WE

Miras Tolepbergen

2023-11-29

```r
install.packages("Rtsne", repos='http://cran.us.r-project.org')

install.packages("text2vec", repos='http://cran.us.r-project.org')

install.packages("plotly", repos='http://cran.us.r-project.org')

install.packages("umap", repos='http://cran.us.r-project.org')

devtools::install_github("oscarkjell/text")

rm(list=ls(all=TRUE))
setwd("C:/Users/Miras/Desktop/u_m/1st/big_data_analytics/Labs/projects")
getwd()

## [1] "C:/Users/Miras/Desktop/u_m/1st/big_data_analytics/Labs/projects"

library(quanteda)

library(readtext)

library(text2vec)

library(quanteda.textplots)
library(Rtsne)

library(ggplot2)

library(plotly)

library(umap)

library(dplyr)

library(ranger)

library(caret)

library(cvTools)

library(lsa)

library(LSAfun)

tot <- read.csv("clothing_reviews23.csv")
tot$text <- gsub("'"," ",tot$text)
```

```
myCorpus <- corpus(tot)

tok2 <- tokens(myCorpus , remove_punct = TRUE, remove_numbers=TRUE,
remove_symbols = TRUE, split_hyphens = TRUE, remove_separators = TRUE)
tok2 <- tokens_remove(tok2, stopwords("en"))

Dfm <- dfm(tok2 )
Dfm <- dfm_remove(Dfm , min_nchar=2)
topfeatures(Dfm )

##  dress   love   size    top    fit  great   like   wear   just fabric
##   2600   2299   1900   1805   1750   1709   1553   1434   1254   1175

Dfm <- dfm_trim(Dfm,  min_termfreq = 5, verbose=TRUE)

## Removing features occurring:

##   - fewer than 5 times: 4,987

##   Total features removed: 4,987 (68.2%).

# Applying the GloVe algorithm via Quanteda

# Let's first extract the vocabulary from our Dfm
Dfm_vocab <- featnames(Dfm )
str(Dfm_vocab)

##  chr [1:2328] "absolutely" "wonderful" "silky" "sexy" "comfortable" "love"
...

# Then let's select the tokens that are present in our previously defined
corpus
mov_tokens <- tokens(myCorpus)
mov_tokens

## Tokens consisting of 5,000 documents and 2 docvars.
## text1 :
## [1] "Absolutely"  "wonderful"   "-"           "silky"       "and"
## [6] "sexy"        "and"         "comfortable"
##
## text2 :
##  [1] "Love"     "this"     "dress"    "!"        "it"       "s"
##  [7] "sooo"     "pretty"   "."        "i"        "happened" "to"
## [ ... and 62 more ]
##
## text3 :
##  [1] "Some"   "major"  "design" "flaws"  "I"      "had"    "such"   "high"
##  [9] "hopes"  "for"    "this"   "dress"
## [ ... and 103 more ]
##
## text4 :
##  [1] "My"       "favorite" "buy"      "!"        "I"        "love"
##  [7] ","        "love"     ","        "love"     "this"     "jumpsuit"
## [ ... and 23 more ]
##
## text5 :
##  [1] "Flattering" "shirt"      "This"       "shirt"      "is"
##  [6] "very"       "flattering" "to"         "all"        "due"
## [11] "to"         "the"
## [ ... and 31 more ]
##
## text6 :
##  [1] "Not"     "for"     "the"     "very"    "petite"  "I"       "love"
##  [8] "tracy"   "reese"   "dresses" ","       "but"
## [ ... and 100 more ]
##
## [ reached max_ndoc ... 4,994 more documents ]
```

