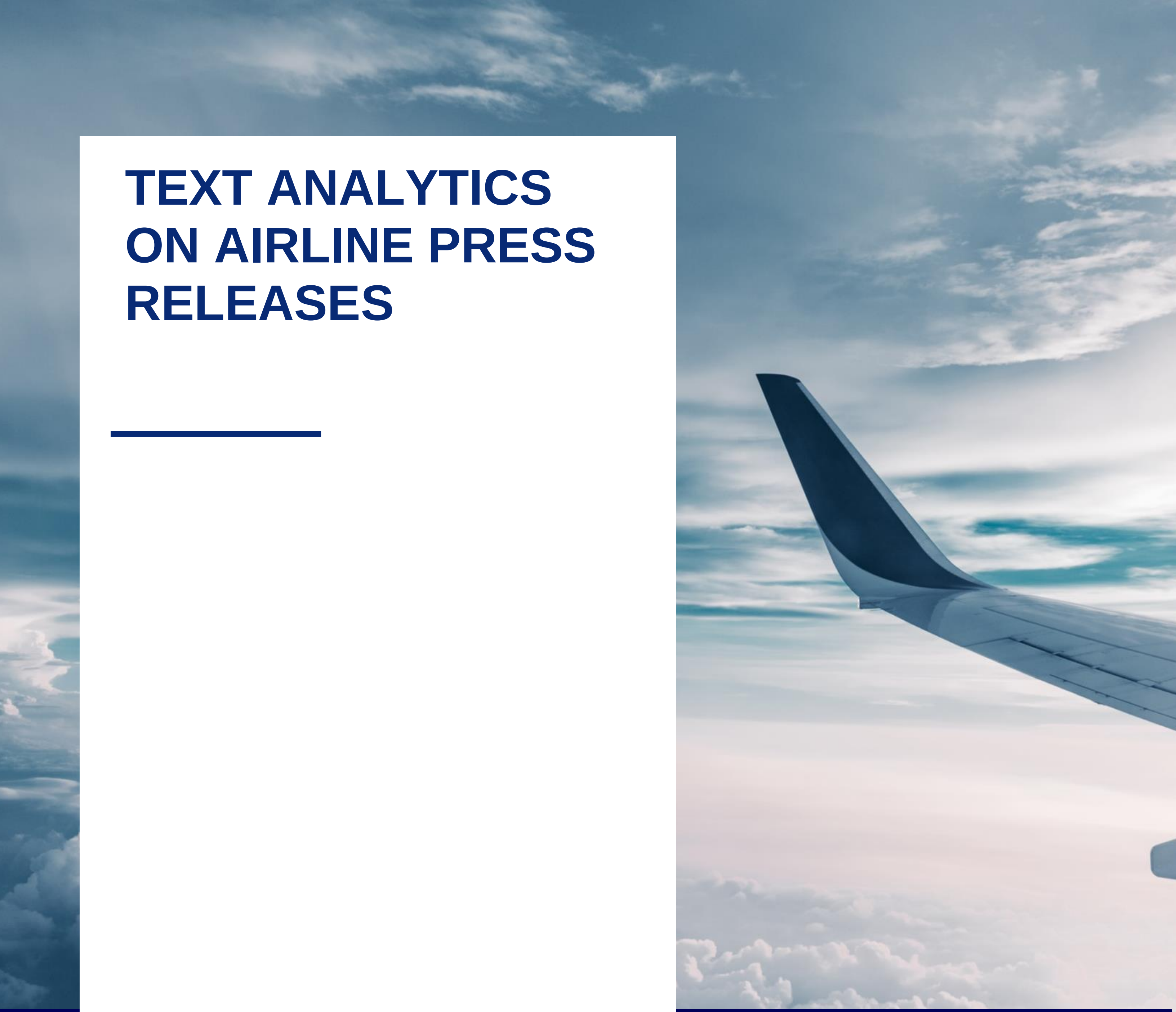
TEXT ANALYTICS ON AIRLINE PRESS RELEASES



FEBRUARY 10, 2021

Clemens Weisgram

Sentiment Analysis & Visualization of Publications of Airlines in Crisis Mode

Most airlines have spent the better part of the last 12 months fighting for economic survival. The industry has been hit by the COVID-19 pandemic like no other industry. Fear of traveling and legal restrictions to operate aircraft on various international routes caused major parts of the global air travel network to collapse.

Understanding the full context of the situation and the uncertainties ahead is crucial for creating a crisis response plan with the goal of quickly returning to the levels of connectivity present before the pandemic. Analysis of the sentiment of public statements can give valuable insights into how an airline – consciously or subconsciously – paints a picture around what investors can expect in the periods ahead.

Sentiment of quarterly earnings press releases serves as forward guidance. Even though mandatory publications of stock listed corporations are meant to review the most recent completed quarter's performance, the sentiment can indicate expectations about next quarter as well. In the case of airlines in crisis mode, communication to stakeholders (including shareholders, governments, and passengers) is key to assess risks and potential mitigating measures.

Not surprisingly, research indicates that the sentiment of press releases (as measured with the AFINN lexicon assigning scores between -5 and 5) takes a turn to the negative once the first effects of a (health, political, natural, etc.) crisis hit in. What is rather surprising is that the sentiment – on the decline and the rise – is leading the passenger volume by approximately one quarter.

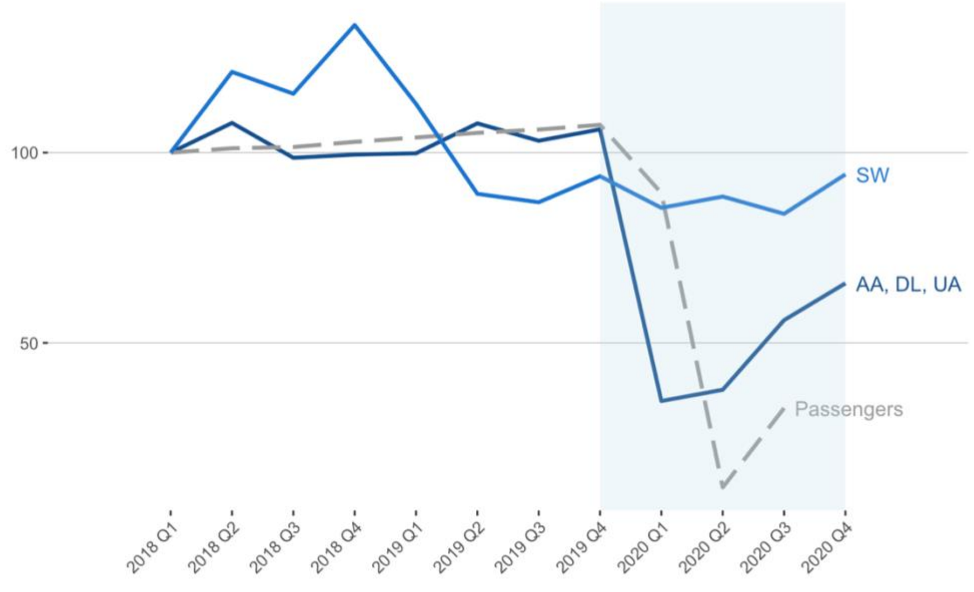
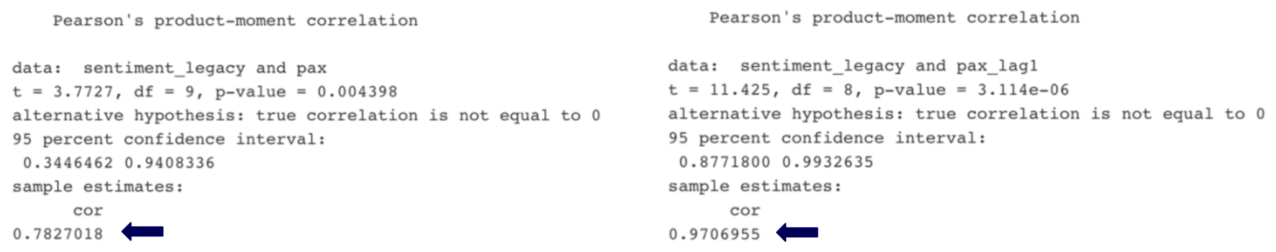


Figure 1: Airline sentiment and passenger volume (indexed, legacy carriers consolidated)[1]

[1] SW: Southwest Airlines, AA: American Airlines, DL: Delta Air Lines, UA: United Airlines

As visualized in *Figure 1*, especially the sentiment of quarterly statements of legacy carriers (in the US: American Airlines, Delta Air Lines, United Airlines) serves as a forward guidance for the next quarter's passenger volume.

