Altersklassifikation auf deutschen Tweets

Basierend auf dem Paper von Nguyen: “How Old Do You Think I Am?”

Das Ziel dieses Projektes ist es, das Alter eines Twitter-Users nur anhand eines Tweets herauszufinden.

Tweets runterladen und User klassifizieren (Ordner TweetCollecting)

In dem Paper von Nguyen wird beschrieben, wie über einen längeren Zeitraum hinweg viele User von Hand klassifiziert wurden. Da ich die Zeit dafür nicht hatte, musste ich einen anderen Weg finden, um User einer Altersgruppe zuordnen zu können. Ich habe deshalb nach Tweets aus der letzten Woche gesucht, in denen "Alles Gute zum xy" vorkommt, wobei xy ein bestimmtes Alter ist. Ich beschränkte mich hierbei auf folgende Alter: 16, 18, 20, 25, 30, 35, 40, 45, 50. Die extrahierten Tweets bin ich von Hand durchgegangen und habe passende User rausgesucht.

Geeignete Tweets sind zum Beispiel:

"Hey Juli mein bester, alles gute zum 18. Wünsch ich dir! @poepelhjuli_fch"

"@Willi1887 Alles liebe und gute zum 50. Geburtstag Nachbar!"

"Alles Gute zum 30. Geburtstag @McPucki"

Fehleranalyse und Verbesserungsvorschläge

- Bei manchen Tweets wurde die Person nicht getaggt:

  "Alles Gute zum 30., Harry. Altes Haus. #DanielRadcliffe"

  Diese ungeeignete Tweets könnte ich nächstes Mal direkt filtern, indem ich nur nach Tweets mit Mentions suche.

- Es wird bei Geburtstagsglückwünschen nur selten das Alter dazu geschrieben - vor allem bei den älteren Menschen. Viele Tweets habe ich hingegen gefunden zum 18. Geburtstag.

- Etwas seltener gab es auch Tweets, bei denen "Alles Gute zum xy" vorkam, doch es sich nicht um einen Glückwunsch zum Geburtstag handelte.

"**Alles** fertig? **Gute** Frage! Vor 1,5 Jahren habe ich meinen #Camper-Ausbau begonnen - ein wirklich großes Projekt: vom #Rettungswagen **zum** #Reisemobil. Im neuen VLOG seht Ihr, ob mein #LandRover #Defender \"Emily\" reisefertig ist... Viel Spaß mit Episode **35**!"

Wortfrequenzen berechnen (age_trainer.py)

In dieser Klasse zählte ich alle Wörter aller User und konnte so pro Altersgruppe die Wortfrequenzen berechnen. Diese speicherte ich ab, um sie später verwenden zu können (unter *Modelle/word_freq.json*). Außerdem zählte/berechnete ich folgende Features für jede Altersgruppe (*Modelle/special_freq.json*):

- Anzahl Retweets
- Anzahl Mentions
- Anzahl Hashtags
- Anzahl Links
- Durchschnittliche Wortlänge
- Durchschnittliche Tweetlänge
- Anzahl Wörter in Großbuchstaben
- Anzahl der Kommas
- Anzahl der Tweets

Werte der Features berechnen (age_values.py)

Ich wollte für jedes Wort einen Wert berechnen, wobei ein negativer Wert für ein niedriges Alter und ein positiver Wert für ein höheres Alter spricht. Das mittlere Alter aller Tweets war 30 Jahre. Demnach gab ich der Altersgruppe 30 den Wert 0. Von da an legte ich die Werte für alle anderen Altersgruppen fest:

```
"16": -14
"18": -12
"20": -10
"25": -5
"30": 0
"35": 5
"40": 10
"45": 15
"50": 20
```

Ich bin alle Tweets aller Altersgruppen durchgegangen und habe die Werte für die Wörter aufaddiert. Folgend teilte ich sie durch die Anzahl der Wörter.

Beispielberechnung

Wort: *haha*
Vorkommen: 16: 6 mal, 18: 4 mal, 35: 5 mal, 45: 2 mal
Summierung: (6 x -14) + (4 x -12) + (5 x 5) + (2 x 15) = -84 -48 + 25 + 30 = -77
Division: -77 / 17 = -4,53
Das Wort *haha* wird demnach bei jüngeren Usern öfter benutzt.

