Technology
Arts Sciences
TH Köln

INFOMOTION
we love data

BACHELORARBEIT

Text-Mining auf Basis von SAP HANA am Beispiel von Social-Media-Beiträgen eines Handelsunternehmens

Bachelorarbeit zur Erlangung des Bachelorgrades
Bachelor of Science im Studiengang Wirtschaftsinformatik
an der Fakultät für Informatik und Ingenieurwissenschaften
der Technischen Hochschule Köln

Vorgelegt von: Herr Arkin Kizilarslan
Matrikelnummer: 11110316

Erster Prüfer: Frau Prof. Dr. Birgit Bertelsmeier
Zweiter Prüfer : Herr Daniel Artmann

Gummersbach, 24.09.2018

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AFL	Application Function Library
BFL	Business Function Library
CPU	Central Processing Unit
HANA	High Performance Analytic Appliance
IMDB	In-Memory-Datenbank
KDD	Knowledge Discovery in Databases
LDA	Latent Dirichlet Allocation
MDX	Multidimensional Expressions
PAL	Predictive Analysis Library
RAM	Random Access Memory
TF-IDF	Term Frequency-Inverse Document Frequency

1 Einleitung

Im Zeitalter der Digitalisierung sind Menschen in unterschiedlichster Art und Weise mit Maschinen in Kontakt. Die Nutzung von digitalen Lösungen durch Menschen und die Kommunikation von Maschinen untereinander, dienen Durchführung abgegrenzter Anwendungsfälle in einer spezifischen Domäne. Diese erzeugen strukturierte und unstrukturierte Daten in verschiedenen Variationen. Die Entwicklungen und Fortschritte im Bereich der Digitalisierung werden weiterhin ausgebaut und es entstehen, neben konventionellen Geschäftsmodellen, zunehmend neue innovative Geschäftsmodelle, die auf digitale Dienstleistungen beruhen, ohne von spezifischer Hardware abhängig zu sein.
Unternehmen wie Facebook, Instagram, Snapchat und Whatsapp basieren auf innovativen Geschäftsmodellen und haben das digitale soziale Netzwerk geschaffen, um das Sozialisieren der Menschen auf Basis des Internets zu ermöglichen. Der Einfluss der Social-Media-Plattformen ist signifikant und hat ein neues Modell menschlicher Interaktion geschaffen. Die Geschäftsmodelle sind sehr beliebt. Social-Media-Plattformen, wie Facebook, sind ein digitaler Treffpunkt, in dem Nachrichten, Veranstaltungen, Bilder, Videos, eigene Vorlieben und vieles weitere ausgetauscht werden. Auch Unternehmen haben das Potential von Social-Media-Plattformen erkannt, sind nun daran interessiert Kunden auf diesen Plattformen zu begegnen und gezieltere Marketing-Aktivitäten auszuführen. Laut Statitsta sind durchschnittlich 1,47 Milliarden Nutzer täglich auf Facebook aktiv [Vgl. Vgl. Statista GmbH, 2018c].

Mit dem kontinuierlich fortschreitendem Umstieg auf das Medium Internet entstehen extensive Datenmengen, die vielfältig sind und in hoher Geschwindigkeit erzeugt werden. Dieses Phänomen wird Big Data bezeichnet und birgt viele ausschöpfbare Potentiale, wenn die Expertise und Ressourcen zur Datenanalyse vorhanden sind. The Economist unterstreicht die Bedeutung von Daten und konstatiert, dass Öl nicht mehr die wertvollste Ressource auf der Welt ist, sondern Daten [Vgl. The Economist, 2017].

Laut der Studie von International Data Corporation wird für 2025 eine weltweite Datenmenge von insgesamt 163 Zettabytes vorhergesagt. Dies entspricht 163 Trillionen Gigabytes. Zudem nennt die Studie einen Schlüsseltrend im Kontext des Umgangs mit der riesigen Datenmenge und sagt aus, dass die Echtzeitdatenverfügbarkeit eine immer wichtiger werdende Rolle erfahren wird. [Vgl. Reinsel et al., 2017, S. 3 f.]

Es werden In-Memory-Datenbanken (IMDB) entwickelt, die Anforderungen der Echtzeitdatenverfügbarkeit und -verarbeitung erfüllen können, um den effektiven und effizienten Umgang mit den stetig wachsenden Daten zu gewährleisten. IMDB persistieren Daten nicht auf gewöhnlichen Festplatten, sondern auf der Random Access Memory (RAM), welche einen deutlich schnelleren Datenzugriff realisiert. Die IMDB HANA von SAP wird seit längerem weiterentwickelt und gegenwärtig in die IT-Infrastruktur von Unternehmen integriert, um Geschäftsprozesse in Echtzeit ausführbar zu machen. SAP möchte die HANA für viele

Anwendungen zugänglich machen, so dass angestoßene Prozesse schneller ausgeführt und Wartezeiten minimiert werden. Zudem bietet SAP einige anwendungsfallspezifische Bibliotheken an, wobei die Möglichkeit besteht, auf der HANA-Datenbank ressourcenlastige Prozesse auszuführen ohne externe Anwendungen zu verwenden.

Unternehmen haben großes Interesse daran Big Data möglichst automatisiert zu analysieren und einen Mehrwert aus den großen Datenmengen zu generieren. Im globalen Markt können sich Käufer schnell über Angebote anderer Anbieter informieren, sowie Rezensionen lesen, um bessere Kaufentscheidungen zu treffen. Somit ergibt sich oft die Herausforderung den Markt zu verstehen und das Dienstleistungsportfolio an die Kundenbedürfnisse zuzuschneiden. Um Kundenbedürfnisse zu adressieren, möchten Entscheidungsträger in Unternehmen zur richtigen Zeit die richtigen Informationen haben, damit relevante Marktveränderungen wahrgenommen und Handlungsmaßnahmen erarbeitet werden können. Eine taktische Auslegung der Handlungsmaßnahmen kann Unternehmen im besten Fall in die Lage versetzen, Wettbewerbsvorteile zu erlangen und sichern.

Data Mining ist die Disziplin, welche die automatisierte Analyse von strukturierten Daten ermöglicht. Text-Mining kann als eine Teildisziplin von Data Mining aufgefasst werden, die sich mit der automatisierten Analyse von unstrukturierten Daten befasst. Beide Disziplinen verfolgen das Ziel verborgene Zusammenhänge, Regel- und Gesetzmäßigkeiten in Daten zu finden, um wertvolle, profitable Informationen aus den extensiven Datenbeständen zu kristallisieren und die Wissensentdeckung zu fördern.

Die wesentlichen Hauptthemen, die in der vorliegenden Arbeit miteinander in Zusammenhang stehen, sind die Anwendung von Verfahren im Text-Mining und die IMDB, HANA, des europäischen Softwareherstellers SAP. Hierbei soll die HANA-Technologieplattform als Basis verwendet werden, um ein Text-Mining-Anwendungsfall zu bearbeiten, die die Analyse von Social-Media-Beiträgen vorsieht.

Die zugrundeliegende Arbeit setzt ein Grundlagenverständnis über Datenbankmodellierung und die Datenbanksprache SQL (Structured Query Language) voraus.

1.1 Relevanz und Motivation der Thematik

Text-Mining wird grundsätzlich für umfangreiche Sammlungen von Texten bzw. unstrukturierten Daten verwendet. Dabei können die unstrukturierten Daten vorab existieren oder in kontinuierlichen Zeitabständen in die Infrastruktur der Text-Mining-Prozesse geladen werden. Unternehmen wollen im Markt bestehen bleiben und haben das Ziel mehr Marktanteile zu erringen. Aufgrund der ständig wachsenden Daten innerhalb und außerhalb von Unternehmen, können durch das Text-Mining entscheidende Chancen und Optimierungsmöglichkeiten wahrgenommen werden. Dazu müssen oftmals den Entscheidungsträgern immer die aktuellsten Daten vorliegen.

Herkömmliche relationale Datenbanken können die Anforderung des Datenabrufs in Echtzeit nicht erfüllen. Auf der Random Access Memory (RAM) basierende IMDB, können hingegen solch Anforderungen unterstützen. Der schnelle Abruf von Daten ist für viele Anwendungsfälle eine essenzielle Eigenschaft. In der IT-Branche wird davon ausgegangen, dass IMDB in Zukunft mehr Einsatzszenarien bedienen und für schnell auszuführende Geschäftsprozesse in die Infrastruktur zunehmend mehr Unternehmen integriert werden.
SAP hat die IMDB, HANA entwickelt und sie als eine Technologiepattform zugänglich gemacht. SAP bewirbt die HANA aktuell sehr stark und bietet sie auch als Cloud-Variante an. Zudem ermöglicht SAP die HANA auch für andere Zwecke als die Datenhaltung und –bereitstellung zu verwenden. Demnach wurde die HANA mit der Predictive Analysis Library (PAL) ausgestattet, damit Text- und Data-Mining-Projekte direkt auf der HANA entwickelt werden können. Infolgedessen wird der Gebrauch weiterer Software, für die Datenanalyse mit Text- und Data-Mining-Algorithmen, obsolet. Die Echtzeitanalyse wird verbessert aufgrund der schnellen Verfügbarkeit der Daten, mittels der Nutzung der In-Memory Technologie. Zusätzlich wird die Echtzeitanalyse optimiert, da die Ausführung der anspruchsvollen Text- und Data-Mining-Prozesse in der HANA-Umgebung stattfinden.

Die Motivation zur Bearbeitung dieser Arbeit geht aus der Popularität der HANA in Zusammenhang mit den Herausforderungen von Big Data und der immer bedeutsam werdenden Text-Mining-Disziplin hervor. Das betreuende Unternehmen dieser Arbeit ist die Infomotion GmbH. Im Rahmen des Wissensmanagements hat die Infomotion GmbH das Interesse zu prüfen, ob Text-Mining mit der HANA bei Kunden, die eine HANA-Datenbank betreiben oder neu integrieren wollen, eingesetzt werden kann. Das Beratungsunternehmen ist im Bereich Business Intelligence und Data Analytics tätig und möchte eventuell das Dienstleistungsportfolio, um Text-Mining-Kompetenzen auf Basis der HANA erweitern.

1.2 Themenabgrenzung

Das Projekt befasst sich mit der Anwendung von Text-Mining auf der HANA-Datenbank. Data-Mining wird bis auf die Vermittlung von Grundlagen, nicht tiefer behandelt. Dahingegen werden Text-Mining-Grundlagen ausführlicher vermittelt. Die zu untersuchenden Funktionalitäten der HANA beschränken sich auf die Predictive Analysis Library (PAL), welche die Algorithmen für Data- und Text-Mining, sowie Textanalyse-Methoden zur Datenaufbereitung und Sentiment-Analyse, enthalten.
Aus der PAL wird ausschließlich ein Text-Mining-Algorithmus verwendet. Die anderen Algorithmen der PAL werden in dieser Arbeit nicht näher beleuchtet, da sie nicht für den vorgesehenen Anwendungsfall in Betracht gezogen wurden. Außerdem werden die Methoden der HANA zur Aufbereitung von unstrukturierten Daten erklärt und angewendet. Die Methoden der Aufbereitung von strukturierten Daten in der HANA-Umgebung werden nicht thematisiert. Die Prozesse des Text-Minings auf der HANA werden über die Client-Lösung SAP HANA Studio mit der Abfragesprache SQL entwickelt und ausgeführt. Ausschließlich einer kurzen

Vorstellung des SAP HANA Studios, werden keine weiteren technischen Eigenschaften benannt.
Auch wenn die HANA im Kern eine Datenbank darstellt, soll diese Arbeit nicht die gewöhnlichen Datenbankfunktionen bewerten. Neben den Text-Mining-Kompetenzen, werden lediglich technische Eigenschaften über die Architektur der HANA in zusammengefasster Form erklärt.

Die zu analysierenden Daten sind Social-Media-Beiträge einer Firma, welcher in der Handelsbranche tätig ist. Die Extraktion von Social-Media-Beiträgen geschieht über die bereitgestellte Schnittstelle der ausgewählten Social-Media-Plattform. Bestandteil der Arbeit ist jedoch nicht die funktionalen Eigenschaften der Schnittstelle zu untersuchen, sondern in erster Linie sie zur Datenextraktion zu verwenden.

Die SAP-Lösung Lumira Discovery soll in diesem Projekt ausschließlich dazu dienen, die generierten Daten bzw. Ergebnisse zu visualisieren und zu interpretieren.

1.3 Forschungsziele und Methodik

Die Ziele der vorliegenden Arbeit beinhalten primäre und sekundäre Komponenten.

Das primäre Ziel soll die Eignung der IMDB HANA als Entwicklungswerkzeug für Text-Mining bewerten und befasst sich mit den technischen Prozessen zur Durchführung des Projektes. Außerdem soll herausgestellt werden, in welchem Ausmaß mit SAP-Produkten, ohne die Berücksichtigung von Lösungen anderer Hersteller, Text-Mining-Projekte effektiv und effizient durchgeführt werden können. Die technische Bewertung wird hierbei für jeden Teilprozess des Vorgehensplans für Text-Mining (siehe Kapitel 2.3.2) durchgeführt.

Das sekundäre Ziel beschränkt sich auf die fachlichen Aspekte der Analyseergebnisse von Social-Media-Beiträgen. Es wird ein Text-Mining-Algorithmus, zur Findung von verborgenen Themen, auf textuellen Beiträgen ausgeführt. Anschließend werden die Kommentare der Social-Media-Beiträge verwendet, um eine Sentiment-Analyse auszuführen. Im Rahmen des sekundären Ziels werden die Ergebnisse der beiden Analysen evaluiert und Aussagen über die Qualität der Ergebnisse getroffen.

Das Projekt war sehr praxisorientiert und beruhte auf empirisch geprägte Arbeit. Zu Beginn wurden die Forschungsziele der Arbeit mit dem betreuenden Unternehmen festgelegt. Anschließend wurden erste Recherchen zur Aneignung eines Grundlagenverständnisses über IMDB und Text-Mining durchgeführt. Die Recherchequellen waren Literaturen, interne Quellen und Mitarbeiter des betreuenden Unternehmens, sowie wissenschaftliche Artikel und das Internet. Nach dem die Inhalte der Themen des Projektumfeldes besser aufgenommen wurden, musste eine Analyse der potentiellen Datenquellen für Unternehmen der Handelsbranche durchgeführt werden. Die berücksichtigten Datenquellen waren Facebook und

Twitter. Zur Auswahl einer geeigneten Datenquelle und eines Unternehmens wurden folgenden Fragestellungen beantwortet:

- Welche Nutzergruppen befinden sich auf den Plattformen?
- Verfügt das Unternehmen über genug Kunden, die über die Plattform mit dem Unternehmen kommunizieren?
- Sind die textuellen Daten Aussagekräftig?

Folglich konnte Facebook und auch einige Firmen aus der Handelsbranche als geeignete Social-Media-Plattform ausgewählt werden. Anschließend musste ein Anwendungsfall unter Berücksichtigung der möglichen Text-Mining-Kompetenzen der HANA definiert werden. Der Anwendungsfall war zunächst nicht auf ein Unternehmen zugeschnitten, definierte jedoch die anzuwendenden Algorithmen.

Danach wurden Informationen über die zur Verfügung gestellte Schnittstelle von Facebook gesammelt und der Antrag für einen Zugangsschlüssel eingereicht. Es folgte zur selben Zeit die Einrichtung der Programme, die im Projekt verwendet werden sollten. Die Verwendung der Programme Lumira Discovery und SAP HANA Studio wurden vorab bei Zieldefinierung mit dem betreuenden Unternehmen abgesprochen. Die HANA-Datenbank war schon auf einem Server aufgesetzt worden. Einige Programme zur Extraktion und anschließender Integration der Social-Media-Beiträge in die HANA-Datenbank wurden bewertet und die Lösung Spyder, eine Entwicklungsumgebung für Python, wurde ausgewählt.

Zudem wurden die recherchierten Grundlagen den wissenschaftlichen Anforderungen entsprechend dokumentiert (siehe Kapitel 2). Im Anschluss daran wurde die Analyse- und Planungsphase niedergeschrieben (siehe Kapitel 3). Zur selben Zeit musste im Laufe des Projektes mit Testdaten gearbeitet werden, da auf den Beantragungsprozess von Facebook gewartet werden musste und erste praktische Kenntnisse erlernt werden sollten. Aufgrund des ausstehenden Beantragungsprozesses war die Entwicklung sehr dynamisch. Anhand der Testdaten wurden die Programmcodes geschrieben, welche nach der Genehmigung des Zugangsschlüssels, an die zu extrahierenden Daten angepasst werden sollten. Die Testentwicklung orientierte sich an der Analyse- und Planungsphase. Der Prozess des Antrags auf den Zugangsschlüssel hat viel Zeit in Anspruch genommen. Aufgrund der Überschreitung der Zeitplanung wurde entschieden einen existierenden Datensatz der Infomotion GmbH zu verwenden.

Daraufhin wurde die Umsetzung (siehe Kapitel 4), die Social-Media-Beiträge aufzubereiten und zu analysieren, realisiert. Die Daten wurden in die HANA-Datenbank geladen, woraufhin die Datenexploration gemacht wurde. Inhalte der Tabellenwerte und die Attribute der Tabellen wurden daraufhin nachvollzogen. Danach wurden Prozesse zur Aufbereitung der textuellen Beiträge vorgenommen, welche, der Reihenfolge nach, die Segmentierung, Normalisierung, Wortstammbildung und Entfernung von Stopwords vornehmen sollten. Infolgedessen fand eine Ausführung des ausgewählten Text-Mining-Algorithmus und die Sentiment Analyse

nacheinander statt. Die Resultate wurden visualisiert und letztlich interpretiert. Die gewonnenen Erfahrungen in der Umsetzungsphase wurden im Anschluss in Kapitel 5 in Form einer kritischen Würdigung, hinsichtlich technischer und fachlicher Aspekte, niedergeschrieben.

1.4 Aufbau der Arbeit

Das Projekt beginnt zunächst mit der Vermittlung von Grundlageninhalte, die für die vorliegende Arbeit relevant sind und demnach werden am Anfang wichtige Begriffe eingeführt. Es werden In-Memory Computing (siehe Kapitel 2.1.1), Big Data (siehe Kapitel 2.1.2), Data-Mining (siehe Kapitel 2.1.3) und Text-Mining (siehe Kapitel 2.1.4) definiert. Zudem wird der Lebensmitteleinzelhandel angeführt, die die Domäne der Datengrundlage darstellt (siehe Kapitel 2.2).

In Kapitel 2.3.1 wird das Vorgehen im Data-Mining nach dem Knowledge Discovery in Databases (KDD) erklärt, wobei eine Einteilung in die Teilprozesse des KDDs gemacht wird. Somit werden die Teilprozesse des Vorgehens nach dem KDD in Selektion (siehe Kapitel 2.3.1.1), Vorverarbeitung und Transformation (siehe Kapitel 2.3.1.2), Data-Mining-Prozess (siehe Kapitel 2.3.1.3) und Evaluierung und Interpretation(siehe Kapitel 2.3.1.4) gegliedert und die anwendbaren Verfahren, sowie der Zweck und das Zusammenspielen der Teilprozesse näher beschrieben. Im Anschluss werden in Kapitel 2.3.1.5 Informationen über das Ziel des KDDs, die Wissensentdeckung, vermittelt.
Im Kontext des Text-Minings wird ebenfalls ein an das KDD angelehntes Vorgehensmodell vorgestellt (siehe Kapitel 2.3.2). Als Unterkapitel werden die Datenaufbereitungstechniken im Text-Mining erläutert (siehe Kapitel 2.3.2.1) und ein wichtiges Verfahren zur Umwandlung von Texten in eine repräsentative numerische Vektordarstellung dargelegt (siehe Kapitel 2.3.2.2).

Kapitel 2.4 wird im Rahmen des Grundlagenkapitels Informationen über die technische Plattform HANA einführen. Es folgt ein Kapitel 2.4.1 über die Historie und Entwicklung der SAP HANA, sowie einige Unterkapitel zu der architektonischen Zusammensetzung der HANA (siehe Kapitel 2.4.2). Nachfolgend werden die relevanten Verfahren zur optimierten Speichermethode (siehe Kapitel 2.4.2.1) und die Parallelverarbeitung in der HANA (siehe Kapitel 2.4.2.2) beschrieben. Des Weiteren werden im Kontext der HANA-Umgebung Bestandteile der Textanalyse (siehe Kapitel 2.4.2.3) und die Integrationsmöglichkeiten der Programmiersprachen R und Python (siehe Kapitel 2.4.2.4) charakterisiert.

Das Kapitel 2.5 gibt einen Einblick in die Entwicklungsumgebung des Projektes und legt die Gründe der Konstellation der Komponenten dar. Demnach wird das Programm SAP HANA Studio, welches die Entwicklung auf der HANA ermöglicht (siehe Kapitel 2.5.1) und eine allgemeine Beschreibung der Python-Entwicklungsumgebung Spyder, welches zur Extraktion von Social-Media-Daten aus Facebook dient, (siehe Kapitel 2.5.2) umrissen. Im weiteren

Verlauf werden die Lösungen Lumira Discovery, zur Illustration von Analysedaten (siehe Kapitel 2.5.3) und die Schnittstelle von Facebook (siehe Kapitel 2.5.4) vorgestellt.

Hauptkapitel 3 konzentriert sich auf die Kommunikation der Analyse- und Planungstätigkeiten des Text-Mining-Projektes auf Basis der HANA-Datenbank. Dabei orientiert sich die Chronologie des Kapitels an das Text-Mining-Vorgehen von Hippner und Rentzmann. Es wird das Ziel des Projektes formuliert (siehe Kapitel 3.1) und eine Recherche zur Auswahl der potentiellen Datenquelle vorgenommen (siehe Kapitel 3.2). Damit ein besseres Verständnis über die zu analysierenden Daten gewonnen werden kann, wird eine Datenexploration unternommen und Charakteristiken der Daten werden erläutert (siehe Kapitel 3.3). Daraufhin wird der Datenintegrationsprozess der Facebook-Beiträge analysiert und der Transport der Daten in die HANA näher spezifiziert (siehe Kapitel 3.4). Nachfolgend werden die Teilprozesse der Textdatenaufbereitung aufgezählt und die anzuwendenden Methoden der HANA zur Aufbereitung der Texte wiedergegeben (siehe Kapitel 3.5). Nach der Planung der Vorverarbeitungsmethoden der Social-Media-Beiträge, folgt eine detaillierte Erläuterung über die Verfahrensweise des Clustering-Algorithmus, Latent Dirichlet Allocation (LDA), welcher in der PAL enthalten ist und verborgene Themen in den Facebook-Beiträgen finden soll (siehe Kapitel 3.6). Anschließend werden in Kapitel 3.7 Methoden zur Extraktion von Sentiments und weiterer Entitäten eruiert, welche auf Kommentaren, die zu den Facebook-Beiträgen gemacht wurden, angewendet werden sollen (siehe Kapitel 3.7). Letztlich wird geplant, wie der Weg hin zur Visualisierung auf technischer Ebene umgesetzt werden soll (siehe Kapitel 3.8).

Das Hauptkapitel 4 hat den Fokus auf der Umsetzung und soll die Effektivität und Effizienz, der HANA-Kompetenzen im Kontext der geplanten Schritte zur Durchführung des Text-Mining-Projekts, aufzeigen. Zunächst wird die Integration der Facebook-Beiträge und Kommentare in die HANA vollzogen (siehe Kapitel 4.1). Anschließend wird die Vorverarbeitung ausschließlich der Facebook-Beiträge in der HANA-Umgebung vorgenommen (siehe Kapitel 4.2), woraufhin die Facebook-Beiträge mit dem Latent Dirichlet Allocation-Algorithmus (LDA-Algorithmus) der PAL auf verborgene Themen untersucht werden (siehe Kapitel 4.3). Letztlich werden die Kommentare mit dem Entity und Fact Extraction analysiert, wobei die Faktenextraktion die Sentiment-Analyse umfasst (siehe Kapitel 4.4). Demzufolge wird die Visualisierung der erzeugten Daten der Analysen realisiert und abgebildet (siehe Kapitel 4.5).

Das letzte Kapitel 5 des Hauptteils der Arbeit zeigt die Ergebnisse der wahrgenommenen Erfahrungen im Text-Mining-Projekt auf und bewertet die Ergebnisse der Unternehmungen im Hinblick auf die definierten Forschungsziele. Im Kapitel 5.1 werden die technisch orientierten Teilprozesse, insbesondere der Prozesse auf der HANA, gewürdigt, wohingegen Kapitel 5.2 die fachliche Bewertung der Visualisierungen der Analyseergebnisse vorsieht.

2 Grundlagen des Projektes

In den nachstehenden Unterkapiteln werden die relevanten Begriffe In-Memory Computing (siehe Kapitel 2.1.1), Big Data (siehe Kapitel 2.1.2), Data-Mining (siehe Kapitel 2.1.3) und Text-Mining (siehe Kapitel 2.1.4) erörtert. Anschließend werden einige Zahlen über einen spezifischen Bereich der Handelsbranche vermittelt (siehe Kapitel 2.2). Des Weiteren wird der Leser mit den Vorgehensmodellen im Data-Mining (siehe Kapitel 2.3.1) und Text-Mining (siehe Kapitel 2.3.2) vertraut gemacht. Dazu werden die aufeinander aufbauenden Bestandteile der beiden Vorgehensmodelle dem Leser im Einzelnen kommuniziert. Nach der Vermittlung allgemeiner Grundlagen wird im Anschluss die Technologieplattform HANA vorgestellt. Es wird auf die Historie und erreichten Entwicklungen der IMDB HANA eingegangen (siehe Kapitel 2.4.1), sowie die architektonischen Komponenten der HANA im Detail angeführt (siehe Kapitel 2.4.2). Darüber hinaus werden Einsatzszenarien der HANA aufgeführt und abgegrenzt, für welche Anwendungsbereiche die HANA nicht entwickelt wurde (siehe Kapitel 2.4.3). Letztendlich wird über die Entwicklungsumgebung des Projektes informiert und grundsätzliche Informationen über die Technologien und ihr Zusammenwirken vermittelt (siehe Kapitel).

2.1 Grundlegende Begriffe

Bevor die Inhalte der Ausarbeitung detaillierter erläutert werden, sollen zentrale Begriffe in diesem Kapitel vorab beschrieben werden. Die Beschreibung der Begriffe soll dem besseren Verständnis der nachfolgenden Kapitel dienen und bei der Einordnung der Themen helfen.

2.1.1 In-Memory Computing

In-Memory-Datenbanken (IMDB) nutzen die Möglichkeit des In-Memory Computings, um extensive Datenbestände nicht auf einer Festplatte, sondern dauerhaft im Hauptspeicher, dem RAM, zu persistieren. Die Datenbank ermöglicht aufgrund der Eigenschaft des RAMs eine schnelle Datenbereitstellung für Anwendungen in großen Unternehmen [Vgl. Silvia et al., 2017, S. 21]. Datenbanken mit Caching[1] können hingegen nicht dieselbe Reaktionszeit aufgrund des Vorhaltens der Datenmengen in den Festplatten erzielen. Dabei stellt die Festplatte den Hauptspeicher in der Datenbank dar. Grund dafür ist, dass der Schreib- und Lesezugriff auf Festplatten deutlich langsamer ist und der Cache immer neu verarbeitet werden muss [Vgl. Silvia et al., 2017, S. 21].

Auf der Hardware-Ebene wird zur Ausführung eines Prozesses die klassische Kombination von Komponenten verwendet. Die Kombination setzt sich aus Central Processing Unit (CPU), Hard Disk Drive-Speicher (HDD-Speicher) und einem Bus-System zusammen. Dabei verantwortet das Bus-System den Transport der Informationen zwischen CPU und dem Speichermedium. In

[1] *Caching dient dazu oft verwendete Informationen bzw. Daten in einem schnellen Puffer-Speicher zu halten. Die geforderten Daten können somit schneller abgerufen werden und müssen nicht erneut aus der langsamen Quelle geladen werden. Der Cache besitzt im Vergleich zum RAM kleinere Speicherkapazitäten.*

der Regel ist die zeitintensivste Komponente die Festplatte, welche einen Flaschenhalseffekt hervorruft. [Vgl. Grandpierre et al., 2013, S. 3]
Nachfolgend wird eine vereinfachte Hardwarearchitektur abgebildet:

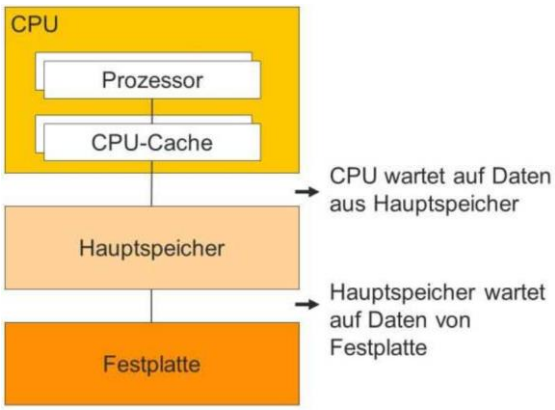


Abbildung 1: Datenengpass einer klassischen Hardwarearchitektur [Alterauge, o.J.]

In-Memory-Computing kann die Anforderungen der Echtzeit Datenabrufe bewerkstelligen, indem es folgenden Grundsatz befolgt:

> *„Der Datenzugriff wird beschleunigt und die Datenbewegungen werden minimiert" [Silvia et al., 2017, S. 23].*

Der Zugriff auf den RAM kann 100.000-mal schneller erfolgen im Vergleich zu herkömmlichen Festplattenspeichern [Vgl. Silvia et al., 2017, S. 23]. Die Geschwindigkeit resultiert aufgrund der Beschaffenheit des RAMs und der Haltung der gesamten Daten in dem RAM. Die CPU kann die im RAM abgelegten Daten direkt ansprechen und muss nicht darauf warten, dass die Daten zunächst in das RAM geladen werden müssen. Eine IMDB befähigt das Management binnen kurzer Zeit Berichte aufzurufen, anspruchsvolle Berechnungen und Analysen auszuführen, um auf interne und/oder externe Ereignisse zu reagieren.

IMDB sind aufgrund der Verwendung des RAMs als Speichermedium eine teure Ressource im Unternehmen. Außerdem wird das RAM vor allem in Rechenzentren, Konsolen, TVs und für Krypto-Mining eingesetzt [Vgl. Resch, 2018]. Die zu erwartende steigende Nachfrage von bis zu 25% für dieses Jahr im Vergleich zum Vorjahr erhöht wiederum den Preis der Speicherressource RAM [Vgl. Resch, 2018]. Allerdings kann eine bedarfsgerechte Skalierung von IMDB durch Cloud-Angebote in Betracht gezogen werden. Entscheidungsträger sollten nicht alle Daten pauschal in der HANA halten, zumal dieses Vorgehen viele monetäre Ressourcen binden würde. Vielmehr sollten nur oft angefragte Daten in der HANA existieren.

Im Folgenden sehen wir ein beispielhaftes Szenario, welches ein traditionelles Data Warehouse mit einem In-Memory-Computing anwendenden Data Warehouse vergleicht:

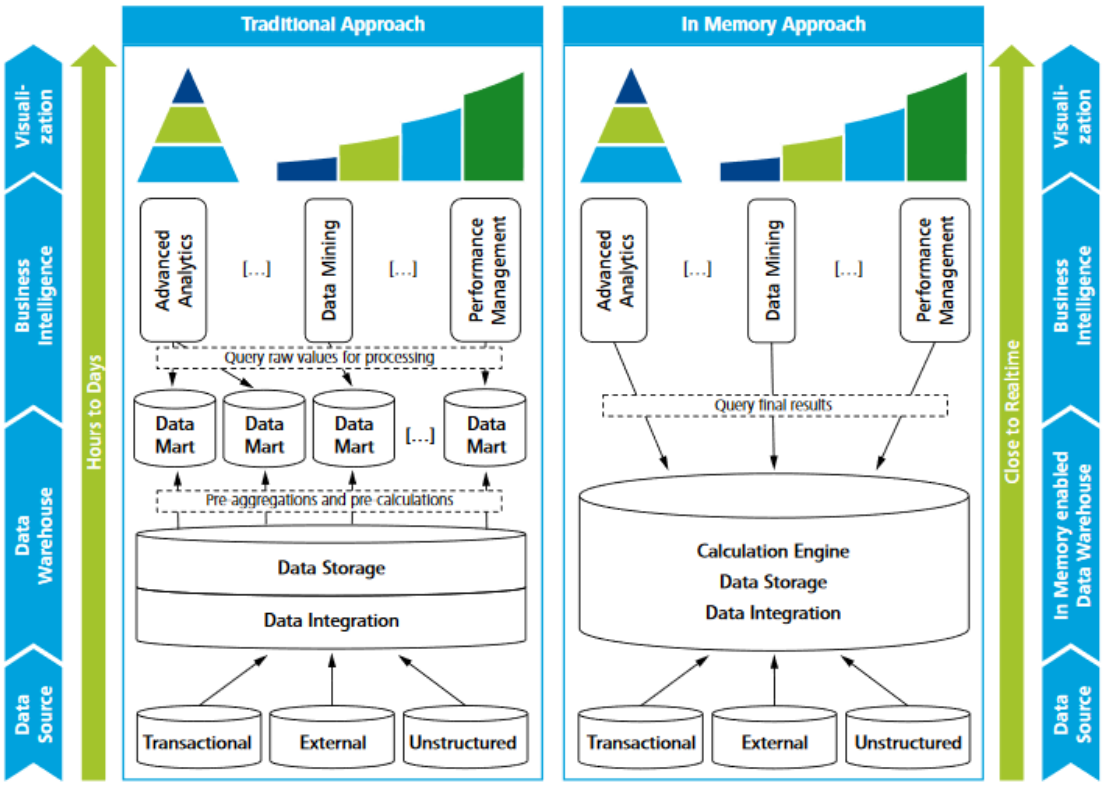


Abbildung 2: Beispielhafter Ansatz eines traditionellen Data Warehouses im Vergleich zu einem In-Memory getriebenem Data Warehouse [Grandpierre et al., 2013, S. 4]

Im traditionellen Data Warehouse werden in der Regel mehrere aufeinander aufbauende Datenhaltungsschichten oder Schemata eingesetzt. Die Daten werden im Rahmen des ETL-Prozesses (siehe Kapitel XXX) aus verschiedenen Quellsystemen extrahiert, transformiert und anschließend in die Speicherressourcen geladen. Auf den Datenspeichern setzen verschiedene Data Marts[2] auf. Letztlich werden einzelne Data Marts von diversen Reporting- und Analyse-Lösungen angefragt, um die Daten zu lesen und dem Anwender in aggregierter Form anzuzeigen. Je nach Größe und Komplexität der Infrastruktur der Systemlandschaft und Menge an Daten finden langwierige ETL-Prozesse zwischen den Datenquellen statt. Es gibt nur begrenzte Möglichkeiten in Echtzeit auf Marktveränderungen zu agieren.

Das In-Memory Data Warehouse wird auch von den Quellsystemen beladen, verwaltet alle Daten des Quellsystems jedoch in weniger persistenten Datenschichten. Diese Speicherschichten bestehen zudem aus dem schnellen RAM. Außerdem werden weitere Mechanismen, wie zum Beispiel Calculation Engines und Komprimierungsverfahren

[2] *Data Marts beinhalten Datenbestände, welche für dedizierte Abteilungen, Rollen oder Benutzer zugeschnitten sind. Die Daten sind den Anforderungen entsprechend transformiert. Zugriffsberechtigte greifen mit Applikationen auf die Data Marts zu, um ihre operativen Tätigkeiten (Analyse, Steuerung, Organisation, Reporting) auszuführen. [Vgl. Kemper et al., 2010, S. 26]*

eingesetzt, um höhere Geschwindigkeiten zu erreichen. Reporting- und Analyse-Lösungen können die rohen Schichten des In-Memory-Data Warehouse direkt anfragen und bekommen mit Hilfe der Calculation Engine der IMDB die angefragten Daten schnell in verarbeitetem Zustand zurück [Vgl. Grandpierre et al., 2013, S. 3]. Die anfragenden Berichterstellungslösungen können somit aktuelle Berichte und Analysen sofort generieren.

Während der traditionelle Ansatz Stunden bis Tage benötigt, um die entscheidungsrelevanten Daten zusammenzustellen, können In-Memory Data Warehouses die Datenbereitstellung innerhalb Sekunden realisieren. Die Geschwindigkeit kann mit der In-Memory-Technologie und zusätzlichen internen Mechanismen begründet werden. Außerdem kann die schnelle Reaktionszeit durch die Reduzierung der Datenhaltungsschichten und der einheitlichen Zusammenführung in eine zentrale Datenhaltungsschicht erklärt werden.

In Kapitel XXX folgen Informationen über die Architektur einer konkreten IMDB. Sie ist von SAP entwickelt worden und trägt den Namen HANA. Es wird charakterisiert, welche zusätzlichen Eigenschaften die HANA besitzt, die die traditionellen Datenbanksysteme nicht aufweisen.

2.1.2 Big Data

Die täglich produzierten extensiven Mengen an Daten sind für Unternehmen und ihre strategischen Ziele von großer Bedeutung.

> *„Über 90% der weltweiten Daten wurden zwischen 2009 und 2012 generiert und die Menge soll sich alle 18 Monate verdoppeln“ [Silvia et al., 2017, S. 29].*

Laut Definition des SAS Institute beschreibt der Begriff Big Data den Wachstum an strukturierten und unstrukturierten Daten und dessen Nutzung zum Vorteil der Unternehmensinteressen, indem tiefe Erkenntnisse zum Treffen bessere Entscheidungen und strategische Handeln gewonnen werden [Vgl. SAS Institute Inc., o.J.].

Daten im Kontext von Big Data sind nach Gartner hoch voluminös, vielfältig und/oder sammeln sich in einem hohen Tempo an. Zudem erfordert die Informationsverarbeitung von großen Datenbeständen einen kosteneffizienten und innovativen Umgang. [Vgl. Gartner Inc., o.J.]

Das Durchforsten der einzelnen Berichte ist nicht mehr stark im Fokus, was nicht heißen soll, dass die Berichte bedeutungslos sind. Jedoch sind Unternehmer nunmehr daran interessiert Kundenverhalten zu analysieren, zukünftige Szenarien vorherzusagen, Trends zu ermitteln und Zusammenhänge, sowie Muster in den Rohdaten[3] zu erkennen. Ziel ist es, der Konkurrenz einen Schritt voraus zu sein, um Wettbewerbsvorteile zu erringen. Gegenwärtige und in der Entwicklungsphase befindlichen Technologien der EDV und das Angebot an internetbasierten Dienstleistungen schaffen die Voraussetzungen, um Big Data-Projekte anzugehen. Data- und

[3] *Rohdaten sind im Sinne dieser Ausarbeitung, Daten die in ihrem ursprünglich erhobenen bzw. erzeugten Zustand sind und für Analyseprozesse zunächst vorverarbeitet werden müssen.*

Text-Mining sind zwei wichtige Disziplinen, in denen riesen Mengen an Daten analysiert werden und wertvolle Erkenntnisse aus Big Data gezogen werden.

Der Begriff Big Data wird auch als ein Sammelbegriff für digitale Technologien verwendet, die eine neue Art der Verarbeitung von Daten und digitaler Kommunikation ermöglichen, sowie einen gesellschaftlichen Umbruch im Hinblick auf soziale Aspekte verursachen. Demnach werden mit Big Data assoziierte EDV-Systeme nicht ausschließlich als neutrale und passive Lösungen wahrgenommen, sondern auch als Systeme, die die Gesellschaft und Menschen beeinflussen. [Vgl. Reichert, 2014, S. 9]

2.1.3 Der Data-Mining-Begriff

Knapp ausgedrückt ist Data-Mining eine Disziplin in der aus einer großen strukturierten Datenmenge Wissen extrahiert wird. Eine dem Ziel entsprechend angemessenere Bezeichnung für Data-Mining würde somit Knowledge Mining sein. Jedoch hat sich im Bereich der Datenanalyse die Bezeichnung Data-Mining durchgesetzt. [Vgl. Han und Kamber, 2010, S. 5]

Einer der führenden Forscher im Gebiet Data-Mining und Machine Learning definieren Data-Mining wie folgt:

> *„Data mining is the application of specific algorithms for extracting patterns from data.“ [Fayyad et al., 1996, S. 39]*

Der Data-Mining-Prozess behandelt nach Fayyad ausschließlich die Anwendung von Data-Mining-Algorithmen auf Daten. Data-Mining-Algorithmen nutzen Verfahren der Statistik, der künstlichen Intelligenz und des maschinellen Lernens, um Muster in Rohdaten zu erkennen [Vgl. Kotu, 2015, S. 3].

Nachfolgend werden kurze Beschreibungen der Anwendungsbereiche von Data-Mining-Algorithmen erläutert. Dabei werden ausschließlich zwei gegensätzliche Aufgabenbereiche behandelt:

- **Klassifikation**: Ein Klassifikationsmodell wird dazu genutzt, um neu registrierte Datenobjekte Kategorien zuzuordnen. Um das Modell anzulernen bzw. zu trainieren, werden bereits klassifizierte Datenobjekte verwendet. Da die Eingabetabelle klassifiziert ist, wird von überwachtem Lernen (supervised Learning) gesprochen. Zudem stellt das resultierende Klassifikationsmodell ein vorhersagendes Data-Mining-Modell dar, welches versucht einen Wert aus einer gegebenen Menge vorherzusagen (Predictive Data-Mining).
  Beispielsweise kann für das Abfangen von Spam-E-Mails ein trainiertes Klassifizierungsmodell angewendet werden, welches eingehende E-Mails als „Spam“ oder „nicht-Spam“ kategorisiert.
- **Clustering/Segmentierung**: Clustering-Algorithmen versuchen möglichst voneinander gut unterscheidbare Cluster von Datenobjekten zu bilden. Es wird die Maximierung der Ähnlichkeit der Datenobjekte innerhalb eines Clusters angestrebt.

Gleichzeitig wird die Minimierung der Ähnlichkeit der Datenobjekte in einem Cluster zu Datenobjekten anderer Cluster beabsichtigt. Im Vergleich zur Klassifizierung muss beim Clustering explorativ die optimale Anzahl an Clustern ermittelt werden, da keine klassifizierten Datenobjekte vorhanden sind. Clustering-Algorithmen müssen eigenständig Datenobjekte gruppieren, weswegen Clustering sich als unbewachtes Lernschema (unsupervised Learning) auszeichnet. Das Segmentierungsmodell ist ein beschreibendes Modell, da es in einer Menge von Datenobjekten versteckte Muster aufdeckt und ihre Charakteristiken beschreibt (Descriptive Data-Mining).
Eine Segmentierung kann beispielsweise Eigenschaften von Kunden eines Autoherstellers in Kombination mit den verkauften Automodellen analysieren. Das gewonnene Wissen kann Einsichten über bisher unbekannte Kundensegmente, die bestimmte Automodelle bevorzugen, ermöglichen.

Data-Mining wird oft mit Big Data (siehe Kapitel XXX) in Zusammenhang gebracht. Allerdings besteht eine klare Abgrenzung zueinander. Data-Mining beschreibt den Analyseprozess von kleinen und großen Datenmengen. Big Data hingegen beschreibt die Tatsache des rasanten Datenanstiegs von jeglichen Datentypen und bezieht sich auf die extensive Ansammlung von Datenmengen. Auch wenn Data-Mining-Verfahren häufig auf große Datenmengen angewendet werden, impliziert dies nicht, dass Data-Mining nur auf Big Data begrenzt ist. [Vgl. Litzel, 2016]

2.1.4 Der Text-Mining-Begriff

Gegenwärtig wird geschätzt, dass über 80% der Daten, die in einer Organisation anfallen, unstrukturierte Daten sind. Die meisten Textdateien sind von diesem Anteil. [Vgl. Laudon et al., 2010, S. 312]

Unternehmen sind daran interessiert, die großen Mengen an textuellen Daten zu analysieren und Erkenntnisse zu gewinnen.
Die Begriffe, “Text“ und “Wissen“, werden wie folgt miteinander in Zusammenhang gebracht:

> *„Text repräsentiert Wissen.“ [Heyer et al., 2008, S. 8]*

Schriften verschiedenster Arten gab es schon vor Jahrtausenden und konnten dazu beitragen, Wissen niederzuschreiben und persistent zu halten. In der heutigen Form haben Texte eine Struktur[4] und können einem gewissen (semistrukturierten) Textformat[5] unterliegen. Wir verwenden für jede Sprache eigenständige Grammatikregeln und können Schriften in digitaler Form festhalten. Linguisten versuchen zusammen mit Mathematikern Texte, unter

[4] *Die Struktur von Texten kann anhand von Sätzen, Absätzen, Kapiteln und Abschnitte beschrieben werden.*

[5] *Beispiele für Textformate: E-Mail, Post, Zeitungsartikel, Buch, Lebenslauf, Stellenausschreibung, Rechnung*.