In order to prove this mathematically, a new variable is introduced: *passenger volume lag-1*. This variable represents the time-series of *passenger volume* moved backwards by one quarter. As the correlation of *the passenger volume lag-1* is higher than the *passenger volume* on the non-adjusted time-series, there is the predictive power of the sentiment on next quarter's passenger volume.



```
        Pearson's product-moment correlation

data:  sentiment_legacy and pax
t = 3.7727, df = 9, p-value = 0.004398
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3446462 0.9408336
sample estimates:
      cor
0.7827018
```

```
        Pearson's product-moment correlation

data:  sentiment_legacy and pax_lag1
t = 11.425, df = 8, p-value = 3.114e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8771800 0.9932635
sample estimates:
      cor
0.9706955
```

Figure 2: Correlation analysis of sentiment with (1) normal time-series and (2) lag-1 time-series

Using this information, equity researchers and other observers of the financial markets can adjust their expectations about the upcoming performance based on the sentiment of the most recent press releases. More details about the sentiment gap between low-cost carriers (Southwest Airlines) and legacy carriers (American Airlines, Delta Air Lines, United Airlines) are described in the following paragraphs.

Utilizing press releases as opportunities to demonstrate achievements rather than only reporting results. While the legal requirement of giving access to the financial conduct of a corporation cannot be neglected, there is potential to send signals to the market through the sentiment of the reporting. The choice of words gives the observers an understanding of what is important.

Figure 3 demonstrates the differences in words that are most prominently used in press releases of airlines. The findings are in line with *Figure 1*, that Southwest Airlines maintained a more positive sentiment throughout the crisis whereas legacy carriers took a deep dive into more negative terminology.

The metric to build the word clouds are trigrams which showed more meaningful results after also reviewing bigrams and quadrograms for the respective periods. For the size of the words, the term frequency–inverse document frequency (tf_idf) in decreasing order was used. This metric highlights the trigrams that are most unique and meaningful to the press releases dependent on the year and the airline.

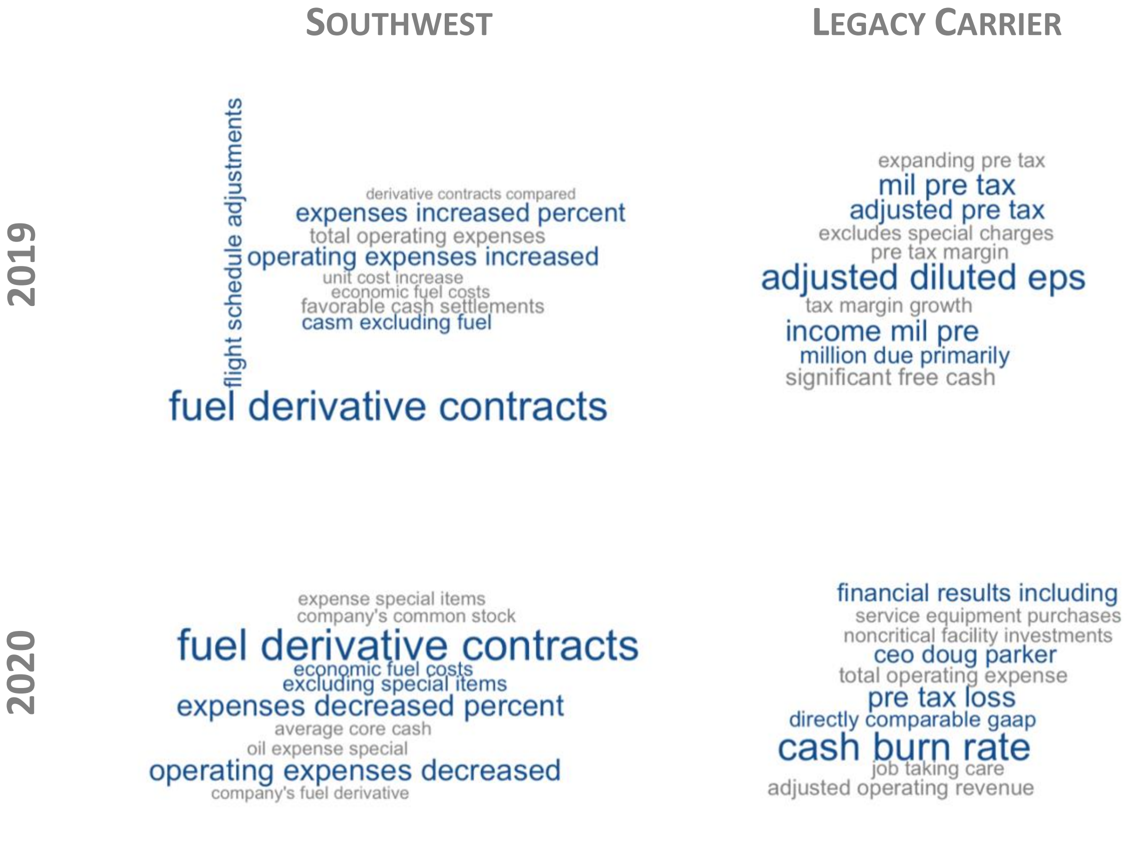


Figure 3: Word clouds per year and airline (word size: decreasing tf_idf)

Overall, the word clouds demonstrate that the concerns of airlines quickly changed from 2019 to 2020. While operational matters such as adjustments to the flight schedule or the increase of operating expenses were relevant in 2019, the situation quickly changed. Similarly, reporting financial matters about significant free cash and earnings per share (eps) are out of reach in 2020. In contrast, the most recent reportings are coined by the effects of the COVID-19 pandemic with a clear focus on decreasing expenses, managing the cash burn rate and a pre-tax loss.

Differentiating between the type of carrier reveals that Southwest Airlines used the publications to prominently highlight the achievements such as decreasing the expenses with noun-verb combinations in the 2020. In contrast, the word cloud of the legacy carriers shows more noun-only combinations of accounting terms such as cash burn rate or pre-tax loss.

In order to maintain better public perceptions, airlines – even in a crisis – are recommended to use press releases as means of demonstrating their achievements in dealing with challenges rather than purely disseminating accounting terminology (which can be better placed in the 10-Q and 10-K reports).

Airlines can use press releases to gauge the sentiment about their business. Outsiders can draw additional information from the choice of words. This report highlights the underlying information that can be drawn from the sentiment of quarterly press releases of airlines in a crisis. Using sentiment analysis and visualization tools for the results, provides a surprising method to approximate next quarter’s expected passenger volume in a crisis. On the other hand, airlines can consciously decide which words they want to communicate to the public to shape an opinion about their most recent and expected upcoming performance. By focusing on achievements and reducing the accounting terminology to a regulatory required minimum bears the potential of more positive sentiment and better public perception.

References:

Bureau of Transportation Statistics. (2020). *Monthly Passengers on U.S. Scheduled Airlines (Domestic + International), Seasonally Adjusted, October 2017 - October 2020.* Retrieved from https://www.bts.gov/figure-1-monthly-passengers-us-scheduled-airlines-domestic-international-seasonally-adjusted-october

A3_Business_Insight_Report_Clemens_Weisgram

Clemens Weisgram

2/7/2021

Set-up

```
# setting working directory
setwd("/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)
/A3_Business Insight Report")


# simplifying my life later
options(stringsAsFactors = FALSE)
```

Loading Packages

```
# loading all necessary libraries
library(textreadr)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidytext)
library(stringr)
library(tm)
```

```
## Loading required package: NLP
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(directlabels)
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## The following object is masked from 'package:tidyr':
##
##     crossing
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(ggraph)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(RcppRoll)
```

Defining own stopwords

Adding airline-specific and (selected) accounting-specific stopwords to the standard stopwords

```
# defining lexicon (= dataframe) with own stopwords
custom_lex <- data_frame(word=c("united", "american", "airlines", "delta", "southwest",
                                "quarter", "january", "february", "march", "april", "may",
                                "june", "july", "august", "september", "october", "november",
                                "december", "dallas", "texas", "net"),
                         lexicon="own"
                         )


#combining both lexicons the stop_words and the custom_lex
binded_lexicons <- rbind(stop_words, custom_lex)
```

