Анализ неструктурированных данных

2. Морфологический анализ

Екатерина Черняк

echernyak@hse.ru

Национальный Исследовательский Университет – Высшая Школа Экономики
НУЛ Интеллектуальных систем и структурного анализа

September 18, 2017

Морфологический анализ

Основные морфологического анализа:

- **Разбор слова**
  - **Лемматизация** – определение нормальной формы слова (леммы)
  - Определение грамматических характеристик слова (POS-tagging, частеречная разметка)
  - **Стемминг** – определение (псевдо)основны слова (стема)
- **Синтез слова** — генерация слова по заданным грамматическим характеристикам

Зачем?

- Для классификации / кластеризации для отбора признаков
  - Лемматизация и стемминг помогают сократить количество признаков (одно слово – один признак)
  - Фильтрация по частям речи тоже помогает сократить количество признаков
  - Извлечение групп [англ. chunking] (именных групп, глагольных групп) помогает добавить “умные” признаки
- Для более сложных задач обработки текста и речи в качестве предобработки:
  - Машинный перевод
  - Распознавание и генерация речи
  - Поиск

Части речи и их грамматические характеристики [по документации MyStem]

A	прилагательное	падеж, число, форма, степень сравнения, род	горячий, холодный
ADV	наречие		кисло, сладко
ADVPRO	местоименное наречие		почему, поэтому
ANUM	числительное-прилагательное	падеж, число, род	первый, третий
APRO	местоимение-прилагательное	падеж, число, род	мой, твой
COMP	часть композита		
CONJ	союз		и, но
INTJ	междометие		ах, ну
NUM	числительное	падеж	двадцать, пять
PART	частица		бы, же
PR	предлог		в, на
S	существительное	род, число, падеж, одушевленность	гусь, топор
SPRO	местоимение-существительное	лицо, число, падеж	ты, вы
V	глагод	лицо, число, время, вид, репрезентация, залог, переходность	идти, смотреть

https://tech.yandex.ru/mystem/doc/grammemes-values-docpage/

Ресурсы

- НКРЯ http://ruscorpora.ru/
- ГИКРЯ http://www.webcorpora.ru/
- Открытый корпус http://opencorpora.org/
- MorphoRuEval-2017
  http://www.dialog-21.ru/evaluation/2017/morphology/

Морфологические процессоры для русского языка

Mystem3 (https://tech.yandex.ru/mystem/)

```
In[1]:  from pymystem3 import Mystem
In[2]:  text = “На востоке Москвы неизвестные ограбили
ювелирный магазин”
In[3]:  m = Mystem()
In[4]:  lemmas = m.lemmatize(text)
```

pymorphy2 (https://github.com/kmike/pymorphy2)

```
In[1]:  from pymorphy2 import MorphAnalyzer
In[2]:  m = MorphAnalyzer()
In[3]:  lemmas = [m.parse(word)[0].normal_form for word in
text.split()]
```

Стемминг для русского языка

nltk.stem.snowball.RussianStemmer

```
In[1]:  from nltk.stem.snowball import RussianStemmer
In[2]:  stemmer = RussianStemmer()
In[3]:  stem = stemmer.stem(‘‘оптимизация’’)
```

Лемматизация

Каждой **словоформе** соответствует **лемма** (нормальная форма):

- кошке, кошку, кошкам, кошкой $\Longrightarrow$ кошка
- бежал, бежит, бегу $\Longrightarrow$ бежать
- белому, белым, белыми $\Longrightarrow$ белый

Парадигма

Словоизменительная парадигма — список словоформ, принадлежащих одной лексеме и имеющих разные грамматические значения.

пальто-	плакать	рук-а
	плач-у	рук-и
	плач-ешь	рук-е
	плач-ет	рук-у
	плач-ем	рук-ой
	плач-ете	о рук-е
	плач-ут	

Стемминг

Слова состоят из морфем: $\texttt{word} = \texttt{stem} + \texttt{affixes}$. Стемминг позволяет отбросить аффиксы (чаще всего – только суффиксы).

