Практическое задание 2
Приложение: постобработка результата разметки

курс «Математические методы анализа текстов»

Модель HMM и её связь с CRF

Для обучения нейросетевой модели разметки (например, BiLSTM) используется поэлементная кросс-энтропия. При использовании на этапе инференса функции $\arg\max$ для получения выходной последовательности, мы не можем гарантировать согласованность предсказаний. Для согласованности необходимо вместо $\arg\max$ использовать другие функции получения предсказаний.

В модели CRF (conditional random field) для получения предсказаний используется алгоритм Витерби. Напомним, что модель CRF модлирует вероятность последовательности $y \in Y^n$ при условии $x \in X^n$ линейной моделью с вектором весов $w \in \mathbb{R}^d$, которая после некоторых преобразований записывается следующим образом:

$$p(y|x,w) = \frac{1}{Z(x,w)} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{d} w_j f_j(y_{i-1}, y_i, x_i, i)\right) = \frac{1}{Z(x,w)} \exp\left(\sum_{i=1}^{n} G_{x,i}[y_{i-1}, y_i]\right)$$

Модель CRF принадлежит к семейству графичеких моделей (probabilistic graphical models). Ещё одна модель из этого семейства - HMM (hidden markov model), моделирует совместную вероятность последовательностей x и y, используя частотные оценки вероятностей:

$$p(x,y) = p(y)p(x|y) = p(y_1)\prod_{i=2}^{n} p(y_i|y_{i-1})\prod_{i=1}^{n} p(x_i|y_i)$$

Если принять, что y_0 означает специальный токен <START>, встречающийся только в начале последовательности, модель можно записать так:

$$p(x,y) = \prod_{i=1}^{n} p(y_i|y_{i-1})\prod_{i=1}^{n} p(x_i|y_i)$$

Параметры модели и оценки, получаемые на них через метод максимального правдоподобия:

- матрица переходов $A \in \mathbb{R}^{|Y|\times|Y|}$

$$A_{vu} = \frac{\sum_y \sum_{i=2}^{|y|} \mathbb{I}[y_i = v, y_{i-1} = u]}{\sum_y \sum_{i=2}^{|y|} \mathbb{I}[y_{i-1} = u]}$$

- матрица выходных вероятностей $B \in \mathbb{R}^{|X|\times|Y|}$

$$B_{zu} = \frac{\sum_{y,x} \sum_{i=1}^{|y|} \mathbb{I}[x_i = z, y_i = u]}{\sum_{y,x} \sum_{i=1}^{|y|} \mathbb{I}[y_i = u]}$$

- матрица начальных вероятностей $C \in \mathbb{R}^{|Y|}$:

$$C_v = \frac{\sum_y \mathbb{I}[y_1 = v]}{\sum_y 1}$$

  Значения C можно хранить в матрице A, введя специальный токен <START>, хранить в отдельной матрице или вообще использовать равномерное приближение.

Модель HMM является генеративной, однако если использовать её только для получения выходной последовательности y по входной x, окажется, что модель является частным случаем CRF:

$$p(y|x) = \frac{1}{Z(x)} p(y)p(x|y) = \frac{1}{Z(x)} \exp(\log p(y) + \log p(x|y)) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{n} (\log p(y_i|y_{i-1}) + \log p(x_i|y_i))\right)$$

Это легко показать, если ввести признаки:

$f_1(y_{i-1}, y_i, x_i, i) = \log B[z = x_i, u = y_i]$

$f_2(y_{i-1}, y_i, x_i, i) = \log A[v = y_i, u = y_{i-1}]\mathbb{I}[i > 1] \times \log C[v = y_i]\mathbb{I}[i = 1]$

Дополнительно, можно ввести вес на каждый из признаков. Т.к. веса всего два, их легко подобрать по отложенной выборке. Таким образом, вероятность y при условии x записывается так:

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{n} (w_1 f_1(y_{i-1}, y_i, x_i, i) + w_2 f_2(y_{i-1}, y_i, x_i, i))\right)$$

Заметим, что при таком переходе мы отошли от вероятностной постановки через совместную вероятность, но получили большую свободу за счёт возможности выбора весов. Также заметим, что такой способ задания признаков напоминает различные способы кодирования признаков, использующиеся, например, при решении задач машинного обучения ансамблями решающих деревьев.

