

Text Plots

Jan Wijffels

Abstract

The textplot R package allows one to visualise complex relations in texts. This is done by providing functionalities for displaying text co-occurrence networks, text correlation networks, dependency relationships as well as text clustering. In this vignette, some example visualisations of these are shown.

Keywords: Text, network, co-occurrence, correlation, text clustering, dependency parsing, visualisation.

1. General

1.1. Overview

The package allows you to visualise

- Text frequencies
- Text correlations
- Text cooccurrences
- Text clusters
- Text embeddings
- Dependency parsing results

Source code repository

The source code of the package is on github at <https://github.com/bnosac/textplot>.
The R package is distributed under the GPL-2 license.

2. Example visualisations

2.1. Dependency Parser

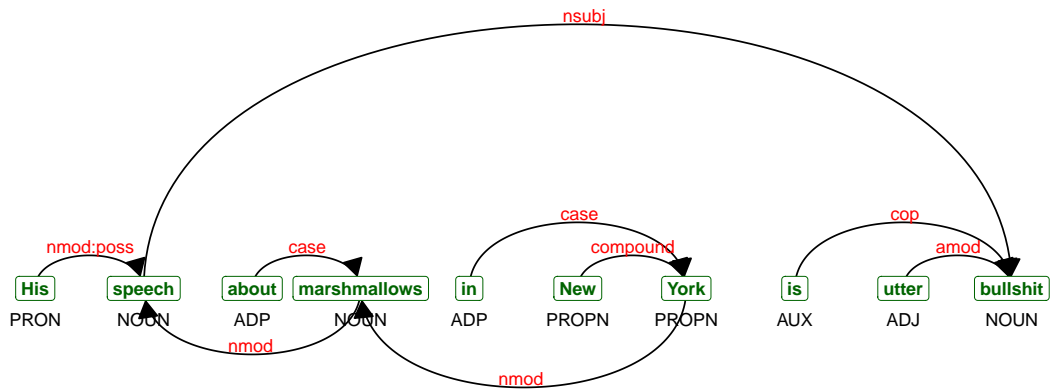
Example 1

This example visualises the result of a text annotation which provides parts of speech tags and dependency relationships.

```
library(textplot)
library(udpipe)
library(ggraph)
library(ggplot2)
library(igraph)
x <- udpipe("His speech about marshmallows in New York is utter bullshit",
            "english")
plt <- textplot_dependencyparser(x, size = 4)
plt
```

Dependency Parser

tokenisation, parts of speech tagging & dependency relations



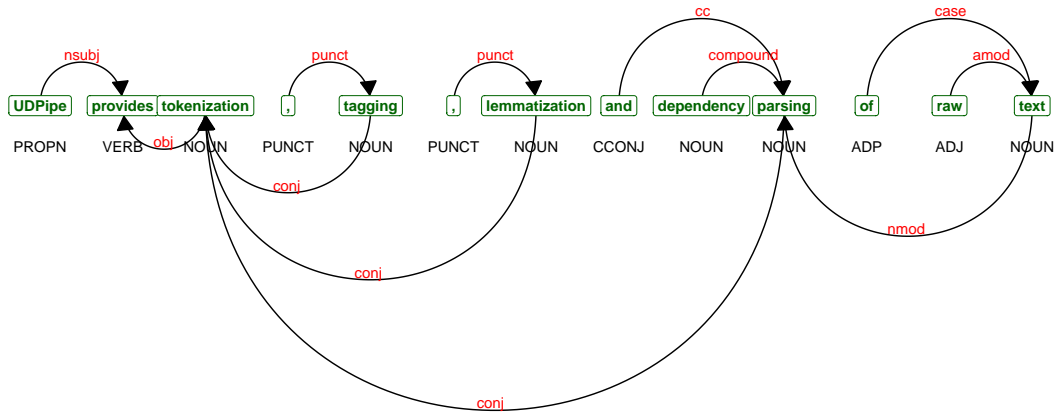
Example 2

The following visualisation displays the dependency parser results on some larger sentence. Note that this function works only on 1 sentence.

```
x <- udpipe("UDPipe provides tokenization, tagging, lemmatization and
            dependency parsing of raw text", "english")
plt <- textplot_dependencyparser(x, size = 4)
plt
```

