Основы обработки текстов

Лекция #10:
Извлечение отношений

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

План лекции

- Постановки задачи извлечения отношений
- Области применения извлечения отношений
- Методы на основе деревьев зависимостей
- Методы, работающие на «чистом» тексте
- Механизм внимания
- Извлечение сложных отношений
- Оценка качества извлечения отношений

Извлечение отношений (relation extraction)

- Отношение описывает смысловую взаимосвязь между сущностями, выраженную автором в тексте

Дмитрий стал руководителем разработки в Microsoft

Назначение	на должность
сотрудник	Дмитрий
должность	руководителем разработки
компания	Microsoft

Извлечение отношений (relation extraction)

- Отношение описывает смысловую взаимосвязь между сущностями, выраженную автором в тексте

Без двигателя не может быть автомобиля

Часть-целое	
часть	двигателя
целое	автомобиля

Сквозное извлечение отношений

- Извлекаем «одновременно» и сущности, и отношения

Дмитрий стал руководителем разработки в Microsoft

Назначение	на должность
сотрудник	Дмитрий
должность	руководителем разработки
компания	Microsoft

Классификация отношений

- Известно, что отношение есть, необходимо определить лишь его тип

Без двигателя не может быть автомобиля

Области применения извлечения отношений

- Извлечение сущностей и отношений позволяет построить по массиву текстов базу фактов (взаимосвязей сущностей), т.е. перейти от слабоструктурированной (текста) к полностью структурированной информации
- Такую базу можно применять для:
  - Вопросно-ответных систем: «Где родился Пушкин?»
  - Сопоставления фактов, полученных из нескольких источников: поиск ошибок в документах, выявление манипуляций читателями и т.п.
  - Обучения новичков в области: они могут обращаться к базе на формальном языке для быстрого поиска информации

Методы извлечения отношений

- Типовой подход – попарная классификация сущностей: отношение типа R_1, R_2, …, R_n или отсутствие отношения
- Проблема: «отрицательных» примеров много больше, чем «положительных» => сильный перекос обучающей выборки
- Возможные решения:
  - Дублирование «положительных» примеров «как есть»
  - Аугментация «положительных» примеров: sampling аргументов

    оригинал: Дмитрий работает в Microsoft. Василий работает в Google.

    результат: Василий работает в Microsoft. Дмитрий работает в Google.
  - Фильтрация малоинформативных «отрицательных» примеров

Типовые признаки классификатора пар

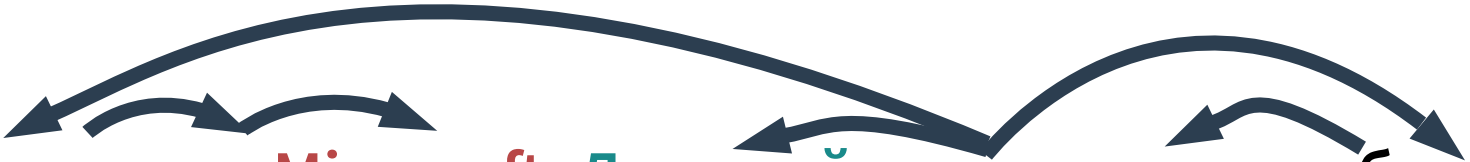
Весной Дмитрий устроился в отделение Microsoft в Берлине

- Контексты:
  - Левый: «Весной»
  - Средний: «устроился», «в», «отделение»
  - Правый: «в», «Берлине»
- Признаки контекста:
  - Ngramы слов
  - Ngramы частей речи
  - Вхождение сущностей различных типов: «Берлине»
- Типы сущностей — аргументов

Методы на основе синтаксической структуры

Дмитрий недавно устроился в отделение Microsoft в Берлине

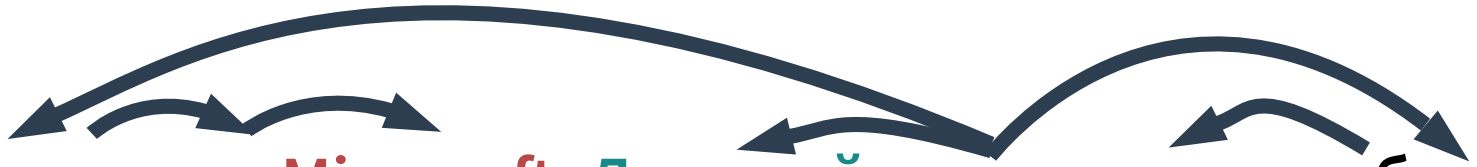
- Слова на кратчайшем пути между сущностями вносят наибольший вклад в решение классификатора
- Удаление других слов может снизить уровень шума



Устроившись в Microsoft, Дмитрий решил свои проблемы

- Из-за свободного порядка слов слова на кратчайшем пути не всегда находятся между сущностями

Признаки на основе синтаксической структуры



Устроившись в Microsoft, Дмитрий решил свои проблемы

- Кратчайший путь: «решил», «Устроившись», «в»
- Ngramы:
  - слов кратчайшего пути
  - частей речи
  - меток дуг в дереве зависимостей
- Сверточная сеть по кратчайшему пути
- Рекуррентная сеть

