Математические модели в морфологии

Условные случайные поля. Система UDPipe.

Алексей Андреевич Сорокин

спецкурс, ОТИПЛ МГУ,
осенний семестр 2017–2018 учебного года
07 ноября 2017 г.

Определение наилучшей последовательности

- Морфологический разбор сводится к нахождению наилучшей последовательности состояний.
- Состояния нельзя находить полным перебором.

Определение наилучшей последовательности

- Морфологический разбор сводится к нахождению наилучшей последовательности состояний.
- Состояния нельзя находить полным перебором.
- Логарифм вероятности выходной последовательности:

$$\log p(\mathbf{q}|\mathbf{w}) = -\log Z(\mathbf{w}) + \sum_{t=1}^{n} \sum_{k} \theta_k f_k(q_t, q_{t-1}, \mathbf{w}, t)$$

- Логарифм вероятности начала последовательности:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = -\log Z(\mathbf{w}) + \sum_{t=1}^{m} \sum_{k} \theta_k f_k(q_t, q_{t-1}, \mathbf{w}, t)$$

Определение наилучшей последовательности

- Морфологический разбор сводится к нахождению наилучшей последовательности состояний.
- Состояния нельзя находить полным перебором.
- Логарифм вероятности выходной последовательности:

$$\log p(\mathbf{q}|\mathbf{w}) = -\log Z(\mathbf{w}) + \sum_{t=1}^{n} \sum_{k} \theta_k f_k(q_t, q_{t-1}, \mathbf{w}, t)$$

- Логарифм вероятности начала последовательности:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = -\log Z(\mathbf{w}) + \sum_{t=1}^{m} \sum_{k} \theta_k f_k(q_t, q_{t-1}, \mathbf{w}, t)$$

- Формула пересчёта вероятностей:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = \log p(\mathbf{q}_{1,m-1}|\mathbf{w}) + \sum_{k} \theta_k f_k(q_m, q_{m-1}, \mathbf{w}, m)$$

Пересчёт вероятностей

- Формула пересчёта вероятностей:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = \log p(\mathbf{q}_{1,m-1}|\mathbf{w}) + \sum_k \theta_k f_k(q_m, q_{m-1}, \mathbf{w}, m)$$

Пересчёт вероятностей

- Формула пересчёта вероятностей:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = \log p(\mathbf{q}_{1,m-1}|\mathbf{w}) + \sum_k \theta_k f_k(q_m, q_{m-1}, \mathbf{w}, m)$$

- $L(q_{m-1}, q_m, m)$ — штраф за переход из q_{m-1} в q_m в момент m.
- Штраф не зависит от предыдущего пути (только от последнего состояния).

Пересчёт вероятностей

- Формула пересчёта вероятностей:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = \log p(\mathbf{q}_{1,m-1}|\mathbf{w}) + \sum_k \theta_k f_k(q_m, q_{m-1}, \mathbf{w}, m)$$

- $L(q_{m-1}, q_m, m)$ — штраф за переход из q_{m-1} в q_m в момент m.
- Штраф не зависит от предыдущего пути (только от последнего состояния).
- Можно запомнить последнее состояние и применить алгоритм Витерби (почти так же, как в марковских моделях).

Пересчёт вероятностей

- Формула пересчёта вероятностей:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = \log p(\mathbf{q}_{1,m-1}|\mathbf{w}) + \sum_k \theta_k f_k(q_m, q_{m-1}, \mathbf{w}, m)$$

- $L(q_{m-1}, q_m, m)$ — штраф за переход из q_{m-1} в q_m в момент m.
- Штраф не зависит от предыдущего пути (только от последнего состояния).
- Можно запомнить последнее состояние и применить алгоритм Витерби (почти так же, как в марковских моделях).
- Введём частичные штрафы $\alpha_{m,i}$:

$$\alpha_{m,i} = \max_{\mathbf{q_{1,m-1}}, q_m = i} \log p(\mathbf{q}_{1,m}|\mathbf{w})$$