```
mov_tokens2 <- tokens_select(mov_tokens, Dfm_vocab, padding = TRUE)
```

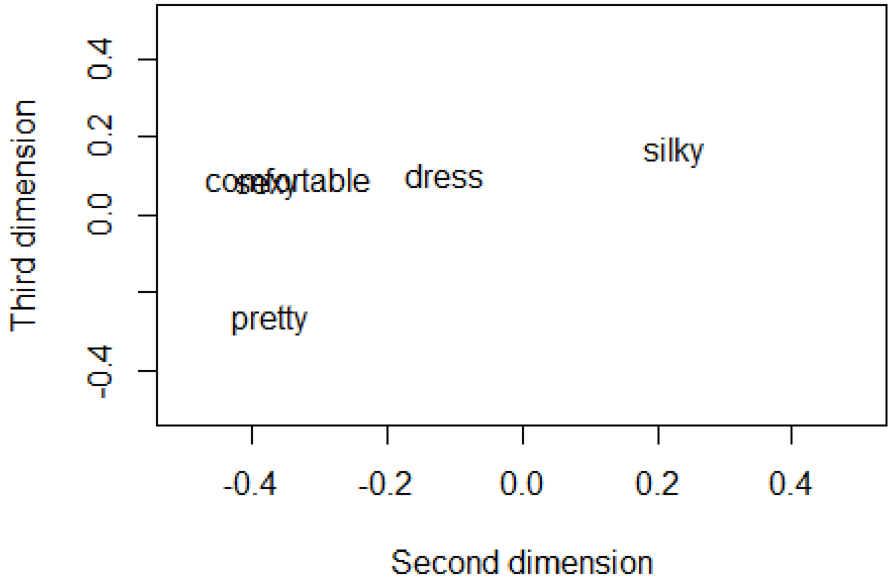
```
fcmat_news <- fcm(mov_tokens2, context = "window", count = "weighted",
weights = 1/(1:5))
fcmat_news
```

```
## Feature co-occurrence matrix of: 2,878 by 2,878 features.
##               features
## features      Absolutely wonderful     silky sexy comfortable     Love
##   Absolutely   0.6666667 3.7500000 0.3333333 0.20   0.3333333 0.500000
##   wonderful    0         0.6666667 0.5000000 0.25   0.6500000 0
##   silky        0         0         0         0.50   0.2500000 0
##   sexy         0         0         0         0.50   2.1666667 0
##   comfortable  0         0         0         0      1.3000000 2.600000
##   Love         0         0         0         0      0         9.333333
##   dress        0         0         0         0      0         0
##   sooo         0         0         0         0      0         0
##   pretty       0         0         0         0      0         0
##   happened     0         0         0         0      0         0
##               features
## features          dress sooo     pretty  happened
##   Absolutely   1.200000 0     0         0
##   wonderful    3.416667 0     0         0
##   silky        0        0     0         0
##   sexy         1.666667 0     0.3333333 0
##   comfortable 35.666667 1.00  2.1666667 0
##   Love        44.466667 0     0         0
##   dress       87.133333 0.25 22.4500000 0.6666667
##   sooo         0        0     1.0000000 0.2500000
##   pretty       0        0     3.8000000 0.5833333
##   happened     0        0     0         0
## [ reached max_feat ... 2,868 more features, reached max_nfeat ... 2,868
more features ]
```

```
# Let's estimate WE via Glove

glove <- GlobalVectors$new(rank=100, x_max=10)

set.seed(123)
system.time(glove_main <- glove$fit_transform(fcmat_news, n_iter = 50,
convergence_tol = 0.01, n_threads = 1))

## INFO  [19:54:04.978] epoch 1, loss 0.2223
##    user  system elapsed
##   24.11    0.06   25.78

str(glove_main)

##  num [1:2878, 1:100] -0.436 0.136 -0.191 -0.176 0.492 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:2878] "Absolutely" "wonderful" "silky" "sexy" ...