Defining Helper Functions

```
# getting file name in file list
get_file_name <- function(file_list, index){
    file_name = file_list[index]
    file_name = gsub(".txt","",file_name)
    return(file_name)
}


# getting sentiment score with afinn lexicon (score between -5 and 5) & preliminary cleaning
get_sentiment <- function(file_list, txt){

    # creating empty dataframe to store results
    sentiment_df <- data.frame(Label = character(),
                           Sentiment = integer())

    # iterating over all text elements
    for (i in seq(1, length(which(txt == txt)), by=1)){

        # getting file name as location information
        file_name <- get_file_name(file_list ,i)

        # transforming data to a dataframe
        mydf <- data.frame(text = txt[i])

        # tokenising with afinn (on range from -5 to +5) and calculating frequency
        frequencies_tokens_nostop <- mydf %>%
                                unnest_tokens(word, text) %>%
                                anti_join(binded_lexicons) %>%
                                inner_join(get_sentiments("afinn")) %>%
                                count(word, value, sort=TRUE)

        # summing frequencies
        sentiment_mean <- mean(frequencies_tokens_nostop$value)

        # storing the frequency sum
        sentiment_df[nrow(sentiment_df) + 1,] = c(file_name, sentiment_mean)
    }
    return(sentiment_df)
}


# bigram
bigram <- function(txt){

    # tokenizing
    txt_bigram <- txt %>%
                    unnest_tokens(bigram, text, token = "ngrams", n = 2)
    # separating bigram into each word for individual analysis
    bigrams_separated <- txt_bigram %>%
                           separate(bigram, c("word1", "word2"), sep = " ")

    # removing bigrams with at least one stopword
    bigrams_filtered <- bigrams_separated %>%
      filter(!word1 %in% binded_lexicons$word) %>%
      filter(!word2 %in% binded_lexicons$word)

    # re-uniting words to bigram
    bigram_united <- bigrams_filtered %>%
      unite(bigram, word1, word2, sep=" ")

    # calculating tf_idf
    bigram_tf_idf <- bigram_united %>%
      count(airline, bigram) %>%
      bind_tf_idf(bigram, airline, n) %>%
      arrange(desc(tf_idf))

    return(bigram_tf_idf)
}


# trigram
trigram <- function(txt){

    # tokenizing
    txt_trigram <- txt %>%
                     unnest_tokens(trigram, text, token = "ngrams", n = 3)

    # separating trigram into each word for individual analysis
    trigrams_separated <- txt_trigram %>%
                           separate(trigram, c("word1", "word2", "word3"), sep = " ")

    # removing trigrams with at least one stopword
    trigrams_filtered <- trigrams_separated %>%
      filter(!word1 %in% binded_lexicons$word) %>%
      filter(!word2 %in% binded_lexicons$word) %>%
      filter(!word3 %in% binded_lexicons$word)

    # re-uniting words to trigram
    trigram_united <- trigrams_filtered %>%
      unite(trigram, word1, word2, word3, sep=" ")

    # calculating tf_idf
    trigram_tf_idf <- trigram_united %>%
      count(airline, trigram) %>%
      bind_tf_idf(trigram, airline, n) %>%
      arrange(desc(tf_idf))

    return(trigram_tf_idf)
}


# quadrogram
quadrogram <- function(txt){

    # tokenizing
    txt_quadrogram <- txt %>%
                         unnest_tokens(quadrogram, text, token = "ngrams", n = 4)

    # separating trigram into each word for individual analysis
    quadrograms_separated <- txt_quadrogram %>%
                        separate(quadrogram, c("word1", "word2", "word3", "word4"), sep = " ")

    # removing quadrograms with at least one stopword
    quadrograms_filtered <- quadrograms_separated %>%
      filter(!word1 %in% binded_lexicons$word) %>%
      filter(!word2 %in% binded_lexicons$word) %>%
      filter(!word3 %in% binded_lexicons$word) %>%
      filter(!word4 %in% binded_lexicons$word)

    # re-uniting words to quadrogram
    quadrogram_united <- quadrograms_filtered %>%
      unite(quadrogram, word1, word2, word3, word4, sep=" ")

    # calculating tf_idf
    quadrogram_tf_idf <- quadrogram_united %>%
      count(airline, quadrogram) %>%
      bind_tf_idf(quadrogram, airline, n) %>%
      arrange(desc(tf_idf))

    return(quadrogram_tf_idf)
}


# data preparation and harmonization
data_prep <- function(txt_data, airline_label){

    # converting to dataframe
    txt_data <- as.data.frame(txt_data)

    # setting period labels
    txt_data$period <- c("2018 Q1", "2018 Q2", "2018 Q3", "2018 Q4", "2019 Q1", "2019 Q2", "2019 Q3", "201
9 Q4", "2020 Q1", "2020 Q2", "2020 Q3", "2020 Q4")

    # setting year labels
    txt_data$year <- c("2018", "2018", "2018", "2018", "2019", "2019", "2019", "2019", "2020", "2020", "20
20", "2020")

    # adjusting number of airline labels
    txt_data$airline <- rep(airline_label, 12)

    # setting column names
    colnames(txt_data) <- c("text", "period", "year", "airline")

    # adjusting data types
    txt_data$airline <- factor(txt_data$airline, c("AA", "DL", "SW", "UA"))
    txt_data$year <- factor(txt_data$year, c("2018", "2019", "2020"))
    txt_data$period <- factor(txt_data$period, c("2018 Q1", "2018 Q2", "2018 Q3", "2018 Q4", "2019 Q1", "2
019 Q2", "2019 Q3", "2019 Q4", "2020 Q1", "2020 Q2", "2020 Q3", "2020 Q4"))

    return(txt_data)
}
```

Data Import and Preparation

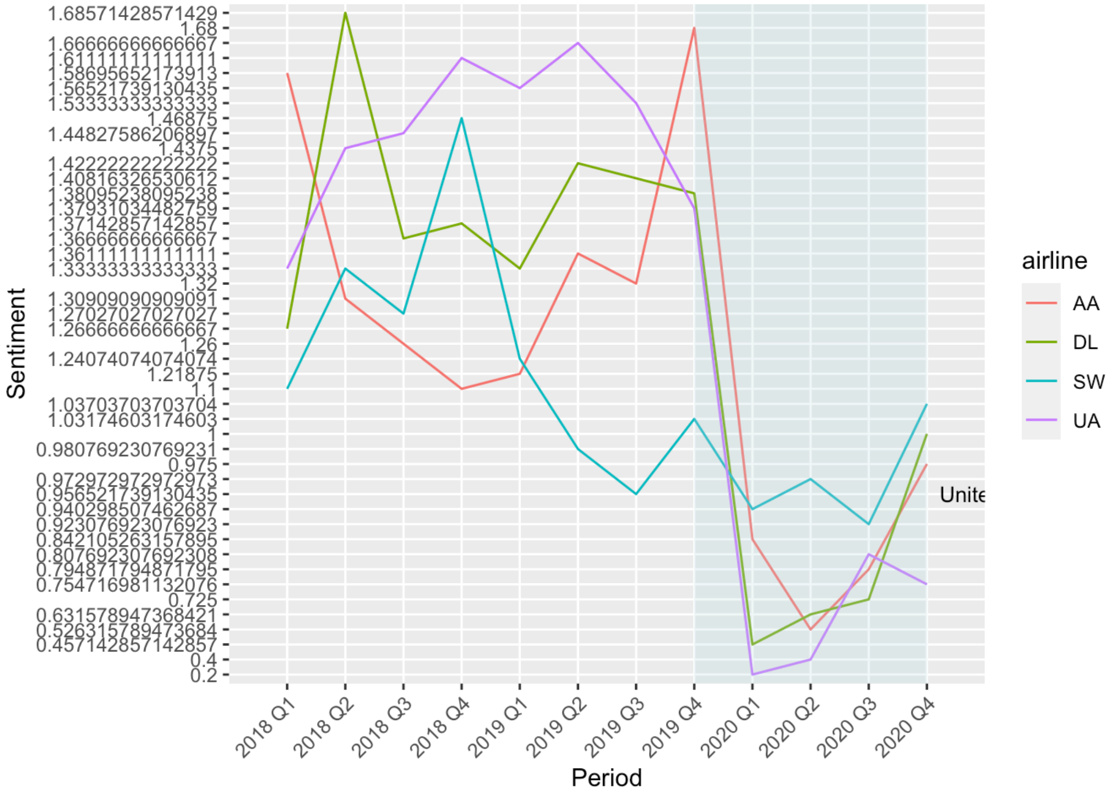
```
# setting destinations of stored files
file_dest_AA <- "/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)/A3_Business Insight Report/files/AA"
file_dest_DL <- "/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)/A3_Business Insight Report/files/DL"
file_dest_SW <- "/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)/A3_Business Insight Report/files/SW"
file_dest_UA <- "/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)/A3_Business Insight Report/files/UA"


# importing press releases from American Airlines (AA)
setwd(file_dest_AA)
file_list_american <- list.files()

txt_american <- do.call(rbind, lapply(file_list_american, function(x) paste(read_document(file=x), collapse = " ")))


# importing press releases from Delta Airlines (DL)
setwd(file_dest_DL)
file_list_delta <- list.files()

txt_delta <- do.call(rbind, lapply(file_list_delta, function(x) paste(read_document(file=x), collapse = " ")))


# importing press releases from Southwest Airlines (SW)
setwd("/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)/A3_Business Insight Report/files/SW")
file_list_southwest <- list.files()

txt_southwest <- do.call(rbind, lapply(file_list_southwest, function(x) paste(read_document(file=x), collapse = " ")))


# importing press releases from United Airlines (UA)
setwd(file_dest_UA)
file_list_united <- list.files()

txt_united <- do.call(rbind, lapply(file_list_united, function(x) paste(read_document(file=x), collapse = " ")))


# resetting wd to defaul for this project
setwd("/Users/clemensweisgram/HULT/MsBA/_Spring Term/_Text Analytics and Natural Language Processing (NLP)/A3_Business Insight Report")
```

