Основы обработки текстов

Лекция #12:
Перенос знаний, совместное обучение

Лектор: м.н.с. ИСП РАН Андрианов Иван Алексеевич

План лекции

- Перенос знаний
- Предобучение символьных представлений слов
- Предобучение контекстно-зависимых представлений слов
- Языковые модели
- Оценка качества языковых моделей
- Совместное обучение в нейронных сетях

Обучение с учителем (supervised learning)

В курсе рассматривалось обучение моделей, решающих конкретную задачу на данных из конкретного распределения (язык, предметная область, …):

$\{x_i\}$ — тренировочная выборка из распределения X
$\{y_i\}$ — метки (задают задачу)
L — ф-ция потерь (задает цель оптимизации)

ищется ф-ция f: $\Sigma_i L(f(x_i), y_i)$ минимальна (обучение)
далее для всех x_j из X y_j полагается $f(x_j)$ (предсказание)

Перенос знаний (transfer learning)

- Можно ли использовать найденную ф-цию f для других задач и других распределений? Да!
- Для той же задачи в том же распределении — supervised learning
- Для других задач в том же распределении — inductive transfer learning
- Для той же задачи в другом распределении — transductive transfer learning или domain adaptation
- Для других задач в другом распределении — unsupervised transfer learning

Мотивации переноса знаний

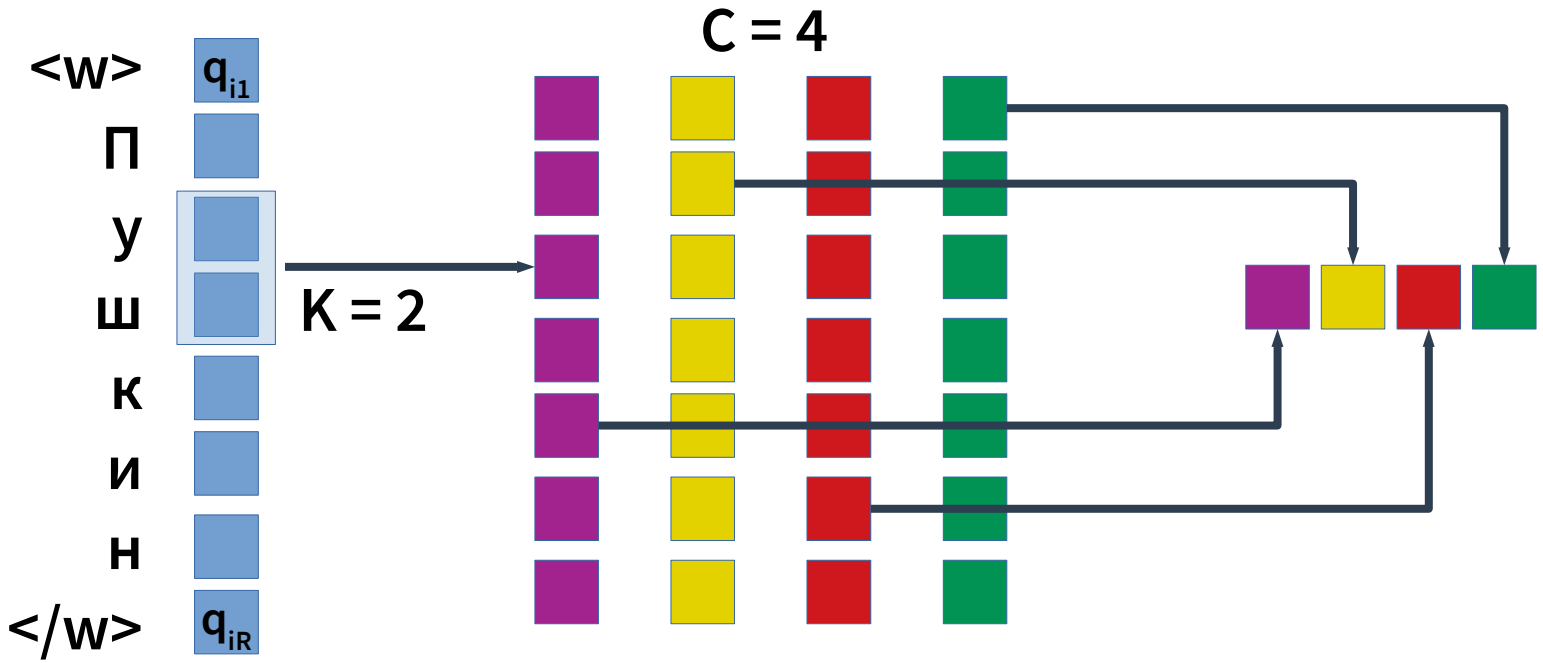
- Идеологические взаимосвязи между задачами:
  - части речи и деревья зависимостей сильно коррелируют
  - сущности и отношения коррелируют
- Наличие более объемного (репрезентативного) корпуса для другой задачи и/или предметной области:
  - корпуса для частей речи >> корпусов для сущностей
  - социо-политические корпуса >> IT корпусов
- Рассмотренные ранее модели skip-gram, GloVe и др. — это перенос знаний о совместной встречаемости слов из неразмеченного (заведомо >>, чем размеченный) корпуса