- павлиний, павлиньи, павлиньим $\Longrightarrow$ павлин
- пакет, пакетом, пакеты $\Longrightarrow$ пакет

Основные проблемы

- Морфологическая неоднозначность
  - Существительное или глагол: стали, стекло, течь, белила, падали
  - Прилагательное или существительное: мороженое, простой
  - Существительное или существительное: черепах
- Новые слова

Основные подходы: лемматизация, POS-tagging

- unigram tagging: (правила и словари) выбираем самый частый / вероятный разбор
- ngram tagging: анализируем контекст текущего слова – n предыдущих слов
  - Учет окна фиксированной длины: SVM, MaxEnt, SENNA
  - Модели последовательностей [sequence labelling]: HMM, MEMM, CRF, RNN

1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

Основные подходы: стемминг, алгоритм Портера

Алгоритм Портера состоит из 5 циклов команд, на каждом цикле – операция удаления / замены суффикса. Возможны вероятностные расширения алгоритма.
Ошибки:

- белый, белка, белье $\Longrightarrow$ бел
- трудность $\Longrightarrow$ трудност, трудный $\Longrightarrow$ труд
- быстрый, быстрее $\Longrightarrow$ быст, побыстрее $\Longrightarrow$ побыст

`http://snowball.tartarus.org/algorithms/russian/stemmer.html`

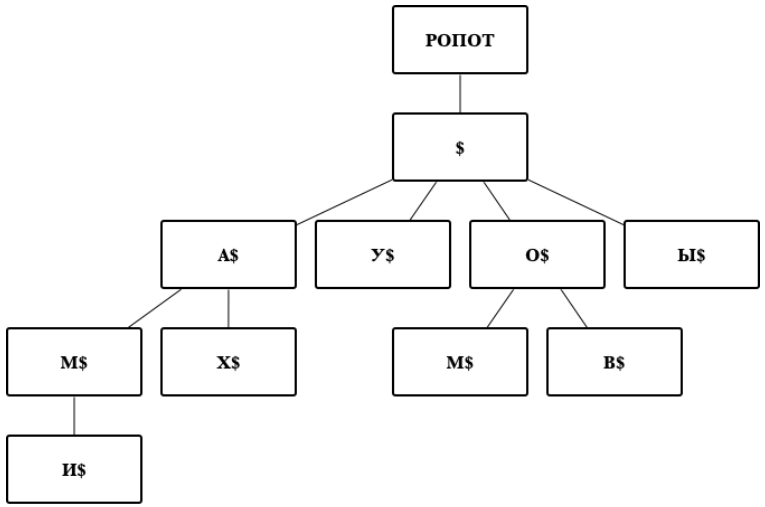
1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

Поиск в словаре [Segalovich, 2003]

Словарь представлен как префиксное дерево (trie) инвертированных основ и дополнительное префиксное дерево для хранения окончаний. Форма записи слова: топор – “ропот$A”, где “A” – парадигма (например, -и, -ами, -ом). Разбор слова:

1. Начиная с правого конца слова найти все возможные разбиения на основу + окончание
2. Повторить следующие шаги, начиная с самого длинного возможного окончания (короткой основы)
3. Найти основу в префиксных деревьях для основ, проверить, есть ли форма с нужным окончанием. Если есть, разбор найден, если нет – перейти к следующему разбиению.



