Accenture Applied Intelligence

TIDY TEXT MINING WORKSHOP R LADIES

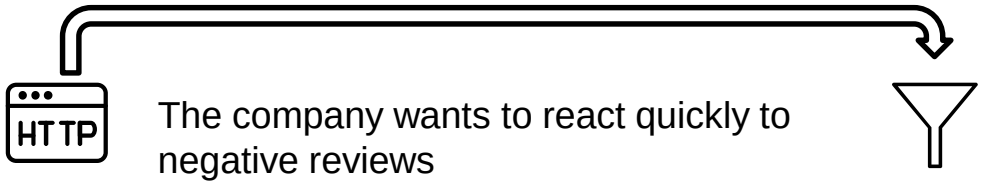
INTRODUCTION INTO TEXT MINING

Process unstructured textual information

Turn text into computer-readable numbers. Summarize, analyze these numbers and use the outcomes as input for predictive data mining or clustering.

Big unstructured text data (takes hours to understand by yourself)

Structured text data, for everyone easy to understand



Reviews on websites	The company wants to react quickly to negative reviews	A list of negative reviews to react on
Newspapers	Get an overview of the main topics of all the newspapers	A trader needs to know quickly what's going on in the world
Social Media	What do social media user think about our prime minister Rutte	Predict the outcome of an election
Emails	Understand the content of the emails. Classify which are important to read first	React first on customers who want to churn

Clean the data
Summarize the data
Analyze the data
Cluster the data

Agenda

- Introduction to the client
- The data set
- After this workshop you are able to:
  - Text mining the tidy way!
  - Extract strings from text files
  - Gather the most common words from text files
  - Bring words back to its stem (e.g. waits, waited → wait) and tag words (wait is a verb)
  - Tips and tricks to clean the data
  - Derive the sentiment of the text
- Translate the insights into business value
- Further examples

THE CASE

What are the strengths and weaknesses of United Airlines compared to other airlines based on social media posts?

The client

Competitors

TWITTER DATA

14640 tweets from 7700 users (February 2015)

tweet_id	text
570306133677760000	@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA
570301130888122000	@VirginAmerica - the passenger in 7D, Flt 338 that assaulted me shouldn't have flown. I trust he's banned. Crew filed report to @FAANews
570301083672813000	@VirginAmerica soooo are you guys going to leave the seatbelt light on all flight? You can barely call this turbulence :-)
570301031407624000	@VirginAmerica I am in seat 4C and I cannot even open my laptop
570300817074462000	@VirginAmerica does Virgin America fly direct from Seattle to NYC or Boston?
570300767074181000	@united crashed trying to check in.
570300616901320000	@united YOU GUYS ARE HORRIBLE.
570300248553349000	@united Could you update me on the suitcase please? The online and phone tracking told me nothing. I was told I'd have it back yesterday!
570299953286942000	@united Really....you charge me $25 to check a bag and then you put it on a different flight....still Don't have my bag!!!

STEP 1: OPEN R-STUDIO & GET STARTED

Checklist

Check if you have Wi-Fi

Type the following code at the beginning of your R script:

```
options(stringsAsFactors = F)
options(scipen = 999)
```

Check if you have the following datasets

Datasets
twitter_airline_sentiment.csv
df_airlines.feather
df_mini.feather
tidy_lemma.feather
tidy_lemma_time.feather

Check if you have to following packages

Library
Library(tidyverse)
Library(tidytext)
Library(udpipe)
Library(feather)

To-do

read in the first csv file

```
"twitter_airline_sentiment.csv"
```

Get a small book with code, it will guide you through the assignments

Use a notebook! They are fun!

THE FEATHER PACKAGE

`Imports:` Rcpp, tibble, hms

Why we use it	Function
1) Load csv files faster, 2) Data format remains the same, 3) Load the data in different programming languages (e.g. python)	read_feather(path, columns = NULL)
	write_feather(x, path)

THE TIDYVERSE PACKAGE

Imports: Broom, cli, crayon, **dplyr,** dbplyr, forcats, **ggplot2,** haven, hms, httr, jsonlite, lubridate, magrittr, modelr, purrr, **readr,** ,readxl\n(>=, reprex, rlang, rstudioapi, rvest, **stringr,** tibble, tidyr, xml2, tidyverse

Package	Why we use it	Function
readr	Load the data	read_csv("namefile.csv")
dplyr	To write in the tidy way	%>%
stringr	Extract strings from the text	str_extract_all(texst,regexp)
ggplot2	Visualize the outcomes	ggplot(aes(x, y))

