Основы обработки текстов

Лекция #1:
Распознавание именованных сущностей,
машинное обучение с учителем

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

Действующие лица

- Преподаватель:
  - к.ф.-м.н., доцент кафедры СП Турдаков Денис Юрьевич
- 3 лектора:
  - м.н.с. ИСП РАН Андрианов Иван Алексеевич
  - м.н.с. ИСП РАН Майоров Владимир Дмитриевич
  - н.с. ИСП РАН Сысоев Андрей Анатольевич
- Команда Texterra (https://texterra.ispras.ru/) - часть отдела информационных систем ИСП РАН

Структура курса

- http://tpc.at.ispras.ru/
- 12 лекций:
  - 9 лекций «основы»: полностью выносятся на экзамен
  - 3 «продвинутых» лекции: на экзамен выносятся постановки задач
- 2 практических задания:
  - сроки сдачи — 31.10 и 20.12
  - «удовл» — за main class, +1 балл — за baseline
  - +1 балл на экзамен — за «оригинальное» решение

План лекции

- Задача распознавания именованных сущностей и ее практические применения
- Важные для решения задачи сигналы
- Основы машинного обучения с учителем
- Линейные классификаторы и формирование признакового пространства
- Оценка качества методов распознавания именованных сущностей

Распознавание именованных сущностей (NER)

- На входе: текст, разбитый на предложения и токены
- На выходе: множество сущностей (начало, конец, тип)

Александр Пушкин родился в Москве, столице России

личность город страна

Microsoft — один из крупнейших производителей ПО в мире

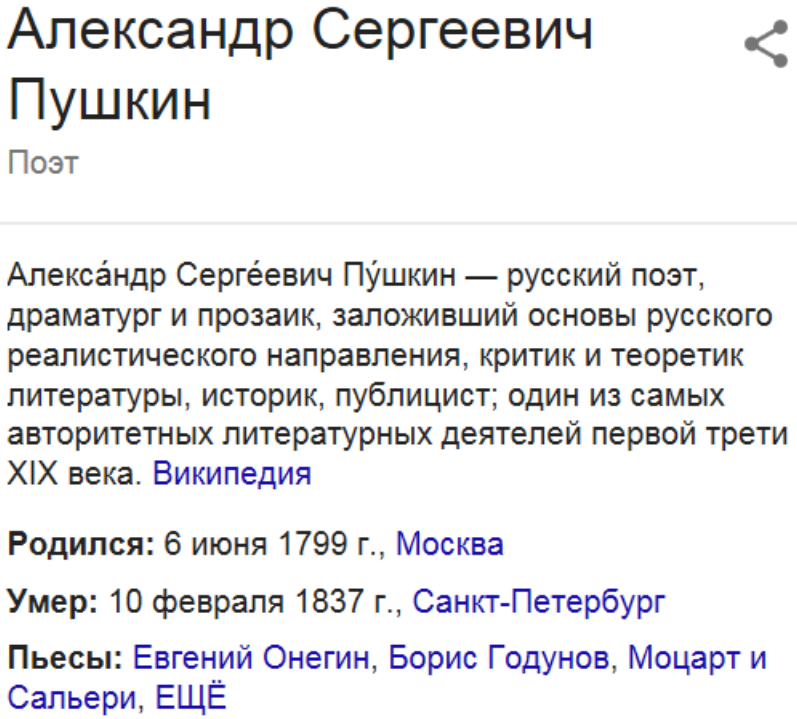
компания

Обработка текстов — область на стыке ИИ и лингвистики

научная дисциплина НД НД

Практические применения NER

- **Информационный поиск и вопросно-ответные системы**
  - **сущности как правило важнее «обычных» слов**
  - **сущности могут иметь описание на Википедии и подобных ресурсах**



Практические применения NER

- Извлечение отношений
  - сущности выступают в роли аргументов отношения
  - родился(личность, страна)

    Василий Пупкин появился на свет в тихом уголке России
  - штаб-квартира(компания, город)

    Крупнейший и основной офис Microsoft расположился неподалеку от Сиэттла, в Редмонде
- Реферирование
  - текст скорее всего посвящен именно сущностям

Решение NER через словари

Александр Пушкин родился в Москве

Министерство путей сообщения было упразднено в 2004 году

- Можно составить словари для актуальных типов сущностей и сопоставлять словосочетания в тексте с ними
- Разреженность сущностей:
  - Словари всегда будут неполными
  - Во многих языках слова склоняются => лемматизация
  - Имена и фамилии могут встречаться в произвольном порядке
  - Имеются явные шаблоны «министерство …» => правила

