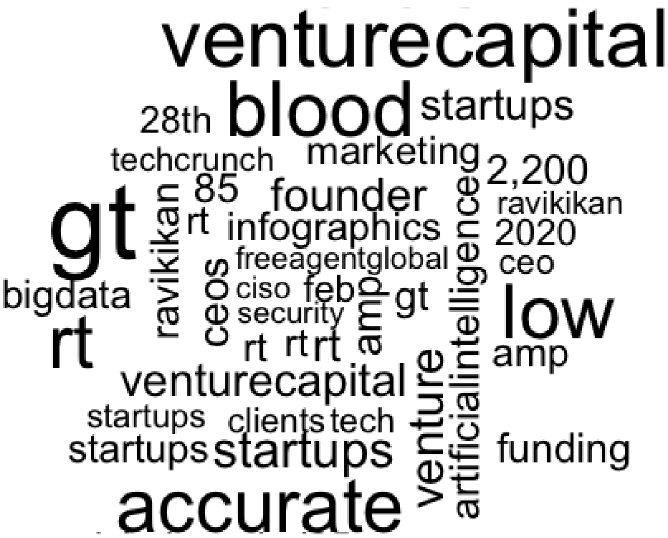
HULT INTERNATIONAL BUSINESS SCHOOL

Demeanor towards Venture Capital: an overview and a comparison through social media and magazines

Text Analytics: Business Insight Report



Prof: Thomas Kurnicki

Paolo Schirru

Table of Contents

Introduction

The scope of this work is to gather a broad view of the current happenings in the Venture Capital world; in particular, the author wishes to compare the demeanor towards Venture Capital firms in social media to that of notable magazines.

Are magazines' views aligned to those of social media active persons?
Are magazines' opinions having an impact on social media users?
Are the two groups focusing on the same topics and issues?

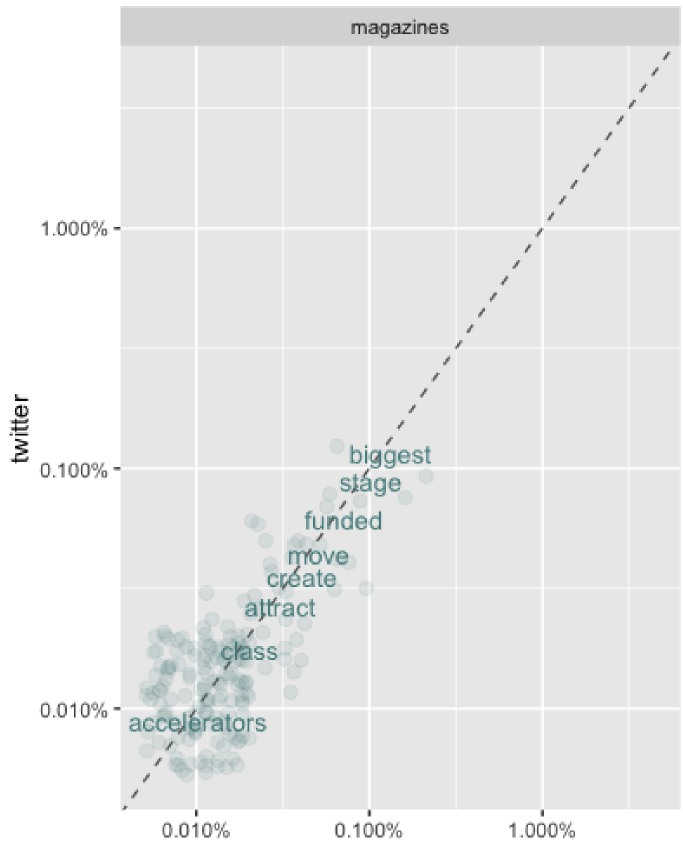
These are the kind of questions that shaped the run-through of the analysis, and that the analysis will try to answer, using a data science approach.

Data Collection

For the purpose of this study, the author has collected data on the matter, that had been posted on social media, in the current and in the previous year (2020 and 2019); the data obtained from respectable magazines, had also been published in the current and in the previous year (2020 and 2019).

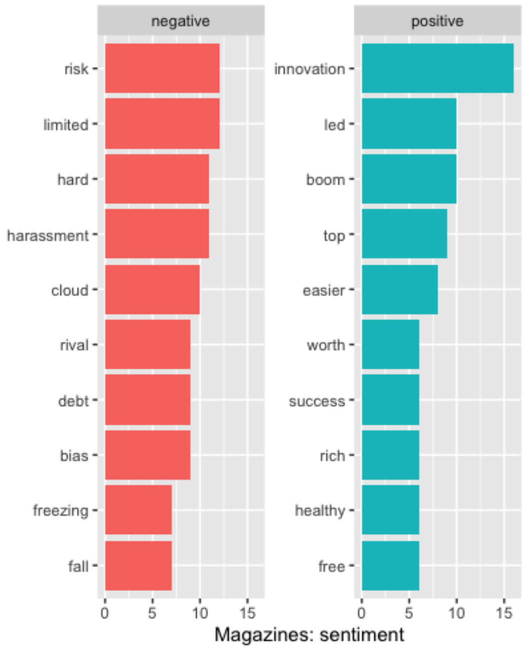
500 tweets have been randomly pulled out from Twitter, while 50 pages of articles have been randomly collected, mainly from "The Economist" (2020), but also from "The Harvard Business Review" (2020) and "The Financial Times" (2020).