Цепи Маркова и модель биграм

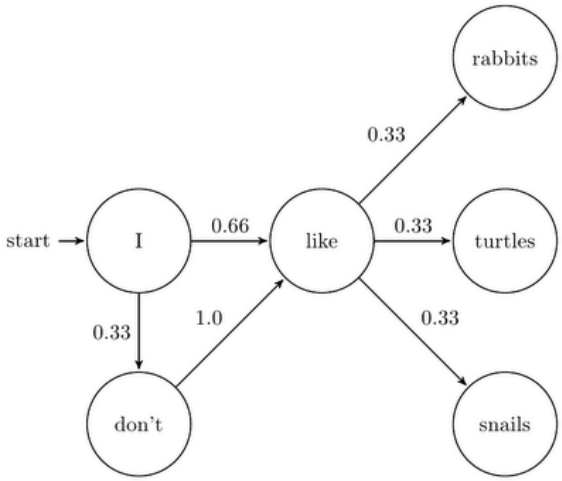
Цепь Маркова порядка 2 – только одна предыдущая предыстория (состояние) важно

1. Вероятность последовательности слов

$$p(w_l^1) = \prod_{k=1}^{l} p(w_k | w_{k-1})$$

2. Вероятность следующего слова

$$= \frac{p(w_{i-1} w_i}{w_{i-1}}$$



Оценки вероятностей в модели биграм

ММП оценка вероятностей в модели биграм:

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

Если появляется новое слово, возникает проблема нулевых вероятностей:

$$\text{count}(w^2) = \text{count}(w_{i-1}w_i) = 0$$

1. Преобразование Лапласа: $\widehat{p(w_i|w_{i-1})} = \frac{\text{count}(w^2)+\alpha}{\text{count}(w_{i-1})+\alpha|V|}$
2. Преобразование Гуд-Тьюринга: $\text{count}(w^2) = \frac{(\text{count}(w_{i-1}w_i)+1)*N_{c+1}}{N_c}$
   N_c – количество биграмм, которые встречаются $\text{count}(w^2)$ раз

Использование языковой модели для генерации псевдослучайного текста

Markovify

```
In[1]: corpus = open("sherlock.txt").read()
In[2]: text_model = markovify.Text(corpus, state_size=3)
In[3]: text_model.make_short_sentence(140)
Out[1]:“It cost me something in foolscap, and I had no idea
that he was a man of evil reputation among women.”
```

https://github.com/jsvine/markovify

1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

Скрытая цепь Маркова

Скрытая цепь Маркова [Hidden Markov Model, HMM]

$$\hat{T} = \arg\max_T P(T|W)$$

$$\arg\max_T P(W|T)P(T)$$

$$\arg\max_T \prod_i P(w_i|t_i) \prod_i (t_i|t_{i-1})$$

T – конечное множество частеречных тегов

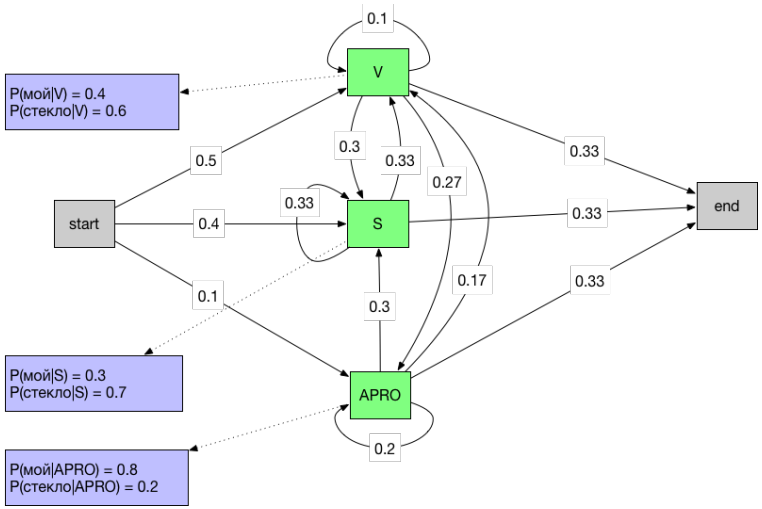
W – конечное множество слов

Скрытая цепь Маркова

$< Q, A, O, B, q_0, q_F >$:

- $Q = q_1, \ldots, q_N$ – конечное множество состояний;
- A – матрица вероятностей переходов размером $|Q| \times |Q|$, $0 \leq a_{ij} \leq 1$;
- O – конечное множество наблюдений;
- B – вероятности наблюдений, $b_i \to \mathbb{R}, \sum_{o \in O} b_i(o) = 1, 1 \leq i \leq |Q|$;
- q_0, q_F – специальные начальные и конечные символы и соответствующие им вероятности переходов a_{0i}, a_{iF}, $0 \leq a_{0i}, a_{iF} \leq 1$, $1 \leq i \leq |Q|$;

$\sum_{j=1}^{|Q|} a_{ij} + a_{iF} = 1, 0 \leq i \leq |Q|$



Скрытая цепь Маркова

Марковские допущения о независимости:

1. Текущее состояние зависит только от предыдущего состояния:

$$p(q_{i_n}|q_{i_1}\dots q_{i_{n-1}}) = p(q_{i_n}|q_{i_{n-1}})(= a_{i_{n-1}i_n})$$

2. Текущее наблюдение зависит только от текущего состояния:

$$p(o_{i_j}|q_{i_1}\dots q_{i_{n-1}}, o_{i_1}\dots o_{i_{n-1}}) = p(o_{i_j}|q_{i_j})(= b_{i_j}(o_{i_j}))$$

Три задачи скрытых цепей Маркова

1. Оценить вероятность последовательности наблюдений в модели;
2. Найти последовательность состояний, которая с наибольшей вероятностью порождает данную последовательность наблюдений;
3. Оценить параметры модели (обучение по реальным данным).

Первая задача

По последовательности наблюдений $o = o_1 \dots o_n$ оценить вероятность последовательности o. Мы знаем, что:

$$p(o, q) = p(o|q)p(q)$$

Используем допущения о независимости:

$$p(o, q) = \prod_{i=1}^{n} p(o_i|q_i) \prod_{i=1}^{n} p(q_i|q_{i-1})$$

Тогда для всей последовательности наблюдений o:

$$p(o) = \sum_{q \in Q^n} \prod_{i=1}^{n} p(o_i|q_i) \prod_{i=1}^{n} p(q_i|q_{i-1}) p(q_F|q_n)$$

Прямой проход

Идея: используем динамическое программирование для вычисления $n \times |Q|$ значений $\alpha_{ij} = p(o_1 \dots o_i, q_i)$:

1. Инициализация

$$\alpha_{1j} = a_{0j} b(o_1), 1 \leq j \leq |Q|$$

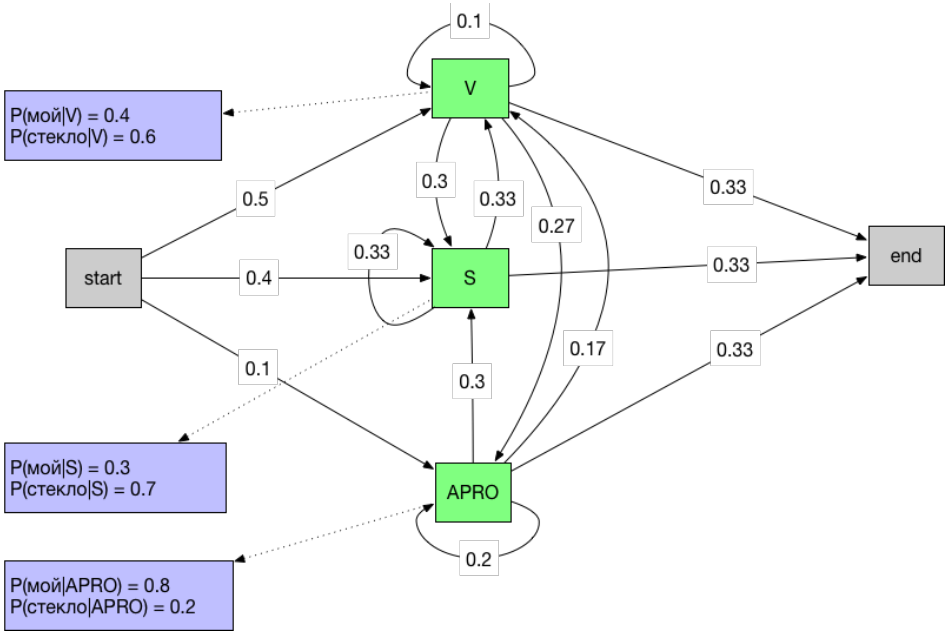
2. Шаг рекурсии

$$\alpha_{ij} = \sum_{k=1}^{|Q|} \alpha_{i-1k} a_{kj} b_j(o_i), 1 \leq i \leq n, 1 \leq j \leq |Q|$$

3. Завершение

$$p(o) = \sum_{k=1}^{|Q|} \alpha_{nk} a_{kF}$$

Вычисление вероятности последовательности наблюдений “мой стекло”



	start	мой	стекло	end
V	0.5	0.25	0.1219	0.0402
S	0.4	0.12	0.0970	0.0320
APRO	0.1	0.08	0.0167	0.0055

$P(\text{“мой стекло”}) = 0.07775$

Обратный проход

Идея: используем динамическое программирование для вычисления $n \times |Q|$ значений $\beta ij = p(o_{i+1}) \dots o_n, q_i)$:

1. Инициализация

$$\beta_{nj} = a_{jF}, 1 \leq j \leq |Q|$$

2. Шаг рекурсии

$$\beta_{ij} = \sum_{k=1}^{|Q|} \beta_{i+1k} a_{jk} b_k(o_{i+1}), 1 \leq i \leq n, 1 \leq j \leq |Q|$$

3. Завершение

$$p(o) = \sum_{k=1}^{|Q|} a_{0k} b_k(o_1) \beta_{1k}$$

Декодирование

По последовательности наблюдений $o = o_1 \dots o_n$ определить наиболее вероятную последовательность $q = q_1 \dots q_n \in Q^n$:

$$\texttt{argmax}_{q \in Q^n} p(o, q) = \texttt{argmax}_{q \in Q^n} p(o|q)p(q)$$

Используем допущения о независимости:

$$\texttt{argmax}_{q \in Q^n} p(o, q) = \texttt{argmax}_{q \in Q^n} \prod_{i=1}^{n} p(o_i|q_i) \prod_{i=1}^{n} p(q_i|q_{i-1})$$

Алгоритм Витерби

Идея: используем динамическое программирование для вычисления $n \times |Q|$ значений $v_{ij} = \max_{q \in Q^{i-1}} p(o_1 \dots o_i, q_1 \dots q_i)$:

1. Инициализация

$$v_{1j} = a_{0j} b(o_1), 1 \leq j \leq |Q|$$

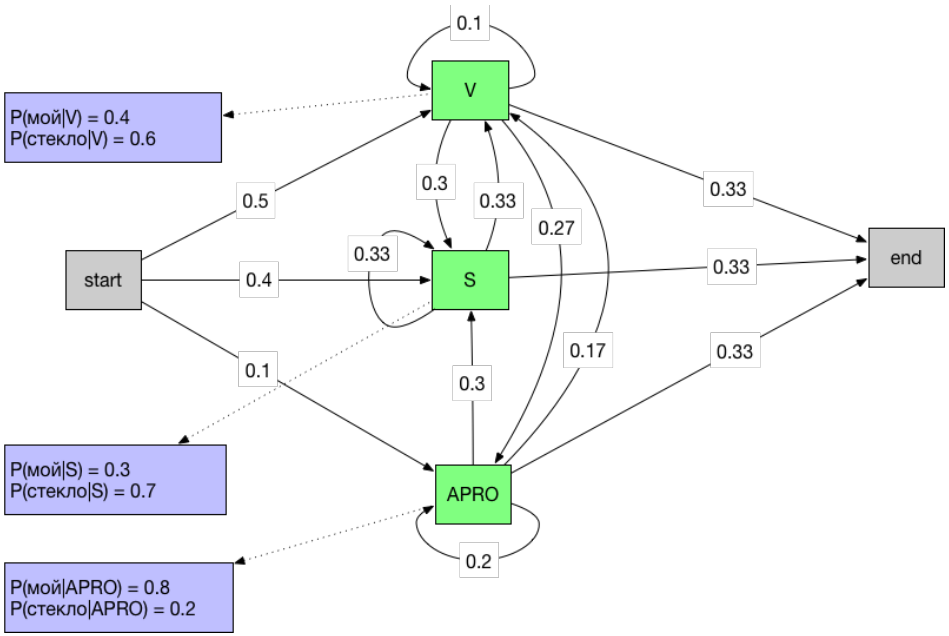
2. Шаг рекурсии

$$v_{ij} = \max v_{i-1k} a_{kj} b_j(o_i), 1 \leq i \leq n, 1 \leq j \leq |Q|$$

3. Завершение

$$\max_{q \in Q^n} p(o, q) = \max_{1 \leq k \leq |Q|} v_{nk} a_{kF}$$

Декодирование последовательности наблюдений “мой стекло”



	start	мой	стекло	end
V	0.5	**0.25**, start	0.015, V	0.0046, S
S	0.4	0.12, start	**0.0525**, V	**0.0177**, S
APRO	0.1	0.08, start	0.0135 V	0.0045, S

наиболее вероятная последовательность скрытых состояний: V S

p(“мой стекло”, V S) = 0.0177

TnT POS-tagger [Brants, 2000]

TnT использует скрытую Марковскую цепь второго порядка для того, чтобы найти частеречные тэги:

$$\arg\max[\prod_{j}[p(o_i|t_{o-1}, t_{o-2})p(q_i|o_i)]P(o_{T+1}|o_T)$$

Вероятность тэга для данного слова определяется как линейная интерполяция вероятностей, полученных из трех Марковских цепей::

$$P(o_i|o_{i-1}, o_{i-2}) = l_1 * P(o_i) + l_2 * P(o_i|o_{i-1}) + l_3 * P(o_i|o_{i-1}, o_{i-2})$$

nltk.tag.tnt

```
In[1]: from nltk.tag import tnt
In[2]: tnt_pos_tagger = tnt.TnT()
In[3]: tnt_pos_tagger.train(train_data)
In[4]: tnt_pos_tagger.evaluate(test_data)
```

1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

Марковская модель максимальной энтропии [McCallum, 2000], [Toutanova,2003]

Марковская модель максимальной энтропии [Maximum-entropy Markov model, MEMM]

$$\hat{T} = \arg\max_T P(T|W)$$

$$\arg\max_T \prod_i P(t_i) \prod_i (t_i|w_i, t_{i-1})$$

T – конечное множество частеречных тегов

W – конечное множество слов

Метод максимальной энтропии, MaxEnt

Индикаторные признаки:

У/PR страха/S глаза/S велики/(S или A) ./PUNCT

$$f_{11}(c,x) = \begin{cases} 1, & if\ \ t_{-1} = S, c = S \\ 0, & otherwise \end{cases} \qquad f_{12}(c,x) = \begin{cases} 1, & if\ \ t_{-1} = S, c = A \\ 0, & otherwise \end{cases}$$

$$f_{21}(c,x) = \begin{cases} 1, & if\ \ w_{-1}[:-1] = a, c = S \\ 0, & otherwise \end{cases} \qquad f_{22}(c,x) = \begin{cases} 1, & if\ \ w_{-1}[:-1] = a, c = A \\ 0, & otherwise \end{cases}$$

$$f_{31}(c,x) = \begin{cases} 1, & if\ \ w_{+1} = \text{“.”}, c = S \\ 0, & otherwise \end{cases} \qquad f_{32}(c,x) = \begin{cases} 1, & if\ \ w_{+1} = \text{“.”}, c = A \\ 0, & otherwise \end{cases}$$

$$f_{41}(c,x) = \begin{cases} 1, & if\ \ w_{+1} = \text{“?”}, c = S \\ 0, & otherwise \end{cases} \qquad f_{34}(c,x) = \begin{cases} 1, & if\ \ w_{+1} = \text{“?”}, c = A \\ 0, & otherwise \end{cases}$$

