National College of Ireland

Configuration Manual

MSc Research Project
MSc in FinTech

Mudit Agarwal
Student ID: x18108202

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Mudit Agarwal

Student ID: X18108202

Programme: MSc in FinTech Year: 2019

Module: MSc Research Project

Lecturer: Victor Del Rosal

Submission Due Date: 12/08/2019

Project Title: Analysing the Evolution of FinTech Research Topics in Academia

Word Count: 1397 Page Count: 7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: 16/09/2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Mudit Agarwal
Student ID: x18108202

1 Introduction

This configuration manual specifies the system setup including software and hardware requirements needed to perform the research undertaken for the thesis titled “Analysing the Evolution of FinTech Research Topics in Academia”. The steps to be followed in order to replicate the experiment/research performed are also laid out in an easy to follow manner to ensure the research can be easily reproduced to analyse the results by others. Code snippets and outputs are also portrayed and discussed to provide a deeper insight into the inner workings of the code. This further enhances the replicability of the experiment.

2 Dataset

The IEEE Xplore Digital Library (IEEE, 2019) is chosen as the primary data source for the purpose of this research as it is a vast research database for accessing journal articles, magazines and conference proceedings among other materials related to the field of computer science, electrical engineering and electronics and similar fields. One of the major advantages of IEEE Xplore Digital Library over other prominent research databases is that it offers a very easy and intuitive way of exporting relatively large amounts of data with a single click as a csv file. All search results against a search string can be selected at once and exported as a csv file for the purpose of this research.

Technical papers like journal articles and conference proceedings on financial technology (FinTech) are needed for this research as the aim is to analyse the hot FinTech topics in academic research. To this end, we search for journal articles and conference proceedings in the IEEE Xplore Digital Library using various buzzwords/keywords from the FinTech domain. These include blockchain, distributed ledger, smart contracts, ICO, cryptocurrency, initial coin offering, DLT, regtech, insurtech, fintech, financial technology and financial services. A total of 14,116 unique publications are obtained upon cleaning and pre-processing the data.

3 Hardware Requirements

Laptop Model: Lenovo Y510p

Processor: Intel® Core™ i7-4700MQ Quad-Core CPU @ 2.40GHz

Installed Memory (RAM): 8GB

Storage Space: 1TB HDD (Hard Disk Drive)

GPU (Graphics Processing Unit): NVIDIA GeForce GT 750M 2GB

System Type: 64-bit Operating System, x64-based processor

4 Software Requirements

R Version: 3.5.1

RStudio Version: 1.1.456

R Libraries Used: textmineR, pacman

Operating System: Windows 10

MS Excel Version: 2016

5 Replication

- Make sure the hardware and software requirements laid out are met.
- Download csv data from IEEE Xplore Digital Library by searching for the following keywords: blockchain, cryptocurrency, ICO, smart contract, DLT, initial coin offering, distributed ledger, fintech, financial services, financial technology, insurtech and regtech. The data taken for this research is from 2010 to 2019.
- Aggregate the downloaded csv data into one csv file by copying and pasting using MS Excel. Clean the data by removing missing and numerical values for Year of Publication and Abstract. This can be done by creating a filter on these two columns, filtering the observations to be deleted and deleting them using the delete key.
- In the R code, change the line that reads the data so that the directory is same as the one where the aggregated, cleaned data is stored. The filename should also be correct.
- Run the R code provided as part of this research.