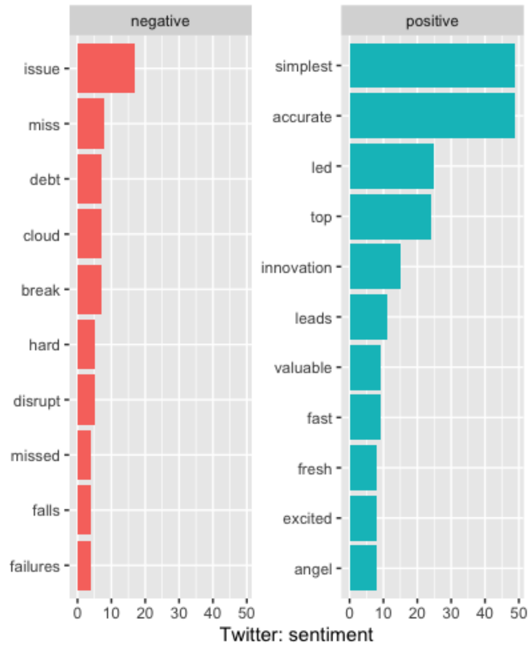
Similarities



The correlogram shows similarities between the two datasets. However, this should not surprise the reader, as both datasets focus on the same topic. Both Twitter users and magazines' authors address accelerators, funding stages, moves and attraction.

Sentiment Analysis





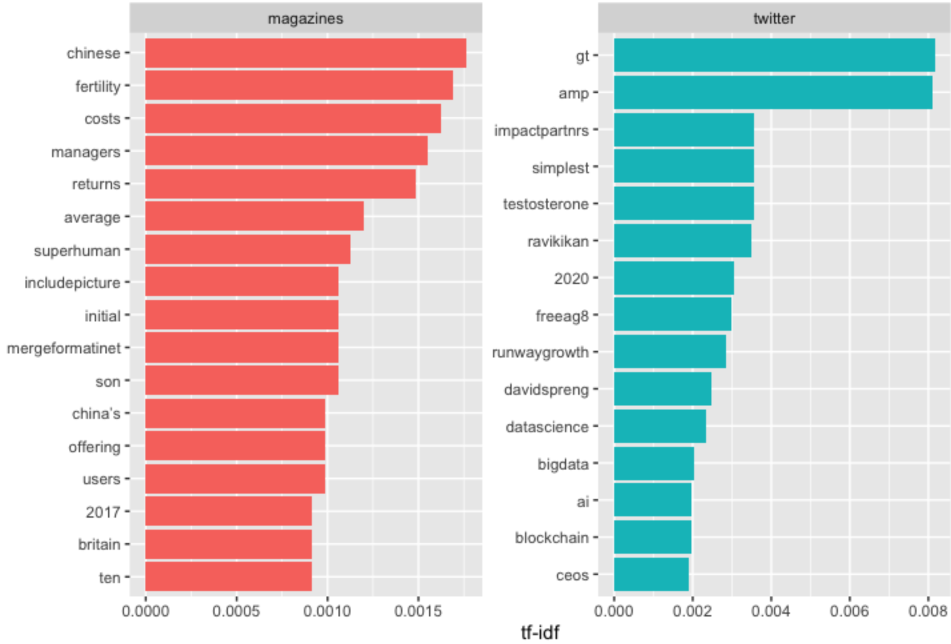
These two plots answer the initial questions about the views and opinion of social media users, compared to those of magazines' authors.
The sentiment analysis shows how Twitter's users are proportionally much more positive when writing about Venture Capital, compared to magazine's authors. Magazines' opinions are almost balanced between positive and negative, slightly tending towards negative opinions.

Twitter's users tend to focus on the individual gains and losses; for instance, one area of focus is the fear of missing opportunities, while magazines focus more on debt and on the limits of VCs. This might refer to the elite environment that surrounds VCs; magazines, even refer to harassment.

On the positive aspect, both sides consider VCs as something leading and innovative.
On Twitter, the resounding sentiments are those of excitement and value, while on magazines, sentiments of worth and success.

In both datasets, words like: "disrupt" can be considered positive or negative. Cloud is mistakenly considered a sentiment, while is referring to cloud technology. Angel is instead, not a sentiment, but refers to angel investors.

Tf-Idf



The tf-idf focuses on what makes unique and distinguishes each dataset, eventually a category or a group, and on what sets apart that group from the others.

In this case, what makes unique the twitter users is the focus on new technologies, such as data science, big data, ai, and blockchain. There are a few actors that are particularly active or taken into consideration among the users on Twitter: these are venture capital firms or even single users. The focus is on the year 2020.