STEP 2: EXTRACT SPECIFIC STRINGS FROM THE TWEETS BY USING REGEX (A SEQUENCE OF CHARACTERS THAT DEFINE A SEARCH PATTERN)

Tweet
nice rt @virginamerica: vibe with the moodlight from takeoff to touchdown. #moodlitmonday #sciencebehindtheexperience http://t.co/y7o0unxtqp

Load the data:
twitter_airline_sentiment.csv

Use the function:
str_extract_all(text, “regexp”)
From the package tidyverse

Try to extract:
1. All digits
2. The airlines (@)

Tip: Try one tweet first

Time: 10 min

Airline	#	Websites	Tweet
virginamerica	#moodlitmonday #sciencebehindtheexperience	http://t.co/y7o0unxtqp	nice rt @virginamerica: vibe with the moodlight from takeoff to touchdown. #moodlitmonday #sciencebehindtheexperience http://t.co/y7o0unxtqp

Regexp	Matches
\t	tab
\s	Any whitespace
\S	Non whitespace
\d	Any digit
\w	Any word character
[:digit:]	digits
[:alpha:]	letters
[:lower:]	lowercase
[:upper:]	uppercase
[:punct:]	punctuation
[:space:]	space
a+	One or more
(?<=b)a	preceded by

Example: (?<=#)\\w+ → get one or more word characters after the #

THE TIDYTEXT PACKAGE

Imports: Rlang, **dplyr**, **stringr**, hunspell, broom, Matrix, **tokenizers**, janeaustenr, purrr (>= 0.1.1), methods, **stopwords**

Package	Why we use it	Function
tokenizers	Splitting text into tokens	unnest_tokens(input, token = "words”)
tokenizers	Get the sentiment of a text	get_sentiments(lexicon = c("afinn", "bing", "nrc", "loughran"))
stopwords	To remove unnecessary words	get_stopwords(language = "en", source = "snowball")

STEP 3: GET THE MOST COMMONLY USED WORDS

The tidytext package transforms the text into a tidy text format (splitting the text into tokens). A token can be a word (or a n-gram, sentence, paragraph). The tidy text format has a one-token-per-row structure.

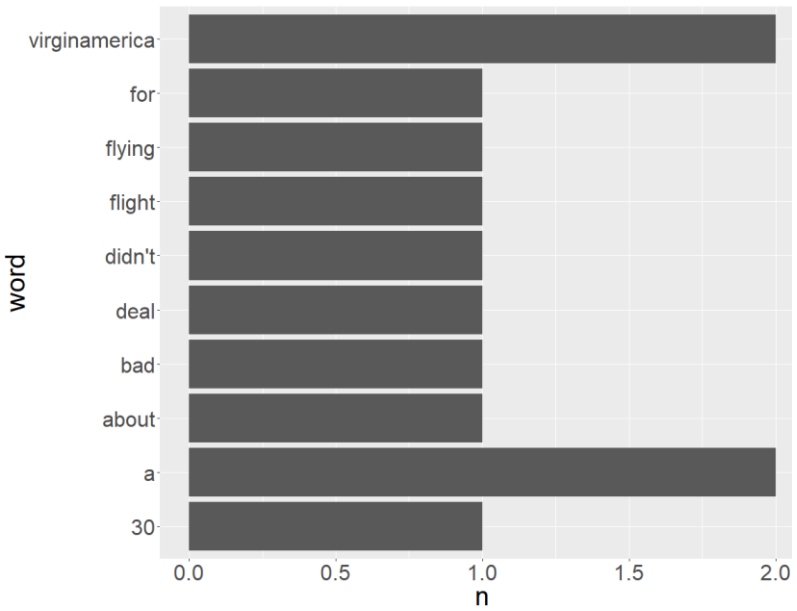
Tweets → unnest_tokens(word, tweet) → Tidytweet → count(word) → Summarized tweet → ggplot2(word,n) → Visualizations

@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA

@virginamerica it is such a great deal!

Token
@virginamerica
seriously
would
pay
$
30
a
flight
for
seats
that
did
n't
have
this
playing
.
it
's
really
the
only
bad
thing
about
flying
va
@virginamerica
it
is
such
a
great
deal
!

Token	Count of token
@virginamerica	2
a	2
this	2
really	1
n't	1
such	1
!	1
pay	1
$	1
seats	1
the	1
.	1
va	1
About	1
30	1
...	...



