Übungen 2: Supervised Learning

Abgabedatum Sonntag, 1. April 2018, 13:00h MEZ

Setup

Installiere Anaconda für Python 3.6 wie auf der Web-Site[1] beschrieben. Um die benötigten Packages zu installieren, führe im *Terminal* oder in der *Eingabeaufforderung* folgenden Befehl aus:

```
conda install nb_conda
conda env create -f ki2_fs2018_env.yml
```

Um die Daten nutzen zu können, müssen wir diese über das NLTK-Toolkit herunterladen. Führe dazu folgende Befehle aus:

```
$ python
>>> import nltk
>>> nltk.download("names")
>>> nltk.download("europarl_raw")
```

Um das Notebook zu starten gehe im Terminal zum Pfad in dem das *Language Classification.ipynb* und das *Sklearn Intro.ipynb* Notebook gespeichert sind und führe folgenden Befehl aus:

```
jupyter notebook
```

1 Toxic Comment Classification Challenge

In dieser Aufgabe werden wir einen Hate-Speech Klassifikator bauen. Dafür verwenden wir die Kaggle-Daten[2]. Der Datenatz besteht aus Wikipedia Kommentaren, die mit verschiedenen binären Labels versehen sind. Die Labels sind: Toxic, Severe Toxic, Obscene, Threat, Insult, und Identity Hate.

Das Ziel dieser Übung ist es, die Supervised Learning Techniken auf diese Daten anzuwenden. Als Metrik werden wir den F1-Score auf der Toxic Klasse verwenden. Das bedeutet:

```
from sklearn.metrics import f1_score
f1_score(y_true, y_pred, average='binary', pos_label=1)
```

Dieser Datensatz ist stark unbalanciert, dh ca 90% der Daten sind nicht Toxic. Beachtet, dass einige Klassifikatoren sehr sensibel darauf reagieren und man etwas *class balancing* betreiben muss.

1.1 Verstehen der Codevorlage

1. In den Unterlagen findet ihr das Notebook `Toxic Comment Classification Challenge.ipynb`. Dort findet ihr eine einfache Baseline auf der ihr aufbauen werdet.

2. Falls ihr `gensim` nicht kennt, studiert die Dokumentation [3].

[1] https://www.anaconda.com/download/
[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data
[3] `https://radimrehurek.com/gensim/`

3. Für Wordembeddings gibt es sehr viel Auswahl, im Code werden die GloVe embeddings verwendet mit 25 dimensionen. Für mehr Embeddings könnt ihr Word2Vec[4], GloVe[5] oder FastText[6] anschauen.

1.2 Aufgabe

Viele Tasks im Maschinellen Lernen finden im Rahmen eines sogenannten *shared task* statt. Dabei wird ein Datensatz veröffentlicht und ein Wettbewerb ausgeschrieben, wer den besten score auf dem Datensatz erzielt. Solche Tasks fördern die Forschung auf einem Gebiet. Für diese Aufgabe werden wir einen mini-shared-task durchführen. Wir werden uns auf das *Toxic* Label fokussieren. Das Ziel ist es den F1-Score zu maximieren. Dazu könnt ihr:

- Neue Features entwickeln: TF-IDF, andere Word Embeddings, neue Word Embeddings trainieren, nach Wortlisten suchen, etc..
- Modelle testen: Die Sklean library duchgehen und den Classifier finden, der am besten Funktioniert.
- Ensembing: mehrere Modelle trainieren und anschliessend deren Output kombinieren.
- Testet eure eigenen coolen Ideen :)

2 Abgabe

- Als Abgabe erwarten wir einen kurzen Bericht über euer System: welche Features habt ihr verwendet, welche Modelle, etc. Es soll nachvollziehrbar sein, dh jemand sollte in der Lage sein den Bericht zu lesen und euer System nachzubauen.
- Das ausgefüllte Testset, damit wir herausfinden können wer das beste System hat. Benutzt dafür den vorgegebenen Code, welches ein CSV File generiert. Das CSV File besteht aus: *Id, text, predicted label*.
- Die Abgabe erfolg über ein Zip File, wo die beiden oben genannten Datein enthalten sind. Stellt sicher, dass ihr in der Abgabe die Namen und Kürzel aller Teammitglieder reinschreibt.

3 Feedback

Bitte melde kurz zurück, welche Aufgaben du hilfreich und welche du nicht hilfreich fandest, um dein Lernen zu unterstützen.

Viel Spass und Erfolg beim Programmieren!

[4]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit
[5]https://nlp.stanford.edu/projects/glove/
[6]https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md