$\lambda_{11} = 0.3, \lambda_{21} = 0.4, \lambda_{31} = 0.1, \lambda_{41} = 0.2$

$\lambda_{12} = 0, \lambda_{22} = 0.2, \lambda_{32} = 0, \lambda_{42} = 0.1.$

$$P(S|\text{велики}) = \frac{e^{0.3+0.1+0.4}}{e^{0.3+0.1+0.4}+e^{0.2}}$$

$$P(A|\text{велики}) = \frac{e^{0.2}}{e^{0.3+0.1+0.4}+e^{0.2}}$$

$$P(S|\text{велики}) > P(A|\text{велики})$$

Марковская модель максимальной энтропии

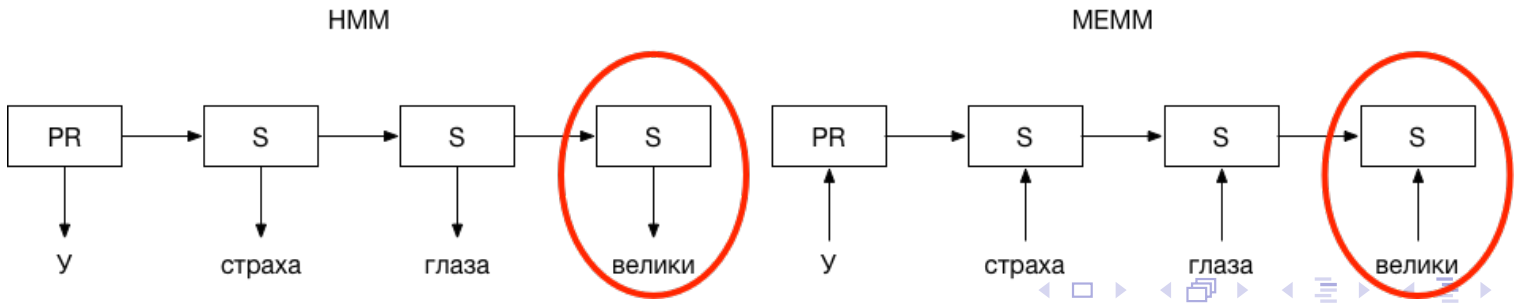
По аналогии с HMM и MaxEnt:

$$P(Q|O) = \prod_{i=1}^{n} P(q|q_{i-1}, o_i)$$

$$P(q|q', o) = \frac{e^{\sum_i w_i f_i(o,q)}}{Z(o, q')}$$

Сравнение HMM и MEMM

1. HMM и MEMM моделируют последовательности: существуют скрытые состояния (частеречные теги), порождающие наблюдения (слова). По последовательности наблюдений требуется определить, какие скрытые состояния их породили;
2. Для декодирования HMM и MEMM используется алгоритмы Витерби, для обучения – EM алгоритм;
3. MEMM позволяет ввести дополнительные индикаторные признаки, поэтому может считаться расширением HMM;
4. HMM – генеративная модель и моделирует $P(O, Q)$, MEMM – дискриминативная и моделирует $P(Q|O)$, что и требуется для декодирования;
5. В MEMM используется локальная нормировка на Z и преимущество получают состояния с меньшей энтропией – меньшим числом переходов, т.н. “label bias problem”.