Load in the data:
df_airlines.feather

Use the package tidytext

- What are the top 10 most frequent words?
- What are the top 10 most frequent words per airline

Visualize your outcomes!
Time: 10 min

ALL THE INFLECTED FORMS OF A WORD ARE NOT INFORMATIVE EITHER

Token	Count of token
Fly	1200
Flying	1000
Flies	300
flown	2500
Sit	59
Sits	1
Seated	100

Token	Count of token
Fly	5000
Sit	160

STEP 4: LEMMATIZATION & TAGGING

Lemmatization: grouping together the inflected forms of a word

@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA

Token	Lemma	Tagging
@virginamerica	@virginamerican	AUX
seriously	seriously	ADV
would	would	AUX
pay	pay	VERB
$	$	SYM
30	30	NUM
a	a	DET
flight	flight	NOUN
for	for	ADP
seats	seat	NOUN
that	that	PRON
did	do	AUX
n't	not	PART
have	have	VERB
this	this	DET
playing	Play	NOUN
.	.	PUNCT
it	it	PRON
's	be	AUX
really	really	ADV
the	the	DET
only	only	ADJ
bad	bad	ADJ
thing	thing	NOUN
about	about	SCONJ
flying	fly	VERB
va	va	NOUN

Universal Pos Tags (Upos)	Definition	Examples
ADJ	adjective	big, old, green
ADP	adposition	in, to, during
ADV	adverb	very,well, exactly, tomorrow, up, down
AUX	auxiliary	has, is, will, was, got, should, can
CCONJ	coordinating conjunction	and, or, but
DET	determiner	this, which, the
INTJ	interjection	psst, ouch, bravo, hello
NOUN	noun	girl, cat, air
NUM	numeral	0, 1, 2, three
PART	particle	‘s, n't
PRON	pronoun	I, you, he, she, we, they, what, my
PROPN	proper noun	Mary, John, America
PUNCT	punctuation	.?!
SCONJ	subordinating conjunction	that, if, while
SYM	symbol	$, #, ;)
VERB	verb	run, eat, fly
X	other	dwqjdwo dwqo

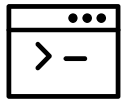
THE UDPIPE PACKAGE

Imports: Rcpp (>= 0.11.5), data.table (>= 1.9.6), Matrix, methods

Why we use it	Function
Choose a language and download it (52 different languages)	udpipe_download_model(language = c("afrikaans", "ancient_greek-proiel", "ancient_greek", "arabic", "basque", "belarusian", "bulgarian", "catalan", "chinese", "coptic", "croatian", "czech-cac", "czech-cltt", "czech", "danish", "dutch-lassysmall", "dutch", "english-lines", "english-partut", "english", "estonian", "finnish-ftb", "finnish", "french-partut", "french-sequoia", "french", "galician-treegal", "galician", "german", "gothic", "greek", "hebrew", "hindi", "hungarian", "indonesian", "irish", "italian", "japanese", "kazakh", "korean", "latin-ittb", "latin-proiel", "latin", "latvian", "lithuanian", "norwegian-bokmaal", "norwegian-nynorsk", "old_church_slavonic", "persian", "polish", "portuguese-br", "portuguese", "romanian", "russian-syntagrus", "russian", "sanskrit", "serbian", "slovak", "slovenian-sst", "slovenian", "spanish-ancora", "spanish", "swedish-lines", "swedish", "tamil", "turkish", "ukrainian", "urdu", "uyghur", "vietnamese")
To lemmatize and tokenize	udpipe_annotate(object, x, doc_id = paste("doc", seq_along(x), sep = ""), tokenizer = "tokenizer", tagger = c("default", "none"), parser = c("default", "none"), trace = FALSE, ...)

NOW TRY IT YOURSELF!

Use the “udpipe” package



Use the following code:

```
# load the tagging models
dl <- udpipe_download_model(language = "english")
udmodel_english <- udpipe_load_model(file = "english-ud-2.0-
170801.udpipe")

# load the data
df_mini <- read_feather("df_mini.feather")
df_mini$text[1]

# do for one tweet
lemma_example <- udpipe_annotate(udmodel_english,  x = df_mini$text[1],
doc_id =             df_mini$tweet_id[1], parser = "none", tagger =
"default ", trace = FALSE)
str(lemma_example)

# in a tidy format and add airline
lemma_example <- as.data.frame(lemma_example) %>% mutate(airline =
df_mini$airline[1])

View(lemma_example)
```

Experiment with 7 tweets: which Upos are good to keep, which are less important. Clean up the data as good as possible!