Berücksichtigung der Grammatikregeln, für die Maschinen lesbar zu machen.
Die Bestrebung ist die enorm große Datenmenge in Form von Zeichenfolgen automatisiert zu verarbeiten, analysieren und Wissen zu gewinnen. Text-Mining heißt die Disziplin, die das Ziel verfolgt, die Wissensgewinnung zu ermöglichen. Sie wendet statistische und musterbasierte Verfahren an und ist ein Teilgebiet von Data-Mining.

Eine Definition von Text-Mining beschreibt, dass die Text-Mining-Verfahren linguistische Analysen anwenden:

> *„Mit dem Terminus Text Mining werden computergestützte Verfahren für die semantische Analyse von Texten bezeichnet, welche die automatische bzw. semi-automatische Strukturierung von Texten, insbesondere sehr großen Mengen von Texten, unterstützen." [Heyer et al., 2008, S. 3]*

Nach Tan ist Text-Mining eine Erweiterung von KDD oder Data-Mining und deckt eine Reihe von Disziplinen ab, um unstrukturierte Daten zu analysieren:

> *„Text mining, also known as text data mining or knowledge discovery from textual databases, refers generally to the process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents. It can be viewed as an extension of data mining or knowledge discovery from (structured) databases. (…) Text mining is a multidisciplinary field, involving information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning, and data mining." [Tan, 1999, S. 1]*

Nach der Definition von Tan ist der Begriff Text-Mining ein multidisziplinärer Bereich, welcher unter anderem auch die Textanalyse einschließt. Die Textanalyse ist die Anwendung von Analysetechniken, um maschinenlesbare Fakten zu extrahieren. Der Zweck der Textanalyse besteht also darin, eine strukturierte Repräsentation der unstrukturierten heterogenen Texte bzw. Dokumente zu erzeugen.

Data-Mining und Text-Mining haben **Gemeinsamkeiten und Unterschiede**. Text-Mining kann als eine Unterform von Data-Mining angesehen werden [Vgl. Geierhos, 2018]. Anders als beim Data-Mining, welches strukturierte Daten analysiert, nutzt Text-Mining freie Texte bzw. unstrukturierte Daten als Datenbasis. Wie oben erwähnt, haben Texte eine gewisse Struktur und folgen gewissen Grammatikregeln. Anhand der Struktur und Grammatikregeln können Vorverarbeitungsalgorithmen die unstrukturierten Daten in eine strukturierte Form überführen. Somit können die Daten einfacher verwaltet werden und sind in die Datenbankstruktur integriert, woraus neue Möglichkeiten der Verarbeitung und Analyse entstehen.
Text-Mining kann auch im KDD integriert werden. Wie beim klassischen Data-Mining, muss beim Text-Mining eine Vorverarbeitung der Texte, zwecks der Strukturierung, geschehen.

Anschließend werden im Text-Mining-Prozess auf mathematische, statistische und linguistische Grundlagen beruhende Algorithmen angewendet.

Die Anwendungsbereiche des Text-Minings sind breit und schneiden sich mit zwei Anwendungsbereichen des Data-Minings, Clustering und Classification [Vgl. Miner et al., 2012, S. 31]. Nachfolgend sind die Anwendungsbereiche im Text-Mining tabellarisch aufgeführt:

Anwendungsbereich	Aufgabenstellung
Information Retrieval	Speicherung und Abruf von Text Dokumente, Suchmaschinen und Schlüsselwortsuchen.
Document Clustering	Gruppierung und Kategorisierung von Termen, Textbruchstücken, Abschnitten oder Dokumenten unter Verwendung von Data Mining Clustering Verfahren.
Document Classification	Gruppierung und Kategorisierung von Textbruchstücken, Abschnitten oder Dokumenten unter Verwendung von Data Mining Klassifikationsverfahren.
Information extraction	Identifikation und Extraktion relevanter Fakten und Beziehungen; Erstellen strukturierter Daten aus unstrukturierten und semistrukturierten Daten.
Natural Language Processing	Einfache Sprachverarbeitungs- und erkennungsaufgaben (zum Beispiel Part-of-Speech Tagging).
Concept Extraction	Anordnung von Wörtern und Phrasen in semantisch ähnliche Gruppen.
Web Mining	Data und Text Mining im Internet mit speziellen Fokus auf die Vernetzung.

Tabelle 1:Anwendungsbereiche im Text-Mining [Miner et al., 2012, S. 32]

Die sieben Anwendungsbereiche des Text-Minings überschneiden sich in der Praxis. Viele praktische Text-Mining-Aufgaben liegen an der Schnittstelle mehrerer Anwendungsbereiche. So liegt zum Beispiel die Aufgabe „Entity Extraction“ (siehe Kapitel 3.7) an der Schnittstelle von „Document Classification“ und „Information Extraction“. Dabei wird das Dokument analysiert und mit vordefinierten Entitätstypen gekennzeichnet, die es erlauben nähere Informationen über den Inhalt des Dokumentes zu extrahieren und das Dokument zu klassifizieren. Es folgt ein Venn-Diagramm, welches die überschneidenden Text-Mining-Anwendungsbereiche und die Aufgaben der Anwendungsbereiche veranschaulichen:

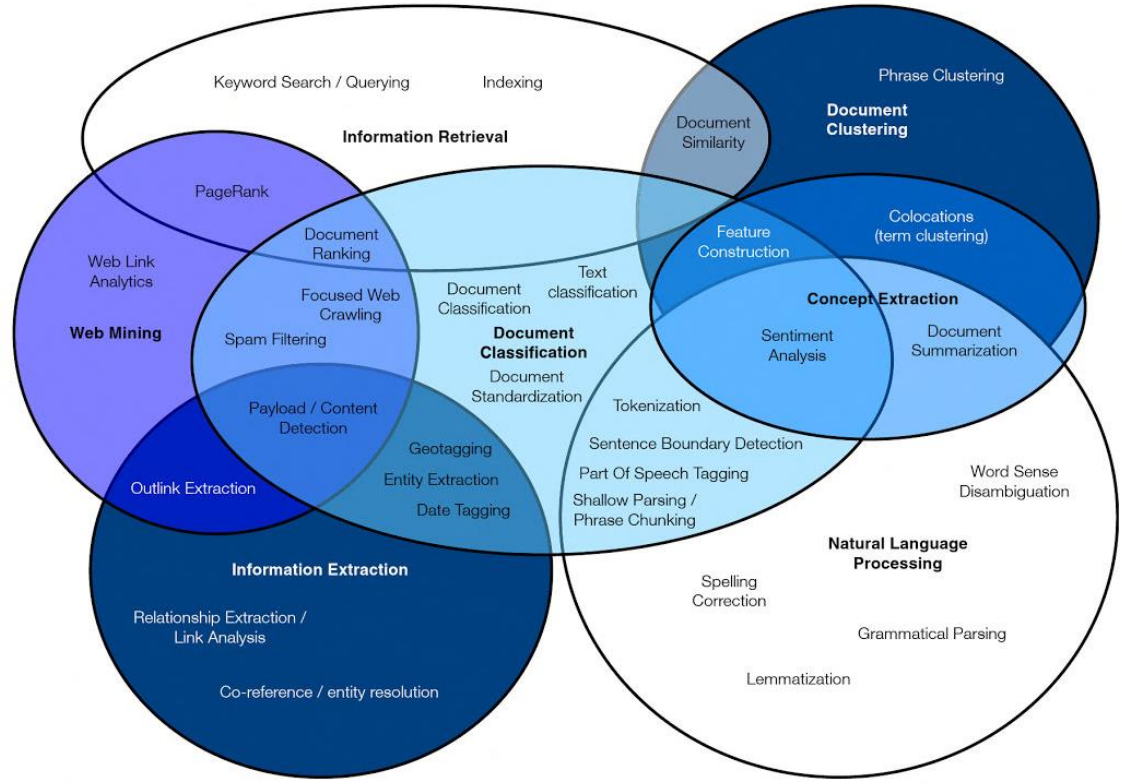


Abbildung 3: Überschneidungen der 7 Anwendungsbereiche des Text-Minings und ihrer Aufgaben (innerhalb der Kreise) anhand eines Venn-Diagramms [Miner et al., 2012, S. 38]

Aus den Anwendungsbereichen des Text-Minings ergeben sich auch interessante Einsatzszenarien, die Unternehmen in Geschäftsprozesse integrieren. Qualitative Text-Mining-Ergebnisse können anhand valider Datengrundlagen helfen richtige Entscheidungen zu treffen. Allerdings können auch Regierungen und öffentliche Einrichtungen die Technologien des Text-Minings nutzen. Es besteht auch die Option, dass einzigartige Wettbewerbsvorteile erzielt werden können. Es folgt eine Auflistung von Einsatzszenarien des Text-Minings:

- Pflege der Wissensmanagement-Datenbank, indem die Mengen an textuellen Daten kategorisiert, mit Schlagwörtern gekennzeichnet und archiviert werden.
- Risikomanagement, indem eine anormale Anhäufung von negativen Ereignissen über ein Produkt oder Dienstleistungen schnell erkannt werden und die Möglichkeit geben Maßnahmen zu treffen.
- Text-Mining als Internet-Kriminalprävention für Unternehmen, Strafverfolgungsbehörden oder Geheimdienste.
- Kontextbezogene Werbung auf Webseiten durch die Identifizierung der Inhalte der Webseite.
- Social-Media-Datenanalyse, um Kundenbedürfnisse zu analysieren, vorherzusagen zu treffen, die Wahrnehmung ihrer Marke zu verstehen, den Erfolg eingeführter Aktionen zu messen oder bisher unbekannte Informationen über die eigne Domäne zu entdecken.

2.2 Lebensmitteleinzelhandel

Nach dem Gabler Wirtschaftslexikon wird der Begriff Handel als solcher definiert, welcher „die Aufgabe [übernimmt], räumliche, zeitliche, qualitative und quantitative Spannungen zwischen der Produktion und der Konsumtion auszugleichen.“ [Hennig und Schneider, 2018].

Dabei geht es essentiell um den An- und Verkauf von hergestellten Produkten. Hierbei ist es möglich den Handel in zwei Elemente zu unterteilen, nämlich dem Groß-und Einzelhandel.

Während unter anderem die Lebensmittelbranche einen Teil des Handels beansprucht, werden die Lebensmittelkonzerne durch die fünf signifikanten Unternehmensgruppen, Edeka, Schwarz-Gruppe, Rewe-Group, Aldi und Metro gekennzeichnet.[6] Laut einer Recherche der Lebensmittelzeitung LZ Retailytics, nahmen diese 2017, rund 60 Prozent des Marktanteils in Deutschland ein. [Vgl. Deutscher Fachverlag GmbH, 2018]

Werden die Konsumausgaben der privaten Haushalte in Deutschland für Nahrungsmittel in den Jahren 1991 bis 2017 beobachtet, so kann festgestellt werden, dass diese um etwa 50 Milliarden Euro gestiegen sind. Demnach gaben die privaten Haushalte im Jahr 2017 in Deutschland rund 154,7 Milliarden Euro für Lebensmittel aus. [Vgl. Statista GmbH, 2018b]

Nach Angaben des statistischen Bundesamtes, betrugen die Ausgaben im Jahre 2016, pro Haushalt durchschnittlich 342 Euro im Monat für die Ernährung. [Vgl. Statistische Bundesamt, 2017]

In der Lebensmittelbranche gewinnen Tiefkühlprodukte immer mehr an Popularität. Durch die Einführung der tiefgekühlten Lebensmittel stieg der Pro-Kopf-Verbrauch an Tiefkühlprodukten fortwährend an. Neben der Flexibilität in der Küche spielte ebenso die saisonunabhängige Erreichbarkeit der vielfältigen Tiefkühlprodukte eine wichtige Rolle für diesen Anstieg. So konsumierte jeder Bundesbürger im Jahr 2017 rund 46,3 Kilogramm Tiefkühlprodukte. [Vgl. Deutsches Tiefkühlinstitut e.V., o.J.]

2.3 Analyseverfahren von strukturierten und unstrukturierten Daten

Der Knowledge Discovery in Databases-Prozess (KDD-Prozess) ist im Bereich der Datenanylse ein bewährter Vorgehensplan und wird im Folgenden im Kontext des Data- und Text-Minings thematisiert (siehe Kapitel 2.3.1 und 2.3.2). Dabei werden für beide Disziplinen die Teilprozesse des KDD näher erklärt, voneinander abgegrenzt, sowie der konkrete Analyseprozess des Data- und Text-Minings in den Vorgehensplan eingeordnet. Infolgedessen wird ein weit verbreitetes Verfahren zur Umwandlung von textuellen Daten in numerische Daten ausgelegt (siehe Kapitel 2.3.2.2).

[6] *Sortiert nach der Größe an Marktanteilen der Lebensmittelbranche.*

2.3.1 Vorgehen im Data-Mining nach Knowledge Discovery in Databases

In dem Artikel „From Data Mining to Knowledge Discovery in Databases“ wird Knowledge Discovery in Databases (KDD) wie nachstehend beschrieben:

> *“Knowledge Discovery in Databases describes the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.” [Fayyad et al., 1996, 40 f.]*

Nach Fayyad et al. ist der KDD-Prozess ein Vorgehen, um valides Wissen aus Daten zu generieren [Vgl. Fayyad et al., 1996, S. 39].
Eine Definition des Begriffes Wissen lautet folgendermaßen:

> *„Der Terminus Wissen bezeichnet dabei die meist auf Erfahrung beruhende und objektiv nachprüfbare Kenntnis von Fakten und Zusammenhängen eines Weltausschnitts, die Personen zur Lösung von Problemen einsetzten.“ [Heyer et al., 2008, S. 2]*

Die Wissensgenerierung im Kontext des Text-Minings wird mit produktiven Daten durchgeführt, die in der Vergangenheit gesammelt wurden. Die produktiven Daten bilden deshalb die Erfahrung ab. Der Algorithmus erzeugt, auf Basis der produktiven Daten, die Ergebnisse. Also bilden die Ergebnisse evaluierter Analysemodelle die Fakten und Zusammenhänge realer Geschäftsprozesse ab. Anschließend wird der generierte Output interpretiert und ggfls. visualisiert. Bei der Interpretation der Daten hat die Subjektivität keinen großen Einfluss, da das Analysemodell mit echten Daten trainiert wird und nicht durch Vermutungen und Einschätzungen von Menschen geprägt ist.

Außerdem vertreten Fayyad et al. die Meinung, dass Data-Mining einen konkreten Teilprozess im KDD umsetzt [Vgl. Fayyad et al., 1996, S. 39]. In der Praxis werden die Begriffe KDD und Data-Mining auch synonym genutzt [Vgl. Piatetsky-Shapiro, 2007, S. 100]. Der Ausdruck Data-Mining wird in der vorliegenden Arbeit als ein Teilprozess des KDDs verstanden, welcher Algorithmen zur Entdeckung unbekannter Muster in Daten anwendet.

Nachstehend wird der Vorgehensplan des KDD-Prozesses abgebildet und anschließend die Teilprozesse näher erläutert:

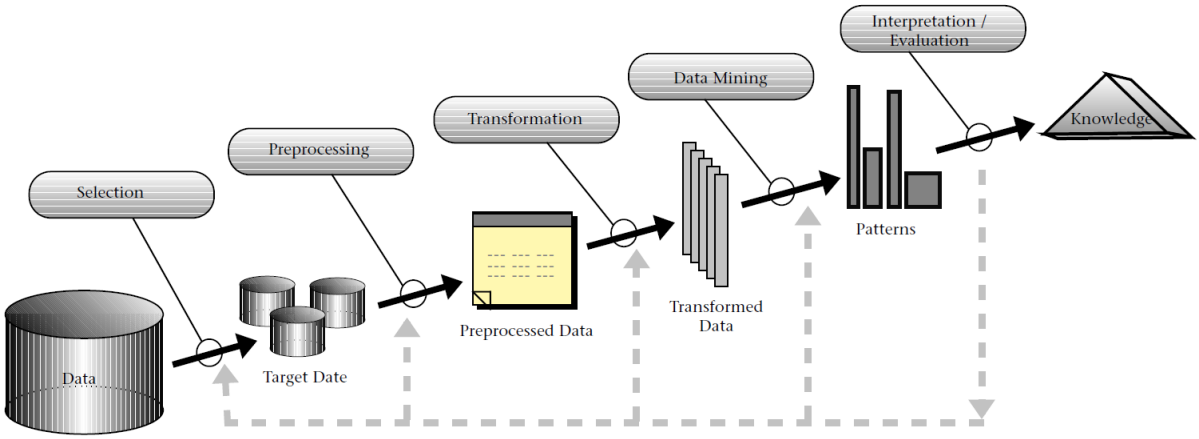


Abbildung 4: Data-Mining als Teilprozess im Knowledge Discovery in Databases [Fayyad et al., 1996, S. 41]

Wie aus der Abbildung hervorgeht, ist der Knowledge Discovery in Databases-Prozess ein iteratives Vorgehen. Dies liegt daran, dass Data-Mining-Algorithmen je nach Parametrisierung und Maß der validen Datengrundlage unterschiedliche Ergebnisse erzeugen. Data-Mining stellt somit eine explorative Disziplin dar, dessen Ergebnisse iterativ evaluiert werden müssen.

2.3.1.1 Selektion

Bevor mit der Selektion von zu analysierenden Daten begonnen wird, muss der Datenanalyst ein Verständnis über die Domäne der Projektumgebung entwickeln und Hintergrundwissen aneignen. Außerdem muss aus der Sicht des Kunden das Ziel der anstehenden Datenanalyse definiert werden. [Vgl. Chang und Chen, 2006, S. 309]

Daraufhin folgt die **Selektion** von relevanten Rohdaten aus einer oder mehreren Datenquellen. Die Selektion kann sich auf den gesamten Rohdatenbestand, auf eine Teilmenge des Rohdatenbestandes und/oder auf bestimmte Attribute von Tabellen beziehen. Die selektierten Rohdaten werden im Rahmen des KDD als Zieldaten bezeichnet.

Außerdem muss ein Unternehmen behutsam vorgehen, wenn personenbezogene oder sensible Daten zur Verarbeitung bzw. Analyse berücksichtigt werden. Die Datenanalyse darf nicht für den Erhebungsgrund abweichende Zwecke verwendet werden. Ohne besondere Umstände, spezielle Gesetzgebungen oder die Einwilligung der Betroffenen[7] (Grundprinzip Verbot mit Erlaubnisvorbehalt) [Vgl. Bitkom e. V., 2016, S. 7] sollte ein Unternehmen die Analyse solcher Daten unterbinden.

[7] *Diejenigen Personen, dessen Daten durch die Einwilligung bei einem Dienstleister erhoben und persistiert werden.*

2.3.1.2 Vorverarbeitung & Transformation

Nach der Selektion erfolgt der Prozess der **Vorverarbeitung** von den Zieldaten, welche die Aufgaben „(…) data cleaning, data integration, data reduction, and data transformation“ umfasst [Han und Kamber, 2010, S. 85].

Ziel der Vorverarbeitung ist es, qualitative Daten zu erzeugen. Han und Kamber definieren die Eigenschaften von qualitativen Daten wie folgt:

> *„Data quality is defined in terms of accuracy, completeness, consistency, timeliness, believability, and interpretabilty. These qualities are assessed based on the intended use of the data.“ [Han und Kamber, 2010, S. 120]*

Bei der **Datensäuberung** werden Geräusche[8], fehlende und inkonsistente Werte, sowie Ausreißer behandelt [Vgl. Han und Kamber, 2010, S. 85]. Das Wort Noise, im Kontext der Datenbeschaffenheit, wird von Han und Kamber wie folgt beschrieben:

> „(…) noise is a random error or variance in a measured variable.“ [Han und Kamber, 2010, S. 89]

Wir können also Geräusche als fehlerhafte Informationen ohne semantische Aussagekraft oder als Ausreißer ansehen. Ausreißer sind Anomalien in Datensätzen[9] [Vgl. Kotu, 2015, S. 25]. In dem Prozess der Datensäuberung werden Methoden verwendet, die die Varianz der Werte in den Daten und die fehlerhaften Werte glätten[10]. Das Glätten kann helfen, die Anzahl der eindeutigen Werte in einer Spalte zu verringern [Vgl. Han und Kamber, 2010, S. 90].
Sie können fehlerhaft sein oder einen Existenzgrund haben, der vom Menschen interpretiert werden muss [Vgl. Kotu, 2015, S. 25].
Die fehlenden Werte können behandelt werden, indem sie ignoriert oder manuell geändert werden [Vgl. Han und Kamber, 2010, S. 88]. Außerdem kann der Durchschnitt oder das Maximum der semantisch korrekten Werte, oder ein konstanter Wert für die fehlenden Werte eingesetzt werden [Vgl. Han und Kamber, 2010, S. 88].

Der **Datenintegrationsprozess** soll die unterschiedlichen Datenquellen, die dieselben Daten repräsentieren und Inkonsistenzen aufweisen, in eine einheitliche Datengrundlage überführen. Für unterschiedliche Spaltennamen, wie „Dokument_id“ und „Text_ID“, mit gleichem Inhalt wird nur ein Spaltenname verwendet, ein Datentyp definiert und nur eine Spalte weiterhin im Zieldatensatz aufgeführt. Es kann auch sein, dass eine Datentabelle den Vornamen und Nachnamen in getrennten Spalten aufführen und eine andere Datentabelle diese zwei Attribute zusammen in einer Spalte speichern. Hier gilt, sich auf eine Art der Repräsentation des Namens zu einigen.

---

[8] In Englisch noise genannt.

[9] Ein Ausreißer kann zum Beispiel ein ungewöhnlich großer Betrag beim Einkauf von Lebensmitteln sein, der über dem durchschnittlich zu zahlenden Betrag der Kunden liegt.

[10] In Englisch smoothing genannt.

Duplikate auf Zeilen- bzw. Tupelebene ergeben sich oft, wenn Tabellen in die dritte Normalform aufgelöst werden [Vgl. Han und Kamber, 2010, S. 98].
Inkonsistenzen zwischen Duplikaten entstehen häufig durch fehlerhafte Eingaben[11] von Daten [Vgl. Han und Kamber, 2010, S. 98]. Inkonsistenzen entstehen unter anderem aufgrund der Aktualisierung einer Tabelle, aber nicht einer anderen unmittelbar zusammenhängenden oder identischen Tabelle [Vgl. Han und Kamber, 2010, S. 98].

Der Prozess der **Datenreduktion** besteht darin, einen voluminösen Datensatz auf eine kleinere Größe zu reduzieren. Ziel der Reduktion ist es im Vergleich zum vollständigen Datensatz einen kleineren Datensatz mit gleichem analytischem Aussagegehalt zu erhalten. [Vgl. Han und Kamber, 2010, S. 86]

Die Selektion und Vorverarbeitung der Daten im KDD können bis zu 80% Aufwand für die Entwicklung eines robusten und zielführenden analytischen Modells darstellen [Vgl. Baesens et al., 2015, S. 27].

Der Prozess der **Datentransformation** wird angewendet, um die zu analysierenden Daten für die Auswahl eines Data-Mining-Algorithmus anzupassen und zu konsolidieren. Die Datentransformation umfasst unterschiedliche Strategien und leistet einen Beitrag zur Effizienzsteigerung der Analyselaufzeit, sowie zur besseren Verständlichkeit der resultierenden Datenmuster. [Vgl. Han und Kamber, 2010, S. 111]

[11] *Beispiel: falsche Formatierung, Tippfehler oder kein Eintrag von Daten.*

Die Strategien der Datentransformation sind nachstehend gelistet [Vgl. Han und Kamber, 2010, S. 112]:

a. Glätten der Daten, um Geräusche zu unterdrücken und Verzerrungen[12] der Werte zu verhindern.
b. Hinzufügen neuer Attribute bzw. Spalten, welche aus den gegebenen Datensätzen abgeleitet bzw. errechnet werden. Die neuen Variablen können erzeugt werden, wenn sie zur Zielerreichung beitragen.
c. Aggregationen von Datensätzen, falls auf einer höheren Ebene eine multidimensionale Analyse[13] durchgeführt werden soll.[14]
d. Normalisierungen von Variablen zur Umwandlung von Datensätzen mit einem großen Werteintervall in Datensätze mit einem kleineren Werteintervall.[15]
e. Diskretisierung ersetzt einzelne Werte in einem Datensatz durch Intervalle der Wertemenge.[16]
f. Konkrete Werte, die durch Intervalle der Wertemenge ersetzt werden, können auch hierarchisch geordnet werden, wenn mehr als eine Generalisierungsebene entwickelt wird.[17]

Die Aufzählungspunkte a bis c können sich mit den oben genannten Teilschritten der Vorverarbeitung, die Datensäuberung und Datenreduktion überschneiden [Vgl. Han und Kamber, 2010, S. 112 f.].

Für Informationen über die technischen Umsetzungsmöglichkeiten der Datenaufbereitung wird auf das Buch „Data Mining Concepts and Techniques“ von Han und Kamber verwiesen.

2.3.1.3 Data-Mining-Prozess

Bevor ein technischer Data-Mining-Prozess angestoßen wird, muss der Datenanalyst einen Algorithmus, unter Berücksichtigung des definierten Ziels, auswählen [Vgl. Chang und Chen, 2006, S. 310]. Dazu wird ein Data-Mining-Anwendungsbereich und anschließend ein passender Algorithmus aus diesem Bereich ausgewählt. Die gebräuchlichen Data-Mining-Anwendungsbereiche sind die Klassifizierung, Segmentierung, Regression, Assoziationsanalyse, Zusammenfassung und Sequenzanalyse [Vgl. Chang und Chen, 2006, S. 311]. Die Klassifizierung und Segmentierung wurden in Kapitel 2.1.3 näher erläutert. Der

---

[12] Unter dem Begriff Verzerrung ist der Effekt der Minderung der Relevanz bzw. Gewichtung aller Wörter im gesamten Dokument zu verstehen. Dies hat zur Folge, dass die Kerninhalte des Dokumentes nicht gut analysiert werden können.

[13] Unter dem Begriff multidimensionale Analyse wird das Einbeziehen mehrerer voneinander abhängigen Variablen im Data-Mining-Prozess verstanden.

[14] Beispiel: Täglichen Kurswert einer Aktie auf wöchentlichen Kurswert aggregieren.

[15] Beispiel: Bewertungsskala von 1 bis 100 auf den Wertebereich 0 bis 1 normalisieren.

[16] Beispiel: Eindeutige Gehälter in Intervalle von 10.000 bis 20.000, 20.001 bis 30.000, usw. generalisieren.

[17] Beispiel: Hierarchie bilden, indem der täglicher Umsatz auf die Generalisierungsebenen Monat, Quartal und Jahr aggregiert werden.

ausgewählte Algorithmus wird anschließend mit initialen Parametern parametrisiert und ausgeführt.

2.3.1.4 Evaluierung & Interpretation

Unter Berücksichtigung von Kennzahlen und/oder Visualisierungen[18], welche die Qualität des Modells beschreiben, wird das Resultat geprüft und gegebenenfalls eine erneute Ausführung des Algorithmus mit anderen Parametern angewendet.
Aus dem Resultat des Data-Mining-Prozesses wird versucht, Muster und Regeln zu erkennen, die diese im Kontext der Domäne interpretiert und Hypothesen erstellt. Je nach angewendetem Algorithmus, werden unterschiedliche Ausgabedaten erzeugt. Es kann auch die Möglichkeit bestehen, dass das Modell und die aufgedeckten Muster visualisiert werden können und den Prozess der Interpretation und Hypothesenbildung beschleunigen [Vgl. Chang und Chen, 2006, S. 310].
Die Evaluierung des Algorithmus hat einen technischen Schwerpunkt, wohingegen die Interpretation der Visualisierungen und der Modelle Hintergrundwissen über die Domäne erfordert, auf die anschließend Hypothesen aufgestellt werden. Wird die Evaluierung und Interpretation erfolgreich durchgeführt, kann anschließend eine Konfiguration von Parametern festgelegt werden.

2.3.1.5 Wissen

Alle Teilprozesse im KDD bauen aufeinander auf und sollen zum Ziel der Wissensgewinnung führen.
Das Wissen soll den Kunden befähigen, geeignete Handlungen bzw. Maßnahmen zu erarbeiten, um Vorteile oder Chancen wahrzunehmen, die unter anderem idealerweise einzigartige Wettbewerbsvorteile ermöglichen können. Im Folgenden wird gezeigt, wie das Wissen eines berühmten Datenanalyse-Ergebnisses aus Kanada verwendet werden konnte [Vgl. Alpar und Niedereichholz, 2000, S. 8]:

- **Analyseergebnis interpretieren und Hypothese entwickeln:** Das Ergebnis der Datenanalyse beschreibt eine Abhängigkeit zwischen den Produkten Babywindeln und Bier. Aus den Resultaten ist erkennbar, dass die Produkte Babywindeln und Bier oft zusammen gekauft werden. Diese Erkenntnis wird als Hypothese aufgestellt.
- **Wissen generieren:** Es wird darauf geschlossen, dass junge Väter beim Kauf von Bier von der Frau oft die Aufgabe bekommen, ebenfalls Windeln einzukaufen. Somit ist die Hypothese bewiesen.

---

[18] *Kennzahlen sind zum Beispiel numerische oder visuelle Resultate einer Berechnungsmethode, die die Vorhersagegenauigkeit eines Modells repräsentieren.*
Beispiel: Zur Evaluierung von Cluster-Modellen kann der der Silhouettenkoeffizient und der Silhouettenplot genutzt werden, um die Qualität der resultierenden Cluster zu evaluieren.

– **Handlungsoptionen erarbeiten:** Aus dem Wissen werden auf Handlungsmöglichkeiten geschlossen. Demnach wurden zwei Alternativen vorgeschlagen. Eine Alternative ist die nahe Anordnung der Babywindeln und das Bier, damit der Kunde bequem einkaufen kann. Die Zweite Alternative ist die entfernte Platzierung der zwei Produkte, damit der Kunde auf dem Weg andere Produkte wahrnimmt und eventuell weitere Produkte einkauft.

2.3.2 Vorgehen im Text-Mining angelehnt an Knowledge Discovery in Databases

Das Text-Mining-Vorgehensmodell von Hippner und Rentzmann ähnelt dem KDD-Vorgehensmodell des klassischen Data-Minings. Im Folgenden werden die Teilprozesse im Text-Mining illustriert und näher beleuchtet:

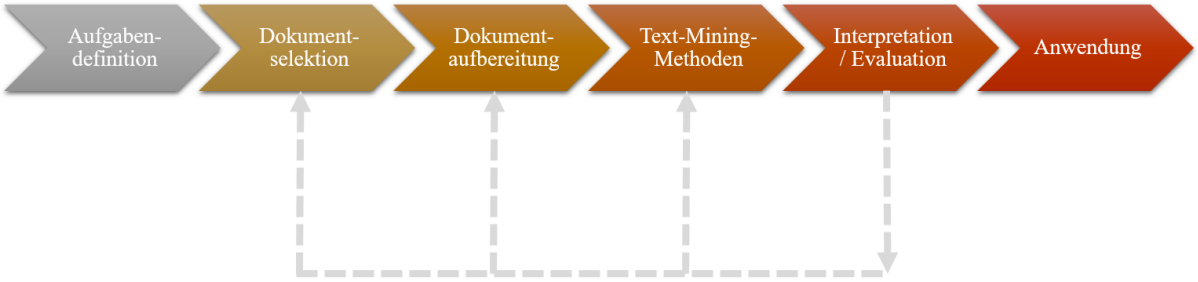


Abbildung 5: Text-Mining-Vorgehensplan (Quelle: Mit geringfügigen Veränderungen entnommen aus [Hippner und Rentzmann, 2006, S. 288])

Für das Ausführen einer erfolgreichen Textanalyse ist Hintergrundwissen über die Domäne von grundlegender Bedeutung. In der Abbildung des KDDs von Fayyad ist kein Prozess der **Aufgabendefinition** aufgeführt, wird jedoch implizit vorausgesetzt. Das Text-Mining-Vorgehensmodell nach Hippner und Rentzmann bildet den Schritt der Aufgabendefinition explizit ab. Das zu lösende Problem bzw. die Anforderungen werden ermittelt und daraus werden Ziele abgeleitet, die mit Text-Mining gelöst werden können [Vgl. Hippner und Rentzmann, 2006, S. 288].

Anschließend folgt die **Selektion** von Dokumenten. Sie geschieht unter Berücksichtigung des zuvor definierten Ziels der Textanalyse. Texte können aus internen[19] und externen[20] Quellen verwendet und zusammengetragen werden. Das Resultat dieses Prozesses ist eine Sammlung von Dokumenten, welches im weiteren Verlauf dieser Arbeit als Textkorpus bzw. Dokumentenkollektion bezeichnet wird.

Nach der Auswahl von geeigneten Dokumenten werden die Dokumente, welche unstrukturierte Daten darstellen, in der **Datenaufbereitungsphase** mittels Algorithmen der Merkmalsextraktion in strukturierte Daten transformiert. Dabei werden die Texte in einzelne

[19] *Zum Beispiel Kontaktanfragen und Fehlermeldungen.*

[20] *Zum Beispiel Texte aus Social-Media-Plattformen und öffentlich zugänglichen Datenbanken.*

Terme zerlegt und weiteren Methoden der natürlichen Sprachverarbeitung unterzogen (siehe Kapitel XXX(gesamte Textvorverarbeitungskapitel)).

Wird die strukturelle Repräsentation der Dokumente erzeugt, können in dem nächsten Teilprozess Text-Mining-Methoden verwendet werden, die das ursprünglich in der Phase der Aufgabedefinition definierte Ziel unterstützen. Text-Mining-Algorithmen können auf Dokument- oder Termbasis klassifizieren, segmentieren und/oder analysieren. Die breiten Anwendungsbereiche im Text-Mining wurden in Tabelle 1 vorgestellt.

Nach der Ausführung des Text-Mining-Algorithmus wird das resultierende Modell evaluiert und im Kontext der Domäne interpretiert, sowie Hypothesen gebildet. Ist die Evaluierung des Modells und die Interpretation nicht zielführend und nicht verständlich, wird das Text-Mining-Vorgehensmodell erneut durchlaufen. Ist die Parametrisierung des Algorithmus nicht gut gewählt, wird der Text-Mining-Algorithmus mit anderen Parametern erneut ausgeführt. Wird festgestellt, dass die Vorverarbeitung der Texte nicht qualitativ genug ist, muss die Vorverarbeitung erneut durchlaufen und anschließend der Text-Mining-Prozess ausgeführt werden. Schließlich wird nach mehreren Iterationen ein Text-Mining-Modell erzeugt, welches Einblicke in interessante, nicht-triviale Regeln und Muster ermöglicht und erlaubt sinnvolle Hypothesen zu bilden. Folglich wird die Wissensgenerierung unterstützt und das Text-Mining-Modell kann in die Geschäftsprozesse des Kunden integriert und angewendet werden.

2.3.2.1 Datenaufbereitungstechniken für Text-Mining

Die aufwändigste Tätigkeit in Text-Mining-Projekten zur strukturellen Repräsentation der Dokumente ist die Vorverarbeitung des Textkorpus. Der Text- und Data-Mining-Prozess aus den beiden Vorgehensmodellen ähneln sich. Allerdings unterscheiden sich beide Disziplinen in der Datenaufbereitung. Die Datenaufbereitungsmethoden des Text-Minings wenden statistische und linguistische Verfahren an.

Die Datenaufbereitung von Texten ist ein anspruchsvoller und aufwändiger Prozess. Unstrukturierte Daten liegen oft inkonsistent, nicht vollständig und mehrdeutig vor. Texte haben unterschiedliche Formen und können je nach Faktoren, wie Alter des Verfassers, Genre und Domäne, unterschiedlich aufgebaut sein. Von Nutzern geteilte Texte auf Social-Media-Plattformen sind oft in Alltags- und Umgangssprache, kurz verfasst und haben kein einheitliches Textformat. Im Vergleich dazu sind wissenschaftliche Ausarbeitungen oder Zeitungsartikel in Fachsprache geschrieben und unterliegen bestimmten Textformaten, die zum Beispiel Angaben über (Unter-)Titel, Datum, Autor und Genre enthalten. Zudem muss die Textdatenaufbereitung Abkürzungen, Emoticons, Ironie, Sarkasmus und gleichbedeutende Begriffe, die unterschiedlich geschrieben werden, erkennen und in einen strukturierten Datensatz umwandeln. [Vgl. Geierhos, 2018]

Nachstehend wird schematisch der Prozess der Datenaufbereitung abgebildet:

Abbildung 6: Vorgehensmodell der Datenaufbereitung im Rahmen des Text-Minings (Quelle: Eigene Darstellung)

Die unstrukturierten Daten liegen anfangs als Ansammlung von selektierten Dokumenten vor, welche die Zieldaten bzw. den Textkorpus darstellen. Der erste Schritt, der gemacht werden muss, ist die **Extraktion** der Wörter. Das Verfahren wird auf Englisch auch „Tokenizing" genannt. Alle Wörter und Interpunktionen aus den Dokumenten werden zerlegt[21]. Das Resultat sind einzelne Wörter und Interpunktionen, die Tokens genannt werden. Je nach Anforderungen wird in der Praxis jedes Wort, Wortpaare oder Tupel, bestehend aus drei Wörtern, bei der Extraktion zerlegt. Bei Schriftformen, die Leerzeichen verwenden, um die Sequenz an Wörtern in einem Satz zu kennzeichnen, kann das Leerzeichen als ein Trennzeichen dienen [Vgl. Miner et al., 2012, S. 47]. Wortpaare werden genutzt, wenn wichtige Begriffe, wie beispielshalber „Data Mining", extrahiert werden sollen, damit die Bedeutung des Wortpaars erhalten bleibt.

Anschließend folgt die Entfernung von so genannten **Stopwords[22]**. Stopwords sind sehr häufig vorkommende Wörter, die im gesamten Textkorpus verteilt sind. Typische Stopwords sind Artikeln, Präpositionen und Pronomen [Vgl. Vijayarani et al., 2015, S. 9]. Es wird die Annahme getroffen, dass Wörter, die in sehr vielen Dokumenten auftauchen, eine geringe Relevanz für die Beurteilung des Inhalts der Dokumente darstellen. Eine klassische Methode, Stopwords zu bereinigen, ist es eine Datentabelle mit Stopwords zu erstellen und anschließend mit den Tokens auf Gleichheit zu prüfen und ggfls. zu löschen. Laut duden.de sind die ersten zehn am häufigsten verwendeten Wörter in der deutschen Sprache *der*, *die*, *das*, *in*, *und*, *sein*, *ein*, *zu*,

[21] *Für die Zerlegung müssen nicht alle Interpunktionszeichen berücksichtigt werden. Möglicherweise beinhaltet ein Textkorpus viele Begriffe, welche mit Bindestrichen zusammengesetzt sind, wie beispielshalber Umsatzsteuer-Tabelle, Text Mining-Verfahren oder nicht-funktionale Anforderungen. Hier muss der Datenanalyst entscheiden, welche Interpunktionszeichen nicht berücksichtigt werden sollen.*

[22] *Auf Deutsch Stoppwörter.*

von, *haben*, *werden* und *mit* [Vgl. Bibliographisches Institut GmbH, o.J.]. Für Datenanalysten haben Stopwords keinen Aussagegehalt über die inhaltlichen Aspekte eines Textes. Das Ignorieren der Stopwords könnte das Ergebnis des Text-Mining-Prozesses verzerren und unschärfer machen, da statistische Methoden verwendet werden, um die Relevanz von Tokens zu bestimmen. Aus diesem Grund ist es eine Notwendigkeit Stopwords zu entfernen. Außerdem wird dadurch die Dimensionalität des Vektorraums der Terme reduziert [Vgl. Vijayarani et al., 2015, S. 9]. Die Reduzierung der Anzahl an Tokens im Textkorpus kann Speicherressourcen schonen und trägt zur schnelleren Verarbeitung von Daten bei [Vgl. Miner et al., 2012, S. 47].

Ist der Textkorpus extrahiert und von Stopwords bereinigt, folgt das Zurückführen der Wortausprägungen auf den Wortstamm bzw. die Grundform. Der Prozess wird **Stemming** genannt. Präfixes, Suffixes und Pluralisierungen der Wörter werden beim Stemming entfernt [Vgl. Miner et al., 2012, S. 47]. Dabei wird zum Beispiel das Genitiv der Tokens „Hauses“ oder „Häuser“ auf die Grundform „Haus“ zurückgeführt. Der Stemming-Prozess ist notwendig. Andernfalls würde jede grammatikalische Form von dem Wort „Haus“ im Vektorraum als voneinander unabhängige Wörter in die statistischen Berechnungen zur Ermittlung der Relevanz der Wörter in Dokumenten einfließen. Die Stemming-Methode nimmt an, dass die morphologische Form der Wörter eine semantische Abhängigkeit zueinander haben [Vgl. Vijayarani et al., 2015, S. 10]. Einige Stemming-Methoden nutzen eine Datentabelle, welche Stemming-Regeln zur Identifizierung von morphologischen Formen von Wörtern beinhalten. Das Zurückführen der Tokens in die Grundform ermöglicht in den Text-Mining-Anwendungsbereichen, Klassifizierung, Segmentierung und Suchindexierung, präzisere Ergebnisse zu generieren [Vgl. Miner et al., 2012, S. 48]. Außerdem werden durch Stemming ein weiteres Mal die Dimensionen verkleinert und infolgedessen die nachträgliche Ausführung des Text-Mining-Algorithmus verbessert [Vgl. Miner et al., 2012, S. 48].

Nach dem Stemming werden die Tokens normalisiert. Die **Normalisierung** nach Miner et al. ist in zwei Arten unterteilt. Spelling Normalization betrifft die Identifizierung von semantisch gleichbedeutenden Tokens, welche anders geschrieben werden [Vgl. Miner et al., 2012, S. 48 ff.]. Andere Schreibweisen ergeben sich zum Beispiel aus den Unterschieden zwischen amerikanischem und britischem Englisch, sowie aufgrund Tippfehler von Nutzern. Case Normalization beschreibt die Identifizierung unterschiedlicher Groß und- Kleinschreibungen und die anschließende Umwandlung aller Tokens in allen Dokumente in die Groß- oder Kleinschreibung [Vgl. Miner et al., 2012, S. 49 f.]. Die Normalisierungsprozesse der Datenaufbereitungsphase sind ebenfalls Maßnahmen, um den multidimensionalen Vektorraum der Tokens zu verringern. Außerdem ist der Zweck der Normalisierung die Tokens, die dieselbe semantische Bedeutung haben, zusammenhängend zu betrachten und in den Berechnungen der Text-Mining-Algorithmen zu berücksichtigen.

2.3.2.2 Umwandlung von vorverarbeiteten Texten in numerische Vektoren

Der Textkorpus bzw. die Dokumentenkollektion, welche in der Phase der Selektion unter Berücksichtigung des definierten Ziels entstanden ist, wird in der Datenaufbereitungsphase vorverarbeitet. Anschließend müssen die Tokens, welche zu Texten bzw. Dokumenten zugeordnet sind, in eine geeignete Repräsentation umgewandelt werden, damit sie mit dem numerischen Dateneingabeformat der Text-Mining-Algorithmen konform sind. Eine Vielzahl von Text-Mining-Algorithmen können nur mit numerischen Repräsentationen von Texten arbeiten.

Es gibt drei Möglichkeiten, um Dokumente als numerische Vektoren darzustellen. Es kann eine binäre, ganzzahlige oder Gleitkommawert-Repräsentation angewendet werden. Die Zusammenführung der Vektor-Repräsentationen der Dokumente in einer Tabelle wird im Kontext des Text-Minings als Term-Document-Matrix bezeichnet. Die Term-Document-Matrix bildet für Text-Mining-Algorithmen die Grundlage zur Ausführung der Operationen dar.

Im Folgenden werden drei Dokumente als Beispiel angeführt und in die genannten Vektor-Repräsentationen überführt, wobei Letztere eine mit TF-IDF gewichtete Vektor-Repräsentation ausdrückt. Die beispielhaften Dokumente im Textkorpus haben nur klein geschriebene Tokens. Jedoch enthalten die Dokumente Stopwords und die Tokens wurden nicht auf den Wortstamm reduziert, wobei nur eine grammatikalische Wortform der gleichbedeutenden Wörter existiert. Deswegen hat das Ignorieren des Stemming-Prozesses keine Auswirkungen auf die Genauigkeit des Ergebnisses. Das Ziel ist, dem Leser ausschließlich die Rechenmethoden zur Umwandlung der Dokumente in die Vektor-Repräsentation zu erläutern. Außerdem kann die Verzerrung der Gewichtungen nachvollzogen werden, da die Stopwords in den Dokumenten enthalten sind:

- **Dokument 1**: menschen können gute und böse dinge machen.
- **Dokument 2**: es gibt menschen die böse absichten haben.
- **Dokument 3**: es gibt gute menschen und böse menschen.

