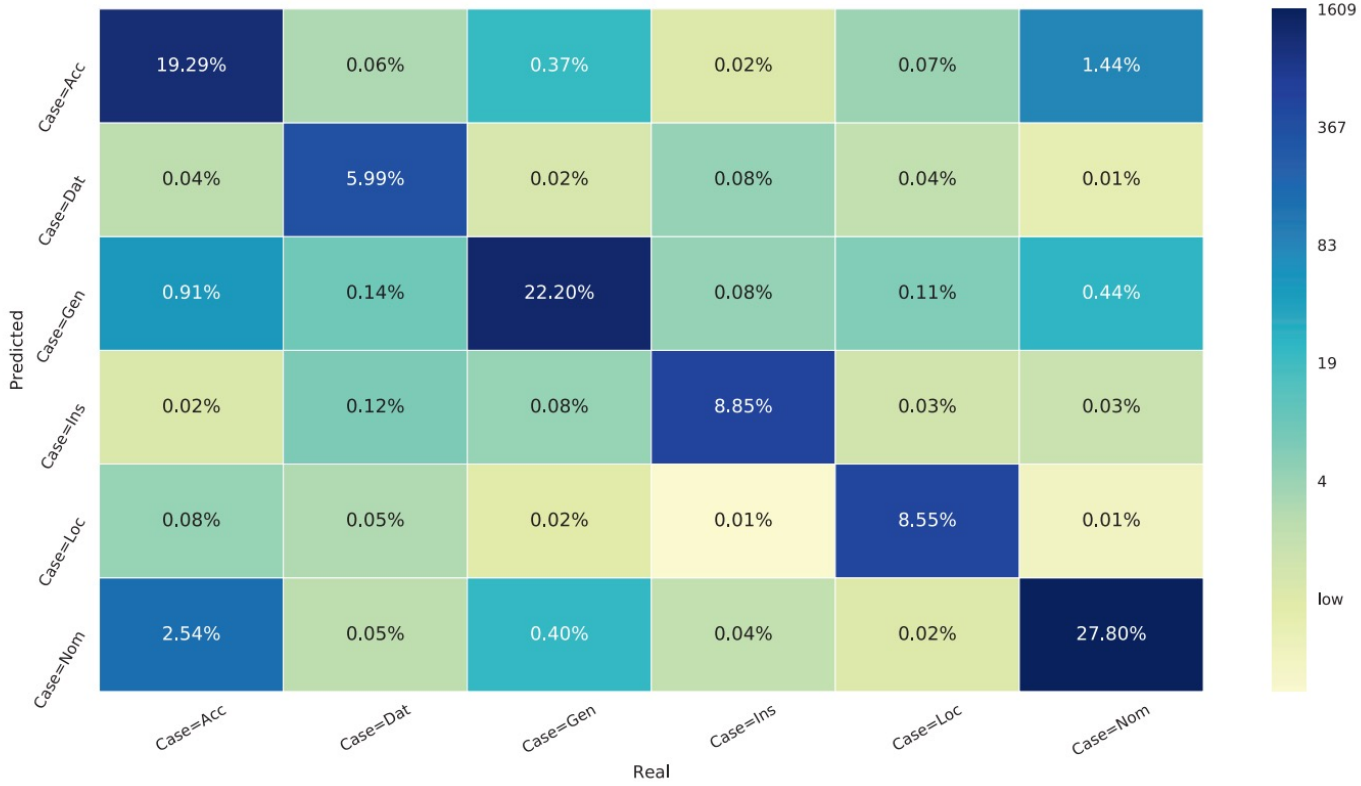
- Accuracy на уровне POS
- Accuracy на уровне полного тега
- Accuracy per tag / class
- Accuracy per sentence

Полезно помнить:

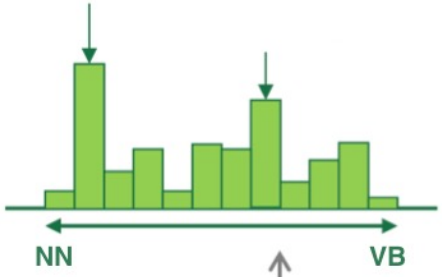
- согласованность разметки (Penn Treebank) – 97% (*human ceiling*)
- unigram baseline – см. слайд 11