Time: 15 min

Universal Pos Tags (Upos)	Definition	Examples
ADJ	adjective	big, old, green
ADP	adposition	in, to, during
ADV	adverb	very,well, exactly, tomorrow, up, down
AUX	auxiliary	has, is, will, was, got, should, can
CCONJ	coordinating conjunction	and, or, but
DET	determiner	this, which, the
INTJ	interjection	psst, ouch, bravo, hello
NOUN	noun	girl, cat, air
NUM	numeral	0, 1, 2, three
PART	particle	‘s, n't
PRON	pronoun	I, you, he, she, we, they, what, my
PROPN	proper noun	Mary, John, America
PUNCT	punctuation	.?!
SCONJ	subordinating conjunction	that, if, while
SYM	symbol	$, #, ;)
VERB	verb	run, eat, fly
X	other	dwqjdwo dwqo

STEP 5: VISUALIZE THE CLEANED DATA

Look again at the commonly used words, do you see any improvement?

Data: tidy_lemma.feather

Open the cleaned data set and test whether the top most common 10 words (per airline) are more informative now. Visualize your outcomes!

Time: 5 min

Remember to use the following functions from the package tidytext:

unnest_tokens(word, tweet) → count(word) → ggplot2(word,n)

SENTIMENT ANALYSIS

Approach the emotional content of a text programmatically

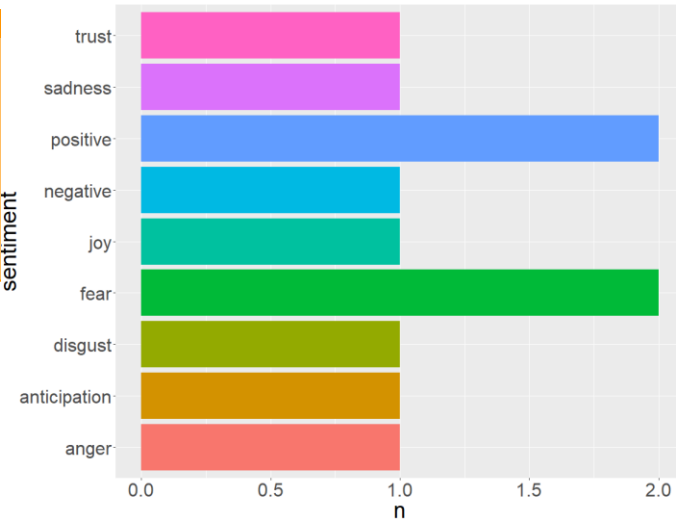
Tweets → unnest_tokens() → Tidytweet → get_sentiments() / Inner_join() → Lexicon → count() → Summarized tweet → ggplot2() → Visualizations

@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA

Token
@virginamerica
seriously
would
pay
$
30
a
flight
For
seats
that
did
n't
have
this
playing
.
it
's
really
the
only
bad
thing
about
flying
va

Token	Sentiment
@virginamerica	Neutral
seriously	Neutral
would	Neutral
pay	Aticitpation, trust,positive
$	Neutral
30	Neutral
a	Neutral
flight	Neutral
For	Neutral
seats	Neutral
that	Neutral
did	Neutral
n't	Neutral
have	Neutral
this	Neutral
playing	Neutral
.	Neutral
it	Neutral
's	Neutral
really	Neutral
the	Neutral
only	Neutral
bad	Anger, disgust, fear,negative,sadness
thing	Neutral
about	Neutral
flying	Fear, positive
va	Neutral

Sentiment	Count of sentiment
Positive	2
Sadness	1
Trust	1
Negative	1
Joy	1
Fear	2
Disgust	1
Anticipation	1
Anger	1



Different kind of lexicons:

1) Afinn (Finn Arup Nielsen)
2) Bing (Bing Liu and collabraters)
3) NRC (Saif Mohammed and Peter Turney)

STEP 6: FIND THE SENTIMENT OF THE TWEETS!