Die Dokumente werden als Vektoren dargestellt. Dabei stellen alle Tokens, die im Textkorpus existieren, einen Eintrag im Vektor bzw. eine Spalte in der Tabelle dar:

	es	gibt	die	menschen	können	gute	und	böse	dinge	haben	machen	absichten
Dok. 1	0	0	0	1	1	1	1	1	1	0	1	0
Dok. 2	1	1	1	1	0	0	0	1	0	1	0	1
Dok. 3	1	1	0	1	0	1	1	1	0	0	0	0

Tabelle 2: Beispiel der binären Vektor-Repräsentation von Dokumenten in einer Term-Document-Matrix (Quelle: Eigene Tabelle)

Die binäre Vektor-Repräsentation kann nur zwei Zustände abbilden. Die Eins wird für Tokens vergeben, die mindestens einmal im Dokument vorkommen. Wenn ein Token im Dokument mehr als zwei Mal vorkommt, ist dies allerdings nicht aus der binären Vektor-Repräsentation erkennbar. Die Null wird für Tokens vergeben, die im Dokument nicht vorkommen.

	es	gibt	die	menschen	können	gute	und	böse	dinge	haben	machen	absichten
Dok. 1	0	0	0	1	1	1	1	1	1	0	1	0
Dok. 2	1	1	1	1	0	0	0	1	0	1	0	1
Dok. 3	1	1	0	2	0	1	1	1	0	0	0	0

Tabelle 3: Beispiel einer ganzzahligen Vektor-Repräsentation von Dokumenten in einer Term-Document-Matrix (Quelle: Eigene Tabelle)

Die ganzzahlige Vektor-Repräsentation bildet die Anzahl für jeden Token innerhalb eines Dokumentes ab. Nun kann erkannt werden, dass das Dokument 3 wahrscheinlich ein Thema über „menschen“ behandelt. In der binären Vektor-Repräsentation konnten wir nur feststellen, welche Wörter in Dokument 3 vorkamen. Jedoch konnten wir nicht anhand der Häufigkeiten erkennen, welcher Token den primären Inhalt wiedergibt. Es gilt die Annahme, dass je öfter ein Token in einem Dokument vorkommt, desto höher ist die Wahrscheinlichkeit, dass dieser Token den Inhalt wiedergibt.

Ein Best Practise-Verfahren zur besseren Vektor-Repräsentation der Relevanz von Dokumenten über den gesamten Textkorpus ist das statistische Verfahren Term Frequency-Inverse Document Frequency (TF-IDF).
Die Annahme bei der TF-IDF-Rechenmethode wird wie folgt definiert:

> *Je höher der TF-Wert für ein Token ist, desto relevanter ist der Token. Es sei denn der DF-Wert ist auch hoch, denn dann wird der Token als ein gewöhnliches Wort angenommen. [Miner et al., 2012, S. 50]*

Die Term Frequency (TF) ist das Verhältnis der Anzahl eines Tokens d_t[23] in einem Dokument zu der gesamten Anzahl von Tokens d in dem Dokument. Die Division wird angewendet, da jedes Dokument unterschiedlich viele Tokens hat. Somit kann eine relative Vorkommens-Häufigkeit errechnet und ein angemessener Vergleich zwischen unterschiedlich langen Dokumenten bewerkstelligt werden:

$$TF = d_t / d$$

[23] *t ist ein konkreter Token aus dem Dokument d.*

Die Document Frequency (DF) beschreibt das Verhältnis der Anzahl der gesamten Dokumente D zu der Anzahl der Dokumente, die den Token D_t enthalten:

$$DF = D / D_t$$

Die Inverse Document Frequency wendet auf den DF-Wert den Logarithmus an. Die Verwendung der Logarithmusfunktion stellt sicher, dass Tokens, die in vielen Dokumenten vorkommen, eine geringe Gewichtung nahe 0 bekommen. Demnach lautet die Formel für IDF:

$$IDF = log_2 (D / D_t)$$

Letztendlich gibt der TF-IDF-Wert die Relevanz von einem Token in einem Dokument einer Dokumentenkollektion wieder. Der TF-IDF-Wert wird mit folgender Formel für jedes Token in jedem Dokument berechnet:

$$TF * IDF = d_t / d * log_2 (D / D_t)$$

Es wird das Beispiel von oben weitergeführt. Dokument 3 hat für den Token „menschen“ den TF-Wert 0,29[24] und DF-Wert 1[25]. Daraus ergibt sich für den Token „menschen“ der TF-IDF-Wert bzw. die Gewichtung 0. Das Ergebnis sagt aus, dass das Token gegenüber dem Dokument 3 im Textkorpus keine Relevanz darstellt.

Nachstehend werden die gewichteten Vektoren dargestellt. Je höher der TF-IDF-Wert eines Tokens ist, desto relevanter ist dieses Token für das Dokument in dem Textkorpus:

	es	gibt	die	menschen	können	gute	und	böse	dinge	haben	machen	absichten
Dok. 1	0	0	0	0	0,27	0,08	0,08	0	0,27	0	0,27	0
Dok. 2	0,08	0,08	0,27	0	0	0	0	0	0	0,27	0	0,27
Dok. 3	0,08	0,08	0	0	0	0,08	0,08	0	0	0	0	0

Tabelle 4: Beispiel einer mit TF-IDF gewichteten Vektor-Repräsentation von Dokumenten in einer Term-Document-Matrix (Quelle: Eigene Tabelle)

2.4 Die technische Plattform HANA

Die folgenden Unterkapitel behandeln die IMDB HANA und sollen auf fachlicher Ebene ein Grundlagenverständnis über die HANA erzeugen. Es wird auf die Entwicklung und Historie der HANA eingegangen (siehe Kapitel 2.4.1). Danach folgen Informationen über die Architekturkomponenten und ihr Zusammenwirken, um eine hohe Performance zu gewährleisten (siehe Kapitel 2.4.2). Letztlich werden Einsatzszenarien und Anwendungsbereiche, die die HANA nicht bedient, vorgestellt (siehe Kapitel 2.4.3).

[24] $d_t = 2, d = 7.$

[25] $N = 3, N_t = 3.$

2.4.1 Historie und Entwicklung der SAP HANA

Die HANA wurde ursprünglich für die Ausführung von Echtzeit-Analysen implementiert und war als Appliance[26] verfügbar. Aufgrund dessen entstand der Name „High Performance Analytic Appliance“. Gegenwärtig ist die HANA in verschiedenen Varianten als Produkt verfügbar und findet nicht nur für Analysezwecke Anwendung. Die HANA wird unter anderem auch auf der Cloud bereitgestellt. Im Laufe der Zeit hat sich das Appliance-Modell geändert. Die HANA wird seither als alleinstehender Markenname verwendet und hat keinen Bezug zur ursprünglichen Abkürzung. [Vgl. Gahm et al., 2016, S. 31]

Erste Entwicklungen der SAP HANA fingen im Jahr 2008 an. SAP hatte in Kooperation mit der Stanford University und dem Hasso-Plattner-Institut angefangen eine Datenbank auf Basis der In-Memory-Technik zu entwickeln. [Vgl. Litzel, 2017]
2011 wurde die HANA erstmals als eine revolutionierende Datenbank am Markt eingeführt [Vgl. Schmitz, 2015]. Anschließend begann SAP, die erfolgreiche IMDB HANA strategisch in ihr Produktportfolio zu integrieren. Den Kunden sollten die Performancevorteile der HANA in Zusammenhang mit den bestehenden SAP Modulen offeriert werden. Somit kam 2012 „SAP BW powered by SAP HANA“[27] auf den Markt, welches Data Warehousing, Reporting und Analysen auf Grundlage der HANA-Datenbank ermöglichte [Vgl. Litzel, 2017].
Im Jahre 2013 kündete SAP die HANA Plattform an [Vgl. Litzel, 2017]. Seither ist die HANA mit der In-Memory-Technik der Kern der Entwicklungs- und Technologieplattform von SAP. Auf dieser Plattform werden SAP- Geschäftsanwendungen betrieben und entwickelt.

Zudem sind SAP-Anwendungen auch in der Cloud. Die Cloud baut auf der HANA auf und heißt HANA Cloud Plattform (HCP). Softwareprodukte anderer Hersteller können auch die Plattform verwenden, um Nutzen aus den Vorteilen der IMDB, HANA zu ziehen.
Mit der Weiterentwicklung der HANA, zur Beschleunigung von Anwendungen, verfolgt SAP die Strategie eigene Lösungen zu entwickeln und etablieren, statt Unternehmen und ihre Lösungen aufzukaufen [Vgl. Wiehr, 2017].

Lizensiert ein Kunde die HANA, kann er auch von den Zusatzfunktionalitäten profitieren. Um das wertvolle Wissen in den großen Mengen an strukturierten und unstrukturierten Rohdaten zu extrahieren, werden auf der HANA-Plattform Funktionen bereitgestellt, die den Umgang mit Big Data erleichtern. Vorausschauende Analysen mit Hilfe von Data-Mining-Algorithmen, sowie Textanalysen bzw. Analysen von unstrukturierten Daten sind einige dieser Werkzeuge, die direkt auf der HANA ausgeführt werden können. [Vgl. Wiehr, 2017]

Die Themen, Cloud-Lösungen und HANA werden seitens der SAP medial oft betont. Als seriöser Softwarehersteller geht das europäische Softwarehaus mit den Herausforderungen von

---

[26] *Appliance bezeichnet eine eng miteinander abgestimmte Kombination von Soft- und Hardware, um einen konkret abgegrenzten Anwendungsfall zu realisieren [Vgl. Martins und Kobylinska, 2018].*

[27] *Der Name wurde nachträglich geändert und lautet nun „SAP NetWeaver Business Intelligence“.*

Big Data seit langem um und hat seine Kompetenzen erweitert. Die HANA ist als Appliance, Cloud- und Hybrid-Lösung in die IT-Infrastruktur des Unternehmens integrierbar. Die HANA, die das RAM als Datenspeicher verwendet, hat die Möglichkeit eines schnellen Data Warehousings und einer Echtzeit-Datenverarbeitung geschaffen.

2.4.2 Architektur der HANA In-Memory-Datenbank

Die HANA ist eine spaltenbasierte relationale Datenbank und funktioniert somit hybrid. Grundsätzlich geschieht die Verarbeitung der Daten im Arbeitsspeicher, dem RAM, da die HANA eine In-Memory-Datenbank ist. [Vgl. Begerow, o.J.]

In diesem Kapitel wird auf die Architektur der HANA-Datenbank und dem Zusammenspiel der Komponenten und Techniken eingegangen. Die Eigenschaften der Komponenten werden aufzeigen, wie eine starke Zugriffs- und Verarbeitungsgeschwindigkeit erzielt wird. Im Folgenden sind die wesentlichen Architekturkomponenten schematisch veranschaulicht und erörtert:

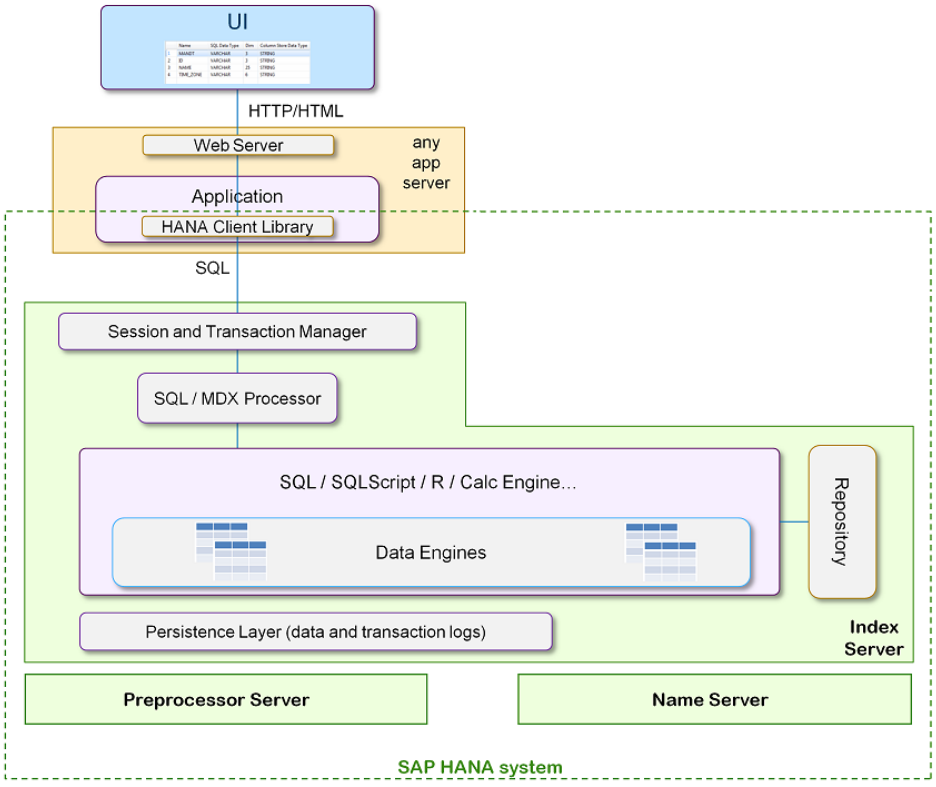


Abbildung 7: Die SAP HANA-Datenbank Architektur [SAP SE, 2018c, S. 11]

Der Speichertyp RAM ist ein flüchtiges Speichermedium. Wird die Stromzufuhr zum RAM unterbrochen, werden die Daten nicht gespeichert und sind auch nicht wiederherstellbar. Die **persistente Schicht** ist für diesen speziellen und geschäftskritischen Fall vorhanden. Werden Transaktionen angestoßen und Daten verändert, werden die betroffenen Seiten im RAM markiert und es findet ein Schreibvorgang in das nichtflüchtige Speichermedium statt [Vgl. Silvia et al., 2017, S. 25]. Die Logdateien der Transaktionen sollen die Dauerhaftigkeit der

Transaktionen sicherstellen und beinhalten alle Änderungen, die an der Datenbank gemacht werden [Vgl. Silvia et al., 2017, S. 25].

Die Topologie der HANA-Datenbank werden in dem **Name Server** verwaltet. Der Name Server registriert, wo die Hosts liegen[28] und welche Daten die Hosts verwalten. [Vgl. Gahm et al., 2016, S. 60]

Mit dem Update der HANA auf SPS9 hat SAP die Möglichkeit gegeben, innerhalb einer HANA-Instanz voneinander isolierte Datenbanken zu betreiben und verwalten. Diese Funktion ist interessant, wenn unabhängige Anwendungen eine eigene Datenbank auf einem HANA-System haben sollen. Auf diese Weise können Ressourcen individuell verwaltet werden und die Wartung der Datenbank wird übersichtlicher. SAP nennt diese Betriebsart Multitenant Database. Für Cloud-Umgebungen ist diese Eigenschaft sehr wertvoll. Bis auf die Datenbankinhalte bzw. Datenbankschemata teilen sich die Teilnehmer in der Multitenant Database die CPU, das Betriebssystem und die Datenbanksoftware. [Vgl. Gahm et al., 2016, S. 59]

Der **Präprozessor-Server** wird vom Index-Server aufgerufen, wenn mit Textdaten operiert werden soll [Vgl. SAP SE, 2018c, S. 12]. Er kann Textdaten analysieren und individuell konfigurierte Textsuche-Prozesse anstoßen [Vgl. SAP SE, 2018c, S. 12]. Sie ist für die Nutzung der Text-Mining- und Textanalyse-Funktionen die Schlüsselkomponente.

Der **Index Server** ist die wichtigste Architekturkomponente in der HANA-Infrastruktur. Über die HANA-Client-Library können jegliche Applikationsserver mit der HANA kommunizieren. Der SQL-Parser nimmt SQL-Befehle entgegen und kann in Zusammenarbeit mit dem Row- oder Column-Store die Abfrageergebnisse liefern [Vgl. Gahm et al., 2016, S. 59]. Aktive im RAM-Speicher befindliche Daten können vom Column- und Row-Store sofort abgerufen werden [Vgl. Gahm et al., 2016, S. 59]. Anfragen an die HANA können auch über die Datenbanksprache Multidimensional Expressions (MDX) gemacht werden [Vgl. SAP SE, 2018c, S. 12].

Anfragen, die komplexe Berechnungen erfordern, werden von anwendungsfallspezifischen Engines weiterverarbeitet. Die Engines können je nach Anwendungsfall nachträglich durch die Lizensierung einer höheren HANA-Version erweitert werden. Wesentliche Engines im Index Server sind die Calculation-, Planning-, MDX- und Stored Procedure-Engines [Vgl. Gahm et al., 2016, S. 59].

Die HANA Architektur erlaubt statistische Berechnungen mit Skripten der Programmiersprache R innerhalb der HANA-Umgebung zu deklarieren und aufzurufen. Als

[28] *Von einer Topologie im Kontext der HANA-Architektur wird unter der Voraussetzung, falls die HANA dezentral betrieben wird, gesprochen.*

Bestandteil der Calculation Engine können benutzerdefinierte R-Operatoren verwendet werden. [Vgl. SAP SE, 2018e, S. 5 f.]

Der **Sitzungsmanager** initialisiert und verwaltet die Sitzungen, sowie die Datenbankverbindungen. Der **Transaktionsmanager** koordiniert ausgeführte Transaktionen und registriert laufende und abgeschlossene Transaktionen. Die resultierenden Änderungen aufgrund der Transaktionen werden den Speicher-Engines, also dem Row- oder Column-Store, mitgeteilt, die anschließend weitere Prozesse anstoßen. [Vgl. tutorialspoint, o.J.]

2.4.2.1 Das spalten- und zeilenbasierte Speichermodell

Die HANA zeichnet sich im Vergleich zu den klassischen zeilenorientierten Datenbanken performanter aus, da es die Daten in Spalten organisiert. Außerdem verfügt die HANA über effiziente Kompressionsmethoden. Daraus ergeben sich technisch Vorteile, die sich in der Geschwindigkeit wiederspiegeln. Nachteile hat das spaltenbasierte Speichermodell auch, jedoch werden diese durch interne Verfahren auf technischer Ebene minimiert. Die HANA unterstützt beide Speichermethoden, ist jedoch für die spaltenorientierte Speicherung optimiert [Vgl. SAP SE, 2018a, S. 12].

Nachstehend wird die zeilen- und spaltenorientierte Speicherung illustriert:

Abbildung 8: Datenmodell der zeilen- und spaltenorientierten Speichernutzung [Alterauge, o.J.]

Die **zeilenorientierte Speichermethode** legt jede Zeile in der Tabelle nacheinander in einer Folge im Row-Store des Index Servers ab.
Die **spaltenorientierte Speichermethode** speichert die in einer Spalte vorliegenden Werte einer Tabelle im Column-Store des Index Servers sequenziell ab.

Im Folgenden wird eine tabellarische Ansicht der Vor- und Nachteile beider Speichermodelle gezeigt:

	Zeilenbasierte Speicherung	Spaltenbasierte Speicherung
Vorteile	Daten werden zusammenhängend gespeichert und können einfach eingefügt bzw. aktualisiert werden.	Nur die relevanten Spalten werden beim Auswahlprozess gelesen, und jede Spalte kann als Index oder Schlüssel zum Datenabruf dienen.
Nachteile	Bei der Auswahl müssen alle Daten gelesen werde.	Datenaktualisierungen sind bei der spaltenbasierten Speicherung nicht so effizient wie bei der zeilenbasierten Speicherung.

Tabelle 5: Vergleich der Vor- und Nachteile der spalten- und zeilenbasierten Speicherung [Silvia et al., 2017, S. 35]

Der Nachteil der zeilenorientierten Speicherung kommt zustande, da alle Zeilen samt den zusammenhängenden Attributen gelesen werden müssen bis die WHERE-Bedingung wahr ist. Hier kann der spaltenorientierte Ansatz über die in der WHERE-Bedingung angegebenen Spalte sofort suchen und wird schnell fündig. Außerdem gelingt das Aggregieren der Daten schneller.

Die spaltenorientierte Speicherung hat bei der Aktualisierung von Datensätzen folgenden Nachteil: Es muss erstmal die richtige Spalte und anschließend die richtige Zeile gefunden werden, um einen Wert zu aktualisieren. Die HANA behebt diesen Nachteil mit einem zeilenbasierten Deltapuffer[29]. [Vgl. Silvia et al., 2017, S. 35]

Wie oben erwähnt, ist die HANA für die spaltenbasierte Speicherung optimiert. Eine Spalte in einer Tabelle muss nicht indexiert werden, da das Speichern von Daten spaltenbasiert und sequenziell geschieht [Vgl. SAP SE, 2018a, S. 12]. Außerdem nutzt die HANA Kompressionsverfahren und kann die Datenverarbeitung noch schneller ausführen. Diese Kompressionsmethoden heißen run-length encoding, cluster coding und dictionary coding [Vgl. SAP SE, 2018a, S. 12].

Dictionary coding speichert die Spalten, die einen String-Datentyp haben, in eine bit-kodierte Sequenz ab, so dass Vergleichsoperatoren über ganzzahlige Werte ausgeführt werden können [Vgl. SAP SE, 2018a, S. 12]. Ein Vergleich über String-Werte wäre deutlich Ressourcenbelastender.

2.4.2.2 Parallelverarbeitung

Durch das Nutzen der spaltenoptimierten Speichermethode ergibt sich für HANA die Gelegenheit effiziente Parallelverarbeitungen von Daten zu exekutieren. In Kombination mit den oben erwähnten Kompressionsverfahren werden zusätzlich höhere Geschwindigkeiten erzielt.

Operationen wie Joins, Aggregationen und Lesezugriffe können parallel ausgeführt werden. Die spaltenorientierten Datensätze können auf mehrkernigen Prozessoren verteilt werden. Muss

[29] *Detaillierte Informationen über den Deltapuffer und ihre Verfahrensweise sind im Buch „SAP HANA: Die neue Einführung" nachzulesen.*

eine Spalte aggregiert und eine andere durchsucht werden, können beide Prozesse auf jeweils einem Kern des Prozessors verteilt werden. Um das volle Potenzial einer CPU auszuschöpfen, können große spaltenbasierte Daten auf mehrere Kerne verteilt werden, so dass die Daten verteilt werden und von mehreren Prozessorkernen verarbeitet werden können. Wird ein dezentrales HANA-System betrieben, können die Daten über mehrere CPU-Instanzen verteilter Hosts verarbeitet werden [Vgl. Sonnenschein, 2013]. Parallele Verarbeitungen zeilenorientierter Daten sind hingegen nicht geeignet, da die Daten der angefragten Spalte über den gesamten Hauptspeicher verteilt und die Werte einer Spalte nicht sequenziell angeordnet sind. [Vgl. SAP SE, 2018a, S. 13]

Nachstehend wird schematisch die Verteilung von spaltenorientierten Daten auf einer vierkernigen CPU abgebildet:

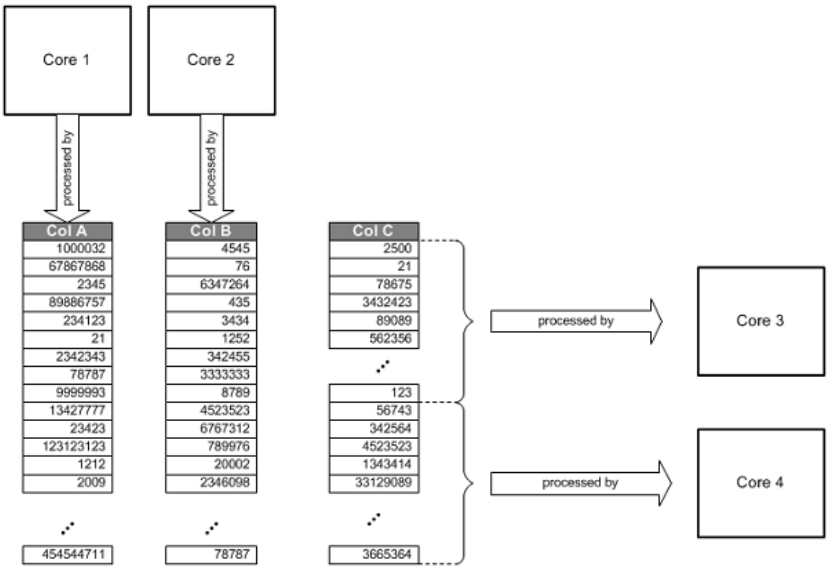


Abbildung 9: Schematische Darstellung der Parallelverarbeitung spaltenorientierter Daten auf der HANA [SAP SE, 2018a, S. 13]

2.4.2.3 Text Analytics-Umgebung der HANA

SAP unterscheidet die zwei Begriffe Textanalyse und Text-Mining beim Vorgang der Generierung von Wissen aus Textdaten mit der HANA-Infrastruktur. Es soll betont werden, dass diese Begriffsunterscheidung im Sinne der HANA-Umgebung für die Entwicklung von Text-Mining-Projekten eine Bedeutung hat und abgegrenzte Funktionalitäten darstellen.

Die Textanalyse-Funktionalitäten behandeln den Anwendungsfall der Vorverarbeitung der Textdaten (siehe Kapitel 3.5). Des Weiteren sind die Extraktionen von Fakten und Sentiments, sowie die Erkennung von Grammatikbestandteilen, Prozesse der semantischen Textanalyse. Die Funktionalitäten des Text-Minings spiegeln im HANA-Kontext die Umwandlung der Texte in Zahlen wieder (siehe Kapitel 2.3.2.2) und bewirken das Löschen von Stopwords. Der Zweck der Umwandlung besteht darin, die Anwendung von PAL-Algorithmen der HANA zu ermöglichen. Nahezu alle PAL-Algorithmen nehmen ausschließlich numerische Eingabedaten an.

Es folgen generelle Aussagen über die Kompetenzen spezieller Bibliotheken in der HANA-Umgebung.

In der HANA können für abgegrenzte Anwendungsgebiete speziell für die IMBD-Architektur optimierte funktionale Bibliotheken installiert werden. Diese werden je nach anzuwendender Operation in den verschiedenen Engines des Index Servers verarbeitet [Vgl. SAP SE, 2018a, S. 14 f.]. Die Anwendungsfunktionen sind in C++ geschrieben und können über das SQL-Skript als Prozedur aufgerufen werden [Vgl. SAP SE, 2018d, S. 8]. Im Prozeduraufruf müssen als Parameter die Eingabetabellen und die Ausgabetabellen angegeben werden. Nach der Exekution der Prozedur werden die Resultate in Form von physische Tabellen generiert.

Die **Application Function Library** (AFL) von SAP beinhaltet zwei Bibliotheken, die eine Gruppe von Funktionen mitliefern, welche für abgegrenzte Einsatzzwecke genutzt werden können:

- Die **Business Function Library** (BFL) ist eine Anwendungsbibliothek, die viele Geschäftsfunktionen im finanziellen Bereich abdeckt und leistungskritische Aufgaben auf der HANA meistert. Jährliche Abschreibungen, interne Zinsmethode und gewichteter Durchschnitt sind einige der anwendbaren Funktionen. [Vgl. Gahm et al., 2016, S. 518 f.]
- Die **Predicitve Analytics Library** (PAL) beinhaltet eine Gruppe von Funktionen für fortgeschrittene Analysen. Mit der PAL können Klassifikations-, Clustering-, Zeitreihen- und Assoziationsfunktionen auf umfangreichen, numerischen und kategorischen Datenmengen angewendet werden.
  Die PAL ist ein Set von Werkzeugen, welches die Geschwindigkeit der IMDB ausnutzt und komplexe Rechnungen auf der HANA ermöglicht. Big Data Tätigkeitsbereiche, wie Data-Mining und vorhersagende Analysen können auf der HANA zuverlässig umgesetzt werden. Bewährte Algorithmen, wie der K-Mean, Decision Tree und Apriori, sind integriert und für viele Anwendungsfälle nutzbar. Damit vertrauenswürdige Resultate erzielt werden und angemessen umfangreiche Eingabedaten erstellt werden können, sind Algorithmen für die Vorverarbeitung von Eingabedaten in der PAL integriert. Diese sind Algorithmen, wie Binning, Partitition, Principal Component Analysis und Substitute Missing Values.

Das **Document Analysis Toolkit** im Präprozessor-Server führt die Analysen und Extraktionen auf Textdaten aus [Vgl. Gahm et al., 2016, S. 458]. Damit das Document Analysis Toolkit die unstrukturierten Daten bzw. Textdaten analysieren kann, muss zuerst auf eine spaltenorientierte Textspalte ein so genannter Full-Text-Index angewendet werden. Der Full-Text-Index zerlegt

jeden Satz in Wörter[30] und indexiert diese. Dieser Prozess stellt die Vorbedingung zum Arbeiten mit Textdaten dar und findet in dem Column-Store des Index Servers statt [Vgl. Gahm et al., 2016, S. 458]. Vor dem Erstellen des Full-Text-Indexes können Parameter spezifiziert werden, auf die im späteren Kapitel 4.2 eingegangen wird.

Auf Grundlage der Tokens kann die Text-Engine der HANA die Textsuche[31] und -analyse auf unstrukturierten Daten anwenden. Die Text-Engine befähigt Nutzer dazu Textsuchmodelle zu entwickeln und semantische Textanalysen durchzuführen.
Semantische Textanalysen können die Bedeutung der Tokens erkennen. SAP nennt diesen Vorgang Entity Extraction (siehe Kapitel 3.7). Die Tokens werden den vordefinierten oder von den Entwicklern spezifizierten Themengebieten zugeordnet. Beispielhafte Themengebiete[32] sind Topic, negative/neutral/positive Sentiment, Request, URL, Organization und Locality.

2.4.2.4 R- und Python-Integration auf der HANA

Die in der Calculation Engine integrierte R-Komponente bietet Erweiterungen für statistische Analysen. So können umfangreiche Bibliotheken der Programmiersprache R genutzt werden und als SQL-Skript auf der HANA gestartet werden. Die gestartete Prozedur übergibt den R-Code und die Eingabedaten an die externe R-Umgebung und wartet auf die Antwort. Der Rückgabetyp der Antwort ist ein Data Frame[33]. [Vgl. SAP SE, 2018e, S. 5 f.]

Python kann auch dazu verwendet werden, um bestimmte Aufgaben zu übernehmen, die die HANA nicht bereitstellen kann. Python verfügt über eine große Community und viele Bibliotheken, die unter anderem auch Einsatzszenarien des Text-Minings unterstützen. Dabei fungiert die Bibliothek pyhdb in der Programmiersprache Python als Schnittstelle. Die deklarierte Verbindungsinstanz der pyhdb erstellt eine Datenbanksitzung und kommuniziert über die Java Database Connectivity-Schnittstelle (JDBC-Schnittstelle) mit der HANA.

2.4.3 Einsatzszenarien der In-Memory-Datenbank HANA

SAP hat den Einsatzzweck der HANA nicht nur auf das Verwalten und Bereitstellen von Produktivdaten beschränkt. SAP versteht die HANA unter anderem als eine Technologie, die die Voraussetzung für schnelles Verarbeiten von Daten für breite Anwendungsfelder schafft. Aufgrund der Vielzahl an Funktionen, die auf der HANA laufen und die weitere Einbeziehung von Anwendungen überflüssig machen, ergeben sich verschiedene Einsatzszenarien. Der strategische Einsatz der HANA-Umgebung kann Unternehmen Chancen aufzeigen und

---

[30] Im Bereich Text Analytics werden die Wörter, welche aus der Zerlegung der Dokumente/Texte resultieren, als Token bezeichnet.

[31] Nähere Informationen zur Textsuche sollen bewusst nicht aufgegriffen werden. Der Fokus liegt auf den Textanalyse- und Text Mining-Funktionen der SAP HANA.

[32] Die Elemente in einer Menge von Themengebieten werden in diesem Anwendungskontext von der SAP als Entitäten betitelt.

[33] In der R-Umgebung ist Data Frame eine Liste von Vektoren, die gleich lang sind. Ein Data Frame akzeptiert unterschiedliche Datentypen.

Wettbewerbsvorteile schaffen. Die Fähigkeiten von HANA können in verschiedenen Bereichen ausgeschöpft werden. Hierzu gehören Handel, Finanzwesen, Telekommunikation, Fertigung, öffentlicher Verkehr, Vertrieb und Forschung/Entwicklung.

Allgemeine Anwendungen, die auf Basis von HANA realisiert werden, sind Echtzeitanalysen, Data Warehousing, Erstellung oder Migration optimierter Applikationen für HANA, sowie die Nutzung der HANA als Accelerator [Vgl. Gahm et al., 2016, S. 61].

Von einem HANA basierenden Accelerator[34] wird gesprochen, wenn Anwendungen aufgrund der begrenzten Verarbeitungskapazität rechenintensive Anfragen an die HANA auslagern und die Vorteile der Leistung und Skalierbarkeit proaktiv nutzen. Der datenlastige Vorgang wird durch Replikation der Daten in den Hauptspeicher bzw. RAM der HANA ausgeführt. Bei der Verwendung der SAP Business Suite Anwendung kann ein Mechanismus die großen Tabellen kennzeichnen und zukünftige Ressourcenlastige Anfragen an die HANA übergeben. [Vgl. SAP SE, 2018b, S. 10 f.]

Einsatzszenarien der HANA sind vielfältig und mit fortschreitender Entwicklung der Algorithmik und Technologien werden in Zukunft weitere Einsatzszenarien hinzukommen. Nachkommend werden in Unternehmen praktizierte Einsatzszenarien aufgeführt [Vgl. Silvia et al., 2017, S. 45]:

- Verkaufsortanalyse
- Qualitäts-und Produktanalyse
- Radio-Frequency Identification-Tracking und -Analyse
- Betrugs- /Risikomanagement und –modellierung
- Prognosemodellierung
- Was-wäre-wenn-Szenarien
- Preisoptimierung
- Analyse von Kundenanforderungen und -gewohnheiten

Die in Kapitel 2.4.2.3. genannten Funktionsbibliotheken machen die HANA zu einer primären Komponente in der Infrastruktur von Einsatzszenarien. Neben anderen sekundären Programmen, wie ETL- und Visualisierungslösungen, kann die HANA das formulierte Einsatzszenario in nahezu Echtzeit in ein profitables Resultat überführen. Beispielsweise kann die HANA zusammen mit der PAL Anwendungsfälle mit Data-Mining-Techniken eigenständig lösen.

[34] *Auf Deutsch Akzelerator oder Beschleuniger .*

Nachdem ein Grundverständnis über die Potentiale der HANA erarbeitet wurde, soll nun aufgezeigt werden, welche Anwendungszwecke die HANA nicht unterstützt [Vgl. Silvia et al., 2017, S. 45 f.]:

- Die HANA stellt im Kern eine Datenbank dar, welche auf den vorhandenen Daten Abfragen und Operationen ausführen kann. Damit Daten in die HANA geladen werden können, müssen ETL-Werkzeuge, wie SAP Data Services oder diverse andere ETL-Lösungen, genutzt werden.
- Die HANA unterstützt keine integrierte Funktion, um Berichte zu erstellen. Diese Funktionalität kann speziell für die Berichterstellung entwickelte SAP Produkte, wie SAP Business Objects, oder Berichterstellungslösungen anderer Hersteller, wie Sisense, bereitstellen.
- Die HANA ist kein Werkzeug für die Modellierung von Daten.
- Die HANA ist nicht gleichzusetzen mit dem Modul von SAP ERP, welches ein Transaktionssystem darstellt.
- Die HANA kann auch nicht mit dem Modul SAP Business Warehouse (SAP BW) verglichen werden. SAP BW nutzt hochgradig strukturierte Datenmodelle und basiert auf dem Sternschemakonzept.
- Die HANA erfüllt nicht die Anforderungen für das Betreiben von Qualitätsmanagement.

2.5 Entwicklungsumgebung

Die Infrastruktur der Entwicklungsumgebung des Projektes sieht wie folgt aus:

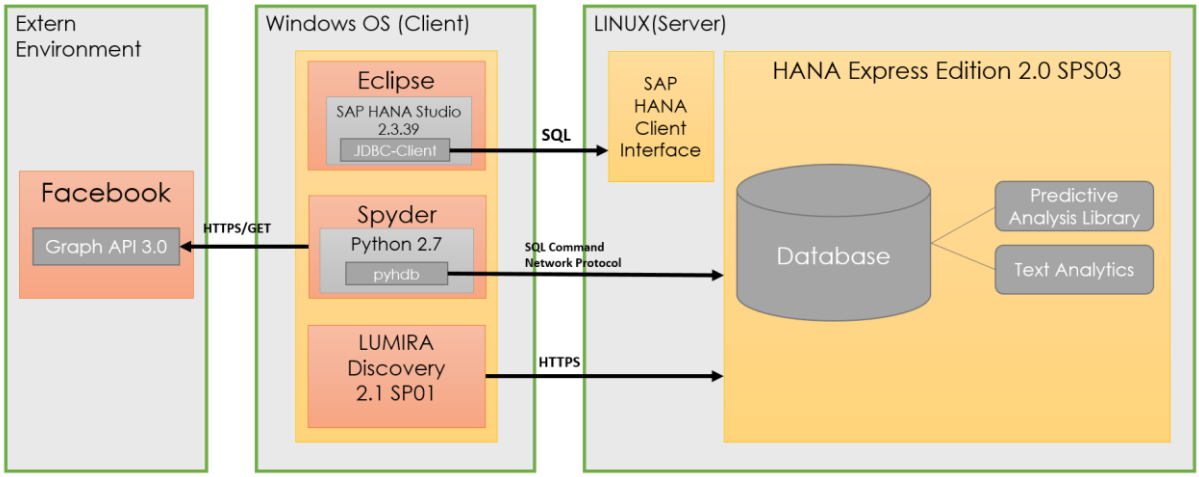


Abbildung 10: Entwicklungsumgebung des vorliegenden Text-Mining-Projektes (Quelle: Eigene Darstellung)

Das HANA-System befindet sich auf einem Linux-Server. Die installierte Version der HANA ist die HANA Express Edition 2.0 SPS03. Die HANA Express Edition ist eine weniger umfangreiche Version der HANA. Einige spezielle Funktionen werden nicht auf der HANA Express Edition unterstützt, jedoch verfügt sie über alle Funktionalitäten, um Data und Text-Mining-Analysen auszuführen. Wie in der Abbildung zu erkennen ist, sind die PAL und die Funktionalitäten für Text Analytics und Data-Mining verfügbar.

Die Clientinfrastruktur wird auf einem Windows Betriebssystem betrieben. Auf dem Client werden die Entwicklungsumgebungen Eclipse (siehe Kapitel 2.5.1) und Spyder (siehe Kapitel 2.5.2), sowie die Datenvisualisierungs- und Analysesoftware Lumira Discovery von SAP (siehe Kapitel 2.5.3) ausgeführt. Eclipse kommuniziert über Java Database Connectivity (JDBC) unter Verwendung der Abfragesprache SQL mit der HANA-Datenbank. Spyder spricht die HANA über die Bibliothek pyhdb, unter der Nutzung der SQL Command Network Protocol, an. Die Kommunikation zwischen Lumira Discovery und HANA wird über HTTPS[35] gewährleistet. Über die Entwicklungsumgebung Spyder wird per HTTPS eine GET-Anfrage an die Schnittstelle Graph API[36], welches von Facebook bereitgestellt wird, übergeben (siehe Kapitel 2.5.4). Das Resultat der GET-Anfrage wird anschließend zurückgegeben.

Da die HANA-Datenbank in Kapitel 2.4 ausführlich behandelt wurde, werden die Komponenten der Linux-Umgebung nicht weiter detailliert. Es folgen Unterkapitel mit Informationen über die restlichen Komponenten in der Entwicklungsumgebung.

2.5.1 Das SAP HANA Studio und die Eclipse-Plattform

Eclipse stellt ein quelloffenes Framework dar, welches sich als eine Plattform für Entwicklungsumgebungen und -werkzeuge etabliert hat und eine einheitliche Benutzeroberfläche bietet. Eine Eigenschaft, die Eclipse auszeichnet, ist die Möglichkeit verschiedene Entwicklungswerkzeuge in eine Installation zu bündeln. Daraus folgt der Mehrwert einer homogenen Entwicklungsumgebung für den Nutzer. [Vgl. Gahm et al., 2016, S. 75 f.]

SAP ist Mitglied der Eclipse Foundation und hat das strategische Ziel, Eclipse als neue Umgebung zur Entwicklung von SAP-Lösungen zu nutzen. SAP HANA Studio ist eines der Werkzeuge, welches auf der Eclipse-Plattform verfügbar ist. Mit SAP HANA Studio kann die HANA-Datenbank administriert und entwickelt werden. Beispielsweise können mit SAP HANA Studio Datensichten[37] modelliert und Datenbankprozeduren entwickelt werden. [Vgl. Gahm et al., 2016, S. 34 f.]

Data und Text-Mining-Funktionalitäten und Textanalysen werden in SAP HANA Studio über SQL aufgerufen. Die ausgeführten Prozeduren erzeugen physische Ausgabetabellen und können im SAP HANA Studio angesehen werden. Zudem können Konfigurationen und Parametrisierungen der Vorverarbeitungs- und Text-Mining-Methoden vorgenommen werden.

---

[35] *Hypertext Transfer Protocol Secure.*

[36] *Application Programming Interface (auf Deutsch Programmierschnittstelle). API beschreibt einen Programmabschnitt, der anderen Programmen die technische Möglichkeit gibt, sich mit dem System, welches die API bereitstellt, zu verbinden. Anschließend können anwendungsspezifische Schnittstellen-Funktionen von Programmen, die sich über die API anbinden, aufgerufen werden.*

[37] *Auf Englisch Views genannt.*

Diese Anpassungen werden im XML-Dateiformat gemacht. Diese genannten Funktionalitäten werden in Kapitel 4 verwendet und beschrieben.

Das SAP HANA Studio ermöglicht über die integrierte SQL-Konsole Data Query Language-Anweisungen (DQL-Anweisungen) auszuführen und die Datenbank zu verändern[38]. Die SQL-Anweisungen können innerhalb der Perspektive „SAP HANA Development“ genutzt werden. Auf der HANA werden anwendungsfallspezifische Funktionen in Perspektiven gruppiert. So gibt es im SAP-Umfeld beispielsweise die Perspektiven ABAP, SAP HANA Modeler, SAP HANA Administration Console und Business Warehouse Modeling. Es folgt eine Abbildung der Eclipse Workbench mit geöffneter SAP HANA Development-Perspektive:

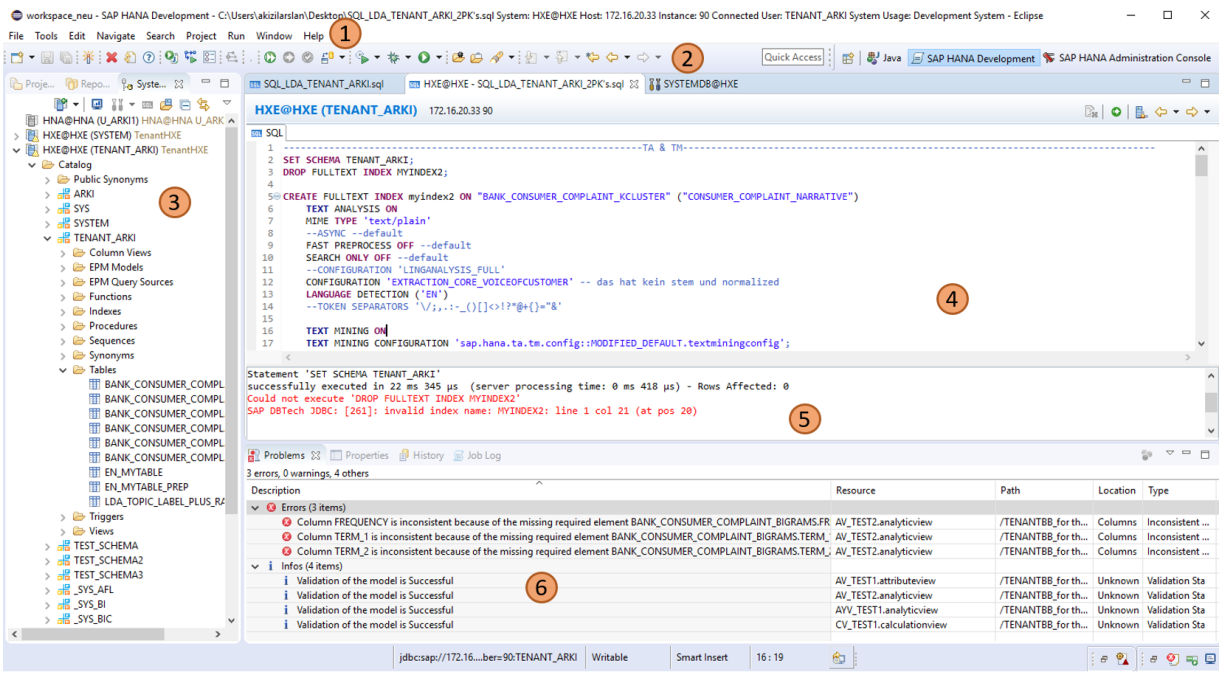


Abbildung 11: Eclipse Workbench (SAP HANA Development-Perspektive) (Quelle: Eigene Darstellung)

Die Eclipse-Oberfläche für die SAP HANA Development-Perspektive beinhaltet folgende Graphical User Interface-Bausteine (GUI-Bausteine):

1. Eine Menüleiste zum Ausführen von Standardfunktionen, wie die Öffnung von Dateien, Programmeinstellungen, Fensteranpassungen, Ausführung von Skripten, und dergleichen.
2. Eine Symbolleiste, die zum Beispiel Funktionen für die Fehleranalyse und Skriptausführung bereithält.
3. Ansichten zur Navigation über Datenbankobjekte.
4. Die SQL-Konsole zum Ausführen von SQL-Anweisungen.
5. Die Ausgabekonsole der ausgeführten SQL-Anweisungen.
6. Ansichten zum Einsehen in Fehlermeldungen, Ausführungsprotokolle und der Eigenschaften der aktuellen SQL-Sitzung.

