TopicAnalysis

Julio Portella

8/8/2021

# Topic analysis

The goal in this task is to create an algorithm that classifies abstracts into a topic. Indeed, your goal is to group abstracts based on their semantic similarity. Given that this is an unsupervised learning task we can set the number of topics at convenience but not the type of topics.

## Solution description

This is a NLP topic modelling task. For this case, most of the libraries are written in R and even if I have some personal bias towards Python. I prefer to work with R in this case. The process is summarized in the following

**1.** Data extraction

**2.** Data frame creation

**3.** Natural language processing functions

**4.** Topic analysis

**5.** Conclusion

## Data Extraction

As standard procedure, the first thing to do is to import the libraries

library(xml2)  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.3 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidyverse) # general utility & workflow functions  
library(tidytext) # tidy implimentation of NLP methods  
library(topicmodels) # for LDA topic modelling   
library(tm) # general text mining functions, making document term matrixes

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(SnowballC) # for stemming  
library(tm)  
library(tmap)

The first thing done was to get an abstract from a file

## [1] "This is the abstract"

## [1] "Head and heart development are closely intertwined during embryonic development in vertebrates. They share molecular regulatory mechanism as well as some progenitor cell populations. Both head and heart, are made from a multitude of cells, which include mesodermal cells that form musculature and neural crest cells which form connective tissue. Mesodermal and neural crest cells are also important for the development of cartilages and bones in head and neck and for proper septation of the heart. The identity of cells that constitute these tissues is specified very early during embryogenesis via a specific set of genes (specifiers), then follow their trajectory to become muscle, connective tissue, etc.<br/>One early specifier, gastrulation-brain-homeobox 2 (Gbx2), is essential for the migration (movement) and survival of neural crest cells. Neural crest cells and mesodermal cells interact during their migration to form head and heart structures. While mesodermal cells are not directly effected by change in Gbx2 expression, neural crest cells are effected and will alter the communication with mesodermal cells, leading to changes in mesoderm derived structures (e.g., muscles). This project aims to understand how Gbx2 guides neural crest development, and how changes in Gbx2 expression influences head (muscle, cartilage, bone, cranial nerve) and heart development. <br/>Traditionally the fields of comparative, developmental and evolutionary biology have been unsuccessful in recruiting students and scientists of diverse backrounds. This project will promote collaborations between researchers from non-traditional backgrounds in these areas of biology and historically black universities, Howard University and University of the District of Columbia. This award is funded by the NSF Excellence in Research Program.<br/><br/><br/>Technical paragraph<br/>Cranial neural crest cells (CNCCs) are involved in the development of cranial ganglia, cranial nerves, cranium, connective tissue of head muscles, heart septation, pharyngeal arch artery development, etc. The gastrulation-brain-homeobox (Gbx) transcription factor family member, Gbx2, regulates diverse developmental processes, including anteroposterior patterning within the anterior hindbrain and migration and survival of CNCCs. Since CNCCs are involved in many different developmental processes during head and heart development, this project aims to analyze target genes under direct control of Gbx2, that regulate migration and survival of CNCCs in homozygous Gbx2neo embryos. Furthermore, this project will include analyses of the impact of changes of Gbx2 expression on the morphogenesis of the neural system, and craniofacial and cardiovascular structures. The latter part is of significance as neural crest cells influence the development and differentiation of surrounding tissues and vice versa. During head and haert development the interaction of CNCCs and mesoderm is of particular interest for this project because the mesodermal progenitor cells for most of the head and heart musculature derives from a common progenitor field, called the cardiopharyngeal field. In case Gbx2 is somehow involved in the gene regulatory network underlying the differentiation of that mesodermal progenitor field, directly via altering the transcription of genes or indirectly via altering the neural crest - mesodermal interaction during development, this project will lead to insights into mechanisms regulating head and heart development as well as cranial ganglia and nerve development.<br/><br/>This award reflects NSF's statutory mission and has been deemed worthy of support through evaluation using the Foundation's intellectual merit and broader impacts review criteria."

