«Анализ неструктурированных данных»

Семинар 6

Анализ тональности текстов

Мурат Апишев (great-mel@yandex.ru)

10-11 октября, 2018

Задачи сантимент-анализа

Три типа задач (по возрастанию сложности):

1. Полярная тональность (positive / negative / neutral)
2. Ранжированная тональность («звёздочки» от 1 до N)
3. Выявление источника / цели или более сложные типы

http://www.greenbookblog.org/2012/01/02/from-sentiment-analysis-to-enterprise-applications/

Ключевые моменты

- Токенизацию и лемматизацию нужно производить аккуратно, как и фильтрацию словаря
- С опечатками можно бороться с помощью буквенных N-грамм
- `не_` лучше приклеивать к впередистоящему слову, инвертируя его тональность
- Очень важно выделять смайлы и экспрессивную пунктуацию (регулярные выражения)
- Обучение и тестирование нужно производить на схожих данных
- Для учёта редких слов можно логарифмировать частоты

Сложности

Помимо чисто технических проблем, возникают также более сложные семантические:

- Отзывы могут иметь ясный смысл, но при этом не содержать позитивных или негативных слов:

  Это фильм заставляет прочувствовать всю гамму эмоций от «А» до «Я».

- Отзыв может содержать позитивные слова, но на самом деле выражает ожидание:

  Это фильм должен был быть супер крутым. Но не был.

Подходы к решению

- Правила (точно, трудозатратно)

  Я люблю кофе – если сказуемое в группе положительных глаголов и нет отрицаний то positive

- Словари (просто, зависит от предметной области)

слово i из текста	соответствующая тональность $a_i \in [1\text{-}9]$ из словаря
хороший	7.47
счастливый	8.21
...	...
скучный	2.95

$$a_{text} = \frac{\sum_{i=1}^{n} a_i}{n}$$

Подходы к решению

- ML: обучение с учителем (классификация) (точно при достаточной обучающей выборке, требуется данные для обучения)
  - Получаем размеченную коллекцию текстов
  - Выделяем признаки (специфичные для тональности)
  - Обучаем классификатор
- ML: обучение без учителя (просто, не требуются данные для обучения, нужен словарь, низкая точность)
  - Выделить ключевые слова (например, по TF-IDF)
  - Определить тональность ключевых слов
  - Сделать вывод о тональности предложения/текста

Сантимент-лексикон

- Существуют наборы слов с оценённой степенью позитивности/негативности, активности.пассивности и т.п.

  Примеры[1]: MPQA subjectivity cues lexicon, SentiWordNet

- Сантимент-лексикон можно размечать самостоятельно, это задача частичного обучения. Необходимо разметить часть примеров и описать правила пополнения.

  Пример: `bad and ugly; cruel but amazing`

  Если два слова связаны and, и тональность одного нам известна, то второе тоже будет иметь такую тональность.

  Если через but — то противоположную.

[1] Ссылка на ресурсы

О классификации

Naive Bayes:

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, p(c) \prod_{w \in d} \hat{p}(w|c)$$

$$\hat{p}(w|c) = \frac{count(w, c) + 1}{count(c) + |W|}$$

- $count(w, c)$ — число раз, когда слово w встретилось в документе с классом c;
- $count(c)$ — общее число слов в документах с классом c.

Пример классификации

```
1  from nltk.classify.util import accuracy
2  from nltk.classify import NaiveBayesClassifier
3  from nltk.corpus import movie_reviews
4
5  def get_feats(tokens):
6      return dict([(token, True) for token in tokens])
7
8  def create_sample(tag, train_ratio):
9      ids = movie_reviews.fileids(tag)
10     feats = [(get_feats(
11                 movie_reviews.words(fileids=[f])),
12                 tag) for f in ids]
13
14     idx = int(len(feats) * train_ratio)
15     train, test = feats[: idx], feats[idx: ]
16     return train, test
```