Решение NER через правила

Юрий Левитан родился во Владимире

Всеволод Юрьевич «Большое Гнездо» умер во Владимире

Игорь Тальков убит в Санкт-Петербурге

Инцидент произошел в Туле

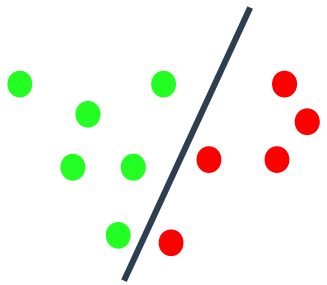
- Многозначность сущностей:
  - «Владимир» - это «личность» или «город»? => анализ контекста
- Разреженность контекста:
  - «(родился|умер|убит|...) в(|о) <город>» => (квази-)синонимия
  - «<глагол> в(|о) <город>» => части речи

Машинное обучение с учителем

- Составление правил — крайне трудоемкий процесс: в случае задачи NER могут потребоваться тысячи правил
- Разработанные правила скорее всего будут специфичны для какого-то набора типов сущностей, их будет сложно адаптировать к другому набору типов
- Вместо правил мы можем составить выборку текстов, в которой будут вручную размечены примеры сущностей
- Нам понадобится алгоритм, который, «просмотрев» выборку примеров, будет выделять сущности на произвольном тексте

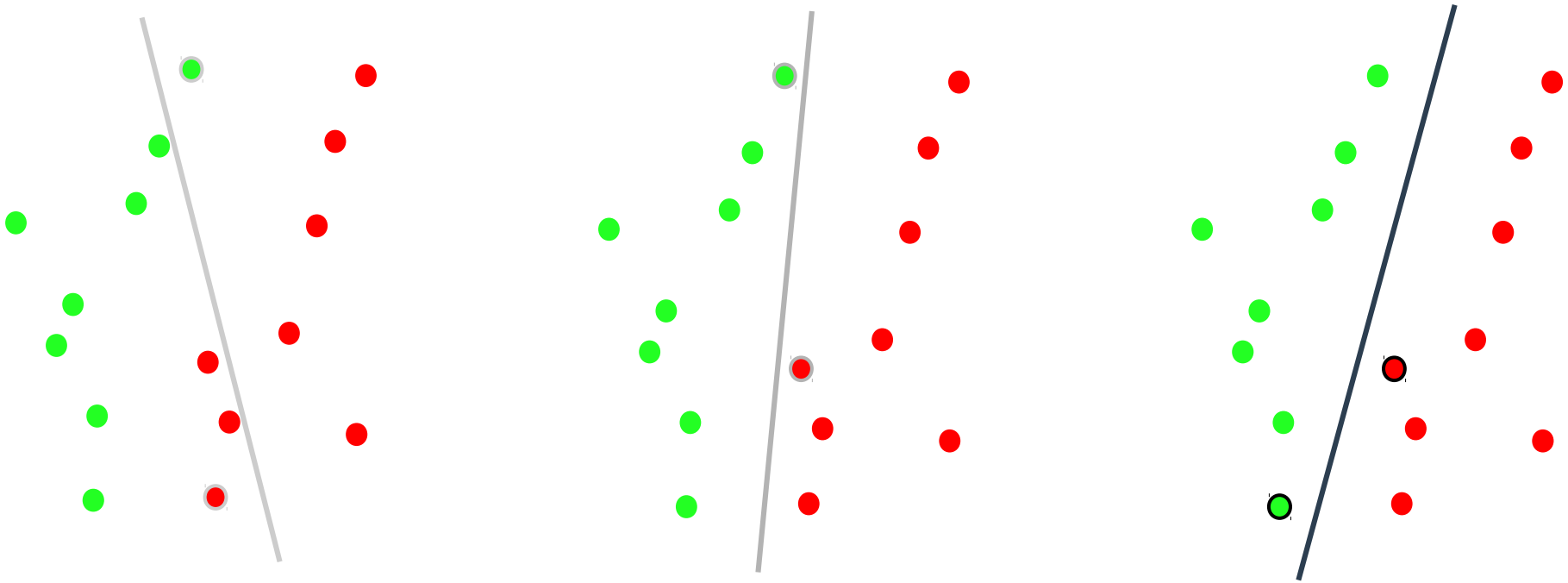
Линейные классификаторы

- Классификация на 2 класса «+» и «-»: слово в контексте — имя человека? текст — спам? будет ли сегодня дождь?
- Каждый классифицируемый объект (слово в контексте, текст, изображение неба) представляется точкой в «признаковом» N-мерном пространстве
- По обучающей выборке строится гиперплоскость, разделяющая точки из противоположных классов