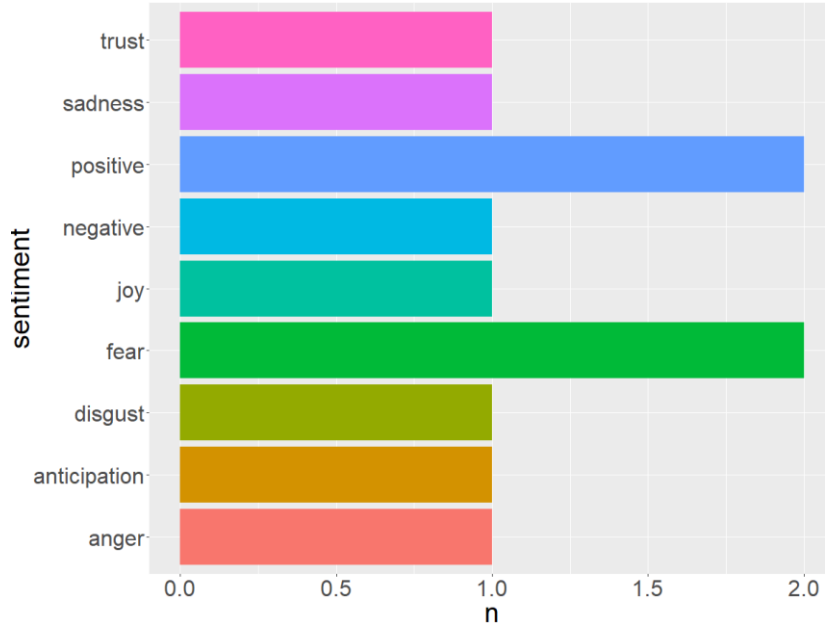
Use the code below

Load the data: tidy_lemma.feather

Use the package tidytext.

For every airline; What is % of positive and negative sentiment with the NRC dictionary. Visualize your outcomes!

Time: (10 min)




```
tidy_tweet <- data_set %>%
  select(text)%>%
  unnest_tokens(word, text)
```

→

```
sentiments <- get_sentiments("nrc") %>%
  inner_join(tidy_tweet)
```

→

```
sentiments %>%
  count(sentiment,)%>%  ggplot(aes(sentiment, n, fill =
sentiment))+  geom_col(show.legend = F)+  coord_flip
```


SENTIMENT ANALYSIS

Approach the emotional content of a text programmatically

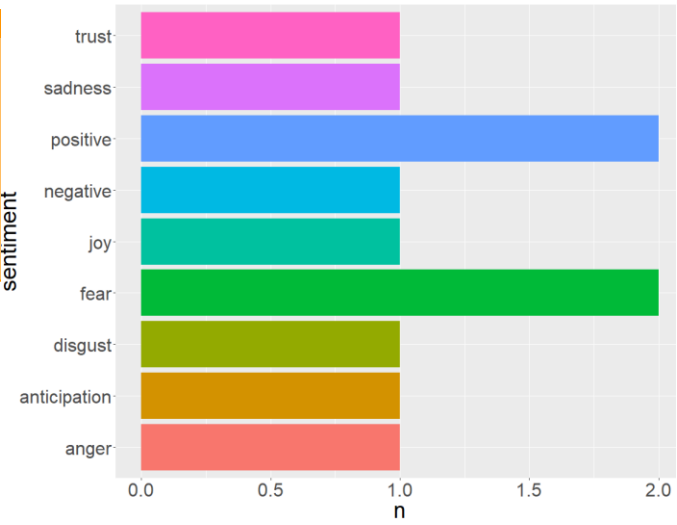
Tweets → unnest_tokens() → Tidytweet → get_sentiments() / Inner_join() → Lexicon → count() → Summarized tweet → ggplot2() → Visualizations

@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA

Token
@virginamerica
seriously
would
pay
$
30
a
flight
For
seats
that
did
n't
have
this
playing
.
it
's
really
the
only
bad
thing
about
flying
va

Token	Sentiment
@virginamerica	Neutral
seriously	Neutral
would	Neutral
pay	Aticitpation, trust,positive
$	Neutral
30	Neutral
a	Neutral
flight	Neutral
For	Neutral
seats	Neutral
that	Neutral
did	Neutral
n't	Neutral
have	Neutral
this	Neutral
playing	Neutral
.	Neutral
it	Neutral
's	Neutral
really	Neutral
the	Neutral
only	Neutral
bad	Anger, disgust, fear,negative,sadness
thing	Neutral
about	Neutral
flying	Fear, positive
va	Neutral

Sentiment	Count of sentiment
Positive	2
Sadness	1
Trust	1
Negative	1
Joy	1
Fear	2
Disgust	1
Anticipation	1
Anger	1



Data: all dataset can be used!

(For every airline); what are their strengths and weaknesses ? Visualize your outcomes!