Magazines focus instead on the Chinese economy, on Britain, probably referring to the effects of Brexit. Another common theme is fertility; in fact, VCs' investments are booming in this area. Only in 2019, Femtech received a bit less than $ 800 million in funding (Jaramillo, 2020). Magazines refer to this as "fertility", while on Twitter, there are more referrals to "testosterone".
There is a also a focus on returns and offerings as would be expected from VC firms.

Bigrams

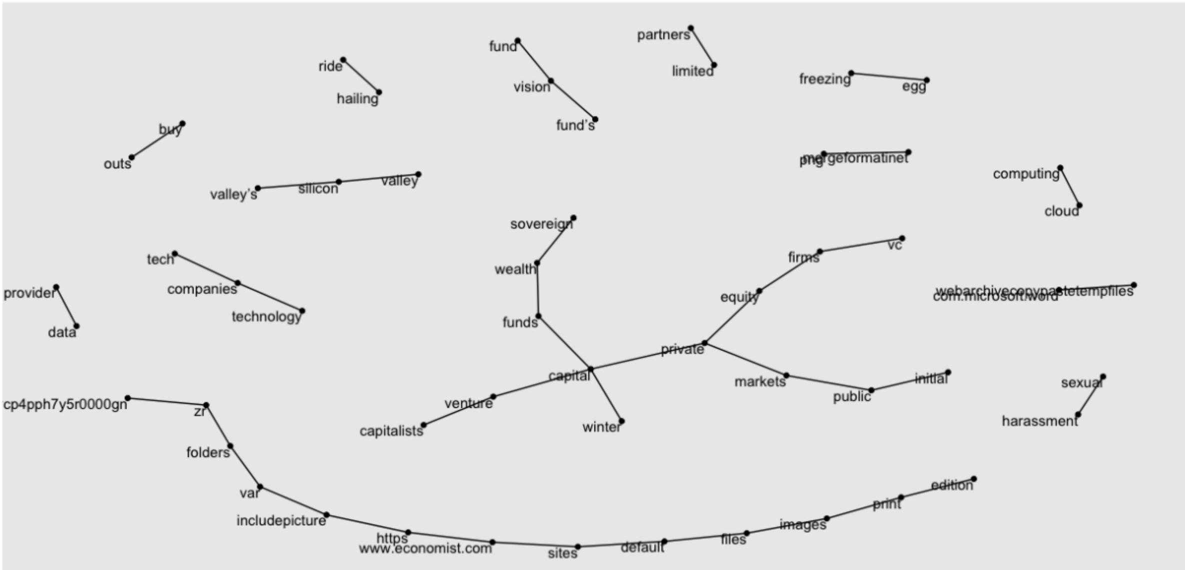
The bigrams show the words are related in each data set.

Magazines data

Despite rumors about an upcoming techquake, that sees the Silicon Valley no longer receiving special treatments from the U.S. government and being bridle by the Californian government (Suich Bass, 2019), the Silicon Valley still covers an important role in the U.S. and global economy. In fact, magazines still write about the valley.
Another reference is to data providers; in fact, nowadays, data is seen as the new gold (The economist, 2017). Thus, the reader should not be surprised if VCs have invested and continue to invest in data technologies.

There is also a connection that regards sexual harassment, that might regard what Nitasha Tiku (2017) reported on Wired, regarding female startup founders being harassed from venture capitalists and investors.
The most connected group regards common topics about private and public markets and investments; wealth and funds.

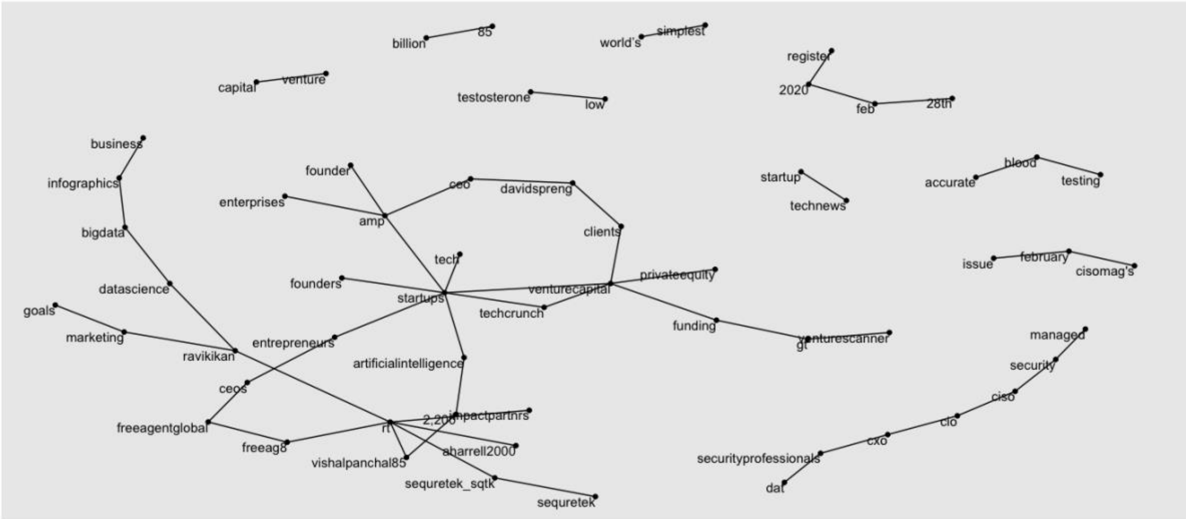
Other connections regard technology, cloud and ride (probably sharing) technologies.