Пересчёт вероятностей

- Формула пересчёта вероятностей:

$$\log p(\mathbf{q}_{1,m}|\mathbf{w}) = \log p(\mathbf{q}_{1,m-1}|\mathbf{w}) + \sum_k \theta_k f_k(q_m, q_{m-1}, \mathbf{w}, m)$$

- $L(q_{m-1}, q_m, m)$ — штраф за переход из q_{m-1} в q_m в момент m.
- Штраф не зависит от предыдущего пути (только от последнего состояния).
- Можно запомнить последнее состояние и применить алгоритм Витерби (почти так же, как в марковских моделях).
- Введём частичные штрафы $\alpha_{m,i}$:

$$\alpha_{m,i} = \max_{\mathbf{q}_{1,m-1}, q_m = i} \log p(\mathbf{q}_{1,m}|\mathbf{w})$$

- Формула пересчёта:

$$\begin{aligned} \alpha_{m+1,i} &= \max_j \alpha_{m,j} + L(j, i, m+1) \\ L(j, i, m+1) &= \sum_k \theta_k f_k(q_i, q_j, \mathbf{w}, m+1) \end{aligned}$$

Декодирование оптимальной последовательности

Нахождение оптимальной выходной последовательности:

- Вычислить частичные штрафы и обратные ссылки:

$$\begin{aligned}
\alpha_{0,j} &= [\![j = 0]\!], \\
\alpha_{m,i} &= \max_j \alpha_{m,j} + \sum_k \theta_k f_k(j, i, \mathbf{w}, m), \ m = 1, \ldots, n, \\
\delta_{m,i} &= \operatorname{argmax}_j \alpha_{m,j} + \sum_k \theta_k f_k(j, i, \mathbf{w}, m)
\end{aligned}$$

Декодирование оптимальной последовательности

Нахождение оптимальной выходной последовательности:

- Вычислить частичные штрафы и обратные ссылки:

$$\begin{aligned} \alpha_{0,j} &= [\![j = 0]\!], \\ \alpha_{m,i} &= \max_j \alpha_{m,j} + \sum_k \theta_k f_k(j, i, \mathbf{w}, m), \ m = 1, \ldots, n, \\ \delta_{m,i} &= \operatorname{argmax}_j \alpha_{m,j} + \sum_k \theta_k f_k(j, i, \mathbf{w}, m) \end{aligned}$$

- Восстановить последовательность по обратным ссылкам:

$$\begin{aligned} q(n) &= \operatorname{argmax}_i A_{n,i} \\ q(t-1) &= \delta_{t,q(t)} \end{aligned}$$

- По состояниям восстанавливается разбор.

Декодирование оптимальной последовательности

- Сложность декодирования: $L * M * R * K$.
  - L — длина последовательности,
  - M — максимальное число “активных” состояний,
  - R — количество способов продолжить состояние,
  - K — число признаков.

Декодирование оптимальной последовательности

- Сложность декодирования: $L * M * R * K$.
  - L — длина последовательности,
  - M — максимальное число “активных” состояний,
  - R — количество способов продолжить состояние,
  - K — число признаков.
- Если состояние — энграмма порядка m, то $M \sim R^m$, где R — максимальная степень неоднозначности слова.

Декодирование оптимальной последовательности

- Сложность декодирования: $L * M * R * K$.
  - L — длина последовательности,
  - M — максимальное число “активных” состояний,
  - R — количество способов продолжить состояние,
  - K — число признаков.
- Если состояние — энграмма порядка m, то $M \sim R^m$, где R — максимальная степень неоднозначности слова.
- Число признаков тоже растёт экспоненциально по m.

- Следствие: $m > 2$ нереализуемо на практике, с $m = 2$ проблемы для языков с развитой морфологией.