[38] *Die Veränderung der Datenbankinfrastruktur wird durch die SQL-Kommandoarten, Data Definition Language (DDL) und Data Manipulation Language (DML), erbracht.*

Das in Eclipse integrierte SAP HANA Studio ist ein wichtiges Clientseitiges Werkzeug, um die HANA zu bedienen und auf der HANA zu entwickeln. SAP ist in enger Zusammenarbeit mit Eclipse daran interessiert, SAP HANA Studio auf der Eclipse-Plattform auszubauen. In diesem Projekt wird deshalb Eclipse in die Entwicklungsumgebung integriert. SAP HANA Studio ist die zentrale Technologie zur Interaktion mit der HANA und Realisierung der Teilprozesse im Vorgehensplan des Text-Minings (siehe Kapitel 2.3.2).

2.5.2 Spyder

Spyder ist eine plattformübergreifende Entwicklungsumgebung für die Programmiersprache Python. Python ist eine universelle Programmiersprache[39], welche auf viele verschiedene Problemklassen angewendet werden kann [Vgl. Python Software Foundation, 2018]. Spyder integriert eine Reihe von bekannten Paketen im wissenschaftlichen Python-Stack. Außerdem kann Spyder benutzerfreundlich bedient werden und erlaubt die deklarierten Variablen über die Benutzeroberfläche zu erforschen und bearbeiten.

In diesem Projekt soll auf eine Bibliothek von Python zurückgegriffen werden, um die Textdaten aus Facebook per GET-Anfrage zu extrahieren. Danach sollen die Daten von Spyder über das SQL Command Network Protocol in ein vorgesehenes Schema der HANA-Datenbank geladen werden (siehe Kapitel 4.1).

2.5.3 SAP Lumira Discovery

SAP Lumira Discovery ist eine Frontend-Lösung, welche zur Erstellung von Dashboards und Analysen entwickelt wurde. Echtzeitanalysen, in Kombination mit der HANA und interaktiven Self-Service-Datenvisualisierungen, können über alle Geschäftsbereiche hinweg mit SAP Lumira Discovery verwirklicht werden. Zudem können die aus der Anwendung von Data-Mining-Prozessen gewonnenen Daten mit SAP Lumira Discovery untersucht werden. Das Programm stellt ausgereifte Visualisierungsarten bereit, um den Nutzer beim Erforschen von Mustern und Zusammenhängen zu unterstützen. Aus dem gewonnen Wissen werden im Rahmen des Projektes Handlungsmaßnahmen getroffen (siehe Kapitel 5.2).

Das primäre Ziel der Arbeit ist, zu untersuchen, wie gut ein Text-Mining-Projekt mit SAP-Lösungen bewältigt werden kann. Aufgrund des Reifegrades der Software und dem Beitrag zum primären Ziel wird das Frontend-Werkzeug von SAP in die Entwicklungsumgebung mitberücksichtigt.

2.5.4 Facebook Graph Application Programming Interface

Die Facebook Graph API ist eine Programmierschnittstelle, die von Facebook entwickelt wurde und die Möglichkeit bietet, einen Datenfluss von der Datenbank von Facebook zu anderen Applikationen auszulösen. Die Graph API stellt also, unter Voraussetzung eines gültigen

[39] *Auf Englisch General Purpose Language genannt.*

Zugangsschlüssels, eine anknüpfbare Schnittstelle dar, um Anfragen an die Datenbank von Facebook zu senden. Die empfangenen Daten können danach in andere Programme geladen werden. Anschließend können weitere Algorithmen bzw. Funktionen die empfangenen Daten verarbeiten und einen Mehrwert schaffen.

Es wird vorausgesetzt, dass ein Zugangsschlüssel mit bestimmten Berechtigungen benötigt wird. Außerdem sind nur diejenigen Daten extrahierbar, die öffentlich zugänglich sind. Es können von denjenigen Facebook-Konten Informationen extrahiert werden, die Drittanbietern[40] die Erlaubnis für das Extrahieren der eigenen Daten erteilt haben.

Im Folgenden werden einige aus Facebook extrahierbare Informationen aufgelistet:

- Beiträge in Form von Fotos, Videos, Standorte, Veranstaltungen und Umfragen
- Kommentare zu Beiträgen
- Anzahl der Gefällt Mir-Angaben oder Reaktionen[41] zu Beiträgen und Kommentaren
- Namen der Freunde eines Facebook-Mitglieds
- Geburtstag eines Facebook-Mitglieds
- Gefällt Mir-Angabe zu Bücher, Filme und Persönlichkeiten
- Abonnements von diversen Gruppen oder Facebook-Konten

[40] *Unter Drittanbieter werden Nutzer der Graph API verstanden.*

[41] *Im Kontext der Nutzung von Facebook sind Reaktionen Emoticons, die anstelle von Gefällt Mir-Angaben, für Beiträge und Kommentare vergeben werden können.*

3 Analyse & Planung des Text-Mining-Projekts

In diesem Kapitel wird das Vorgehen der Analyse- und Planungsphase des Projektes im Rahmen der zu analysierenden Datenquelle chronologisch behandelt.

Nach dem KDD wird das Projektziel definiert und die zeitgemäßen Erwartungen an Kompetenzen der HANA gestellt (siehe Kapitel 3.1). Daraufhin findet eine Recherche über den Auswahlprozess der Datengrundlage eines reellen Handelsunternehmens statt (siehe Kapitel 3.2). Um die Anonymität des Unternehmens zu gewährleisten, bezeichnen wir das Unternehmen im weiteren Verlauf der Arbeit als Tradefood GmbH, welcher in der Handelsbranche in Deutschland tätig ist. Danach wird die Extraktion der zu analysierenden Daten aus der Quelle in die HANA im Kontext eines ETL-Prozesses angeführt (siehe Kapitel 3.4). Die HANA stellt in der Entwicklungsumgebung die zentrale Technologie, auf dem die Kernprozesse für das Text-Mining ausgeführt werden sollen, dar. Es werden die Möglichkeiten der Vorverarbeitungsmethoden der HANA für unstrukturierte Daten eruiert und die Methodenauswahl getroffen (siehe Kapitel 3.5). Daraufhin wird die Theorie eines ausgewählten Algorithmus erklärt und der daraus zu ziehende Nutzen erläutert (siehe Kapitel 3.6). Im Anschluss wird die HANA-Funktion zur Extraktion von Entitäten und Sentiments vorgestellt, die verwendet werden, um Kunden-Kommentare der Tradefood GmbH zu analysieren (siehe Kapitel 3.7). Schließlich wird auf technischer Ebene die Analyse und Planung der zu nutzenden Visualisierungslösung beschrieben (siehe Kapitel 3.8).

Die konkrete technische Umsetzung des geplanten Vorgehens, insbesondere auf Basis der HANA, wird in dem nachfolgenden Hauptkapitel erarbeitet (siehe Kapitel 4).

3.1 Zieldefinition

Es werden in diesem Projekt zwei Ziele verfolgt, die in Zusammenarbeit mit der Infomotion GmbH, definiert wurden. Das primäre Ziel ist technisch orientiert und behandelt die Evaluierung von Text-Mining-Kapazitäten der HANA, im Hinblick auf effektive und effiziente Einsetzbarkeit im Unternehmensumfeld. Dabei wird versucht, möglichst viele Prozesse mit den SAP-Standards abzudecken, ohne andere Lösungen zur Hilfe zu nehmen.

Das sekundäre Ziel ist fachlich orientiert und beschränkt sich auf die Datenanalyse von Social-Media-Beiträgen und –Kommentaren von Tradefood GmbH. Dabei soll ein Clustering Algorithmus verwendet werden, um verborgenen Themen in den Facebook-Beiträgen zu finden und den Beiträgen zuzuordnen. Anschließend sollen die gefundenen themenspezifischen Beiträge mit den Kunden-Kommentaren verknüpft werden, damit eine differenzierte Analyse nach Themen erfolgen kann. In den Kunden-Kommentaren werden anschließend nach Fakten und Sentiments gesucht.
Das sekundäre Ziel gibt somit einen Rahmen für das primäre Ziel vor, welches die Auswahl an Methoden der technischen Umsetzungsmöglichkeiten eingrenzt. Das bedeutet, dass maßgeblich

nur die zur Erreichung des sekundären Ziels benötigten Textanalyse-Methoden der HANA evaluiert werden.

Die Social-Media-Beiträge stellen eine Primärquelle dar, auf der die Anwendung von Entity und Fact Extraction angestrebt wird. Unternehmen, wie Tradefood GmbH, die im Einzelhandel tätig sind, haben Interesse daran, den Kunden genauer zuzuhören. Das Zuhören soll Aufschluss über die Wünsche, Bedürfnisse und Kritik der Kunden geben und die Stimmungslage ableitbar machen. Die gewonnenen Informationen können strategisch eingesetzt werden, um die Kundenzufriedenheit zu erhöhen, sowie eine Gewinnmaximierung durch höhere Absätze anzustreben. Unternehmen, zum Beispiel aus der Handelsbranche, werden fähig, Marktänderungen schnell zu identifizieren und Handlungsmaßnahmen einzuleiten.

Die Disziplin des Text-Minings ist nicht neu. Erste Entwicklungen wurden schon im letzten Jahrzehnt des 20. Jahrhunderts gemacht. Bis dato wurden in diesem Bereich viele Forschungen betrieben und es wurden anspruchsvolle Text-Mining-Projekte in der Vergangenheit bewältigt. Wird der gegenwärtige Stand der Technik und die Kompetenz des Softwareherstellers SAP betrachtet, sollte das Sprachmodul der HANA gute Ergebnisse liefern und einen hohen Reifegrad erreicht haben, um die Anforderungen von Unternehmen umzusetzen bzw. zu unterstützen.

Da die HANA eine hohe Verarbeitungsgeschwindigkeit aufweist, könnte sie in Kombination mit Text-Mining strategisch eingesetzt werden. Der strategische Ansatz, der dabei verfolgt wird, ist das Erübrigen der Einbeziehung weiterer Lösungen, die die IT-Landschaft von Unternehmen weniger komplizieren. Aufgrund der Ausführung der Text-Mining-Prozesse auf der IMDB, könnte die Verarbeitungsgeschwindigkeit erhöht und somit auch für spezielle Einsatzgebiete, wie die des Text-Minings, die Anforderung der nahezu Echtzeitverarbeitung gewährleisten. Weitere Aufwandsfaktoren, wie Lizenzgebühren, Zeitaufwände und eventuelle Fehlerbehebungen, aufgrund der notwendigen Nutzung von ETL-Prozessen (siehe Kapitel 3.4) zur Datenübertragung, würden durch die Verwendung der Text-Mining-Kapazitäten auf dem HANA-System entfallen.

3.2 Recherche zur Auswahl der Datengrundlage

Es wird die automatisierte Analyse von öffentlichen zugänglichen Textdaten von Tradefood GmbH angestrebt, der in der Handelsbranche tätig ist und Lebensmittel an private Haushalte vertreibt. Es wurde eine Recherche auf den digitalen Social-Media-Konten dieses Unternehmens durchgeführt. Das Resultat der Recherche ist, dass viele Kunden den Social-Media-Kanal Facebook nutzen, um mit Tradefood GmbH zu interagieren, Meinungen und auch Wünsche zu äußern. Außerdem präferiert das Unternehmen auch Facebook als Kommunikationskanal. Es besteht ein gegenseitiger Austausch zwischen beiden Parteien.

Facebook verfügt über eine große Anzahl von aktiven Nutzern (siehe Abbildung 12). Facebook-Mitglieder sind über Gruppen, Foren, und Freundschaften miteinander vernetzt. Sie stehen regelmäßig im Informationsaustausch und teilen Beiträge, wie zum Beispiel Fotos, Videos, Standorte, schreiben Kommentare und teilen ihre Ereignisse. Des Weiteren hinterlassen Nutzer „Gefällt mir“-Angaben und teilen Reaktionen über Beiträge anderer privater Personen und Firmen.

Aufgrund der großen Community sind Firmen auch daran interessiert, in dem dynamischen Netzwerk mit Kunden in Kontakt zu treten. Firmen wollen gezielt über Facebook Werbung machen und wollen verstehen, was die Kunden über die Firmen, Konkurrenten und das Angebot an Produkten denken. Darüber hinaus möchten Firmen unter anderem ihren Erfolg messen, wenn sie besondere Ereignisse, Angebote und Neuprodukte ankündigen. Überdies ist eine weitere Anstrengung der Firmen positive und negative Beiträge zu identifizieren und dessen Ursachen zu verstehen. Allgemeine Ziele der Unternehmen sind die Entwicklung von Maßnahme, um Risiken zu meiden, Chancen wahrzunehmen, indem sie den von den Käufern geprägten Markt verstehen.

Im Folgenden wird die Anzahl der wöchentlich aktiven Nutzer von populären Social-Media-Plattformen in Deutschland angezeigt:

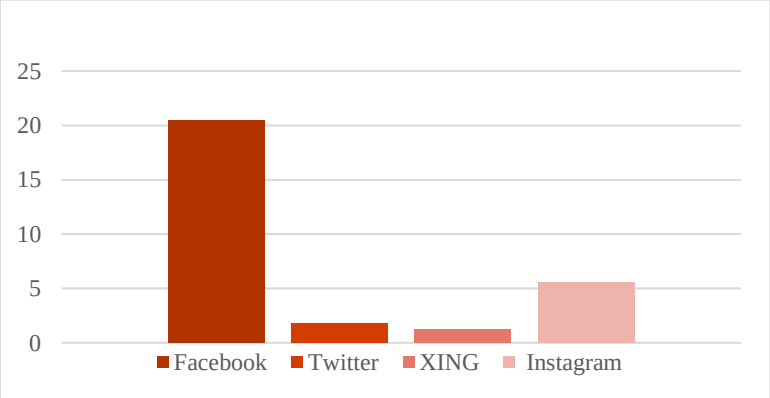


Abbildung 12: Gegenüberstellung der wöchentlich aktiven Nutzer von populären Social-Media-Plattformen in Deutschland (Quelle: Eigene Darstellung)

Aus der Grafik geht eine wöchentlich aktive Nutzeranzahl von über 20 Millionen Facebook-Nutzern in Deutschland hervor. Die Social-Media-Plattformen Twitter, Xing und Instagram haben hingegen deutlich weniger Nutzer in Deutschland, wobei Instagram mit ca. sechs Millionen aktiven Nutzern auch beachtenswert ist. Xing ist eine Plattform zum Vernetzen mit Personen aus dem Arbeitsumfeld und Verfolgen von Nachrichten. Twitter nutzen in der Regel populäre oder wichtige Persönlichkeiten und Unternehmen. Private Nutzer sind auf Twitter selten aktiv und folgen eher den vorhin genannten Nutzergruppen.

In der vorliegenden Arbeit, wird geplant Textdaten aus der Facebook-Seite von Tradefood GmbH zu verwenden. Die Textdaten sollen hauptsächlich Beiträge des Unternehmens und die zugehörigen Kommentare von Kunden beinhalten. Zu den Beiträgen und Kommentaren soll zusätzlich das Datum berücksichtigt werden.

3.3 Datenexploration

Im Rahmen des zugrundeliegenden Projektes kann die Exploration der Daten schon in der Planungsphase erfolgen, um ein besseres Verständnis über die Domäne und die Facebook-Seite von Tradefood GmbH zu erhalten. Daher wurde die Facebook-Seite von Tradefood GmbH durchforstet.

Es folgen Auszüge der geteilten Beiträge von dem Unternehmen, sowie die zugehörigen Kommentare der Kunden. Ziel ist, die Ermittlung einiger Charakteristiken und Inhalte der Daten. Der Name des Unternehmens ist durch die Zeichenkette „tradefood“ ersetzt worden:

1. Sommer-frischer Eisgenuss – unser neues Waldmeister-Vanille-Eis ist da! Herrliches Apfelfruchteis mit Waldmeister-Geschmack und einem Kern aus cremigem Vanilleeis. Das müssen Sie probieren! http://www.tradefood.de/Produkte/Eis-Eis-Eis/Waldmeister-Vanille/pu9244
    a. Hab erst jetzt gesehen das es noch andere Sorten gibt
    b. Voll Lecker
    c. Maren Konrad😍
2. Jacke an, Mütze auf – und rein ins Winter-Grill-Vergnügen! Zum Beispiel mit unseren saftigen XXL Steakhouse Burgern. Denn wer sagt, dass Grillen nur was für den Sommer ist? Einfach mal ausprobieren! Und dann: verraten, wie’s war! http://www. tradefood.de/Produkte/Hackfleisch-zubereitungen/XXLSteakhouseBurger/pu7896413
    a. Was ein Blödsinn. Muss man alles machen . Nur weils IN is.
    b. Neu? Habt ihr die letzten 5 bis 10 Jahre verschlafen? ☺😂
    c. ist einfach genial
3. 8.792.843 Euro – wir freuen uns mit RTL - Wir helfen Kindern - Spendenmarathon über das tolle Ergebnis des diesjährigen Spendenmarathons! Und natürlich auch über unseren Anteil daran. Vielen Dank an alle Helfer, Paten, Unterstützer und das gesamte RTL-Team! #dankeschön #jedercentkommtan #rtlspendenmarathon #spendenmarathon # tradefood _deutschland #miteinanderaugenauf
    a. Danke, dass ihr dabei gewesen seid und uns so sehr in unserer Arbeit für die Kinder unterstützt. Nur so ist es möglich
    b. Es gibt genug Armut auf der Welt, auch in Deutschland
4. Egal ob tradefood*Wagen, tradefood*Boot oder tradefood*Raumschiff: Machen Sie ein Foto von tradefood* on Tour und laden Sie es in die Kommentare. Schon landen Sie im Lostopf um einen tradefood*Gutschein im Wert von 500 Euro! Das Gewinnspiel endet am 15.03. Hier gibt es alle Teilnahmebedingungen: www.tradefood.de/teilnahmebedingungen
    a. Frische und Genuss für den kleinen und großen Hunger! #tradefoodschnappschuss
    b. tradefood Deutschland ist einfach super! Vielen Dank für das tolle Gewinnspiel! Eure tollen Flitzer sehen wir fast jeden Tag und falls wir gewinnen sollten, würden wir eine Großbestellung für ein Schulfest bei uns aufgeben :::-)
    c. Frei Haus Lieferung auch in den Außenbezirk.... nun ist die Tiefkühtruhe wieder voll. #tradefoodschnappschuss
    d. Bei Wind und Wetter ist unser Fahrer unterwegs und versorgt uns immer mit einem Lächeln und natürlich mit der tradefoodware....🍱🍱🍗🍲🍝🍛🍩🍦🍡🍡🍰🍤🍮🍴

Im Folgenden werden einige identifizierbare Merkmale der Beiträge und Kommentare aus der Facebook-Seite von Tradefood GmbH aufgelistet:

- Rhetorische Fragen
- Kontraktionen und Zusammensetzungen von Wörtern
- Emoticons, die Gefühle/Stimmungen und Objekte illustrieren
- Emoticons, bestehend aus einer Zusammensetzung von Interpunktionen
- Kundenmeinungen über Beiträge, Produkte, Aktionen und weitere Dienstleistungen
- Internetadressen und Hashtags
- Rechtschreib- und Tippfehler
- Sätze/Wörter in Umgangssprache verfasst
- Kurze Texte bzw. Kommentare
- Erwähnung/Markierung von Facebook-Mitgliedern in Kommentaren
- Nennung von Nahrungsmitteln oder geworbenen Produkten

Die oben genannten sprachlichen Charakteristiken ergeben sich aus dem Bereich der sozialen Medien. Sie sind typische Herausforderungen im Text-Mining, die das Sprachmodul der HANA mit geeigneten Vorverarbeitungsmethoden und Analysen bewältigen soll.

3.4 Datenintegrationsprozess der Social-Media-Beiträge

Die Integration von heterogenen externen Daten, die in verschiedenen Systemen liegen, werden durch Adapter der SAP HANA Smart Data Integration bereitgestellt. Adapter zur Datenbereitstellung können eine Verbindung mit einer Vielzahl von Datenquellen herstellen und Daten in die HANA transportieren. Die Entwicklung der SAP HANA Smart Data Integration sollte demnach die Möglichkeit geben, eine effektive und kontinuierliche Datenverbindung zu gewährleisten, damit der Datenstrom nicht unterbrochen wird. Um das Bedürfnis der Echtzeitanalyse auf der HANA zu befriedigen, stellt deshalb SAP HANA Smart Data Integration eine wichtige Kompetenz dar.

Im Rahmen eines Projektes zur Weiterentwicklung des Wissensmanagements der Infomotion GmbH wurde die Graph API in Zusammenhang mit der HANA-Datenbank verwendet. Dabei wurde der Camel Facebook Adapter von der Apache Softwarestiftung genutzt und auf Effizienz und Effektivität geprüft. Die Nutzung des Adapters erbrachte keinen Mehrwert und wurde nicht für den Einsatz bei Kunden der Infomotion GmbH empfohlen.

Aufgrund dessen wird Python zur Extraktion der Daten von Facebook in die HANA als Alternative in Erwägung gezogen. Der Datenintegrationsprozess kann auch als ein ETL-Prozess angesehen werden.

Zunächst folgt eine allgemeine Begriffsdefinition von ETL-Prozessen. Der ETL-Prozess beschreibt ein Vorgehen zur Beladung eines Data Warehouses[42], welcher in der Regel für Analysezwecke optimiert ist. Das Vorgehen der Beladung wird in drei Teilprozesse gegliedert [Vgl. Kimball und Ross, 2013, S. 19 f.]:

- Extraktion: Extrahieren von Daten aus unterschiedlich strukturierten bzw. heterogenen Datenquellen.
- Transformation: Umwandlung von Daten, indem die Teilprozesse Filterung, Harmonisierung, Aggregation, sowie Anreicherung angewendet werden [Vgl. Kemper et al., 2010, S. 27 f.].
- Laden: Beladung der transformierten Daten in die Zieldatenbank.

Python fungiert in der Entwicklungsumgebung Spyder im Kontext von ETL-Prozessen als ein ETL-Werkzeug. Im Rahmen der Planung des vorliegenden Projektes extrahiert Spyder über das Kommunikationsprotokoll HTTPS die Social-Media-Beiträge unter Verwendung der Graph API und behält sie zunächst in seiner Umgebung vor. Nachfolgend werden die Daten ohne einen Transformationsprozess in die Zieldatenbank HANA geladen.

Der Beladungszyklus der HANA mit Social-Media-Daten sollte nach Bedarf angestoßen werden. Von Maschinen erzeugte Sensordaten oder Transaktionsdaten werden in der Praxis kontinuierlich in festen Zeitabständen extrahiert. Dahingegen ist ein fester Beladungszyklus für dieses Projektes nicht notwendig. Der Bedarf für Tradefood GmbH zu einer erneuten Beladung ergibt sich zum Beispiel durch die Organisation einer besonderen Aktion oder eines Sonderangebots. Deswegen sollte ein weiterer Beladungsprozess erst angestoßen werden, wenn besondere Ereignisse eintreten, die viel Resonanz in Form von Beiträgen und Kommentaren von Kunden, verursachen. Für die Durchführung dieses Projekts sollen Daten einmalig über die Graph API extrahiert und in die HANA zur Ausführung der geplanten Textanalyse geladen werden.

3.5 Methoden der Textdatenaufbereitung & linguistischen Textanalyse mittels der HANA

Das vorliegende Kapitel wird einige Funktionen der HANA vorstellen, die zur Textdatenaufbereitung und linguistischen Textanalyse der Beiträge von Tradefood GmbH verwendet werden sollen. Die Kunden-Kommentare werden in diesem Kapitel noch nicht berücksichtigt. Die Aufbereitung und linguistische Analyse stellen die Grundlage für die nachfolgende Analyse mit dem LDA-Algorithmus dar, welcher zur Findung von verborgenen Themen in den Beiträgen angewendet werden soll.

Textdatenaufbereitungs- und linguistische Textanalyseoptionen werden in einer XML-Konfigurationsdatei zusammengefasst (siehe Anhang 4) und beim Erzeugen des Full-Text-

[42] *Auf Deutsch Datenlager.*

Index als Parameter mitgegeben (siehe Kapitel 3.5.1). Es wird die Konfigurationsdatei LINGANALYSIS_STEMS verwendet, um die Extraktion (siehe Kapitel 3.5.2) und die nachfolgenden Normalisierung von Tokens(siehe Kapitel 3.5.3), sowie das Stemming der Tokens (siehe Kapitel 3.5.3) durchzuführen. Die Funktion zur Identifizierung von Wortklassen[43] wird in diesem Projekt nicht ausgeführt, da keine Analyse von Grammatikstrukturen beabsichtigt ist.

Der Prozess zur Bereinigung der Stopwords, der zur Datenaufbereitung angehört, wird in einer separaten Konfigurationsdatei abgelegt. Zu Testzwecken wird erstmal die Standardkonfiguration, DEFAULT, verwendet, bevor weitere Stopwords eingepflegt werden. Die Entfernung von Stopwords wird nicht näher ausgeführt, da es im HANA-Umfeld keinen komplexen Algorithmus darstellt. Die Bereinigung von Stopwords wird durch einen einfachen Abgleich der Tokens mit einer Liste von Stopwords verwirklicht.

Konfigurationsdateien sind im SAP HANA-Umfeld im XML-Format gespeichert und können nach den projektspezifischen Anforderungen angepasst werden. Es können auch von SAP vorkonfigurierte XML-Dateien genutzt werden[44]. Bestandteile der anzuwendenden Konfigurationsdateien lösen Prozesse aus, welche in den folgenden Unterkapiteln erläutert werden.[45]

3.5.1 Indizierung der Textspalte

Wir nehmen an, dass die Textdaten in die HANA geladen wurden und in einer physischen Tabelle in spaltenorientierter Speicherform vorliegen. Die zu analysierende Textspalte der Tabelle mit den Beiträgen der Tradefood GmbH wird manuell mit dem sogenannten Full-Text Index durch eine SQL-Anweisung indiziert.

Es wird der Zusammenhang zwischen einer Quelltabelle und einem Full-Text Index dargestellt:

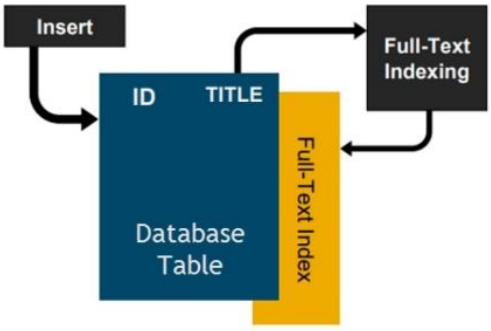


Abbildung 13: Der Zusammenhang zwischen einer Datentabelle und dem Full-Text Index [Rajpal, 2018]

[43] *Auf Englisch Part-of-Speech Tagging.*

[44] *Die vorkonfigurierten XML-Dateien für Textanalysen heißen LINGANALYSIS_BASIC, LINGANALYSIS_STEMS, LINGANALYSIS_FULL, EXTRACTION_CORE und EXTRACTION_CORE_VOICEOFCUSTOMER. Sie werden in der SQL-Anweisung mit dem Dateinamen angesprochen und als Parameter bei der Anlegung des Full-Text-Index mitgegeben.*

[45] *Es werden hauptsächlich Quellen von SAP berücksichtigt, da diese in der Regel verlässlicher sind und die aktuelle Version 2.0 SPS03 der SAP HANA Plattform abbilden.*

Der Full-Text Index generiert implizit eine interne verborgene Spalte, die den gleichen Typ der indizierten Textspalte erhält. Der Full-Text Index enthält die gleichen Daten wie in der Stammtabelle, die jedoch, wie im Folgenden abgebildet, optimiert verarbeitet werden:

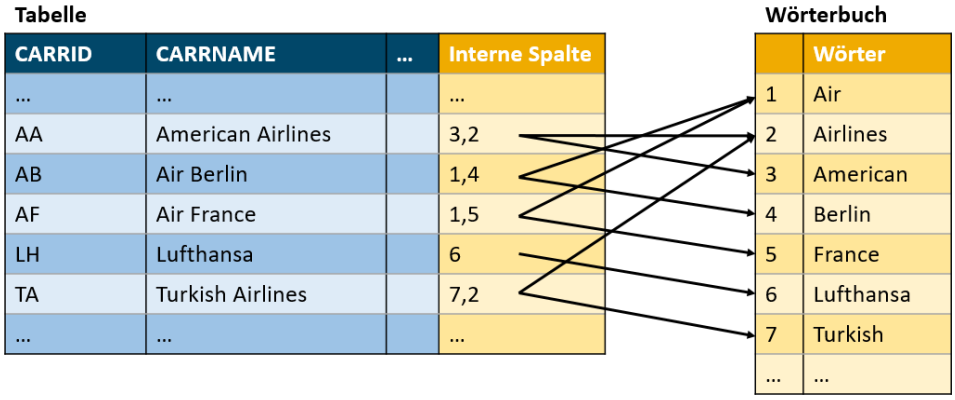


Abbildung 14: Die Full-Text Index Erzeugung (Quelle: Mit geringfügigen Veränderungen entnommen aus [Bicu, o.J.])

Der Full-Text Index erzeugt eine Wörterbuch-Tabelle, welche die Tokens der indizierten Textspalte abbildet. Die interne Spalte der Quelltabelle beinhaltet eine Sequenz von ganzen Zahlen, die auf die Wörterbuch-Tabelle referenzieren.

Die Vorverarbeitung und linguistische Textanalyse wird ausgelöst, wenn der Full-Text Index auf der Textspalte mit existierenden Daten erzeugt wird und die zugehörigen Parameter in der SQL-Anweisung mitgegeben werden. Werden neue Daten in die Textspalte geschrieben oder Aktualisierungen vorgenommen, wird die Vorverarbeitung und Textanalyse erneut ausgelöst. Der Aufruf findet in der SQL-Konsole des SAP HANA Studios statt (siehe Abbildung 11).

3.5.2 Extraktion

Die Beiträge der Tradefood GmbH müssen zunächst in Tokens zerlegt werden. Die HANA verwendet allgemeingültige Regeln zur Extraktion von Tokens aus Texten, die in Sprachen geschrieben sind, die Leerzeichen zur Worttrennung nutzen. Außerdem sind spezifische Regeln für viele verschiedene Sprachen implementiert. Im Folgenden werden einige von der Sprache unabhängige Regeln bzw. Muster vorgestellt, die bei Bedarf verändert werden können [Vgl. SAP SE, 2018g, S. 13 f.]:

- Idiomatische Phrasen, wie zum Beispiel *Schwein gehabt*, *Keine Ursache* und *out-of-the-box* werden identifiziert und als eine ganze Einheit bzw. ein Token erkannt.
- Wörter, die mit Bindestrichen zusammengesetzt sind, werden nicht in einzelne Tokens extrahiert, da sie sonst ihre Bedeutung verlieren.
- Folgt einem Wort eine Interpunktion und anschließend kein Leerzeichen, sondern eine weitere Zeichenfolge, so wird diese Phrase nicht extrahiert. Als Beispiel können die Wörter *U.S.A.* oder Dateinamen wie *text_mining.sql* genannt werden.
- Abkürzungen, wie *usw.*, *etc.* und *co.*, werden nicht extrahiert, trotz des folgenden Leerzeichens nach der Interpunktion.

– Apostrophen werden je nach Sprache unterschiedlich extrahiert.

Zusätzlich werden folgende spezifische Regeln der Sprache Deutsch im Extraktionsverfahren hinzugezogen [Vgl. SAP SE, 2018g, S. 21 f.]:

– Wörter, die ein Apostroph beinhalten, wie beispielsweise *wie's*, werden in zwei Tokens gespalten, so dass *wie* und *'s* entstehen. Wörter, wohingegen das Apostroph zum Namen gehört, werden nicht gespalten.
– Der Bindestrich von zwei Wörtern, die mit einer Konjunktion verbunden sind, wird bei der Extraktion beibehalten, es sei denn, nach dem Bindestrich folgt ein groß geschriebenes Wort.
– Abkürzungen mit nachfolgender Interpunktion, wie *bzgl.* oder *Mrd.*, werden nicht extrahiert. Das gleiche gilt auch für Ordinalzahlen.

Die Regeln der Extraktion können je nach Anforderungen der Domäne ergänzt werden. Außerdem können die Regeln entfernt werden. Beispielsweise kann der Extraktionsvorgang so gestaltet werden, dass die Apostrophe oder Bindestriche nicht als ein Zeichen zur Worttrennung angesehen werden sollen.

3.5.3 Case Normalization & Stemming

Die HANA wendet die Normalisierung von Tokens mit verschiedenen Groß- und Kleinschreibungen an. Dieser Prozess wird Case Normalization genannt. Einzigartige Namen, wie SAP, IBM, Oracle und MongoDB, werden laut SAP von dem Normalisierungsprozess ausgeschlossen. Alle Titel, wie zum Beispiel von Zeitungsartikeln, werden in die Kleinschreibung überführt. Außerdem wird das erste Wort in einem Satzbeginn in die Kleinschreibweise überführt, wohingegen die Groß- und Kleinschreibweise der Wörter die nach dem ersten Wort folgen, nicht normalisiert werden. [Vgl. SAP SE, 2018g, S. 39 f.] Spelling Normalization (siehe Kapitel 2.3.2.1) wird auf der HANA gegenwärtig nicht unterstützt.

Die Wortstammbildung bzw. das Stemming in der HANA wird mit Hilfe eines digitalen Wörterbuchs ausgeführt. Das Wörterbuch kann zusätzlich ergänzt werden, indem alle möglichen Varianten der Grundform des Wortes deklariert werden. Anhang XXX zeigt die Syntax der Grundform von Worten und ihrer möglichen Varianten. Da im Vorhinein die Normalisierung durchgeführt wird, sind alle Worte im Wörterbuch in der Kleinschreibweise aufgeführt. Wörter, dessen Wortstamm unbekannt sind, werden von dem sogenannten Stemmer Guesser verarbeitet. Der Stemmer Guesser besitzt morphologische Regeln, die den Wortstamm des unbekannten Wortes vorhersagen können [Vgl. SAP SE, 2018g, S. 78].

Zusammengezogene Präpositionen werden in ihre Bestandteile überführt. Beispielsweise wird das Wort *ins* zu *in=das* umgewandelt. Es resultiert ein Token, wobei das Gleichheitszeichen die gleichrangige Wichtigkeit beider Begriffe symbolisiert. Optionale Bindestriche in Wörtern,

wie *Kaffee-Ersatz*, werden zu *Kaffeeersatz* umgeformt. Abkürzungen, Akronyme, Zahlen und Konjunktionen werden nicht beeinflusst beim Stemming. [Vgl. SAP SE, 2018g, S. 54 f.]

In der deutschen Sprache nutzen wir die diakritischen Zeichen, um die Umlaut-Punkte auf den drei Buchstaben *a, o* und *u* abzubilden. Enthalten Tokens die Umlaute *ae, ue* und *oe,* wandelt der Stemmer der deutschen Sprache die Umlaute in *ä, ü* und *ö* um.[46] So wird auch die unterschiedliche Schreibweise auf eine einheitliche Darstellung überführt.

Die HANA ist fähig, zusammengesetzte Wörter zu identifizieren und sie in ihre Komponenten zu zerlegen. Die Komponenten werden anschließend auf den Wortstamm zurückgeführt und präsentiert. Zusammengesetzte Wörter ergeben sich in der deutschen Sprache durch die Kombinierung von verschiedenen Wortarten. Demnach werden Substantive mit Substantiven, Substantive mit Adjektiven, Verben mit Substantiven, und so weiter miteinander kombiniert.

Im Folgenden sind mögliche Kombinationsarten in der deutschen Sprache und die zugehörige Repräsentation des Wortstamms abgebildet [Vgl. SAP SE, 2018g, S. 80 ff.]:

- Zusammengesetzte Wörter aus der Wortart Nomen haben Fugenelemente, wie *en* oder *s,* die beide Wörter zusammenfügen. Somit wird zum Beispiel das Wort *Lieblingsgericht* zu L*iebling#Gericht* oder *Bohnenkeimlinge* zu *Bohne#Keimling* umgewandelt, wobei das Fugenelement entfernt wird. Es gibt auch ohne Fugenelemente bestehende Wörter aus Nomen, wie das Wort *Apfelfruchteis*, welches nach dem Stemming in die Zeichenfolge *Apfel#Frucht#Eis* geformt wird.
- Eine Zusammensetzung von zwei Wörtern aus Adjektiven; oder Substantive und Adjektiven sind in der deutschen Sprache auch üblich. Demnach werden Wörter, wie *blaugrün* oder *tiefgefroren,* in die Kompositionsglieder *blau#grün* und *tief#frieren* umgewandelt.
- Zusammensetzungen bei dem das zweite Kompositionsglied einen Substantiv und das erste Kompositionsglied Adjektive, Adjektivpartizipien, Adverbien, Verbstämme oder Eigennamen darstellen, werden von der HANA auch erkannt. Beispiele für Kombinationen dieser Art sind Wörter, wie *Waschmaschine*, *Goethehaus* und *Tiefkühltruhe*. Diese werden in die Zeichenfolgen *waschen#Maschine*, *Goethe#Haus* und *tief#kühl#Truhe* überführt.
- Nachdem das Wort *wie's* bei der Extraktion in *wie* und *'s* umgewandelt wurde, wird es beim Stemming in die Tokens *wie* und *es* verarbeitet.

3.6 Latent Dirichlet Allocation Clustering-Algorithmus

Der Clustering-Algorithmus Latent Dirichlet Allocation soll verwendet werden, um verborgene Themen in den Facebook-Beiträgen der Tradefood GmbH ausfindig zu machen. Nachdem die

[46] *Der Grund für die unterschiedliche Schreibweise der Umlaute liegt zum Teil daran, dass diverse Programme ältere ISO-Kodierungen verwenden.*

Beiträge nach Themen sortiert wurden, wird angestrebt, diese mit den Kunden-Kommentaren zu verknüpfen. Die Umsetzung des LDA-Algorithmus in der SAP HANA Studio-Umgebung kann in Kapitel 4.3 betrachtet werden. In diesem Kapitel wird der LDA-Algorithmus eingeführt und die Funktionsweise, sowie die Anwendungsbereiche näher beschrieben.

Der Latent Dirichlet Allocation-Algorithmus (LDA-Algorithmus) wendet Verfahren der Statistik an. LDA ist ein generatives probabilistisches Modell [Vgl. Blei et al., 2003, S. 996] und gehört unter anderem zu der Klasse der Themen-Modellierung. Das LDA-Verfahren verfolgt das Ziel, in einer großen Menge an Dokumenten latente Themen (auf Deutsch Topics) automatisch zu entdecken. Dabei wird mit Hilfe statistischer Verfahren jedes Wort in jedem Dokument analysiert. Dadurch entstehen Möglichkeiten, die das Erkennen von Zusammenhängen zwischen einzelnen Themen, sowie das Beobachten von Veränderungen der Themen und der verwendeten Wörter ermöglichen.

Es gibt den Anwendungsfall der Dokumentensuche über einen Schlüsselbegriff, welcher eine Menge an Dokumenten zurückgibt, die in Relation zu der gesuchten Zeichenfolge stehen. Als Beispiel kann die Google-Suchmaschine angeführt werden, welches die Suche über Schlüsselbegriffe anbietet. Im Vergleich dazu können Implementierungen des LDA-Algorithmus den Nutzer dazu befähigen, auf Themenebene eine Ansicht über themenspezifische Dokumenten zu schaffen und anschließend eine Dokumentenauswahl zu treffen.

Die PAL der HANA hat unter anderem den LDA-Algorithmus, um Themenmodellierungen zu ermöglichen und gehört zu der Familie der Clustering-Algorithmen. LDA wird im Kontext des Text-Minings den Clustering-Algorithmen zugeordnet, weil keine kategorisierten Daten zum Trainieren des Algorithmus vorhanden sind. Anders ausgedrückt, es besteht kein Vorwissen darüber, was die Ausgabewerte für unsere Proben sein sollten. Da der LDA-Algorithmus in nicht kategorisierten Daten Gruppen von ähnlichen Datenobjekten finden kann, wird LDA den Algorithmen des unbewachten bzw. unbeaufsichtigten Lernens zugesprochen.

Das LDA-Verfahren beruht auf der Intuition, dass Dokumente in einem Testkorpus mehrere Themen enthalten [Vgl. Blei, 2011, S. 2]. Beispielsweise kann ein ausführlicher Artikel über Data-Mining Informationen über Data-Mining-Prozesse, Visualisierungen und bestimmten Domänen enthalten, die thematisch getrennt betrachtet werden können. Es folgt eine Abbildung und eine chronologische Erläuterung über den generativen Prozess, wie sich das LDA-Modell das Entstehen eines Dokumentes vorstellt [Vgl. Blei, 2011, S. 3 f.]:

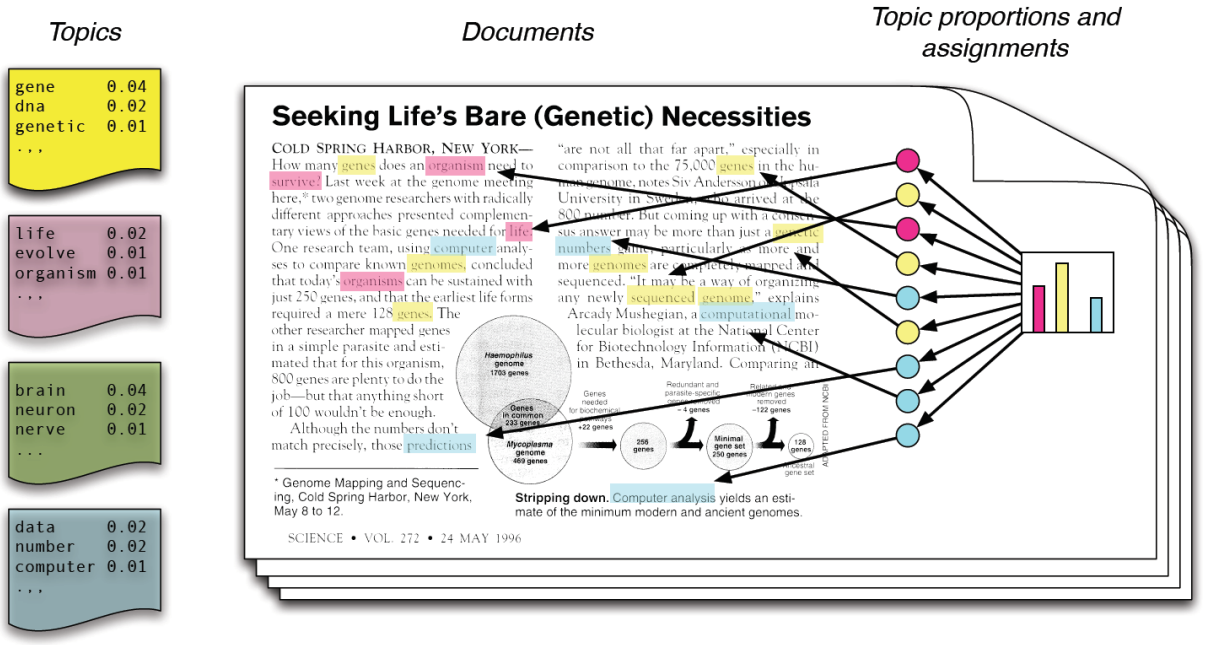


Abbildung 15: Arbeitsweise der Latent Dirichlet Allocation [Blei, 2011, S. 3]

Im LDA stellen Themen eine Verteilung über bestimmte Wörter bzw. Terme dar. Es wird zunächst angenommen, dass eine Anzahl an Themen[47] über die definierte Menge an Wörtern im Korpus existiert. Dabei ist jedes Wort, welches einem Topic zugeordnet ist, mit Wahrscheinlichkeiten bzw. Gewichten gekennzeichnet (links: Topics mit Wortverteilung). Also, das Modell nimmt an, dass zuerst die Topics und danach die Dokumente generiert werden. Die generativen Schritte, die durchgeführt werden, sehen demnach wie folgt aus [Vgl. Blei, 2011, S. 3 f.]:

1. Es wird eine feste Anzahl von Themen für den gesamten Textkorpus definiert. Allen Themen werden Wörter mit Wahrscheinlichkeiten zugeordneten (links: Anzahl der Themen).
2. Für jedes Dokument d wird die Themenverteilung anhand der Dirichlet-Verteilung[48] berechnet (rechts: grafische Darstellung der Häufigkeitsverteilung).
3. Für jedes Wort w im Dokument d:
   i. Wird zufällig ein Thema entsprechend der Themenverteilung in Schritt 2 ausgewählt (farbige Kreise mit Zeiger auf die Wörter).
   ii. Wird zufällig das Wort selbst entsprechend der Themenverteilung in Schritt 2 generiert.