Dependency Parser

tokenisation, parts of speech tagging & dependency relations



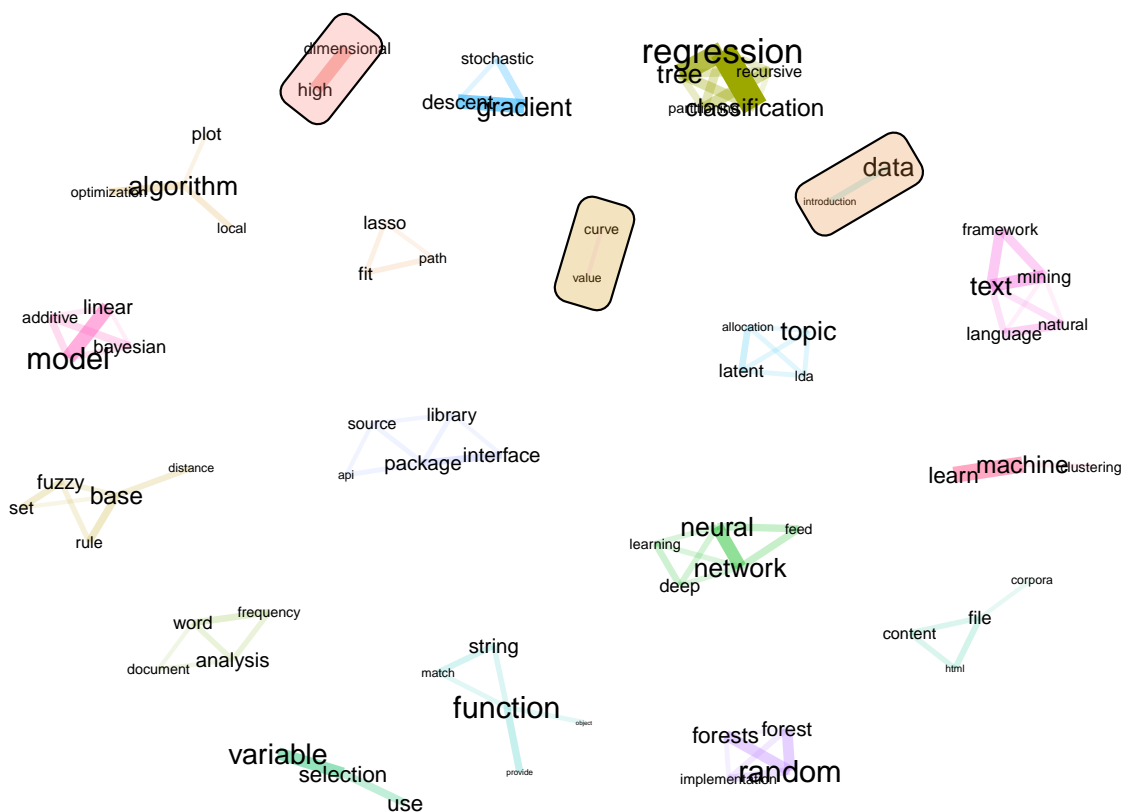
2.2. Biterm Topic Model plots

Example 1

This example shows plotting a biterm topic model which was pretrained and put in the package as an example.

