Assignment 10 by Team 3

Ashutosh Agarwal, Shun-Lung Chang, Pooja Natu

This study was conducted in R. The source code can be found here.

1. Convert the html to text files and separate the individual news items. The individual press release items serve as documents.

The data was retrieved from the Jacobs University press release archives, and then compiled into a dataframe with three variables:

- id: serial number for each press release
- text: title and content of each press release
- year: year of issue

```r
colnames(text_df)
```

```
[1] "id"   "text" "year"
```

```r
text_df[2]
```

```
# A tibble: 638 x 1
                                                                       text
                                                                      <chr>
 1 Wissenschaft jenseits von Science Fiction: Jacobs University beteiligt sich
 2 Sozialer Mehrwert durch Musik? Begleitstudie der Jacobs University zur Symb
 3 Leibniz-Preis für Jacobs-Professorin Antje Boetius Dec , Antje Boetius, sei
 4 Neuer Förderpreis der Stiftung Mercator für Studierende der Jacobs Universi
 5 Management mit Zukunft: TiasNimbas Business School und Jacobs University st
 6 Deutscher Hochschulverband ernennt Katja Windt zur »Hochschullehrerin des J
 7 Der persönliche Eindruck zählt: Studienberater aus vier Kontinenten informi
 8 Spintronik: Physikerteam gelingt Nachweis eines nano-mechanischen Torsionse
 9 „Neue malerische Wendungen": University Club der Jacobs University zeigt ab
10 RWE startet CO-Konversions-Pilotanlage auf Basis einer von der Jacobs Unive
# ... with 628 more rows
```

2. Remove stop words and perform stemming.

Stop words can be discarded by `tokenizers::stopwords()`. Stemming can be carried out by `SnowballC::wordStem()`. The results are shown below.

```r
t <- text_df %>%
    unnest_tokens(word, text) %>%
    anti_join(tibble(word = c(stopwords("de"), stopwords("en")))) %>%
    mutate(stemmed_word = wordStem(word))
```

```
Joining, by = "word"
```

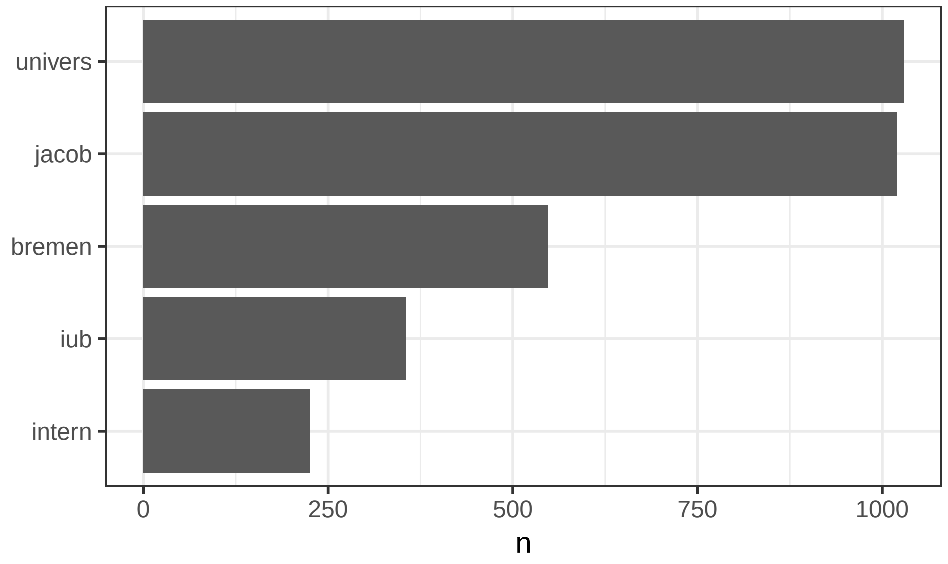
id	year	word	stemmed_word
1	2008	wissenschaft	wissenschaft
1	2008	jenseits	jenseit
1	2008	science	scienc

id	year	word	stemmed_word
1	2008	fiction	fiction
1	2008	jacobs	jacob
1	2008	university	univers

3. Perform a frequency analysis to compute the term-document (TD) matrix. What are the most common terms?

After the frequency analysis is performed by `dplyr::count()`, its results indicates that the most frequent three words in these press releases are 'univers', 'jacob', and 'bremen', which come as no surprise.

```r
top_5_words <- t %>%
    group_by(stemmed_word) %>%
    count(sort = TRUE) %>%
    ungroup() %>%
    slice(1:5)
```