[47] *Unter dem Begriff Themen bzw. Topics wird nicht der zusammenfassende Begriff bzw. die zusammenfassende Überschrift eines Themas verstanden. Vielmehr ist ein Thema eine Menge von Wörtern mit unterschiedlichen Wahrscheinlichkeiten bzw. Gewichtungen. Eine zusammenfassende Bezeichnung für das Thema muss vom Datenanalysten oder Fachexperten interpretiert werden.*

[48] *Die Dirichlet-Verteilung gibt an mit welcher Wahrscheinlichkeit eine multinomiale Verteilung (zum Beispiel die Wahrscheinlichkeit beim Würfeln die Augenzahlen 1,2,3,4,5,6 usw. zu ziehen) auftreten wird. Sie gibt also die Wahrscheinlichkeit über das Vorkommen einer Wahrscheinlichkeitsverteilung an.*

Auf diese Weise nimmt LDA die Generierung von Dokumenten wahr. Beispielhaft nehmen wir an, dass ein Dokument die Themen *Data Analysis* und E*volution* beinhaltet. Demnach könnte eine Verteilung nach Schritt 1 wie folgt aussehen:

- *Data Analysis:* predicition (0,4), data (0,25), number (0,15), computer (0,1), …
- *Evolution:* genome (0,45), genes (0,4), genetic (0,3), dna (0,2), …

Außerdem nehmen wir an, dass das zu generierende Dokument zu 70% *Data Analysis* und 30% *Evolution* thematisiert. Nach Schritt a und b werden nun die Wörter den Themen zugeordnet und die Wörter des Dokumentes der Verteilung entsprechend generiert. Das generierte Dokument könnte somit wie folgt lauten: *„data number dna prediction genome computer prediction genes computer“*. Das Dokument wird generiert, indem die Auswahl der Wörter durch die Wahrscheinlichkeiten der Wörter der Themen realisiert wird. Aufgrund dieser Eigenschaft wird bei dem LDA-Verfahren auch von einem Bag-of-Words[49]-Modell gesprochen. Die Reihenfolge der Zeichenketten bzw. Wörter im Dokument sind nicht relevant. Das generierte Dokument könnte auch eine andere Wortreihenfolge aufweisen. Außerdem ist die Option gegeben, dass durch den Bag-of-Words-Ansatz Wörter, die sinngemäß zusammengehören und unter einem Thema zugeordnet werden sollten, in unterschiedliche Themen auftauchen können [Vgl. Blei et al., 2003, S. 1008].

Es wurde erläutert, wie der generative Prozess des LDA sich das Entstehen von Dokumenten vorstellt. Das LDA-Verfahren wird in der Praxis angewendet, indem die Schritte im generativen Prozess umgekehrt angegangen werden. Dieses Mal wird davon ausgegangen, dass ein Textkorpus existiert, während für jedes Dokument die Themen und die Zuweisung der Wörter zu den Themen, sowie ihre Gewichtung gesucht bzw. unbekannt sind. Von den Dokumenten werden die Themen zurückverfolgt, um eine Menge von Themen zu finden, die wahrscheinlich den Korpus generiert haben.

An dieser Stelle kann die Erläuterung des Namens Latent Dirichlet Allocation angeführt werden: Dirichlet Allocation ist der Prozess der Zuordnung der Wörter bzw. Terme eines Dokumentes zu verschiedenen Themen anhand der Dirichlet-Verteilung. Während die Wörter der Dokumente beobachtbar sind, sind hingegen die Themen, die Themenverteilung pro Dokument und Themenzuordnung zu jedem Wort pro Dokument verborgen bzw. latent. [Vgl. Blei, 2011, S. 4]

Die Herausforderung liegt darin, diese verborgenen Strukturen im Korpus unter Zuhilfenahme der erkennbaren bzw. gegebenen Dokumente aufzudecken. Im Folgenden werden die Schritte des LDA-Verfahren erläutert:

---

[49] *Bag-of-Word beschreibt eine Multimenge von Wörtern, die in einem Dokument auftauchen [Vgl. Han und Kamber, 2010, S. 26]. Die Wörter in einem Dokument werden hierbei isoliert betrachtet. Das heißt, das Bag-of-Word-Modell berücksichtigt nicht die Grammatik und Reihenfolge. Es wird jedoch die Eigenschaft der Vielfachheit in der Multimenge herangezogen.*

1. Es wird die Anzahl an gewünschten Themen zu Beginn festgelegt. Ist die Anzahl an Themen aufgrund von Hintergrundwissen bekannt, kann dies den Aufwand der Exploration einer geeigneten Themenanzahl ersparen. Die Themenanzahl sollte nicht zu hoch (zu detaillierte Themen) und nicht zu niedrig sein (Vermengung von Themen).
2. Allen Wörtern in jedem Dokument wird zufällig ein temporäres Thema zugewiesen, da in Schritt 4 die Themenzuordnung iterativ aktualisiert wird. Die zufällige Zuweisung erzeugt für alle Dokumente eine initiale Themenverteilung und Zuweisung der Themen zu jedem Wort. Jedoch spiegelt dieser Prozess vorerst keine gute Zuordnung wieder. [Vgl. Chen, 2011]
3. Für jedes Wort w im Dokument d:
    a. Wird die bedingte Wahrscheinlichkeit P(w|t) berechnet. Die Wahrscheinlichkeit P(w|t) berechnet sich anhand des Verhältnisses von der Anzahl der Zuordnung des Wortes w zum Thema t in allen Dokumente d zur Häufigkeit des Vorkommens des Wortes w im gesamten Textkorpus. Das Ergebnis repräsentiert den Anteil der Zuordnung des Wortes w zum Thema t über alle Dokumente [Vgl. Bansal, 2016].
    b. Wird die bedingte Wahrscheinlichkeit P(t|d) berechnet. Die Wahrscheinlichkeit P(t|d) berechnet sich anhand des Verhältnisses von der Anzahl von Worten im Dokument d, die dem Thema t zugewiesen sind, zu der gesamten Anzahl von Worten in Dokument d. Das Ergebnis repräsentiert den Anteil der Wörter in Dokument d, die dem Thema t zugeordnet sind [Vgl. Bansal, 2016].
4. Durch die Multiplikation der bedingten Wahrscheinlichkeiten P(w|t) und P(t|d) wird dem Wort w die neue Wahrscheinlichkeit P(w|t) gegeben, welches eine bessere Annährung der Themenzuordnung darstellt. Dabei wird angenommen, dass alle Themenzuordnungen der Wörter bis auf die des Wortes w, welches in Schritt 3 betrachtet wird, richtig sind. Dieser Schritt stellt in Bezug zu unserem generativen Modell die Wahrscheinlichkeit dar, dass die Anteile der Themen in den Dokumenten die Wörter generieren. [Vgl. Bansal, 2016]

Ab Schritt 3 wird für jedes Wort w in jedem Dokument d Iterationen ausgeführt, bis die Themenzuordnung der Wörter schließlich konvergiert und eine gute Annährung erreicht wird. Eine semantisch korrekte Abbildung der Themen erreicht LDA, indem es das Markov-Ketten-Monte-Carlo-Verfahren, Gipps-Sampling, in den Rechenschritten 3.a. und 3.b. anwendet. Gipps-Sampling zeichnet sich durch das Verwenden von dem Dirichlet-Hyperparameter α und dem Themen-Hyperparameter η[50] aus. Demnach wird folgende Formel angewendet, um die kaum bis wenig aussagekräftigen Zuordnungen in Schritt 2 in semantisch korrekte

[50] *Wird in anderen Quellen auch als β angegeben.*

Themenzuweisungen zu überführen. Anschließend folgt eine Erläuterung der Bestandteile der Formel [Vgl. Liu, 2015]:

$$P(z_i = j | z_{-i}, w_i, d_i) = \frac{C_{w_i j}^{WT} + \eta}{\sum_{w=1}^{W} C_{wj}^{WT} + W\eta} \times \frac{C_{d_i j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d_i t}^{DT} + T\alpha}$$

- $\mathbf{P(z_i = j)}$**:** Die Wahrscheinlichkeit, dass das Wort z_i dem Thema j zugewiesen ist.
- $\boldsymbol{z_{-i}}$**:** Stellt Themenzuordnungen aller anderen Wörter dar, bis auf das aktuell betrachtete Wort z_i.
- $\boldsymbol{w_i}$**:** Wort-Index des i-ten Wortes.
- $\boldsymbol{d_i}$**:** Dokument, dass das i-te Wort enthält.
- $\boldsymbol{C_{w_i j}^{WT}}$**:** Anzahl der Zuordnungen des Wortes w_i zum Thema j für alle Dokumente.
- $\boldsymbol{\sum_{w=1}^{W} C_{wj}^{WT}}$**:** Gesamtanzahl der Wörter, die dem Thema j zugeordnet sind.
- $\boldsymbol{W}$**:** Gesamtzahl aller Wörter in dem Korpus.
- $\boldsymbol{C_{d_i j}^{DT}}$**:** Anzahl der Zuordnungen des Wortes w im Dokument d, die dem Thema j zugewiesen sind.
- $\boldsymbol{\sum_{t=1}^{T} C_{d_i t}^{DT}}$**:** Gesamtanzahl von Worten in Dokument d.
- $\boldsymbol{T}$**:** Anzahl der definierten Themen.

Nachstehend wird die Dirichlet-Verteilung von Themen mit verschiedenen Dirichlet-Hyperparametern α abgebildet. Daraufhin folgt eine Erläuterung über die resultierenden Effekte, die bei der Veränderung des Wertes des Dirichlet-Hyperparameters α und Themen-Hyperparameters η auftreten:

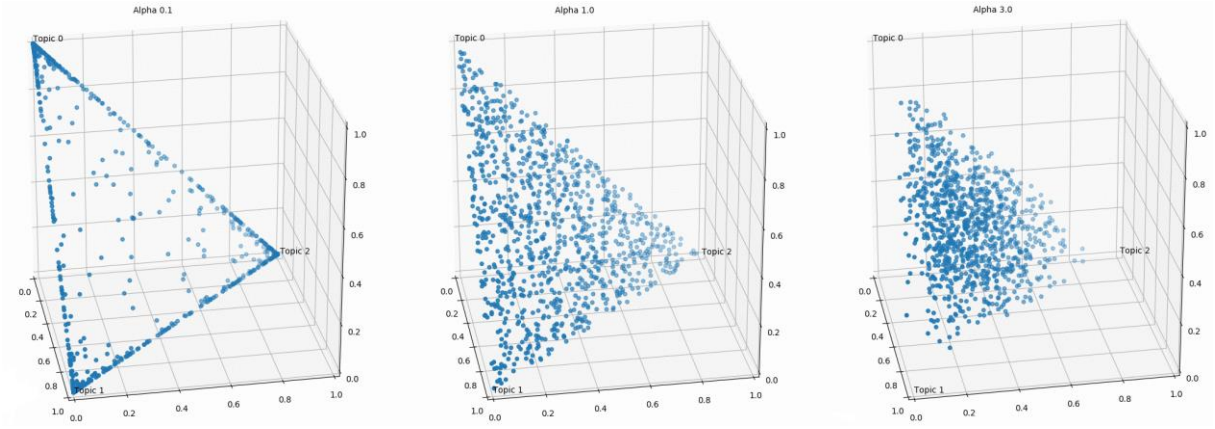


Abbildung 16: Dirichlet-Verteilung von 1000 Proben (Dirichlet-Hyperparameter α = 0.1, 1.0, 3.0) [Lettier, 2018]

Die Abbildung visualisiert die Verteilungen von 1000 Dokumenten auf drei Themen, unter Festlegung unterschiedlicher Dirichlet-Hyperparameter. Dabei stellt jeder Punkt ein Dokument und jede Ecke ein Thema dar. Je nach Positionierung des Punktes werden dem Dokument verschiedene Themenanteile zugeordnet.

Je höher der α Wert, desto höher ist die Wahrscheinlichkeit, dass jedes Dokument eine Mischung aus den meisten Themen, anstelle eines einzelnen Themas enthält. Das Ziel ist, aus

den einzelnen Dokumenten eine adäquate Anzahl an Themen zu filtern und dabei nicht zu detailliert oder oberflächlich zu untersuchen.
Wird für alle Dokumente $\alpha < 1$ festgelegt, sind die Punkte in der Nähe der Ecken positioniert. Ein sehr niedriger α Wert nahe 0, wie zum Beispiel die Koordinate (0.99, 0.0, 0.0), bedeutet, dass das Dokument nahezu ausschließlich einem Topic zugeordnet wird.
Wird $\alpha = 1$ gewählt, sind alle Punkte auf der Fläche bzw. dem 2-Simplex gleichmäßig verteilt. Somit würden für einige Dokumente Themenzuordnungen resultieren, die nahezu für alle Themen gleiche Anteile haben, überwiegend einem Thema zugeordnet wären oder Variationen zwischen den beiden letztgenannten Themenverteilungen aufweisen würden.
Ist der Wert $\alpha > 1$, so sammeln sich alle Punkte mit zunehmender Erhöhung des α-Wertes in der Mitte des 2-Simplexes. Daraus resultiert für alle Dokumente eine gleichmäßige Verteilung der Themen. [Vgl. Lettier, 2018]

Der η-Hyperparameter beeinflusst die Themenverteilung der Wörter innerhalb eines Dokumentes. Je höher der η-Wert ist, desto höher ist die Wahrscheinlichkeit, dass jedes Thema eine Mischung aus den meisten Wörtern und folglich keine spezifische Wörter enthält [Vgl. Liu, 2015]. Die Absicht ist, Themen zu generieren, die durch eine angemessene Wortanzahl repräsentiert werden und zur Interpretation und Definition des möglichen Hauptthemas beitragen.

Neben Topic Modelling kann LDA unter anderem auch für andere Anwendungsbereiche angewendet werden. So kann das Analysieren mit dem LDA-Verfahren durch leichte Anpassungen aus Umfragedaten, Audio- und Musik-Dateien, Computercodes, Bilddateien und sozialen Netzwerken wertvolle Informationen generieren [Vgl. Blei, 2011, S. 11 f.]. Beispielsweise kann ein LDA-Modell trainiert werden, damit es Objekte auf Bilder erkennt und die Objektnamen wiedergibt. Ein LDA-Modell kann auch spektrale Eigenschaften von Pflanzen, die unter Trockenstress leiden, identifizieren.

3.7 Entity Extraction & Fact Extraction mittels der HANA

Ziel der Einplanung von Entity und Fact Extraction ist die Analyse der Kunden-Kommentare von Tradefood GmbH. Die Kunden-Kommentare wurden zuvor mit Hilfe des LDA-Clustering-Algorithmus den themenspezifischen Beiträgen zugeordnet.

Entity Extraction stellt auch eine XML-Konfigurationsdatei in SAP HANA Studio dar. Sie wird während der Erzeugung des Full-Text Indexes in der SQL-Konsole des SAP HANA Studio aufgerufen. Die Konfigurationsdatei heißt „EXTRACTION_CORE“. Dieser ermöglicht aggregierte Informationen zu generieren und visualisieren. Der Extraktionsprozess kennzeichnet einzelne Tokens mit einem Entitätstyp. Anders als bei der linguistischen Analyse werden keine Schritte zur Textdatenvorverarbeitung unternommen. Die mit dem Entitätstyp gekennzeichneten Tokens haben eine unterschiedliche Anzahl an Worten. Entity Extraction greift auf eine vordefinierte oder benutzerspezifische Liste zu, die unter Verwendung von

Regeln die Zuordnung von Entitätstyp und Token realisiert. Entitätstypen stellen allgemeine Oberbegriffe dar. Es folgt eine Auflistung von einigen extrahierbaren Entitätstypen [Vgl. SAP SE, 2018g, S. 275 ff.]:

- ADDRESS
- GEO_AREA, COUNTRY, LOCALITY
- CURRENCY
- DATE, TIME,
- FACILITY, VEHICLE, WEAPON,
- HOLIDAY
- ORGANIZATION
- PEOPLE
- SOCIAL_MEDIA
- URL
- PHONE

Einige der oben aufgelisteten Entitätstypen stellen die allgemeinste Form dar. Nach der Ausführung resultieren für einige Entitätstypen detailliertere Ergebnisse, wie beispielsweise FACILITY@BUILDING, ORGANIZATION@SPORTS, URL@EMAIL oder VEHICLE@AIR. Die Entitätsextraktion erlaubt somit zu sehen, welche Entitäten in welchen Texten vorkommen und wie häufig sie auftreten.

Für eine einheitliche Darstellung von unterschiedlichen Schreibweisen ist der Teilprozess **Entity Normalization** bei der Entity Extraction mit einbegriffen. Die Normalisierung wird nach dem International Standards Organization-Standard (ISO-Standard) angewendet. Die Vereinheitlichung wird bei den Entitätstypen CURRENCY, DATE, MEASURE, PERCENT, TIME_PERIOD gebraucht. Die Tokens dieser Entitätstypen besitzen numerische (Mengen-) angaben, welche unter anderem als Zahl repräsentiert oder ausgeschrieben vorliegen können. Die HANA kann die ausgeschriebenen Werte in die numerische Darstellung überführen. Die Syntax zur Umwandlung der Entitätstypen in die jeweiligen ISO-Standards wurde in eine Tabelle zusammengefasst und sieht wie folgt aus:

CURRENCY	ISO 4217:2015	<Anzahl><Leerzeichen><drei Zeichen der Währungseinheit>
DATE	ISO 8601	<YYYY>-<MM>-<DD>
MEASURE	Einheit nach NIST[51] normalisiert	<Wert><Leerzeichen><Einheit>
PERCENT	Keine ISO-Standards vorhanden	<Anzahl>%
TIME_PERIOD	ISO 8601	PnnW; PnnYnnMnnD; PTnnHnnMnnS [52]

Tabelle 6: Syntax der Entity Normalization nach ISO-Standards [SAP SE, 2018g, S. 287 ff.]

[51] *National Institute of Standards and Technology.*

[52] *PnnW: Periode von nn Wochen; PnnYnnMnnD: Periode von nn Jahren nn Monaten nn Tagen; PTnnHnnMnnS= Zeitraum von nn Stunden nn Minuten nn Sekunden. Der Platzhalter nn steht für die Repräsentation von ganzen Zahlen, welche auch ausgeschrieben im Text vorkommen können.*

Die **Fact Extraction** kann als eine Erweiterung zur Entity Extraction gesehen werden. Die HANA hat drei XML-Konfigurationsdateien, welche dasselbe Resultat, wie die Entity Extraction generieren und zusätzlich je nach Konfigurationsdatei unterschiedliche Fakten extrahieren. Demnach sind die Konfigurationsdateien der Fact Extraction wie folgt unterteilt [Vgl. SAP SE, 2018g, S. 293].

- EXTRACTION_CORE_VOICEOFCUSTOMER
- EXTRACTION_CORE_ENTERPRISE und
- EXTRACTION_CORE_PUBLIC_SECTOR

Die zwei Letzteren sind nur in englischer Sprache verfügbar und werden deshalb nicht näher betrachtet [Vgl. SAP SE, 2018g, S. 330] [Vgl. SAP SE, 2018g, S. 350]. Die XML-Konfigurationsdatei, EXTRACTION_CORE_VOICEOFCUSTOMER, ist für die deutsche Sprache verfügbar und wurde im Rahmen dieses Projektes angewendet, um die Kunden-Kommentare der Tradefood GmbH zu analysieren.

Die Faktenextraktion wendet linguistische Analysen an und vergleicht Muster miteinander. Dazu werden Teilprozesse ausgeführt, die unter anderem das Verarbeiten von Sprachteilen, syntaktischen Mustern und Negationen umfassen, um relevante Muster zu identifizieren. [Vgl. SAP SE, 2018g, S. 293]

EXTRACTION_CORE_VOICEOFCUSTOMER soll aufgerufen werden, um eine Sentiment-Analyse auf den Kunden-Kommentaren durchzuführen. Dadurch sollen Informationen über die Gefühle und Handlungen von Kunden im Bereich der zugrundeliegenden Domäne extrahiert werden. Die Sentiment-Analyse extrahiert aus dem Textkorpus mit Hilfe von Entitätstypen und Regeln die Sentiments, Emoticons, Anfragen und Kraftausdrücke.

In der Standard-HANA-Konfiguration, EXTRACTION_CORE_VOICEOFCUSTOMER, werden folgende Entitätstypen erfasst:

Sentiment	MajorProblem	StrongNegativeEmoticon
StrongPositiveSentiment	Topic (Sentiment)	Request
WeakPositiveSentiment	Emoticon	GeneralRequest
NeutralSentiment	StrongPositiveEmoticon	ContactRequest
WeakNegativeSentiment	WeakPositiveEmoticon	Topic (Request)
StrongNegativeSentiment	NeutralEmoticon	AMBIGUOUSPROFANITY
MinorProblem	WeakNegativeEmoticon	UNAMBIGUOUSPROFANITY

Tabelle 7: Anzahl vordefinierter Entitätstypen der „EXTRACTION_CORE_VOICEOFCUSTOMER“ Konfiguration [SAP SE, 2018g, S. 361 ff.]

Dabei werden die Hauptentitäten Sentiment, Emoticon und Request weiter differenziert. So ergibt sich zum Beispiel für Tokens, die Sentiments abbilden, eine Einordnung in neutrale, negative und positive Sentiments. Positive und negative Sentiments werden wiederum in schwache und starke Sentiments unterschieden. Die Unterscheidung von Emoticons wird analog zu den Sentiments durchgeführt. Außerdem wird zwischen allgemeinen Anfragen und

Anfragen, welche eine Kontaktierung erwünschen bzw. erfordern, unterschieden. Zudem werden auch kleine und große Probleme aus dem Text extrahiert. Diese Extraktion von Fakten ermöglicht den Unternehmen aus großen Datenmengen gezielt kritische Bemerkungen von Kunden ausfindig zu machen und angemessen zu agieren.

Bei Bedarf kann das von SAP vordefinierte sogenannte „thesaurus“ Wörterbuch und die Extraktionsregeln[53] an eigene Anforderungen angepasst werden. Extraktionsregeln werden in einer musterbasierten Sprache geschrieben, die die Durchführung von Mustervergleichen mit Token- oder Zeichenbasierten regulären Ausdrücken, in Kombination mit linguistischen Attributen, erlaubt. Mit CGUL-Regeln können somit benutzerdefinierte Entitätstypen angelegt werden. [Vgl. SAP SE, 2018f, S. 30]

Es können weitere Sentiment-Schlüsselwörter hinzugefügt und (Sub-)Entitätstypen nach gegebenen Anforderungen angepasst werden. So können beispielsweise Tokens, die im Standard einem MinorProblem (kleines Problem) zugeordnet werden, zu einem MajorProblem (großes Problem) umgeändert werden. Die Entscheidung für diese Änderungen ist davon abhängig, inwiefern ein Unternehmen den als MinorProblem markierten Token als MajorProblem ansieht.
Ein anderes Beispiel: Die Phrase „verdammt geil“ nehmen Menschen als eine besonders positive Bemerkung wahr. Die HANA würde dieses Sentiment als neutral Stufen, da laut HANAs Definition „verdammt“ eine negative und „geil“ eine positive Bemerkung darstellt. Dieses Interpretationsproblem kann mit der CGUL-Regel berichtigt werden. Diese Situation kann auch behoben werden, indem die Phrase „verdammt geil“, wie im Anhang 1 abgebildet, in das Wörterbuch eingepflegt wird.

Unter Berücksichtigung der Abgabefrist des vorliegenden Projektes werden benutzerdefinierte Regeln und Wörterbücher im Rahmen der Entity und Fact Extraction nicht konzipiert.

3.8 Visualisierung

Die Visualisierung der resultierenden Ausgabedaten der Clustering-Analyse, sowie der Fact und Entity Extraction soll mit der Software Lumira Discovery von SAP umgesetzt werden. Die Nutzung einer Lösung von SAP für die Durchführung dieser Aufgabe trägt dazu bei, dem primären Ziel des Projektes weiterhin zu folgen. Außerdem hat sich Lumira Discovery als ein verbreitetes unternehmenstaugliches Produkt ausgezeichnet, welches erfolgversprechende Indikatoren für die Reife und Effektivität eines Produktes sind.

Lumira Discovery greift nicht direkt auf die physischen Tabellen in der HANA zu. Die Visualisierungslösung führt vielmehr einen lesenden Zugriff auf modellierte Datensichten (Views) aus. Dies stellt die Notwendigkeit dar, dass, nach den abgeschlossenen

[53] *Von SAP als Custom Grouper User Language-Regeln (CGUL-Regeln) bezeichnet.*

Analyseprozessen, Views, der zu visualisierenden Tabellen, in SAP HANA Studio erstellt werden müssen.

Für das Projekt wird die Entwicklung des View-Typs, Calculation View, in der SAP HANA Studio-Umgebung eingeplant. Die Calculation View soll graphisch modelliert werden. Die Tätigkeiten zur Entwicklung der Calculation View und nachfolgender Visualisierung kann in Kapitel 4.5 eingesehen werden.

Vom Funktionsumfang betrachtet, ist neben den View-Typen, Attribute und Analytic View, die Calculation View der umfangreichste View-Typ. Die Calculation View kann eine Kombination von beliebig vielen anderen View-Typen realisieren. Dahingegen kann die Attribute View nur Datenbanktabellen mit Join-Operatoren verknüpfen. Die Analytic View nutzt Attribute Views, um sie in einem Sternschema als Dimensionen zu verwenden, welche mit einer zentralen physischen Faktentabelle über Schlüsselbeziehungen verbunden sind.[54] [Vgl. Gahm et al., 2016, S. 228 f.]

Damit Lumira Disvocery die modellierten Views in der HANA lesen kann, müssen Leseberechtigungen auf das Schema, welches die Views vorhält, erteilt werden.

Nach der Erzeugung der Visualisierungen und Berücksichtigung der resultierenden Modelle werden die Darstellungen im Kontext der betroffenen Domäne, unter Hinzuziehung von Hintergrundwissen, interpretiert. Der Prozess der Interpretation resultiert in der Aufstellung von Hypothesen. Werden die Hypothesen belegt, entsteht Wissen, aus denen Handlungsmaßnahmen abgeleitet werden können.

[54] *Ein Starschema ist eine logische Datenmodellierungsvarianten eines multidimensionalen Datenmodells, welche sich durch Dimensionen auszeichnet, die an einer zentralen Faktentabelle gebunden sind. Dimensionen können beispielsweise Zeitangaben, Produkte, Standorte oder Kunden beschreiben. Dimensionen bewirken unterschiedliche Sichten auf die Fakten. Fakten sind betriebswirtschaftliche Kennzahlen die zum Beispiel Warenbestände, Umsatzerlöse und –mengen, sowie Stückkosten beschreiben. [Vgl. Kemper et al., 2010, S. 66 f.]*

4 Umsetzung

Es folgt nun das Kapitel zur technischen Umsetzung der geplanten Unternehmungen auf der HANA. Die geplanten Teilprozesse zur Erreichung des primären und sekundären Ziels in Kapitel 3.1, werden chronologisch in die Praxis umgesetzt. Zudem wird auch festgestellt, ob der Plan befolgt werden konnte und an welcher Stelle Komplikationen bzw. Abweichungen zustande gekommen sind. Die Teilprozesse bauen aufeinander auf. Beim Eintritt von Komplikationen wird versucht, selbstentwickelte Alternativen zu erzeugen, damit die Folgeprozesse ausgeführt werden können. Die Ausgabedaten jeder Teilprozesse werden ebenfalls erläutert.

Die technische Umsetzung fängt mit der Extraktion, der geteilten Beiträge und Kunden-Kommentare von dem Facebook-Konto der Tradefood GmbH an (Kapitel 4.1). Anschließend wird auf der HANA die Vorverarbeitung der Beiträge, ohne die Berücksichtigung der Kunden-Kommentare, ausgeführt (Kapitel 4.2). Danach werden die aufbereiteten Textbeiträge, die seitens Tradefood GmbH auf Facebook geteilt wurden, mit dem Latent Dirichlet Allocation-Clustering-Algorithmus auf latente Themen untersucht (Kapitel 4.3). Nach erfolgreicher Evaluierung der Ergebnisse des LDA-Algorithmus werden die Kunden-Kommentare mit den themenspezifischen Beiträgen verknüpft. Folglich wird eine nach Themen differenzierte Entity und Fact Extraction der Kunden-Kommentare vorgenommen (Kapitel 4.4). Letztlich erfolgt die Visualisierung der generierten Analyseergebnisse (Kapitel 4.5), wobei die Interpretation und Hypothesen-Aufstellung in Kapitel 5.2 vollführt wird.

4.1 Nutzung der Graph API

Um Daten andere Facebook-Seiten aus der Graph API extrahieren zu können, muss zunächst ein Zugangsschlüssel von Facebook beantragt werden. Für den Zugangsschlüssel wurde der sogenannte „Seitenzugriff auf öffentlichen Content“ angefragt, um auf anonymisierte öffentliche Daten einen Lesezugriff auszuführen. Öffentliche Daten, im Sinne von Facebook, sind Metadaten zu Unternehmen, sowie öffentliche Kommentare, Beiträge und Bewertungen auf anderen Seiten.

Es wurde vorab, um die Wartezeit des Antrags zu überbrücken, die Entwicklungsumgebung Spyder testweise verwendet. Dabei wurde die Datenbankverbindung zur HANA instanziiert. Im weiteren Verlauf wurde die Graph API in der Entwicklungsumgebung von Spyder genutzt, um die eigenen Daten aus Facebook testweise über eine Request-Anfrage bzw. HTTPS-Anfrage zu extrahieren und anschließend in die HANA zu laden.

Trotz früher Beantragung eines Zugangsschlüssels, steht die Prüfung seitens Facebook weiterhin aus. Unter Betrachtung der Abgabefrist der vorliegenden Arbeit wurde die Entscheidung getroffen, auf zuvor von der Infomotion GmbH extrahierte Facebook-Beiträge und die zugehörigen Kommentaren zuzugreifen. Die öffentlichen Daten sind von dem Facebook-Konto der Tradefood GmbH extrahiert worden. Diese Daten sind in zwei CSV-

Dateien abgespeichert und wurden über den Assistenten von SAP HANA Studio in das Schema TENANT_ARKI der HANA importiert.

Anhang 2 zeigt das Assistenten-Dialogfenster zum Importieren einer Tabelle. Die Tabellen wurden spaltenorientiert abgespeichert, die Attributbezeichnungen wurden in die Großschreibweise überführt und die Datentypen der Attribute wurden festgelegt. Die Namen der importierten Quelltabellen lauten CORPORATION_RAW_POST_DATA und CORPORATION_RAW_COMMENT_DATA.

Es besteht eine Abhängigkeit zwischen den Tabellen, die sich durch die Beiträge und Kommentare ergeben. Die Abhängigkeit kann im Kontext der Datenmodellierung als eine 1:n-Beziehung zwischen Beiträgen und Kommentaren angesehen werden. Das heißt, ein Beitrag kann mit mehreren Kommentaren verknüpft sein. Ein Kommentar muss hingegen immer zu einem Beitrag geordnet sein.

CORPORATION_RAW_POST_DATA beinhaltet die Facebook-Beiträge von Tradefood GmbH und hat folgende Attribute und Datentypen:

- POST_ID (NVARCHAR): Eine eindeutige Identitätsspalte der Beiträge, die alle anderen Spalten der Tabelle identifiziert. Die POST_ID wird als Primärschlüssel deklariert.
- POST_CREATED_TIME (DATE): Das Datum, an dem der Beitrag geteilt wurde. Der Zeitraum, den die Spalte abbildet ist vom August 2013 bis Februar 2018.
- POST_MESSAGE (NVARCHAR): Der textuelle Inhalt des geteilten Beitrags.

CORPORATION_RAW_COMMENT_DATA beinhaltet die Kommentare der Kunden in Bezug auf die Beiträge von Tradefood GmbH. Der Primärschlüssel und die Datentypen sehen wie folgt aus:

- ID (INTEGER): Eine fortlaufende Zahl beginnend bei 100, die alle restlichen Spalten identifiziert.
- POST_ID (NVARCHAR): Die Identitätsspalte der Beiträge.
- COMMENT_ID (NVARCHAR): Eine eindeutige Identitätsspalte der Kommentare, die alle anderen Spalten der Tabelle identifiziert. Die COMMENT_ID wird als Primärschlüssel deklariert.
- COMMENT_CREATED_TIME (DATE): Das Datum, an dem die Kommentare geteilt wurden. Der Zeitraum den die Spalte abbildet ist ebenfalls vom August 2013 bis Februar 2018.
- COMMENT_FROM_NAME (NVARCHAR): Die Namen der Facebook-Mitglieder, die die Kommentare geschrieben haben.
- COMMENT_FROM_ID (NVARCHAR): Eine eindeutige Identitätsspalte der Spalte COMMENT_FROM_NAME.
- COMMENT_MESSAGE(NVARCHAR): Der textuelle Inhalt des geteilten Kommentars.

4.2 Vorverarbeitung der Beiträge

In dem vorliegenden Kapitel werden die in der SAP HANA Studio-Umgebung angewendeten Verfahren vorgestellt, um die Spalte POST_MESSAGE der Tabelle CORPORATION_RAW_-POST_DATA aufzubereiten, damit diese von dem LDA-Algorithmus (siehe Kapitel 4.3) verarbeitet werden kann.

Zunächst wurden durch eine SQL-Anweisung die leeren Zellen der Spalte, POST_MESSAGE, entfernt. Die Datum-Spalte, POST_CREATED_TIME, wies keine leeren Werte auf. Der Primärschlüssel POST_ID wurde nicht weiter geprüft, da sie eindeutig ist und bei der Erzeugung des Primärschlüssels keine leeren Werte erlaubt. Anschließend wurde der Full-Text-Index (siehe Kapitel 3.5.1) auf die Spalte, COMMENT_MESSAGE, der Tabelle, CORPORATION_RAW_POST_DATA, angewendet. Während der Erzeugung des Indexes wurden einige Parameter mitgegeben, damit die linguistische Textanalyse ausgeführt wird und die Stopwords entfernt werden. Der Code für das Löschen der leeren Zellen und der Erzeugung des Full-Text-Index kann im Anhang 3 eingesehen werden. Es folgt eine kurze Beschreibung der verwendeten Parameter:

- FAST PREPROCESS OFF: Die Funktion dieses Parameters konnte nicht herausgefunden werden. Die Dokumentation von SAP gibt nur einen Hinweis, ob dieser Parameter ein- oder ausgeschaltet werden soll. Wenn Textanalysen ausgeführt werden sollen, muss dieser Parameter explizit deaktiviert werden.
- MIME TYPE 'text/plain': Es wird der Medientyp der indizierten Spalte angegeben. Der Spalte wird der Medientyp plain text (einfacher Text) zugeordnet, da die Beiträge unter Verwendung eines Zeichencodes unstrukturierte Daten abbilden und keinem bestimmten Format, wie beispielhalber HTML oder PDF, unterliegen.
- ASYNC: Die Ausführung der auslösenden Anweisung wartet nicht auf die Beendigung der Vorverarbeitung und die Kontrolle kehrt zur weiteren Ausführung sofort zur Anwendung zurück.
- TEXT ANALYSIS ON: Aktivierung der Textanalyse-Funktionalität.
- CONFIGURATION 'LINGANALYSIS_STEMS': Mitteilung der zu verwendenden XML-Konfigurationsdatei zur linguistischen Textanalyse.
- LANGUAGE DETECTION ('DE'): Mitteilung der verwendeten Sprache in der indizierten Spalte. Tradefood GmbH ist in Deutschland tätig. Die textuellen Daten liegen in deutscher Sprache vor. Aufgrund dessen wird der Sprachcode „DE“ verwendet.
- TOKEN SEPARATORS ' \ / ; , . : - _ ( ) [ ] < > ! ? * @ + { } = " & ': Eine Gruppe von Zeichen, die als Indikator für die Trennung von Tokens verwendet werden.
- TEXT MINING ON: Aktivierung der Funktionalität zur Entfernung von Stopwords.
- TEXT MINING CONFIGURATION 'DEFAULT': Mitteilung der zu verwendenden Konfigurationsdatei. DEFAULT stellt eine vorkonfigurierte XML-Datei dar, die Stopwords verschiedener Sprachen enthält.

Die XML-Konfigurationsdatei, LINGANALYSIS_STEMS, normalisiert die Tokens und führt jeden Token auf den Wortstamm zurück (siehe Kapitel 3.5.3). Der Stemmer Guesser wurde auch aktiviert. Die Parametrisierung und Erläuterung der Programmzeilen der XML-Konfigurationsdatei werden in Anhang 4 beschrieben.

In der XML-Datei ist auch ein Funktionsbereich zur Erkennung der verwendeten Sprachen der indizierten Spalte und der Kennzeichnung dieser als solche aufgeführt. Diese Funktionalität wurde für unseren Anwendungsfall nicht benötigt und folglich nicht weiter erläutert. Die Spracherkennung muss allerdings obligatorisch in der XML-Datei vorhanden sein.

Die automatisch resultierende Ausgabetabelle $TA_MYINDEX2 nach der Erzeugung des Full-Text-Indexes wird im Anhang 5 abgebildet.[55] Die Tabelle mit dem Präfix, $TA_, weist auf die Ergebnisse der Prozesse des Parameters TEXT ANALYSIS CONFIGURATION hin.

Der Full-Text-Index wurde auf die Spalte POST_MESSAGE und der Tabelle CORPORATION_RAW_POST_DATA angewendet. Es folgt die Beschreibung der Spalten der Ausgabetabelle. Dabei werden die Spalten beschrieben, die für die Folgeprozesse weiterhin eine Bedeutung haben:

- POST_ID, TA_RULE, TA_COUNTER: Stellen den Primärschlüssel der Tabelle dar. Die POST_ID ordnet die Tokens zu den Beiträgen zu. TA_RULE gibt an, welche Art von Analysen gemacht wurden. Die Analysearten sind zum Beispiel die linguistische-, grammatikalische Analyse oder Entity Extraction. TA_COUNTER zeigt die Reihenfolge aller Tokens in einem Dokument auf.
- TA_TOKEN enthält die extrahierten Tokens.
- TA_NORMALIZED hält die normalisierten Tokens vor.
- TA_STEM beschreibt auf den Wortstamm reduzierten Tokens.

Es lagen nun normalisierte und auf den Wortstamm zurückgeführte Tokens vor. Jedoch musste festgestellt werden, dass das Stemming an einige Stellen nicht erwartungsgemäß funktionierte. So kam es vor, dass einige identische Tokens in der Spalte TA_TOKEN korrekt und fehlerhaft auf den Wortstamm zurückgeführt wurden. Beim einlesen in die Tabelle $TA_MYINDEX2 wurde auch festgestellt, dass das Stemming die ursprüngliche Bedeutung von Tokens verfälscht hatte. Überdies konnte, trotz der Aktivierung des Stemmer Guesser, keine Änderungen, im Vergleich zu der Anzahl von Wortstammbildungen von Tokens ohne den Stemmer Guesser, festgestellt werden.

Zudem wurde erkannt, dass Tokens, die schon normalisiert vorliegen oder erst normalisiert werden müssen und den Wortstamm schon abbilden, in den Spalten TA_TOKEN und TA_NORMALIZED belassen wurden, statt in die Spalte TA_STEM eingefügt zu werden.

---

[55] *Jede Anlegung eines Full-Text-Index, unter Verwendung der Parameter TEXT ANALYSIS und TEXT ANALYSIS CONFIGURATION <Name der XML-Konfigurationsdatei>, erzeugt automatisch eine Tabelle $TA_<Indexname>.*

Es folgen einige Beispiele aus der Tabelle $TA_MYINDEX2 die die Variationen der Normalisierung und des Stemmings darstellen:

Variationen	TA_TOKEN	TA_NORMALIZED	TA_STEM
normalisieren ∧ Wortstamm bilden	Dosen	dosen	Dose
normalisieren ∧ Wortstamm schon gebildet	Eintopf	eintopf	?
schon normalisiert ∧ Wortstamm bilden	empfohlen	empfohlen	Empfehlen
schon normalisiert ∧ Wortstamm schon gebildet	engagieren	engagieren	?

Tabelle 8: Gegenüberstellung der Variationen der Tokens nach Anwendung des Normalisierungs- und Stemming-Vorgangs (Quelle: Eigene Tabelle)

Für die weiteren Verarbeitungen der Tokens musste deshalb eine SQL-Anweisung erstellt werden, die alle normalisierten und auf den Wortstamm umgewandelten Tokens in einer Spalte einer neuen Tabelle zusammenführt. Der SQL-Code kann im Anhang 6 nachvollzogen werden. Die erstellte Tabelle CORPORATION_POST_DATA_STEMMED_AND_WITH_-STOPWORDS wird im Anhang 7 abgebildet. Nun liegen alle Tokens, welche auf drei Spalten[56] verteilt waren, in einer Spalte POST_MESSAGE vor und sind jeweils der entsprechenden POST_ID zugeteilt. Die Spalten TA_COUNTER und TA_RULE wurden in der Tabelle CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS nicht mitaufgenommen. TA_RULE stellte einen konstanten Wert dar, der die Art der Analyse angab. TA_COUNTER beschrieb die Reihenfolge der Tokens in einem Dokument. Für den LDA-Algorithmus ist die Reihenfolge der Tokens im Dokument allerdings nicht relevant.

Die Tokens der Spalte POST_MESSAGE der Tabelle CORPORATION_POST_-DATA_STEMMED_AND_WITH_STOPWORDS enthielt jedoch auch Stopwords. Nach dem Vorgehensmodell aus Kapitel 2.3.2.1 erfolgt die Bereinigung von Stopwords vor dem Stemming und der Normalisierung von Tokens. Mit dem Hinzufügen der zusätzlichen Parameter TEXT MINING ON und TEXT MINING CONFIGURATION 'DEFAULT', bei der Anlegung des Full-Text-Indexes wurde eine Konfigurationsdatei aufgerufen, welche Stopwords bereinigte. Die resultierende Tabelle beinhaltete für jede POST_ID die normalisierten und von Stopwords bereinigten Tokens, welche jedoch nicht die die Grundform von Tokens abbildete. Außerdem wurde in einem Beitrag öfter erwähnte Tokens nur einmal aufgeführt. Des Weiteren konnte erkannt werden, dass viele Emoticons gelöscht wurden.

Somit mussten zur Entfernung der Stopwords weitere SQL-Logiken verbaut werden (siehe Anhang 8), welche auf der Tabelle CORPORATION_POST_DATA_STEMMED_AND-_WITH_STOPWORDS die Häufigkeiten der Tokens ermittelte. Anhand der Häufigkeiten der Tokens wurden alle Tokens gelöscht, die über einer selbstdefinierten Häufigkeitsgrenze lagen. Außerdem wurde eine neue Tabelle STOPWORD_DICT erstellt, welche typische Stopwords enthielt. Daraufhin wurden die Häufigkeiten der eindeutigen Tokens im Textkorpus, die unter der Häufigkeitsgrenze lagen, angeschaut (siehe Anhang 9) und gegebenenfalls manuell in

[56] *TA_TOKEN, TA_NORMALIZED, TA_STEM.*

STOPWORD_DICT eingepflegt. Die gesammelten Stopwords wurden mit den Tokens in der Spalte POST_MESSAGE und der Tabelle CORPORATION_POST_DATA_STEMMED-_AND_WITH_STOPWORDS auf Gleichgeit geprüft. Bei erfolgreicher Übereinstimmung, wurden die Tokens aus der Tabelle CORPORATION_POST_DATA_STEMMED-_AND_WITH_STOPWORDS gelöscht.

Letztlich wurden die Daten der Tabelle CORPORATION_POST_DATA_STEMMED-_AND_WITH_STOPWORDS in eine neue Tabelle CORPORATION_POST_DATA_PREPROCESSED übertragen. Somit wurde eine Tabelle generiert, die die Texte bzw. Beiträge jeder POST_ID in vorverarbeitetem Zustand abbilden konnte.