Декодирование оптимальной последовательности

- Сложность декодирования: $L * M * R * K$.
  - L — длина последовательности,
  - M — максимальное число “активных” состояний,
  - R — количество способов продолжить состояние,
  - K — число признаков.
- Если состояние — энграмма порядка m, то $M \sim R^m$, где R — максимальная степень неоднозначности слова.
- Число признаков тоже растёт экспоненциально по m.
- Сложность обучения: $T*N*C_0$, где C_0 — сложность декодирования, T — число эпох обучения, N — размер обучающей выборки.
- Следствие: $m > 2$ нереализуемо на практике, с $m = 2$ проблемы для языков с развитой морфологией.
- Часто CRF реализуют иерархически: сначала грубая классификация (части речи), потом более точная.

Недостатки CRF

- Преимущества CRF:
  - Большая гибкость по сравнению с марковскими моделями.
  - Локальные признаки произвольной природы (в том числе лексические).

Недостатки CRF

- Преимущества CRF:
  - Большая гибкость по сравнению с марковскими моделями.
  - Локальные признаки произвольной природы (в том числе лексические).
- Недостатки CRF:
  - Большие затраты (время и память) на обучение.
  - Большое количество признаков (в том числе избыточных).

Недостатки CRF

- Преимущества CRF:
  - Большая гибкость по сравнению с марковскими моделями.
  - Локальные признаки произвольной природы (в том числе лексические).
- Недостатки CRF:
  - Большие затраты (время и память) на обучение.
  - Большое количество признаков (в том числе избыточных).
  - Невозможность учитывать удалённый контекст.

Недостатки CRF

- Преимущества CRF:
  - Большая гибкость по сравнению с марковскими моделями.
  - Локальные признаки произвольной природы (в том числе лексические).
- Недостатки CRF:
  - Большие затраты (время и память) на обучение.
  - Большое количество признаков (в том числе избыточных).
  - Невозможность учитывать удалённый контекст.
- CRF применимы к произвольной задаче разметке последовательно распознавание именованных сущностей, разбиение на морфемы и т. д.

Недостатки CRF

- Преимущества CRF:
  - Большая гибкость по сравнению с марковскими моделями.
  - Локальные признаки произвольной природы (в том числе лексические).
- Недостатки CRF:
  - Большие затраты (время и память) на обучение.
  - Большое количество признаков (в том числе избыточных).
  - Невозможность учитывать удалённый контекст.
- CRF применимы к произвольной задаче разметке последовательно распознавание именованных сущностей, разбиение на морфемы и т. д.
- В последние годы уступают нейронным моделям.

Система UDPipe

- Наиболее эффективная из доступных в открытом доступе систем — система проекта UDPipe (Карлов университет, Прага, UFAL).

Система UDPipe

- Наиболее эффективная из доступных в открытом доступе систем — система проекта UDPipe (Карлов университет, Прага, UFAL).
- Основана на условных случайных полях.
- Обучается с помощью усреднённого персептрона.

Система UDPipe

- Наиболее эффективная из доступных в открытом доступе систем — система проекта UDPipe (Карлов университет, Прага, UFAL).
- Основана на условных случайных полях.
- Обучается с помощью усреднённого персептрона.
- Использует развёрнутые шаблоны признаков (изначально разрабатывалась для чешского).

Система UDPipe

- Наиболее эффективная из доступных в открытом доступе систем — система проекта UDPipe (Карлов университет, Прага, UFAL).
- Основана на условных случайных полях.
- Обучается с помощью усреднённого персептрона.
- Использует развёрнутые шаблоны признаков (изначально разрабатывалась для чешского).
- Предсказываются несколько меток: UPOSTAG (часть речи), XPOSTAG (детализованная часть речи: наклонение глагола, одушевлённость сущ-го, ...), FEATS (признаковое описание).

Система UDPipe

- Наиболее эффективная из доступных в открытом доступе систем — система проекта UDPipe (Карлов университет, Прага, UFAL).
- Основана на условных случайных полях.
- Обучается с помощью усреднённого персептрона.
- Использует развёрнутые шаблоны признаков (изначально разрабатывалась для чешского).
- Предсказываются несколько меток: UPOSTAG (часть речи), XPOSTAG (детализованная часть речи: наклонение глагола, одушевлённость сущ-го, ...), FEATS (признаковое описание).
- Лемматизатор использует ту же математическую модель, но предсказывает шаблон словоизменения.