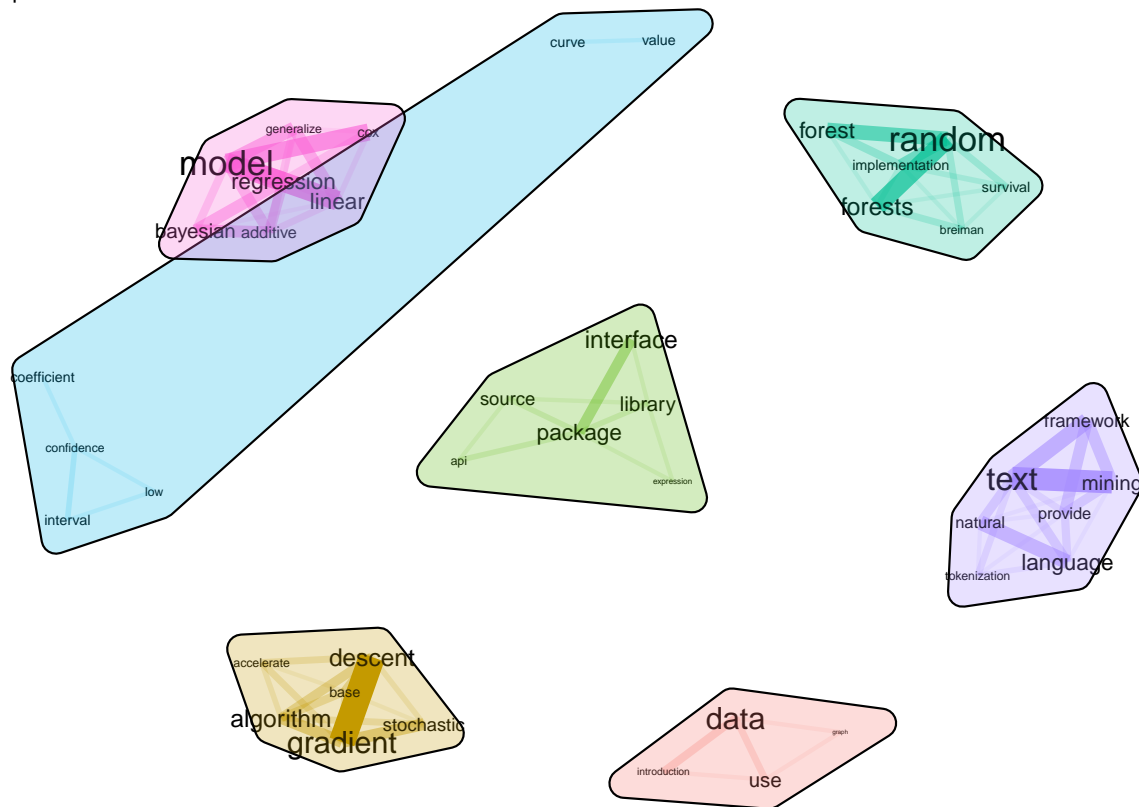
```
library(BTM)
library(ggplot2)
library(gggraph)
library(ggforce)
library(concaveman)
library(igraph)
data(example_btm, package = 'textplot')
model <- example_btm
plt <- plot(model, title = "BTM model", top_n = 5)
plt
```

BTM model



```
plt <- plot(model, title = "Biterm topic model", subtitle = "Topics 2 to 8",
            which = 2:8, top_n = 7)
plt
```

Biterm topic model
Topics 2 to 8



Example 2

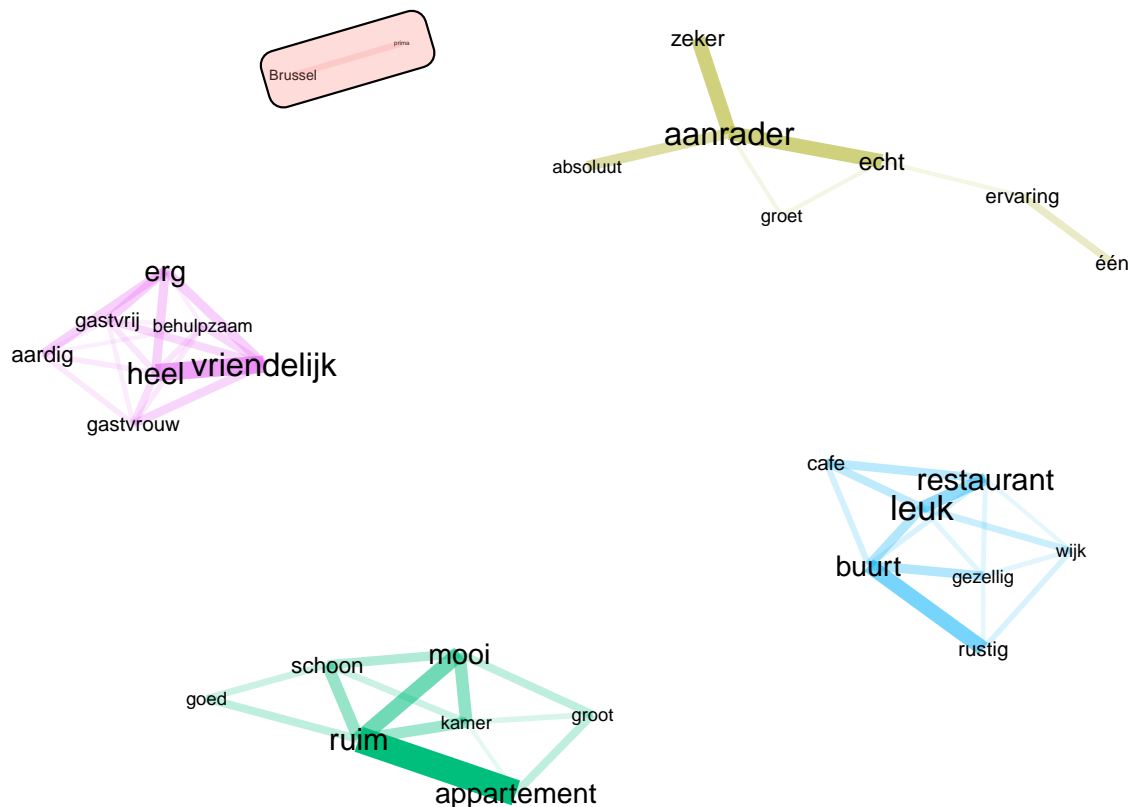
This example shows building a biterm topic model on nouns, adjectives and proper nouns occurring in the neighbourhood of one another and next plotting this model.

```
library(data.table)
library(udpipe)
## Annotate text with parts of speech tags
data("brussels_reviews", package = "udpipe")
anno <- subset(brussels_reviews, language %in% "nl")
anno <- data.frame(doc_id = anno$id, text = anno$feedback, stringsAsFactors = FALSE)
anno <- udpipe(anno, "dutch", trace = 10)
## Get cooccurrences of nouns / adjectives and proper nouns
biterms <- as.data.table(anno)
biterms <- biterms[, cooccurrence(x = lemma,
                                relevant = upos %in% c("NOUN", "PROPN", "ADJ"),
                                skipgram = 2),
```