Twitter data

Twitter's bigram is far more complex and connected than the previous one. However, one common theme, with the previous bigram is the relation with technology, here much more present. The emphasized topics are artificial intelligence, big data, data science, infographics, etc.
Another side of the graph underlines the growing importance of data, data management and data security. The connection highlights management, CIO (Chief Information Officer), security, etc. Indeed, with an increasing amount of data, fraud chances are also increasing. Thus, companies such as Paypal backed, Arkose Labs, are born with the scope of preventing frauds and protecting businesses and clients.

The graph also connects founders and CEOs, stating the importance of a good team, for a startup that aims for a Venture Capital seed investment. Enterprises are also referred. A few actors in the industry are also linked.



Summary and conclusion

The two datasets show similar themes, that stretch from innovation, value, technology and data science. Nevertheless, magazines' most common themes have a wider and more negative range, that vary from sexual harassment, to an upcoming techquake.

In terms of sentiment, Twitter users are definitely more positive about VCs than magazines' counterparts. On a scale that goes from -5 to +5, Twitter users have an average sentiment of 0.99, while magazines' positions, account for a 0.35 average sentiment. Both are positive; however, Twitter users communicate a more positive view on Venture Capital.

My Code

```
#Set working directory and import libraries
setwd("/Users/paoloschirru/Desktop/Venture Capital/")
library(dbplyr)
library(tinytex)
library(textreadr)
library(tidytext)
library(tidyverse)
library(twitteR)
library(tm)

#Import data and convert it to a data frame
vc_magazines <- read_document(file="All.docx")
vc_mag_df <- data_frame(text=vc_magazines)

#tibble of words that only make noise in our analysis, They will be remove
d later
noisy_words <- tibble(word = c("he","rt","accordiing","venturecapital","be
cause","capital","venture","fund"))

#Tokenise and remove stop words/noisy words
vc_mag_token <- vc_mag_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(noisy_words)

#Count and sort words
vc_mag_token %>%
  count ( word , sort = TRUE )

#Set keys to access Twitter
consumer_key <- 'XXXX'
consumer_secret <- 'XXXX'
access_token <- 'XXXX'
access_secret <- 'XXXX'


#Load keys
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_se
cret)

#Search twitter
twitter_search <- twitteR::searchTwitter('#venturecapital', n = 1000, lang
= 'en', since = '2019-06-01', retryOnRateLimit = 1e3)
vc_twitter = twitteR::twListToDF(twitter_search)

#remove noise
vc_twitter$text <- gsub("http[^[:space:]]*","",  vc_twitter$text)
vc_twitter$text <- gsub("http[^[:space:]]*","", vc_twitter$text)

#tokenize twitter data and remove unnecessary words
```

```
vc_tweet_token <- vc_twitter %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(noisy_words)

#save twitter data to excel
#library(openxlsx)
#write.xlsx(vc_tweet_token, 'vc_tweet_tokenss.xlsx')

#obtain frequency to  later plot a corellogram
frequency <- bind_rows(mutate(vc_tweet_token, author="twitter"),
                       mutate(vc_mag_token, author= "magazines")) %>%
  mutate(word=str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n))%>%
  spread(author, proportion) %>%
  gather(author, proportion, `magazines`)

library(scales)

#plot frequency in order to get a corellogram
ggplot(frequency, aes(x=proportion, y=`twitter`,
                      color = abs(`twitter`- proportion)))+
  geom_abline(color="grey40", lty=2)+
  geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
  geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels= percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high =
"gray75")+
  facet_wrap(~author, ncol=2)+
  theme(legend.position = "none")+
  labs(y= "twitter", x=NULL)

#get sentiments for the two datasets

vc_mag_token %>%
  inner_join(get_sentiments("afinn"))%>%
  group_by(word) %>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

vc_tweet_token %>%
  inner_join(get_sentiments("afinn"))%>%
  group_by(id) %>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

