«Анализ неструктурированных данных»

Семинар 7

Выявление именованных сущностей

Мурат Апишев (great-mel@yandex.ru)

17-18 октября, 2018

Содержание занятия

- Постановка задачи
- Метрики качества
- Rule-based подход
- CRF
- BiLSTM + CRF
- Сети-трансформеры + CRF
- Подготовка данных

Name Entity Recognition

Постановка задачи «sequence labeling»:

- Дан корпус текстов D
- Каждый текст представляет собой последовательность токенов
- Каждому токену присвоена метка из некоторого множества V

В зависимости от множества меток V получаем разные типы подзадач:

- для $V ==$ множество частей речи $\Rightarrow$ POS-теггинг
- для $V ==$ множество типов имен. сущностей $\Rightarrow$ NER

Именованная сущность – фрагмент текста, обозначающий некоторый интересный объект.

Примеры

Trump's ORG victory traces back to June DATE , when Mr. Strzok PERSON 's conduct was laid out in a wide-ranging inspector general's report on how the F.B.I. GPE handled the investigation of Hillary Clinton's PERSON emails in the run-up to the 2016 DATE election., The report was critical of Mr. Strzok PERSON 's conduct in sending the texts, and the bureau's, Office of Professional Responsibility ORG said that Mr. Strzok PERSON should be suspended for 60 days DATE and demoted., Mr. Strzok PERSON had testified before the House ORG in July DATE about how he had not allowed his political views to interfere with the investigations he was overseeing., But Mr. Strzok

- Имена людей
- Названия географических объектов
- Названия организаций
- Специальная терминология
- Названия торговых брендов
- Даты
- Любое интересное в решаемой задаче слово/словосочетание

Метрики качества

- Существует множество подходов к оцениванию качества решения задачи NER
- Можно проверять тип сущности, её начало, длину в словах
- Самый простой и общий вариант: проверять метку сущности отдельно для каждого слова в каждом предложении
- На основании этого для каждой метки по-отдельности и для всех в совокупности можно посчитать точность/полноту/f-меру
- Слова, не являющиеся сущностями, в этом случае должны иметь соответствующую метку

Rule-based approach

- Пусть есть выборка текстов, но для неё отсутствует разметка
- Известен список именованных сущностей, есть достаточно примеров
- Методом «пристального» взгляда и некоторого анализа можно подобрать эвристические признаки:
  - правила на основе морфологии/синтаксиса
  - регулярные выражения
- И с их помощью выделять из текста те или иные типы сущностей
- Пример инструмента для русского языка – Natasha

Примеры правил

1. если подряд идут двузначное число и название месяца, то это дата
2. если одна заглавная буква заканчивается точкой, и за ней идёт слово с заглавной буквы, то это имя
3. если перед словом с заглавной буквы посреди предложения стоит предлог, то это название географического объекта или организации

Правила обычно неточны и неполны, но если

- их много
- каждое из них полезно

то результат может получиться очень хорошим

Conditional Random Fields

- Марковское случайное поле: неориентированный граф $G = (V, E)$ и множество функций $\{\phi_k\}$
- V – множество случайных переменных-вершин, E – множество рёбер, отражающих попарные зависимости между переменными
- Содержимое $\{\phi_k\}$ – *потенциальные функции*, по одной на каждую клику G (полный подграф), область значений $\phi_k \subseteq \mathbb{R}_+$
- Вершины, не являющиеся смежными, должны соответствовать условно независимым случайным величинам
- Группа смежных вершин образует клику, набор состояний вершин является аргументом соответствующей потенциальной функции

Ссылка на источник

Conditional Random Fields

- $V = X \cup Y$, X – наблюдаемые переменные, Y – предсказываемые
- CRF – дискриминативная модель: в отличие от HMM она строит не совместное распределении $p(y, x)$, а условное $p(y \mid x)$, которое обычно и нужно в задачах ML
- В *линейном условном случайном поле* потенциальная функция имеет вид

$$\phi_k(x_k) = \exp(\sum_s \lambda_s f_s(y_t, y_{t-1}, x_t)),$$

где s – индекс признаковой функции f_s, $\lambda_s \in \mathbb{R}$

- Строим распределение

$$p(y \mid x_t) = \frac{1}{Z(x)} \prod_{k \in K} \exp(\sum_s \lambda_s f_s(y_t, y_{t-1}, x_t)),$$