Furthermore, the term-document matrix below shows that in the first ten press releases, the three most common terms did appear several times.

```r
word_counts <- t %>%
    group_by(id, stemmed_word) %>%
    count() %>%
    arrange(id, -n) %>%
    ungroup()

td <- word_counts %>% spread(stemmed_word, n, fill = 0) %>%
    select(-id) %>%
    as.matrix()
```

	univers	jacob	bremen	iub	intern
1	3	3	0	0	0
2	2	2	5	0	0
3	1	2	2	0	0
4	1	2	0	0	1

	univers	jacob	bremen	iub	intern
5	2	2	2	0	0
6	1	1	1	0	0
7	2	3	1	0	0
8	2	1	1	0	0
9	4	3	1	0	0
10	3	3	1	0	0

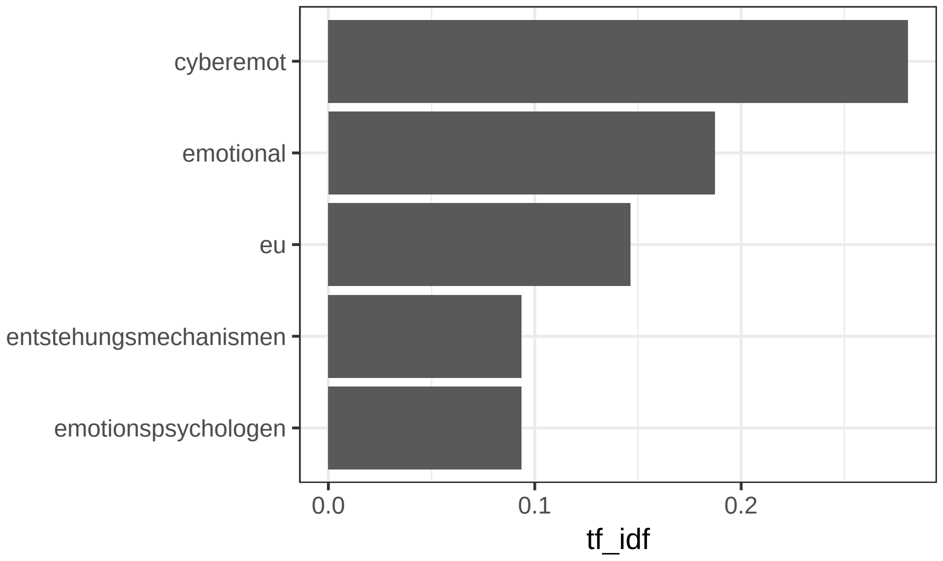
4. Compute inverse-document frequency (IDF) and term importance (TI). What are now the most common terms?

Inverse-document frequency and term importance in each press release can be acquired by `tidytext::bind_tf_idf()`. The following table shows that 'univers' gained low importance that resulted from its ubiquitousness in the whole corpus.

```r
tf_idf <- word_counts %>%
    bind_tf_idf(term = stemmed_word, document = id, n = n)
```

id	stemmed_word	n	tf	idf	tf_idf
1	cyberemot	3	0.0434783	6.4583383	0.2807973
1	eu	3	0.0434783	3.3672958	0.1464042
1	jacob	3	0.0434783	0.3490907	0.0151779
1	univers	3	0.0434783	0.1022306	0.0044448
1	emotional	2	0.0289855	6.4583383	0.1871982
1	projekt	2	0.0289855	2.4693542	0.0715755

A closer look at the term importance of the first document reveals that important terms are surrounding the topics of cyber and emotion. And if people check the press title, which is "Wissenschaft jenseits von Science Fiction: Jacobs University beteiligt sich am CyberEmotions-Project der EU", they can immediately understand the reason behind the results.



5. Compute pairwise cosine and Euclidean distance between all documents.

Applying `text2vec::dist2` on the term-document matrix can assist us in obtaining pairwise similarities (measured by cosine or euclidean distance) of two documents.

```r
cos_dist <- dist2(td, method = 'cosine')
euc_dist <- dist2(td, method = 'euclidean')
```

The following are cosine distance matrix and euclidean distance matrix, respectively. Both of them show the first 3 x 3 submatrix. The diagonal elements are all 0 since they are comparisons of identical documents. And lower off-diagonal values suggest more similar documents.

```r
cos_dist[1:3, 1:3] %>% kable()
```

1	2	3
0.0000000	0.8578515	0.8455115
0.8578515	0.0000000	0.7659177
0.8455115	0.7659177	0.0000000

```r
euc_dist[1:3, 1:3] %>% kable()
```