vc_mag_sentiment <- vc_mag_token %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T)

#plot sentiments
```

```
vc_mag_sentiment %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Magazines: sentiment", x=NULL)+
  coord_flip()

vc_twitter_sentiment <- vc_tweet_token %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T)

vc_twitter_sentiment %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Twitter: sentiment", x=NULL)+
  coord_flip()

vc_mag_token %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

vc_tweet_token %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

twitter_words <- lengths(vc_tweet_token)

############################################
## TF-IDF analysis
#############################################
#combine the data
combined_sources <- bind_rows(mutate(vc_twitter, from="twitter"),
                           mutate(vc_mag_df, from= "magazines")
)

#unnest and count words
twitt_modif <- combined_sources %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(noisy_words) %>%
  count(from, word, sort=TRUE) %>%
  ungroup()
```

```
#grouping
twitt_modif2 <- twitt_modif %>%
  group_by(from) %>%
  summarize(total=sum(n))

#left join the two datasets
sources_leftjoined <- left_join(twitt_modif, twitt_modif2)

tidy_twitt_tfidf <- sources_leftjoined %>%
  bind_tf_idf(word, from, n)

tidy_twitt_tfidf

#order descending
tidy_twitt_tfidf %>%
  arrange(desc(tf_idf))


#ploting tf-idf
tidy_twitt_tfidf %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(from) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=from))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~from, ncol=2, scales="free")+
  coord_flip()

#Creating Bigrams
vc_mag_bigrams <- vc_mag_df %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

vc_mag_bigrams %>%
  count(bigram, sort = TRUE)

library(tidyr)

#magazines bigrams
bigrams_mag_separated <- vc_mag_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_mag_filtered <- bigrams_mag_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigram_mag_counts <- bigrams_mag_filtered %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
bigram_mag_counts

#twitter bigrams
```

```
vc_twitt_bigrams <- vc_twitter %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

vc_twitt_bigrams %>%
  count(bigram, sort = TRUE) #this has many stop words, need to remove the
m

#to remove stop words from the bigram data, we need to use the separate fu
nction:
library(tidyr)
bigrams_twitt_separated <- vc_twitt_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_twitt_filtered <- bigrams_twitt_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
bigram_twitt_counts <- bigrams_twitt_filtered %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
bigram_twitt_counts

#plotting

library(igraph)
mag_graph <- bigram_mag_counts %>%
  filter(n>20) %>%
  graph_from_data_frame()

mag_graph

library(ggraph)
ggraph(mag_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

twitt_graph <- bigram_twitt_counts %>%
  filter(n>20) %>%
  graph_from_data_frame()

ggraph(twitt_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

#Bigrams
big_twitt_n <- bigram_twitt_counts %>%
  filter(n > 12)

big_mag_n <- bigram_mag_counts %>%
  filter(n>5)
```

```
big_joined <- bigram_twitt_counts %>%
  full_join(bigram_mag_counts) %>%
  filter(n>5)

ggraph(big_twitt_n, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

ggraph(big_mag_n, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

twitt_graph <- bigram_twitt_counts %>%
  filter(n>20) %>%
  graph_from_data_frame()

ggraph(twitt_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

#######################################################
#COVER PAGE
#Wordclouds for the cover page
library(wordcloud)

vc_cloud <- vc_mag_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word)

wordcloud(
  words = vc_cloud$word,
  freq = vc_cloud$n,
  max.words = 45,
  colors = 'blue',
  ordered.colors = TRUE
)

bigram_twitt_counts

wordcloud(
  words = bigram_twitt_counts$word1,
  freq = bigram_twitt_counts$n,
  max.words = 45
)



#other

vc_mag_token %>%
```

```
  inner_join(get_sentiments("afinn"))%>%

  summarise(sentiment=mean(value)) %>%

  mutate(method="AFINN")


vc_tweet_token %>%

  inner_join(get_sentiments("afinn"))%>%

  summarise(sentiment=mean(value)) %>%

  mutate(method="AFINN")
```