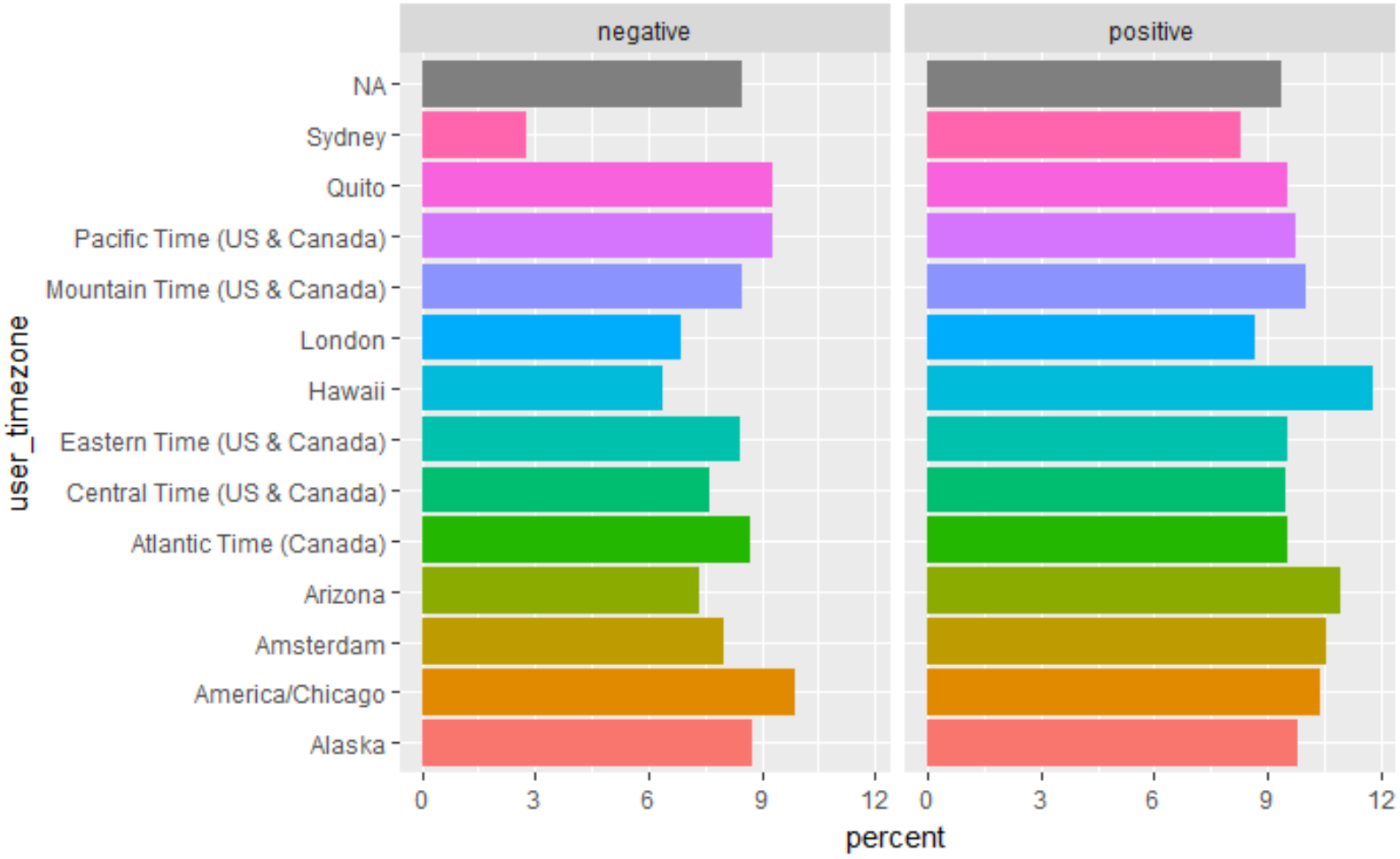
Time: (25 min)