Folgende Wörter waren nach dieser Berechnung besonders signifikant (*Modelle/wortwerte.txt*):

Junge User (16 Jahre)	
xd	-25.044941931217167
^	-20.552061386305812
:d	-19.781304519844028
<3	-18.996779909514974
:(	-17.317958816687835
minecraft	-16.365156942317757
✨	-15.28261939924989
uff	-14.98635846842045

Ältere User (50 Jahre)	
;-)	25.118035126902768
:-)	21.5340955389254
csu	20.16550161543144
spd	20.15467329682824
bundestag	20.123801214968616
bundesregierung	20.11413475564596
merkel	20.086827441644736
europa	19.8673480283111

Weiterhin ergaben sich für die anderen Features folgende Werte (*Modelle/spezialwerte.json*):


```
"MENTIONS": -1.150186083611065,
"CAPITALIZED": -0.06579758621660581,
"HASHTAGS": 0.6782970223825797,
"RETWEETS": 3.2719958810875194,
"LINKS": 3.8280287578292036
```


Jüngere User benutzen demnach öfter Mentions und schreiben Wörter in Großbuchstaben (LOL). Hashtags, Retweets und Links sind Zeichen für ältere User.

Bis auf die Retweets überschneiden sich diese Ergebnisse mit denen aus Nguyens Paper (dort waren Retweets ein Zeichen für jüngere User). Auch die Wort- und Tweetlänge stieg in meinen Daten, wenn auch nur minimal, mit dem Alter an.

Klassifizierung von Usern/Tweets (age_classifier.py)

Ich habe auf dem Trainset den durchschnittlichen Wert für einen Tweet für alle Altersgruppen berechnet. Dazu summierte ich die Werte aller Wörter auf und teilte diesen durch die Anzahl aller Wörter. War das Wort "besonders" (Hashtag, Link, etc.) benutzt ich den Wert aus spezialwerte.json. Das Ergebnis sah wie folgt aus (*Modelle/werte_alter.json*):

Alter	Größter Wert	Kleinster Wert	Durchschnitt
16	10.554	-19.062	0,255
18	12.599	-20.617	0,679
20	21,468	-20,618	1,136
25	14.440	-10.106	1.207
30	12.247	-13.353	1.274
35	11.951	-17.383	2.642
40	11.980	-10.106	1.818
45	14.440	-5.291	2.364
50	21.468	-10.498	2.513

Es ist zu erkennen, dass die Tweets von jüngeren Usern durchschnittlich einen niedrigeren Wert haben, als die von älteren Usern. Allerdings ist leider keine lineare Entwicklung von jung zu alt erkennbar. Während der durchschnittliche Wert von 16 bis 35 Jahren konstant ansteigt, sinkt er bei 40 wieder stark ab. Das liegt wahrscheinlich vor allem daran, dass ich nur zehn User im Alter von 40 Jahren zum Training benutzen konnte, und der Wert dadurch von wenigen Usern stark beeinfluss wird. Auch ist zu sehen, dass der größte Durchschnitt bei Usern im Alter von 35 Jahren gemessen wurde. Hier hatte ich ebenfalls nur fünf User, was offensichtlich nicht aussreicht, um ein gutes Abbild eines "typischen" 35-jährigen zu erfassen.
Trotzdem legte ich anhand dieser Werte für jede Altersgruppe ein Maximum fest, bis zu dem ein Tweet noch zu einer Altersgruppe gehören kann.

`0.3: **16**, 0.7: **18**, 1: **20**, 1.2: **25**, 1.5: **30**, 2: **35**, 2.2: **40**, 3: **45**`

D.h. wenn ich für einen Tweet einen Wert von 0.8 berechnet habe, dann gehört er nicht zur Gruppe der 16- oder 18-jährigen, denn dort liegt er schon über dem Maximum. Stattdessen würde dieser Tweet zur Gruppe der 20-jährigen eingeordnet werden.