Признаковое описание для английского языка

Context predicting whole tag	
Tags	Previous tag Previous two tags First letter of previous tag
Word forms	Current word form Previous word form Previous two word forms Following word form Following two word forms Last but one word form
Current word affixes	Prefixes of length 1-9 Suffixes of length 1-9
Current word features	Contains number Contains dash Contains upper case letter

Всего $\sim 2,0$ млн. признаков.

Признаковое описание для чешского языка

Признак	Описание
t_0	текущая метка
$t_{-1}\,t_0$	биграмма меток
t_{-2}; $t_{-2}\,t_{-1}\,t_0$	триграмма меток
$w_{-2}, w_{-1}, w_0, w_1, w_2$	слово и его контекст
	позиция слова
$l(V_L), t(V_L)$	метка и лемма ближайшего глагола слева
$l(V_R), t(V_R)$	метка и лемма ближайшего глагола справа
	капитализация

Всего $\sim 8,4$ млн. признаков.

Условные случайные поля

- Условные случайные поля могут решать любую задачу, сводящуюся к разметке последовательностей.

Условные случайные поля

- Условные случайные поля могут решать любую задачу, сводящуюся к разметке последовательностей.
- Пример: выделение именованных сущностей.
- Границы групп задаются метками B(начало), M(середина), E(конец), S(однословная группа), O(вне групп).

Андрей	Николаевич	Колмогоров	работал	в	Московском	Университете
B-PERS	M-PERS	E-PERS	O	O	B-ORG	E-ORG

Условные случайные поля

- Условные случайные поля могут решать любую задачу, сводящуюся к разметке последовательностей.
- Пример: выделение именованных сущностей.
- Границы групп задаются метками B(начало), M(середина), E(конец), S(однословная группа), O(вне групп).

Андрей	Николаевич	Колмогоров	работал	в	Московском	Университете
B-PERS	M-PERS	E-PERS	O	O	B-ORG	E-ORG

- Основные признаки: капитализация, частеречные метки, сами слова (в окне ширины 5).

Лемматизация: постановка задачи

- Пока мы что решали только задачу нахождения морфологической метки.
- Однако чаще всего нужна ещё и начальная форма (извлечение информации).

Лемматизация: постановка задачи

- Пока мы что решали только задачу нахождения морфологической метки.
- Однако чаще всего нужна ещё и начальная форма (извлечение информации).
- Если словоформа словарная и известна морфологическая метка, то достаточно просто посмотреть в словаре.
- Исключений очень мало:

$$\textit{лечу}+\mathrm{V}+\mathrm{Pres}+\mathrm{Sg}+1 \mapsto \text{лечить / лететь.}$$

Лемматизация: постановка задачи

- Пока мы что решали только задачу нахождения морфологической метки.
- Однако чаще всего нужна ещё и начальная форма (извлечение информации).
- Если словоформа словарная и известна морфологическая метка, то достаточно просто посмотреть в словаре.
- Исключений очень мало:

$$\textit{лечу}+\mathrm{V}+\mathrm{Pres}+\mathrm{Sg}+1 \mapsto \text{лечить / лететь.}$$

- Что делать с несловарными словами?

Лемматизация: постановка задачи

- Пока мы что решали только задачу нахождения морфологической метки.
- Однако чаще всего нужна ещё и начальная форма (извлечение информации).
- Если словоформа словарная и известна морфологическая метка, то достаточно просто посмотреть в словаре.
- Исключений очень мало:

$$\textit{лечу}+\mathrm{V}+\mathrm{Pres}+\mathrm{Sg}+1 \mapsto \text{лечить / лететь.}$$

- Что делать с несловарными словами?
- Для них априори неизвестен даже список возможных меток и оценки их вероятностей.