Im letzten Teilprozess des beschriebenen Vorgehensmodells zur Aufbereitung von Texten wurde die Term-Document-Matrix (siehe Kapitel 2.3.2.2) erstellt. Der LDA-Algorithmus operierte auf der Term-Document-Matrix, die aus der Tabelle CORPORATION_POST_DATA_PREPROCESSED generiert wurde. Die Erzeugung der Term-Document-Matrix geschah im Hintergrund, als die Prozedur zur Ausführung des Text-Mining-Algorithmus angestoßen wurde. Somit musste keine explizite Entwicklung von Programmcodes vorgenommen werden.

4.3 Einsatz des LDA-Algorithmus zur Themenfindung

Anhand der Tabelle CORPORATION_POST_DATA_PREPROCESSED wird in diesem Kapitel die Anwendung des Clustering-Algorithmus, Latent Dirichlet Allocation (siehe Kapitel 3.6), aus der PAL vorgestellt. Ziel ist, die Aufdeckung von latenten Themen der aufbereiteten Beiträge der Tradefood GmbH über die in den Beiträgen gesprochen wird. Dabei sind die Anzahl der Themen und der Inhalt der Themen verborgen. Anschließend werden die nach Themen gekennzeichneten Beiträge mit den Kunden-Kommentaren aus der Quelltabelle CORPORATION_RAW_COMMENT_DATA zusammengeführt. Folglich erhalten wir eine Tabelle, die das Thema von jedem Beitrag und die zugehörigen Kunden-Kommentare aufzeigen. Dies ermöglicht später auf themenspezifische Kommentaren die Sentiment-Analyse und Faktenextraktion auszuführen.

Der verwendete Programmcode zur Durchführung des LDA-Clustering-Algorithmus ist im Anhang 10 einzusehen. Es wurden erstmal benutzerdefinierte Datentypen erstellt. Danach wurden Ein- und Ausgabetabellen anhand der definierten Datentypen erzeugt. Diese Tabellen wurden der LDA-Prozedur als Parameter mitgegeben und anschließend wurde die Themenanalyse mit LDA durchgeführt. Die Anforderungen seitens SAP an die Anzahl der Tabellen und die Definition der Metadaten der Tabellen müssen eingehalten werden, damit sie von der LDA-Prozedur interpretiert werden können. Die Anforderungen sind für die Algorithmen der PAL unterschiedlich. Detaillierte Anforderungen zur korrekten Erstellung der

Ein- und Ausgabetabellen für den LDA-Prozeduraufruf können in [Vgl. SAP SE, 2018d, S. 125 ff.] nachgelesen werden.

4.3.1 Beschreibung der Eingabetabellen

Die virtuelle Tabelle (View) **LDA_T_DATA** beinhaltete die Attribute und Daten der physischen Tabelle bzw. Relation CORPORATION_POST_DATA_PREPROCESSED. Die View wurde im Prozeduraufruf als Datengrundlage mitgegeben, um die latenten Themen ausfindig zu machen. Es wurden folgende Attribute für die View-Erstellung selektiert:

- POST_ID zur Identifikation der aufbereiteten Facebook-Beiträge.
- POST_MESSAGE, welche die aufbereiteten Facebook-Beiträge enthält.

Die temporäre Tabelle **#LDA_T_PARAMS** wurde verwendet, um die Parameter des LDA-Algorithmus zu setzen. Sie wurde als temporäre Tabelle erstellt, da nur zum Zeitpunkt der Prozedurausführung die Parameter vorhanden sein müssen und anschließend wieder vom Speicher gelöscht werden können. Die Spalte PARAM_NAME enthielt die Namen der anzulegenden Parameter. INT_VALUE, DOUBLE_VALUE und STRING-VALUE beinhalteten die temporären Werte, die für die Parameter belegt wurden. Die Erläuterung der Parameter [Vgl. SAP SE, 2018d, S. 126 f.] und die belegten Parameter-Werte für das validierte LDA-Modell werden im Folgenden ausgeführt:

- TOPICS dient zur Festlegung der Anzahl an Themen, die gefunden und ausgegeben werden sollen. Der Parameter wurde mit dem Wert 11 belegt.
- BURNIN beschreibt die Anzahl der am Anfang ausgelassenen Gibbs-Iterationen. Der BURNIN-Parameter wurde mit dem Wert 20 besetzt.
- THIN beschreibt die Anzahl der zwischen Anfang und Ende ausgelassenen Gibbs-Iterationen. Den THIN-Parameter wurde der Wert 20 verliehen.
- ITERATION gibt die Anzahl an Gibbs-Iterationen an. Es wurde der Wert 2000 festgelegt.
- SEED legt den Initialwert für den zufälligen Iterationsprozess fest. Wenn kein konstanter Wert angeben wird, resultieren bei jeder Neuausführung des Programmcodes bei gleicher Parametrisierung leicht unterschiedliche LDA-Modelle. Der Parameter-Wert wurde auf 0 gesetzt, welcher einen Wert aus der aktuellen Uhrzeit des verwendeten Systems ableitete.
- ALPHA wurde in Kapitel 3.6 beschrieben. Es wurde der Wert 0.1 gewählt. Die Themenverteilung der Dokumente sollte möglichst spezifisch sein.
- BETA wurde ebenfalls in Kapitel 3.6 erklärt. Im Projekt wurde der Wert 0.01 deklariert, damit alle Themen eine Mischung aus möglichst spezifischen Wörtern erhalten sollte.
- THRESHOLD_TOP_WORDS gibt die Wörter mit der höchsten Wahrscheinlichkeit für jedes Thema aus, wenn die Wahrscheinlichkeit größer als der deklarierte

Schwellenwert ist. Im vorliegenden Projekt wurde ein Schwellenwert von 0.01 gewählt, um mehr Wörter für die Beurteilung des semantischen Inhalts der Themen auszugeben.

- INIT wird verwendet, um Gipps-Sampling zu aktivieren. Der Parameter wurde mit dem Wert 1 belegt, um Gipps-Sampling zu initialisieren.

4.3.2 Generierte Ausgabetabellen

Nach der Ausführung der LDA-Prozedur wurden in die Ausgabetabellen die Resultate registriert. Die generierten Ausgabetabellen werden im Folgenden erörtert:

Alle eindeutigen Wörter des Textkorpus wurden in der Ausgabetabelle **LDA_T_DICT** vorgehalten. Die Tabelle besteht aus den zwei Spalten WORD_ID und WORDS. Jedem eindeutigen Wort wird eine identifizierende Zahl zugeteilt. Die Tabelle kann als Wörterbuch aufgefasst werden und wird im Anhang 11 illustriert.

LDA_T_DOCUMENT_TOPIC_DISTRIBUTION setzt sich aus den Attributen POST_ID, TOPIC_ID und PROBABLITY zusammen. Die Tabelle zeigt die Wahrscheinlichkeiten der Themenzuordnung für jeden Beitrag mit jedem Topic an. Im Anhang 12 wird die Tabelle nach den Beiträgen mit den höchsten Wahrscheinlichkeiten für ein spezifisches Topic sortiert dargestellt.

Eine weitere Ausgabetabelle ist **LDA_T_TOPIC_TOP_WORDS**. Diese hat die Spalten TOPIC_ID und WORDS. Die Tabelle zeigte für jedes der 11 Topics die Wörter, die am Wahrscheinlichsten das Topic beschrieben, an (siehe Anhang 13).

Die Ausgabetabelle **LDA_T_TOPIC_WORD_DISTRIBUTION** besitzt die Attribute TOPIC_ID, WORD_ID, PROBABILITY. Sie bildet die Wahrscheinlichkeiten der Zuordnung von jedem eindeutigen Wort des Textkorpus zu jedem Topic ab. Im Anhang 14 ist die Tabelle deutlich aufgewiesen.

Eine weitere wichtige Ausgabetabelle ist die **LDA_T_STATS**, welche nur drei statistische Kennzahlen beinhaltet. Nach der Ausführung wurden folgenden statistischen Werte erfasst:

- DOCUMENTS beschreibt die Anzahl der Beiträge in dem Textkorpus, die in der View LDA_T_DATA enthalten waren und der LDA-Prozedur als Parameter übergeben wurden. Die Anzahl der Beiträge waren 999.
- VOCABULARY_SIZE zeigt die Anzahl der eindeutigen Wörter, die von der LDA-Prozedur aufgenommen wurden, auf. Die Gesamtanzahl der eindeutigen Wörter war demnach 6022.
- LOG_LIKELIHOOD beschreibt eine mathematische Funktion aus der Statistik, die eine Kennzahl zur Bewertung der Qualität des LDA-Modells berechnet. Je höher die Kennzahl, desto besser ist das generierte LDA-Modell. Nach der Exekution der LDA-Prozedur wurde der LOG_LIKELIHOOD-Wert -162534.76 errechnet.

Eine andere Ausgabetabelle im Rahmen der Anwendung des LDA beinhaltet die verwendeten Parameter, die für den LDA-Prozeduraufruf eingetragen wurden. Die Tabelle heißt **LDA_T_CV_PARAMETER** und kann im Anhang 15 angesehen werden.

4.3.3 Nutzung der Ausgabetabellen

Nach dem die LDA-Prozedur exekutiert und die in Kapitel 4.3.2 beschriebenen Tabellen ausgegeben wurden, musste ein neue Tabelle X_LDA_TOP_TOPIC_PER_DOCUMENT erstellt werden. Die neue Tabelle leitete sich aus der Tabelle LDA_T_DOCUMENT_TOPIC_DISTRIBUTION ab. Es wurde eine SQL-Anweisung geschrieben (siehe Anhang 16), welche für jeden Beitrag ausschließlich die Themenzuordnung mit der höchsten Wahrscheinlichkeiten selektierte. Da zur Erstellung der neuen Tabelle nur Zeilen selektiert wurden, blieben die Attribute POST_ID, TOPIC_ID und PROBABILTY weiterhin bestehen.

Die SQL-Anweisung musste entwickelt werden, da die ursprüngliche Ausgabetabelle LDA_T_DOCUMENT_TOPIC_DISTRIBUTION die Wahrscheinlichkeiten der Themenzuordnung für jeden Beitrag mit jedem Topic registriert hatte.

Die neue Tabelle beinhaltete insgesamt 1039 Beiträge, wovon 999 eindeutige Beiträge waren. Dies wies darauf hin, dass einige Beiträge mehr als einem Thema zugeordnet wurden.

Tabellen, die generiert wurden, wurden zur Erstellung von Calculation Views verwendet, damit die Views von der Visualisierungslösung Lumira Discovery gelesen werden konnten (siehe Kapitel 4.5).

Überdies folgte die Erstellung einer weiteren Tabelle, welche anhand der Verknüpfung der Quelltabelle CORPORATION_RAW_COMMENT_DATA und der Tabelle X_LDA_TOP-_TOPIC_PER_DOCUMENT auf Basis der Spalte POST_ID umgesetzt wurde. Die Tabelle wurde erstellt, um die nach Themen gekennzeichneten Beiträge mit den Kunde-Kommentaren zusammenzuführen. Es war somit möglich die Kunden-Kommentare nach Themen einzugrenzen und themenspezifische Analysen der Kommentare durchzuführen.
Dazu wurde eine SQL-Anweisung in SAP HANA Studio ausgeführt, die die Verbindung der Tabellen realisierte und die neue Tabelle X_LDA_FOUND_TOPICS_JOIN-_WITH_COMMENTS generierte (siehe Anhang 17). Die Tabelle enthielt die Spalten COMMENT_ID, POST_ID, TOPIC_ID, COMMENT_CREATED_TIME, COMMENT-_MESSAGE, welche in Kapitel 4.1 erörtert wurden. Zudem wurde die neue Spalte PROCESSED erstellt, dessen Zweck in Kapitel 4.4 beschrieben ist. COMMENT_ID und TOPIC_ID wurden als Primärschlüssel definiert, damit die Tabelle mit den Full-Text-Index indiziert werden kann. Ein Ausschnitt der Daten ist im Anhang 18 abgebildet. Diese Tabelle diente als Datenbasis für Entity und Fact Extraction.

4.4 Anwendung der Entity & Fact Extraction

Entity und Fact Extraction stellen die Analysearten dar, die wertvolle Informationen aus den Kunden-Kommentaren generieren sollen. Es sollte wiederholt werden, dass Fact Extraction auch die Sentiment-Analyse umfasst.

Die Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS wurde im Rahmen einer weiteren Erzeugung eines Full-Text-Indexes verwendet. Dabei wurde diesmal die XML-Konfigurationsdatei EXTRACTION_CORE_VOICEOFCUSTOMER zur Sentiment-Analyse herangezogen. Die indizierte Spalte war COMMENT_MESSAGE. Der Full-Text-Index konnte fehlerfrei ausgeführt werden und die Tabelle mit den Ergebnissen der Analyse wurde erzeugt. Allerdings war die Tabelle leer. Nach mehreren Versuchen und Fehleranalysen wurde zum Entschluss gekommen, dass eine bestimmte Anzahl der vielfältigen Emoticons und Symbolen in den Kunden-Kommentaren einen internen Konflikt auslösten, der nicht gemeldet wurde.

Daraufhin wurden nicht geplante Entwicklungen eingeleitet, dessen Schritte und Ergebnisse wie folgt aussahen:

- Es wurde der Full-Text-Index INDEX_COMMENT auf die Spalte COMMENT_MESSAGE der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH-_COMMENTS angewendet (siehe Anhang 19). Dabei wurde die XML-Konfigurationsdatei, LINGANALYSIS_BASIC, aufgerufen, welche die Tokens nur extrahiert und normalisiert, jedoch nicht auf den Wortstamm zurückführt.
- Die resultierende Tabelle $TA_INDEX_COMMENT wurde verwendet, um alle eindeutigen Tokens und dessen Unicode-Werte in einer separaten Tabelle X_EMOTICON_WITH_UNICODE zu speichern. Die Definition der Tabelle $TA_INDEX_COMMENT war identisch zu der Tabelle im Anhang 5 aufgebaut. Doch dieses Mal wurden statt Beiträge, Kommentare indiziert und die Spalte TA_STEM blieb leer.
- Anschließend wurden aus der Tabelle X_EMOTICON_WITH_UNICODE alle Tokens, die gewöhnliche Zeichen darstellten, über die Adressierung des Unicodes gelöscht. Es wurden auch oft benutzte Emoticons, die Gesichtsausdrücke und diverse andere Emoticons, wie Herzen, abbildeten, gelöscht. Somit blieben Emoticons und Symbole übrig, die Objekte oder andere Inhalte darstellten, die vermeintlich die Funktionsuntüchtigkeit der Sentiment-Analyse hervorgerufen hatten (siehe Anhang 20).
- Danach wurde die PL-SQL-Prozedur DELETE_EMOJI geschrieben und aufgerufen, welche alle Zeichen der Kommentare in der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS mit den Zeichen in der Tabelle X_EMOTICON_WITH_UNICODE vergleichte und bei Übereinstimmung gelöscht hatte (siehe Anhang 21). Außerdem wurde für jeden von der DELETE_EMOJI-Prozedur bearbeiteten Kommentar, die zugehörige Spalte PROCESSED mit dem Zeichen „X“ belegt, um den Fortschritt der Prozeduren zu verfolgen. Zugleich konnte die Prozedur

aufgrund der Spalte PROCESSED bei unerwarteten Abbrüchen, an dem zuletzt verarbeiteten Kommentar weiterarbeiten, ohne die verarbeiteten Kommentare erneut zu durchlaufen und prüfen.

Nach der erfolgreichen Ausführung der DELETE_EMOJI-Prozedur, wurde die Spalte COMMEN_MESSAGE auf leere Werte untersucht und bei Übereinstimmung gelöscht. Die leeren Werte entstanden, weil es Kommentare gab, die ausschließlich aus Emoticons und Symbolen bestanden und von der Prozedur gelöscht wurden.

Letztlich wurde die von der Prozedur verarbeitete Spalte COMMENT_MESSAGE und die anderen Spalten der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS verwendet, um den gesamten Inhalt in die neue Tabelle X_LDA_FOUND_TOPICS_JOIN-_WITH_COMMENTS_VOICEOFCUSTOMER zu kopieren. Die Spalten TOPIC_ID und COMMENT_ID wurden als Primärschlüssel deklariert.

Zum Schluss lagen nun Kommentare vor, welche keine Emoticons oder Symbole beinhalteten, die die Sentiment-Analyse beeinträchtigten.
Der Grund der Erstellung einer neuen Tabelle war, die schon indizierte Spalte der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS. Die neue Tabelle musste zur Durchführung der Entity und Fact Extraction ebenfalls indiziert werden.

Folglich wurde der Full-Text-Index INDEX_COMMENT_VOC auf der Spalte COMMENT_MESSAGE der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH-_COMMENTS_VOICEOFCUSTOMER ausgeführt. Damit die Sentiment-Analyse ausgeführt werden konnte, wurde die XML-Konfigurationsdatei EXTRACTION_CORE-_VOICEOFCUSTOMER beim Erzeugen des Full-Text-Indexes als Parameter mitgegeben (siehe Anhang 22).
Dabei ist zu beachten, dass keine Vorverarbeitungsprozesse, wie die Entfernung von Stopwords, Segmentierung, Normalisierung und Wortstammbildung von Tokens notwendig ist. Eine Vorverarbeitung der Kunden-Kommentare könnte evtl. die Anzahl der potentiell extrahierbaren Entitäten verringern. Zum Beispiel könnte durch die Segmentierung oder Stopword Bereinigung Emoticons, wie zum Beispiel „ :-) “ oder „ :(“, aufgelöst werden.

Die Ausgabe der automatisch generierten Tabelle $TA_INDEX_COMMENT_VOC ist im Anhang 23 dargestellt. In der Spalte TA_TYPE wurden alle extrahierten Entitätstypen ausgewiesen, die einem Token zugeteilt und von den Attributen TOPIC_ID und COMMENT_ID identifiziert wurden. Die extrahierbaren Entitätstypen wurden in Kapitel 3.7 erläutert.

Die Tabelle $TA_INDEX_COMMENT_VOC erlaubte im Anschluss, unter Hinzuziehung einer Zeitdimension, geeignete Visualisierungen zur Abbildung zeitlicher Entwicklungen von Entitätstypen darzustellen (siehe nachfolgendes Kapitel 4.5).

4.5 Visualisierung der Ausgabedaten

Der Aussagegehalt der in Kapitel 4.3 und 4.4 erstellten Tabellen wurde durchforstet und es wurden Gedanken über eine sinnvolle Zusammenführung der Tabellen gemacht. Die Konstruktion adäquater Verknüpfungen von Tabellen verfolgte das Ziel, Tabellen zu erstellen, die den Nutzer visuelle Einblicke in die Strukturen und Muster der Beiträge und Kommentare erlauben. In SAP HANA Studio wurden graphische Calculation Views erstellt, auf die über Lumira Discovery zugegriffen und anschließend Visualisierungen generiert wurden. Zudem wurde in der Calculation View die Definition geeigneter Dimensionen und Kennzahlen, welche die Attribute der Tabellen darstellten, vorgenommen. Die Dimensionen und Kennzahlen wurden anschließend in Lumira Discovery zur Parametrisierung der Visualisierungen verwendet.

An erster Stelle wurden Leseberechtigungen auf das Schema, auf dem alle im Rahmen dieser Arbeit erstellten Tabellen lagen, gegeben. Das Schema, auf dem entwickelt wurde, hieß TENANT_ARKI. Die Rechte, genauer gesagt die Objektprivilegien, wurden über die SAP HANA Administration Console-Perspektive in SAP HANA Studio an den Benutzer vergeben, so dass Lumira Discovery alle Views dieses Schemas lesen konnte.

Eine im vorliegenden Projekt erstellte Calculation View umfasste die Verbindung des Erstellungsdatums der Beiträge mit den nach Themen gekennzeichneten Beiträgen. Dazu wurde die Quelltabelle CORPORATION_RAW_POST_DATA mit der Tabelle X_LDA_TOP_TOPIC_PER_DOCUMENT, auf Grundlage der Spalte POST_ID, zusammengeführt.
Es folgte demnach die Entwicklung der Calculation View CV_TOPICS_PER_DOCUMENT (siehe Anhang 24). Dabei wurde zur effektiveren Interpretation der Daten die Datum-Spalte POST_CREATED_TIME als weitere Dimension verwendet, um Analysen unter Berücksichtigung von Zeiträumen zu erstellen.

Die Calculation View CV_TOPICS_PER_DOCUMENT ermöglichte folgende Visualisierungen:

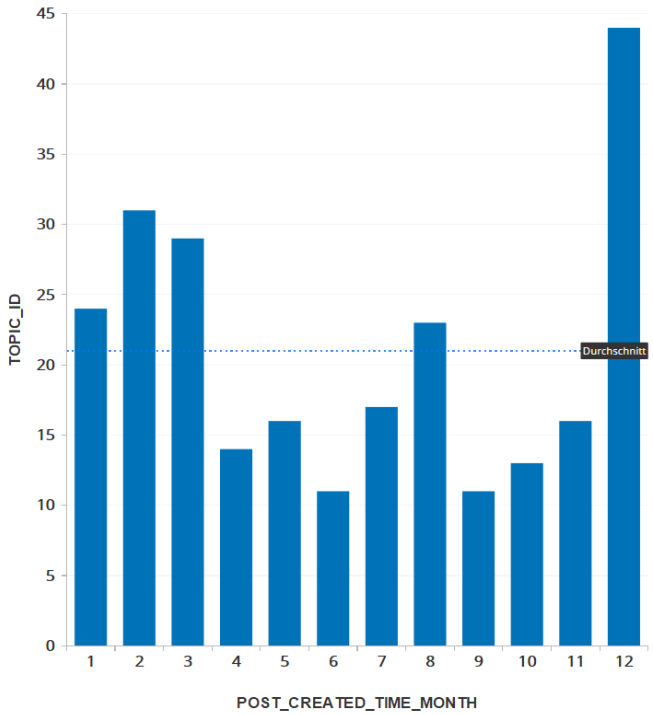


Abbildung 17: Anzahl der Gewinnspiel-Beiträge nach Monaten (2013-2018) (Quelle: Eigene Darstellung)

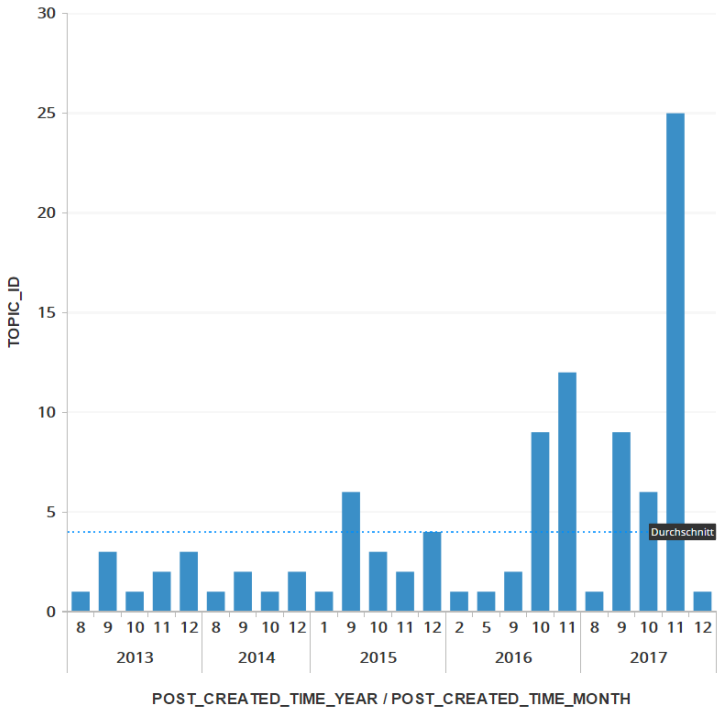


Abbildung 18: Anzahl der Beiträge über das Thema Spende für Kinder in den Jahren 2013 - 2017 (Quelle: Eigene Darstellung)

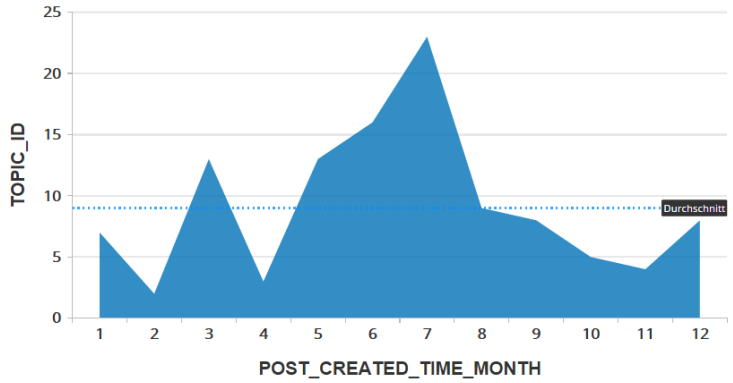


Abbildung 19: Anzahl der Beiträge über Eis nach Monaten (2013-2018) (Quelle: Eigene Darstellung)

Des Weiteren wurde eine Calculation View CV_TOP_WORDS_PER_TOPIC generiert (siehe Anhang 25). Die View wurde entwickelt, indem die Tabellen LDA_T_DICT und LDA_T_TOPIC_WORD_DISTRIBUTION, auf Basis der Spalte WORD_ID, zusammengesetzt wurden. Die in der virtuellen Tabelle zu verwendenden Attribute waren WORD_ID, WORD und PROBABILITY. Für die nachfolgenden Interpretationen und Visualisierungen der Inhalte der Themen spielte die Einbeziehung der Wörter eine wichtige Rolle.

Daraufhin wurde die Calculation View CV_TOP_WORDS_PER_TOPIC über Lumira Discovery gelesen und folgende zwei Diagramme erstellt:

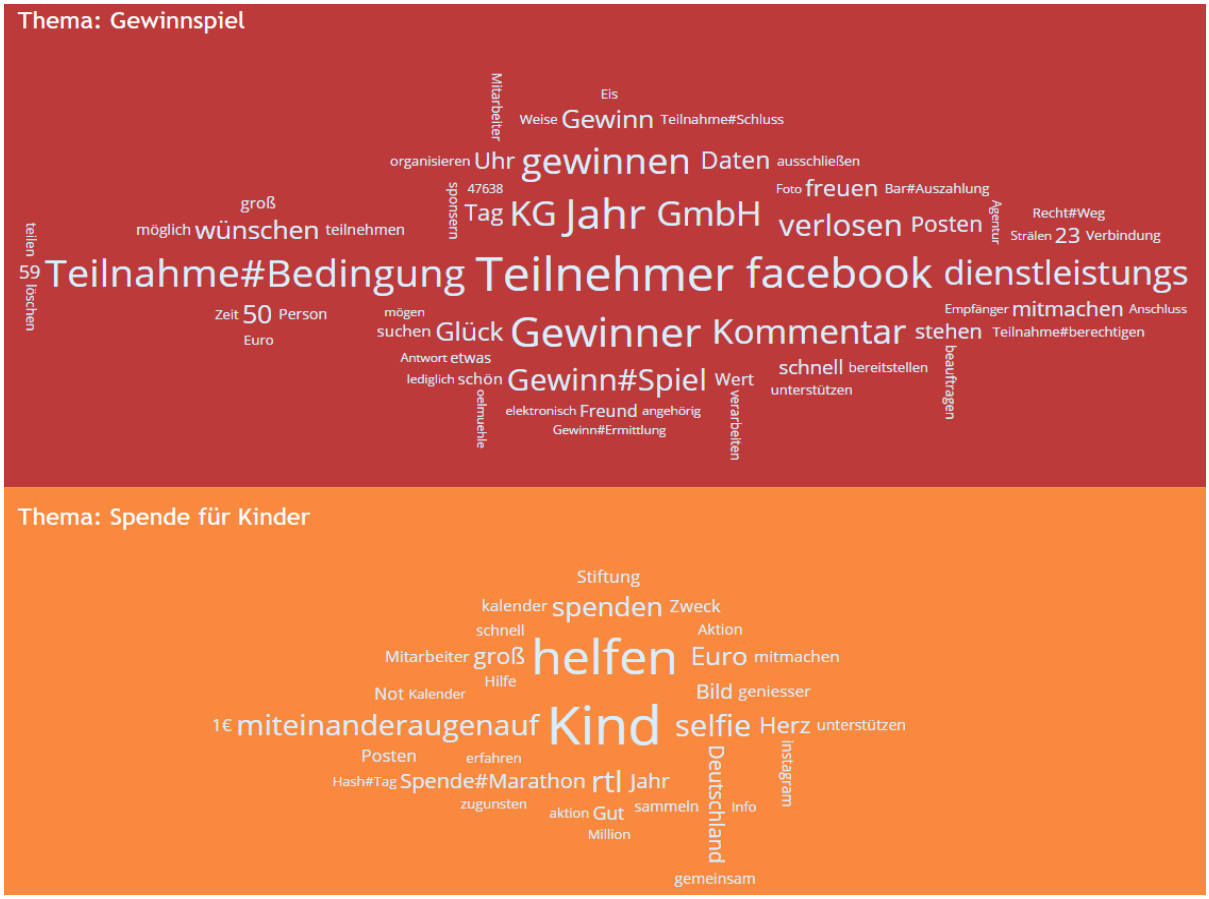


Abbildung 20: Visualisierung von Wort-Wolken über die Themen Gewinnspiel und Spende für Kinder (Wortgröße nach Wahrscheinlichkeit, dass das Wort das Thema beschreibt, geordnet) (Quelle: Eigene Darstellung)

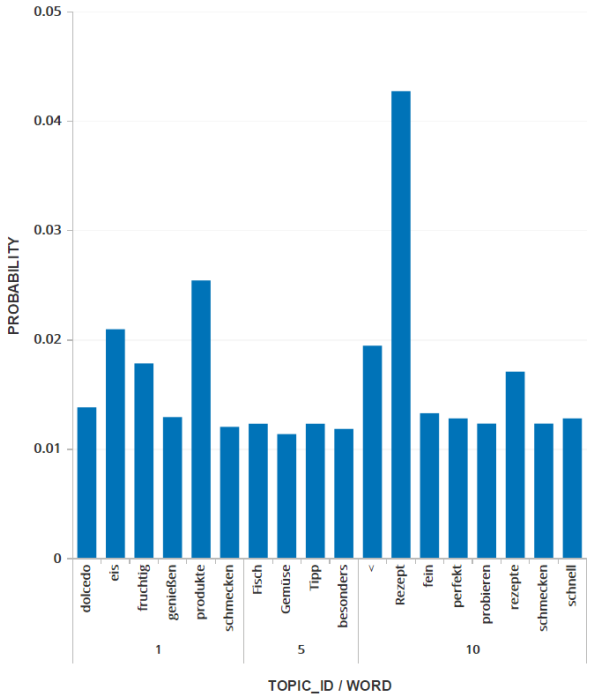


Abbildung 21: Gegenüberstellung der Wörter, die am Wahrscheinlichsten die Topics 1(Eis), 5 (Gesundes Essen) und 10 (Rezept-Ideen) abbilden (Quelle: Eigene Darstellung)

Die oben aufgeführten Abbildungen veranschaulichen Informationen, die auf Grundlage der Quelltabelle, welche die Beiträge der Tradefood GmbH darstellen, entstanden sind.
Anhand der durchgeführten Fact und Entity Extraction, wovon das erstere die Sentiment-Analyse umfasst, konnten aus den Kunden-Kommentaren der Facebook-Seite von Tradefood GmbH Informationen über die Kunden gewonnen werden.

Zu diesem Zweck wurde die Tabelle $TA_INDEX_COMMENT_VOC und die Quelltabelle CORPORATION_RAW_COMMENT_DATA auf Basis des Attributs COMMENT_ID, in der Calculation View CV_VOC zusammengefügt. In der Calculation View wurden nur relevante Attribute der zusammenzuführenden physischen Tabellen berücksichtigt. Demnach wurden in der View die Attribute COMMENT_ID, TOPIC_ID, TA_TOKEN, TA_TYPE und COMMENT_CREATED_TIME aufgenommen.[57] Die Datum-Spalte erlaubte die Tokens und Entitätstypen in einer Zeitdimension zu betrachten. Anhang 26 verdeutlicht, aus welchen Tabellen die Attribute der View gewählt und welche Verknüpfungsart (Join) verwendet wurde.

[57] *Die Beschreibung der Attribute ist in Kapitel 4.2 aufgelistet.*

Das Auslesen der Calculation View CV_VOC in Lumira Discovery erlaubte die nachfolgenden themenspezifischen Visualisierungen der extrahierten Entitätstypen der Kunden-Kommentare anzufertigen:

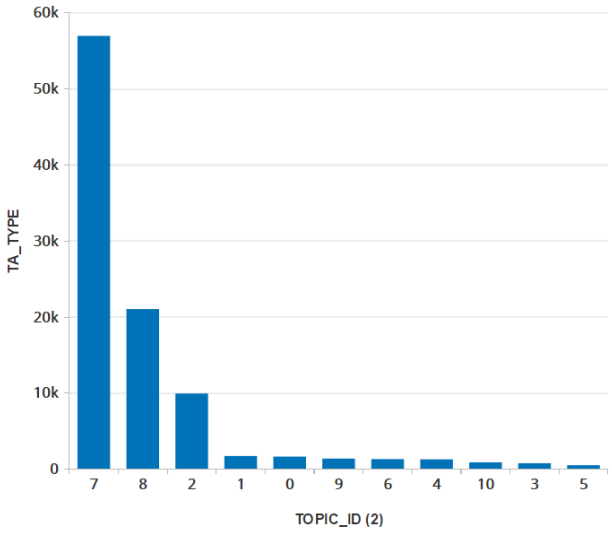


Abbildung 22: Anzahl aller extrahierten Entitätstypen pro Topic (Quelle: Eigene Darstellung)

Abbildung 23: Wort-Wolke über das Thema Gewinnspiele (Quelle: Eigene Darstellung)

Weitere Darstellungen der analysierten Kunden-Kommentare sind in Anhang 27, Anhang 28 und Anhang 29 dargestellt.

5 Ergebnis

Die zu unternehmenden Tätigkeiten und verwendenden Technologien im zugrunde liegenden Projekt wurden in Kapitel 3 analysiert und geplant. Die geplanten Ausführungseinheiten bzw. Prozesse auf den vorgestellten Technologien stellten zur Durchführung des Text-Mining-Projektes den Idealfall dar. Die Formulierung des Idealfalls wurde nach dem Status quo der existierenden Ressourcen im Bereich der Informationsverarbeitung definiert (siehe Kapitel 3.1). In der Phase der technischen Umsetzung wurden im Laufe des Projektes Erfahrungen gesammelt, die dazu gelten, diese zu bewerten. Das Kapitel 5.1 widmet sich der kritischen Würdigung der unternommenen technischen Prozesse und adressiert das primäre Ziel der Arbeit (siehe Kapitel 3.1). Die Bewertung der Qualität der fachlichen Erkenntnisse anhand der erstellten Visualisierungen werden im Hinblick auf das sekundäre Ziel in Kapitel 5.2 ausgetragen.

5.1 Evaluierung des Mehrwerts von Text-Mining auf Basis der HANA

Im Folgenden wird die technische Bewertung der HANA, im Kontext von Text-Mining, bezüglich des Mehrwerts evaluiert. Dabei werden auch Verbesserungswünsche, die im Vergleich zu gegenwärtigen alternativen Lösungen schon umgesetzt wurden, niedergeschrieben. Da SAP einer der größten Softwarehersteller neben Microsoft und IBM ist, wurden für dieses Projekt dementsprechend hohe Erwartungen an die HANA zugesprochen.

Außerdem muss beigefügt werden, dass wenige Quellen, bis auf die von SAP, zur Beschaffung von weitere Informationen existieren. Die Dokumentationen von SAP über HANA sind gelegentlich nicht ausreichend informativ, da Erläuterungen zu gewissen Funktionen, Datentypen oder Verfahren fehlen. Es wurde erkannt, dass die Community im Bereich Text-Mining auf Basis der HANA nicht groß ist und somit oft mit Quellen von SAP gearbeitet werden musste.

5.1.1 Datenbewirtschaftung der HANA

SAP bietet unter SAP HANA Smart Data Integration die Möglichkeit unter Verwendung von Datenbereitstellungsadapter eine Verbindung mit einer Vielzahl von Quellen herzustellen und Daten in SAP HANA zu integrieren. Im Bereich Social-Media-Mining wurde in Kapitel 3.4 erklärt, dass der Datenbereitstellungsadapter, Camel Facebook Adapter, keinen Mehrwert für Projekte erbrachte. SAP bietet für viele verschieden Quellen, wie Twitter, Oracle, Excel oder Outlook, Adapter an und befähigt Entwickler eigene Adapter mit einer Sammlung von Software-Entwicklungswerkzeugen zu konzipieren.

Wird in Zukunft geplant Social-Media-Beiträge zu extrahieren, um Funktionen zu testen, eine Datenbewirtschaftung zu vollführen oder dergleichen, sollte möglichst früh vor Projektbeginn ein Zugangsschlüssel bei Facebook beantragt werden. Aufgrund des Cambridge Analytica-Datenskandals hat Facebook Maßnahmen getroffen den Datenschutz zu verschärfen, dessen

Auswirkungen sich auch im Forschungsumfeld von öffentlichen Institutionen bemerkbar gemacht hat [Vgl. Bastos und Walker, 2018].

Die Python-Bibliothek pyhdb erlaubt einen einfachen Zugriff auf die HANA. Besitzt ein Entwickler viel Erfahrung in der Sprache Python, kann er zum Zweck der Datenverarbeitung fehlende oder nicht ausgereifte Funktionalitäten mit Python ergänzen. Jedoch kann keine Delegation eines Python-Skripts über SAP HANA Studio in die Python-Umgebung erfolgen.

Der Datenaustausch zwischen den SAP Produkten HANA und Lumira Discovery funktionierte zuverlässig und mühelos. Sollte ein kontinuierlicher Datenfluss erforderlich sein, kann die Datenübertragung in Echtzeit bewerkstelligt werden.

5.1.2 Datenaufbereitungsprozesse

Die Funktionen der Aufbereitung von Facebook-Beiträgen konnten nicht überzeugen. Außerdem fehlen einige grundsätzliche Funktionen.

HANA hat Funktionen zur Erkennung der Sprache. Diese Funktion ist für die Verarbeitung von multilingualen Texten in einer Textkorpora notwendig, falls keine Daten existieren, die die Sprache der Texte vorab gekennzeichnet haben. Nach mehreren Modifikationen der XML-Konfigurationsdatei zur Änderung der Parameter der Spracherkennung hat sich erwiesen, dass die resultierenden Zuordnungen der Sprachen nicht korrekt waren. Die Spracherkennung ist gegenwärtig keine anspruchsvolle Herausforderung und wird in digitalen Übersetzern effektiv und in Echtzeit betrieben.

Die Segmentierung von Wörtern bzw. Tokens aus den Texten kann die HANA gewährleisten. In HANA gibt es nur die Möglichkeit Fragmente bestehend aus einem Wort zu segmentieren. Es gibt allerdings auch Methoden, die Fragmente bestehend aus zwei oder drei Wörtern, segmentieren können. Diese Funktionalität ist eine Basiskompetenz im Rahmen des Text-Minings. Anhand dieser funktionalen Einschränkung kann gesehen werden, dass sehr einfache Methoden, die andere Analysemöglichkeiten eröffnen könnten, nicht in der HANA gegeben sind. Außerdem wurden Abkürzungen, wie „z.B.“, auch in einzelne Tokens segmentiert, obwohl die Dokumentation aussagt, dass Abkürzungen als Ganzes segmentiert werden.

Eine weitere Kompetenz im Rahmen der Vorverarbeitung von Texten, dessen korrekte Implementierung eine Selbstverständlichkeit darstellt, ist das Normalisieren von Tokens. HANA unterscheidet im Normalisierungsverfahren die Position eines Worts in einem Satz und wandelt dementsprechend die Großschreibweise eines Wortes bzw. Tokens in die Kleinschreibweise um. Jedoch führte dies dazu, dass der angewendete LDA-Clustering-Algorithmus dieselben Wörter, die in unterschiedliche Groß- und Kleinschreibung abgebildet wurden, nicht als gleichbedeutende Tokens erkennen konnte. Somit wurde die bessere Unterscheidbarkeit der gebildeten Themen nicht gewährleistet und ein unpräziseres Ergebnis

kam zustande.[58] Der richtige Ansatz wäre alle Tokens in die Klein- oder Großschreibweise zu überführen.

Die Wortstammbildung bzw. das Stemming von Tokens ist komplexer als die soeben angeführten Vorverarbeitungsschritte. Andererseits ist dieser Teilprozess des Text-Minings kein unerforschtes Gebiet. Stemming-Regeln sollte das Sprachmodul der HANA effektiv und schnell umsetzen können. Die HANA konnte im Projekt die Wortstammbildung von deutschsprachigen Facebook-Beiträgen nicht zufriedenstellend umsetzen. Somit wurde festgestellt, dass einige identische Tokens der Grundform entsprechend umgewandelt wurden und andere nicht auf den Wortstamm zurückgeführt worden sind.[59] Der Stemming Prozess hatte auch die semantische Bedeutung von einigen Tokens geändert.[60] Es konnte auch festgestellt werden, dass identische Tokens nach der Wortstammbildung unterschiedlich vorlagen.[61] Trotz Aktivierung des Stemmer Guessers konnten keine Änderungen der Anzahl an auf die Grundform überführten Tokens festgestellt werden.
Ein weiteres Problem, welches sich erwies, war die Art der Datenbewirtschaftung der automatisch erzeugten Tabelle, $TA_MYINDEX. Es musste eine SQL-Anweisung geschrieben werden, damit die nicht vollkommen zufriedenstellenden Ergebnisse der Datenaufbereitung in eine geeignete, weiter verwendbare Tabelle überführt werden. Das Problem der geeigneten Darstellung der Tabelle wurde in Kapitel 4.2 behandelt.
Das Stemming auf der HANA konnte einige gewöhnliche Tokens nicht auf die korrekte Grundform überführen. Dadurch resultierten mehrere eindeutige Tokens, die im Grunde dasselbe aussagten. Aufgrund dessen entsprach das Stemming nicht den Erwartungen.

Die Parameter TEXT MINING ON und TEXT MINING CONFIGURATION wurden im Rahmen des Projektes verwendet, um Stopwords zu entfernen. Die Parameter generierten insgesamt drei Tabellen, wovon zwei Tabellen laut SAP nicht verwendet werden sollten, da dessen Inhalte nicht gewährleistet sind [Vgl. SAP SE, 2018h, S. 22]. Die Namen der Tabellen haben das Präfix $TM_DOCUMENTS_ und $TM_TERMS_ und wurden im Rahmen des Projektes nicht weiter berücksichtigt.
Die dritte in Betracht gezogene Tabelle mit dem Präfix $TM_MATRIX_ führte die Tokens, welche normalisiert waren und von Stopwords bereinigt wurden, in einer gemeinsamen Spalte

[58] *Diese Auswirkung ist nicht nur auf den LDA-Algorithmus beschränkt und beeinflusst eine Vielzahl von Text Mining-Algorithmen.*

[59] *Es wurde wahrgenommen, dass die Tokens, Zwiebelkuchen, auf den Wortstamm, Zwiebel#Kuchen, umgewandelt wurden und ein anderes identisches Token nicht auf den Wortstamm zurückgeführt worden ist. Das normalisierte Token, Kaese, wurde an einer Stelle in das Token, Käse, und an anderer Stelle gar nicht umgewandelt.*

[60] *Aus dem Token, Brotsorte, wurde der Wortstamm, Brot#Ort, gebildet.*

[61] *Das Token, entspannen, wurde zu entspinnen und das Token, entspannt, wurde hingegen korrekt zu entspannen umgewandelt.*

auf.[62] Die Grundform der Tokens wurde trotz der entsprechenden Konfiguration in der Ausgabetabelle nicht berücksichtigt. Außerdem wurden viele Emoticons gelöscht. Zudem wurden Tokens, die in einem Facebook-Beitrag öfter vorkamen, nur einmal aufgeführt. Das Ergebnis der Anwendung der Parameter TEXT MINING ON und TEXT MINING CONFIGURATION war nicht zielführend und wurde deshalb nicht verwendet. Mittels SQL-Anweisungen wurde eine Alternative entwickelt, die die Stopwords entfernte (siehe Kapitel 4.2).
In der HANA wird die Entfernung von Stopwords nach der Normalisierung und Wortstammbildung durchgeführt. Ein performantes Vorgehen würde sich ergeben, wenn vor der Normalisierung und Wortstammbildung die Entfernung von Stopwords vorgenommen wird. Folglich würden vor dem Beginn der Normalisierung und des Stemmings weniger Tokens vorliegen. Diese Reihenfolge der Aufbereitungsschritte würde auch dem in Kapitel 2.3.2.1 vorgestellten Vorgehensplan entsprechen.