##   ..$ : NULL

# Plotting words in the WE dimensional space

# Let's create a dataframe out of the Glove results
glove_dataframe <- as.data.frame(glove_main)
nrow(glove_dataframe)

## [1] 2878

# the same # of words as in our co-occurance matrix of course!
nrow(fcmat_news)

## [1] 2878

colnames(glove_dataframe )

##   [1] "V1"   "V2"   "V3"   "V4"   "V5"   "V6"   "V7"   "V8"   "V9"   "V10"
##  [11] "V11"  "V12"  "V13"  "V14"  "V15"  "V16"  "V17"  "V18"  "V19"  "V20"
##  [21] "V21"  "V22"  "V23"  "V24"  "V25"  "V26"  "V27"  "V28"  "V29"  "V30"
##  [31] "V31"  "V32"  "V33"  "V34"  "V35"  "V36"  "V37"  "V38"  "V39"  "V40"
##  [41] "V41"  "V42"  "V43"  "V44"  "V45"  "V46"  "V47"  "V48"  "V49"  "V50"
##  [51] "V51"  "V52"  "V53"  "V54"  "V55"  "V56"  "V57"  "V58"  "V59"  "V60"
##  [61] "V61"  "V62"  "V63"  "V64"  "V65"  "V66"  "V67"  "V68"  "V69"  "V70"
##  [71] "V71"  "V72"  "V73"  "V74"  "V75"  "V76"  "V77"  "V78"  "V79"  "V80"
##  [81] "V81"  "V82"  "V83"  "V84"  "V85"  "V86"  "V87"  "V88"  "V89"  "V90"
##  [91] "V91"  "V92"  "V93"  "V94"  "V95"  "V96"  "V97"  "V98"  "V99"
"V100"

# let's add to glove_dataframe a specific column called "word" with the list
of features
glove_dataframe$word <- row.names(glove_dataframe )
colnames(glove_dataframe )
```

```
##   [1] "V1"   "V2"   "V3"   "V4"   "V5"   "V6"   "V7"   "V8"   "V9"   "V10"
##  [11] "V11"  "V12"  "V13"  "V14"  "V15"  "V16"  "V17"  "V18"  "V19"  "V20"
##  [21] "V21"  "V22"  "V23"  "V24"  "V25"  "V26"  "V27"  "V28"  "V29"  "V30"
##  [31] "V31"  "V32"  "V33"  "V34"  "V35"  "V36"  "V37"  "V38"  "V39"  "V40"
##  [41] "V41"  "V42"  "V43"  "V44"  "V45"  "V46"  "V47"  "V48"  "V49"  "V50"
##  [51] "V51"  "V52"  "V53"  "V54"  "V55"  "V56"  "V57"  "V58"  "V59"  "V60"
##  [61] "V61"  "V62"  "V63"  "V64"  "V65"  "V66"  "V67"  "V68"  "V69"  "V70"
##  [71] "V71"  "V72"  "V73"  "V74"  "V75"  "V76"  "V77"  "V78"  "V79"  "V80"
##  [81] "V81"  "V82"  "V83"  "V84"  "V85"  "V86"  "V87"  "V88"  "V89"  "V90"
##  [91] "V91"  "V92"  "V93"  "V94"  "V95"  "V96"  "V97"  "V98"  "V99"
"V100"
## [101] "word"
```

```
# let's define a plot function for the second and third dimension for example


plot_words <- function(words, glove_dataframe){
  # empty plot
  plot(0, 0, xlim=c(-0.5, 0.5), ylim=c(-0.5,0.5), type="n",
       xlab="Second dimension", ylab="Third dimension")
  for (word in words){
    # extract second and third dimensions
    vector <- as.numeric(glove_dataframe[glove_dataframe$word==word,2:3])
    # add to plot
    text(vector[1], vector[2], labels=word)
  }
}

plot_words(c("dress", "sexy", "silky", "comfortable", "love", "pretty"),
glove_dataframe)
```



```
set.seed(123)
system.time(tsne <-  Rtsne(glove_main[1:500,], perplexity = 50))

##    user  system elapsed 
##    3.00    0.10    3.27

str(tsne)

## List of 14
##  $ N                  : int 500
##  $ Y                  : num [1:500, 1:2] 4.22 2.11 -2.93 -1.14 -2.18 ...
##  $ costs              : num [1:500] 0.00279 0.00293 0.00292 0.00953 0.002
...