<u>Different kind of lexicons:</u>

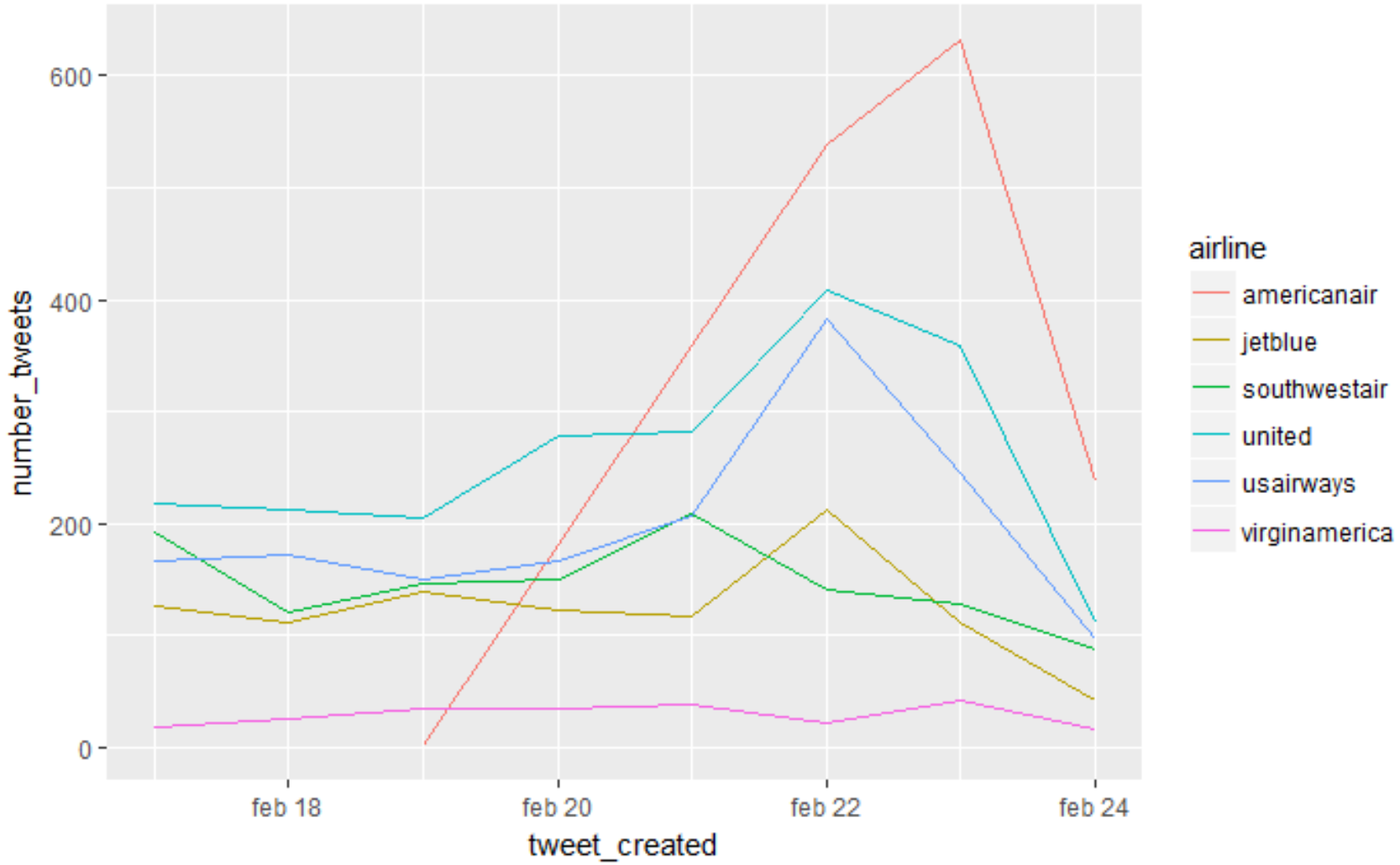
1) Afinn (Finn Arup Nielsen)
2) Bing (Bing Liu and collabraters)
3) NRC (Saif Mohammed and Peter Turney)