R Output

```
> #Set working directory and import libraries
> setwd("/Users/paoloschirru/Desktop/Venture Capital/")
> library(dbplyr)
> library(tinytex)
> library(textreadr)
> library(tidytext)
> library(tidyverse)
> library(twitteR)
> library(tm)
> #Import data and convert it to a data frame
> vc_magazines <- read_document(file="All.docx")
> vc_mag_df <- data_frame(text=vc_magazines)
> #tibble of words that only make noise in our analysis, They will be removed later
> noisy_words <- tibble(word =
c("he","rt","accordiing","venturecapital","because","capital","venture","fund"))
> #Tokenise and remove stop words/noisy words
> vc_mag_token <- vc_mag_df %>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words) %>%
+   anti_join(noisy_words)
Joining, by = "word"
Joining, by = "word"
> #Count and sort words
> vc_mag_token %>%
+   count ( word , sort = TRUE )
# A tibble: 3,887 x 2
   word        n
   <chr>     <int>
```

```
 1 firms       107
 2 private     100
 3 investors    88
 4 public       74
 5 funds        59
 6 startups     59
 7 firm         56
 8 tech         50
 9 companies    49
10 silicon      45
# … with 3,877 more rows
> #Set keys to access Twitter
> consumer_key <-
> consumer_secret <-
> access_token <-
> access_secret <-

> #Load keys
> setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
[1] "Using direct authentication"
> #Search twitter
> twitter_search <- twitteR::searchTwitter('#venturecapital', n = 1000, lang = 'en', since =
'2019-06-01', retryOnRateLimit = 1e3)
> vc_twitter = twitteR::twListToDF(twitter_search)
> #remove noise
> vc_twitter$text <- gsub("http[^[:space:]]*","",  vc_twitter$text)
> vc_twitter$text <- gsub("http[^[:space:]]*","", vc_twitter$text)
> #tokenize twitter data and remove unnecessary words
> vc_tweet_token <- vc_twitter %>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words) %>%
+   anti_join(noisy_words)
Joining, by = "word"
Joining, by = "word"
> #obtain frequency to  later plot a corellogram
> frequency <- bind_rows(mutate(vc_tweet_token, author="twitter"),
+                        mutate(vc_mag_token, author= "magazines")) %>%
+   mutate(word=str_extract(word, "[a-z']+")) %>%
+   count(author, word) %>%
+   group_by(author) %>%
+   mutate(proportion = n/sum(n))%>%
+   spread(author, proportion) %>%
+   gather(author, proportion, `magazines`)
> library(scales)
> #plot frequency in order to get a corellogram
> ggplot(frequency, aes(x=proportion, y=`twitter`,
+                       color = abs(`twitter`- proportion)))+
+   geom_abline(color="grey40", lty=2)+
+   geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
+   geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
```

```
+   scale_x_log10(labels = percent_format())+
+   scale_y_log10(labels= percent_format())+
+   scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
+   facet_wrap(~author, ncol=2)+
+   theme(legend.position = "none")+
+   labs(y= "twitter", x=NULL)
Warning messages:
1: Removed 5732 rows containing missing values (geom_point).
2: Removed 5732 rows containing missing values (geom_text).
> vc_mag_token %>%
+   inner_join(get_sentiments("afinn"))%>%
+   group_by(word) %>%
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 342 x 3
   word         sentiment method
   <chr>            <dbl> <chr>
 1 abandon             -4 AFINN
 2 abilities            2 AFINN
 3 ability              2 AFINN
 4 accept               2 AFINN
 5 accidentally        -2 AFINN
 6 accomplished         4 AFINN
 7 accused             -6 AFINN
 8 accusing            -2 AFINN
 9 active               2 AFINN
10 admit               -2 AFINN
# … with 332 more rows
> vc_tweet_token %>%
+   inner_join(get_sentiments("afinn"))%>%
+   group_by(id) %>%
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 317 x 3
   id                  sentiment method
   <chr>                   <dbl> <chr>
 1 1225717706906898433        -2 AFINN
 2 1225720963532296192         2 AFINN
 3 1225721113277337600         1 AFINN
 4 1225723792460263424         2 AFINN
 5 1225725057374269440         2 AFINN
 6 1225728393129418752         2 AFINN
 7 1225728415040405505         2 AFINN
 8 1225728603943571457         2 AFINN
 9 1225729374160330752         2 AFINN
10 1225729493131640832         3 AFINN
# … with 307 more rows
> vc_mag_sentiment <- vc_mag_token %>%
```

```
+   inner_join(get_sentiments("bing")) %>%
+   count(word, sentiment, sort=T)
Joining, by = "word"
> #plot sentiments
> vc_mag_sentiment %>%
+   group_by(sentiment) %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(word=reorder(word, n)) %>%
+   ggplot(aes(word, n, fill=sentiment)) +
+   geom_col(show.legend = FALSE) +
+   facet_wrap(~sentiment, scales = "free_y")+
+   labs(y="Magazines: sentiment", x=NULL)+
+   coord_flip()
Selecting by n
> vc_twitter_sentiment <- vc_tweet_token %>%
+   inner_join(get_sentiments("bing")) %>%
+   count(word, sentiment, sort=T)
Joining, by = "word"
> vc_twitter_sentiment %>%
+   group_by(sentiment) %>%
+   top_n(10) %>%
+   ungroup() %>%
+   mutate(word=reorder(word, n)) %>%
