Assignment 10 by Team

Shun-Lung Chang

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

```
colnames(text_df)
```

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

```
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.

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

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

```
head(t)
```

```
# A tibble: 6 x 4
     id year           word stemmed_word
  <int> <chr>         <chr>        <chr>
1     1 2008   wissenschaft wissenschaft
2     1 2008       jenseits      jenseit
3     1 2008        science        scienc
4     1 2008        fiction       fiction
5     1 2008         jacobs        jacob
6     1 2008     university       univers
```

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

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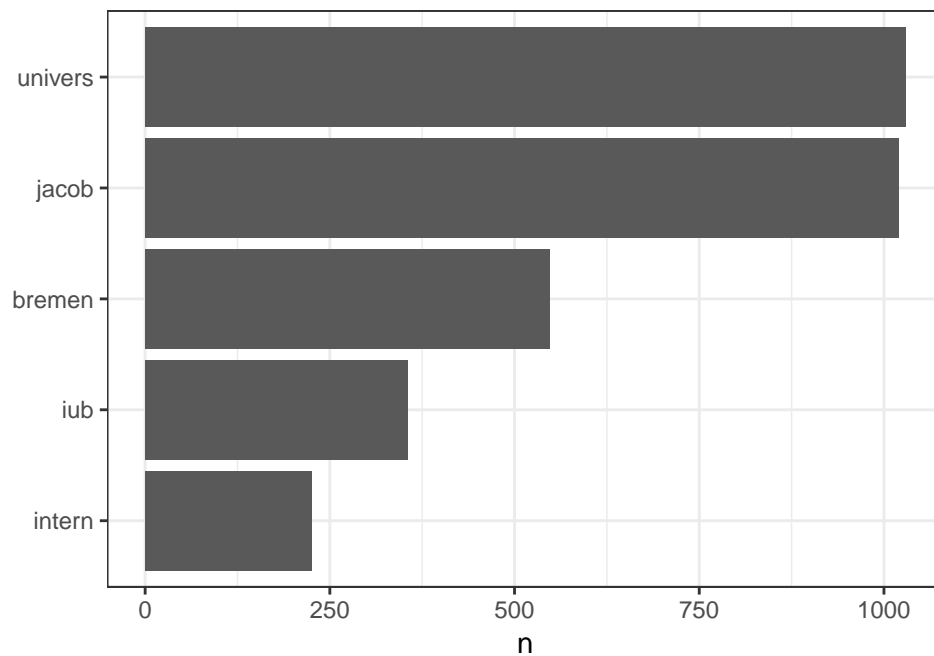


```r
word_counts <- t %>%
    group_by(id, stemmed_word) %>%
    count() %>%
    arrange(id, -n) %>%
    ungroup()

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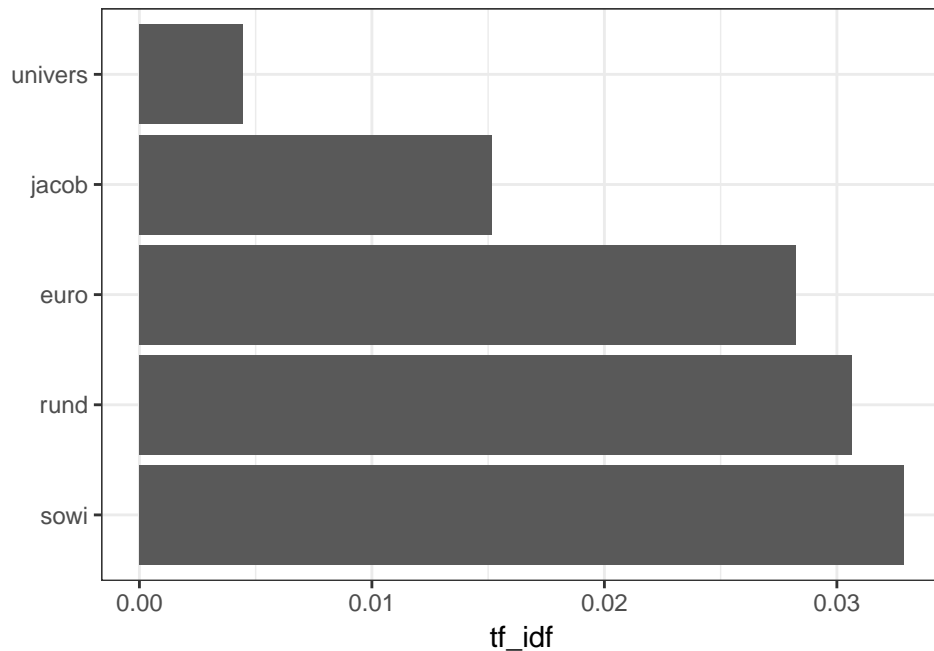
4. Compute inverse-document frequency (IDF) and term importance (TI). What are now the most common terms?