##  $ itercosts          : num [1:20] 58.1 58.6 58.9 58.8 59.6 ...
##  $ origD              : int 50
##  $ perplexity         : num 50
##  $ theta              : num 0.5
##  $ max_iter           : num 1000
##  $ stop_lying_iter    : int 250
##  $ mom_switch_iter    : int 250
##  $ momentum           : num 0.5
##  $ final_momentum     : num 0.8
##  $ eta                : num 200
##  $ exaggeration_factor: num 12
##  - attr(*, "class")= chr [1:2] "Rtsne" "list"
```

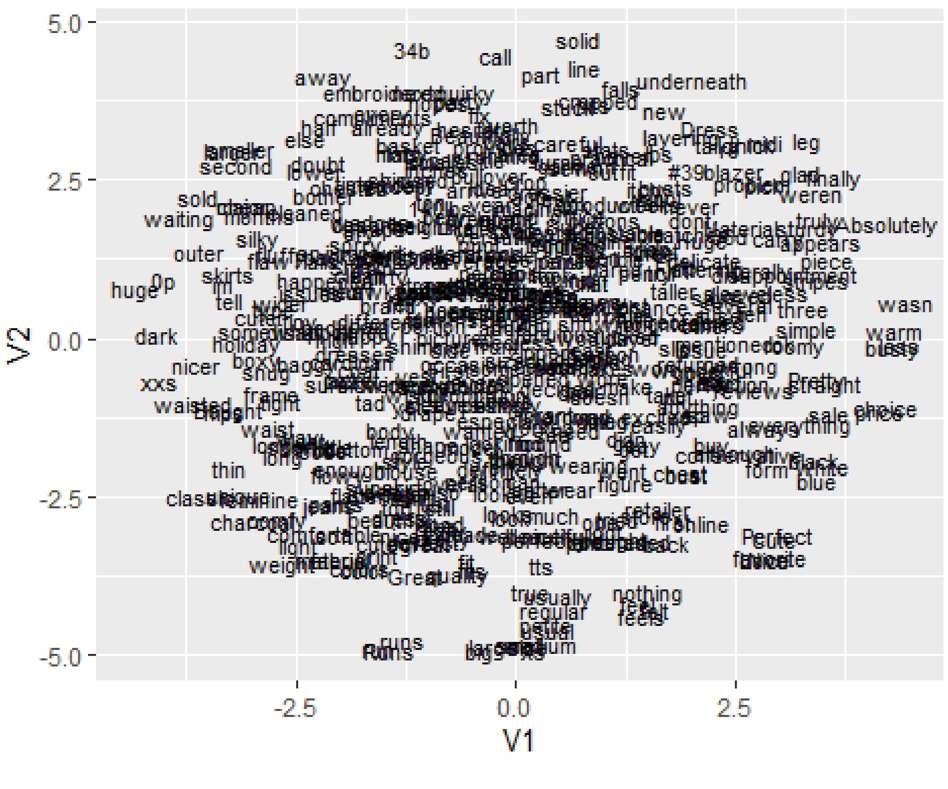
```
tsne_plot <- tsne$Y
tsne_plot  <- as.data.frame(tsne_plot)
str(tsne_plot)
```

```
## 'data.frame':    500 obs. of  2 variables:
##  $ V1: num  4.22 2.11 -2.93 -1.14 -2.18 ...
##  $ V2: num  1.82 -0.482 1.605 0.684 -3.06 ...
```

```
tsne_plot$word  <- row.names(glove_main)[1:500]
str(tsne_plot)
```

