1. Implementierung einer Teilmenge ausgewählten Ansätzen

Dieses Kapitel befasst sich mit der Implementierung von zwei ausgewählten Ansätze zur Dokumentähnlichkeitsbestimmung, die auf Bayesscher Statistik basieren, nämlich BayesLSH und BayesLSHLite. Diese Ansätze, deren Pseudocode schon im vorherigen Kapitel dargestellt wurden, werden in der Programmiersprache R implementiert.

Die Programmiersoftware und -sprache R

R ist eine Programmiersprache, die dazu dient statistische Berechnungen durchzuführen und Graphiken darzustellen. R wird durch die freie Software „R“ betrieben und ist ähnlich zu der S Programmiersprache. Unter „freie Software“ wird eine Nutzerfreiheit und -gemeinschaft respektierende Software verstanden. Die R Software ist kostenlos im Internet verfügbar und kann sowohl auf UNIX Plattformen, als auch auf Windows und MacOS Betriebssysteme betrieben werden. R ermöglicht es sowohl statistische Methoden wie lineare und nicht lineare Modellierung, klassische statistische Testen, Time-series Analyse Klassifikation oder noch Clustering, als auch graphische Methoden durchzuführen.

Mit R sind viele Möglichkeiten für den Programmierer verfügbar. Daten können unter anderem effektiv verarbeitet werden und der Speicher eingerichtet werden. Es bestehen Operatoren mit denen Arrays (Matrizen) berechnet werden können sowie eine Sammlung von Zwischenwerkzeugen und grafische Einrichtungen, die sich für die Datenanalyse eignen. Wie die meisten Programmiersprachen ist die Anwendung mit Bedingungen, Schleifen, benutzerdefinierte rekursive Funktionen, sowie Eingabe- und Ausgabefunktionen versehen. R ist flexibel und erweiterbar, d. h es verfügt außerdem auch über Pakete, die sowohl durch der R Distribution als auch durch die CRAN-Familie im Internet erhältlich sind.

Die Firma Dibuco GmbH entwickelt normalerweise Softwarelösungen in die Programmiersprache Java oder Scala aber die verwendete Programmiersprache für diese Thesis ist R, und die Software dazu ist Rstudio. Da R Statistik mit Datenvisualisierung kombiniert wird und das zentrale Thema dieser Arbeit beruht auf Bayesscher Statistik, wurde es entschieden diese Programmiersprache anzuwenden. Dies hat unter anderem unterschiedliche Gründe. Es bestehen schon in R Bibliotheken und vorgefertigte Algorithmen, die nur zur Anwendung bereit sind. R kann unter anderem sowohl mit Spark (Big Data Framework für Analytik) als auch mit KNIME (Analytik Plattform) betrieben werden und fast alle Datenbanken können von der ausgelesen werden. Das Textreuse Paket, das später benutzt wird für die Implementierung des Ansatzes kommt vom MIT und es besteht eine große Community (Weltweit bestehend), die Pakete zur Verfügung stellt. Ein weiterer Vorteil von R wie schon im Abschnitt (1.1) erwähnt, ist dass, sie eine open Source ist. Außerdem viele wissenschaftliche Arbeiten können in R erstellt werden, bzw. Berechnungen und Graphiken können direkt in wissenschaftlichen Arbeiten gegeben werden.

1.1. Die Algorithmen

Der für diesen Implementierungsteil angewendete Algorithmus ist der BayesLSH Algorithmus, dessen Pseudocode beschrieben wurde im Kapitel 2. Es ist hinzuweisen, dass diesen Algorithmus als Leitfaden für die Implementierung des Ansatzes genommen wird, bzw. für das Schreiben des Codes. Weitere Details werden im Abschnitt (Code noch hinzufügen) erläutert.

1.2. Wichtige Pakete

Wichtige Pakete für die Implementierung dieser Arbeit sind die Pakete textreuse, rjson und pdftools

1.2.1. Das Paket textreuse

Das Paket textreuse ist verfügbar auf die R Plattform und enthält eine Menge von Funktionen, zur Ähnlichkeitsmessung zwischen Dokumenten sowie Erkennung von wiederverwendeten Textpassage in Dokumente. Das Paket textreuse implementiert Minhash, Locality Sensitive Hashing Funktionen sowie Ähnlichkeitsfunktionen.

1.2.2. Der Code

Der Code dieser Implementierung wird in R geschrieben und anhand vom existierenden Paket Textreuse durchgeführt. Es werden auch ein Paar in R herunterzuladende Bibliothekpakete wie „pdftools“, zum Auslesen von PDF-Dateien, rjson zum Auslesen von JSON Dokumente.

```r
#library(pdftools)
library(tidyverse)
library(textreuse)
library(readtext)
library(readxl)



minhash <- minhash_generator(n=200, seed = 234)
corpus <- TextReuseCorpus(dir = paste0(getwd(),"/mix/experimental_data/test-set_json_equal-and-greater-than-200kb"),
                          tokenizer = tokenize_ngrams, n = 2,
                          minhash_func = minhash, progress = FALSE,
                          skip_short = FALSE)
# Threshold for default values
#lsh_threshold(h = 200, b = 40)
#buckets <- lsh(corpus, bands = 50)


#candidates <- lsh_candidates(buckets)
comparisons_jsim <- pairwise_compare(corpus, jaccard_similarity)
```

Abbildung 1: Der Code zur Implementierung des BayesLSH mit dem Jaccard Index

[RPR18] https://www.r-project.org/ (Letzter Abruf:21.06.2018)

[GNU18] http://www.gnu.org/philosophy/free-sw (Letzter Abruf:21.06.2018)

[HPS18] HP Support Assistant, integriertes Service in HP Pavilion Notebook

[RDO18] https://www.rdocumentation.org/packages/textreuse/versions/0.1.4 (Letzter Abruf:24.06.2018)

[RPT18] https://cran.r-project.org/web/packages/textreuse/textreuse.pdf