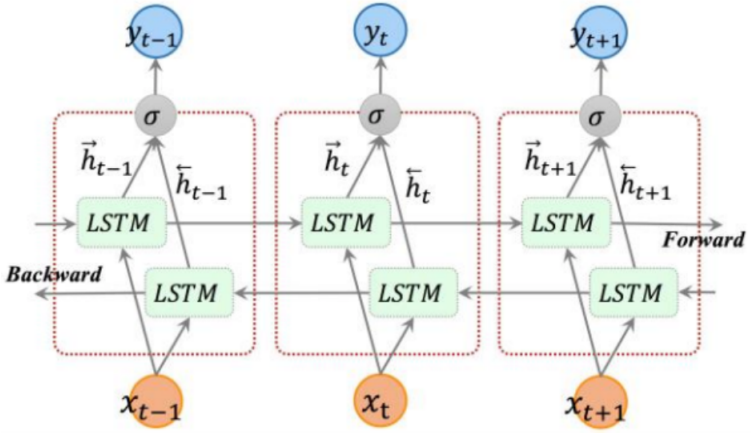
где t – индекс очередного элемента последовательности, K – множество клик G. $Z(x)$ получается суммированием числителя по всем y

Conditional Random Fields

- CRF лишена Label Bias Problem – ситуации в MEMM, когда наибольшее предпочтение получают состояния с меньшим числом переходов в другие (подробнее тут)
- Качество сильно зависит от выбора признаков f_s
- Один из лучших методов для NER и POS-теггинга
- Очень долго обучается
- Хорошо работает в связке с рекуррентными нейросетями, моделирует совместное распределение на всей последовательности выходов сети одновременно

Архитектура BiLSTM

Обычная LSTM учитывает только прошлый контекст, двунаправленная учитывает и будущий:



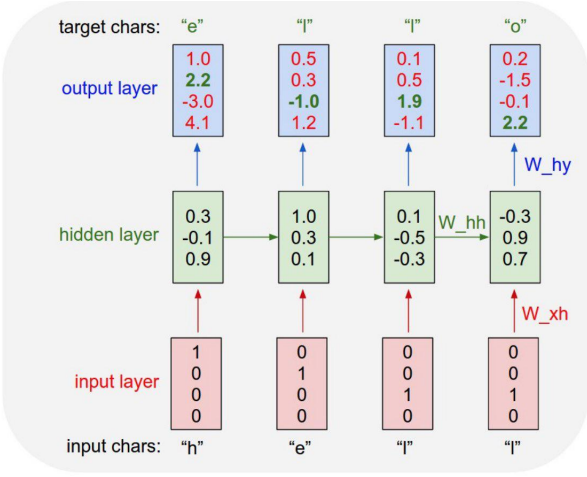
Ссылка на источник картинки

BiLSTM + CRF для задачи NER

Основные шаги:

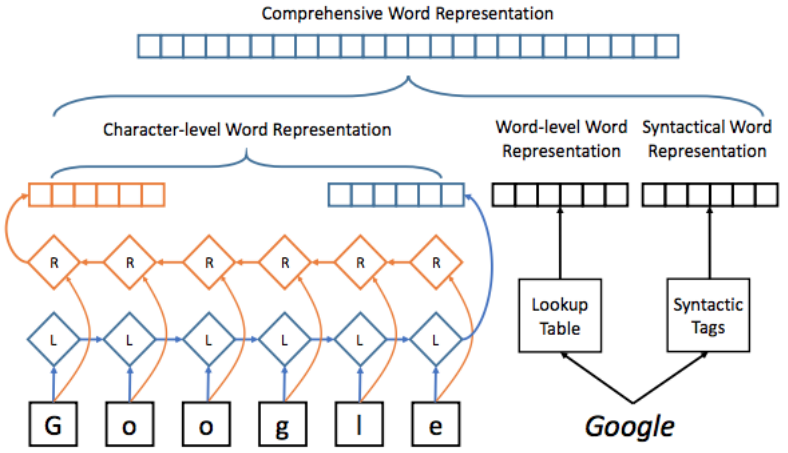
- Получить предобученные эмбеддинги слов коллекции (word2vec, GloVe)
- Обучить символьные эмбеддинги (char-BiLSTM)
- Составить для каждого слова морфологические/синтаксические признаки (POS-тег, роль в предложении и т.п.)
- Объединить всё это и подать на вход основной сети (BiLSTM)
- Выходы h_t для всех слов предложения подавать на вход классификатору, который будет предсказывать NER-тег (CRF)
- Готовые реализации: torchnlp и код из хорошего туториала

Символьные эмбеддинги



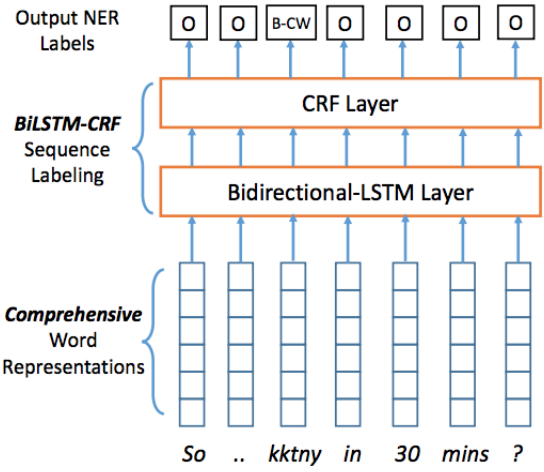
Ссылка на источник картинки

Комплексное представление слова



Ссылка на источник картинки

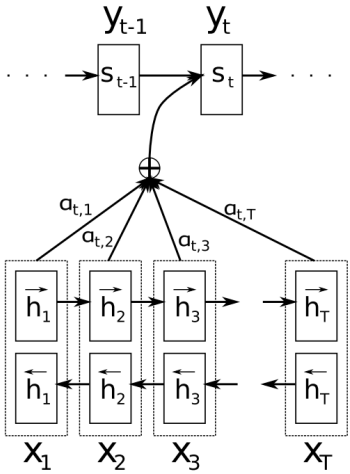
Общая схема модели



Ссылка на источник картинки

Attention в глубинном обучении

- Кодирование производится с помощью BiLSTM (не принципиально)
- Веса a_i обычно суммируются в 1
- Визуализируя веса можно понимать стратегию получения результат (от каких частей контекста зависело появление данного типа именованной сущности)



Ссылка на источник картинки

Attention в глубинном обучении

- При использовании BiLSTM каждый вектор h_j хранит информацию о всей последовательности, но в наибольшей степени о j-м слове и его соседях

- Далее при текущем выходе y_{t-1} (с вектором s_{t-1}) для вектора каждого входного слова h_j считается a_{tj} – вклад в генерацию следующего выходного вектора (attention):

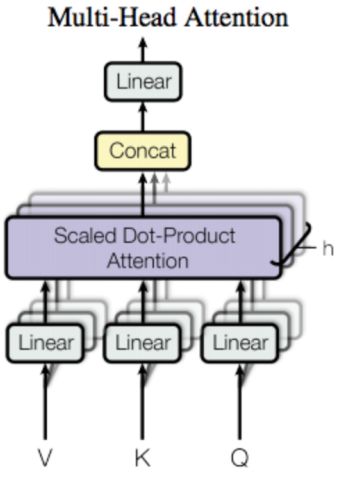
$$a_{tj} = \frac{\exp(e_{tj})}{\sum_k \exp(e_{tk})},$$

