There is an interesting challenge running on Kaggle at the moment. It has been designed in cooperation with the Coleridge Initiative (https://coleridgeinitiative.org/) . This initiative is established at the New York University, it's goal is to facilitate data driven decision making by governments. In the challenge we get to optimize automated dataset detection In policy papers. The ultimate goal? We need to help civil servants make better and more efficient decisions.

What is the key to making better decisions? Use the most up to date scientific research and data. Data Science can help here to make this process more efficient. Specifically we can use Natural Language Processing (NLP) techniques to predict what papers use a dataset. This is the essence of the Kaggle challenge. If it succeeds, it can facilitate government decision making transparency.

In this initial blog I want to discuss some of my considerations when thinking about this challenge. There will also be some quick analysis on the dataset provided.

Challenge overview

So you are working on a problem, how many times have you looked for a good scientific source to find a solution? Put yourself in the position of someone working for a local government. You have to come up with a local policy impact analysis on the councils climate change efforts. You've been looking for evidence and scientific research all morning. Isn't there a tool that immediately identifies all similar research where similar datasets are used?

I can imagine these decisions are made on a daily basis, all around the world. Designing better systems to navigate this huge information source is the key to data driven decision making. To improve such mass scale decision making, with direct societal impact. It triggers my imagination.

Next the dataset, we are given roughly 20.000 papers, and have to predict what datasets are used in these documents. There are 3 key technical aspects I want to discuss, that in my opinion are vital to reaching a solution.

- Feature addition
- Similarity index of scientific papers
- Transfer learning

Feature addition

The dataset literally consists of 5 columns. A document ID, the publication title, the dataset used in the publication, the institute associated with it and a cleaned label for the dataset. Firstly one of the potential challenges with text data is to find reliable additional structured data from the raw text provided. We want to see if there are some other features available!

There are multiple reasons for attempting to find structured additional data from the provided text. If we for example can extract the date of the publication from the text, we can see if this date of publication is a relevant factor in the usage of different datasets. one of the most important reasons is that it makes our ultimate solution easier to understand for policy users. Hence it creates a simpler interpretation of our outcome if we can use these structured elements. In our efforts to improve decision making it is our duty as the data scientist to maximize understanding of our solution.

From a data science perspective, creating additional metadata also helps us improve our own

understanding of the problem. In one of the upcoming blogs I will attempt to use the text to create these metadata elements. From the top of my head important elements are:

- Publication title
- Author / Organization
- Total citations
- Citated papers
- Text length

There is probably more potential for data enrichment, but this is a starting consideration for me when going over this challenge.

Similarity index of scientific papers

When we look for different papers that cover similar datasets, we assume that there is some similarity. There are similar technical terms used and introduced, similar domain knowledge. It is logical that a dataset that covers economic indicators, has similar explanations of these indicators. A dataset has a theoretical limit to the angles it can be discussed in.

Aside from the challenge of predicting datasets used in the articles, it should be an interesting business case to cluster the scientific papers, and find similar articles. Are there for example very different articles that cover the same dataset? By zeroing in on these effects we can more optimally cover all the relevant knowledge on a given dataset. It also just interest me a lot this question, and i'm curious if there are very different articles that cover similar data.

One of the ways to approach this simply is to use a string matching alghorithm. Using this method is an interesting baseline approach for the challenge. When I start a challenge I am always looking for a baseline. What I find most helpful in establishing a simple baseline early is clarity on the challenge complexity. For me it answers the question, how difficult is this really? When we create a simple intuitive model it helps understand how humans work the problem in practice. Finally It will further help to understand the mountain we have to climb with our more advanced predictive model.

Transfer learning

When I read this challenge part of what peaked my interest is the potential to use transfer learning. In the field of Natural Language Processing there has been a big buzz around transfer learning. For a scientific breakdown see here , another great link is found here . One of the most promising recent models is BERT.

Personally I've never used transfer learning and am eager to give it a try in this challenge. For me It would be a great victory to beat my string matching baseline with a transfer learning method. This will take some studying on the links mentioned in the previous paragraph, but there is alot of great content available.