Extracting Sentiments and Cleaning Data

```
# refer to helper function for details of sentiment analysis

# American Airlines
sentiment_df_american <- get_sentiment(file_list = file_list_american, txt = txt_american)

# Delta Airlines
sentiment_df_delta <- get_sentiment(file_list = file_list_delta, txt = txt_delta)

# Southwest Airlines
sentiment_df_southwest <- get_sentiment(file_list = file_list_southwest, txt = txt_southwest)

# United Airlines
sentiment_df_united <- get_sentiment(file_list = file_list_united, txt = txt_united)
```

Data Visualization on Original Scale

```
# creating empty dataframe
sentiment_df_all <- data.frame(period = character(48),
                               airline = character(48),
                               sentiment = integer(48))


# defining period labels (quarters)
sentiment_df_all$period <- c("2018 Q1", "2018 Q2", "2018 Q3", "2018 Q4", "2019 Q1", "2019 Q2", "2019 Q3",
"2019 Q4", "2020 Q1", "2020 Q2", "2020 Q3", "2020 Q4")


# assigning sentiment scores to dataframe
sentiment_df_all$sentiment[1:12] <- sentiment_df_american$Sentiment
sentiment_df_all$airline[1:12] <- "AA"

sentiment_df_all$sentiment[13:24] <- sentiment_df_delta$Sentiment
sentiment_df_all$airline[13:24] <- "DL"

sentiment_df_all$sentiment[25:36] <- sentiment_df_southwest$Sentiment
sentiment_df_all$airline[25:36] <- "SW"

sentiment_df_all$sentiment[37:48] = sentiment_df_united$Sentiment
sentiment_df_all$airline[37:48] = "UA"


# visualizing time-series of sentiments per airline [NOT FORMATTED]
ggplot(data = sentiment_df_all, aes(x = period, y = sentiment, group = airline)) +
    geom_line(aes(color = airline)) +
    theme(axis.text.x=element_text(angle=45,hjust=1)) +
    #coord_cartesian(ylim = c(0, 50)) +
    #scale_y_continuous(limits=c(-5, 5), breaks=c(-5,-4,-3,-2,-1,0,1,2,3,4,5)) +
    xlab("Period") +
    ylab("Sentiment") +
    annotate("rect", xmin = 8, xmax = 12, ymin = -Inf, ymax = Inf, fill = "lightblue", alpha = .2 )+
    scale_x_discrete(expand=c(0, 1)) +
    geom_dl(aes(label = c("United Airlines")), method = list(dl.trans(x = x + 0.2), "last.points", cex = 0
.8))
```



Data Visualization on Indexed Scale

Making variables on different scales comparable by indexing (start = 100)

Indexing Airline Sentiment Scores

```
# creating empty dataframe in horizonal format
sentiment_df_all_horizontal <- data.frame(period = character(12),
                                          AA = numeric(12),
                                          DL = numeric(12),
                                          SW = numeric(12),
                                          UA = numeric(12))

# transferring airline sentiment data (previously all in one column for easier visualization) to individua
l columns
sentiment_df_all_horizontal$AA <- sentiment_df_all[which(sentiment_df_all$airline == "AA"), ]$sentiment
sentiment_df_all_horizontal$DL <- sentiment_df_all[which(sentiment_df_all$airline == "DL"), ]$sentiment
sentiment_df_all_horizontal$SW <- sentiment_df_all[which(sentiment_df_all$airline == "SW"), ]$sentiment
sentiment_df_all_horizontal$UA <- sentiment_df_all[which(sentiment_df_all$airline == "UA"), ]$sentiment


# creating empty dataframe to store indexed values
sentiment_df_all_indexed <- data.frame(period = character(12),
                                          AA = numeric(12),
                                          DL = numeric(12),
                                          SW = numeric(12),
                                          UA = numeric(12),
                                          pax = numeric(12),
                                          pax_lag_minus1 = numeric(12))

# setting location information equal to previous dataframe
sentiment_df_all_indexed$period <- sentiment_df_all$period[1:12]


# setting initial value of all variables to 100
sentiment_df_all_indexed[1,2:7] <- c(100,100,100,100,100,100)


# indexing
for (j in c("AA", "DL", "SW", "UA")){
    for (i in seq(2, nrow(sentiment_df_all_indexed))){
    sentiment_df_all_indexed[i, j] <-
      sentiment_df_all_indexed[i-1,j]*(as.numeric(sentiment_df_all_horizontal[i,j]) /
                                              as.numeric(sentiment_df_all_horizontal[i-1,j]))
    }
}
```

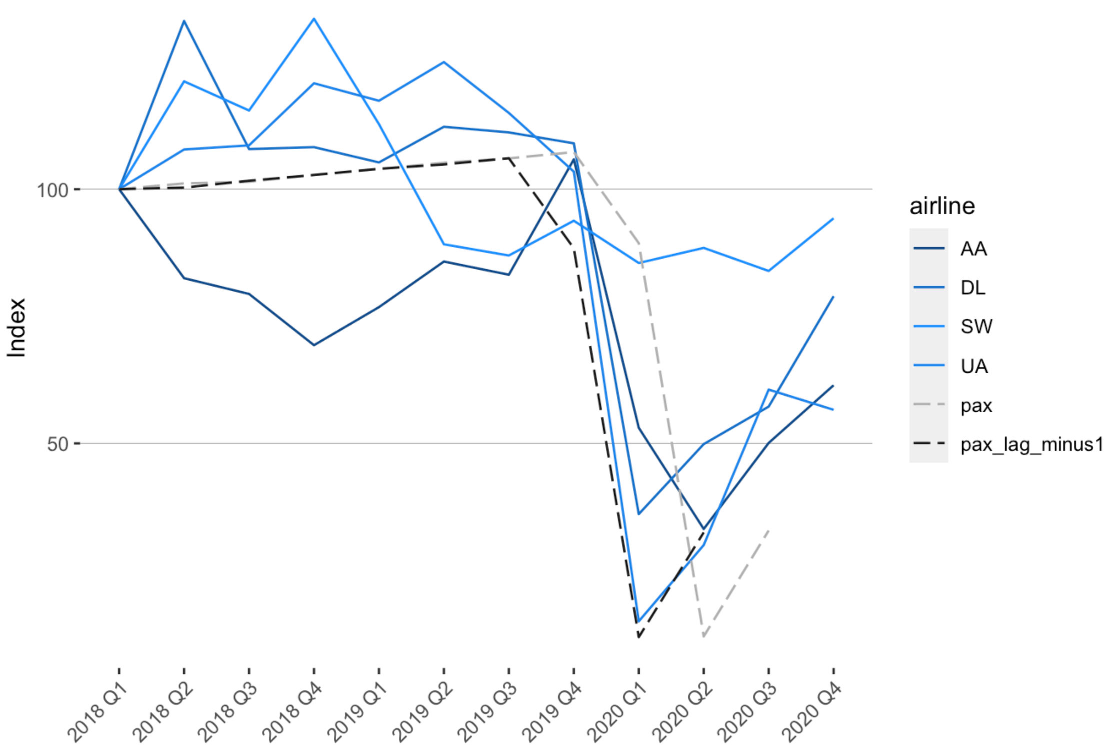
Adding Data About Passenger Volume

As the 4 major airlines hold a market share of close to 70%, the overall passenger volume is used to approximate airline performance

```
# importing data from csv file
pax_monthly <- read.csv("US_transportation_stats_Oct2020.csv")


# setting variable names for easier handling
colnames(pax_monthly) <- c("period", "pax")


# adjusting monthly data to quarterly data
    # calculating rolling sum
    pax <- pax_monthly %>%
        mutate(roll_sum = roll_sum(pax, 3, align = "right", fill = NA))

    # retaining only data points at end of quarters
    pax <- pax[which(as.numeric(rownames(pax_monthly)) %% 3 == 0),]

    # dropping unnecessary variable
    pax <- pax[-2]

    # setting variable names for easier handling
    colnames(pax) <- c("period", "pax")

    # adding empty row because data is not yet reported for most recent quarter
    pax[nrow(pax) + 1,] <- NA

    # setting period labels equal to other dataframe
    pax$period <- sentiment_df_all$period[1:12]


# indexing
for (i in seq(2, nrow(sentiment_df_all_indexed))){
sentiment_df_all_indexed[i, "pax"] <-
  sentiment_df_all_indexed[i-1,"pax"]*(as.numeric(pax[i,"pax"]) /
                                        as.numeric(pax[i-1,"pax"]))
}


# creating new, indexed variable with passenger volume moved ahead by one quarter
for (i in seq(2, nrow(sentiment_df_all_indexed))){
sentiment_df_all_indexed[i, "pax_lag_minus1"] <-
  sentiment_df_all_indexed[i-1,"pax_lag_minus1"]*(as.numeric(pax[i+1,"pax"]) /
                                        as.numeric(pax[i,"pax"]))
}
```