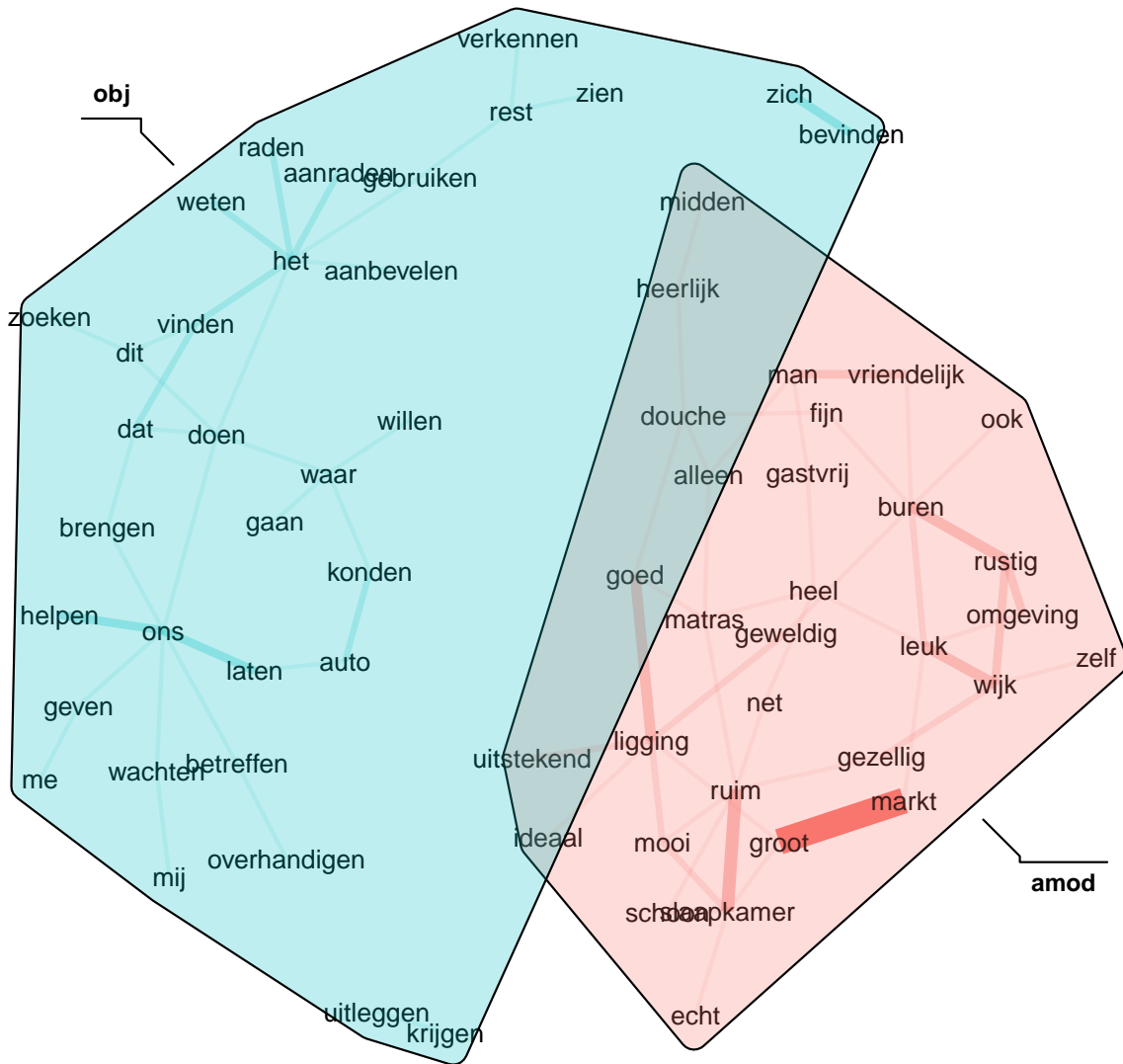
```
by = list(doc_id)]
```

```
library(BTM)
library(ggplot2)
library(ggraph)
library(ggforce)
library(concaveman)
library(igraph)
## Build the BTM model
set.seed(123456)
x <- subset(anno, upos %in% c("NOUN", "PROPN", "ADJ"))
x <- x[, c("doc_id", "lemma")]
model <- BTM(x, k = 5, beta = 0.01, iter = 2000, background = TRUE,
             biterms = biterms, trace = 100)
plt <- plot(model)
plt
```

