Maschinelle Sprachverarbeitung

Übung

Aufgabe 4: Klassifikation von Filmbewertungen

Mario Sänger

mario.saenger@informatik.hu-berlin.de

Textklassifikation

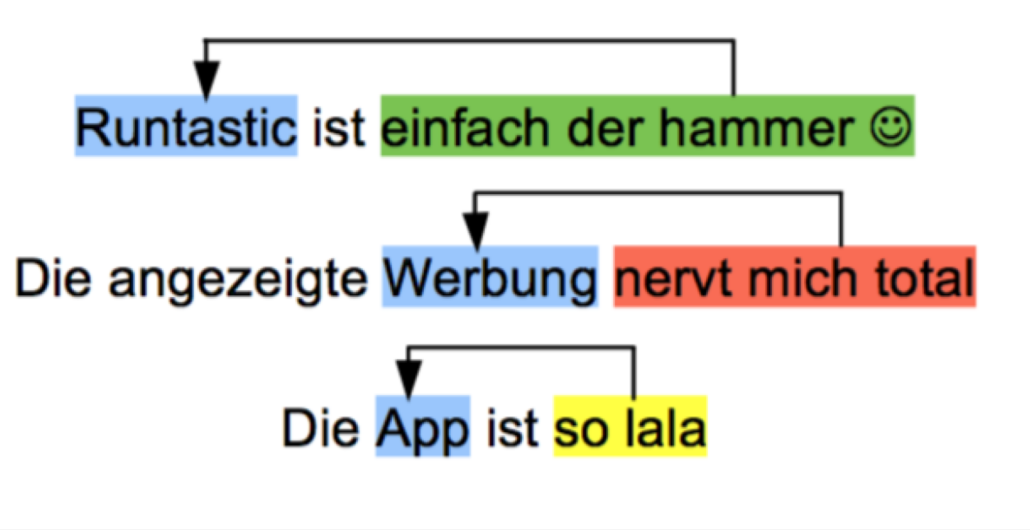
- Sentiment Analyse: Automatische Stimmungs- bzw. Meinungsanalyse von Texten
  - Erkennung der grundlegenden Haltung des Texts (positiv/negativ)
  - Fein-granulare Abstufungen (bspw. 5 Sternbewertung) und Analysemöglichkeiten (Meinungshalter? Zielobjekt?)

Schickes Gerät für wenig Geld. Inbetriebnahme läuft unkompliziert und das Design ist einfach aber zeitlos. [...]

Bild- und Tonqualität waren schlechter, als bei meinem 4 Jahre alten Smart-TV in ähnlicher Preisklasse.[...]

Textklassifikation

- Sentiment Analyse: Automatische Stimmungs- bzw. Meinungsanalyse von Texten
  - Erkennung der grundlegenden Haltung des Texts (positiv/negativ)
  - Fein-granulare Abstufungen (bspw. 5 Sternbewertung) und Analysemöglichkeiten (Meinungshalter? Zielobjekt?)



Aufgabenstellung

- Sentiment Analyse von Filmbewertungen
  - Ist eine gegebene Filmbewertung grundlegend positiv oder negativ?
- Implementierung eines Klassifikationsprogramms zur automatischen Analyse der Bewertungen
  - Bereitstellung eines Datensatzes mit Trainingsbeispielen
- Grundlage der (autom.) Entscheidung ist der Text
  - Keine weiteren Metadaten (bspw. Nutzer etc)!

