Препроцессинг: Сегментация и токенизация корпуса рецензий к фильмам

...

Разработчик, аналитик (описание общей организации программы): Сонина Полина
Тестировщик, аналитик (резюме статьи): Стырина Соня

3 курс ФиКЛ, Школа лингвистики, ВШЭ
Проект в рамках курса “Автоматическая обработка естественного языка”

Аналитическая часть:

краткое резюме статьи

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Задача — сегментация на предложения в текстах, полученных с помощью автоматической обработки записи речи (ASR — Automatic speech recognition systems)

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

ASR-тексты:

- обычно целиком в одном регистре
- отсутствует пунктуация
- возможны ошибки в транскрипции
- при сегментации на предложения вручную обнаружена значительная разница между разметкой текстов в одном регистре и текстов в смешанных регистрах (mixed case text) (во втором случае размечали значительно лучше)
- также было выявлено, что различные размечающие не всегда делят текст на предложения одинаково

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Золотой стандарт:

- тексты из Wall Street Journal, с разметкой POS и с пунктуацией

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Метод:
Timbl memory-based learning algorithm (case-based / lazy learning) (Daelemans et al., 1999):

- берется набор примеров для обучения
- новые данные сравниваются с набором изученных примеров и на основании этого помечаются как “sentence_boundary” или “no_boundary”
- для данных, не имеющих точного аналога в наборе изученных примеров, находится максимально похожий из последних

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Метод:

- 90% текстов — для обучения, во время которого была собрана информация о 13 признаках
- 10% текстов — тестовый корпус со следующими преобразованиями:

    1. удаление пунктуации
    2. маркировка первых 12 признаков
    3. при повторном эксперименте: конвертация в uppercase (удаление 6, 12 признаков)

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Используемые признаки:

1. предыдущее слово
2. вероятность того, что предыдущее слово — последнее в предложении
3. часть речи предыдущего слова
4. вероятность того, что часть речи (3-й признак) приписана последнему слову в предложении
5. маркер того, является ли предыдущее слово “stop word”
6. маркер того, пишется ли предыдущее слово с заглавной буквы
7. следующее слово

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Используемые признаки:

8. вероятность того, что следующее слово стоит в начале предложения

9. часть речи следующего слова

10. вероятность того, что часть речи (9-й признак) приписана слову в начале предложения

11. маркер того, является ли следующее слово “stop word”

12. маркер того, пишется ли следующее слово с заглавной буквы

13. маркер “sentence_boundary” или “no_boundary”

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Результаты:

алгоритм разбил mixed-case тексты на предложения с 76% точностью

same-case тексты — с 35% точностью

Case information	P	R	F
Applied	78	75	76
Not applied	36	35	35

Table 3: Results of the sentence boundary detection program

“Experiments on Sentence Boundary Detection”

Mark Stevenson and Robert Gaizauskas

Результаты:

- сегментация на предложения — задача, значительно более сложная для ASR-текстов, чем для стандартных текстов (mixed case, punctuated)
- эксперимент помог осветить данную задачу как нетривиальную и актуальную в сфере NLP

Аналитическая часть:
Описание системы

Описание системы

Формат данных на входе: коллекция из 400 файлов .txt с текстами рецензий (более 100,000 словоупотреблений)

Формат выдачи: файл tokens.txt

Формат строки данных на выходе: ID документа ID токена токен

Описание системы: особенности и сложности

Сокращения с точками:

Т.е., т. е., т. к., т. д., т. п., в т. ч., т. з.

P.S., p.s., P. S., ps, vs., Vs., E.T. (название фильма)

др., и пр., млн., г., гг., физ. (физически)

кот., сек., студ. (студенческий), гл. г фильма

Быт., гл. 18, см. (смотри), тыс., реж. (режиссерская версия или режиссер)

Музык. Оформление

Описание системы: особенности и сложности

Аббревиатуры с заглавными буквами:

- *ТВ3, СТС (канал)*
- *БПС (Бэтмен против Супермена)*
- *США, СССР, КНР*
- *Х23*
- *ФБР*
- *СМИ*
- *ГГ (главный герой)*
- *ЛЛЛ («Ла-Ла-Ленд»)*
- *БМВ*
- *VHS, DC*
- *ПМС*

Описание системы: особенности и сложности

Дефисы в составных словах:

- *Город-сказка, город-мечта, рок-н-ролла, «Я»-человека*
- *ТВ-шоу, роуд-муви, ЛГБТ-движения*
- *Лучшие сезоны — 1-3 и 7.*
- *двух хороших людей-они подумают и поймут* (опечатка)
- *общение сейчас-это просто набор* (опечатка)

Сокращения через дефис:

- *д-иях*
- *мн-ва*

Описание системы: особенности и сложности

Цифровые и буквенно-цифровые шаблоны:

- *18+*
- *100%, 150%, 850%, ...*
- *30+*
- *70 — х (именно с тире)*
- *3-й, 9-ый, 4-ю, 1-го*
- *1961-62 годы*
- *$ 20 миллионов*
- *1,5*
- *20:00*
- *1. 3)*
- *3D 2D*
- *PG-13*
- *1970-х, 90-х, 2-х*
- *начало 20 века*
- *XX века (XX - кириллица)*
- оценка *“8 из 10”*
- *8 из 10и. 9,5 из 10, 8 из 12* (пользователь изменил шаблон)

Описание системы: особенности и сложности

Специфика имен и названий:

- *Шиндлер-Малер-Гропиус-Верфель*
- *МакГрегор*
- *Sur le fil -- название фильма внутри рецензии*
- *Superперцы -- смешение алфавитов*
- инициалы:
  - *О. Дж. Симпсона*
  - *Э. А. По*

Описание системы: особенности и сложности

Слова через / :

- *Весь сезон — 9/10*
- *до 16/20 серии*
- *и пытается увидеть/услышать*
- *и его пути/принятие себя.*
- *на тему бедных/богатых*
- *а особенно пейринг Маринетт/Кот*
- *фанат творчества Дэнни Бойла/ первой части в частности*
- *Sweet / Vicious* (название фильма)

Программная часть:
Описание общеи организации программы

Описание организации программы:

- Текст файла .txt → предварительный список токенов

- Знаки препинания выделяются в токены
  - . , ( ) " ? ! : ; — /
  - ???
  - !!!
  - ?!
  - … (спецсимвол)
  - ©
  - *20:00* *7,5* *9/10* (не делятся)

Описание организации программы:

Создание словарей для более точной токенизации:

- аббревиатур (abbreviations.txt) : *т.д. т.п. P. S. ...*
- слов с дефисом (hyphen_dict.txt) : *Нью-Йорк, человек-паук, альтер-эго, ...*
- неотделяемых частиц : *-то по- во- в- -нибудь ...*

слова с дефисом вне словарей → разбиваются на несколько токенов

Описание организации программы:

- Аббревиатуры выделяются в токены
- Апостроф не делит токен (*Трейнспоттинг`а, Underworld`а, won`t*)
- Слэш делит на токены (*увидеть/услышать*), кроме положения между цифрами (*9/10*)
- Имена собственные из более чем одного слова выделяются как один токен
- Концы предложений выделяются пометкой </se>

Тестирование

Тестирование

- для тестирования сегментации на предложения было взято 129 предложений из корпуса

- тестирование токенизации было произведено на 689 токенах из корпуса

Тестирование

Результаты:

Тест сегментации на предложения:

- Точность программы: 126/129 — 97% маркеров совпадают с тестовым корпусом, размеченным вручную

Тест на токенизацию:

- Точность программы: 686/689 — 99% токенов совпадают

Тестирование

Ошибки программы во время тестирования (сегментация на предложения):

Program results		Manual results
Кинг Конг. Тем	vs.	Кинг Конг .</se>
Сэмюел Л .</se> Джексон	vs.	Сэмюел Л. Джексон
. “	vs.	.</se> “

Тестирование

Ошибки программы во время тестирования (токенизация):

Program results		Manual results
Потому Бога	vs.	Потому Бога
Сначала Логан	vs.	Сначала Логан
На Логана	vs.	На Логана

Тестирование

Замечания по результатам тестирования:

Основные проблемы при сегментации на предложения:

- имена собственные с инициалами
- названия, имена собственные в начале/конце предложений

Основные проблемы при токенизации:

- “склеивание” имен собственных в один токен
- имена собственные с инициалами
- слова с дефисом
- установление нижнего регистра в первом слове предложения (опять же, из-за названий и имен собственных)

Ссылка на проект в GitHub:

https://github.com/SoDipole/hse_nlp3year/tree/master/project_preprocessing