Biterm topic model



Objects of verbs and adjectives–nouns
 Top 50 by group

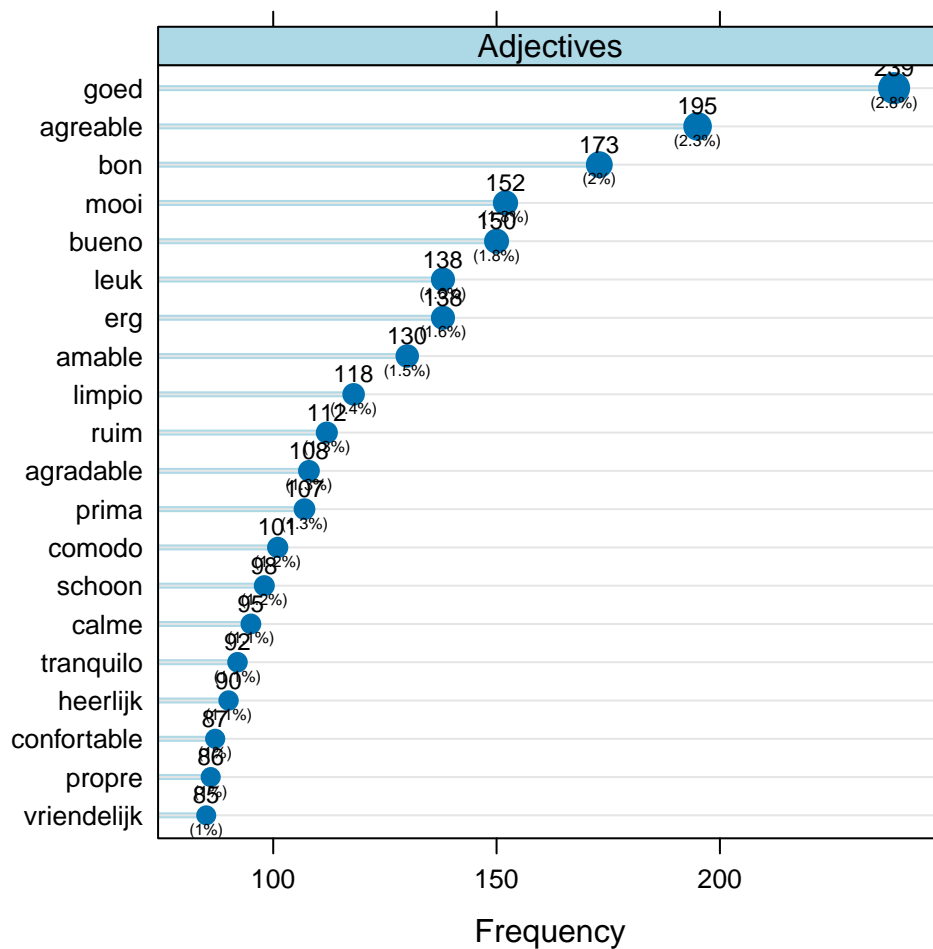


2.4. Bar plots

Example showing frequency of adjectives

The plot below shows a simple barplot which works on the output of table.

```
library(udpipe)
data("brussels_reviews_anno", package = "udpipe")
x <- subset(brussels_reviews_anno, xpos %in% "JJ")
x <- sort(table(x$lemma))
plt <- textplot_bar(x, top = 20,
                    panel = "Adjectives", xlab = "Frequency",
                    col.panel = "lightblue", cextext = 0.75,
                    addpct = TRUE, cexpct = 0.5)
plt
```



2.5. Correlation of texts

Top correlations above a certain threshold

Text correlations are interesting to see, but as there are many, the below function allows one to visualise a subset of these, the ones with the highest correlations above a certain threshold.