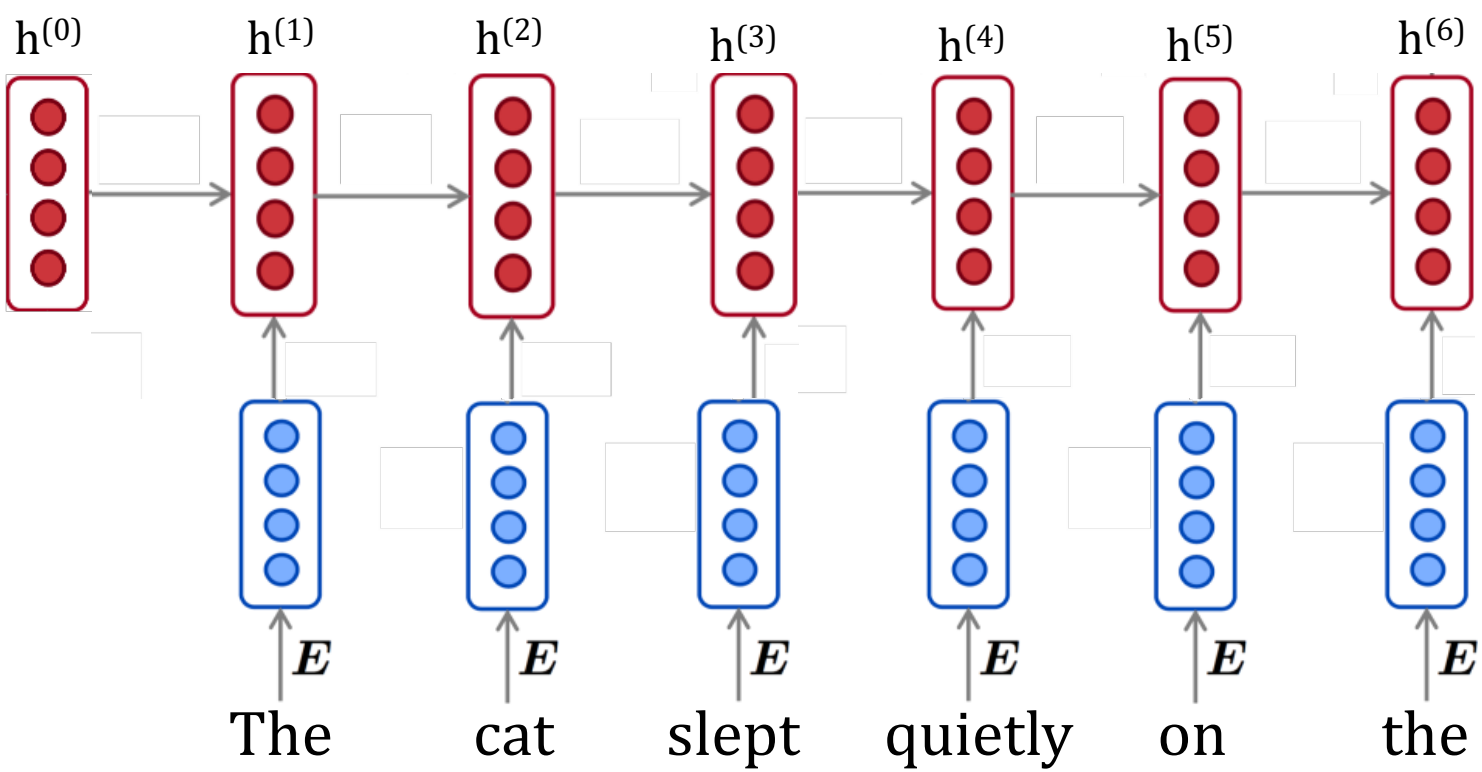
Error Analysis

Confusion matrix / contingency table



RNN-LM





RNN-LM

- Каждое слово превращаем в вектор e_{w_t}
- Скрытый слой

$$h^{(t)} = \sigma(W_h h^{(t-1)} + W_e e_{w_{t-1}} + b_1)$$

- Выходное распределение

$$\hat{y}_t = soft\max(Uh^{(t)} + b_2)$$

Спасибо!

Вопросы?