Ergebnisse

Ich testete auf Tweets von den selben Usern (was auf jeden Fall nicht ausreicht). Mit den Grenzen wie ich sie oben angegeben habe erhielt ich eine Precision von 0,132. Ich sah, dass die meisten Tweets eher zu alt als zu jung eingeschätzt wurden, weshalb ich die Grenzen etwas nach oben hin verschob:

```
0: 16, 2: 18, 2.5: 20, 3.0: 25, 3.3: 30, 3.5: 35, 4.0: 40, 4.5: 45
```

Dadurch erhielt ich eine **Precision von 0,15** und folgende Recall-Werte für die Altersgruppen:

```
16: 0,10 - Tweets gesamt: 12.722
18: 0,25 - Tweets gesamt: 13.066
20: 0,05 - Tweets gesamt:  3.666
25: 0,09 - Tweets gesamt:  3.490
30: 0,12 - Tweets gesamt:  2.101
35: 0,07 - Tweets gesamt:  1.175
40: 0,20 - Tweets gesamt:  2.793
45: 0,16 - Tweets gesamt:  2.138
50: 0,18 - Tweets gesamt:  5.982
```

Die besten Werte wurden bei den 18-jährigen und 40-jährigen erreicht. Die schlechtesten bei den 20-jährigen. Ich kann mir vorstellen, dass es bei den 18-jährigen gute Werte gab, weil ich dort viele Trainingsdaten hatte (22 Personen). Bei den 20-jährigen hingegen hatte ich nur sechs Personen, und auch bei den 35-jährigen nur fünf. Ich könnte leicht die Auswahlgrenzen weiter nach unten schieben und eine bessere Precision erreichen, nur weil dann mehr Tweets von den 16-jährigen richtig klassifiziert werden würden. Doch dann würde der Recall der anderen Altersgruppen sich deutlich verschlechtern.

Einteilung in Altersklassen

Wenn ich die Tweets nicht einem bestimmten Alter, sondern einer Altersklasse (also: unter 20, 20-30, 30-40 und über 40) zuordne, erhalte ich eine **Precision von ca. 0,41**.

Fehleranalyse und Verbesserungsvorschläge

1. Mehr Tweets würden vermutlich zu deutlich besseren Ergebnissen führen. Doch für die manuelle Annotation hatte ich keine Zeit, weshalb ich mich mit der "Alles Gute zum"-Strategie zufrieden geben musste.

2. Sehr kurze Tweets würde ich nächstes Mal bei der Klassifizierung raus lassen:

```
Beispiel: 16 Jahre alt - richtig klassifiziert
@AnkulixC Jo Moment, teleportieren mich kurz xD
Wert: -3.055
Alterszuordnung: 16

Beispiel: 16 Jahre alt (kurz) - falsch klassifiziert
@syntax_manu Und jetzt?
Wert: 5.586
Alterszuordnung: 50
```

3. Weitere Features könnten zu besseren Ergebnissen führen (wie in dem Paper von Nguyen). Zum Beispiel habe ich die Wort- und Tweetlänge bei der Berechnung noch nicht berücksichtigt.

Zusammenfassung Projekt (Skripte)
age_trainer.py – Zum Berechnen der Frequenzen pro Altersgruppe
age_values.py – Zum Berechnen der Werte aller Worte
age_classifier.py – Zum Klassifizieren von Tweets

Zusammenfassung Ordner **TweetCollecting**
AllesGutezum – enthält extrahierte Tweets mit "Alles Gute zum xy"
Namen – Textdateien mit allen Twitter-Namen für die Altersgruppen
Tweets – alle extrahierten Tweets der User aus Namen
Testset – 20 % aller Tweets der User
Trainset – 80 % aller Tweets der User
gettweets.py – Skript zum downloaden aller Tweets eines Users
search_for.py – Skript zum Finden eines Strings in Tweets

Zusammenfassung Ordner **Modelle**
werte_alter.json – Max, Min und Durchschnitt der Tweets aller Altersgruppen
specialfreq.json – Frequenzen der Sondersachen aller Altersgruppen
spezialwerte.json – Werte der Sondersachen (zusammengefasst)
wortwerte.json – Werte der Worte
wortwerte.txt – Werte der Worte (sortiert)

Zusammenfassung Ordner **Ergebnisse**
class_with_special.json – Ergebnisse der Klassifizierung
user_results1.json – Ergebnisse User-Klassifizierung (alt)