Морфологический анализ как задача классификации

- Задача: дано несловарное слово w, найти наиболее вероятную морфологическую метку $t(w)$.
- Для марковских моделей и CRF нужно чуть больше: распределение вероятностей $p(t|w)$ для возможных меток t.

Морфологический анализ как задача классификации

- Задача: дано несловарное слово w, найти наиболее вероятную морфологическую метку $t(w)$.
- Для марковских моделей и CRF нужно чуть больше: распределение вероятностей $p(t|w)$ для возможных меток t.
- Можно решать как задачу классификации:
  - Описать каждое слово вектором признаков. Морфологическая метка — предсказываемый класс.
  - Словарные/корпусные слова — обучающая выборка.

Морфологический анализ как задача классификации

- Стандартный набор признаков: суффиксы слова до m символов ($m \sim 5 - 8$), префиксы (если нужно), капитализация.
- Каждый признак кодируется двоичной схемой:

	-’	-а	-ть	-ль	-ка	-ать	...	Класс
играть	1	0	1	0	0	1	...	VERB, Mood=Inf,...
бегать	1	0	1	0	0	1	...	VERB, Mood=Inf,...
плыть	1	0	1	0	0	0	...	VERB, Mood=Inf,...
кровать	1	0	1	0	0	1	...	NOUN, case=Nom,...
тень	1	0	0	0	0	0	...	NOUN, case=Nom,...
рука	0	1	0	0	1	0	...	NOUN, case=Nom,...
броска	0	1	0	0	1	0	...	NOUN, case=Gen,...

Морфологический анализ как задача классификации

- Стандартный набор признаков: суффиксы слова до m символов ($m \sim 5 - 8$), префиксы (если нужно), капитализация.
- Каждый признак кодируется двоичной схемой:

	-’	-а	-ть	-ль	-ка	-ать	...	Класс
играть	1	0	1	0	0	1	...	VERB, Mood=Inf,...
бегать	1	0	1	0	0	1	...	VERB, Mood=Inf,...
плыть	1	0	1	0	0	0	...	VERB, Mood=Inf,...
кровать	1	0	1	0	0	1	...	NOUN, case=Nom,...
тень	1	0	0	0	0	0	...	NOUN, case=Nom,...
рука	0	1	0	0	1	0	...	NOUN, case=Nom,...
броска	0	1	0	0	1	0	...	NOUN, case=Gen,...

- После этого применим любой алгоритм машинного обучения (метод опорных векторов, персептрон, логистическая регрессия, деревья решений...)

Лемматизация как задача классификации

- Данный подход также позволяет осуществлять лемматизацию.
- Для этого надо предсказывать тип преобразования от словоформы к лексеме.

Лемматизация как задача классификации

- Данный подход также позволяет осуществлять лемматизацию.
- Для этого надо предсказывать тип преобразования от словоформы к лексеме.
- Простейший тип преобразования — изменения суффикса:

$$\text{коленями} \mapsto \text{колено}$$
$$\text{X+ями} \mapsto \text{X+о}$$

Лемматизация как задача классификации

- Данный подход также позволяет осуществлять лемматизацию.
- Для этого надо предсказывать тип преобразования от словоформы к лексеме.
- Простейший тип преобразования — изменения суффикса:

$$\begin{aligned} \text{коленями} &\mapsto \text{колено} \\ \text{X+ями} &\mapsto \text{X+о} \end{aligned}$$

- Можно добавить изменения префиксов:

$$\begin{aligned} \text{наикраснейший} &\mapsto \text{красный} \\ \text{наи+X+ейший} &\mapsto \text{X+ый} \end{aligned}$$

Лемматизация как задача классификации

- Данный подход также позволяет осуществлять лемматизацию.
- Для этого надо предсказывать тип преобразования от словоформы к лексеме.
- Простейший тип преобразования — изменения суффикса:

$$\text{коленями} \mapsto \text{колено}$$
$$\text{X+ями} \mapsto \text{X+о}$$

- Можно добавить изменения префиксов:

$$\text{наикраснейший} \mapsto \text{красный}$$
$$\text{наи+X+ейший} \mapsto \text{X+ый}$$