Die Umwandlung von Texte in eine numerische Vektordarstellung ist ein überschaubares statistisches Verfahren und muss für eine Vielzahl von Text-Mining-Algorithmen, als letzten Schritt in der Datenaufbereitung, umgesetzt werden (siehe Kapitel 2.3.2.2). Dieser Prozess ist notwendig, damit die Texte von den Text-Mining-Algorithmen interpretiert werden können. Die Vektorendarstellung aller Texte wird in einer Term-Document-Matrix abgespeichert.
In der HANA-Umgebung ist die Document-Term-Matrix nicht, wie in der Theorie gelehrt, abgebildet. Eine beispielhafte Term-Documten-Matrix, dessen Werte sich durch das TF-IDF Rechenverfahren ergeben und die Relevanz von Tokens in einem Dokument wiedergeben, wird in der Tabelle 4 veranschaulicht. Zudem besteht auch nicht die Möglichkeit eine Term-Document-Matrix als Parameter im Prozeduraufruf mitzugeben. Somit kann die Document-Term-Matrix nicht für andere PAL-Algorithmen im Kontext des Text-Minings verwendet werden.

Die Teilprozesse der Aufbereitung von Texten im HANA-Umfeld werden in der Regel richtig umgesetzt, auch wenn an einzelne Stellen Tokens falsch verarbeitet wurden. Es mussten jedoch Alternativen entwickelt werden, damit der weitere Folgeprozess die Anwendung von LDA, ausgeführt werden können. In jeder Phase der Aufbereitung gibt es einige Mängel, welche in Konkurrenzprodukten besser implementiert sind. Im Text- und Data-Mining dauert üblicherweise die Aufbereitung der Daten am längsten. Aufgrund der oben angeführten Komplikationen, dauerte die Aufbereitungsphase noch länger.

[62] *Die Tabelle mit dem Präfix $TM_MATRIX_ stellt die Term-Document-Matrix dar (siehe Kapitel 2.3.2.2). Diese war die einzige Tabelle, die berücksichtigt werden konnte, welche von Stopwords bereinigte Tokens enthielt.*

5.1.3 Anwendung von Text-Mining-Algorithmen der PAL

Text-Mining mit der Abfragesprache SQL ist zu Beginn gewöhnungsbedürftig und erfordert vom Datenanalysten Programmierkenntnisse im Bereich Datenbanken. Für die meisten Algorithmen in der PAL müssen die Ausgabetabellen, welche nach der Ausführung automatisch befüllt werden, nach den Anforderungen von SAP erstellt werden. Pro Algorithmus muss einmalig ein Gerüst in Form eines SQL-Skripts geschrieben werden, welches für unterschiedliche Tabellen angepasst wird und dann ausgeführt werden kann. Die verwendeten SQL-Anweisungen zur Ausführung des LDA-Algorithmus sind im Anhang 10 ausgewiesen.

Der LDA-Algorithmus konnte entsprechend der Qualität der Eingabedaten solide Ergebnisse liefern. Die fehlerhaft verarbeiteten Tokens des Normalisierungs- und Stemming-Prozesses waren auch Bestandteil der Eingabedaten. Auch wenn einige Tokens nicht richtig vorverarbeitet wurden, konnte der LDA-Algorithmus Themen generieren, über die tatsächlich in den Facebook-Beiträgen der Tradefood GmbH geschrieben wurde. Die HANA bietet auch die Funktionalität die Themenzuordnung für neue Beiträge, basierend auf den vorherigen LDA-Schätzergebnissen, abzuleiten. Die Funktion der Ableitung von Themenzuornungen für neue Texte adressiert Text-Mining-Projekte, welche eine Echtzeitanalyse bedingen und wurde deshalb nicht in diesem Projekt getestet.
Wünschenswert und effizienter wäre die automatische Erstellung von Views auf Basis der generierten Ausagabedaten, wie in Kapitel 4.3.3 vorgestellt wurde. Infolgedessen könnten die Ausgabetabellen, die zusammengeführt werden sollen, sofort visualisiert werden.

Da die Term-Document-Matrix nicht verwendet werden kann, konnten einfache und effiziente Algorithmen der PAL, wie der K-Mean-Clustering-Algorithmus, nicht ausgeführt werden. In der PAL ist der Clustering-Algorithmus LDA der einzige Algorithmus für Text-Mining, der aufbereitete Textdaten, ohne eine explizite Umwandlung in Vektoren, akzeptierte. Die Generierung der Term-Document-Matrix wird implizit im Hintergrund ausgeführt, woraufhin die Bearbeitungsschritte des LDAs angestoßen wurden.
Der implizite Aufruf sollte in Zukunft auch von anderen Algorithmen unterstützt werden, denn die PAL hat einige effiziente Text-Mining-Algorithmen, die somit auch Anwendung finden könnten. Aktuell können die Algorithmen in der PAL maßgeblich für Data-Mining-Anwendungsfälle verwendet werden.

5.1.4 Entity und Fact Extraction

Einginge relevante Fähigkeiten der Entity und Fact Extraction, welche im Rahmen dieses Projektes getestet wurden, können in Kapitel 3.7 nachgelesen werden. Die Anzahl der extrahierbaren Entitäts- und Faktentypen ist für deutschsprachige Texte, im Vergleich zu englischen und chinesischen Texten, nicht vollkommen ausgereift.

Im Rahmen der Faktenextraktion werden die Konfigurationsdateien EXTRACTION_CORE-_ENTERPRISE und EXTRACTION_CORE_PUBLIC_SECTOR in der deutschen Sprache nicht unterstützt. Somit können beispielshalber Entitätstypen, die Informationen über militärische Einheiten, Organisationen, Veranstaltungen, Produktankündigungen und Fusionen von Unternehmen enthalten, nicht extrahiert werden. [Vgl. SAP SE, 2018g, S. 328 f.] [Vgl. SAP SE, 2018g, S. 349 f.]

Es wurde ermittelt, dass die HANA die Kunden-Kommentare nicht lesen und die Extraktion von Fakten und Entitäten folglich nicht ausführen konnte. Es wird vermutet, dass die HANA bestimmte Emoticons und/oder Symbole nicht verarbeiten kann. Wie in Kapitel 4.4 beschrieben, musste eine Alternative entwickelt werden, um ausgewählte Emoticons und Symbole aus den Kunden-Kommentaren zu entfernen. Nach Anwendung der Prozedur funktionierte die Entity und Fact Extraction.
Die geschriebene Prozedur ist ressourcenlastig und hat ca. 50.000 Kunden-Kommentare in ca. 26 Stunden verarbeiten können. Das heißt, pro Stunde wurden durch die Prozedur etwa 2000 Kunden-Kommentare verarbeitet. Dabei muss berücksichtigt werden, dass die Kommentare auf Social-Media-Plattformen in der Regel kurze Texte abbilden.

Bis auf wenige Ausnahmen wurden Tokens mit den richtigen Entitätstypen gekennzeichnet. Falsch zugeordnete Tokens können herausgefunden und an die eigenen Anforderungen angepasst werden. Dazu können Wörterbücher erweitert oder Regeln definiert werden, die eine korrekte Zuordnung von Token und Entitätstyp erlauben. Einige Tokens, wie „Fehler“ wurden mit dem Entitätstyp MinorProblem assoziiert. Das lag daran, dass ein Beitrag der Tradefood GmbH die Kunden dazu aufforderte, einen Fehler in einem Bild zu finden. An dieser Stelle kann das Wörterbuch erweitert und ein geeigneter Entitätstyp für das Token „Fehler“ definiert werden.

Es gab während der Entity und Fact Extraction nur eine Komplikation, welche die Löschung einiger Emoticons und Symbole vorsah. Die Ergebnisse nach der Ausführung waren zufriedenstellend und informativ, obwohl keine benutzerdefinierte Regeln und Wörterbücher Anwendung gefunden haben.

5.2 Interpretation und Hypothesenformulierung anhand von Visualisierungen

Die Anwendung des Latent Dirichlet Allocation-Algorithmus konnte aus den Facebook-Beiträgen der Tradefood GmbH die latenten Themen aufdecken und die Beiträge den Themen zuordnen. Zudem erstellte der LDA-Prozeduraufruf Ausgabetabellen, die die Abbildung von Wörtern, die am wahrscheinlichsten ein Thema beschrieben, ermöglichte.

Es folgt die Interpretation der Analyseergebnisse und Visualisierungen. Zudem werden an einigen Stellen Hypothesen aufgestellt, dessen Wahrheitsgehalt durch Nachforschung der Tradefood GmbH ermittelt werden können.

Mittels der im Anhang 13 und der View CV_TOP_WORDS abgebildeten Tabellen, die die Inhalte der Themen aufzeigten, konnten folgende Themen zu den Werten der Spalte TOPIC_ID und der Tabelle LDA_T_TOPIC_TOP_WORDS zugewiesen werden:

TOPIC_ID	Thema
0	Gemüse
1	Eis
2	Verlosung von Eis (Gewinnspiel)
3	Rezept- und Bastel-Ideen
4	Grillen und Pizza
5	Gesundes Essen (Fisch, Obst, Gemüse)
6	Süßes Essen, Vor- und Nachspeisen
7 & 8	Gewinnspiel
9	Spende für Kinder
10	Rezept-Ideen

Tabelle 9: Darstellung der interpretierten Themen der Facebook-Beiträge (Quelle: Eigene Darstellung)

Außerdem konnten Abbildung 20 und Abbildung 21 einige Darstellungsmöglichkeiten von Themen abbilden, damit ein genaueres Bild über die themenspezifischen Inhalte der Facebook-Beiträge gemacht werden konnten. Die Abbildungen konnten unter Verwendung der Calculation View CV_TOP_WORDS erstellt werden.

Des Weiteren wurde die Abbildung 17 nachvollzogen. In den Jahren von 2013 bis 2018 stellten sich die Monate, Dezember, Januar, Februar und März als beachtenswert heraus, da in diesen Monaten viele Gewinnspiel-Beiträge geteilt wurden. Im Monat August wurden ebenfalls viele Beiträge über Gewinnspiele, im Vergleich zu den Monaten vor und nach dem August, geteilt. Über die Jahre betrachtet, wurden die meisten Beiträge über Gewinnspiele im Dezember geteilt.

Die Abbildung 18 demonstriert die Anzahl der Beiträge über Spenden für Kinder über die Jahre 2013 bis 2017 zugenommen hat. In dem Monat Dezember wurden vergleichsweise wenige Beiträge über das Thema Spenden für Kinder geteilt.
Die Höhe von Geldspenden steigt in der Regel zum Jahresende. Eine Studie von Statista zeigt auf, dass in Deutschland gegen Ende des Jahres 2017[63] über zwei Milliarden Euro gespendet wurde, wovon über eine Milliarde Euro im Dezember gespendet wurde [Vgl. Statista GmbH, 2018a]. Die Hypothese kann aufgestellt werden, dass Tradefood GmbH eine höhere Spendensumme erzielen wird, wenn im Dezember über den Social-Media-Kanal Facebook mehr Beiträge zum Thema Spenden für Kinder geteilt werden.

In den Sommer-Monaten bewarb Tradefood GmbH 2013 bis 2018 öfter Beiträge über Eis, als im Vergleich zu den Winter-Monaten. Anhang 19 bildet die Anzahl der Beiträge über Eis nach Monaten ab. Aus der Grafik geht hervor, dass im August wenige Beiträge gemacht wurden. Der August ist ein Monat, welcher in der Vergangenheit hohe Temperaturen in Deutschland aufwies. Unter Berücksichtigung der Temperaturen im August, sollte Tradefood GmbH die

[63] *Die berücksichtigten Monate sind Oktober, November und Dezember.*

Anzahl der Beiträge über Eis erhöhen und mehr werben. Es kann die Hypothese aufgestellt werden, dass im August ein höherer Absatz von Eis-Produkten erfolgen wird, wenn Tradefood GmbH über Facebook Werbung macht.
Die erfolgreiche Anwendung der Entity und Fact Extraction[64] ermöglichte aggregierte Informationen über Kunden-Kommentare mittels Visualisierungen zu erstellen. Demnach konnten Fakten und Entitäten, gegliedert nach den interpretierten Themen, dargestellt werden.

Die Abbildung 22 veranschaulicht für jedes Thema die Anzahl von allen extrahierten Entitätstypen in den Jahren 2013 bis 2018. Die Themen 8, 7 und 2 haben die höchste Anzahl an Entitätstypen und beschreiben, unter Beachtung der Tabelle 9, das zusammengefasste Hauptthema Gewinnspiel. Die Anzahl der Entitätstypen der restlichen Themen sind im Verhältnis zu dem Thema Gewinnspiel, sehr gering. Daraufhin wurde entschieden, das Thema Gewinnspiel näher zu betrachten.

Die Anzahl an positiven Kommentaren in Form von Texten und Emoticons für Beiträge über Gewinnspiele ist sehr hoch. Es wurden die Diagramme im Anhang 27 und Anhang 28 miteinander verglichen, welche ausschließlich die Anzahl der Entitätstypen, StrongPositiveSentiment, WeakPositiveSentiment, StrongPositiveEmoticon und Weak-PositiveEmoticon über die Zeit darstellen. Aus dem Vergleich konnte die Schlussfolgerung getroffen werden, dass Kunden der Tradefood GmbH die Gewinnspiel-Aktionen sehr gut empfinden und die meisten positiven Resonanzen aufgrund des Themas Gewinnspiel entstehen. Die Resonanzen wurde in Abbildung 23 als eine Wort-Wolke-Visualisierung dargestellt. Die abgebildeten Tokens sind hierbei keinen spezifischen Entitätstypen zugeordnet.

Die Visualisierungsart Wort-Wolke aus Abbildung 29 zeigt, unabhängig vom Thema, die am häufigsten genannten Tokens, die als Entitätstyp Sentiment extrahiert wurden. Der Entitätstyp Sentiment beschreibt die allgemeine Form von Tokens aus denen andere spezifische Sentiments extrahiert wurden. Da, wie oben festgestellt, der größte Teil der Kunden-Kommentare das Thema Gewinnspiel beinhalten, kann davon ausgegangen werden, dass die abgebildeten Tokens des Entitätstyps Sentiment wahrscheinlich über Gewinnspiele handeln. Demnach sind Kunden von Produkten der Tradefood GmbH überzeugt und beteiligen sich gerne an Gewinnspiel-Aktionen.

Anhand der generierten (virtuellen) Tabellen, im Rahmen der Anwendung des LDAs und der Entity und Fact Extraction, konnte ein besseres Verständnis über die Beiträge und Kunden-Kommentare der Facebook-Seite von Tradefood GmbH gewonnen werden. Die Analyse ermöglichte einige Hypothesen aufzustellen, die seitens Tradefood GmbH geprüft werden können. Werden die Hypothesen bewiesen, so kann das resultierende Wissen taktisch eingesetzt werden, um strategische Ziele zu verfolgen.

---

[64] Die Fact Extraction beschreibt im Rahmen dieses Projektes die Anwendung der Konfigurationsdatei, EXTRACTION_CORE-_VOICEOFCUSTOMER, welche die Ausführung der Sentiment-Analyse bezweckte.

6 Resümee

6.1 Zusammenfassung der Erkenntnisse

Die Aufgabenstellung der vorliegenden Arbeit war die Anwendung der In-Memory-Datenbank, HANA, als Basistechnologie im Rahmen eines konkreten Anwendungsfalls für Text-Mining, um eine Analyse von Social-Media-Beiträgen eines Unternehmens durchzuführen. Die Zielsetzung setzt sich aus einem primären und sekundären Ziel zusammen.

Das primäre Ziel behandelt die Würdigung der Bestandteile des Sprachmoduls von HANA, zur Durchführung von Text-Mining. Dabei soll der Mehrwert der HANA-Technologie evaluiert und eine Aussage getroffen werden, inwiefern das Sprachmodul der HANA, die Prozesse des zugrunde liegenden Text-Mining-Anwendungsfalls abdecken kann.

Python konnte als eine Alternative verwendet werden, welche die Daten über die Graph API extrahierte und dann in die HANA lud. Jedoch konnte zur Erfüllung des primären Ziels aufgrund dessen nicht beigetragen werden.

Die Aufbereitung der Facebook-Beiträge wurde auf der HANA nicht effektiv und effizient ausgeführt. Die Prozesse zur Normalisierung und Wortstammbildung von Tokens funktionierten in der Grundgesamtheit nach der Dokumentation von SAP richtig. Jedoch hinderte die Normalisierungsregel der unterschiedlichen Groß- und Kleinschreibung dem Ziel, die Anzahl der eindeutigen Tokens zu verringern. Die Wortstammbildung von Tokens muss zudem verbessert werden, da sie z.T. gewöhnliche Tokens nicht auf den Wortstamm zurückführen konnte. Eine fehlerhafte Implementierung konnte festgestellt werden, da einige Tokens auf den Wortstamm umgewandelt wurden und einige nicht, obwohl sie identisch vorlagen.

Für die Entfernung von Stopwords mussten eigene Entwicklungen durchgeführt werden, da der von SAP implementierte Ansatz zur Löschung von Stopwords den Aussagegehalt der Tokens minderte und Emoticons löschte.

Die Ausgabetabelle, welche eine Term-Document-Matrix abbildete war nicht, der Theorie entsprechend, korrekt abgebildet. Zudem gab es keine explizite Schnittstelle, die Term-Document-Matrix für Algorithmen der Predicitive Analysis Library anzuwenden. Effiziente Algorithmen der PAL, wie der K-Mean, können somit keine Anwendung finden.

Es konnte darauf geschlossen werden, dass beim Aufruf des Latent Dirichlet Allocation-Algorithmus die Generierung der Term-Document-Matrix implizit angestoßen wurde, woraufhin der Algorithmus auf Grundlage dessen operierte.

Die Ausführung des LDA-Clustering-Algorithmus erbrachte solide Resultate. Die Ausgabetabellen konnten für nachfolgende Analysen weiterverwendet werden. Einige Ausgabetabellen mussten jedoch vorher manuell zusammengeführt werden, damit die Daten visualisiert und besser verstanden werden können.

Die Ausführung der Entity und Fact Extraction konnten valide Zuordnungen von Tokens und Entitätstypen realisieren. Allerdings wurde festgestellt, dass die Ausführung von Entity und Fact Extraction aufgrund bestimmter Emoticons und/oder Symbolen beeinträchtigt wurde. Es musste erstmal eine PL-SQL Prozedur geschrieben werden, welche diejenigen Emoticons und/oder Symbole löschte, die die Störung wahrscheinlich auslösten.

Erwartungsgemäß konnte Lumira Discovery die Daten aus der HANA ohne weitere Probleme lesen und ermöglichte adäquate Visualisierungen zu erstellen.

Das sekundäre Ziel befasst sich mit der Evaluierung der gewonnenen Erkenntnisse über die analysierten Social-Media-Beiträge des angeführten Unternehmens. Demnach konnten aus den Facebook-Beiträgen die relevanten Themen, Gewinnspiel, Spenden für Kinder, Eis, Rezept-Ideen und weitere Themen mittels des LDA-Algorithmus entdeckt werden.

Die Visualisierung der Analyseergebnisse konnten sinnvoll interpretiert werden, woraufhin einige Hypothesen aufgestellt wurden. Interpretationen wiesen darauf hin, dass Gewinnspiele sehr beliebt sind und eine hohe Resonanz auf der Facebook-Seite des Unternehmens bewirken. Im weiteren Verlauf wurden Hypothesen aufgestellt, die Ansätze zur Erhöhung des Absatzes von Eis-Produkten oder von Spendenbeiträgen für Kinder behandelten.

6.2 Bedeutung und Ausblick dieser Arbeit

Nach einer drei Monatigen Auseinandersetzung mit Text-Mining-Funktionalitäten auf Basis der HANA, konnte ein genaueres Bild darüber geschaffen werden, in welchem Umfang die HANA einen Mehrwert für das Analysieren von unstrukturierten Daten erbringen kann.

In dieser Arbeit wurde nachgewiesen, dass aktuell die Text-Mining-Kapazitäten der HANA nicht vollkommen ausgereift sind und zusätzlich die Entwicklung weiterer Programmcodes erfordern, um ein zufriedenstellendes Ergebnis zu generieren.

Die im Rahmen der Arbeit angewendeten Verfahren sind nicht neuartig und wurden im Laufe der Zeit hinsichtlich der Effektivität und Effizienz weiterentwickelt. Die Erwartungen an die Text-Mining-Kompetenzen der HANA waren bemessen an dem aktuellen Stand der Entwicklungen im Bereich des Text-Minings. SAP ist ein kompetenter Softwarehersteller, welcher allerdings die Text-Mining-Komponente der HANA nicht an den Status quo angepasst hat und somit aktuell nicht wettbewerbsfähig ist.

Demzufolge müssen Unternehmen, die einen Mehrwert mittels Text-Mining erzeugen wollen, weiterhin andere Lösungen einsetzen. Es kann als Alternative in Erwägung gezogen werden externen Text-Mining-Lösungen in Zusammenhang mit der HANA zu verwenden, um schnelle Datenzugriffe zu erlauben und in nahezu Echtzeit Analysen aufzurufen. Die HANA bietet viele Adapter an, welche Quellsystemen erlauben sich mit der HANA zu verbinden.

Die Zugriffsgeschwindigkeiten der HANA sind sehr hoch und würden in Zusammenhang mit ausgereiften Text-Mining-Funktionalitäten die Voraussetzungen schaffen Text-Mining in Echtzeit auszuführen, ohne ein externe Lösung einzusetzen. In Anbetracht des signifikant schnellen Datenwachstums werden In-Memory-Datenbanken in Zusammenhang mit effizienten Algorithmen zur automatisierten Datenverarbeitung immer Bedeutsamer und wertvoller für Unternehmen. SAP ist mit der HANA im Markt gut aufgestellt. Eine zusätzliche kompetente Text-Mining-Umgebung kann für Entscheidungsträger ein weiteres Kriterium für den Einsatz der HANA darstellen.

6.3 Offene Fragen

Die HANA-Architektur bietet eine integrierte Umgebung für die Programmiersprache R an. R ist neben Python eine beliebte und ausgereifte Open-Source-Plattform mit einer großen Community. Es können Anwendungsfälle eruiert werden, in denen Eigenschaften der HANA zusammen mit R einen besonderen Mehrwert erbringen können. Dabei soll der Fokus auf der Fragestellung liegen, in welchem Ausmaß die R-Umgebung in der HANA integriert ist, wo die Grenzen liegen und welchen Mehrwert die Nutzung von R auf der HANA in Kontext eines Anwendungsfalls erbringen kann.

Zudem können Datenbereitstellungsadapter der SAP HANA Smart Data Integration auf Robustheit getestet werden. Tiefgründige Analysen im Rahmen von Big Data-Projekten werden immer beliebter und bedingen die Integration von verschiedenen heterogenen Quellsystemen. Unter Berücksichtigung von vorab definierten Kriterien könnten die Adapter getestet und Informationen über die Eignung der Adapter erarbeitet werden.

Als Gegenstück zum Text-Mining kann Data-Mininig auf der HANA evaluiert werden. Die PAL bietet eine große Menge an Algorithmen an, die in einer abgegrenzten Aufgabenstellung eingesetzt werden können. Dabei soll analog zu dieser Arbeit an der, in Kapitel 2.3.1 vorgestellten, Vorgehensweise zur Durchführung eines Data-Mining-Projektes, entlanggearbeitet werden.

Der professionelle Umgang mit vielfältigen, extensiven und immer schneller wachsenden Daten wird immer bedeutungsvoller. Eine konzentrierte Auseinandersetzung mit Technologien, wie der HANA und Datenanalyse-Algorithmen, sind unausweichlich. Resultierend daraus sind Forschungen in diesem Gebiet von großer Wichtigkeit.

Literaturverzeichnis

Alpar, Paul; Niedereichholz, Joachim (2000): Data Mining im praktischen Einsatz. Verfahren und Anwendungsfälle für Marketing, Vertrieb, Controlling und Kundenunterstützung (Business Computing). Wiesbaden: Vieweg+Teubner Verlag.

Alterauge, Markus (o.J.): Einsatzszenarien für SAP HANA. (Hrsg.): mayato GmbH. Online verfügbar unter https://www.mayato.com/wp-content/uploads/2016/04/mayato_WhitePaper_2013_Einsatzszenarien-SAP-HANA_DE-1.pdf, zuletzt geprüft am 14.08.2018.

Baesens, B.; van Vlasselaer, V.; Verbeke, W. (2015): Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques. A Guide to Data Science for Fraud Detection: Wiley.

Bansal, Shivam (2016): Beginners Guide to Topic Modeling in Python. Online verfügbar unter https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/, zuletzt geprüft am 29.08.2018.

Bastos, Marco; Walker, Shawn T. (2018): Facebook's data lockdown is a disaster for academic researchers. (Hrsg.): The Conversation. Online verfügbar unter http://theconversation.com/facebooks-data-lockdown-is-a-disaster-for-academic-researchers-94533, zuletzt geprüft am 20.09.2018.

Begerow, Markus (o.J.): SAP HANA. (Hrsg.): Datenbanken verstehen. Online verfügbar unter http://www.datenbanken-verstehen.de/lexikon/sap-hana/, zuletzt geprüft am 12.08.2018.

Bibliographisches Institut GmbH (o.J.): Die häufigsten Wörter in deutschsprachigen Texten. Online verfügbar unter https://www.duden.de/sprachwissen/sprachratgeber/Die-haufigsten-Worter-deutschsprachigen-Texten, zuletzt geprüft am 22.08.2018.

Bicu, Lulia (o.J.): Text search and analysis in SAP HANA. (Hrsg.): Today Software Magazine. Online verfügbar unter https://www.todaysoftmag.com/article/2144/text-search-and-analysis-in-sap-hana, zuletzt geprüft am 01.09.2018.

Bitkom e. V. (2016): Was muss ich wissen zur EU-Datenschutz Grundverordnung? (Hrsg.): Bitkom e. V. und Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. Online verfügbar unter https://www.bitkom.org/Presse/Anhaenge-an-PIs/2016/160909-EU-DS-GVO-FAQ-03.pdf, zuletzt geprüft am 23.08.2018.

Blei, David M. (2011): Introduction to Probabilistic Topic Models. Online verfügbar unter https://www.seas.harvard.edu/courses/cs281/papers/blei-2011.pdf, zuletzt geprüft am 26.08.2018.

Blei, David M.; Ng, Andrew Y.; Jordan, Michael I. (2003): Latent dirichlet allocation. In: The Journal of Machine Learning Research 3, S. 993–1022. Online verfügbar unter http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Chang, Chan-Chine; Chen, Ruey-Shun (2006): Using data mining technology to solve classification problems. In: The Electronic Library 24 (3), S. 307–321. Online verfügbar unter https://www.researchgate.net/profile/Ruey_shun_Chen/publication/220677141_Using_data_mining_technology_to_solve_classification_problems_A_case_study_of_campus_digital_library/links/0fcfd50c9d617a5d6a000000/Using-data-mining-technology-to-solve-classification-problems-A-case-study-of-campus-digital-library.pdf?origin=publication_detail, zuletzt geprüft am 17.08.2018.

Chen, Edwin (2011): What is a good explanation of Latent Dirichlet Allocation? Online verfügbar unter https://www.quora.com/What-is-a-good-explanation-of-Latent-Dirichlet-Allocation, zuletzt geprüft am 28.08.2018.

Deutscher Fachverlag GmbH (2018): Top 30 Lebensmittelhandel Deutschland 2018. Online verfügbar unter https://www.lebensmittelzeitung.net/handel/Ranking-Top-30-Lebensmittelhandel-Deutschland-2018-134606, zuletzt geprüft am 12.09.2018.

Deutsches Tiefkühlinstitut e.V. (o.J.): Absatzstatistik für Tiefkühlprodukte 2017. Berlin. Online verfügbar unter https://www.tiefkuehlkost.de/tk-fuer-handel/marktueberblick-1/marktdaten1/absatzstatistik2017, zuletzt geprüft am 12.09.2018.

Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996): From Data Mining to Knowledge Discovery in Databases. In: AI Magazine 17 (3), S. 37. Online verfügbar unter https://www.aaai.org/ojs/index.php/aimagazine/article/download/1230/1131.

Gahm, Hermann; Schneider, Thorsten; Swanepoel, Christiaan; Westenberger, Eric (2016): ABAP-Entwicklung mit SAP HANA (SAP press). 2., aktualisierte und erweiterte Auflage. Bonn: Rheinwerk Verlag GmbH.

Gartner Inc. (o.J.): Gartner IT Glossary. Big Data. Online verfügbar unter https://www.gartner.com/it-glossary/big-data, zuletzt geprüft am 10.08.2018.

Geierhos, Michaela (2018): Text Mining. Unter Mitarbeit von Frederik S. Bäumer. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik – Online-Lexikon. Online verfügbar unter http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/technologien-methoden/KI-und-Softcomputing/text-mining, zuletzt geprüft am 20.08.2018.

Grandpierre, Marcel; Buss, Georg; Esser, Ralf (2013): In-Memory Computing technology. The holy grail of analytics? (Hrsg.): Deloitte & Touche GmbH Wirtschaftsprüfungsgesellschaft. Online verfügbar unter https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-

telecommunications/TMT_Studie_In_Memory_Computing.pdf, zuletzt geprüft am 10.08.2018.

Han, Jiawei; Kamber, Micheline (2010): Data mining. Concepts and techniques (The Morgan Kaufmann series in data management systems). 2., Nachdr. Amsterdam: Elsevier/Morgan Kaufmann.

Hennig, Alexander; Schneider, Willy (2018): Handel. Definition. (Hrsg.): Gabler Wirtschaftslexikon. Online verfügbar unter https://wirtschaftslexikon.gabler.de/definition/handel-35491/version-258972, zuletzt geprüft am 12.09.2018.

Heyer, Gerhard; Quasthoff, Uwe; Wittig, Thomas (2008): Text mining: Wissensrohstoff Text. Konzepte, Algorithmen, Ergebnisse (Informatik). Korrigierter Nachdr. Herdecke: W3L-Verlag. Online verfügbar unter http://deposit.ddb.de/cgi-bin/dokserv?id=2783785&prov=M&dok_var=1&dok_ext=htm.

Hippner, Hajo; Rentzmann, René (2006): Text Mining. In: Informatik Spektrum 29 (4), S. 287–290.

Kemper, Hans-Georg; Baars, Henning; Mehanna, Walid (2010): Business Intelligence - Grundlagen und praktische Anwendungen. Eine Einführung in die IT-basierte Managementunterstützung (Studium). 3., überarbeitete und erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden.

Kimball, Ralph; Ross, Margy (2013): The data warehouse toolkit. The definitive guide to dimensional modeling. 3. Aufl. Indianapolis: Wiley.

Kotu, Vijay (2015): Predictive analytics and data mining. Concepts and practice with RapidMiner. Waltham, MA: Morgan Kaufmann.

Laudon, K. C.; Laudon, J. P.; Schoder, D. (2010): Wirtschaftsinformatik. Eine Einführung: Pearson Deutschland.

Lettier, David (2018): Your Easy Guide to Latent Dirichlet Allocation. (Hrsg.): Medium. Online verfügbar unter https://medium.com/@lettier/how-does-lda-work-ill-explain-using-emoji-108abf40fa7d, zuletzt geprüft am 30.08.2018.

Litzel, Nico (2016): Was ist Data Mining? (Hrsg.): BigData-Insider. Online verfügbar unter https://www.bigdata-insider.de/was-ist-data-mining-a-593421/, zuletzt geprüft am 14.08.2018.

Litzel, Nico (2017): Was ist SAP HANA? (Hrsg.): BigData-Insider. Online verfügbar unter https://www.bigdata-insider.de/was-ist-sap-hana-a-617851/, zuletzt geprüft am 10.08.2018.

Liu, Ethen (2015): Latent Dirichlet Allocation Using Gibbs Sampling. Online verfügbar unter https://ethen8181.github.io/machine-learning/clustering_old/topic_model/LDA.html, zuletzt geprüft am 29.08.2018.

Martins, Filipe; Kobylinska, Anna (2018): Die Kaffeemaschine in der IT. Was ist eine Appliance? (Hrsg.): Vogel IT-Medien GmbH. Online verfügbar unter https://www.datacenter-insider.de/was-ist-eine-appliance-a-672239/, zuletzt geprüft am 12.08.2018.

Miner, G.; Elder, J.; Fast, A.; Hill, T.; Nisbet, R.; Delen, D. (2012): Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications: Elsevier Science.

Piatetsky-Shapiro, Gregory (2007): Data mining and knowledge discovery 1996 to 2005. Overcoming the hype and moving from “university” to “business” and “analytics”. In: Data Mining and Knowledge Discovery 15 (1), S. 99–105.

Python Software Foundation (2018): General Python FAQ. Online verfügbar unter https://docs.python.org/3/faq/general.html#what-is-python-good-for, zuletzt geprüft am 23.08.2018.

Rajpal, Esha (2018): SAP HANA Full Text Index. (Hrsg.): SAP SE. Online verfügbar unter https://blogs.sap.com/2018/02/15/sap-hana-full-text-index/, zuletzt geprüft am 01.09.2018.

Reichert, Ramón (Hrsg.) (2014): Big data. Analysen zum digitalen Wandel von Wissen, Macht und Ökonomie. Bielefeld: transcript (Digitale Gesellschaft).

Reinsel, David; Gantz, John; Rydning, John (2017): Data Age 2025. The Evolution of Data to Life-Critical. Don’t Focus on Big Data; Focus on the Data That’s Big. (Hrsg.): International Data Corporation (IDC). Online verfügbar unter https://www.seagate.com/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf, zuletzt geprüft am 22.09.2018.

Resch, René (2018): RAM: Arbeitsspeicher-Preise sollen weiter steigen. (Hrsg.): PC-WELT. Online verfügbar unter https://www.pcwelt.de/a/ram-arbeitsspeicher-preise-sollen-weiter-steigen,3450051, zuletzt geprüft am 11.08.2018.

SAP SE (2018a): SAP HANA Developer Guide 2.0 SPS 03. For SAP HANA Studio. Online verfügbar unter https://help.sap.com/doc/fbb802faa34440b39a5b6e3814c6d3b5/2.0.03/en-US/SAP_HANA_Developer_Guide_for_SAP_HANA_Studio_en.pdf, zuletzt geprüft am 12.08.2018.

SAP SE (2018b): SAP HANA Master Guide 2.0 SPS03. Online verfügbar unter https://help.sap.com/doc/e95f6750b0fd10148ea5c6be75016694/2.0.03/en-US/SAP_HANA_Master_Guide_en.pdf, zuletzt geprüft am 13.08.2018.

SAP SE (2018c): SAP HANA Modeling Guide 2.0 SPS 03. For SAP HANA Studio. Online verfügbar unter https://help.sap.com/doc/fb8f7a9f7860468b84a07eab0a7d0a98/2.0.03/en-US/SAP_HANA_Modeling_Guide_for_SAP_HANA_Studio_en.pdf, zuletzt geprüft am 12.08.2018.

SAP SE (2018d): SAP HANA Predictive Analysis Library (PAL) 2.0 SPS 03. Online verfügbar unter https://help.sap.com/doc/86fb8d26952748debc8d08db756e6c1f/2.0.03/en-US/SAP_HANA_Predictive_Analysis_Library_PAL_en.pdf, zuletzt geprüft am 13.08.2018.

SAP SE (2018e): SAP HANA R Integration Guide 2.0 SPS 03. Online verfügbar unter https://help.sap.com/doc/6f2ff4c50f7e4e4d90b93aa33652d063/2.0.03/en-US/SAP_HANA_R_Integration_Guide_en.pdf, zuletzt geprüft am 13.08.2018.

SAP SE (2018f): SAP HANA Text Analysis Extraction Customization Guide 2.0 SPS 01. Online verfügbar unter https://help.sap.com/doc/c297277d9d884868a7f6eeb3e5b5c4e7/2.0.03/en-US/SAP_HANA_Text_Analysis_Extraction_Customization_Guide_en.pdf, zuletzt geprüft am 03.09.2018.

SAP SE (2018g): SAP HANA Text Analysis Language Reference Guide 2.0 SPS 01. Online verfügbar unter https://help.sap.com/doc/2e76b520f80e4fb0b4c91a756f5f51f7/2.0.03/en-US/SAP_HANA_Text_Analysis_Language_Reference_Guide_en.pdf, zuletzt geprüft am 02.09.2018.

SAP SE (2018h): SAP HANA Text Mining Developer Guide 2.0 SPS 03. Online verfügbar unter https://help.sap.com/doc/c370f830e80541e882247c07862a825d/2.0.03/en-US/SAP_HANA_Text_Mining_Developer_Guide_en.pdf, zuletzt geprüft am 21.09.2018.

SAS Institute Inc. (o.J.): Big Data. What it is and why it matters. Online verfügbar unter https://www.sas.com/en_us/insights/big-data/what-is-big-data.html, zuletzt geprüft am 10.08.2018.

Schmitz, Andreas (2015): Was ist eigentlich SAP HANA? (Hrsg.): SAP News Center. SAP SE. Online verfügbar unter https://news.sap.com/germany/2015/09/ist-eigentlich-sap-hana/, zuletzt geprüft am 10.08.2018.

Silvia, Penny; Frye, Rob; Berg, Bjarne (2017): SAP HANA. Die neue Einführung (Rheinwerk Publishing). 3., aktualisierte und erweiterte Auflage. Bonn: Rheinwerk Verlag GmbH.

Sonnenschein, David (2013): A Look Under the Hood of SAP HANA. (Hrsg.): SAPinsider. Online verfügbar unter https://sapinsider.wispubs.com/Assets/Articles/2013/April/A-Look-Under-The-Hood-Of-SAP-HANA, zuletzt geprüft am 13.08.2018.

Statista GmbH (2018a): Höhe der Geldspenden in Deutschland nach Monaten im Jahr 2017 (in Millionen Euro). Online verfügbar unter https://de.statista.com/statistik/daten/studie/37049/umfrage/spenden-in-deutschland-nach-monaten/, zuletzt geprüft am 22.09.2018.

Statista GmbH (2018b): Konsumausgaben der privaten Haushalte in Deutschland für Nahrungsmittel in den Jahren 1991 bis 2017 (in Milliarden Euro). Online verfügbar unter https://de.statista.com/statistik/daten/studie/296815/umfrage/konsumausgaben-in-deutschland-fuer-nahrungsmittel/, zuletzt geprüft am 12.09.2018.

Statista GmbH (2018c): Number of daily active Facebook users worldwide as of 2nd quarter 2018 (in millions). Online verfügbar unter https://www.statista.com/statistics/346167/facebook-global-dau/, zuletzt geprüft am 22.09.2018.

Statistische Bundesamt (2017): 54 % der Konsumausgaben entfielen 2016 auf Wohnen, Ernährung und Bekleidung. Online verfügbar unter https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2017/12/PD17_463_631.html, zuletzt geprüft am 12.09.2018.

Tan, Ah-Hwee (1999): Text Mining:The state of the art and the challenges. Online verfügbar unter http://www3.ntu.edu.sg/home/ASAHTan/Papers/tm_pakdd99.pdf, zuletzt geprüft am 19.08.2018.

The Economist (2017): The world’s most valuable resource is no longer oil, but data. The data economy demands a new approach to antitrust rules. Online verfügbar unter https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data, zuletzt geprüft am 22.09.2018.

tutorialspoint (o.J.): SAP HANA. Core Architecture. Online verfügbar unter https://www.tutorialspoint.com/sap_hana/sap_hana_core_architecture.htm, zuletzt geprüft am 12.08.2018.

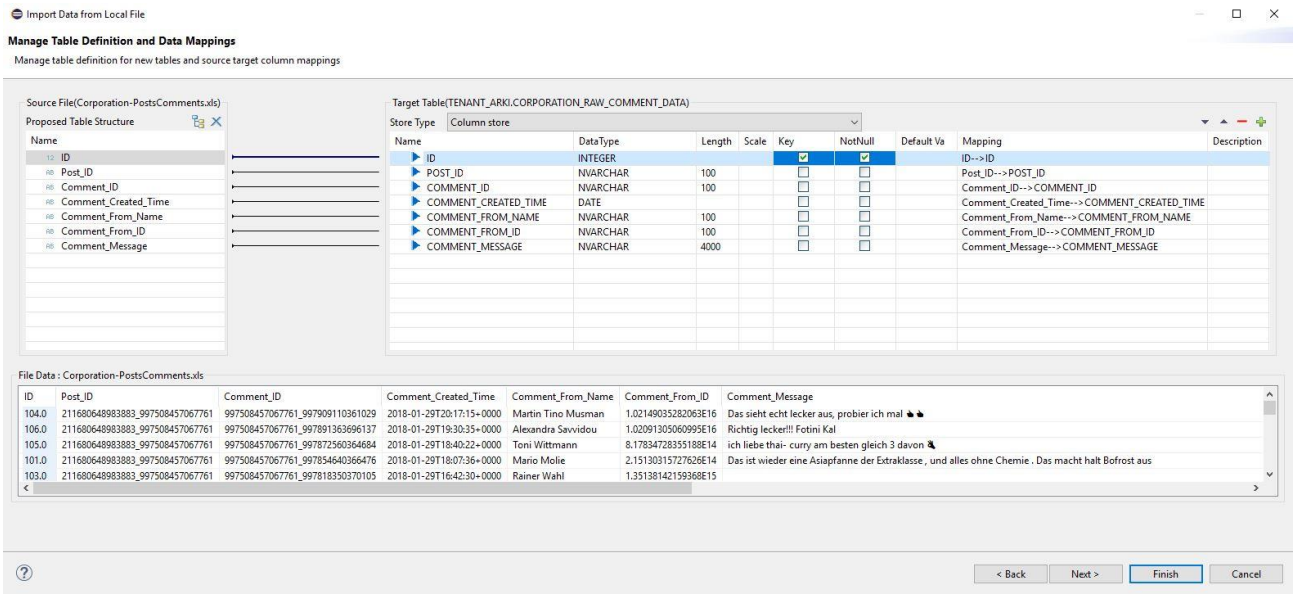
Vijayarani, S.; Ilamathi, J.; Nithya (2015): Preprocessing Techniques for Text Mining - An Overview. (Hrsg.): International Journal of Computer Science & Communication Networks. Online verfügbar unter https://pdfs.semanticscholar.org/1fa1/1c4de09b86a05062127c68a7662e3ba53251.pdf, zuletzt geprüft am 20.08.2018.

Wiehr, Hartmut (2017): In-Memory-Plattform. Wie sich SAP mit HANA neu erfunden hat. (Hrsg.): Neue Mediengesellschaft Ulm mbH. Online verfügbar unter https://www.com-magazin.de/praxis/business-it/sap-hana-neu-erfunden-1208169.html, zuletzt geprüft am 10.08.2018.