Пример классификации

```
1  train_pos, test_pos = create_sample('pos', 0.75)
2  train_neg, test_neg = create_sample('neg', 0.75)
3  train = train_pos + train_neg
4  test = test_pos + test_neg
5
6  print('Train on {} docs, test on P{} docs'.format(
7            len(train), len(test)))
8
9  classifier = NaiveBayesClassifier.train(train)
10
11 print('Accuracy: {}'.format(accuracy(classifier, test)))
12
13 classifier.show_most_informative_features()
```

Результаты

```
Train on 1500 documents, test on 4000 documents
Accuracy: 0.728

Most Informative Features
     magnificent = True      pos : neg = 15.0 : 1.0
     outstanding = True      pos : neg = 13.6 : 1.0
     insulting = True        neg : pos = 13.0 : 1.0
     vulnerable = True       pos : neg = 12.3 : 1.0
     ludicrous = True        neg : pos = 11.8 : 1.0
     avoids = True           pos : neg = 11.7 : 1.0
     uninvolving = True      neg : pos = 11.7 : 1.0
     astounding = True       pos : neg = 10.3 : 1.0
     fascination = True      pos : neg = 10.3 : 1.0
     idiotic = True          neg : pos = 9.8 : 1.0
```

Модели и признаки для классификации

Используется всё, что для других задач классификации:

Модели:

- Лог-регрессия
- Рекурентные сети
- Свёрточные сети
- FastText

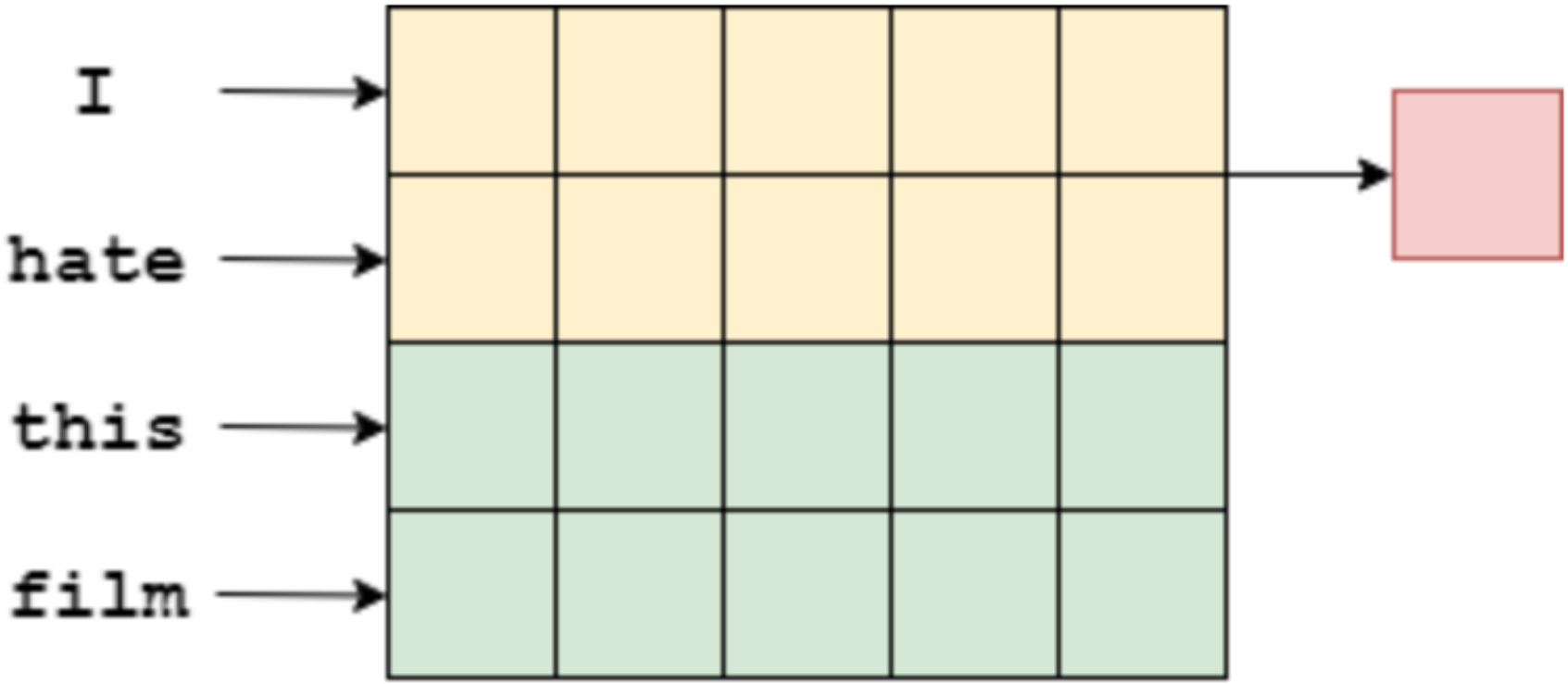
Признаки (в зависимости от выбранной модели):

