Network of bigrams

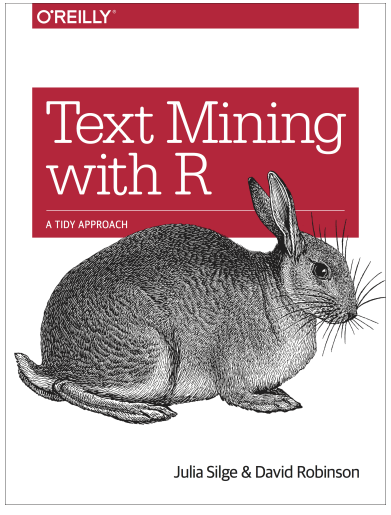
Rei Sanchez-Arias, Ph.D.

Tokenization by n-grams, and graph representation with the `igraph` package

Resources and packages

Book and packages

"Text Mining with R: A Tidy Approach"
by
Julia Silge and
David Robinson



☑ Load the `tidyverse` and `tidytext` packages

```
library(tidyverse)
library(tidytext)
```

The `igraph` package has many powerful functions for manipulating and analyzing networks. `igraph` is a collection of network analysis tools with the emphasis on efficiency, portability and ease of use. `igraph` is open source and free.

The `ggraph` package is an extension of the `ggplot2` API tailored to graph visualizations and provides the same flexible approach to building up plots layer by layer.

- Install it now from your R console: `install.packages("igraph")` , `install.packages("ggraph")`

Example: posts on Florida Poly news website

```
polynews <- read_csv("https://raw.githubusercontent.com/reisanar/datasets/mast
```

	news_date	news_titles	news_summary
1	2020-09-24	Innovative health device takes Florida Poly students to statewide competition finals	Florida Polytechnic University seniors Megan Morano and Ethan Medjuck recently competed as finalists in the Florida Blue Health Innovation Challenge.
2	2020-09-03	Internship with multinational software company blossoms into ongoing opportunity	A summer internship with multinational software company Red Hat turned into a long-term remote work position for Florida Polytechnic University senior Isabel Zimmerman.
3	2020-08-27	Data science student creates nanoHUB tool to enhance remote education	Cindy Nguyen, a senior data science student at Florida Polytechnic University, recently completed an Undergraduate Research Experience (URE) with the Network for Computational Nanotechnology (NCN) at Purdue University. As part of the URE program, she developed an interactive learning tool for scientific computing and data analysis applications in science and engineering.

Motivation

We may be interested in visualizing all of the relationships among words simultaneously, rather than just the top few at a time. As one common visualization, we can arrange the words into a network, or **graph**. Here we refer to a graph not in the sense of a visualization, but as a *combination of connected nodes*. A graph can be constructed from a tidy object since it has three variables

- `from`: the node an edge is coming from
- `to`: the node an edge is going towards
- `weight`: A numeric value associated with each edge

The `igraph` package

The `igraph` package has many powerful functions for manipulating and analyzing networks. `igraph` is a collection of network analysis tools with the emphasis on efficiency, portability and ease of use. `igraph` is open source and free.

One way to create an `igraph` object from tidy data is the `graph_from_data_frame()` function, which takes a data frame of edges with columns for `"from"`, `"to"`, and edge attributes - in this case `n`:

```
library(igraph)
```

Tokenization:

```
token = "ngrams"
```

Tokenization

First, perform tokenization and remove stop-words:

```
poly_filtered <- polynews %>%
    unnest_tokens(bigram, news_summary, token = "ngrams", n = 2) %>%
    separate(bigram, into = c("word1", "word2"), sep = " ") %>%
    filter(!word1 %in% stop_words$word, !is.na(word1)) %>%
    filter(!word2 %in% stop_words$word, !is.na(word2))
```

We then find the most common combination of words:

```
poly_counts <- poly_filtered %>%
  count(word1, word2, sort = TRUE)
poly_counts %>%
  filter(n > 5)
```

```
## # A tibble: 3 x 3
##   word1       word2           n
##   <chr>       <chr>       <int>
## 1 florida     polytechnic    25
## 2 polytechnic university     24
## 3 covid       19              8
```

Building the graph

Bigrams graph

Build graph from data frame:

```
# filter for only relatively
# common combinations
bigram_graph <- poly_counts %>%
  filter(n > 1) %>%
  graph_from_data_frame()
```

```
bigram_graph
```

```
## IGRAPH e46165f DN-- 28 18 --
## + attr: name (v/c), n (e/n)
## + edges from e46165f (vertex names):
##  [1] florida    ->polytechnic polytechn
##  [3] covid      ->19          universit
##  [5] computer   ->science     florida
##  [7] lakeland   ->fla         19
##  [9] 2020       ->semester    academic
## [11] assistant  ->professor   business
## [13] data       ->science     honor
## [15] latin      ->american    science
## + ... omitted several edges
```