6 Code

The following code ingests the dataset into R, transforms the data and creates a data corpus for further examination in R. Next, LDA models are run for different values for k.

```
Dataset <- read.csv(file='C:\\Users\\Mudit\\Desktop\\DATASET.csv', header=T, colClasses="character")

absdata <- dataset

str(absdata)

dtm <- CreateDtm(doc_vec = absdata$Abstract,
                 ngram_window = c(1, 2),
                 stopword_vec = c(stopwords::stopwords("en"),
                 stopwords::stopwords(source = "smart")),
```

```
            lower = TRUE,
            remove_punctuation = TRUE,
            remove_numbers = TRUE,
            verbose = FALSE,
            cpus = 2)

dtm <- dtm[,colSums(dtm) > 2]
set.seed(12345)
model20 <- FitLdaModel(dtm = dtm,
                k = 20,
                iterations = 500,
                burnin = 200,
                alpha = 0.1,
                beta = 0.05,
                optimize_alpha = TRUE,
                calc_likelihood = TRUE,
                calc_coherence = TRUE,
                calc_r2 = TRUE)
```

Furthermore, the following code evaluates the different models using R^2, coherence and log-likelihood.

```
r2_vals <- c(model5$r2, model10$r2, model20$r2, model30$r2, model40$r2, model50$r2, model100$r2, model150$r2, model200$r2)
names(r2_vals) <- c("k=5","k=10","k=20","k=30","k=40","k=50","k=100","k=150","k=200")
r2_vals

summary(model5$coherence)
summary(model10$coherence)
summary(model20$coherence)
summary(model30$coherence)
summary(model40$coherence)
summary(model50$coherence)
summary(model100$coherence)
summary(model150$coherence)
summary(model200$coherence)

model5$log_likelihood[50,]
model10$log_likelihood[50,]
model20$log_likelihood[50,]
model30$log_likelihood[50,]
model40$log_likelihood[50,]
model50$log_likelihood[50,]
model100$log_likelihood[50,]
model150$log_likelihood[50,]
model200$log_likelihood[50,]

str(model50)

model50$top_terms <- GetTopTerms(phi = model50$phi, M = 10)
head(t(model50$top_terms))
```

Then, prevalence vs alpha graph is plotted and the extracted topics are labelled using textmineR library. A summary of the top 10 topics is generated for gaining insight into the most prevalent topics as seen below.

```
model50$prevalence <- colSums(model50$theta) / sum(model50$theta) * 100
plot(model50$prevalence, model50$alpha, xlab = "prevalence", ylab = "alpha")
```

```
model50$labels <- LabelTopics(assignments = model50$theta > 0.05,
                    dtm = dtm,
                    M = 1)
head(model50$labels, n=10)
t(model50$labels)

model50$summary <- data.frame(topic = rownames(model50$phi),
                    label = model50$labels,
                    coherence = round(model50$coherence, 3),
                    prevalence = round(model50$prevalence,3),
                    top_terms = apply(model50$top_terms, 2, function(x){
                      paste(x, collapse = ", ")
                    }),
                    stringsAsFactors = FALSE)

model50$summary[ order(model50$summary$prevalence, decreasing = TRUE) , ][ 1:10 , ]
```