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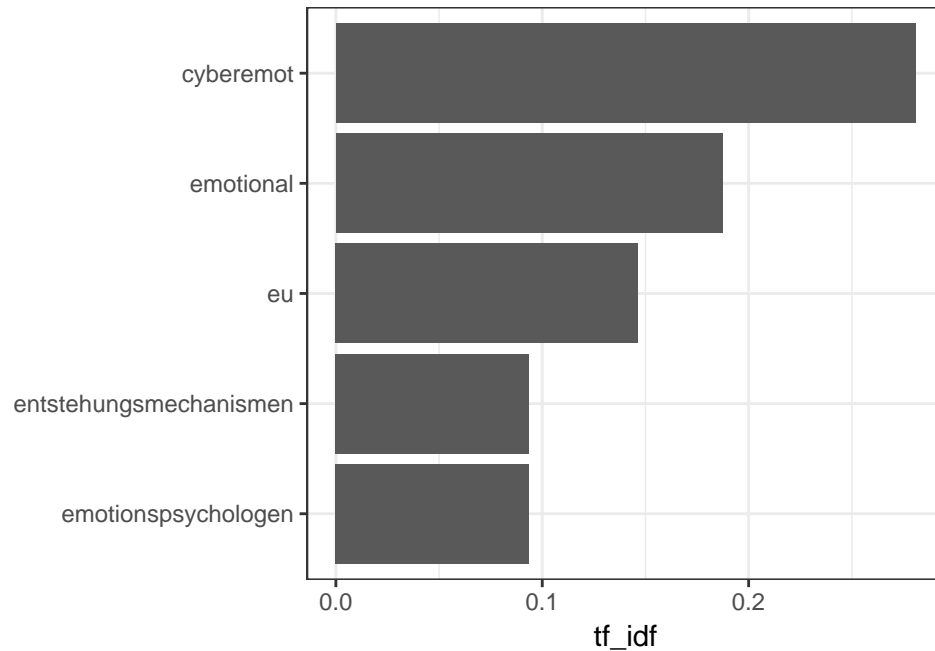


```
tf_idf %>%
    filter(id == 1) %>%
    arrange(tf_idf) %>%
    slice(1:5) %>%
    ggplot() +
    geom_col(aes(x = reorder(stemmed_word, -tf_idf), y = tf_idf)) +
    xlab(NULL) +
    coord_flip() +
    theme_bw()
```

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

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

```
cos_dist[1:3, 1:3]
```

```
            1             2             3
1 9.992007e-16  8.578515e-01  8.455115e-01
2 8.578515e-01 -4.440892e-16  7.659177e-01
3 8.455115e-01  7.659177e-01 -1.332268e-15
```

```
euc_dist[1:3, 1:3]
```

```
         [,1]     [,2]     [,3]
[1,] 0.000000 1.309848 1.300393
[2,] 1.309848 0.000000 1.237673
[3,] 1.300393 1.237673 0.000000
```