- One-hot encoding
- TF-IDF
- Словарные эмбеддинги
- Всё это на N-граммах
- N-символьные эмбеддинги

Классификация свёрточными сетями

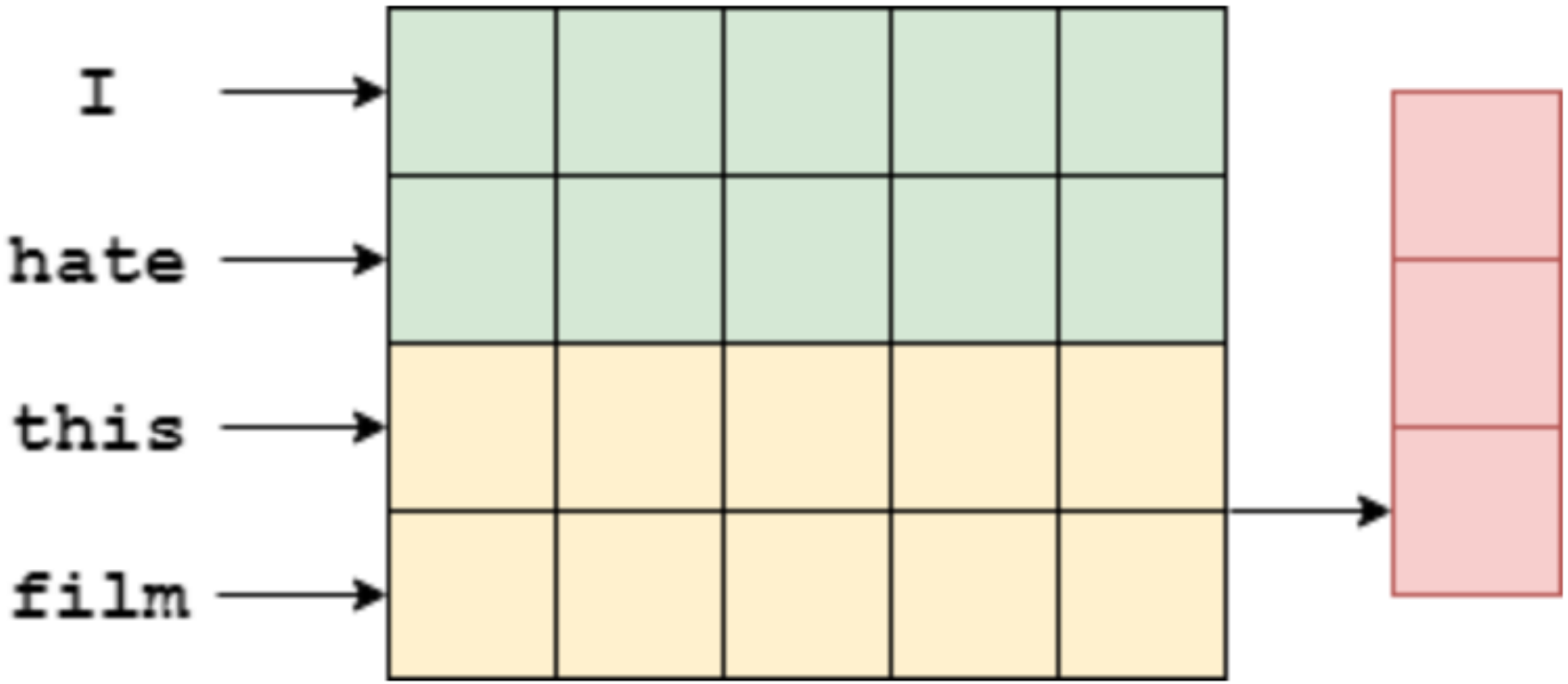
- Для применения свёрточных сетей к текстам необходимо привести текст к матричному виду.
- Для этого можно воспользоваться эмбеддингами
- Все объекты в одном батче должны быть одного размера, поэтому нужно либо нарезать тексты фрагментами фиксированной длины, длины использовать паддинг
- Для работы с текстами используются одномерные свёртки (вторая размерность всегда равна размерности эмбеддингов)