FURTHER EXAMPLES

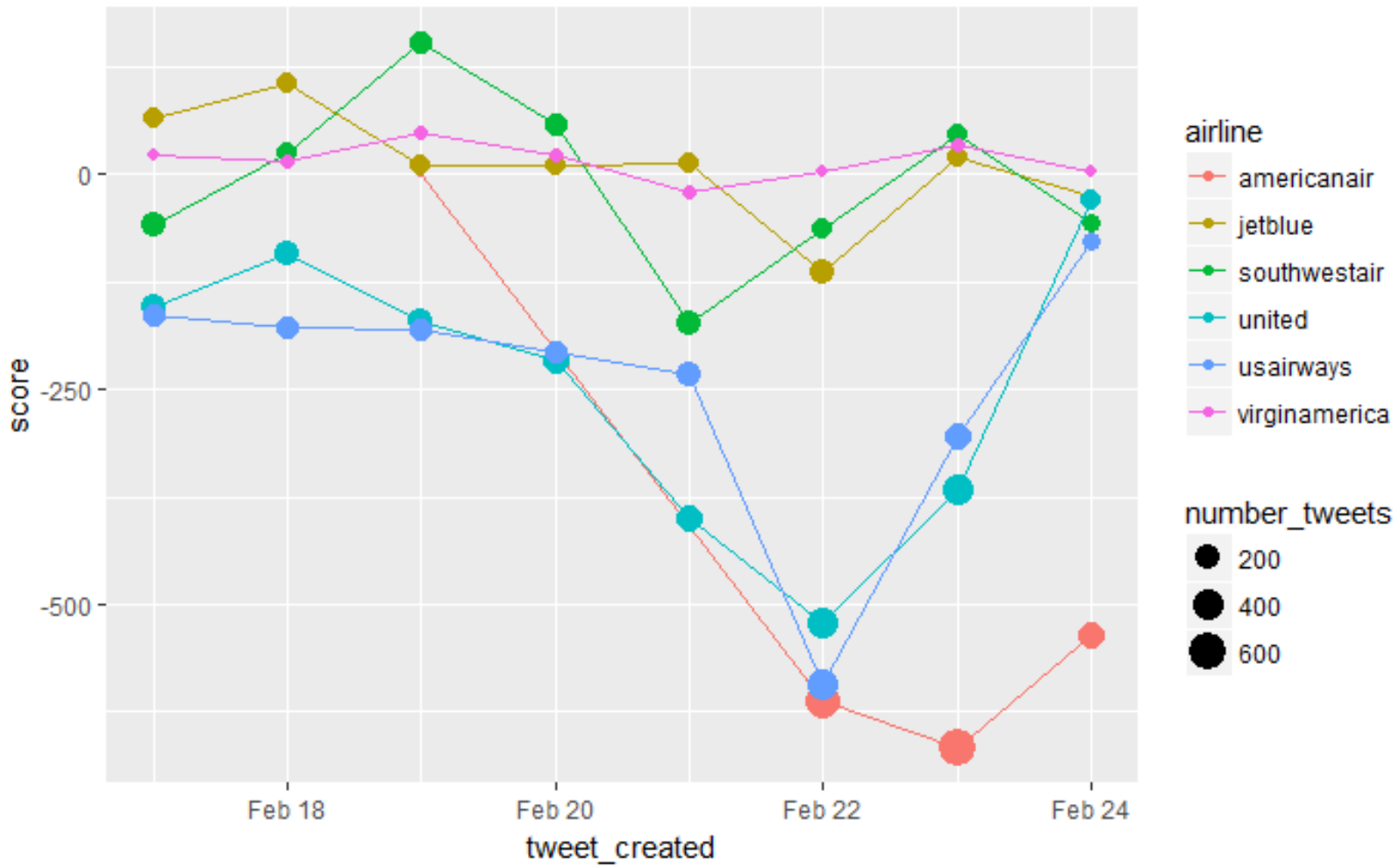
SENTIMENT ANALYSIS OUTCOME (USER TIMEZONE)



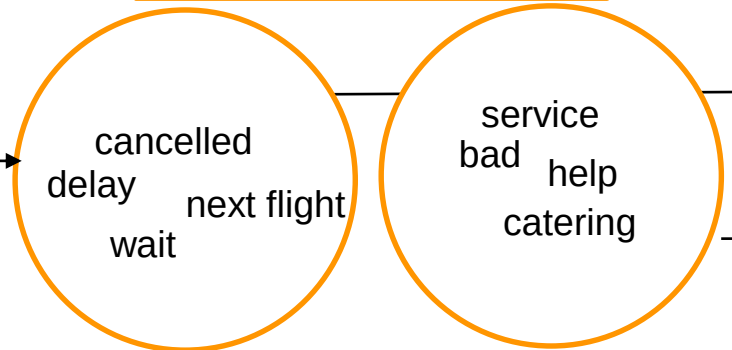
NUMBER OF TWEETS OVER TIME (PER AIRLINE)



SENTIMENT OF TWEETS OVER TIME (PER AIRLINE)



NEXT STEPS

N-grams, analyses groups of words instead of single words	Flight cancelled, delay hour, which gate	More information than one single word
Tf-idf: calculate the relative importance of a word, n-gram compare to the other words within the document and corpus	“Bad customer service” more important than “I ate a salad”	Retaining the core and most important meaning of the document
Word2vec: Make a vector representations of each word by the use of neural nets	Word: “dirty” (filthy, unclean, stained, dusty, disgusting..)	Make your own sentiment lexicon
Prediction: Use the words, n-grams as input for predictive modelling.	Tweet from Justin Bieber: @United be ready I will fly to the Bahamas next week!	Higher the prices of the flight, because demand of the ticket to Bahamas will rise
Topic modelling: Use unsupervised clustering to find the underlying topics within your documents		Topic 1: Problems with a flight Topic 2: Customer service

THANK YOU FOR YOUR ATTENTION!
QUESTIONS?

Any questions left?

Feel free to contact us!

Eline Tjeng
Data scientist

+31 620570147

Eline.tjeng@
accenture.com

Laury van Bedaf
Data scientist

+31 682984506

Laury.van.bedaf@
accenture.com

