

# Text Mining Tool für eine automatisierte Literaturanalyse über Big Data

<b>Themenbereiche:</b>	Big Data, Text Mining, Automatic Tagging, Ontology Learning
<b>Studierende:</b>	Fabrizio Rohrbach
<b>Betreuungsperson:</b>	Michael Kaufmann
<b>Experte:</b>	Matin Burri
<b>Auftraggebende:</b>	Michael Kaufmann
<b>Keywords:</b>	Big Data, Text Mining, Automatic Tagging, Ontology Learning

## 1. Aufgabenstellung

### Problemstellung

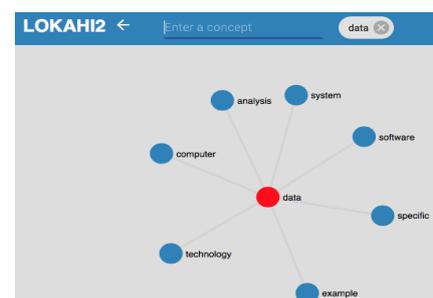
Akademische Datenbanken wie z.B. „ScienceDirect“ der Firma „Elsevier“ bieten den Anwendern die Möglichkeit nach wissenschaftlichen Dokumenten zu suchen und diese einzusehen. Derzeit liefert ScienceDirect 2‘175 wissenschaftliche Papers welche „Big Data“ im Titel nennen (<https://www.sciencedirect.com/search/advanced?title=%22Big%20Data%22>).

Hier stellt sich das Problem, dass ein einzelner Mensch diese Anzahl an wissenschaftlichen Papers in keiner vernünftigen Zeit lesen, die Inhalte erfassen und strukturieren kann.

Mittels „Text Mining“ kann dem Problem entgegengewirkt werden und automatisch aus einem Korpus von unstrukturierten Texten neue Erkenntnisse gewonnen werden.

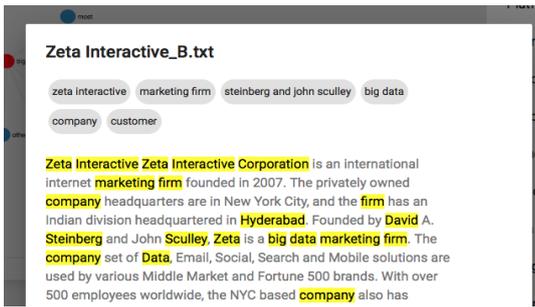
### Ausgangslage

Elsevier bietet für ScienceDirect eine Schnittstelle (API) mit der auf die Datenbank von ScienceDirect bzw. die Metadaten der wissenschaftlichen Papers zugegriffen werden kann (<https://www.elsevier.com/solutions/sciencedirect/support/api>). Diese kann fürs „Text Mining“ verwendet werden. Auch andere akademische Datenbanken wie z.B. „Web of Science“ der Firma „Clarivate Analytics“ bieten Schnittstellen (API) an.



Die Forschungsgruppe der Hochschule Luzern hat ein Prototyp mit dem Namen Lokahi entwickelt, welcher in der Lage ist aus einem Korpus von Textdateien die relevantesten Schlüsselwörter (Konzepte) sowie deren Beziehungen zu anderen Konzepten mittels Algorithmen zu erkennen und diese dem Anwender grafisch darzustellen. Die folgende Abbildung zeigt die Schlüsselwörter zum eingegebenen Begriff „data“ im Korpus mittels PMI.

Lokahi unterstützt neben der Extraktion der Konzepte und deren Beziehungen mittels „Pointwise mutual information“ (PMI) auch die Extraktion mittels „Likelihood ratio“ (LR).



Zudem kann Lokahi dem Anwender die Textdokumente im Korpus anzeigen, in welchen die Konzepte vorkommen sowie die genaue Textstelle mittels Tagging hervorheben.

Derzeit basiert der Korpus von Lokahi auf 500'000 Wikipedia-Artikel die jeweils den Titel sowie den Inhalt des Wikipedia-Artikel beinhalten.

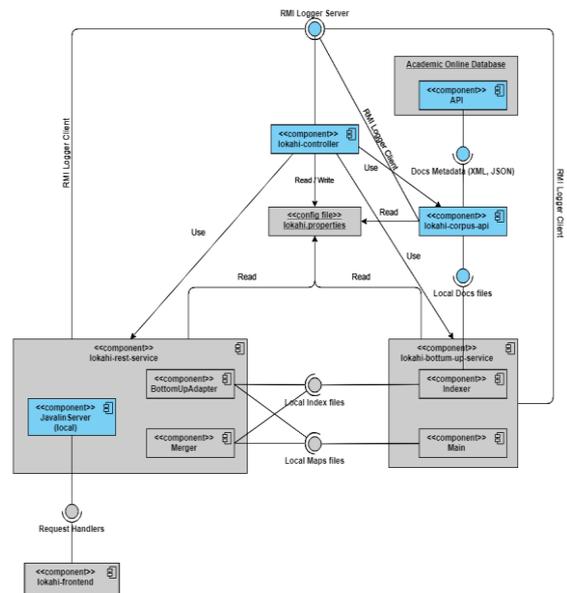
## 2. Ergebnisse

Die grafische Komponente „Controller“ bietet die Möglichkeit die Anwendung zu konfigurieren und zu bedienen. Die Anwendung lässt sich mittels Doppelclicks ausführen, ohne dass zusätzliche Software installiert werden muss.

Der zu analysierende Korpus von Metadaten zu einem Suchbegriff kann mittels Anbindung an die Schnittstelle von „Microsoft Academic“ heruntergeladen werden.