- Иногда изменения не описываются “непрерывным” шаблоном:

$$\text{vuelvo} \mapsto \text{volver}$$
$$\text{броска} \mapsto \text{бросок}$$
$$\text{yiktubun} \mapsto \text{kataba}$$

Абстрактные парадигмы

- Можно ввести “разрывные” шаблоны словоизменения:

$$\begin{aligned} \text{vuelvo} &\mapsto \text{volver} \\ 1{+}\text{ue}{+}2{+}\text{o} &\mapsto 1{+}\text{o}{+}2{+}\text{er} \\ \text{броска} &\mapsto \text{бросок} \\ 1{+}2{+}\text{a} &\mapsto 1{+}\text{o}{+}2 \\ \text{yiktubun} &\mapsto \text{kataba} \\ \text{yi}{+}1{+}2{+}\text{u}{+}3{+}\text{un} &\mapsto 1{+}\text{a}{+}2{+}\text{a}{+}3{+}\text{a} \end{aligned}$$

- Переменная часть шаблона (основа) — наибольшая общая подпоследовательность (НОП) начальной и производной словоформы.

Абстрактные парадигмы

- Можно ввести “разрывные” шаблоны словоизменения:

$$\begin{aligned} \text{vuelvo} &\mapsto \text{volver} \\ 1{+}\text{ue}{+}2{+}\text{o} &\mapsto 1{+}\text{o}{+}2{+}\text{er} \\ \text{броска} &\mapsto \text{бросок} \\ 1{+}2{+}\text{а} &\mapsto 1{+}\text{о}{+}2 \\ \text{yiktubun} &\mapsto \text{kataba} \\ \text{yi}{+}1{+}2{+}\text{u}{+}3{+}\text{un} &\mapsto 1{+}\text{a}{+}2{+}\text{a}{+}3{+}\text{a} \end{aligned}$$

- Переменная часть шаблона (основа) — наибольшая общая подпоследовательность (НОП) начальной и производной словоформы.
- Лингвистическое обоснование:

> *“Основу любой словоформы можно представить как состоящую из двух частей: неизменяемой и изменяемой, то есть такой, которой хотя бы в одной из прочих словоформ парадигмы соответствует некоторая иная цепочка букв”.*
>
> (А. А. Зализняк, “Русское именное словоизменение”)

Абстрактные парадигмы

- Алгоритм извлечения абстрактной парадигмы словоизменения:
  - Найти НОП: *yiktbun*, *kataba* $\mapsto$ k-t-b
  - Выделить в словах элементы НОП:

    yiktubun, kataba

Абстрактные парадигмы

- Алгоритм извлечения абстрактной парадигмы словоизменения:
  - Найти НОП: *yiktbun*, *kataba* $\mapsto$ k-t-b
  - Выделить в словах элементы НОП:

    yiktubun, kataba
  - Заменить i-ый элемент НОП на переменную x_i:

    $\mathrm{k} \mapsto x_1$, $\mathrm{t} \mapsto x_2$, $\mathrm{b} \mapsto x_3$,

    yi $x_1 x_2$ u x_3 un, a x_2 a x_3 a

Абстрактные парадигмы

- Алгоритм извлечения абстрактной парадигмы словоизменения:
  - Найти НОП: *yiktbun*, *kataba* $\mapsto$ k-t-b
  - Выделить в словах элементы НОП:

    yiktubun, kataba
  - Заменить i-ый элемент НОП на переменную x_i:

    $\mathrm{k} \mapsto x_1$, $\mathrm{t} \mapsto x_2$, $\mathrm{b} \mapsto x_3$,

    yi $x_1 x_2$ u x_3 un, a x_2 a x_3 a
  - Вернуть получившийся шаблон
- Проблема: как находить НОП.