Метод опорных векторов (SVM)

- Максимизируем расстояние до гиперплоскости
- $w_1x_1 + w_2x_2 + \ldots + w_nx_n + b = (w, x) + b = 0$



- liblinear (C, Java), scikit-learn (Python), weka (Java)

SVM с ядровыми функциями

- SVM использует скалярное произведение: (w, x) + b = 0
- Можно применять ядровые функции: K(w,x) + b = 0 и строить поверхность вместо гиперплоскости

- libsvm (C), scikit-learn (Python), weka (Java)

Логистическая регрессия

- Вероятности принадлежности к классам вместо расстояний
- Логистическая функция $\sigma(t) = 1 / (1 + e^{-t})$ лежит в (0,1): ее можно считать вероятностью принадлежности к классу «+»
- Логистическая регрессия: минимизируем вероятность неверного класса (кросс-энтропию)

  $l = -\hat{y} \cdot \log y - (1 - \hat{y}) \cdot \log(1 - y)$, где $y = \sigma((w, x) + b)$
- Допускает дообучение

- liblinear (C, Java), scikit-learn (Python), weka (Java)

NER как линейный классификатор

Юрий Левитан родился во Владимире

- Классифицируем каждое слово на «О», «личность», «город»
- Сигналы для слова и его контекста:
  - Словоформа в окне размера w=2

    (родился, во, Владимире, </s>, </s>)
  - Лемма в окне

    (рождаться, в, владимир, </s>, </s>)
  - Часть речи в окне

    (глагол, предлог, существ, </s>, </s>)

Переход от сигналов к признакам: части речи

Юрий Левитан родился во Владимире

- Присвоим каждой части речи уникальный номер

  глагол => 1, предлог => 2, существ => 3, </s> => 4, <s> => 5

- Выделим каждому слову в окне одно измерение в пространстве признаков, значением выберем номер ЧР

  (глагол, предлог, существ, </s>, </s>) => (1, 2, 3, 4, 4)

- Проблема: значения признаков в линейных классификаторах выражают близость примеров друг к другу

Переход от сигналов к признакам: части речи

Юрий Левитан родился во Владимире

- Присвоим каждой части речи уникальный номер

  глагол => 1, предлог => 2, существ => 3, </s> => 4, <s> => 5

- Выделим каждому слову в окне и каждой части речи одно измерение в пространстве признаков, применим one-hot encoding

  (глагол, предлог, существ, </s>, </s>) =>

  (1, 0, 0, 0, 0 | 0, 1, 0, 0, 0 | 0, 0, 1, 0, 0 | 0, 0, 0, 1, 0 | 0, 0, 0, 1, 0)

Переход от сигналов к признакам: словоформы

- В случае словоформ / лемм также применяем one-hot
- Словарь словоформ / лемм неограничен, поэтому добавляем к нему <unk> («неизвестное слово»)
- Редкие (частота в обучающей выборке < 2-5) слова можно заменять на <unk>: слишком малая их частота не позволит модели научиться адекватно реагировать на такой сигнал
- One-hot по словарям > 1k крайне неэффективен из-за чудовищной разреженности => векторные представления слов, учитывающие синонимию

Общая схема решения NER

Александр Пушкин погиб в результате дуэли с Дантесом

- Рассматриваются только метки «О» и «личность»

(<s>, <s>, Александр, Пушкин, погиб)	окно токенов (n=2)
(1, 1, 4, 2, 3)	окно индексов
(1,0,0,0|1,0,0,0|0,0,0,1|0,1,0,0|0,0,1,0)	one-hot признаки (x)
p(y=1|x;w;b) = σ((w, x) + b)	вероятность «лич»

Оценка качества методов NER

Юрий Левитан родился во Владимире

- Все ли распознанные сущности распознаны верно?

  Точность / precision

- Все ли имеющиеся сущности распознаны?

  Полнота / recall

- Сбалансированная мера

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

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

Оценка качества методов NER

Юрий Левитан родился во Владимире

- P = 1, R = 0.5, F_1 = 0.6(6)

Юрий Левитан родился во Владимире

- P = 0.5, R = 0.5, F_1 = 0.5

Юрий Левитан родился во Владимире

- P = 0, R = 0, F_1 = 0/0

Юрий Левитан родился во Владимире

- P = 1, R = 1, F_1 = 1

Следующая лекция

- Разметка последовательности, нейронные сети
- Лектор: Андрианов Иван Алексеевич