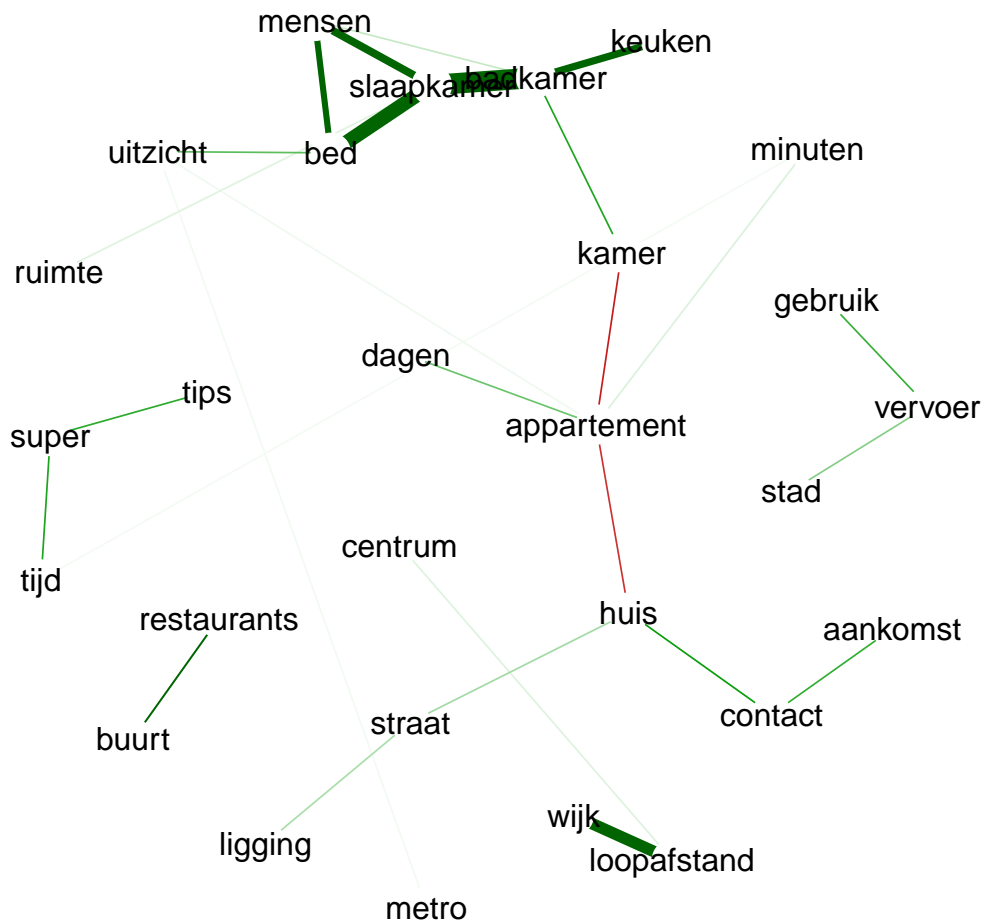
```
library(graph)
library(Rgraphviz)
library(udpipe)
dtm <- subset(anno, upos %in% "ADJ")
dtm <- document_term_frequencies(dtm, document = "doc_id", term = "lemma")
dtm <- document_term_matrix(dtm)
dtm <- dtm_remove_lowfreq(dtm, minfreq = 5)
textplot_correlation_lines(dtm, top_n = 25, threshold = 0.01, lwd = 5, label = TRUE)
```



Correlations which are non-zero after fitting a glasso model

If you have text correlations, you can also fit a glasso model on it. This puts non-relevant correlations to zero, allowing one to plot the correlations in a straightforward way.

```
library(glasso)
library(qgraph)
library(udpipe)
dtm <- subset(anno, upos %in% "NOUN")
dtm <- document_term_frequencies(dtm, document = "doc_id", term = "token")
dtm <- document_term_matrix(dtm)
dtm <- dtm_remove_lowfreq(dtm, minfreq = 20)
dtm <- dtm_remove_tfidf(dtm, top = 100)
term_correlations <- dtm_cor(dtm)
textplot_correlation_glasso(term_correlations, exclude_zero = TRUE)
```