```
## 'data.frame':    500 obs. of  3 variables:
##  $ V1  : num  4.22 2.11 -2.93 -1.14 -2.18 ...
##  $ V2  : num  1.82 -0.482 1.605 0.684 -3.06 ...
##  $ word: chr  "Absolutely" "wonderful" "silky" "sexy" ...
```

```
tsne_plot2 <- ggplot(tsne_plot, aes(x = V1, y = V2, label = word)) +
geom_text(size = 3)
tsne_plot2
```

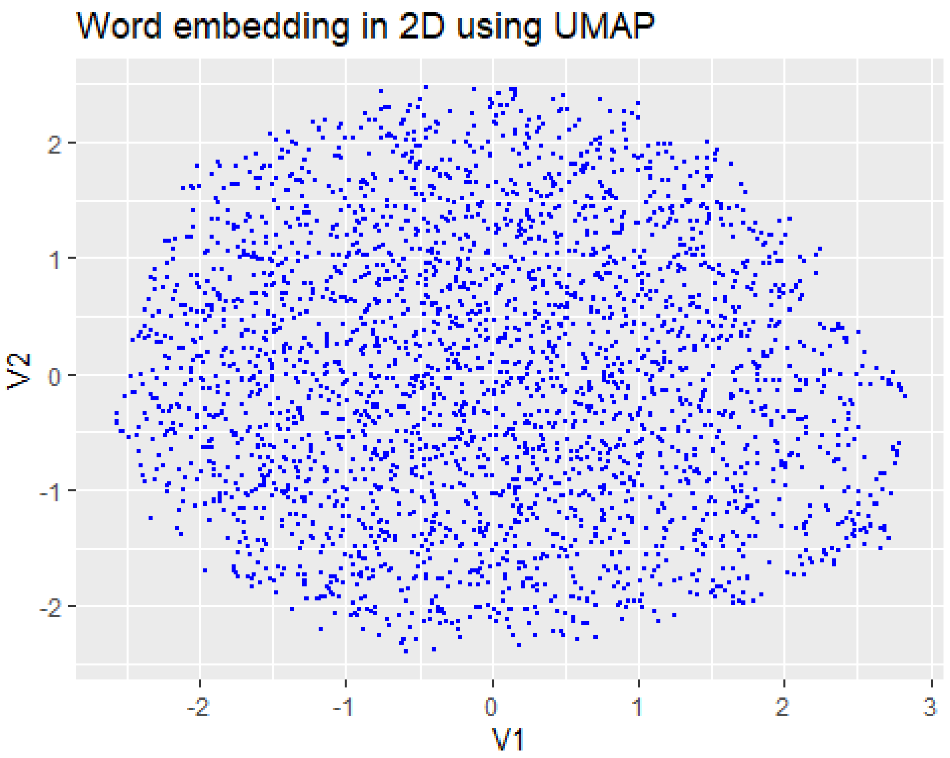


```
tsne_plot[which(tsne_plot$word=="xxs"),]
```

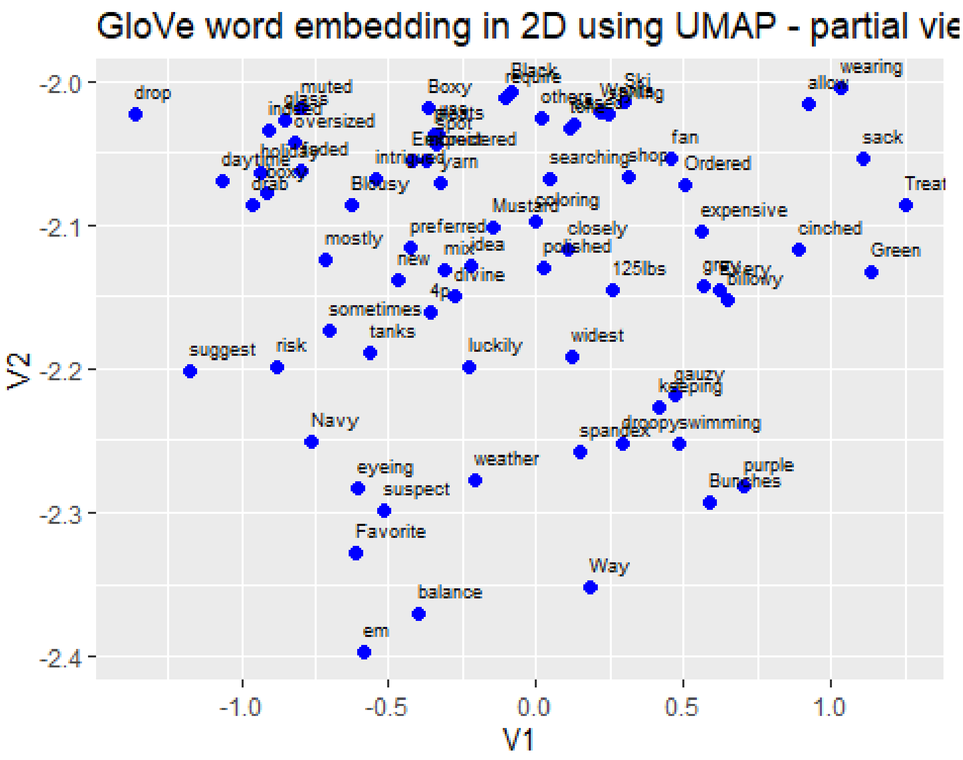
```
##            V1         V2 word
## 162 -4.056662 -0.6536709  xxs
```

```
tsne_plot[which(tsne_plot$word=="figure"),]
```

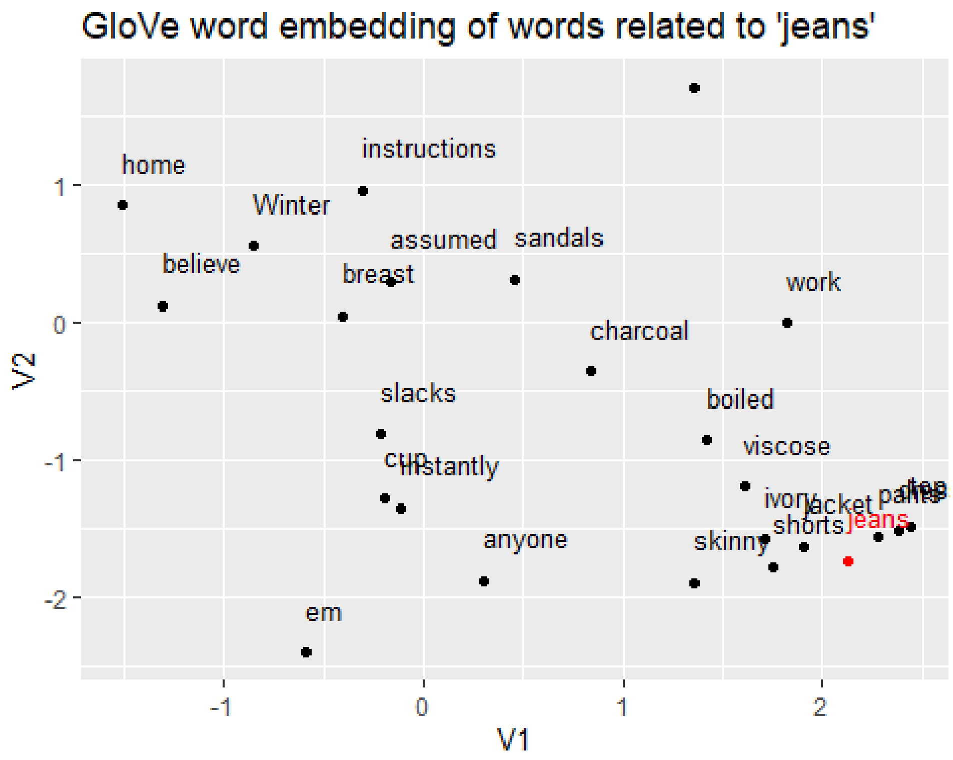
```
##           V1        V2   word
## 275 1.229231 -2.273313 figure

# let's transform the ggplot into an interacting plotly plot
#ggplotly(tsne_plot2)

set.seed(123)
system.time(glove_umap <- umap(glove_main, n_components = 2, metric =
"cosine", n_neighbors = 20, min_dist = 0.1))

##    user  system elapsed
##   47.64    0.55   52.25

saveRDS(glove_umap, file = "glove_umap.rds")

glove_umap <- readRDS("glove_umap.rds")

glove_umap

## umap embedding of 2878 items in 2 dimensions
## object components: layout, data, knn, config

head(glove_umap$layout, 3)

##                  [,1]       [,2]
## Absolutely 0.08238431 -1.4322908
## wonderful  1.16689892 -0.9827055
## silky      0.93713674 -0.1143543

str(glove_umap$layout)

##  num [1:2878, 1:2] 0.0824 1.1669 0.9371 -0.4119 2.4856 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:2878] "Absolutely" "wonderful" "silky" "sexy" ...