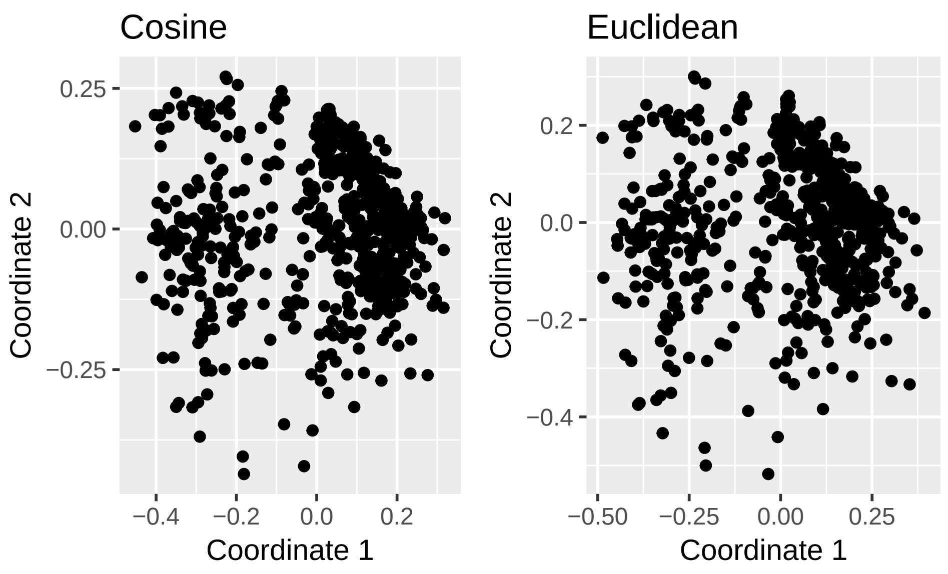
0.000000	1.309848	1.300393
1.309848	0.000000	1.237673
1.300393	1.237673	0.000000

6. Apply a multi-dimensional scaling approach to the distance matrix and render a 2D scatterplot. Compare the two distance metrics.

A multi-dimensional scaling approach can transform a high-dimentional matrix into low-dimentional space. To approach this task, we applied the classical multi-dimensional scaling (by `cmdscale()`) on the two matrices obtained in task 5.

```r
cos_dist_fit <- cmdscale(cos_dist, k = 2)
euc_dist_fit <- cmdscale(euc_dist, k = 2)
```

As can be seen from the scatterplots below, a cluster appear in the left-upper quadrant of both plots. Given that most topics of the press were unrelated, those dissimilar pairs in turn formed the cluster in the two plots.

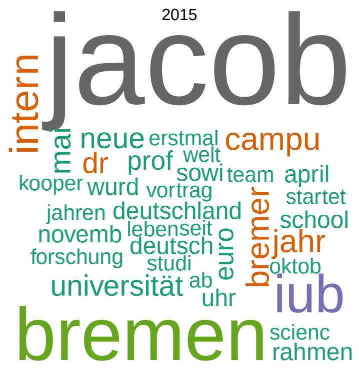


7. Capture the year of release during parsing and color code the scatterplot by time. Produce a Word Cloud for each year.

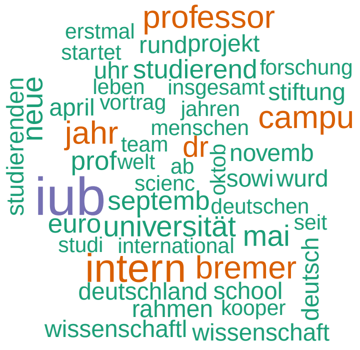
```
create_wordcloud <- function(year) {
    d <- t %>%
        filter(year == year) %>%
        group_by(stemmed_word) %>%
        count() %>%
        ungroup()

    wordcloud(words = d$stemmed_word, freq = d$n,
              max.words = 50, colors = brewer.pal(8, "Dark2"))
    text(x = 0.5, y = 1, cex = 0.5, as.character(year))
}
```

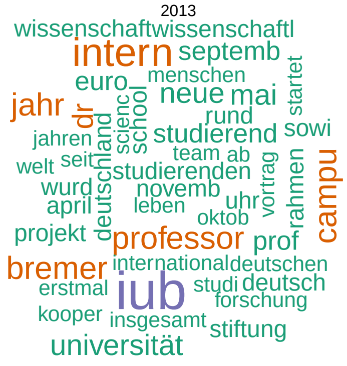
```
for (i in seq(2015, 2013, -1)) {
    create_wordcloud(i)
}
```



2014


professor
erstmal
startet
rundprojekt
uhr studierend forschung
leben insgesamt stiftung
studierenden neue april vortrag jahren campu
jahr menschen
team dr novemb
prof welt ab
oktob sowi wurd
iub scienc
septemb deutschen
euro universität seit
mai
studi international
intern bremer
deutsch
deutschland school
rahmen kooper
wissenschaftl wissenschaft

2013


wissenschaftwissenschaftl
intern septemb
euro menschen
startet
jahr neue mai
dr scienc school
rund
jahren studierend sowi
deutschland
team ab
welt seit
studierenden
vortrag rahmen campu
wurd novemb
april leben uhr
oktob
projekt professor prof
bremer internationaldeutschen
erstmal iub
studi deutsch
forschung
kooper insgesamt
stiftung
universität