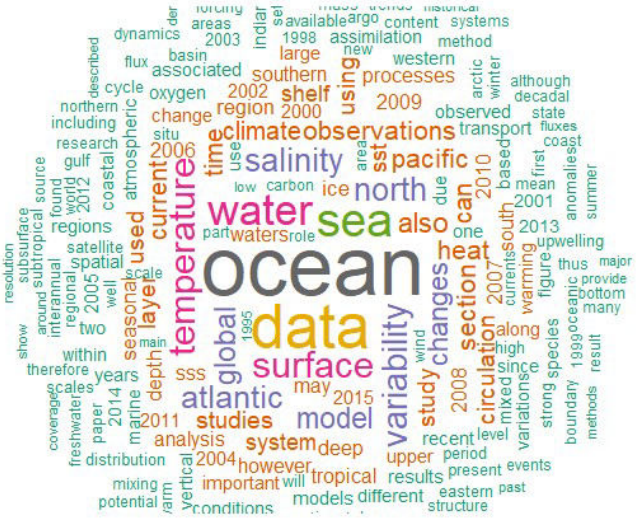
Improving data driven decision making

The purpose of this blog is to write down initial thoughts. I will start off in the challenge by tackling these aspects. It helps me create a structure for my project. Finally it clears the path for the focus of the challenge.

We are focusing on improving data driven decision making for governments. Creating a good prediction model is one part of this, and that is the focus of the Kaggle challenge. But working on this I also want to broaden the impact. Ultimately the prediction model needs to be

explainable and usable to achieve impact. Hence keeping this in the back of our minds as we design it is vital.

To conclude I will leave you with a word cloud of the papers that mention the datasets “World Ocean Database” and “Census of Agriculture”. I expect they look sufficiently different to be of interest. The code used for creating them is below the pictures.



Ocean vs Agriculture dataset wordcloud

```
# Library
library(jsonlite)
library(feather)
library(tm)
library(wordcloud)
```

```
# load and process data

data <- read.csv("C:/Startup/ColeRidge/train.csv")

files_train <- list.files('C:/Startup/ColeRidge/train', full.names = T,
recursive = FALSE)
files_id <- gsub('.json','',basename(files_train))

load_json <- function(file, files_id){
  output             <- jsonlite::fromJSON(file)
  output$publication <- files_id
  output
}

if(file.exists('C:/Startup/ColeRidge/json_files.feather')){
  json_files <- read_feather('C:/Startup/ColeRidge/json_files.feather')
} else {
  json_files <- lapply(1:length(files_train), function(x)
{load_json(files_train[x], files_id[x])})
  json_files <- data.table::rbindlist(json_files)
  write_feather(json_files, path = 'C:/Startup/ColeRidge/json_
files.feather')
}

data <- merge(data, json_files, by.x = 'Id', by.y = 'publication',
all.x = T, all.y = F)

world_ocean <- data[which(data$dataset_title == 'World Ocean Database'
& data$section_title =='Introduction'),]
agri_census <- data[which(data$dataset_title == 'Census of Agriculture'
& data$section_title =='Introduction'),]

build_wordcloud <- function(text){

  data_corp <- Corpus(VectorSource(text))

  # Cleaning:
  data_corp <- tm_map(data_corp, content_transformer(tolower))
  data_corp <- tm_map(data_corp, removeWords, stopwords("english"))
  data_corp <- tm_map(data_corp, removePunctuation)

  # Document term matrix:
  data_corp <- TermDocumentMatrix(data_corp)
  data_corp <- as.matrix(data_corp)
  data_corp <- sort(rowSums(data_corp),decreasing=TRUE)
  data_corp <- data.frame(word = names(data_corp),freq=data_corp)

  set.seed(1234)
  wordcloud(words = data_corp$word, freq = data_corp$freq, min.freq =
1,
            max.words=200, random.order=FALSE, rot.per=0.35,
```

```
            colors=brewer.pal(8, "Dark2"))


}

build_wordcloud(world_ocean$text)
build_wordcloud(agri_census$text)
```