2.6. Co-occurrence of texts

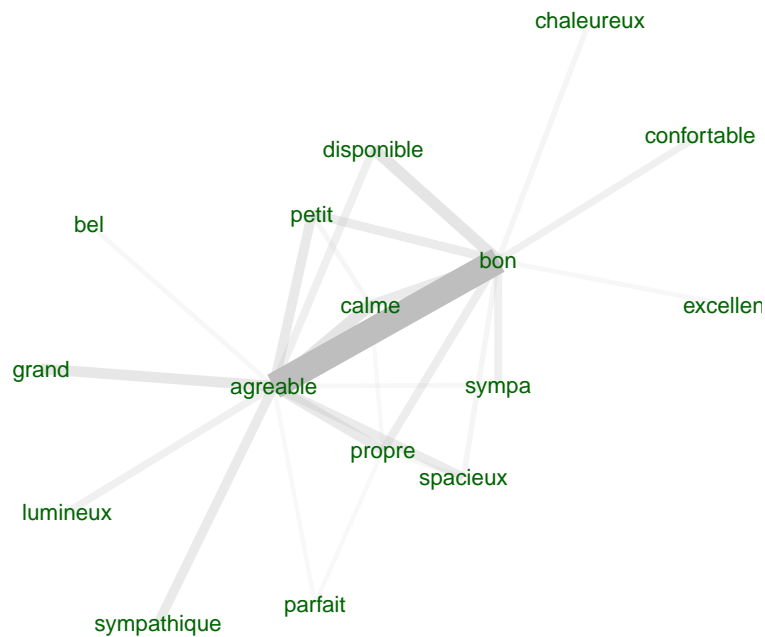
Example showing adjectives occurring in the same document

The following graph shows how frequently adjectives co-occur across all the documents.

```
library(udpipe)
library(igraph)
library(ggraph)
library(ggplot2)
data(brussels_reviews_anno, package = 'udpipe')
x <- subset(brussels_reviews_anno, xpos %in% "JJ" & language %in% "fr")
x <- cooccurrence(x, group = "doc_id", term = "lemma")

plt <- textplot_cooccurrence(x,
                             title = "Adjective co-occurrences", top_n = 25)
plt
```

Adjective co-occurrences



Example showing objects of verbs / adjectives modifying nouns on our annotated dataset

The following graph shows a similar visualisation, but instead focussing on the frequency of objects of verbs and adjectives modifying a noun. For this, we start again from the annotation of the AirBnB data shown in the section 2.2.2.

```
library(udpipe)
library(igraph)
library(ggraph)
library(ggplot2)
library(data.table)
bitterms <- merge(anno, anno,
                  by.x = c("doc_id", "paragraph_id", "sentence_id", "head_token_id"),
                  by.y = c("doc_id", "paragraph_id", "sentence_id", "token_id"),
                  all.x = TRUE, all.y = FALSE, suffixes = c("", "_parent"), sort = FALSE)
bitterms <- setDT(bitterms)
bitterms <- subset(bitterms, dep_rel %in% c("obj", "amod"))
bitterms <- bitterms[, list(cooc = .N), by = list(term1 = lemma, term2 = lemma_parent)]
plt <- textplot_cooccurrence(bitterms,
                              title = "Objects of verbs + Adjectives-nouns",
                              top_n = 75,
                              vertex_color = "orange", edge_color = "black",
                              fontface = "bold")
plt
```

Objects of verbs + Adjectives-nouns



2.7. Text embeddings

Example showing clustered text embeddings

The following graph shows the embeddings of the top 7 words emitted by a sample of topics extracted with the Embedding Topic Modelling clustering algorithm (<https://github.com/bnosac/ETM>).