Visualizing Indexed Data

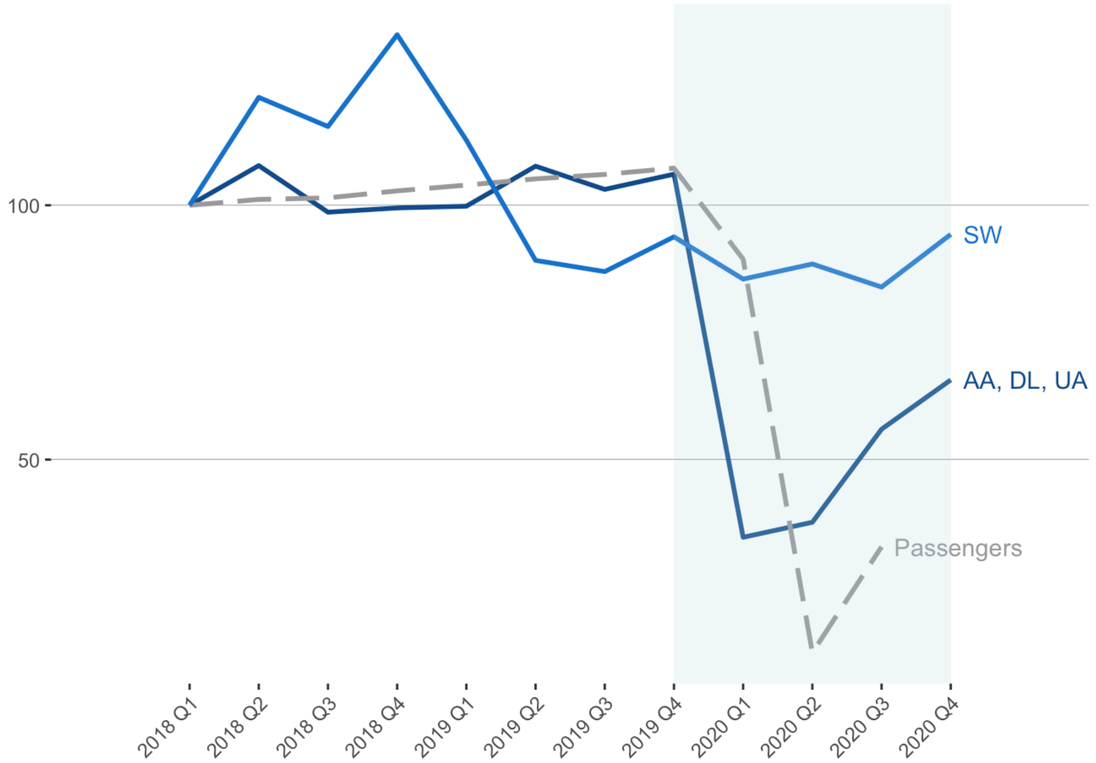
```
# rearranging the data (advised for easier visualization)
visualization_df <- sentiment_df_all_indexed %>%
  select(period, AA, DL, SW, UA, pax, pax_lag_minus1) %>%
  gather(key = "airline", value = "sentiment", -period) %>%
  na.omit()


# changing airline column to factor (advised for easier visualizations)
visualization_df$airline <- factor(visualization_df$airline, c("AA", "DL", "SW", "UA", "pax", "pax_lag_min
us1"))


# visualizing with ggplot2 [NOT FORMATTED]
ggplot(data = visualization_df, aes(x = period, y = sentiment, group = airline)) +
    geom_line(aes(color = airline, linetype = airline)) +
    scale_linetype_manual(values=c("solid","solid","solid","solid", "longdash","longdash")) +
    scale_color_manual(values=c("dodgerblue4","dodgerblue3","dodgerblue1","dodgerblue2","grey70","grey10")
) +
    #theme_bw() +
    theme(panel.background = element_blank(),
          panel.grid.major.x = element_blank(),
          panel.grid.major.y = element_line( size=.1, color="grey30", linetype = "solid"),
          axis.text.x=element_text(angle=45,hjust=1)) +
    #coord_cartesian(ylim = c(0, 50)) +
    #scale_y_continuous(limits=c(-5, 5), breaks=c(-5,-4,-3,-2,-1,0,1,2,3,4,5)) +
    xlab("") +
    ylab("Index")
```



```
    #annotate("rect", xmin = 8, xmax = 12, ymin = -Inf, ymax = Inf, fill = "lightblue", alpha = .2 )
    #scale_x_discrete(expand=c(0, 1)) +
    #geom_dl(aes(label = c("United Airlines")), method = list(dl.trans(x = x + 0.2), "last.points", cex =
0.8))
```

Consolidating & Formatting Visualization

```
# extract relevant data from previous dataframes
cons_vis_df <- sentiment_df_all_indexed %>%
  select(period, SW, pax)


# aggregating AA, DL, UA to a combined variable (because sentiment is highly correlated)
cons_vis_df$legacy <- (sentiment_df_all_indexed$AA + sentiment_df_all_indexed$DL +  sentiment_df_all_indexed$UA ) / 3


# rearranging the data (advised for easier visualization)
cons_vis_df <- cons_vis_df %>%
  select(period, legacy, SW, pax) %>%
  gather(key = "airline", value = "sentiment", -period) %>%
  na.omit()


# changing airline column to factor (advised for easier visualizations)
cons_vis_df$airline <- factor(cons_vis_df$airline, c("legacy", "SW", "pax"))


# adjusting labels
cons_vis_df$airline <- gsub("legacy", "AA, DL, UA", cons_vis_df$airline)
cons_vis_df$airline <- gsub("pax", "Passengers", cons_vis_df$airline)


# visualizing with ggplot2
ggplot(data = cons_vis_df, aes(x = period, y = sentiment, group = airline)) +
    geom_line(aes(color = airline, linetype = airline),size = 1, show.legend = FALSE) +
    scale_linetype_manual(values=c("solid", "longdash", "solid")) +
    scale_color_manual(values=c("dodgerblue4","grey60", "dodgerblue3")) +
    #theme_bw() +
    theme(panel.background = element_blank(),
          panel.grid.major.x = element_blank(),
          panel.grid.major.y = element_line( size=.1, color="grey30", linetype = "solid"),
          axis.text.x=element_text(angle=45,hjust=1)) +
    #coord_cartesian(ylim = c(0, 50)) +
    #scale_y_continuous(limits=c(-5, 5), breaks=c(-5,-4,-3,-2,-1,0,1,2,3,4,5)) +
    xlab("") +
    ylab("") +
    #scale_colour_discrete(guide = 'none') +
    scale_x_discrete(expand=c(0, 2)) +
    geom_dl(aes(label = airline, color = airline), method = list(dl.trans(x = x + 0.2), "last.points", cex = 0.9)) +
    annotate("rect", xmin = 8, xmax = 12, ymin = -Inf, ymax = Inf, fill = "lightblue", alpha = .2 )
```



```
# saving file to local drive
ggsave(filename = "visualization_indexed_formatted.png", width = 12, dpi=700)
```

Correlations

Demonstrating that passenger volume with lag -1 has a higher correlation with sentiment than passenger volume on orginal time-series.

```
# getting data from other parts of the script
sentiment_legacy <- cons_vis_df[which(cons_vis_df$airline == "AA, DL, UA"),]$sentiment
pax <- sentiment_df_all_indexed$pax
pax_lag1 <- sentiment_df_all_indexed$pax_lag_minus1


corr_sentiment_pax <- cor.test(sentiment_legacy, pax)
corr_sentiment_paxlag1 <- cor.test(sentiment_legacy, pax_lag1)

print(corr_sentiment_pax)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sentiment_legacy and pax
## t = 3.7727, df = 9, p-value = 0.004398
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3446462 0.9408336
## sample estimates:
##       cor
## 0.7827018
```

```
print(corr_sentiment_paxlag1)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sentiment_legacy and pax_lag1
## t = 11.425, df = 8, p-value = 3.114e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8771800 0.9932635
## sample estimates:
##       cor
## 0.9706955
```

Preparation for ngrams and other fun stuff

```
# preparing airline data
txt_american <- data_prep(txt_american, "AA")
txt_delta <- data_prep(txt_delta, "DL")
txt_southwest <- data_prep(txt_southwest, "SW")
txt_united <- data_prep(txt_united, "UA")


# combining all data to one structure
txt_full <- rbind(txt_american, txt_delta, txt_southwest, txt_united)


# remove numbers from entire dataset
txt_full$text <- removeNumbers(txt_full$text)


# separating dataset to yearly subsets
txt_full_2018 <- txt_full[which(txt_full$year == "2018"),]
txt_full_2019 <- txt_full[which(txt_full$year == "2019"),]
txt_full_2020 <- txt_full[which(txt_full$year == "2020"),]
```

