Speech-driven graphs

The easy way to “wreck a nice beach” with R

Rainer Hillermann
Chief Data Scientist at Blue Guava Predictive Analytics
rainer@blueguava.com.au

Overview

- Intro
- Speech Systems
- The R + Shiny + annyang system:
  file structure, javascript file, Shiny App
- Demo

Caveats!

- No idea how it will work with a microphone

Who am I?

- Chief Data Scientist at Blue Guava Predictive Analytics, doing R&D in Deep Learning, natural language processing and financial markets with Python and R.

- Delivered Business, IT and Analytics projects around Australia, Asia, India and the Middle East in Telco, Media, Construction, Retail, Digital, Mining and Oil & Gas industries.

- Loves probability, cloud computing, application over theory and open source.

Speech Recognition

Definition: *The ability to recognise and translate spoken language into text.*

Speech Recognition

Definition: *The ability to recognise and translate spoken language into text.*

Characteristics

- Complex problem - speaker variance by accent, pronunciation, pitch, volume, speed
- Very hard to achieve human-level accuracy

Speech Recognition

Definition: *The ability to recognise and translate spoken language into text.*

Characteristics

- Complex problem - speaker variance by accent, pronunciation, pitch, volume, speed
- Very hard to achieve human-level accuracy

Why?

- Environmental factors: background noise, acoustics, multiple speakers
- Speaker dependent (need to train for the speaker)

Speech Recognition

Definition: *The ability to recognise and translate spoken language into text.*

Characteristics

- Complex problem - speaker variance by accent, pronunciation, pitch, volume, speed
- Very hard to achieve human-level accuracy

Why?

- Environmental factors: background noise, acoustics, multiple speakers
- Speaker dependent (need to train for the speaker)

Levels of difficulty

- Discrete words = easy
- Connected words (+ clear articulation) = medium
- Continuous natural speech with large vocabulary = EXTREME

What has changed?

- Deep Learning in Google, Baidu, Microsoft, IBM, Nuance and other companies over the last 5-6 years, accelerating in the last 3 years.

- Latest advances enabled by deep recurrent, Long Short Term Memory (LSTM) networks with character-level embedding and even ignoring phonemes!

My Motivation

Why should one have to learn some technical language or product to be able to ask questions of and get insights from data?

- Excel
- R
- Python
- SQL
- Hadoop
- Tableau

My Motivation

Why should one have to learn some technical language or product to be able to ask questions of and get insights from data?

- Excel
- R
- Python
- SQL
- Hadoop
- Tableau

Business people care about business, not report-building, programming languages, scripts or algorithms!