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

```
scaled_cos_dist <- scale(cos_dist)
scaled_euc_dist <- scale(euc_dist)

pair <- data_frame(cos = scaled_cos_dist[lower.tri(scaled_cos_dist)],
                   euc = scaled_euc_dist[lower.tri(scaled_euc_dist)])
```
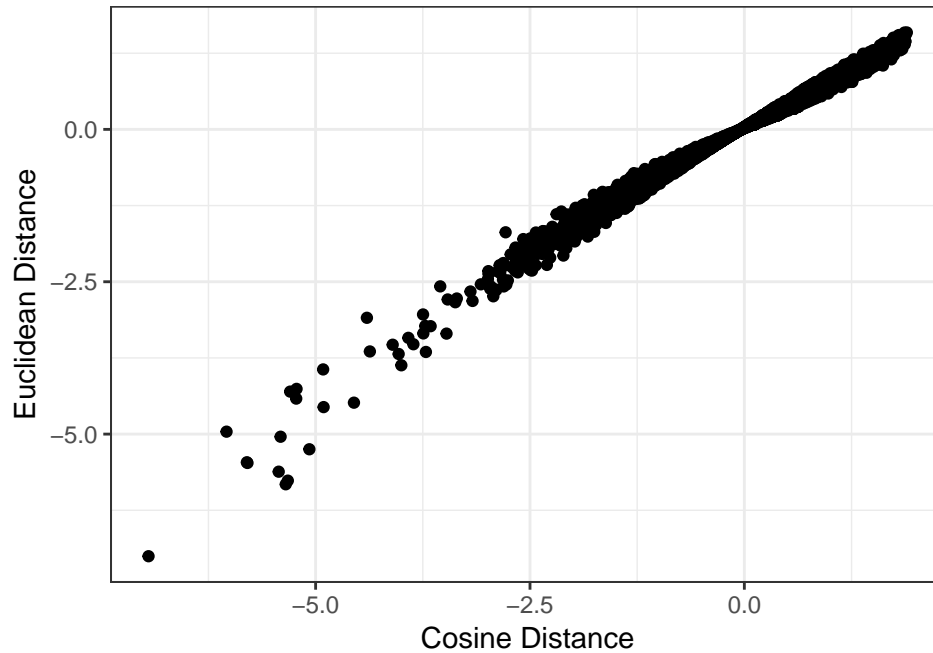
7. Capture the year of release during parsing and color code the scatterplot by time. Produce a Word Cloud for each year.
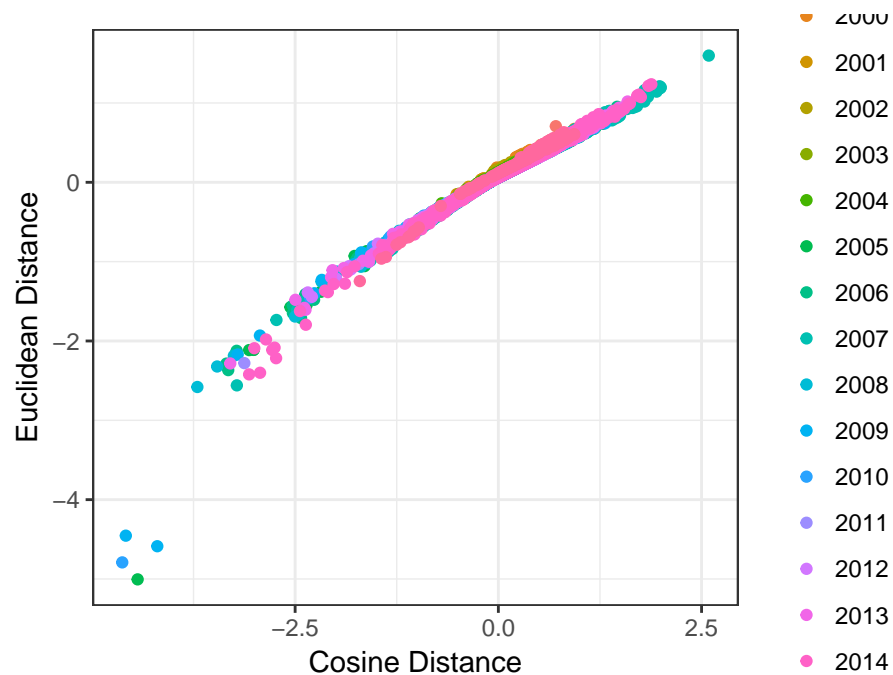

```
create_wordcloud <- function(year) {
    d <- t %>%
        filter(year == year) %>%
        group_by(stemmed_word) %>%
        count() %>%
        ungroup()
```

```
    wordcloud(words = d$stemmed_word, freq = d$n,
              max.words = 50, colors = brewer.pal(8, "Dark2"))
}
```

```
create_wordcloud(2015)
```