

Вопрос 1

Выполнен

Баллов: 1,00 из 1,00

Изучая .csv файл с **оригинальной разметкой** рецептов можно наткнуться в частности на такие примеры:

1^{QTY} teaspoon^{UNIT} baking powder^{NAME}

5^{QTY} cups^{UNIT} kosher or coarse sea^{COMMENT} salt^{NAME}

1^{QTY} cup^{UNIT} plain lowfat^{COMMENT} yogurt^{NAME}

В первом примере baking powder^{NAME} (мука для выпекания) выделена в отдельную сущность, тогда как plain lowfat^{COMMENT} yogurt^{NAME} (обезжиренный йогурт без добавок) и kosher or coarse sea^{COMMENT} salt^{NAME} (кошерная или крупная морская соль) поделены на отдельные сущности.

Какие из следующих методов можно использовать для принятия решения об объединении нескольких слов в именованную сущность?

Выберите один или несколько ответов:

- ☒ a. $pmi(l, w) = \log \frac{p(w, l)}{p(w)p(l)}$
- ☒ b. Языковая модель (например, skip-gram)
- ☒ c. Частота n-граммы

Ваш ответ верный.

Вопрос 2

Выполнен

Баллов: 1,00 из 1,00

Зачем в функции predict_tags() мы делаем вызов torch.no_grad()?

Выберите один ответ:

- ☐ a. для того, чтобы можно было делать backprop не только во время обучения сети, но и на этапе предсказания тегов
- ☐ b. для перевода модели в режим eval (например, чтобы Dropout или BatchNorm слои работали в режиме eval)
- ☒ c. для деактивации autograd (без вычисления градиентов будет использоваться меньше памяти и скорость работы увеличится)
- ☐ d. для автоматического подбора оптимального размера батча

Ваш ответ верный.

Итак, мы определили нашу нейросеть [LSTM](#), задали [функцию потерь](#), и, наконец, запустим процесс обучения нейросети. Он будет происходить вживую в нашем браузере и мы сможем увидеть, как оптимизируется функция потерь. Каждые 500 шагов на график добавляется новая точка, соответствующая значению функции потерь в этот момент. Нейросеть обучается довольно быстро, потому что наш датасет сравнительно мал. Наша нейросеть закончила обучение на тренировочных данных, и теперь мы можем использовать её для предсказания тэгов на рецептах, которых она ранее никогда не видела. Для этого определим функцию `predict_tags`. Она работает следующим образом. Опять же, нейросеть (на этот раз уже обученная) получает на вход слова, находятся их индексы в словаре, они попадают в нейросеть, и мы предсказываем тэги для слов. Рассмотрим 10 реальных примеров предсказания тэгов рецептов. На экране вы увидите настоящие тэги, которые мы взяли из нашего файла, и тэги предсказанные. Мы видим, что наша нейросеть не допустила ошибок на случайно выбранных 10 рецептах, кроме вот этого рецепта, — а именно, длинный тэг "name" для последовательности из трёх слов она разбила на две части — "comment" и "name".

Вопрос 3

Выполнен

Баллов: 1,00 из 1,00

Какие действия выполняет функция `nn.NLLLoss()`?

Выберите из вариантов ответа правильную формулу:

Выберите один ответ:

- ☐ a. $loss = -\log(y)$, где y - вероятность правильного класса.
- ☐ b. $loss = -y$, где y - ненормализованная оценка для правильного класса.
- ☒ c. $loss = -y$, где y - логарифм вероятности правильного класса

Ваш ответ верный.