Методы, работающие на «чистом» тексте

Устроившись в Microsoft, Дмитрий решил свои проблемы

- В обработке текстов наблюдается тенденция «end-to-end» моделей, т.е. моделей, избавленных от предобработки:
  - избегается зашумление моделей: нет накопления ошибок
  - повышается их производительность: T(parser) >> T(rel-ext)
- Кусочно-сверточная сеть (piece-wise CNN):
  - независимая свертка и max pooling по 3 контекстам
  - классификатор на векторных представлениях 3 контекстов

Кусочно-сверточная сеть

- В свертку по контексту подаются:
  - word embeddings (word2vec, GloVe и т.п.)
  - word position embeddings, обучаемые совместно с основной сетью

  Устроившись в Microsoft, Дмитрий решил свои проблемы

  -4 -3 -2 -1 0 1 2 3

- Позиция вычисляется как смещение относительно каждой из сущностей-аргументов
- Такая информация крайне важна, чтобы избежать путаницы для сложных предложений с большим числом сущностей

Механизм внимания

Дмитрий родился в Москве, увлекался компьютерами и информатикой с детства, в 18 лет закончил ВМК МГУ по специальности «Прикладная математика и информатика», в 20 лет устроился в Microsoft руководителем разработки

- Кратчайший путь — 2 токена, между сущностями — 31
- Когда дерево недоступно, необходимо каким-то другим образом устранять малоинформативные части контекста

Механизм внимания. Машинный перевод

- Основная идея: присвоить каждому токену вес, затем брать взвешенную сумму векторов всех токенов в качестве вектора последовательности
- В машинном переводе Bahdanau был предложен механизм внимания, который можно модифицировать для RelExt:

  h_i — результат свертки по контексту

  $\alpha_i = \text{softmax}[i](v_\alpha \tanh(U_\alpha h_i))$

Механизм внимания. Сущности-аргументы

- Специально для RelExt Wang был предложен (аналог Luong) механизм внимания относительно сущностей-аргументов:

  x_i — признаки i-ого слова (embeddings слова и смещений)

  e_1, e_2 — embeddings аргументов:

  - Если аргументы — слова, e_j — их признаки x_k
  - Если аргументы — словосочетания, e_j — усредненные признаки
  - Случайные, обучаются вместе с остальной сетью

  $\alpha_i^j = \text{softmax}[i](e_j A_j x_i)$; $r_i = (\alpha_i^1 + \alpha_i^2) / 2$

- r_i подается в свертку вместо x_i

Извлечение сложных отношений

- Во многих предметных областях отношения возникают на уровне нескольких предложений, а не одного
- Чаще всего это случается для N-арных отношений (N > 2)

  Дмитрий недавно устроился на работу в Microsoft

  Занятая им должность – руководитель разработки
- Можно разрешать анафору, однако это крайне трудная задача, и данный подход не является панацеей

  Занимаемая должность – руководитель разработки
- Требуются более гибкие методы

Иерархические методы

- Сверточные сети с max pooling или механизмом внимания, работающие по «чистому» тексту, не подходят из-за своей чувствительности к длине последовательности
- Решение: переход от последовательности слов к последовательности последовательностей слов
- Сначала применяем свертку и внимание к словам предложения и получаем вектора всех предложений
- Затем к этим векторам применяем аналогичное преобразование уже на уровне последовательности предложений

Обобщение кратчайшего пути

- Кратчайший путь между аргументами построить нельзя, т.к. дерево зависимостей описывает лишь 1 предложение
- Решение: связать корни дугой «следующее предложение»

Представление текста в виде графа

- Стандартный LSTM работает на последовательности
- BiLSTM – на последовательности «вперед» и «назад»
- Это можно рассматривать как граф, где в каждую вершину-токен входят дуги от вершин-соседей

Представление текста в виде графа

- Кроме того, имеются деревья зависимостей и кореферентные связи
- Из всего этого можно построить один ациклический граф

GraphLSTM

- Обобщение LSTM на случай ациклического графа
- Для простоты граф разбивается на две независимые части по направлению дуг: влево и вправо

LSTM «вперед»

$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$ что забываем?

$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$ что запоминаем?

$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$ что выдаем?

$c_t = f_t * c_{t-1} + i_t * \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c)$ ячейка

$h_t = o_t * \tanh(c_t)$ выход

GraphLSTM «вперед»

$f_{tj} = \sigma(W_{fx}x_t + W_{fhm}h_j + b_f)$ что забываем?

$i_t = \sigma(W_{ix}x_t + \Sigma_j W_{ihm}h_j + b_i)$ что запоминаем?

$o_t = \sigma(W_{ox}x_t + \Sigma_j W_{ohm}h_j + b_o)$ что выдаем?

$c_t = \Sigma_j f_{tj} * c_j + i_t * \tanh(W_{cx}x_t + \Sigma_j W_{chm}h_j + b_c)$ ячейка

$h_t = o_t * \tanh(c_t)$ выход

- j — предшественники t, m – тип связи (t, j): сосед, родитель

Оценка качества извлечения отношений

- Отношение представляется как кортеж сущностей и тип
- Все ли извлеченные отношения извлечены верно?

  Точность / precision
- Все ли имеющиеся отношения извлечены?

  Полнота / recall
- Сбалансированная мера

  $F_\beta = (1 + \beta^2) \cdot precision \cdot recall / (\beta^2 \cdot precision + recall)$

  $F_1 = 2 \cdot precision \cdot recall / (precision + recall)$

Следующая лекция

- Привязка к базам знаний
- Лектор: Сысоев Андрей Анатольевич