Классификация свёрточными сетями



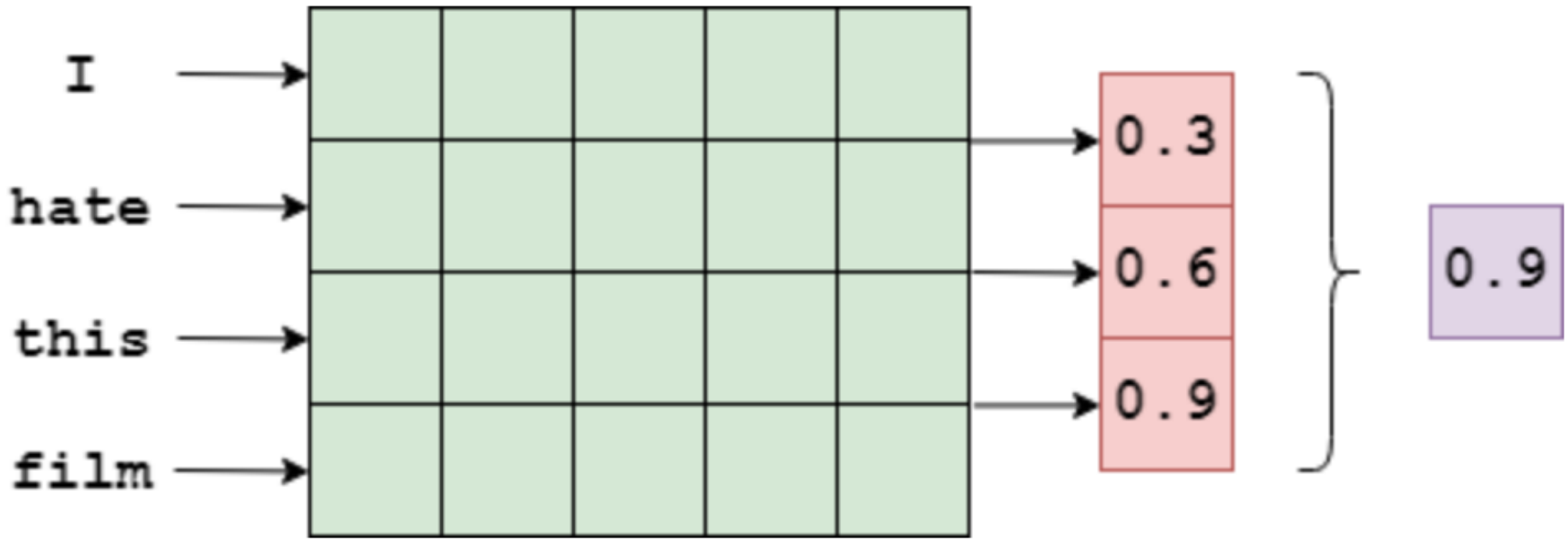
Ссылка на источник картинок

Классификация свёрточными сетями



Классификация свёрточными сетями

Следующий шаг, как и в свёрточных сетях для изображений – max-пуллинг:



Классификация свёрточными сетями

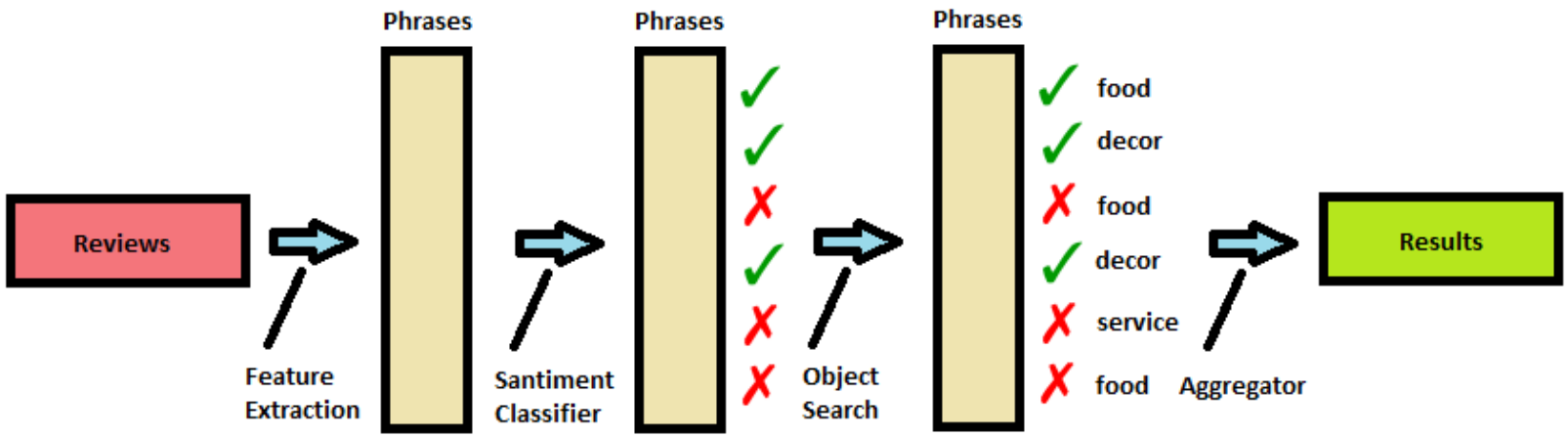
- Основная размерность свёрток может быть разной, что позволяет выделять признаки на N-граммах различной длины
- Идея в том, что наибольшее значение наиболее важного признака соответствует наиболее важной N-грамме текста.
- Получить его можно с помощью max-пуллинга, за счёт изменения backpropagation-ом весов свёрток для максимизации значения признака, наиболее влияющего на предсказание метки класса.
- Число фильтров будет определять размерность выходного эмбеддинга текста, который далее передаётся в линейный слой с софтмаксом на выходе для предсказания тональности.

Более сложные задачи

Поиск атрибутов и их объектов — в одном ревью могут по-разному оцениваться разные вещи. (Здесь могут помочь синтаксические зависимости!)

Для этого ищем частые фразы, которые нам интересны, и смотрим на то, сколько раз они встречаются сразу после сантиментных слов (пример: `great fish tacos`).

Такую работу несложно проделать для ресторанов, отелей и т.п.: food, decor, service и синонимы.



Online-анализаторы тональности

- Sentiment Analysis with Python NLTK Text Classification
  - Использует наивный байесовский классификатор
  - Не работает с русским языком
- http://ston.apphb.com/index.html
  - Размечает слова по тональности (с помощью словарей)
  - Работает с русским языком