##   ..$ : NULL

df_glove_umap <- as.data.frame(glove_umap$layout)
str(df_glove_umap)

## 'data.frame':    2878 obs. of  2 variables:
##  $ V1: num  0.0824 1.1669 0.9371 -0.4119 2.4856 ...
##  $ V2: num  -1.4323 -0.9827 -0.1144 -0.0452 -1.2945 ...

df_glove_umap$word <- row.names(df_glove_umap)

ggplot(df_glove_umap) +
      geom_point(aes(x = V1, y = V2), colour = 'blue', size = 0.05) +
labs(title = "Word embedding in 2D using UMAP")
```



```r
# Plot the bottom part of the GloVe word embedding with labels
ggplot(df_glove_umap[df_glove_umap$V1 > -2.0 & df_glove_umap$V1 < 3 &
df_glove_umap$V2 < -2,]) +
      geom_point(aes(x = V1, y = V2), colour = 'blue', size = 2) +
      geom_text(aes(V1, V2, label =word), size = 2.5, vjust=-1, hjust=0) +
      labs(title = "GloVe word embedding in 2D using UMAP - partial view")
```



```
jeans<- glove_main["jeans", , drop = F] # Let's see what is similar to
"jeans"
cos_sim_great <- sim2(x = glove_main, y = jeans, method = "cosine", norm =
"l2")
head(sort(cos_sim_great[,1], decreasing = T), 10) #most of the similarities
make sense

##     jeans     pants    shorts    breast     dress       Sad   viscose
assumed
## 1.0000000 0.4750380 0.3239235 0.2814735 0.2809963 0.2768964 0.2751924
0.2691859
##    skinny    slacks
## 0.2652978 0.2649339

# let's see the results in our UMAP graph
select <- data.frame(rownames(as.data.frame(head(sort(cos_sim_great[,1],
decreasing = TRUE), 25))))
colnames(select) <- "word"
selected_words <- inner_join(x= df_glove_umap , y=select, by= "word")

ggplot(selected_words, aes(x = V1, y = V2, colour = word == 'jeans')) +
      geom_point(show.legend = FALSE) +
      scale_color_manual(values = c('black', 'red')) +
      geom_text(aes(V1, V2, label = word), show.legend = FALSE, size = 3.5,
vjust=-1.5, hjust=0) +
      labs(title = "GloVe word embedding of words related to 'jeans'")
```



```
# Once we have the vectors for each word, we can also compute the similarity
between a pair of words:

similarity <- function(word1, word2){
    lsa::cosine(
        x=as.numeric(glove_dataframe[glove_dataframe$word==word1,1:100]),
        y=as.numeric(glove_dataframe[glove_dataframe$word==word2,1:100]))
}

similarity("jacket", "jeans")

##          [,1]
## [1,] 0.253448

similarity("home", "jeans")

##           [,1]
## [1,] 0.2330053

similarity("pants", "jeans")

##          [,1]
## [1,] 0.475038

# Machine Learning classification with WE

colnames(glove_dataframe )
```

```