Перенос знаний в нейронных сетях

- Нейронные сети представляют собой сложные ф-ции от x_i
- На каждом уровне формируется векторное представление, например, в задаче извлечения сущностей:
  - векторное представление слов w_i
  - векторное представление слов в контексте h_i (выход BiLSTM)
- Эти представления — наборы информативных признаков
- Основной подход к переносу знаний из нейронных сетей: извлекать из сети промежуточные представления и передавать их в качестве признаков в целевой задаче

Символьные представления слов

Были рассмотрены несколько представлений слов по символам (CNN, BiLSTM, …), они обучались совместно основной задачей



Предобучение символьных представлений слов

Подобные векторные представления можно обучать по совместной встречаемости слов (skip-gram) на большом неразмеченном корпусе

[Во времена правления Ивана Грозного] из …

x_c — центральное слово

x_o — контекстное слово: $o(c) = \{c-2, c-1, c+1, c+2\}$

$v_c = CNN(x_c)$ — символьное представление слова x_c

u_o — вектор слова x_o

$J(\Theta) = \Sigma_c \Sigma_{o(c)} \text{softmax}[x_o](v_c \cdot u_o)$

Предобучение контекстно-зависимых представлений слов

- Морфологические векторные представления слов на основе модели разметки последовательности

  Соответствие второму пункту правил неочевидно

$h_i = BiLSTM(x,i)$

сущ прил сущ сущ прил

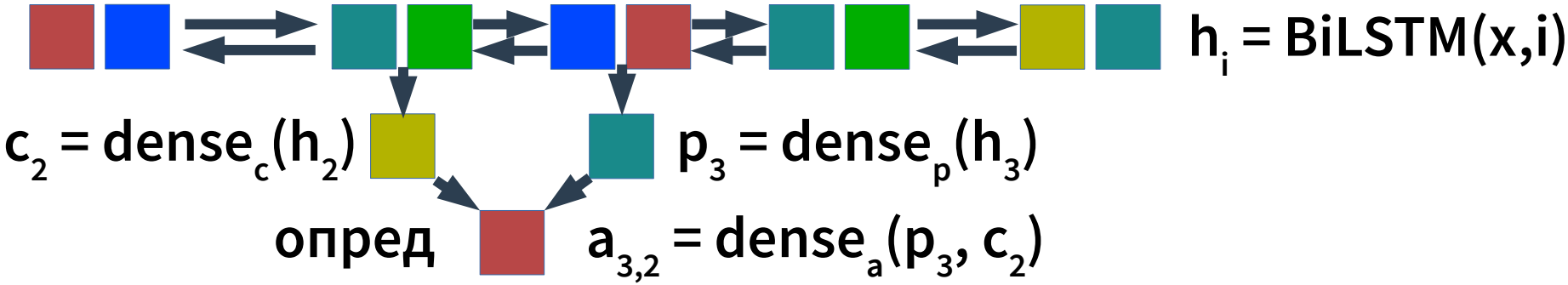
$s_i = CRF(h,i)$

- h_i содержит информацию о морфологии слова x_i в данном контексте (предложении)

Предобучение контекстно-зависимых представлений слов

- Синтаксические векторные представления слов на основе графовой модели разбора по грамматике зависимостей

  Соответствие второму пункту правил неочевидно



- h_i, p_i, c_i содержат информацию о синтаксической роли слова x_i в данном контексте (предложении)

Языковые модели (language model, LM)

Древнерусские крепости снабжались водой через тайники

- Предсказываем следующее слово: forward LM

  {Древнерусские, крепости} => снабжались

  {…, крепости, снабжались} => водой

- Предсказываем предыдущее слово: backward LM

  {…, водой, снабжались} => крепости

  {…, снабжались, крепости} => Древнерусские

- LM должна «кодировать» информацию как о синтаксисе, так и о семантике текстов

Зачем нужны языковые модели?