Bewertungsdatensatz

- Bereitstellung eines Trainingskorpus mit Bewertungen
  - Je 12.500 positive und negative Bewertungen
  - https://hu.berlin/ue_masprach1819_ass4_corpus

- Korpus ist als tab-separierte Datei gegeben
  - Filmbewertungen können
 als „Zeilentrenner“ enthalten
  - Quote-Zeichen ist “

```
ID      Label   Text
20625   neg     I will be short...This film is an embarrassment to
                everyone ...
21350   neg     This is quite possibly the worst film I have ever seen.
                ...
23305   pos     "Honestly, I was expecting to HATE this one ..."
```

Realisierung / Gestaltungsmöglichkeiten

- Auswahl der Klassifikationsmethode
  - Support Vector Machine, Naive Bayes, K-Nearest-Neighbor, Random Forests, Max-Entropy, Künstliche Neuronale Netze, ...
- Verwendung beliebiger Bibliotheken
  - Java: Stanford Core NLP, OpenNLP, Lingpipe, Weka, LibSVM, Mallet, Deep-Learning-4J, ...
  - Python: NLTK, Scikit-Learn, Gensim, Keras, Tensorflow, PyTorch, ..
- Nicht erlaubt: Einsatz spezieller Sentiment-Analysis-Tools!
  - Verwendung von zusätzlichen Ressourcen ggf. möglich
  - Einsatz muss abgesprochen werden!

Realisierung / Gestaltungsmöglichkeiten

- Auswahl und Repräsentation der Features:
  - Darstellung: Binär? TF*IDF? Word Embeddings?
  - Lower-casing? Stopp-Word-Removal?
  - Erkennung von Signalwörtern oder -pattern?
  - Trennung von Plot-Beschreibung und „Meinungsabschnitten“?
  - Position von Meinungsäußerungen relevant?

- Feature-Selection:
  - Manuelle Auswahl? Information Gain? Keine Reduktion?

Aufgabendetails

- Euer Programm unterstützt zwei Modi: Trainings- und Klassifikationsmodus

- 1. Trainingsmodus:
  - Lesen der Trainingsbeispiele und Lernen eines Modells
  - Speichern des Modells in „*model_name*“ im Ausführungsverzeichnis

```
java –jar uebung4-gruppeX.jar train model_name \
      training_tsv_file
```

```
python uebung4-gruppeX.py train model_name \
      training_tsv_file
```

Aufgabendetails

- 2. Klassifikationsmodus
  - Lesen des Modells „*model_name*"aus dem aktuellen Verzeichnis
  - Lesen aller Filmbewertungen aus „*test_tsv_file*"
  - Ergebnisausgabe: „id**\t**pos/neg" in „*result_file*"

```
java –jar uebung4-gruppeX.jar classify model_name \
      test_tsv_file result_file

python uebung4-gruppeX.py classify model_name \
      test_tsv_file result_file
```

Klassifikationsmodus

- Eingabe: Analog zum Trainingskorpus
  - Labels sind jedoch maskiert

```
20625    -        I will be short...This film is an embarrassment to
21350    -        This is quite possibly the worst film I have ever seen.
23305    -        "Honestly, I was expecting to HATE this one ..."
```

- Ausgabe: Nur ID und berechnete Klasse (tab-separiert)
  - Text muss nicht ausgegeben werden!

```
20625    neg
21350    neg
23305    pos
```

Aufgabendetails

- Python: Angabe der verwendeten Version und Auslistung aller Dependencies
  - Bereitstellung einer *requirements.txt* mit allen Abhängigkeiten
  - https://pip.pypa.io/en/stable/reference/pip_install/#requirements-file-format

requirements.txt

```
sklearn

gensim

tensorflow == <version>
```

Abgabe

- Abgabe eines ZIP-Archivs uebung4-gruppeX.zip
  - Archiv enthält ausführbares Programm und den Quellcode
  - Ergebnisse der 10-fach Kreuzvalidierung (inkl. Durchschnitt, Standardabweichung und Median)
  - Python: Auflistung der Dependencies (*requirements.txt*)
- Testet Euer Programm vor der Abgabe!
- Abgabe bis spätestens Do, 10.01.2019, 23:59 Uhr über: https://hu.berlin/ue_masprach1819_ass4

Wettbewerb

- Die Lösung mit der höchsten Genauigkeit gewinnt
  - Genauigkeit = Anteil korrekt klassifizierter Bewertungen
- Geschwindigkeit ist diesmal irrelevant!
- Die drei besten Teams erhalten 5/3/1 Punkte

Fragen?