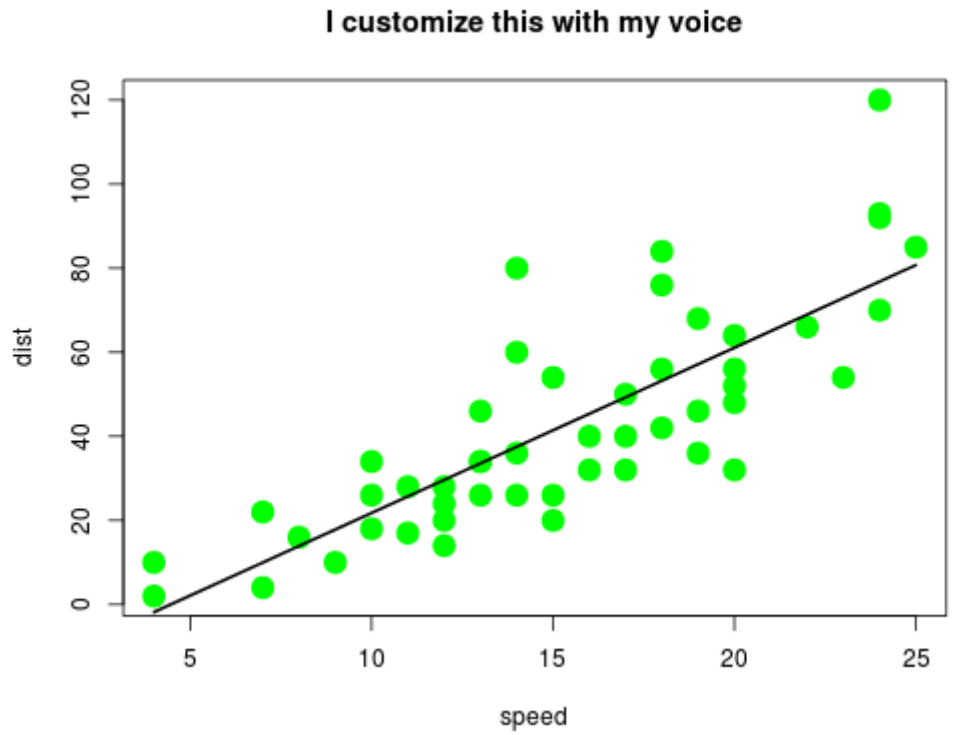
My Use Case

- I want to:

    1. See visualisations of stock prices (using chartSeries)
    2. Interact with my visualisations (with my voice)
    3. See new visualisations that give me what I want (Shiny)

The Inspiration

- Basic example by Yihui Xie#



http://blog.revolutionanalytics.com/2015/01/talk-to-r.html

The Technology

- **Rstudio**: Open source IDE for R
- **Shiny**: interactive visualisations with R
- **Annyang**: speech recognition javascript library wrapped around Google's Web Speech API
- **Chrome browser**: supports Web Speech API
- **Microphone**: karaoke input device

The Limitations

- 1 word = 1 action
- Only 1 action at a time
- Overall state is not saved
- Localisation not for Aussies!
  - *color* does not equal *colour*

My Desired Improvements

	Yihui's example	My enhancements
Speech input	1 command per utterance	1+ command per utterance
Data	1 fixed dataset	Any financial time-series
Natural Language Processing	Hard-coded to direct match in voice input	Regular expression from a string
Visualisation	Default plot function	chartSeries
State	Resets on every command	Updates current graph + can reset

The System


Me
Annyang
ShinyApp
Browser
Create Speech
Speech
Text
Clean up text
Match Conditions
Add features
Define plot
Save Settings
Present plot

Files

Define our speech commands and what gets passed to Shiny

init.js
app.R

Data/

- Config.csv
- Config2.csv

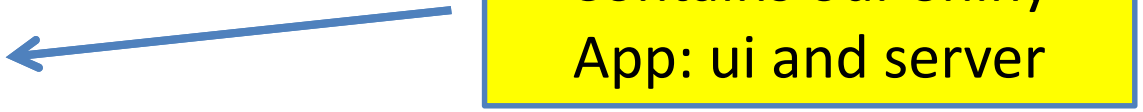
Data/WIKI/

- AAPL.csv
- AMZN.csv
- BIDU.csv
- GOOGL.csv

annyang.min.js: from within ShinyApp UI

Files

init.js
app.R



Data/

- Config.csv
- Config2.csv

Data/WIKI/

- AAPL.csv
- AMZN.csv
- BIDU.csv
- GOOGL.csv

annyang.min.js: from within ShinyApp UI

Files

init.js
app.R

Data/
- Config.csv
- Config2.csv

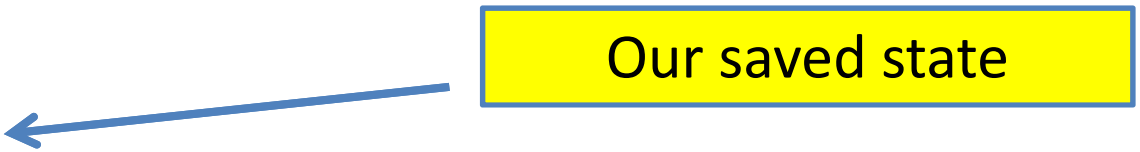
Data/WIKI/
- AAPL.csv
- AMZN.csv
- BIDU.csv
- GOOGL.csv

annyang.min.js: from within ShinyApp UI

Files

init.js
app.R

Data/
- Config.csv
- Config2.csv



Data/WIKI/
- AAPL.csv
- AMZN.csv
- BIDU.csv
- GOOGL.csv

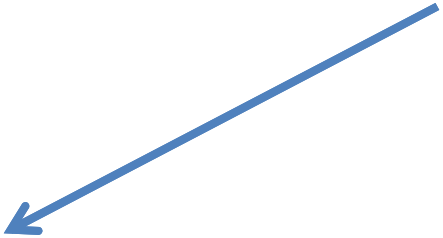
annyang.min.js: from within ShinyApp UI

Files

init.js
app.R

Data/
- Config.csv
- Config2.csv

Data/WIKI/
- AAPL.csv
- AMZN.csv
- BIDU.csv
- GOOGL.csv



Example data for stocks with fields:

Date
Adjusted Open
Adj. High
Adj. Low
Adj. Close
Adj.Volume

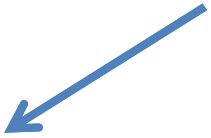
annyang.min.js: from within ShinyApp UI

Files

init.js
app.R
helpers.R

Data/
- Config.csv
- Config2.csv

Data/WIKI/
- AAPL.csv
- AMZN.csv
- BIDU.csv
- GOOGL.csv



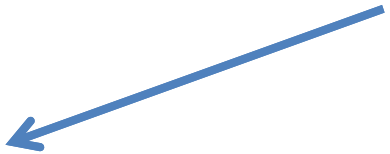
Wrapper around Web Speech API

annyang.min.js: from within ShinyApp UI

init.js file

```
if (annyang) {

  var commands = {
    'now *text': function(text) {
      Shiny.onInputChange('text', text);
    }
  };
  annyang.addCommands(commands);
  annyang.start();
  };
```



After I say “now”, my subsequent speech will be turned to text

init.js file

```
if (annyang) {

  var commands = {
    'now *text': function(text) {
      Shiny.onInputChange('text', text);
    }
  };
  annyang.addCommands(commands);
  annyang.start();
  };
```

After we say “now”, the rest of our speech will be turned to text

And passed back to the Shiny server as the variable “text”

Our R code

Load our libraries

- **stringr** for string manipulation
- **tm** for text mining
- **quantmod** for chartSeries visualisation
- **data.table** to load our data
- **xts** for time-series
- **TTR** for our technical indicators

Our R code

- **setwd(): set our working directory**
- **1 function:**

```
import.stock <- function(datasource){

  file <- paste(datasource, ".csv", sep = "")
  data <- as.data.frame(fread(file))

  names(data) <- c("datetime", "open", "high", "low", "close", "volume")
  data$datetime <- as.Date(strptime(data$datetime, "%Y-%m-%d"))
  data <- as.xts(data[,2:6], order.by = data[,1])

  return (data)
}
```

Anatomy of a Shiny App

UI + **Server**

How it looks

- **The browser**
- **HTML/Javascript**
- **The graph we show**

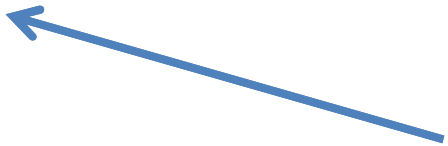
The logic

- **recomputes when we speak**
- **text processing**
- **data manipulation**
- **what features we want to represent**

Our R code

- **Shiny App (UI):**

```
shinyApp(
  ui = fluidPage(
    singleton(tags$head(
      tags$script(src="//cdnjs.cloudflare.com/ajax/libs/annyang/1.4.0/annyang.min.js"),
      includeScript('init.js')
    )),
    div(
      style = 'display: block; margin: auto; width: 100%; max-width: 1000px;',
      plotOutput('plot', height = '700px', width = '100%')
    )
  ),
```



The type of output we want as a variable called ‘plot’

Our R code

Shiny App (server):

1. Load **config.csv** and set default parameters
2. **reactive(input$text)** function receives text from speech.
3. **output$plot** is the output, using the values from our renderPlot() function

Our R code

Shiny App (server):

A reactive function is recalculated whenever the value changes, ie. our input text every time we speak

1. Load **config.csv** and
2. **reactive(input$text)** function receives text from speech.
3. **output$plot** is the output, using the values from our renderPlot() function

Demo Time!

Improvements/Limitations

- ggplot would be better to extend plots
- chartSeries not ideal for use with Shiny (multiple updates to the one plot can't easily be done in a single update)
- Haven't saved states of all parameters
- Could be extended to download and present any financial instrument (eg. from Quandl)
- Some bugs with subsetting data

Resources

- https://www.rstudio.com/
- http://shiny.rstudio.com/
- https://www.talater.com/annyang/
- https://www.quandl.com/

Thanks!

Rainer Hillermann

Chief Data Scientist at
Blue Guava Predictive Analytics
rainer@blueguava.com.au