Алгоритм Витерби

Получение выходной последовательности по входной в HMM устроено так же, как и в CRF, с помощью алгоритма Витерби. Это алгоритм динамического программирования, с помощью которого можно найти наиболее вероятную последовательность скрытых состояний модели для фиксированной последовательности слов:

$$\hat{y} = \arg\max_y p(y|x) = \arg\max_y p(x, y)$$

Определим функцию, определяющую максимальную вероятность последовательности, заканчивающейся на i-ой позиции в состоянии k:

$$\delta(k, i) = \max_{y_1, \dots y_{i-1}} p(x_1, \dots x_i, y_1, \dots y_i = k)$$

Тогда $\max_k \delta(k, n)$ — максимальная вероятность всей последовательности. А состояния, на которых эта вероятность достигается — ответ задачи. Алгоритм Витерби заключается в последовательном пересчете функции $\delta(k, i)$ по формуле:

$$\delta(k, i) = \max_m \delta(m, i-1) p(y_i = k|y_{i-1} = m) p(x_i|y_i = k)$$

Аналогично пересчитывается функция, определяющая, на каком состоянии этот максимум достигается:

$$s(k, i) = \arg\max_m \delta(m, i-1) p(y_i = k|y_{i-1} = m) p(x_i|y_i = k)$$

На практике это означает заполнение двумерных массивов размерности: (длина последовательности) $\times$ (количество возможных состояний). Когда массивы заполнены, $\arg\max_k \delta(k, n)$ говорит о последнем состоянии. Начиная с него можно восстановить все состояния по массиву s. Осталось уточнить, как стартовать последовательный пересчет (чем заполнить первый столбец массива вероятностей):

$$\delta(k, 1) = p(t_1 = k) p(x_1|t_1 = k)$$

При расчёте на компьютере лучше перейти от произведения вероятностей к сумме их логарифмов.

Необучаемый пост-процессинг разметки

Модифицируем описанную выше модель, чтобы использовать её в качестве пост-процессинга. Будем вместо частотной оценки $\log B[z = x_i, u = y_i]$ использовать выходы нашей модели. Пусть после применения модели к последовательности x мы на выходе получаем логарифмы вероятностей $S \in \mathbb{R}^{n \times |Y|}$ (после применения функции $\log \mathrm{softmax}$). Тогда признак будет задаваться следующим образом:

$$f_1(y_{i-1}, y_i, x_i, i) = S_{i, y_i}$$

Почему мы называем пост-процессинг необучаемым? Потому что в отличие от LSTM-CRF модели, рассмотренной на лекции, данный способ не требует совместного обучения нейросети и модели пост-процессинга. К любой обученной нейросети можно применить такой пост-процессинг. Тем не менее, в каком-то смысле обучение всё же происходит — необходимо подсчитать веса матриц A и C по обучающему корпусу и подобрать значения весов модели w_1 и w_2 по кросс-валидации.

Обучаемый пост-процессинг разметки

Обучаемый пост-процессинг был рассмотрен на лекции. Он сводится к использованию модели CRF поверх выходов нейросети. В модели CRF используются два типа признаков:

- $|Y| \times |Y|$ признаков, учитывающих выход модели:

$$\phi_{uv}(y_{i-1}, y_i, x_i) = \mathbb{I}[y_{i-1} = u]\mathbb{I}[y_i = v]S_{i,y_i}$$

- $|Y| \times |Y|$ признаков, учитывающих связь меток:

$$\psi_{uv}(y_{i-1}, y_i) = \mathbb{I}[y_{i-1} = u]\mathbb{I}[y_i = v],$$

Обозначив линейные коэффициенты модели $w(u, v)$ и $a(u, v)$, после преобразований выражения $G_{x,i}[y_{i-1}, y_i]$ получим:

$$G_{x,i}[y_{i-1}, y_i] = \sum_{u \in Y} \sum_{v \in Y} (w(u, v)\phi_{uv}(y_{i-1}, y_i, x_i) + a(u, v)\psi_{uv}(y_{i-1}, y_i)) =$$
$$= \sum_{u \in Y} \sum_{v \in Y} \mathbb{I}[y_{i-1} = u]\mathbb{I}[y_i = v](w(u, v)S_{i,y_i} + a(u, v)) = w(y_{i-1}, y_i)S_{i,y_i} + a(y_{i-1}, y_i)$$

Полученное выражение хорошо подходит и для вычисления оптимальной выходной последовательности в алгоритме Витерби и для вычисления промежуточных векторов в методе вперёд-назад, используемом для подсчёта градиентов функционала. В обучаемом пост-процессинге нейросеть и CRF обучаются в едином пайплайне, CRF по сути выступает отдельным слоём нейросети.