- Подсказка при наборе текста

  Древнерусские к...

  крепости 0.03

  коммуникации 0.01

  колонны 0.000001

  колонки 0

- Оценка моделей natural language understanding:
  не требуются размеченные данные
- Перенос знаний из больших неразмеченных корпусов

Ngramные языковые модели

- Простейший способ построения языковой модели
- Фиксируем размер ngram N (N=3)

  {Древнерусские, крепости, снабжались}

  {крепости, снабжались, водой}

  {снабжались, водой, через}

  {водой, через, тайники}
- Считаем частоты всех ngram в обучающем корпусе
- Для предсказания берем последние N-1 слов входа и ищем в словаре начинающиеся с них ngramы, частоты переводим в вероятности предсказания последнего слова ngramы

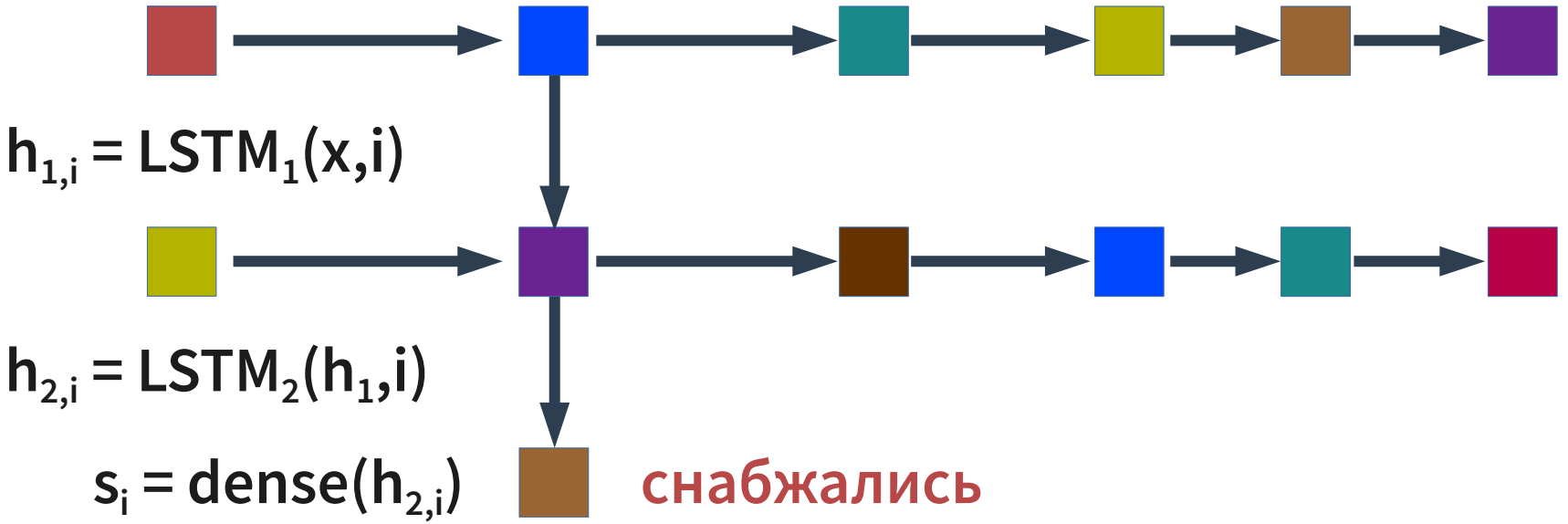
Языковые модели на базе LSTM

- Ngramные языковые модели страдают от разреженности: даже в огромном корпусе частоты ngram при N>1,2 малы
- Ngramные языковые модели не имеют доступа к контексту дальше N-1 слов

Древнерусские крепости снабжались водой через тайники

Перенос знаний из языковых моделей (ELMo)

Древнерусские крепости снабжались водой через тайники



- Двухуровневая LSTM-сеть, обученная на большом неразмеченном корпусе (1 млрд слов) содержит информацию о синтаксисе ($h_{1,i}$) и семантике ($h_{2,i}$)