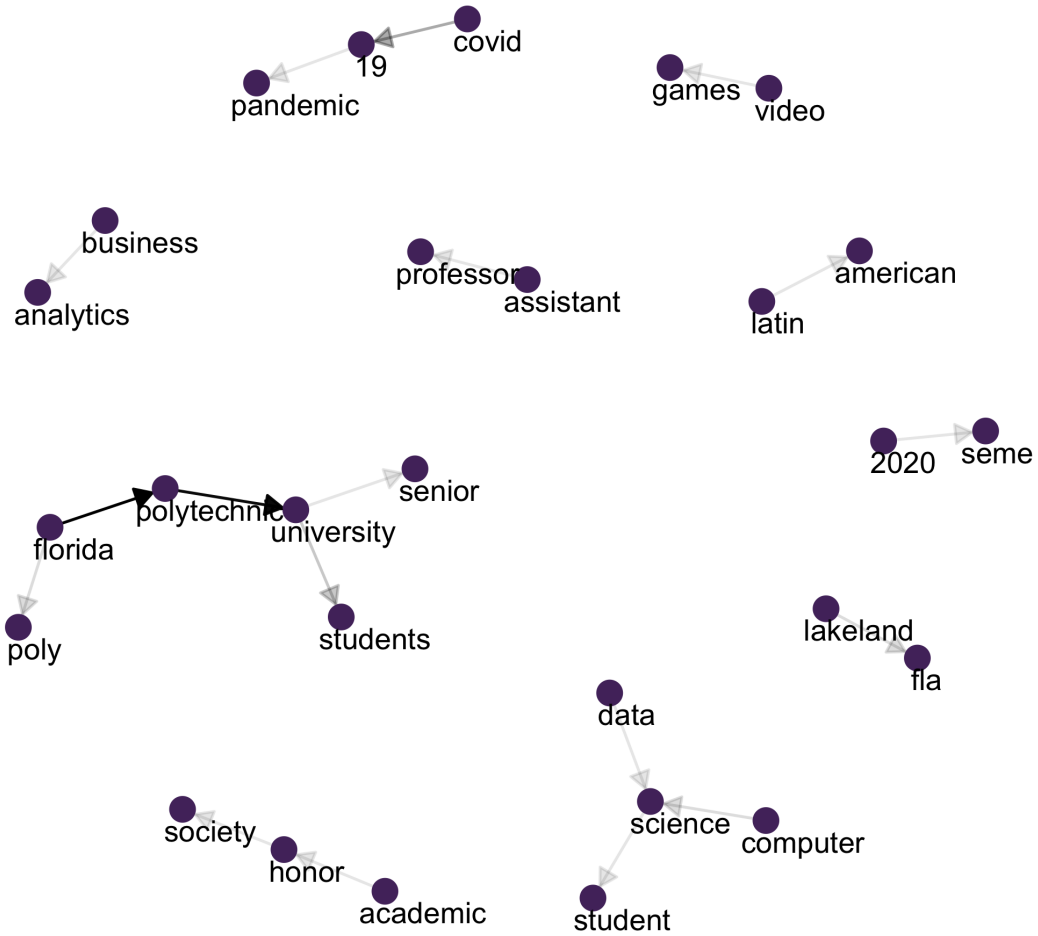
Network of bigrams visualization

`igraph` has plotting functions built in, but they are not what the package is designed to do, so many other packages have developed visualization methods for graph objects.

We recommend the `ggraph` package (Pedersen 2017), because it implements these visualizations in terms of the *grammar of graphics*, which we are already familiar with from `ggplot2`.

```
library(ggraph)
```

```
set.seed(310)
# set arrow
a <- grid::arrow(
  type = "closed",
  length = unit(.10, "inches"))
# plot graph
ggraph(bigram_graph, layout = "fr") +
  geom_edge_link(
    aes(edge_alpha = n),
    show.legend = FALSE, arrow = a,
    end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "#53316B",
                  size = 4) +
  geom_node_text(aes(label = name),
                 vjust = 1.5, hjust = 0.2)
  theme_void()
```



- use `edge_alpha` aesthetic to the link layer to make links transparent based on how common or rare the bigram is.
- add directionality with an arrow, constructed using `grid::arrow()`, including an `end_cap` option that tells the arrow to end before touching the node.