где $e_{tj} = f(s_{t-1}, h_j)$ – модель выравнивания

- Модель выравнивания предсказывает то, насколько хорошо соотносятся входное слово в позиции j и выходное в позиции t (обычно это простая модель, например, однослойная сеть)

Multi-head attention

- Вход: *вектор запроса* и несколько пар *векторов ключей и значений* (ключ и значение обычно совпадают)
- Для запроса и каждого ключа считается вес (линейный слой)
- Значения суммируются с этими весами в итоговый вектор
- **Идея**: обучать несколько attention, в надежде, что они станут отвечать за разные признаки слов
- Результаты пропустим через однослойную сеть, получим на выходе один вектор той же размерности, что и входные

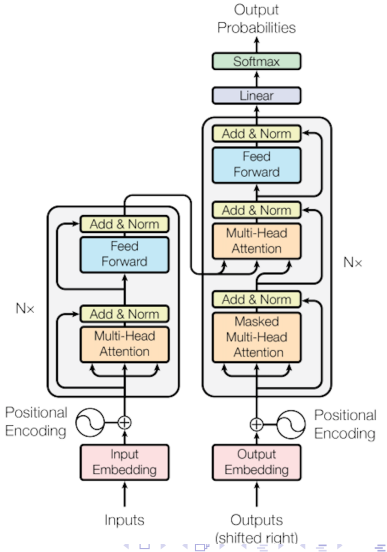


Transformer

- Состоит из encoder и decoder, в каждом используются multi-head attention и полносвязные (свёрточные) слои (нет RNN)
- Для каждого слова в encoder формируется вектор на основе нескольких multi-head attention-слоёв, который передаётся в декодер
- На основании векторов энкодера, а также выходных векторов для уже обработанных слов получается вектор для текущего слова. В обоих случаях используется multi-head attention

---

Ссылка на оригинальную статью



Transformer

- positional encoding – дополнительный вектор признаков для каждого слова, представляющий собой набор значений синусов и косинусов с разными периодами от позиции слова в предложении
- В трансформере и в encoder, и в decoder можно использовать несколько последовательных multi-head attention-слоёв (у сети Google их 6)
- Для большей выразительности и качества добавляют полносвязные слои, а также дропаут, нормализацию по слою и residual connections
- Обучение модели внутри предложения можно распараллелить, в отличие от RNN
- Ещё ссылки на хорошие статьи: medium, хабр

Что делать, если нужно решать NER

Последовательность действие следующая:

1. Собрать корпус текстов по интересующей тематике
2. Нанять экспертов и выдать им инструкции по разметке слов на нужные типы именованных сущностей
3. Определиться с форматом данных и привести к нему полученную разметку
4. Составить правила/выбрать модель
5. Реализовать классификатор самостоятельно или адаптировать существующий код

Форматы разметки данных

- Используйте распространённый формат – проще будет запускать готовые инструменты и презентовать данные и результат
- Существует несколько общепринятых форматов разметки данных для задачи NER
- Наиболее известные – IO и BIO:
  - B – первое слово сущности
  - I – слово-часть сущности
  - O – слово-не часть сущности

Tokens	IO	BIO	BMEWO	BMEWO+
Yesterday	O	O	O	BOS_O
afternoon	0	O	O	O
,	0	O	O	O_PER
John	I_PER	B_PER	B_PER	B_PER
J	I_PER	I_PER	M_PER	M_PER
.	I_PER	I_PER	M_PER	M_PER
Smith	I_PER	I_PER	E_PER	E_PER
traveled	O	0	O	PER_O
to	O	O	O	O_LOC
Washington	I_LOC	B_LOC	W_LOC	W_LOC
.	O	O	O	O_EOS

Коэффициенты согласованности экспертов

- Необходимы как для собственной валидации, так и для презентации результатов разметки
- Существует много способов подсчёта, основные:
  1. Простые статистики: среднее линейное или квадратичное отклонение, дисперсия
  2. Коэффициента ранговой корреляции Спирмэна
  3. Коэффициент конкордации Кенделла
  4. Коэффициент альфа Кронбаха
- Подробности по ссылкам: Хабр, Википедия