Analyzing Tokens per Year

```
# same code applied to yearly datasets

# 2018
txt_tkn_2018 <- txt_full_2018 %>%
  unnest_tokens(word, text) %>%
  anti_join(binded_lexicons) %>%
  count(word, airline , sort=TRUE) %>%
  ungroup()

txt_tkn_2018_total <- txt_tkn_2018 %>%
                        group_by(airline) %>%
                        summarize(total=sum(n))

txt_tkn_2018 <- left_join(txt_tkn_2018, txt_tkn_2018_total)

txt_tkn_2018 <- txt_tkn_2018 %>%
 bind_tf_idf(word, airline, n)

txt_tkn_2018 %>%
  arrange(desc(tf_idf))
```

word	airline	n	total	tf	idf	tf_idf
<chr>	<fct>	<int>	<int>	<dbl>	<dbl>	<dbl>

ual	UA	30	1782	0.0168350168	1.3862944	2.333829e-02
adjusted	DL	79	3339	0.0236597784	0.6931472	1.639971e-02
delta's	DL	35	3339	0.0104821803	1.3862944	1.453139e-02
profitsharing	SW	36	4628	0.0077787381	1.3862944	1.078362e-02
oil	SW	35	4628	0.0075626621	1.3862944	1.048408e-02
contracts	SW	34	4628	0.0073465860	1.3862944	1.018453e-02
refinery	DL	23	3339	0.0068882899	1.3862944	9.549197e-03
ual's	UA	12	1782	0.0067340067	1.3862944	9.335316e-03
derivative	SW	31	4628	0.0066983578	1.3862944	9.285896e-03
american’s	AA	22	3411	0.0064497215	1.3862944	8.941213e-03

1-10 of 3,192 rows Previous **1** 2 3 4 5 6 ... 320 Next

```
# 2019
txt_tkn_2019 <- txt_full_2019 %>%
  unnest_tokens(word, text) %>%
  anti_join(binded_lexicons) %>%
  count(word, airline , sort=TRUE) %>%
  ungroup()

txt_tkn_2019_total <- txt_tkn_2019 %>%
                        group_by(airline) %>%
                        summarize(total=sum(n))

txt_tkn_2019 <- left_join(txt_tkn_2019, txt_tkn_2019_total)

txt_tkn_2019 <- txt_tkn_2019 %>%
 bind_tf_idf(word, airline, n)

txt_tkn_2019 %>%
  arrange(desc(tf_idf))
```

word	airline	n	total	tf	idf	tf_idf
<chr>	<fct>	<int>	<int>	<dbl>	<dbl>	<dbl>
groundings	SW	64	5562	0.0115066523	1.3862944	1.595161e-02
eps	UA	24	2364	0.0101522843	1.3862944	1.407405e-02
delta’s	DL	35	4033	0.0086784032	1.3862944	1.203082e-02
american’s	AA	17	2383	0.0071338649	1.3862944	9.889637e-03
profitsharing	SW	38	5562	0.0068320748	1.3862944	9.471267e-03
items	AA	29	2383	0.0121695342	0.6931472	8.435278e-03
max	SW	146	5562	0.0262495505	0.2876821	7.551525e-03
contracts	SW	30	5562	0.0053937433	1.3862944	7.477316e-03
derivative	SW	30	5562	0.0053937433	1.3862944	7.477316e-03
estimated	SW	29	5562	0.0052139518	1.3862944	7.228072e-03

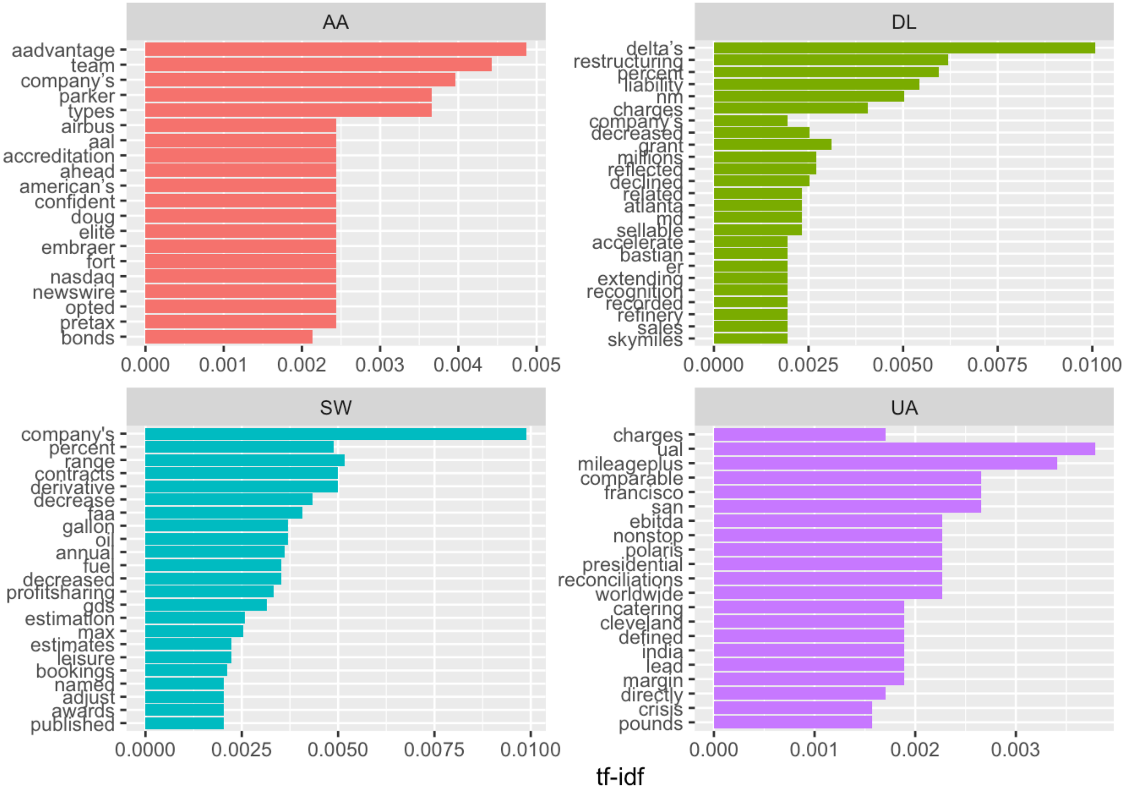
1-10 of 3,251 rows Previous **1** 2 3 4 5 6 ... 326 Next

```
# 2020
txt_tkn_2020 <- txt_full_2020 %>%
  unnest_tokens(word, text) %>%
  anti_join(binded_lexicons) %>%
  count(word, airline , sort=TRUE) %>%
  ungroup()

txt_tkn_2020_total <- txt_tkn_2020 %>%
                        group_by(airline) %>%
                        summarize(total=sum(n))

txt_tkn_2020 <- left_join(txt_tkn_2020, txt_tkn_2020_total)

txt_tkn_2020 <- txt_tkn_2020 %>%
 bind_tf_idf(word, airline, n)

txt_tkn_2020 %>%
  arrange(desc(tf_idf))
```

word	airline	n	total	tf	idf	tf_idf
<chr>	<fct>	<int>	<int>	<dbl>	<dbl>	<dbl>
delta’s	DL	26	3578	0.0072666294	1.3862944	1.007369e-02
company's	SW	107	7497	0.0142723756	0.6931472	9.892857e-03
restructuring	DL	16	3578	0.0044717719	1.3862944	6.199192e-03
percent	DL	74	3578	0.0206819452	0.2876821	5.949825e-03
liability	DL	14	3578	0.0039128004	1.3862944	5.424293e-03
range	SW	28	7497	0.0037348273	1.3862944	5.177570e-03
nm	DL	13	3578	0.0036333147	1.3862944	5.036844e-03
contracts	SW	27	7497	0.0036014406	1.3862944	4.992657e-03
derivative	SW	27	7497	0.0036014406	1.3862944	4.992657e-03
percent	SW	127	7497	0.0169401094	0.2876821	4.873366e-03