Now that we know that we can get an abstract we proceed to the dataframe creation

## Data frame creation

Let’s ge all the xml files from the folder and put it into a variable

filenames <- Sys.glob("D:/Globant/Task1/data/\*.xml")

The next thing to do is to load everything into a dataframe. This dataframe will have the address of the file and teh abstract. Since there are some files that have no abstract, they will have a value that will help to remove them later. This code chunk takes a while to run

The next process is to clean the dataset from the elements that have no abstract. Since the files and the abstract have the same text if there’s no abstract, we can remove in a simple way with the following code

Now that the DataFrame is cleared, we need to pre process the data. We have to remove stopwords, and apply different natural language processing algorithms to get the important words that will allow us to extract the topics. In this case, I added a couple of stop words from the original code, since this is an abstract compilation, it makes sense to remove the word research, project and students.

Compared to the cleared data frame, this one looks not readable for the human but for the machine is clear and it allows to get the topic

## Natural language processing functions

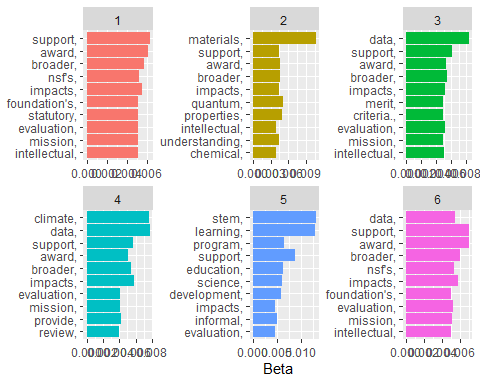
In this function the corpus is created and with the Latent Dirichlet Allocation (LDA), the topics can be modelled.

top\_terms\_by\_topic\_LDA <- function(input\_text, # should be a columm from a dataframe  
 plot = T, # return a plot? TRUE by defult  
 number\_of\_topics = 4) # number of topics (4 by default)  
{   
 # create a corpus (type of object expected by tm) and document term matrix  
 Corpus <- Corpus(VectorSource(input\_text)) # make a corpus object  
 DTM <- DocumentTermMatrix(Corpus) # get the count of words/document  
  
 # remove any empty rows in our document term matrix (if there are any   
 # we'll get an error when we try to run our LDA)  
 unique\_indexes <- unique(DTM$i) # get the index of each unique value  
 DTM <- DTM[unique\_indexes,] # get a subset of only those indexes  
   
 # preform LDA & get the words/topic in a tidy text format  
 lda <- LDA(DTM, k = number\_of\_topics, control = list(seed = 1234))  
 topics <- tidy(lda, matrix = "beta")  
  
 # get the top ten terms for each topic  
 top\_terms <- topics %>% # take the topics data frame and..  
 group\_by(topic) %>% # treat each topic as a different group  
 top\_n(10, beta) %>% # get the top 10 most informative words  
 ungroup() %>% # ungroup  
 arrange(topic, -beta) # arrange words in descending informativeness  
  
 # if the user asks for a plot (TRUE by default)  
 if(plot == T){  
 # plot the top ten terms for each topic in order  
 top\_terms %>% # take the top terms  
 mutate(term = reorder(term, beta)) %>% # sort terms by beta value   
 ggplot(aes(term, beta, fill = factor(topic))) + # plot beta by theme  
 geom\_col(show.legend = FALSE) + # as a bar plot  
 facet\_wrap(~ topic, scales = "free") + # which each topic in a seperate plot  
 labs(x = NULL, y = "Beta") + # no x label, change y label   
 coord\_flip() # turn bars sideways  
 }else{   
 # if the user does not request a plot  
 # return a list of sorted terms instead  
 return(top\_terms)  
 }  
}

## Topic analysis

Now that we have the algorithm ready, let’s model some topics. We’re going to start with 6 topics

top\_terms\_by\_topic\_LDA(cleaned\_documents$terms, number\_of\_topics = 6)

 In this case we can see that the main topic is related to an award from the Naturla Science Foundation. Another important topic is related to the understanding of quantum, probably computing. The third topic is in the evaluation. The fift topic is interesting because it is related to the climate change. While the 6th topic is the education

## Conclusion

NLP provides very useful tools to understand a big group of datasets and have an idea of it. Manually reading the over 2000 abstracts is a tedious tasks while NLP in less time can give an insight

## Aknowledgments

Special thanks for Rachael Tatman for her NLP code that was the reference <https://www.kaggle.com/rtatman/nlp-in-r-topic-modelling/data?select=deceptive-opinion.csv>