Autorship recognition

Sára Štráchalová
May 3rd, 2023
NPFL054

Experimental data set

- 6 authors, 5 books per author
- delexicalized
- split to passages
- xml

n-gram feature extraction

- make.n-grams
- csv

```
* Number of passages processed = 10686
* Number of different (1,2,3)-grams found = 137912
* Total tokens processed = 3560458
*
* N-gram frequencies (sorted by cf, df threshold = 1)
     1          'POS_NOUN'                        cf = 690068     df = 9155       tf{1783} =  238
     2          'POS_VERB'                        cf = 394356     df = 9201       tf{1783} =  123
     3          'POS_ADJ'                         cf = 349516     df = 9021       tf{1783} =  130
     4          ','                               cf = 295614     df = 8905       tf{1783} =   80
     5          'POS_ADV'                         cf = 237158     df = 9030       tf{1783} =  102
     6          '.'                               cf = 208760     df = 8914       tf{1783} =   63
     7          'POS_ADJ POS_NOUN'                cf = 172164     df = 8933       tf{1783} =   59
     8          'POS_NOUN ,'                      cf = 125192     df = 8882       tf{1783} =   28
     9          'a'                               cf = 112712     df = 8898       tf{1783} =   41
    10          'být'                             cf = 109998     df = 8802       tf{1783} =   32
    11          'se'                              cf = 92688      df = 8614       tf{1783} =   19
    12          'POS_PROPN'                       cf = 85314      df = 7626       tf{1783} =   45
    13          'POS_NOUN .'                      cf = 80726      df = 8701       tf{1783} =   34
    14          'POS_NOUN POS_ADJ'                cf = 75208      df = 8359       tf{1783} =   38
    15          'POS_NOUN POS_NOUN'               cf = 67718      df = 8309       tf{1783} =   25
    16          'POS_NOUN POS_VERB'               cf = 65686      df = 8591       tf{1783} =   17
    17          'v'                               cf = 56692      df = 8357       tf{1783} =   25
    18          'POS_VERB ,'                      cf = 55282      df = 7876       tf{1783} =   19
    19          'ten'                             cf = 50998      df = 8083       tf{1783} =   22
    20          'on'                              cf = 48510      df = 7402       tf{1783} =    6
```

n-gram feature extraction

- n-grams from all passages
  - used only 400 most frequent n-grams
- author and frequencies for each passage -> create dataset

- shell script
- python script -> prepares rows, maps via dictionaries

n-gram feature extraction

```
n=$(grep -E '<passage' "$1" | wc -l)

base=$(cat "$1" | head -n 1 | cut -d " " -f 2 | cut -d "=" -f 2
| cut -d "\"" -f 2)

touch "$3"

cat "$2" | ./make.n-grams 1 > /dev/null

dataset_ngrams=$(cat ./ngrams.df1.csv | head -n 401 | cut -f 2 |
tail -n +2)
```

n-gram feature extraction

```
for i in $(seq 1 "$n")
do
    passage_idx=$((base+i-1))
    passage=$(cat "$1" | ./get.passages "pid=\"$passage_idx\"")
    author=$(echo "$passage" | head -n 1 | cut -d " " -f 3 |
            cut -d "=" -f 2 | cut -d "\"" -f 2)

    echo "$passage" | ./make.n-grams 1 > /dev/null
    freqs=$(cat ./ngrams.df1.csv | cut -f 2,3 | tail -n +2)
    python3 table_maker.py "$passage_idx" "$author" "$freqs"
        "$dataset_ngrams" >> "$3"
done
```

n-gram feature extraction

- make.n-grams
- df for n-grams -> needed to calculate idf

```
touch "$3"

cat "$2" | ./make.n-grams 1 > /dev/null

dataset_ngrams=$(cat ./ngrams.df1.csv | head -n 401 | cut -f 2 |
                 tail -n +2)
cat "$1" | ./make.n-grams 1 > /dev/null

df=$(cat ./ngrams.df1.csv | cut -f 2,4 | tail -n +2)


python3 table_maker-text.py "0" "0" "$df" "$dataset_ngrams" >> "$3"
```

Feature values

- in R
- **term frequency (tf)**
  - already completed
- **relative term frequency (rtf)**
  - normalized (divided by number of all n-grams in passage
- **weighted term frequency (tf*idf)**
  - idf = log(N/df)
    - N=total number of passages
    - df=number of passages that contain given n-gram

Feature values: tf

```
data.train <- psg.s200.train
data.train$V2 <- as.factor(data.train$V2)
idf.train <- idf.psg.s200.train
n <- ncol(data.train)

#tf
tf_features.train <- data.train
```

Feature values: rtf

```
rtf_features.train <- data.train


for (row in 1:nrow(rtf_features.train)) {
        all_terms <- sum(rtf_features.train[row,3:n])
        rtf_features.train[row, 3:n] <-
                rtf_features.train[row,3:n]/all_terms
}
```

Feature values: tf*idf

```
idf.train[1, 3:n] <- log(nrow(data.train)/(idf.train[1,3:n] + 1))
tf.idf.features.train <- data.train


for(i in 3:ncol(tf.idf.features.train)) {
        current.idf <- idf.train[1,i]
        tf.idf.features.train[ , i] <-
                                  tf.idf.features.train[ , i]*current.idf
}
```

SVM model & prediction

- LiblineaR
  - https://cran.r-project.org/web/packages/LiblineaR/index.html
- type 2 = L2 regularized L2 loss SVM

```
c=LiblineaR(data=tf_features.train[,3:n],
target=tf_features.train$V2, type=2, findC = TRUE)

tf.model=LiblineaR(data=tf_features.train[,3:n],
target=tf_features.train$V2, type=2, cost=c)


tf.pred=predict(tf.model,tf_features.test[,3:n])
```

Results

200x200	Precision	Recall	F-score
tf	92	92.14	92.07
rtf	84.51	84.69	84.6
tf*idf	89.97	90.12	90.05

200x1000	Precision	Recall	F-score
tf	99.17	99.39	99.28
rtf	96.68	96.98	96.83
tf*idf	91.1	94.08	92.57

1000x200	Precision	Recall	F-score
tf	86.39	86.54	86.47
rtf	81.3	81.15	81.23
tf*idf	82.48	82.82	82.65

1000x1000	Precision	Recall	F-score
tf	98.75	98.8	98.77
rtf	97.3	97.18	97.24
tf*idf	98.53	98.62	98.57

Interpunction

- Is it possible to recognize the authorship based on use of interpunction?

- data preparation
  - grep -E '<token>[\.,;:?!]*</token>|<passage.*|</passage>|<s>|</s>'
- features extracting
  - same as before (except cutting)

Interpunction - Results

200x200	Precision	Recall	F-score
tf	21.69	10.58	14.22
rtf	38.51	26.99	31.74
tf*idf	21.12	21.81	21.46

200x1000	Precision	Recall	F-score
tf	17.63	2.86	4.93
rtf	31.41	3.8	6.79
tf*idf	23.72	11.75	15.71

1000x200	Precision	Recall	F-score
tf	46.73	48.22	47.47
rtf	37.9	29.49	33.17
tf*idf	34.62	48.16	40.28

1000x1000	Precision	Recall	F-score
tf	36.26	3.68	6.67
rtf	42.63	12.45	19.27
tf*idf	50.84	14.69	22.8

Thank you ☺