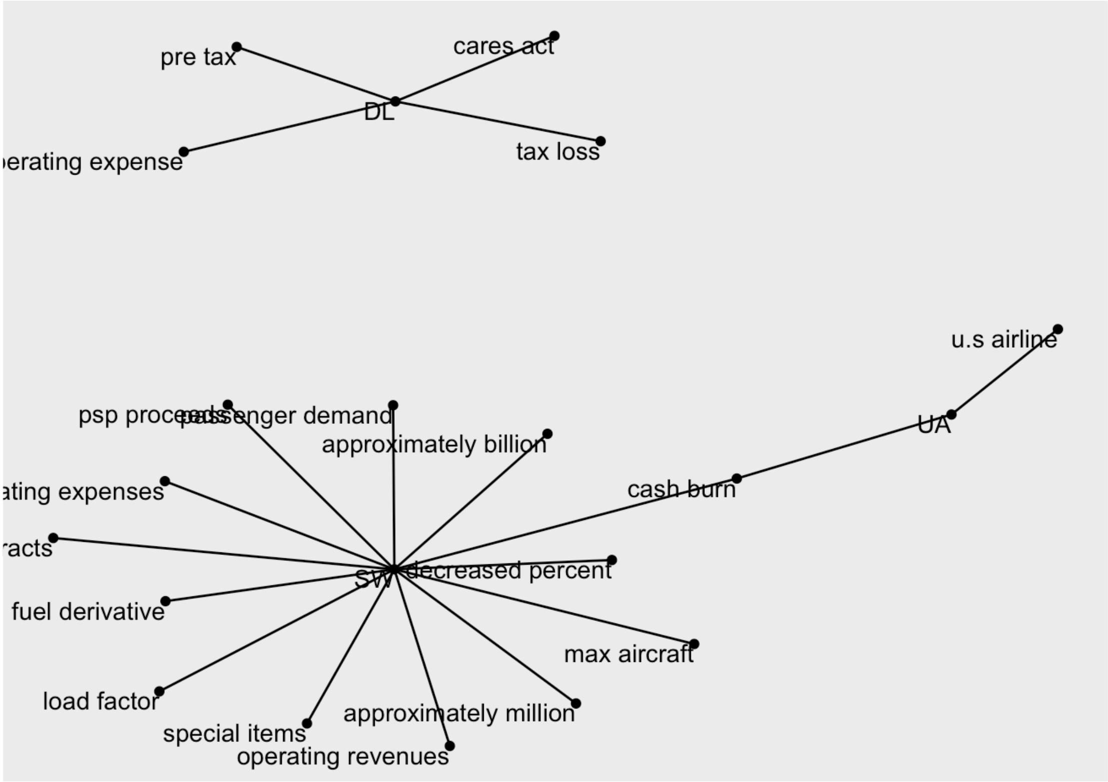
1-10 of 4,254 rows Previous **1** 2 3 4 5 6 ... 426 Next

```
# visualization of 2020 dataset
txt_tkn_2020 %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(airline) %>%
  top_n(20) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=airline))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~airline, ncol=2, scales="free")+
  coord_flip()
```

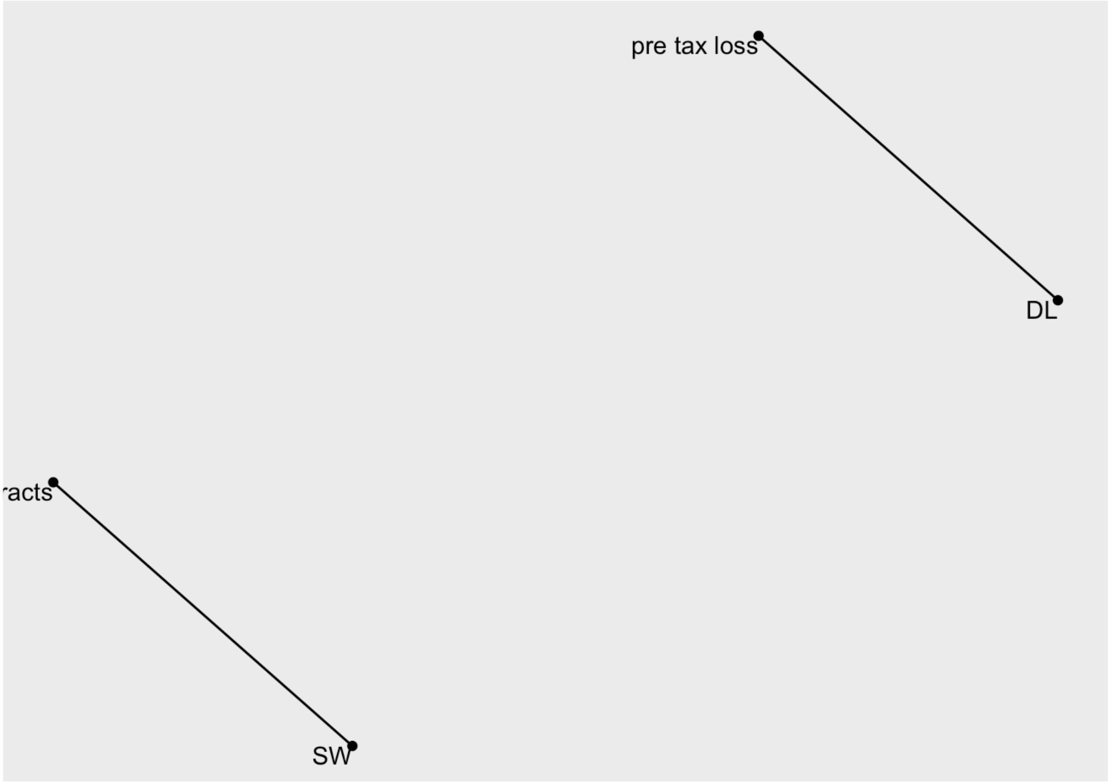


Generating and Visualizing n-grams

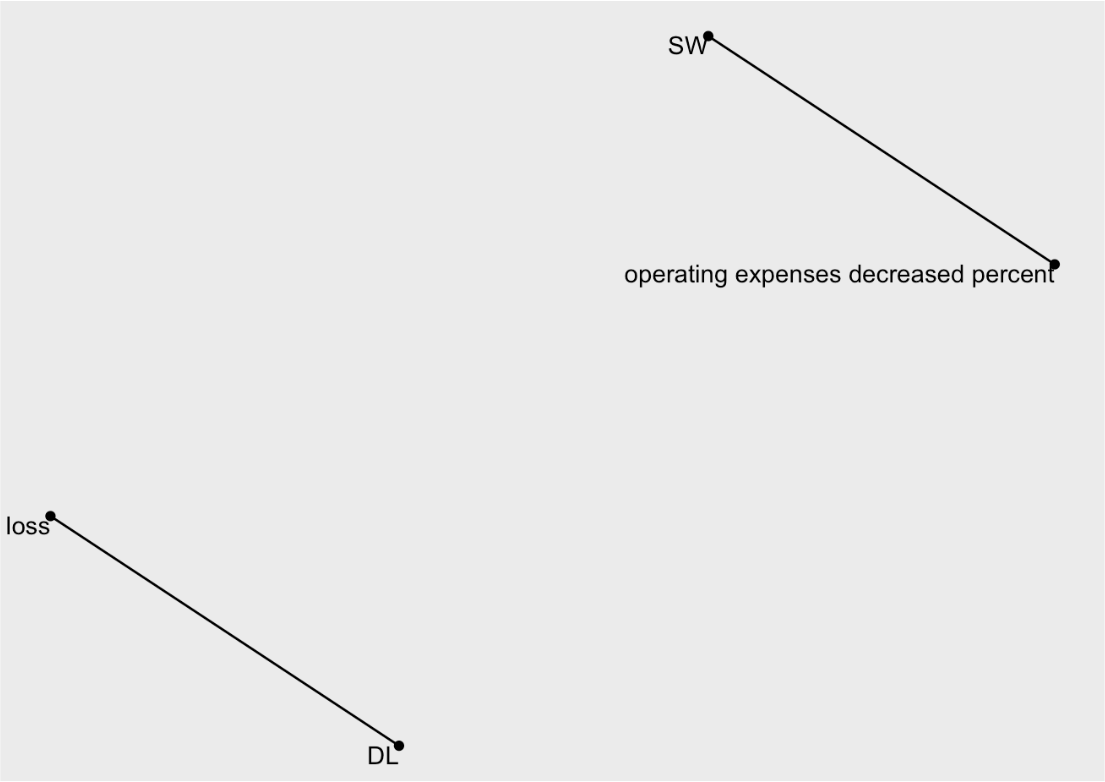
```
# bigram 2020
    # creating n-gram with helper function
    bigram2020 <- bigram(txt_full_2020)

    # preparing visualization by filtering
    bigram_graph <- bigram2020 %>%
                    filter(n>20) %>%
                    graph_from_data_frame()

    # graphing
    ggraph(bigram_graph, layout = "fr") +
      geom_edge_link()+
      geom_node_point()+
      geom_node_text(aes(label=name), vjust =1, hjust=1)
```



```
# trigram 2020
    # creating n-gram with helper function
    trigram2020 <- trigram(txt_full_2020)

    # preparing visualization by filtering
    trigram_graph <- trigram2020 %>%
                     filter(n>20) %>%
                     graph_from_data_frame()

    # graphing
    ggraph(trigram_graph, layout = "fr") +
      geom_edge_link()+
      geom_node_point()+
      geom_node_text(aes(label=name), vjust =1, hjust=1)
```