Перенос знаний из языковых моделей (Flair)

- Спускаемся на уровень символов и обучаем forward / backward LSTM, предсказывающий следующий / предыдущий символ

Древнерусские<w>крепости<w>снабжались<w>водой

{Д,р,е,в,н,е,р,у,с,с} => к

{…,к,и,е,<w>,к,р,е,п,о,с} => т

- В качестве вектора для слова берем выход forward / backward LSTM на последнем / первом символе слова

Оценка качества языковых моделей

- Языковая модель предсказывает слова =>
  множество меток классификатора большое
- Стандартные меры сравнивают метки на совпадение =>
  вырождаются на задаче LM в 0
- У нас есть вероятности всех меток =>
  можно проанализировать, насколько в среднем высоки вероятности «правильных» меток
- Perplexity («удивление»; ниже — лучше):
  $2 \wedge (-(\Sigma_i \log_2 p(x_i)) / N)$

Проблемы переноса знаний

- При плохой корреляции ф-ции потерь переносимой модели с целевой задачей можно получить «шумные» признаки
- Модели, задействующие перенос знаний тяготеют к «монструозности»
- Переносимую ф-цию хотелось бы адаптировать к целевой задаче, а не применять «как есть», однако при наивном подходе это приводит к проблеме «catastrophic forgetting», т.е. удалению всех накопленных знаний из-за «странного» для переносимой модели распространения ошибки в целевой задаче

Взаимосвязи между задачами

- Между различными задачами (POS+DP, NER+RelExt) имеется смысловая связь:
  - Слово-родитель и слово-ребенок должны быть согласованы по роду, числу и падежу
  - Две сущности типа «человек» не могут находится в отношении «родился в»
- Как правило, применяется поэтапный подход: сначала решается задача «попроще», затем ее результаты подаются как «сильные» признаки в задачу «посложнее»:
  - Часть речи, род, число, падеж подаются в DP
  - Типы сущностей-аргументов подаются в RelExt

Совместное обучение

- Можно ли задействовать информацию из обучающей выборки для задачи «посложнее» при обучении модели для задачи «попроще»?
- Совместное обучение — это обучение модели, решающей одновременно несколько задач
- Зачем это нужно:
  - Повышение качества моделей (~~синергия~~ корреляция)
  - Более высокая производительность (общие вычисления)
  - Регуляризация модели для одной из задач, если обучающие данные для нее малые / шумные

Совместное обучение нейронных сетей

- Нейронные сети предоставляют гибкий механизм для обучения произвольных функций преобразования

  $y = f(x)$ — преобразование, $l = \lambda(y, \hat{y})$ — функция потерь

- В большинстве случаев y — это распределение вероятностей меток / классов, а λ — кросс-энтропия

  $y = f(x)$ – «разделяемое» преобразование

  $z_i = g_i(y)$ – преобразование для i-ой задачи

  $l_i = \lambda_i(z_i, \hat{z_i})$ — функция потерь для i-ой задачи

  $l = \Sigma_i \alpha_i l_i$ - «совместная» функция потерь

Попеременное обучение нейронных сетей

- Совместное обучение требует единого обучающего корпуса, в котором имеются метки для всех задач

  $l = \Sigma_i \alpha_i \lambda_i(g_i(f(x)), \hat{z_i})$

- Такие корпуса на практике – редкость: POS+DP, NER+RelExt
- Чтобы смягчить это ограничение, можно применять попеременное обучение: попеременно для всех i обучать сеть с помощью функции потерь: $l_i = \lambda_i(g_i(f(x)), \hat{z_i})$
- Как и в случае совместного обучения, можно отдавать предпочтение какой-либо задаче, делая выбор следующей задачи неравномерным

Проблемы совместного обучения

- Ключевые параметры нейронной сети (скорость обучения и расписание ее затухания; размер mini-batch; кол-во эпох) тщательно подбираются под задачу и обучающие данные:
  - в случае совместного обучения все параметры едины для всех задач
  - в случае попеременного – кол-во шагов обучения разных задач взаимосвязано через уровни предпочтения (чаще всего – равно)
- Как и любая регуляризация, СО способно не бороться с переобучением, а приводить к недообучению модели
- Как выбирать предпочтение между задачами и должно ли оно меняться в процессе обучения?

Следующая лекция