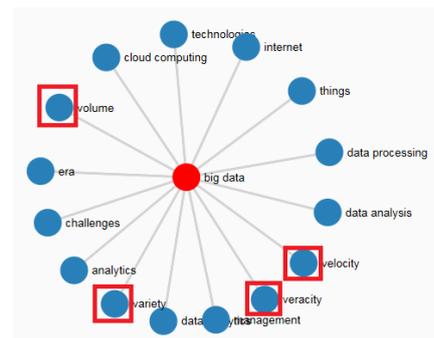
Durch die Optimierung der Extraktion der Konzepte mittels Liste aller englischen Wikipedia Artikel Titel, sowie mittels „Part of Speech“ (POS) Filterung konnten die falsch erkannten Konzepte minimiert werden. Dies erlaubt es dem Anwender den Fokus auf die relevanten Konzepte bzw. Informationen zu legen.

Das Endprodukt ermöglicht dem Anwender mittels „Text Mining“ aus einem vorhandenen oder über die Anwendung heruntergeladenen Korpus die relevantesten Schlüsselwörter (Konzepte) sowie deren Beziehungen zu extrahieren und darzustellen. Dadurch gewinnt der Anwender neue Erkenntnisse über einen Korpus von unstrukturierten Daten (Textdateien).



### Praxisbeispiel

Ein Datenanalyst, dem die Anwendung ausgeliefert wird, kann einen Korpus von Metadaten zum Thema „Big data“ herunterladen und analysieren. Mittels der oben genannten Standardeinstellungen werden dem Datenanalyst die Top 15 Konzepte, die zum Konzept „Big data“ in Relation stehen, dargestellt. Wird z.B. das Konzept „velocity“ angeklickt, so erscheinen die zum Konzept „velocity“ in Relation stehenden Top 15 Konzepte. Unter diesen Konzepten ist auch das Konzept „data“. Wird das Konzept „data“ angeklickt, so erscheinen die Top 15 Konzepte zum Konzept „data“. Wird danach das Konzept „hadoop“ angeklickt, so erscheint unter anderem das Konzept „mapreduce“ sowie „hdfs“ unter den Top 15 Konzepten. Damit lässt sich schliessen, dass das Konzept „hadoop“, „mapreduce“ und „hdfs“ indirekt mit dem Konzept „big data“ in Relation stehen. Dies trifft zu, da „Hadoop“ ein Java Open Source Framework von „Apache“ für grosse Datensätze ist. Darin gibt es zwei Hauptlayers „MapReduce“ und „HDFS“.



### 3. Lösungskonzept

#### Anbindung an akademische Datenbank

Es wurde eine zusätzliche Komponente („lokahi-corpus-api“) entwickelt, welche die Metadaten „Titel“, „Autor“, „Abstract“, „Keywords“ und „Erscheinungsjahr“ der Dokumente aus der akademischen Datenbank „Microsoft Academic“ zu einem vom Anwender festgelegten Suchbegriff herunterlädt. Der heruntergeladene Korpus kann vom „Lokahi Prototyp“ weiter für die Extraktion der Konzepte und deren Beziehungen verwendet werden.

#### Konvertierung in lokale Anwendung

Ursprünglich war der „Lokahi Prototyp“ eine Webanwendung. Um die Anwendung in eine lokale Anwendung zu konvertieren, wurde der Webserver in der „lokahi-rest-service“ Komponente im lokalen Modus („localhost“) verwendet. Zusätzlich werden die Konzepte und deren Beziehungen in einem internen Browser der Anwendung dargestellt.

#### Funktionale Erweiterung

Der „Lokahi Prototyp“ wurde mit einer Filterung der Konzepte mittels Liste aller englischen Wikipedia Artikel sowie „Part of Speech“ (POS) erweitert. Dabei fallen alle Konzepte weg, welche nicht als Titel für Artikel der Seite Wikipedia (<https://en.wikipedia.org>) existieren oder z.B. nur aus Verben bestehen.

### 4. Spezielle Herausforderungen

#### Abhängigkeiten

Um dem Anwender die Installation der Anwendung möglichst einfach zu machen, wurde zum einen eine „Fat Jar“ bzw. eine „Uber Jar“ (Jar Datei, die alle externen Abhängigkeiten enthält) erstellt und zum anderen die „Java Runtime Environment“ (JRE) Version 9.0.4 mittels der Software „packr“ in die Anwendung eingebunden.

#### Geschwindigkeit (Performance) vs. Arbeitsspeicher (Memory)

Es musste entschieden werden, ob zur Filterung die Liste aller englischen Wikipedia Artikel zu Beginn komplett von der Anwendung in den Arbeitsspeicher geladen werden sollte (hohe Arbeitsspeicherbelastung, dafür schnellere Abfrage) oder die Liste bei jeder Prüfung Zeile für Zeile in den Arbeitsspeicher geladen und danach verworfen werden soll (niedrige Arbeitsspeicherbelastung, dafür sehr langsame Abfrage). Zur Lösung des Problems (Erreichen einer guten Geschwindigkeit bei einer nicht zu hohen Arbeitsspeicherbelastung) wurde der arbeitsspeichersparende Dateityp „THashSet“ von „Trove“ verwendet.

### 5. Ausblick

#### Optimierung

Bei der Extraktion der Konzepte werden die Uni Gramme und die N-Gramme separat extrahiert, dies könnte auch in einem einzelnen Schritt durchgeführt werden.

#### Funktionale Erweiterung

In der Zukunft könnten, die in diesem Projekt nicht umgesetzten Anforderungen der funktionalen Erweiterung umgesetzt werden („Keyword Ranking“, „Trendanalyse“, „Clustering von Dokumenten“, „Unterstützung von Word und PDF Dokumenten“, „Term Kurzbeschreibung“ sowie „Optimierung der Graphen Darstellung“).