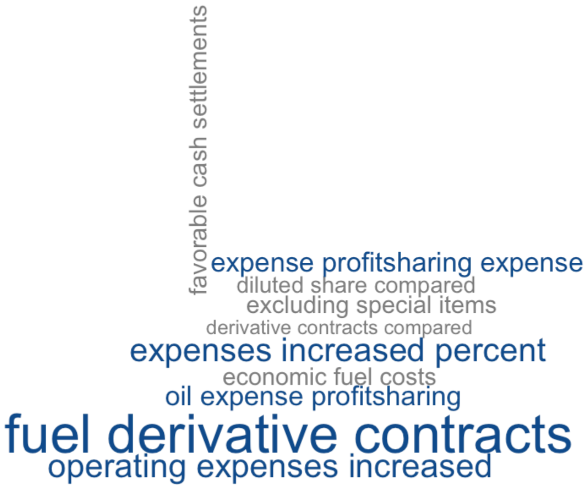
```
# quadrogram 2020
    # creating n-gram with helper function
    quadrogram2020 <- quadrogram(txt_full_2020)

    # preparing visualization by filtering
    quadrogram_graph <- quadrogram2020 %>%
                      filter(n>10) %>%
                      graph_from_data_frame()

    # graphing
    ggraph(quadrogram_graph, layout = "fr") +
      geom_edge_link()+
      geom_node_point()+
      geom_node_text(aes(label=name), vjust =1, hjust=1)
```



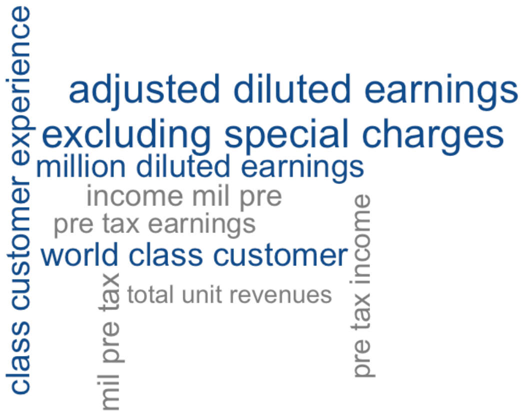
Generating Wordclouds

Wordclouds 2018

```
## wordcloud Southwest 2018
# generating trigram
trigram2018 <- trigram(txt_full_2018)

# setting colorlist for wordcloud (vector with same length as wordcloud_data)
colorlist5 = c(rep(c("dodgerblue4"), times = 5),
               rep(c("grey50"), times =  nrow(trigram2018[which(trigram2018$airline  == "SW"),]) - 5))

# generating wordcloud
#dev.new(width = 100, height = 100, unit = "in")
trigram2018 %>%
  filter(airline == "SW") %>%
  with(wordcloud(words = trigram, freq = tf_idf, scale = c(2,.3), max.words = 10,min.freq=3, colors= colorlist5, ordered.colors = T))
```

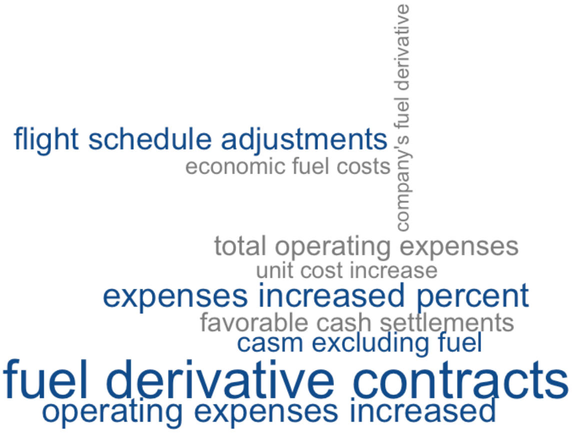


```
# wordcloud legacy carriers 2018
# generating trigram
trigram2018 <- trigram(txt_full_2018)

# setting colorlist for wordcloud (vector with same length as wordcloud_data)
colorlist6 = c(rep(c("dodgerblue4"), times = 5),
               rep(c("grey50"), times =  nrow(trigram2018[which(trigram2018$airline %in% c("AA", "DL", "UA
")),]) - 5))

# generating wordcloud
#dev.new(width = 100, height = 100, unit = "in")
trigram2018 %>%
  filter(airline %in% c("AA", "DL", "UA")) %>%
  with(wordcloud(words = trigram, freq = tf_idf, scale = c(1.5,.3), max.words = 10,min.freq=3, colors= col
orlist6, ordered.colors = T))
```

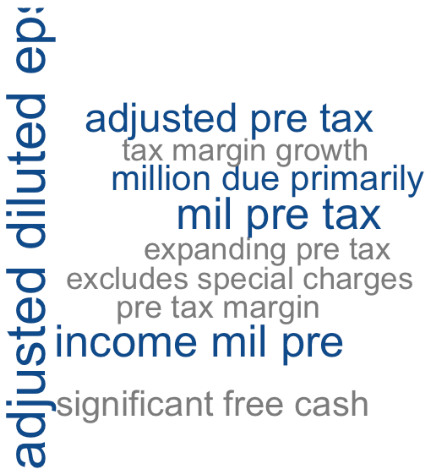


Wordclouds 2019

```
## wordcloud Southwest 2019
# generating trigram
trigram2019 <- trigram(txt_full_2019)

# setting colorlist for wordcloud (vector with same length as wordcloud_data)
colorlist3 = c(rep(c("dodgerblue4"), times = 5),
               rep(c("grey50"), times =  nrow(trigram2019[which(trigram2019$airline  == "SW"),]) - 5))

# generating wordcloud
#dev.new(width = 100, height = 100, unit = "in")
trigram2019 %>%
  filter(airline == "SW") %>%
  with(wordcloud(words = trigram, freq = tf_idf, scale = c(2,.3), max.words = 10,min.freq=3, colors= color
list3, ordered.colors = T))
```

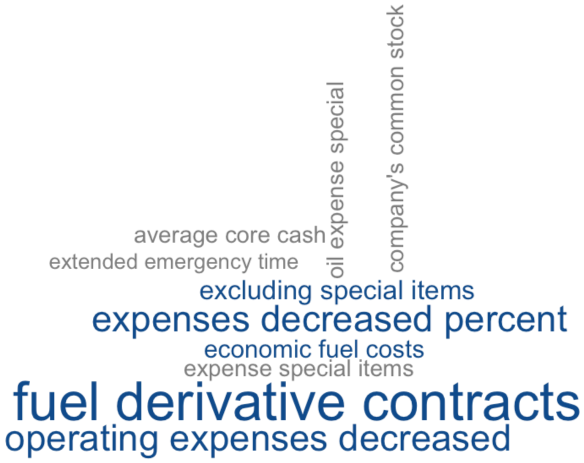


```
# wordcloud legacy carriers 2019
# generating trigram
trigram2019 <- trigram(txt_full_2019)

# setting colorlist for wordcloud (vector with same length as wordcloud_data)
colorlist4 = c(rep(c("dodgerblue4"), times = 5),
               rep(c("grey50"), times =  nrow(trigram2019[which(trigram2019$airline %in% c("AA", "DL", "UA
")),]) - 5))

# generating wordcloud
#dev.new(width = 100, height = 100, unit = "in")
trigram2019 %>%
  filter(airline %in% c("AA", "DL", "UA")) %>%
  with(wordcloud(words = trigram, freq = tf_idf, scale = c(2,.3), max.words = 10,min.freq=3, colors= color
list4, ordered.colors = T))
```

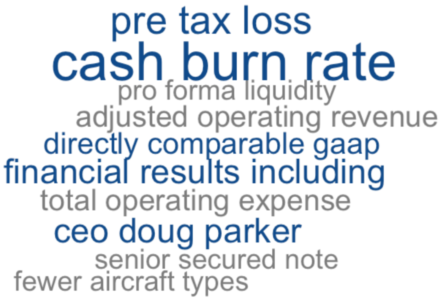


Wordclouds 2020

```
# wordcloud Southwest 2020

# setting colorlist for wordcloud (vector with same length as wordcloud_data)
colorlist1 = c(rep(c("dodgerblue4"), times = 5),
               rep(c("grey50"), times = nrow(trigram2020[which(trigram2020$airline == "SW"),]) - 5))

# generating wordcloud
#dev.new(width = 100, height = 100, unit = "in")
trigram2020 %>%
  filter(airline == "SW") %>%
  with(wordcloud(words = trigram, freq = tf_idf, scale = c(2,.3), max.words = 10,min.freq=3, colors= color
list1, ordered.colors = T))
```



```r
# wordcloud legacy carriers 2020

# setting colorlist for wordcloud (vector with same length as wordcloud_data)
colorlist2 = c(rep(c("dodgerblue4"), times = 5),
               rep(c("grey50"), times = nrow(trigram2020[which(trigram2020$airline %in% c("AA", "DL", "UA"))),]) - 5))

# generating wordcloud
trigram2020 %>%
  filter(airline %in% c("AA", "DL", "UA")) %>%
  with(wordcloud(words = trigram, freq = tf_idf, scale = c(2,.3), max.words = 10,min.freq=3, colors= colorlist2, ordered.colors = T))
```


pre tax loss
cash burn rate
pro forma liquidity
adjusted operating revenue
directly comparable gaap
financial results including
total operating expense
ceo doug parker
senior secured note
fewer aircraft types