+   ggplot(aes(word, n, fill=sentiment)) +
+   geom_col(show.legend = FALSE) +
+   facet_wrap(~sentiment, scales = "free_y")+
+   labs(y="Twitter: sentiment", x=NULL)+
+   coord_flip()
Selecting by n
> vc_mag_token %>%
+   inner_join(get_sentiments("afinn"))%>%
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 1 x 2
  sentiment method
      <dbl> <chr>
1       286 AFINN
> vc_tweet_token %>%
+   inner_join(get_sentiments("afinn"))%>%
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
  sentiment method
1       418  AFINN
> twitter_words <- lengths(vc_tweet_token)
> ###############################################
> ## TF-IDF analysis
> ################################################
```

```
> #combine the data
> combined_sources <- bind_rows(mutate(vc_twitter, from="twitter"),
+                               mutate(vc_mag_df, from= "magazines")
+ )
> #unnest and count words
> twitt_modif <- combined_sources %>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words) %>%
+   anti_join(noisy_words) %>%
+   count(from, word, sort=TRUE) %>%
+   ungroup()
Joining, by = "word"
Joining, by = "word"
> #grouping
> twitt_modif2 <- twitt_modif %>%
+   group_by(from) %>%
+   summarize(total=sum(n))
> #left join the two datasets
> sources_leftjoined <- left_join(twitt_modif, twitt_modif2)
Joining, by = "from"
> tidy_twitt_tfidf <- sources_leftjoined %>%
+   bind_tf_idf(word, from, n)
> tidy_twitt_tfidf
# A tibble: 6,460 x 7
   from      word          n total      tf   idf  tf_idf
   <chr>     <chr>     <int> <int>   <dbl> <dbl>   <dbl>
 1 twitter   startups    221  9406 0.0235  0     0
 2 twitter   startup     152  9406 0.0162  0     0
 3 twitter   gt          114  9406 0.0121  0.693 0.00840
 4 magazines firms       107  9826 0.0109  0     0
 5 twitter   amp         101  9406 0.0107  0.693 0.00744
 6 magazines private     100  9826 0.0102  0     0
 7 twitter   funding      99  9406 0.0105  0     0
 8 magazines investors    88  9826 0.00896 0     0
 9 twitter   tech         75  9406 0.00797 0     0
10 magazines public       74  9826 0.00753 0     0
# … with 6,450 more rows
> #order descending
> tidy_twitt_tfidf %>%
+   arrange(desc(tf_idf))
# A tibble: 6,460 x 7
   from    word              n total      tf   idf  tf_idf
   <chr>   <chr>         <int> <int>   <dbl> <dbl>   <dbl>
 1 twitter gt              114  9406 0.0121  0.693 0.00840
 2 twitter amp             101  9406 0.0107  0.693 0.00744
 3 twitter impactpartnrs    55  9406 0.00585 0.693 0.00405
 4 twitter simplest         49  9406 0.00521 0.693 0.00361
 5 twitter testosterone     49  9406 0.00521 0.693 0.00361
 6 twitter 2020             44  9406 0.00468 0.693 0.00324
 7 twitter freeag8          41  9406 0.00436 0.693 0.00302
```

```
 8 twitter runwaygrowth     40  9406 0.00425 0.693 0.00295
 9 twitter davidspreng      35  9406 0.00372 0.693 0.00258
10 twitter blockchain       30  9406 0.00319 0.693 0.00221
# … with 6,450 more rows
> #ploting tf-idf
> tidy_twitt_tfidf %>%
+   arrange(desc(tf_idf)) %>%
+   mutate(word=factor(word, levels=rev(unique(word)))) %>%
+   group_by(from) %>%
+   top_n(15) %>%
+   ungroup %>%
+   ggplot(aes(word, tf_idf, fill=from))+
+   geom_col(show.legend=FALSE)+
+   labs(x=NULL, y="tf-idf")+
+   facet_wrap(~from, ncol=2, scales="free")+
+   coord_flip()
Selecting by tf_idf
> #Creating Bigrams
> vc_mag_bigrams <- vc_mag_df %>%
+   unnest_tokens(bigram, text, token = "ngrams", n=2)
> vc_mag_bigrams %>%
+   count(bigram, sort = TRUE)
# A tibble: 15,582 x 2
   bigram              n
   <chr>           <int>
 1 of the             85
 2 in the             75
 3 venture capital    47
 4 to the             44
 5 to be              40
 6 of a               36
 7 it is              35
 8 silicon valley     33
 9 in a               26
10 for the            25
# … with 15,572 more rows
> library(tidyr)
> #magazines bigrams
> bigrams_mag_separated <- vc_mag_bigrams %>%
+   separate(bigram, c("word1", "word2"), sep = " ")
> bigrams_mag_filtered <- bigrams_mag_separated %>%
+   filter(!word1 %in% stop_words$word) %>%
+   filter(!word2 %in% stop_words$word)
> bigram_mag_counts <- bigrams_mag_filtered %>%
+   count(word1, word2, sort = TRUE)
> #want to see the new bigrams
> bigram_mag_counts
# A tibble: 3,430 x 3
   word1   word2       n
   <chr>   <chr>   <int>
```

```
 1 venture capital       47
 2 silicon valley        33
 3 private equity        23
 4 vision  fund          22
 5 private capital       20
 6 public  markets       20
 7 venture capitalists   17
 8 private markets       12
 9 ride    hailing       11
10 sexual  harassment    11
# … with 3,420 more rows
> #twitter bigrams
> vc_twitt_bigrams <- vc_twitter %>%
+   unnest_tokens(bigram, text, token = "ngrams", n=2)
> vc_twitt_bigrams %>%
+   count(bigram, sort = TRUE) #this has many stop words, need to remove them
# A tibble: 6,768 x 2
   bigram                  n
   <chr>               <int>
 1 gt gt                  74
 2 one of                 72
 3 venturecapital clients 61
 4 of our                 58
 5 our venturecapital     57