##   [1] "V1"   "V2"   "V3"   "V4"   "V5"   "V6"   "V7"   "V8"   "V9"   "V10"
##  [11] "V11"  "V12"  "V13"  "V14"  "V15"  "V16"  "V17"  "V18"  "V19"  "V20"
##  [21] "V21"  "V22"  "V23"  "V24"  "V25"  "V26"  "V27"  "V28"  "V29"  "V30"
##  [31] "V31"  "V32"  "V33"  "V34"  "V35"  "V36"  "V37"  "V38"  "V39"  "V40"
##  [41] "V41"  "V42"  "V43"  "V44"  "V45"  "V46"  "V47"  "V48"  "V49"  "V50"
##  [51] "V51"  "V52"  "V53"  "V54"  "V55"  "V56"  "V57"  "V58"  "V59"  "V60"
##  [61] "V61"  "V62"  "V63"  "V64"  "V65"  "V66"  "V67"  "V68"  "V69"  "V70"
##  [71] "V71"  "V72"  "V73"  "V74"  "V75"  "V76"  "V77"  "V78"  "V79"  "V80"
##  [81] "V81"  "V82"  "V83"  "V84"  "V85"  "V86"  "V87"  "V88"  "V89"  "V90"
##  [91] "V91"  "V92"  "V93"  "V94"  "V95"  "V96"  "V97"  "V98"  "V99"  
"V100"
## [101] "word"
```

```
glove_dataframe <- select(glove_dataframe, word, everything()) # let's move 
the "word" column to the top
colnames(glove_dataframe )
```

```
##   [1] "word" "V1"   "V2"   "V3"   "V4"   "V5"   "V6"   "V7"   "V8"   "V9"
##  [11] "V10"  "V11"  "V12"  "V13"  "V14"  "V15"  "V16"  "V17"  "V18"  "V19"
##  [21] "V20"  "V21"  "V22"  "V23"  "V24"  "V25"  "V26"  "V27"  "V28"  "V29"
##  [31] "V30"  "V31"  "V32"  "V33"  "V34"  "V35"  "V36"  "V37"  "V38"  "V39"
##  [41] "V40"  "V41"  "V42"  "V43"  "V44"  "V45"  "V46"  "V47"  "V48"  "V49"
##  [51] "V50"  "V51"  "V52"  "V53"  "V54"  "V55"  "V56"  "V57"  "V58"  "V59"
##  [61] "V60"  "V61"  "V62"  "V63"  "V64"  "V65"  "V66"  "V67"  "V68"  "V69"
##  [71] "V70"  "V71"  "V72"  "V73"  "V74"  "V75"  "V76"  "V77"  "V78"  "V79"
##  [81] "V80"  "V81"  "V82"  "V83"  "V84"  "V85"  "V86"  "V87"  "V88"  "V89"
##  [91] "V90"  "V91"  "V92"  "V93"  "V94"  "V95"  "V96"  "V97"  "V98"  "V99"
## [101] "V100"
```

```
glove_dataframe[1:5, 2:11]
```

```
##                     V1          V2          V3           V4         V5
## Absolutely  -0.4363577  0.06143880  0.25482302  0.426368580 0.51128006
## wonderful    0.1363224  0.08799818 -0.28555432  0.414705938 0.27912489
## silky       -0.1907303  0.22561103  0.16408352 -0.074531569 0.09923349
## sexy        -0.1763360 -0.37609145  0.07634876  0.003585817 0.24113904
## comfortable  0.4918008 -0.34670071  0.09589118  0.254491253 0.08746342
##                      V6          V7         V8          V9         V10
## Absolutely  -0.54583375 -0.06467300  0.2062296  0.07223483  0.08524629
## wonderful   -0.10739305  0.45752669  0.4431586 -0.43959993  0.09097157
## silky        0.19667212 -0.15375900 -0.1543391 -0.43959468  0.09586108
## sexy        -0.34918965  0.07899764 -0.1891043  0.06402219 -0.37922220
## comfortable  0.05240412 -0.05810079 -0.2754841 -0.12415825  0.12878827
```

```
# At the moment glove_dataframe is a matrix of 2878 rows (one for each 
feature) and 101 columns (1 column for word and the other 100 for the 100 
# dimensions of WE)
nrow(glove_dataframe)
```

```
## [1] 2878
```

```
ncol(glove_dataframe)
```

```
## [1] 101
```

```
0.006716308
```