The embeddings are mapped onto a 2-dimensional space using UMAP.

```
library(uwot)
set.seed(1234)

## Put embeddings in lower-dimensional space (2D)
data(example_embedding, package = "textplot")
embed.2d <- umap(example_embedding,
                 n_components = 2, metric = "cosine", n_neighbors = 15,
                 fast_sgd = TRUE, n_threads = 2, verbose = FALSE)
embed.2d <- data.frame(term = rownames(example_embedding),
                      x = embed.2d[, 1], y = embed.2d[, 2],
                      stringsAsFactors = FALSE)
head(embed.2d, n = 5)

##           term           x           y
## tribunal      tribunal  3.09024725 -0.60324491
## noodnummers  noodnummers -1.11698290 -0.14421559
## acs           acs       0.94337747  0.01695859
## spi          spi      -2.17797866  2.71778428
## alert        alert    -0.08664509  0.28230094

## Get a dataset with words assigned to each cluster with a certain probability weight
data(example_embedding_clusters, package = "textplot")
terminology <- merge(example_embedding_clusters, embed.2d, by = "term", sort = FALSE)
terminology <- subset(terminology, rank <= 7 & cluster %in% c(1, 3, 4, 10, 15, 19, 17))
head(terminology, n = 10)

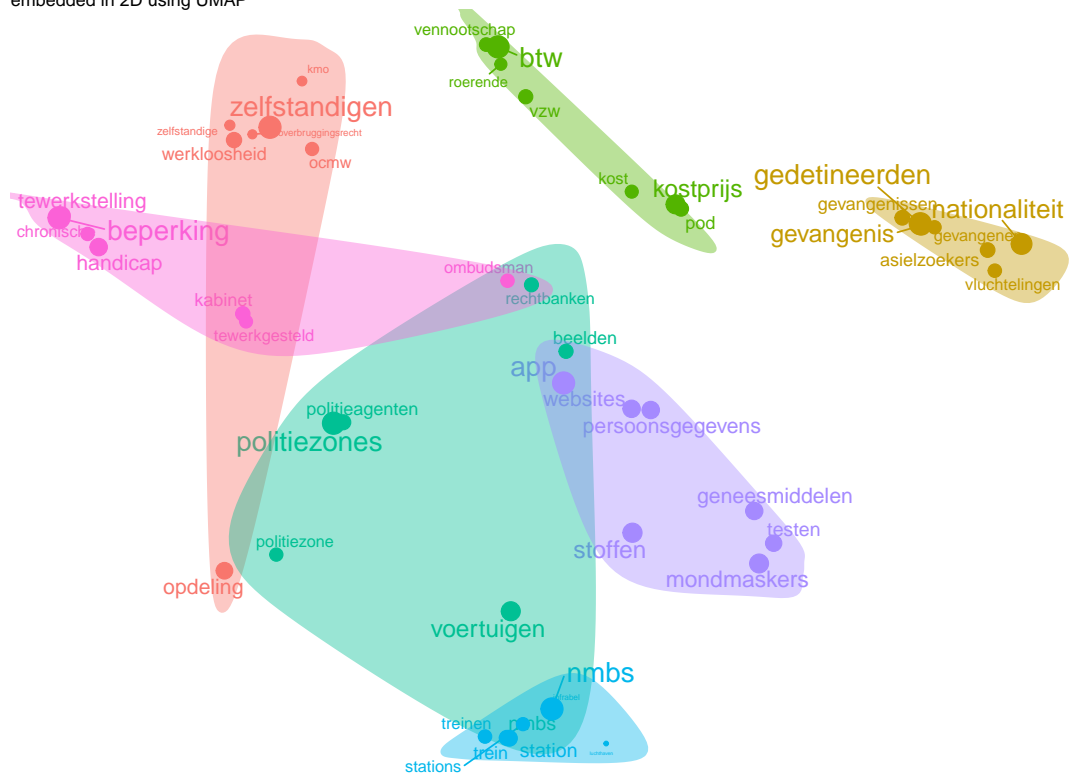
##           term cluster rank  weight           x           y
## 1      zelfstandigen      1     1 1.0000000 -2.0814679  2.544579
## 5           opdeling      1     2 0.5390060 -2.3868655 -1.830962
## 13      werkloosheid      1     3 0.4511878 -2.3225572  2.418448
## 16           ocmw         1     4 0.3379358 -1.7990121  2.330511
## 19      zelfstandige      1     5 0.2172686 -2.3506609  2.564672
## 21           kmo         1     6 0.2013531 -1.8665626  3.000427
## 23  overbruggingsrecht      1     7 0.1851361 -2.1993925  2.474997
## 54           vzw         4     4 0.3867166 -0.3663370  2.845928
## 68           pod         4     3 0.4328151  0.6751950  1.741148
## 211          btw         4     1 1.0000000 -0.5520044  3.337710
```

```

## Plot the relevant embeddings
library(ggplot2)
library(ggrepel)
library(ggalt)
plt <- textplot_embedding_2d(terminology, encircle = TRUE, points = TRUE,
                             title = "Embedding Topic Model clusters",
                             subtitle = "embedded in 2D using UMAP")
plt

```

Embedding Topic Model clusters
embedded in 2D using UMAP



Affiliation:

BNOSAC - Open Analytical Helpers

E-mail: jwijffels@bnosac.be

URL: <http://www.bnosac.be>