Following this, the topics are converted to time series and their slopes are calculated. The slopes may be positive or negative. The hottest terms and coldest terms are identified as shown below.

```
theta_mean_by <- by(model50$theta, absdata$Publication_Year, colMeans)
theta_mean <- do.call("rbind",theta_mean_by)
colnames(theta_mean) = paste(1:50)
theta_mean_ts <- ts(theta_mean, start = 2010)
theta_mean_time <- time(theta_mean)

tm_lm <- apply(theta_mean, 2, function(x) lm(x ~ theta_mean_time))
tm_lm_coef <- lapply(tm_lm,function(x) coef(summary(x)))
tm_lm_coef_sign <- sapply(tm_lm_coef,['',"theta_mean_time","Pr(>|t|)")
tm_lm_coef_slope <- sapply(tm_lm_coef,['',"theta_mean_time","Estimate")

tm_lm_coef_slope_pos <- tm_lm_coef_slope[tm_lm_coef_slope >= 0]
tm_lm_coef_slope_neg <- tm_lm_coef_slope[tm_lm_coef_slope < 0]

p <- c(0.05, 0.01, 0.001, 0.0001)
total_significance <- sapply(p,
                      function(x) (tm_lm_coef_sign[tm_lm_coef_sign < x]))
negative_significance <- sapply(1:length(p),
                    function(x) intersect(names(tm_lm_coef_slope_neg),names(total_significance[[x]])))
positive_significance <- sapply(1:length(p),
                    function(x) intersect(names(tm_lm_coef_slope_pos),names(total_significance[[x]])))

trend_matrix <- rbind(sapply(negative_significance,length),
                sapply(positive_significance,length ),
                sapply (total_significance,length ))
rownames(trend_matrix) <- c("Negative trend", "Positive trend", "Total")
colnames(trend_matrix) <- c("p<0.05", "p<0.01", "p<0.001", "p<0.0001")
trend_matrix

hot_topics <- as.numeric(names(sort(tm_lm_coef_slope[positive_significance[[1]]], decreasing=TRUE)))
cold_topics <- as.numeric(names(sort(tm_lm_coef_slope[negative_significance[[1]]], decreasing=FALSE)))

hot_and_cold_ts <- cbind(theta_mean_ts[,hot_topics[1:10]],
                    theta_mean_ts[,cold_topics[1:10]],
                    deparse.level=0)
colnames(hot_and_cold_ts) <- as.character(c(hot_topics[1:10],cold_topics[1:10]))

hot_ts <- theta_mean_ts[,hot_topics[1:10]]
```

```
cold_ts <- theta_mean_ts[,cold_topics[1:10]]
colnames(hot_ts) <- as.character(hot_topics[1:10])
colnames(cold_ts) <- as.character(cold_topics[1:10])

hot_words <- model50$top_terms[,hot_topics[1:9]]
cold_words <- model50$top_terms[,cold_topics[1:10]]
hot_words
cold_words
```

Finally, multilayer perceptrons (MLPs) are used to identify non-linear trends in topics similar to the research done by Bittermann & Fischer (2018). The topics identified as having non-linear trends are plotted as time series to show their non-linearity.

```
rn=numeric(0)
rlm=numeric(0)

set.seed(282828)
for(i in 1:50){
  ts <- scale(theta_mean_ts[,i])
  theta_mean_time_s <- scale(theta_mean_time)

  n <- nnet(x=theta_mean_time_s,y=ts,size=2,linout=T,trace=F);
  for(j in 1:100){
    m=nnet(x=theta_mean_time_s,y=ts,size=2,linout=T,trace=F);a1<-abs(cor(m$fitted.values,ts));a2<-
abs(cor(n$fitted.values,ts))
    if(is.na(a1)==F && is.na(a2)==F) if( abs(cor(m$fitted.values,ts)) > abs(cor(n$fitted.values,ts))) n<-m

  }
  rn <- c(rn,cor(ts,fitted(n))^2)

  lm <- lm(ts ~ theta_mean_time)
  rlm <- c(rlm,cor(ts,fitted(lm))^2)
}


rq <- data.frame(rlm, rn)
rq$check <- ifelse(rq$rn > 2*rq$rlm, 1, 0)
rq$ratio <- rq$rlm / rq$rn

write.csv2(rq, "rq.csv")


ratio <- rq$ratio
names(ratio) <- rownames(rq)
nonlinear <- head(sort(ratio, decreasing = FALSE), 10)
list.nonlinear <- as.numeric(names(nonlinear))

xyplot(theta_mean_ts[,list.nonlinear])
terms_nonlinear <- model50$top_terms[,list.nonlinear]
terms_nonlinear
```

References

Bittermann, A. and Fischer, A. (2018). How to Identify Hot Topics in Psychology Using Topic Modeling. *Zeitschrift für Psychologie*, 226(1), pp.3-13.

IEEE, 2019. IEEE Xplore. [Online] Available at: https://ieeexplore.ieee.org/Xplore/home.jsp [Accessed 11 August 2019].