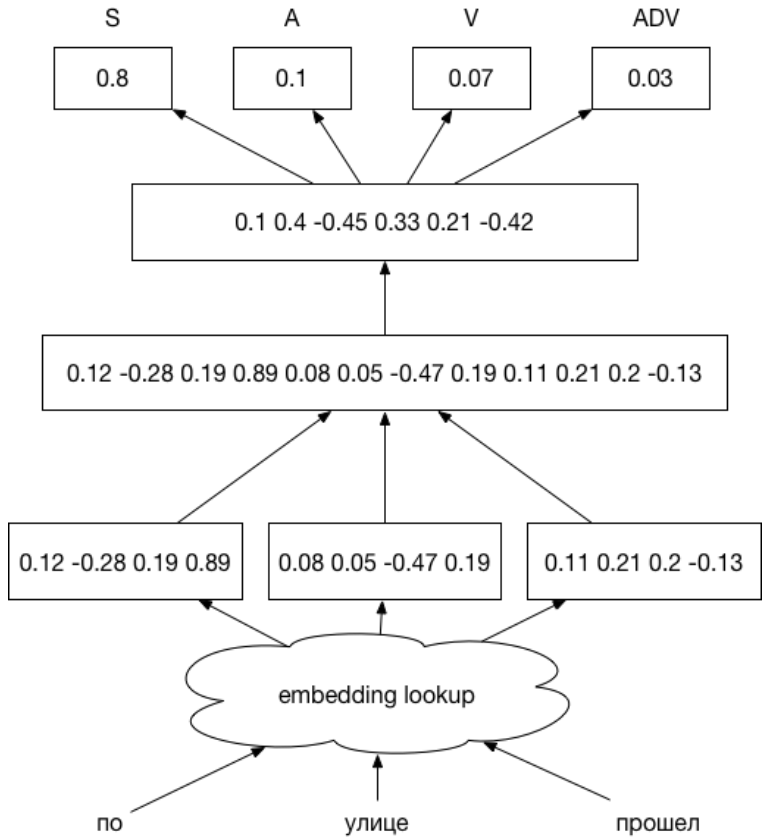


1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

SENNA

Простая архитектура нейронной сети:

- Выбрать слово и определить его контекст (одно-два слова слева и справа)
- Найти векторные представления слова и контекста (например, 100-мерные вектора SGNS)
- Конкатенировать три (или пять) найденных векторов и передать на скрытый слой
- Функция активации скрытого слоя: $h = \tanh(W_1 x + b)$
- `softmax` на выходном слое



Если предобученных векторов слов нет, инициализировать их случайным образом и обучить во время обучения всей нейронной сети

1. Введение
2. Русский язык
3. Основные задачи
4. Основные подходы
   - Алгоритм Портера [Porter, 2001]
   - Поиск в словаре
   - Скрытые цепи Маркова
   - Марковская модель максимальной энтропии
   - SENNA
5. Современные задачи

Современные задачи

1. Morphological reinflection [Cotterell, 2016]: поставить слово в определенную форму
   `http://ryancotterell.github.io/sigmorphon2016/`
2. Решение трудностей, специфичных для конкретных языков:
   - В немецком: составные слова (например, Aktivierungsenergie – энергия активации)
   - В арабском: диглоссия, гибкая пунктуация
   - В иврите: отсутствие огласовок
   - В русском: плавающее ударение

1. Segalovich, Ilya. "A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine." In MLMTA, pp. 273-280. 2003.
2. Porter, Martin F. "Snowball: A language for stemming algorithms." (2001).
3. Brants, Thorsten. "TnT: a statistical part-of-speech tagger." In Proceedings of the sixth conference on Applied natural language processing, pp. 224-231. Association for Computational Linguistics, 2000.
4. McCallum, Andrew, Dayne Freitag, and Fernando C.N. Pereira. "Maximum Entropy Markov Models for Information Extraction and Segmentation." In ICML, vol. 17, pp. 591-598. 2000.
5. Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. "Feature-rich part-of-speech tagging with a cyclic dependency networ." In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pp. 173-180. Association for Computational Linguistics, 2003.
6. Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch" Journal of Machine Learning Research 12, no. Aug (2011): 2493-2537.
7. Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. "The SIGMORPHON 2016 shared task—morphological reinflection." In Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 10-22. 2016.