Exercise in linear models

Tobias Eder

CIS

tobias.eder@in.tum.de

Februar 2018

Übersicht

Task

Wir suchen spezifische Informationen in unstrukturieren (Text-)Daten.

Dataset: Carniege Mellon School of Computer Science (CMU) Seminar Announcements

Beispiel

<0.5.5.93.11.12.56.skees+@SKEES.ADM.CS.CMU.EDU (Jim Skees).0>
Type: cmu.cs.scs
Topic: Healthy Office Seminar
Dates: 12-May-93
Time: <stime>1:00 PM</stime>
PostedBy: skees+ on 5-May-93 at 11:12 from SKEES.ADM.CS.CMU.EDU (Jim Skees)
Abstract:

SCS will sponsor a seminar on ways to achieve a healthy
office environment at <stime>1 p.m</stime>. Wednesday, May 12th, in <location>Wean
Hall 5409</location>.
The speaker will be <speaker>Karlis Mateus</speaker>, a representative of
Steelcase, Inc., the office equipment manufacturer. He will
focus on the ways in which poorly designed office furniture
can contribute to carpal tunnel syndrome and other illnesses.

I have asked <speaker>Mr. Mateus</speaker> to keep the discussion as generic as
possible. However, I do anticipate he will use some illustrations
of Steelcase products during his presentation.

Tags

- stime - Startzeit
- etime - Endzeit
- location - Veranstaltungsort
- speaker - Vortragender

Setup

- Material von der Kurswebsite herunterladen.
- Verzeichnis entpacken: `$ tar xzf IE_exercise2.tgz`
- Ins Verzeichnis `IE_exercise2/` wechseln.

Setup Part 2

- Im Root-Verzeichnis der Übung folgendes Ausführen:
- Dateien vorbereiten: `$ make data`
- Erster Test-Run: `$ make`
- Gibt uns einen F-Measure Score wenn alles geklappt hat:

```
ederto@fruehjahrslorchel.cip.ifi.lmu.de:~/Uni/WiSe1718/IE_exercise1 $ make
Applying rules...done
true positives: 16
false positives: 1046
false negatives: 365
all: 381
Precision: 0.015065913370998116
Recall: 0.04199475065616798
F1: 0.022176022176022176
ederto@fruehjahrslorchel.cip.ifi.lmu.de:~/Uni/WiSe1718/IE_exercise1 $
```

Task Details

- Der Task ist mit IOB-Tags modelliert.
- Ein Token hat das Tag O wenn es auerhalb eines location Tags steht.
- Tag B heisst Beginn einer Location
- Tag I steht für innerhalb einer Location (vor dem Ende-Tag)

Feature Vektoren

- Features müssen als Feature-Vektor repräsentiert werden.
- Jedes Feature ist ein key-value pair in einem Dictionary.
- Key: Name des Features (arbiträr)
- Value: Ausprägung des Features an der konkreten Stelle im Text.
- Features können sich per Index auf vorherige / folgende Token beziehen.
- p features bei n token ergibt nxp matrix.

Feature Extraction

- Datei `feature_extractor.py` im Editor öffnen:

```
20 def feature_extractor(fullset):
21     raw_features = list()
22     for document in fullset:
23         for i, entry in enumerate(document):
24 
25             raw_features.append({'isRoom' : entry == "Room",
26                                  'isUpper' : entry.isupper()})
27 
28     return raw_features
```

- Eigene Feature entwickeln!