 6 rt impactpartnrs       55
 7 you think              55
 8 you can                53
 9 blood testing          51
10 when you               51
# … with 6,758 more rows
> #to remove stop words from the bigram data, we need to use the separate function:
> library(tidyr)
> bigrams_twitt_separated <- vc_twitt_bigrams %>%
+   separate(bigram, c("word1", "word2"), sep = " ")
> bigrams_twitt_filtered <- bigrams_twitt_separated %>%
+   filter(!word1 %in% stop_words$word) %>%
+   filter(!word2 %in% stop_words$word)
> #creating the new bigram, "no-stop-words":
> bigram_twitt_counts <- bigrams_twitt_filtered %>%
+   count(word1, word2, sort = TRUE)
> #want to see the new bigrams
> bigram_twitt_counts
# A tibble: 2,790 x 3
   word1          word2            n
   <chr>          <chr>        <int>
 1 gt             gt              74
 2 venturecapital clients         61
 3 rt             impactpartnrs   55
 4 blood          testing         51
 5 accurate       blood           49
```

```
 6 low           testosterone      49
 7 world's        simplest          49
 8 startups       venturecapital    32
 9 venture        capital           31
10 venturecapital funding           27
# … with 2,780 more rows
> library(igraph)
> mag_graph <- bigram_mag_counts %>%
+   filter(n>20) %>%
+   graph_from_data_frame()
> mag_graph
IGRAPH bb9d1ad DN-- 8 4 --
+ attr: name (v/c), n (e/n)
+ edges from bb9d1ad (vertex names):
[1] venture->capital silicon->valley  private->equity  vision ->fund
> library(ggraph)
> ggraph(mag_graph, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
> twitt_graph <- bigram_twitt_counts %>%
+   filter(n>20) %>%
+   graph_from_data_frame()
> ggraph(twitt_graph, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
> #Bigrams
> big_twitt_n <- bigram_twitt_counts %>%
+   filter(n > 12)
> big_mag_n <- bigram_mag_counts %>%
+   filter(n>5)
> big_joined <- bigram_twitt_counts %>%
+   full_join(bigram_mag_counts) %>%
+   filter(n>5)
Joining, by = c("word1", "word2", "n")
> ggraph(big_twitt_n, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
> ggraph(big_mag_n, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
> twitt_graph <- bigram_twitt_counts %>%
+   filter(n>20) %>%
+   graph_from_data_frame()
> ggraph(twitt_graph, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
```

```
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
>
################################################################################
##############################################
> #COVER PAGE
> #Wordclouds for the cover page
> library(wordcloud)
> vc_cloud <- vc_mag_df %>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words) %>%
+   count(word)
Joining, by = "word"
> wordcloud(
+   words = vc_cloud$word,
+   freq = vc_cloud$n,
+   max.words = 45,
+   colors = 'blue',
+   ordered.colors = TRUE
+ )
There were 16 warnings (use warnings() to see them)
> bigram_twitt_counts
# A tibble: 2,790 x 3
   word1          word2              n
   <chr>          <chr>          <int>
 1 gt             gt                74
 2 venturecapital clients           61
 3 rt             impactpartnrs     55
 4 blood          testing           51
 5 accurate       blood             49
 6 low            testosterone      49
 7 world's        simplest          49
 8 startups       venturecapital    32
 9 venture        capital           31
10 venturecapital funding           27
# … with 2,780 more rows
> wordcloud(
+   words = bigram_twitt_counts$word1,
+   freq = bigram_twitt_counts$n,
+   max.words = 45
+ )

> vc_mag_token %>%
+   inner_join(get_sentiments("afinn"))%>%
+   summarise(sentiment=mean(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 1 x 2
  sentiment method
      <dbl> <chr>
1     0.346 AFINN
```

```
> vc_tweet_token %>%
+   inner_join(get_sentiments("afinn"))%>%
+   summarise(sentiment=mean(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
  sentiment method
1 0.9881797  AFINN
```

List of References

- Harvard Business Review (2020). Accessed 7 February 2020. Retrieved from http://hbr.org
- Jaramillo E. (2020). Femtech in 2020: Investors Share Trends And Opportunities In Women's Health Technology. Accessed 7 February 2020. Retrieved from https://www.forbes.com/sites/estrellajaramillo/2020/01/08/femtech-2020-investors-trends-and-opportunities-in-womens-health-technology/#2d0accdb7d54
- Suich Bass A. (2019). Techquake ahead. The Economist. The World in 2020. 121
- The economist (2017). The world's most valuable resource is no longer oil, but data. Accessed 7 February 2020. Retrieved from https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data
- The Economist (2020). Accessed 7 February 2020. Retrieved from http://economist.com
- The Financial Times (2020). Accessed 7 February 2020. Retrieved from http://ft.com
- Tiku Nitasha (2017). VC Firms Promise to Stamp Out Sexual Harassment. Sounds Familiar. Accessed 7 February 2020. Retrieved from https://www.wired.com/story/venture-capital-sexual-harassment-diversity/
- Twitter (2020). Accessed 7 February 2020. Retrieved from http://twitter.com