Anhang

```
<?xml version="1.0" encoding="utf-8"?>
<dictionary xmlns="http://www.sap.com/ta/4.0">
        <entity_category name="StrongPositiveSentiment">
                <entity_name standard_form="verdammt geil"/>
        </entity_category>
</dictionary>
```

Anhang 1: Wortstammbildung mit einem benutzerdefiniertem Wörterbuch im XML-Format



Anhang 2: Import einer CSV-Datei in die HANA

```
SET SCHEMA TENANT_ARKI;

DELETE FROM "CORPORATION_RAW_POST_DATA"  WHERE POST_MESSAGE ='None' OR POST_CREATED_TIME IS
NULL;

CREATE FULLTEXT INDEX MYINDEX2 ON "CORPORATION_RAW_POST_DATA" ("POST_MESSAGE")
     FAST PREPROCESS OFF
     MIME TYPE 'text/plain'
     ASYNC
     TEXT ANALYSIS ON
     CONFIGURATION 'LINGANALYSIS_STEMS'
     LANGUAGE DETECTION ('DE')
     TOKEN SEPARATORS '\/;,.:-_()[]<>!?*@+{}="&'
     TEXT MINING ON
     TEXT MINING CONFIGURATION 'DEFAULT';
```

Anhang 3: Löschen von leeren Zellen und Erzeugung des Full-Text-Indexes der Tabelle CORPORATION_RAW_POST_DATA

```
<?xml version="1.0" encoding="utf-8" ?>
<tasdk-configuration xmlns="http://www.sap.com/ta/config/4.0">
  <configuration name="SAP.TextAnalysis.AnalysisModel.AggregateAnalyzer.Aggregator">
    <!-- Angeben der Reihenfolge der auszuführenden Textanalyseschritte. -->
    <property name="Analyzers" type="string-list">
     <string-list-
     value>SAP.TextAnalysis.DocumentAnalysis.FormatConversion.FormatConversionAnalyzer.FC</string-list-
     value>
     <string-list-value>SAP.TextAnalysis.DocumentAnalysis.StructureAnalysis.StructureAnalyzer.SA</string-
     list-value>
     <string-list-value>SAP.TextAnalysis.DocumentAnalysis.LinguisticAnalysis.LinguisticAnalyzer.LX</string-
     list-value>
    </property>
  </configuration>

  <!-- Obligatorische Sektion  -->
  <configuration name="CommonSettings" />
  <!-- Obligatorische Sektion  -->
  <configuration name="SAP.TextAnalysis.DocumentAnalysis.FormatConversion.FormatConversionAnalyzer.FC"
based-on="CommonSettings" />

  <!-- Obligatorische Sektion für das Erkennen der Sprachen, welches in dieser Arbeit nicht relevant ist-->
  <configuration name="SAP.TextAnalysis.DocumentAnalysis.StructureAnalysis.StructureAnalyzer.SA" based-
on="CommonSettings">

    <property name="MinimumInputLength" type="integer">
      <integer-value>30</integer-value>
    </property>

    <property name="EvaluationSampleSize" type="integer">
        <integer-value>300</integer-value>
    </property>

    <property name="MinimumConfidence" type="integer">
        <integer-value>50</integer-value>
    </property>

  </configuration>

  <!– Obligatorische Sektion  -->
  <configuration name="SAP.TextAnalysis.DocumentAnalysis.LinguisticAnalysis.LinguisticAnalyzer.LX" based-
on="CommonSettings">

    <!-- Bestimmen der Grundformen für jeden Token -->
    <property name="GetTokenStem" type="boolean">
      <boolean-value>true</boolean-value>
    </property>

    <!-- Erraten der Grundform von Tokens -->
    <property name="EnableStemGuesser" type="boolean">
      <boolean-value>false</boolean-value>
    </property>

    <!-- Identifizierung und Kennzeichnung der Wortklassen bzw. Wortarten von Wörtern -->
    <property name="GetTokenPartOfSpeech" type="boolean">
      <boolean-value>false</boolean-value>
    </property>

     <!-- Gibt an, ob die wahrscheinlichste Wortart in Fällen ausgewählt werden sollte, in denen die
     Wortart nicht eindeutig ist -->
    <property name="DisambiguatePartOfSpeech" type="boolean">
      <boolean-value>false</boolean-value>
    </property>

     <!-- Gibt an, ob die wahrscheinlichste Grundform der Wörter in Fällen ausgewählt werden sollte, in
     denen die Grundform nicht eindeutig ist -->
    <property name="DisambiguateStem" type="boolean">
      <boolean-value>true</boolean-value>
    </property>

    <!-- Nutzung eines benutzerspezifischen linguistischen Wörterbuches -->
    <property name="EnableCustomDictionaries" type="boolean">
      <boolean-value>true</boolean-value>
    </property>

    <!-- Mit der Option VariantString können alternative Implementierungen für die Extraktion und das
Stemming in verschiedenen Sprachen verwendet werden. -->
    <property name="VariantString" type="string">
      <string-value>expanded</string-value>
    </property>

  </configuration>

</tasdk-configuration>
```

Anhang 4: Konfigurationsdatei LINGANALYSIS_STEMS für die Ausführung der linguistischen Textanalyse

HXE@HXE (TENANT_ARKI) 172.16.20.33 90

SQL Result

```
SELECT * FROM "$TA_MYINDEX2" ORDER BY "POST_ID", "TA_COUNTER"ASC
```

	POST_ID	TA_RULE	TA_COU...	TA_TOKEN	TA_LANG...	TA_TYPE	TA_TYPE_EXP...	TA_NORMALIZED	TA_STEM	TA_PAR...	TA_SENT...	TA_CREATED_AT	TA_OFFSET	TA_PARENT
1	211680648983883_1000222023463071	LXP	1	Na	de	interjection	?	na	?	1	1	17.09.2018 10:57:0	0	?
2	211680648983883_1000222023463071	LXP	2	,	de	punctuation	?	?	?	1	1	17.09.2018 10:57:0	2	?
3	211680648983883_1000222023463071	LXP	3	auch	de	adverb	?	auch	?	1	1	17.09.2018 10:57:0	4	?
4	211680648983883_1000222023463071	LXP	4	kein	de	determiner	?	kein	?	1	1	17.09.2018 10:57:0	9	?
5	211680648983883_1000222023463071	LXP	5	Frühaufst...	de	noun	?	fruehaufsteher	Frühaufsteher	1	1	17.09.2018 10:57:0	14	?
6	211680648983883_1000222023463071	LXP	6	?	de	punctuation	?	?	?	1	1	17.09.2018 10:57:0	27	?
7	211680648983883_1000222023463071	LXP	7	Dann	de	adverb	?	dann	?	1	2	17.09.2018 10:57:0	29	?
8	211680648983883_1000222023463071	LXP	8	einfach	de	adjective	?	einfach	?	1	2	17.09.2018 10:57:0	34	?
9	211680648983883_1000222023463071	LXP	9	liegenble...	de	verb	?	liegenbleiben	liegen bleiben	1	2	17.09.2018 10:57:0	42	?
10	211680648983883_1000222023463071	LXP	10	und	de	conjunction	?	und	?	1	2	17.09.2018 10:57:0	56	?
11	211680648983883_1000222023463071	LXP	11	das	de	determiner	?	das	?	1	2	17.09.2018 10:57:0	60	?
12	211680648983883_1000222023463071	LXP	12	Frühstück	de	noun	?	fruehstueck	Frühstück	1	2	17.09.2018 10:57:0	64	?
13	211680648983883_1000222023463071	LXP	13	zum	de	preposition	?	zum	zu=das	1	2	17.09.2018 10:57:0	74	?
14	211680648983883_1000222023463071	LXP	14	Spätstück	de	noun	?	spaetstueck	spät#Stück	1	2	17.09.2018 10:57:0	78	?
15	211680648983883_1000222023463071	LXP	15	machen	de	verb	?	machen	?	1	2	17.09.2018 10:57:0	88	?

Anhang 5: Ausgabetabelle $TA_MYINDEX2 nach der Erzeugung des Full-Text-Indexes

```
CREATE COLUMN TABLE CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS AS (
    SELECT POST_ID,
    CASE WHEN TA_STEM <> '?' THEN TA_STEM
    WHEN  TA_NORMALIZED <> '?' THEN TA_NORMALIZED
    ELSE TA_TOKEN END
    POST_MESSAGE
    FROM "$TA_MYINDEX2");
```

Anhang 6: SQL-Anweisung zur Generierung einer Tabelle die alle normalisierten und auf den Wortstamm zurückgeführten Tokens enthält

File Tools Edit Navigate Search Project Run Window Help

HXE@HXE - Stemming and stopwordfiltering.sql | SQL_LDA_TENANT_ARKI_FINAL.sql | SQL_VOC_TOPIC48_TENANT_ARKI_FINAL.sql | *HXE@HXE -

HXE@HXE (TENANT_ARKI) 172.16.20.33 90

SQL Result

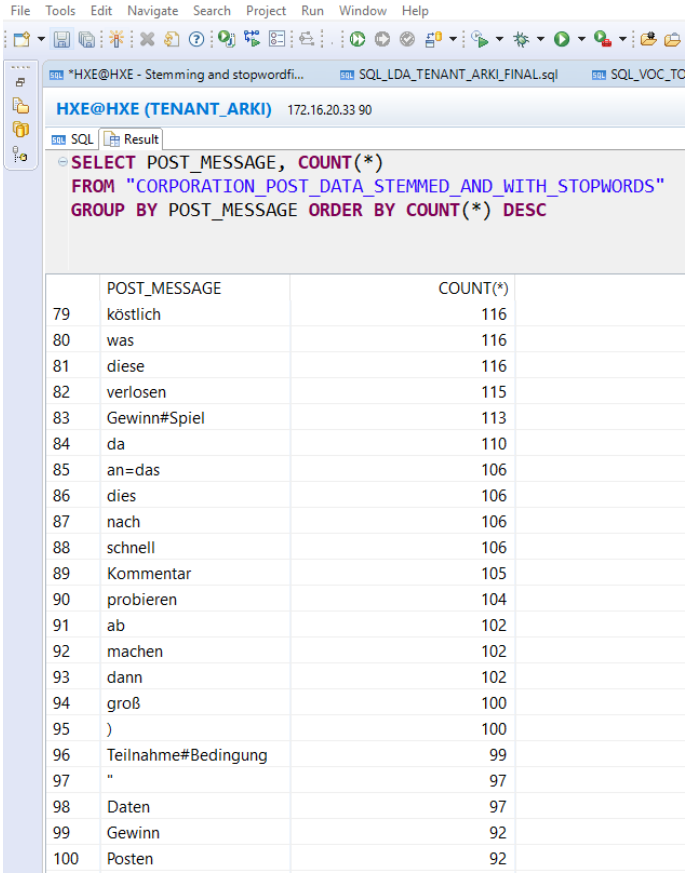
```
SELECT * FROM "CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS" ORDER BY POST_ID ASC
```

	POST_ID	POST_MESSAGE
1	211680648983883_1000222023463071	wunderbar
2	211680648983883_1000222023463071	liegen bleiben
3	211680648983883_1000222023463071	Wochenende
4	211680648983883_1000222023463071	Frühstück
5	211680648983883_1000222023463071	wünschen
6	211680648983883_1000222023463071	spät#Stück
7	211680648983883_1000222023463071	Frühaufsteher
8	211680648983883_1000222023463071	Appetit
9	211680648983883_1397951297129581	erfahren
10	211680648983883_1397951297129581	kaese
11	211680648983883_1397951297129581	grün
12	211680648983883_1397951297129581	Morgen
13	211680648983883_1397951297129581	vergessen
14	211680648983883_1397951297129581	überraschen
15	211680648983883_1397951297129581	Menü
16	211680648983883_1397951297129581	gelingen
17	211680648983883_1397951297129581	ricotta
18	211680648983883_1397951297129581	Februar
19	211680648983883_1397951297129581	Kirsch#Tomate
20	211680648983883_1397951297129581	tortelli
21	211680648983883_1397951297129581	jemand
22	211680648983883_1397951297129581	Valentin#Tag
23	211680648983883_1397951297129581	Nachtisch
24	211680648983883_1397951297129581	Empfehlung
25	211680648983883_1397951297129581	besonder

Anhang 7: Ausgabetabelle CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS

```
DELETE FROM "CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS"
WHERE POST_MESSAGE IN(
     SELECT POST_MESSAGE FROM (
          SELECT POST_MESSAGE, COUNT(*) FROM
          "CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS"
          GROUP BY POST_MESSAGE
          HAVING COUNT(*) > 280) );

DELETE FROM "CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS"
WHERE  POST_MESSAGE = (SELECT STOPWORD FROM "STOPWORD_DICT");
```

Anhang 8: Löschung von Stopwords aus der Tabelle CORPORATION_POST_DATA_STEMMED_AND_WITH_STOPWORDS

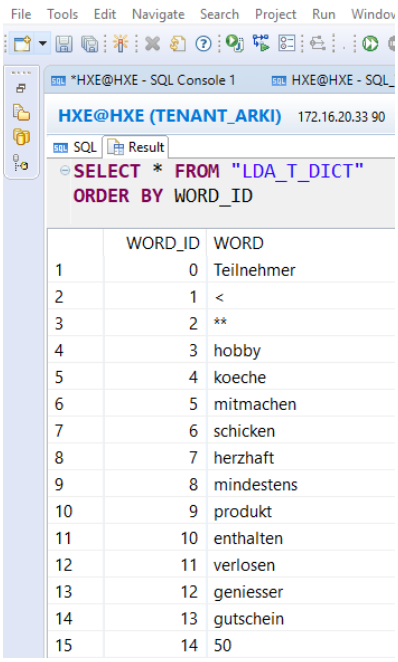


	POST_MESSAGE	COUNT(*)
79	köstlich	116
80	was	116
81	diese	116
82	verlosen	115
83	Gewinn#Spiel	113
84	da	110
85	an=das	106
86	dies	106
87	nach	106
88	schnell	106
89	Kommentar	105
90	probieren	104
91	ab	102
92	machen	102
93	dann	102
94	groß	100
95	)	100
96	Teilnahme#Bedingung	99
97	"	97
98	Daten	97
99	Gewinn	92
100	Posten	92

Anhang 9: Häufigkeiten der eindeutigen Tokens im Textkorpus

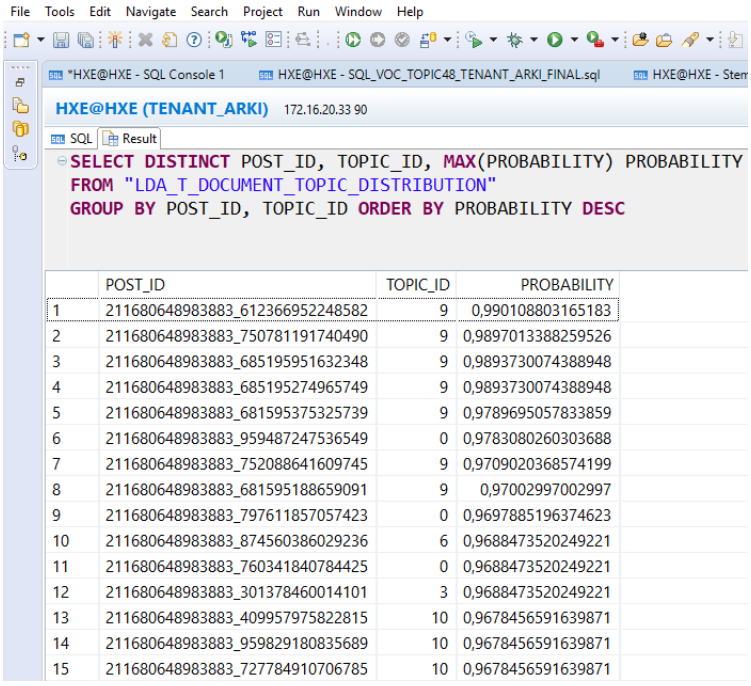
```
-- Erzeugung von benutzerdefinierten Tabellentypen
CREATE TYPE "T_PARAMS" AS TABLE ("NAME" VARCHAR(256), "INTARGS" INTEGER, "DOUBLEARGS" DOUBLE,
     "STRINGARGS" VARCHAR(1000));
CREATE TYPE "T_DICT" AS TABLE ("WORD_ID" INTEGER, "WORD" NVARCHAR(5000));
CREATE TYPE "T_TOPIC_WORD_DISTRIBUTION" AS TABLE ("TOPIC_ID" INTEGER, "WORD_ID" INTEGER,
     "PROBABILITY" DOUBLE);
CREATE TYPE "T_DOCUMENT_TOPIC_DISTRIBUTION" AS TABLE ("POST_ID" NVARCHAR(100), "TOPIC_ID" INTEGER,
     "PROBABILITY" DOUBLE);
CREATE TYPE "T_WORDTOPICASSIGNMENT" AS TABLE ("POST_ID" NVARCHAR(100), "WORD_ID" INTEGER,
     "TOPIC_ID" INTEGER);
CREATE TYPE "T_CV_PARAMETER" AS TABLE ("PARAM_NAME" NVARCHAR(256), "INT_VALUE" INTEGER,
     "DOUBLE_VALUE" DOUBLE, "STRING_VALUE" NVARCHAR(1000));
CREATE TYPE "T_TOPIC_TOP_WORDS" AS TABLE ("TOPIC_ID" INTEGER, "WORDS" NVARCHAR(5000));
CREATE TYPE "T_STATS" AS TABLE ("STAT_NAME" NVARCHAR(256), "STAT_VALUE" NVARCHAR(1000));

-- Erstellung der Eingabetabellen
CREATE VIEW "LDA_T_DATA" AS
     SELECT "POST_ID", "POST_MESSAGE" FROM "CORPORATION_POST_DATA_PREPROCESSED";
CREATE LOCAL TEMPORARY COLUMN TABLE "#LDA_T_PARAMS" LIKE "T_PARAMS";

-- Erstellung der Ausgabetabellen
CREATE COLUMN TABLE "LDA_T_DICT" LIKE "T_DICT";
CREATE COLUMN TABLE "LDA_T_TOPIC_WORD_DISTRIBUTION" LIKE "T_TOPIC_WORD_DISTRIBUTION";
CREATE COLUMN TABLE "LDA_T_DOCUMENT_TOPIC_DISTRIBUTION" LIKE "T_DOCUMENT_TOPIC_DISTRIBUTION";
CREATE COLUMN TABLE "LDA_T_CV_PARAMETER" LIKE "T_CV_PARAMETER";
CREATE COLUMN TABLE "LDA_T_TOPIC_TOP_WORDS" LIKE "T_TOPIC_TOP_WORDS";
CREATE COLUMN TABLE "LDA_T_WORDTOPICASSIGNMENT" LIKE "T_WORDTOPICASSIGNMENT";
CREATE COLUMN TABLE "LDA_T_STATS" LIKE "T_STATS";

-- Vergabe der Parameter des LDA-Algorithmus
INSERT INTO "#LDA_T_PARAMS" VALUES ('TOPICS', 11, NULL, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('BURNIN', 20, NULL, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('THIN', 20, NULL, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('ITERATION', 2000, NULL, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('SEED', 0, NULL, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('ALPHA', NULL, 0.1, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('BETA', NULL, 0.01, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('THRESHOLD_TOP_WORDS', NULL, 0.01, NULL);
INSERT INTO "#LDA_T_PARAMS" VALUES ('INIT', 1, NULL, NULL);

-- Ausführung des LDA-Algorithmus
CALL "_SYS_AFL"."PAL_LATENT_DIRICHLET_ALLOCATION"("LDA_T_DATA", "#LDA_T_PARAMS",
     "LDA_T_DOCUMENT_TOPIC_DISTRIBUTION", "LDA_T_WORDTOPICASSIGNMENT", "LDA_T_TOPIC_TOP_WORDS",
     "LDA_T_TOPIC_WORD_DISTRIBUTION","LDA_T_DICT", "LDA_T_STATS", "LDA_T_CV_PARAMETER") WITH OVERVIEW;
```

Anhang 10: SQL-Skript zur Ausführung des Latent Dirichlet Allocation-Algorithmus



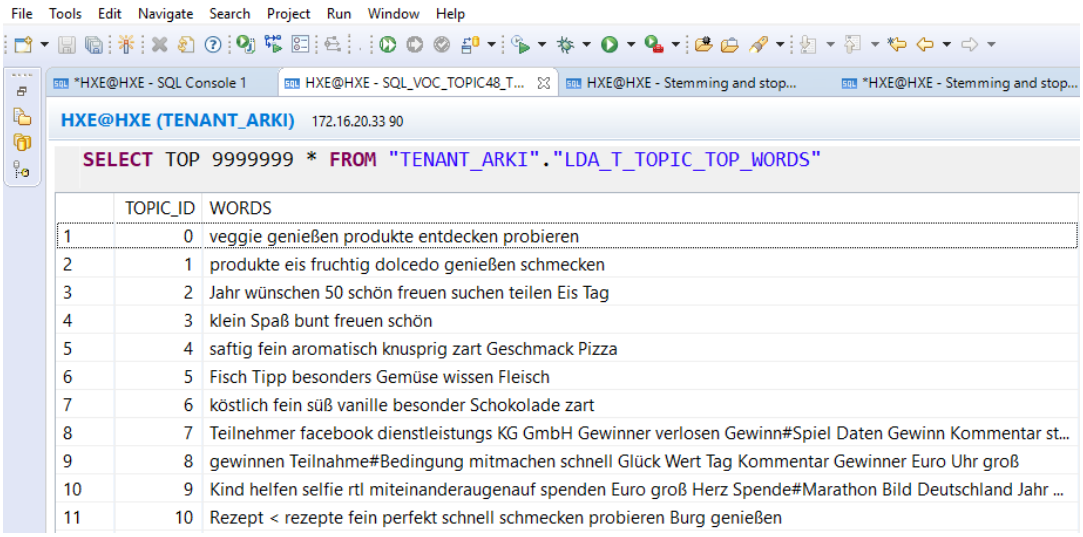
	WORD_ID	WORD
1	0	Teilnehmer
2	1	<
3	2	**
4	3	hobby
5	4	koeche
6	5	mitmachen
7	6	schicken
8	7	herzhaft
9	8	mindestens
10	9	produkt
11	10	enthalten
12	11	verlosen
13	12	geniesser
14	13	gutschein
15	14	50

Anhang 11: Ausgabetabelle LDA_T_DICT



	POST_ID	TOPIC_ID	PROBABILITY
1	211680648983883_612366952248582	9	0,990108803165183
2	211680648983883_750781191740490	9	0,9897013388259526
3	211680648983883_685195951632348	9	0,9893730074388948
4	211680648983883_685195274965749	9	0,9893730074388948
5	211680648983883_681595375325739	9	0,9789695057833859
6	211680648983883_959487247536549	0	0,9783080260303688
7	211680648983883_752088641609745	9	0,9709020368574199
8	211680648983883_681595188659091	9	0,97002997002997
9	211680648983883_797611857057423	0	0,9697885196374623
10	211680648983883_874560386029236	6	0,9688473520249221
11	211680648983883_760341840784425	0	0,9688473520249221
12	211680648983883_301378460014101	3	0,9688473520249221
13	211680648983883_409957975822815	10	0,9678456591639871
14	211680648983883_959829180835689	10	0,9678456591639871
15	211680648983883_727784910706785	10	0,9678456591639871

Anhang 12: Ausgabetabelle LDA_T_DOCUMENT_TOPIC_DISTRIBUTION

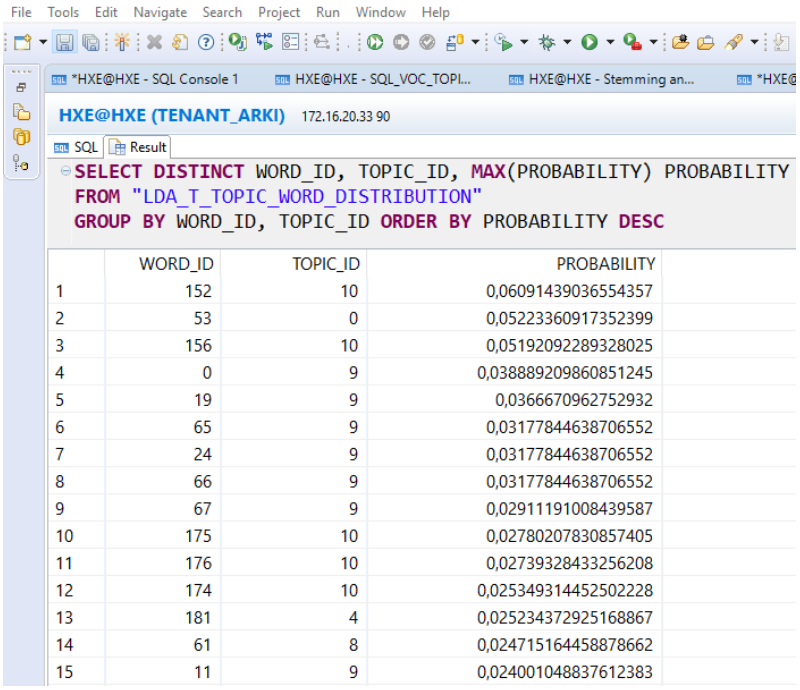


HXE@HXE (TENANT_ARKI) 172.16.20.33 90

```
SELECT TOP 9999999 * FROM "TENANT_ARKI"."LDA_T_TOPIC_TOP_WORDS"
```

	TOPIC_ID	WORDS
1	0	veggie genießen produkte entdecken probieren
2	1	produkte eis fruchtig dolcedo genießen schmecken
3	2	Jahr wünschen 50 schön freuen suchen teilen Eis Tag
4	3	klein Spaß bunt freuen schön
5	4	saftig fein aromatisch knusprig zart Geschmack Pizza
6	5	Fisch Tipp besonders Gemüse wissen Fleisch
7	6	köstlich fein süß vanille besonder Schokolade zart
8	7	Teilnehmer facebook dienstleistungs KG GmbH Gewinner verlosen Gewinn#Spiel Daten Gewinn Kommentar st...
9	8	gewinnen Teilnahme#Bedingung mitmachen schnell Glück Wert Tag Kommentar Gewinner Euro Uhr groß
10	9	Kind helfen selfie rtl miteinanderaugenauf spenden Euro groß Herz Spende#Marathon Bild Deutschland Jahr ...
11	10	Rezept < rezepte fein perfekt schnell schmecken probieren Burg genießen

Anhang 13: Ausgabetabelle LDA_T_TOPIC_TOP_WORDS

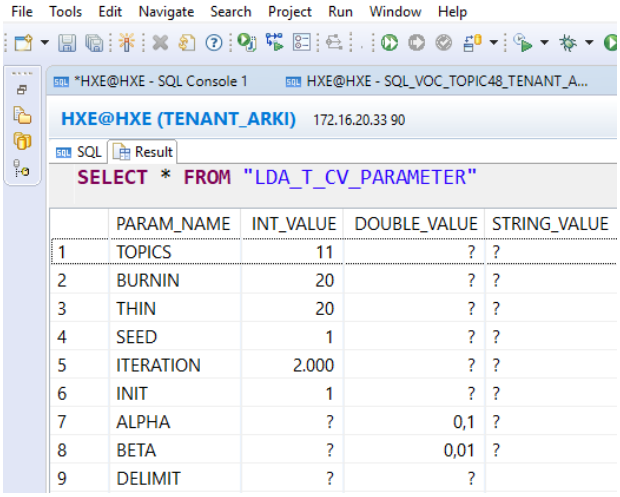


HXE@HXE (TENANT_ARKI) 172.16.20.33 90

```
SELECT DISTINCT WORD_ID, TOPIC_ID, MAX(PROBABILITY) PROBABILITY
FROM "LDA_T_TOPIC_WORD_DISTRIBUTION"
GROUP BY WORD_ID, TOPIC_ID ORDER BY PROBABILITY DESC
```

	WORD_ID	TOPIC_ID	PROBABILITY
1	152	10	0,06091439036554357
2	53	0	0,05223360917352399
3	156	10	0,05192092289328025
4	0	9	0,03888209860851245
5	19	9	0,0366670962752932
6	65	9	0,03177844638706552
7	24	9	0,03177844638706552
8	66	9	0,03177844638706552
9	67	9	0,02911191008439587
10	175	10	0,02780207830857405
11	176	10	0,02739328433256208
12	174	10	0,025349314452502228
13	181	4	0,025234372925168867
14	61	8	0,024715164458878662
15	11	9	0,024001048837612383

Anhang 14: Ausgabetabelle LDA_T_TOPIC_WORD_DISTRIBUTION



HXE@HXE (TENANT_ARKI) 172.16.20.33 90

```
SELECT * FROM "LDA_T_CV_PARAMETER"
```

	PARAM_NAME	INT_VALUE	DOUBLE_VALUE	STRING_VALUE
1	TOPICS	11	?	?
2	BURNIN	20	?	?
3	THIN	20	?	?
4	SEED	1	?	?
5	ITERATION	2.000	?	?
6	INIT	1	?	?
7	ALPHA	?	0,1	?
8	BETA	?	0,01	?
9	DELIMIT	?	?	

Anhang 15: Ausgabetabelle LDA_T_CV_PARAMETER

```
CREATE COLUMN TABLE "X_LDA_TOP_TOPIC_PER_DOCUMENT" AS (
     SELECT POST_ID, TOPIC_ID, PROBABILITY
     FROM "LDA_T_DOCUMENT_TOPIC_DISTRIBUTION"
     WHERE (POST_ID, PROBABILITY) IN (
          SELECT POST_ID, MAX(PROBABILITY) PROBABILITY
          FROM "LDA_T_DOCUMENT_TOPIC_DISTRIBUTION"
          GROUP BY POST_ID));
```

Anhang 16: SQL-Anweisung zur Erstellung der Tabelle X_LDA_TOP_TOPIC_PER_DOCUMENT

```
CREATE COLUMN TABLE "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS" AS (
     SELECT COMMENT_ID, A.POST_ID, CAST (TOPIC_ID AS NVARCHAR) TOPIC_ID,
     COMMENT_CREATED_TIME, COMMENT_MESSAGE
     FROM "X_LDA_TOP_TOPIC_PER_DOCUMENT" AS A
          INNER JOIN
          (SELECT * FROM CORPORATION_RAW_COMMENT_DATA) AS B
          ON A.POST_ID =B.POST_ID);
ALTER TABLE "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS" ADD (PROCESSED NVARCHAR(1));
ALTER TABLE "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS" ADD PRIMARY KEY (TOPIC_ID, COMMENT_ID);
```

Anhang 17: SQL-Anweisung zur Erstellung der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS

HXE@HXE (TENANT_ARKI) 172.16.20.33 90

```
SELECT TOP 9999999 * FROM "TENANT_ARKI"."X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS"
```

	COMMENT_ID	POST_ID	TOPIC_ID	COMMENT_CREATED_TIME	COMMENT_MESSAGE	PROCESSED
13354	737172073101402_739544386197504	211680648983883_737...	7	11.11.2016	die Eis-Schneemänner ❄	?
13355	737172073101402_739543846197558	211680648983883_737...	7	11.11.2016	Gesucht sind die Eis Schneemänner. Meine kleine Maus und ich würden uns riesig ü...	?
13356	737172073101402_739539496197993	211680648983883_737...	7	11.11.2016	Eis-Schneemänner :) LECKER !	?
13357	737172073101402_739539442864665	211680648983883_737...	7	11.11.2016	Eis-Schneemänner Super Aktion Meine Enkelkinder würden sich riesig freuen	?
13358	737172073101402_739538299531446	211680648983883_737...	7	11.11.2016	Mjammjamm ❄ Eis Schneemänner	?
13359	737172073101402_739535252865084	211680648983883_737...	7	11.11.2016	Oh das sieht aber lecker aus *_*! Die Eis-Schneemänner! :) Jasmine Ninnemann guc...	?
13360	737172073101402_739535196198423	211680648983883_737...	7	11.11.2016	die Eis-Schneemänner Franziska Kräupl mach doch auch mit dann klingelt dein Lief...	?
13361	737172073101402_739534742865135	211680648983883_737...	7	11.11.2016	Eis-Schneemänner! Lecker :-)	?
13362	737172073101402_739534512865158	211680648983883_737...	7	11.11.2016	Eis-Schneemänner haben sich versteckt :-)	?
13363	737172073101402_739533749531901	211680648983883_737...	7	11.11.2016	Das sind die leckeren Eis-Schneemänner und ich mache sehr gerne hier mit:-)*smile...	?
13364	737172073101402_739533122865297	211680648983883_737...	7	11.11.2016	Das sind doch Eis-Schneemänner	?
13365	737172073101402_739532739532002	211680648983883_737...	7	11.11.2016	Eis Schneemänner ❄❄❄	?
13366	737172073101402_739532476198695	211680648983883_737...	7	11.11.2016	Eis-Schneemänner	?
13367	737172073101402_739531356198807	211680648983883_737...	7	11.11.2016	das sind Eis-Schneemänner	?
13368	737172073101402_739530939532182	211680648983883_737...	7	11.11.2016	Eis Schneemänner Tolles gewinnspiel ❄	?
13369	737172073101402_739530806198862	211680648983883_737...	7	11.11.2016	Eis schneemänner. Die gibts zu den weihnachten immer bei muttern	?
13370	737172073101402_739528906199052	211680648983883_737...	7	11.11.2016	Super Gewinnspiel :D Das sind Eis Schneemänner :) Ich würde mich riesig über eine...	?

Anhang 18: Ausgabe der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS

```
CREATE FULLTEXT INDEX INDEX_COMMENT ON "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS"
("COMMENT_MESSAGE")
     FAST PREPROCESS OFF
     MIME TYPE 'text/plain'
     ASYNC
     TEXT ANALYSIS ON
     CONFIGURATION 'LINGANALYSIS_BASIC'
     LANGUAGE DETECTION ('DE')
     TOKEN SEPARATORS '\/;,.:-_()[]<>!?*@+{}="&';
```

Anhang 19: Ausführung des Full-Text-Index der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS

	UNICODE_ID	TA_TOKEN
1	8.226	•••
2	8.226	•
3	8.222	„
4	8.230	…
5	127.876	🎄
6	127.794	🌲
7	128.139	🎋
8	127.880	🎈
9	128.104	🎎
10	128.171	🎏
11	128.017	🎐
12	9.994	🎒
13	184	»
14	9.728	🎖
15	129.304	🎗
16	128.037	🎞
17	127.939	🎟
18	9.731	☃❄
19	128.064	👀
20	9.734	☆

Anhang 20: Tabelle X_EMOTICON_WITH_UNICODE

```
CREATE PROCEDURE  "TENANT_ARKI"."DELETE_EMOJI"
AS BEGIN
 DECLARE CURSOR CUR_COMMENT_TABLE FOR SELECT * FROM "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS"
 WHERE PROCESSED IS NULL;
 DECLARE CURSOR CUR_EMOJI_TABLE FOR SELECT * FROM "X_EMOTICON_WITH_UNICODE";
     FOR X AS CUR_COMMENT_TABLE DO
          FOR Y AS CUR_EMOJI_TABLE DO
               UPDATE "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS" SET COMMENT_MESSAGE =
               REPLACE (COMMENT_MESSAGE, Y.TA_TOKEN, '')
               WHERE COMMENT_ID = X.COMMENT_ID;
               UPDATE "X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS" SET PROCESSED = 'X'
               WHERE COMMENT_ID = X.COMMENT_ID;
          END FOR;
          COMMIT;
     END FOR;
END;
```

Anhang 21: PL-SQL-Prozedur DELETE_EMOJI

```
CREATE FULLTEXT INDEX INDEX_COMMENT_VOC ON
"X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS_VOICEOFCUSTOMER" ("COMMENT_MESSAGE")
     FAST PREPROCESS OFF
     MIME TYPE 'text/plain'
     ASYNC
     TEXT ANALYSIS ON
     CONFIGURATION 'EXTRACTION_CORE_VOICEOFCUSTOMER'
     LANGUAGE DETECTION ('DE')
     TOKEN SEPARATORS '\/;,.:-_()[]<>!?*@+{}="&';
```

Anhang 22: Ausführung des Full-Text-Index der Tabelle X_LDA_FOUND_TOPICS_JOIN_WITH_COMMENTS_VOICEOFCUSTOMER

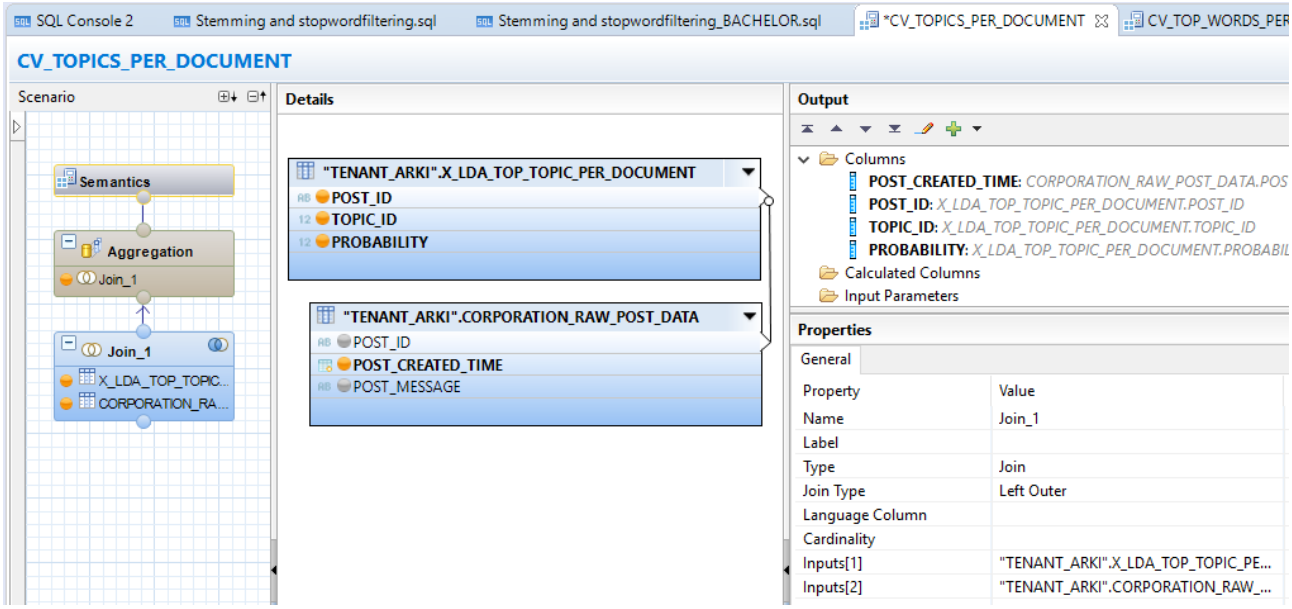
HXE@HXE (TENANT_ARKI) 172.16.20.33 90

SQL Result

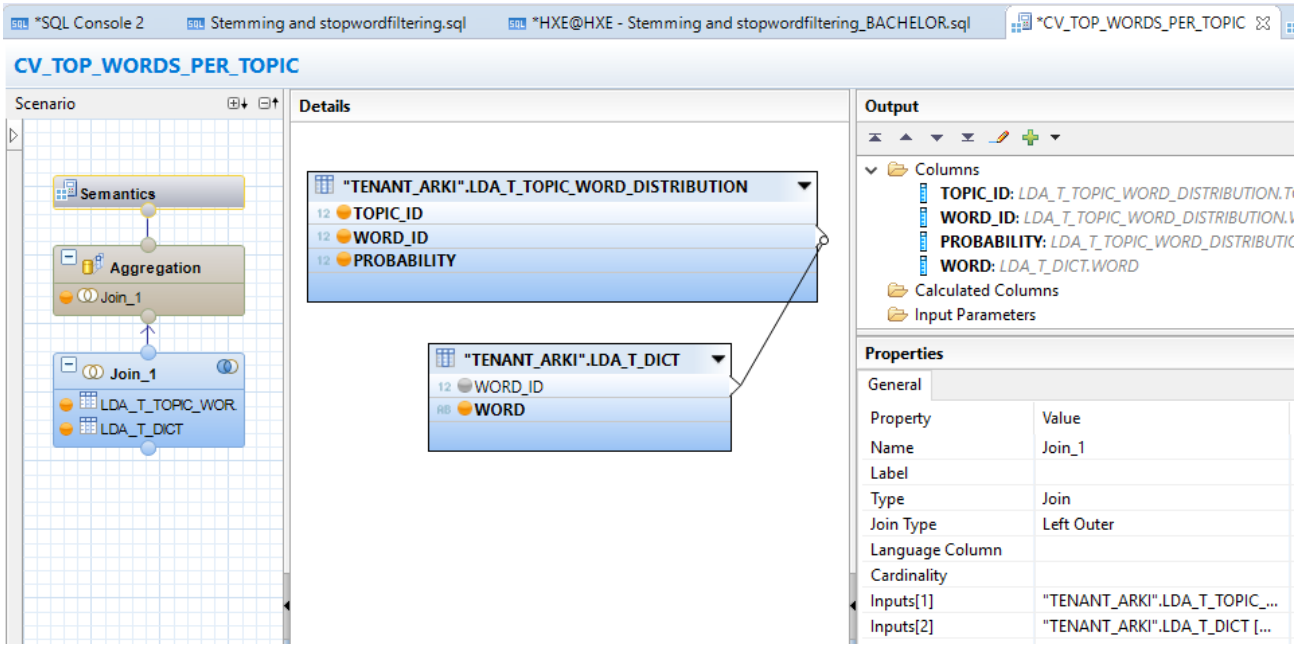
```
SELECT * FROM "$TA_INDEX_COMMENT_VOC"
```

	TOPIC_ID	COMMENT_ID	TA_RULE	TA_COUNTER	TA_TOKEN	TA_LANGUAGE	TA_TYPE
50857	2	251668781651736_465297	Entity Extraction	1	1. Weihnachtsfeiertag	de	HOLIDAY
50858	2	251668781651736_465297	Entity Extraction	3	Heiligabend	de	HOLIDAY
50859	2	251668781651736_465297	Entity Extraction	7	aufwändiger	de	StrongNegativeSentiment
50860	2	251668781651736_464850	Entity Extraction	5	<3 Wir würden dann auch ausführlich auf unserem Testblog darüber berichten :)	de	Emoticon
50861	2	251668781651736_464651	Entity Extraction	8	dreckigem	de	MinorProblem
50862	2	251668781651736_464569	Entity Extraction	1	Mein Schwiegervater kommt und ich kann nicht kochen :-)	de	Emoticon
50863	2	251668781651736_464519	Entity Extraction	2	leider	de	WeakNegativeSentiment
50864	2	251668781651736_464238	Entity Extraction	11	richtig	de	StrongNegativeSentiment
50865	2	251668781651736_463266	Entity Extraction	5	Weihnachten	de	HOLIDAY
50866	2	251668781651736_463266	Entity Extraction	8	richtig	de	WeakNegativeSentiment
50867	2	251668781651736_463146	Entity Extraction	8	einen Tag	de	TIME_PERIOD
50868	2	251668781651736_463146	Entity Extraction	9	Weihnachten	de	HOLIDAY
50869	2	251668781651736_463073	Entity Extraction	4	Weihnachten	de	HOLIDAY
50870	2	251668781651736_462844	Entity Extraction	3	Roetgen	de	LOCALITY
50871	2	251668781651736_462437	Entity Extraction	4	Leider	de	WeakNegativeSentiment
50872	2	251668781651736_462437	Entity Extraction	9	LG	de	PROP_MISC
50873	2	251668781651736_462409	Entity Extraction	2	Es wäre schön wenn ich auch etwas vom Fest dieses Jahr hätte	de	Request
50874	2	251668781651736_462409	Entity Extraction	4	wäre schön	de	GeneralRequest
50875	2	251668781651736_462409	Entity Extraction	8	:)	de	WeakPositiveEmoticon
50876	2	251668781651736_462330	Entity Extraction	6	Und es ist viel einfacher als Ente oder sonstiges selber zuzubereiten :)	de	Emoticon
50877	2	251668781651736_462234	Entity Extraction	8	:D	de	StrongPositiveEmoticon

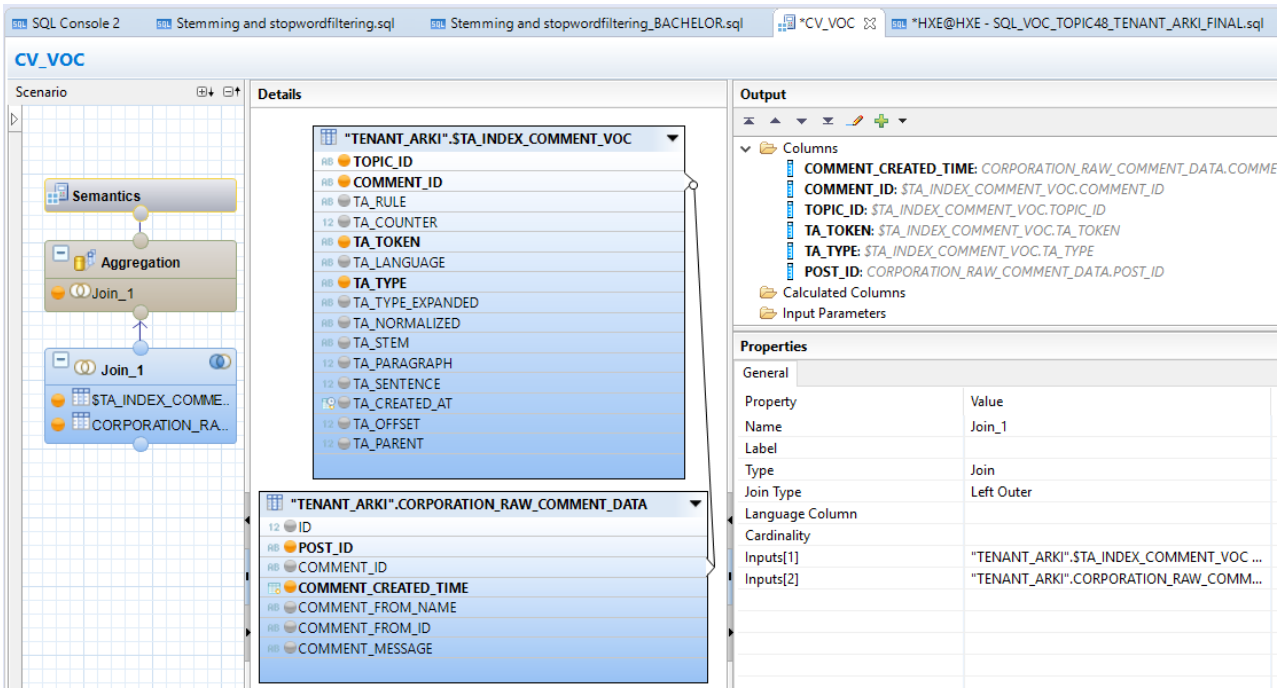
Anhang 23: Ausgabe der Tabelle $TA_INDEX_COMMENT_VOC



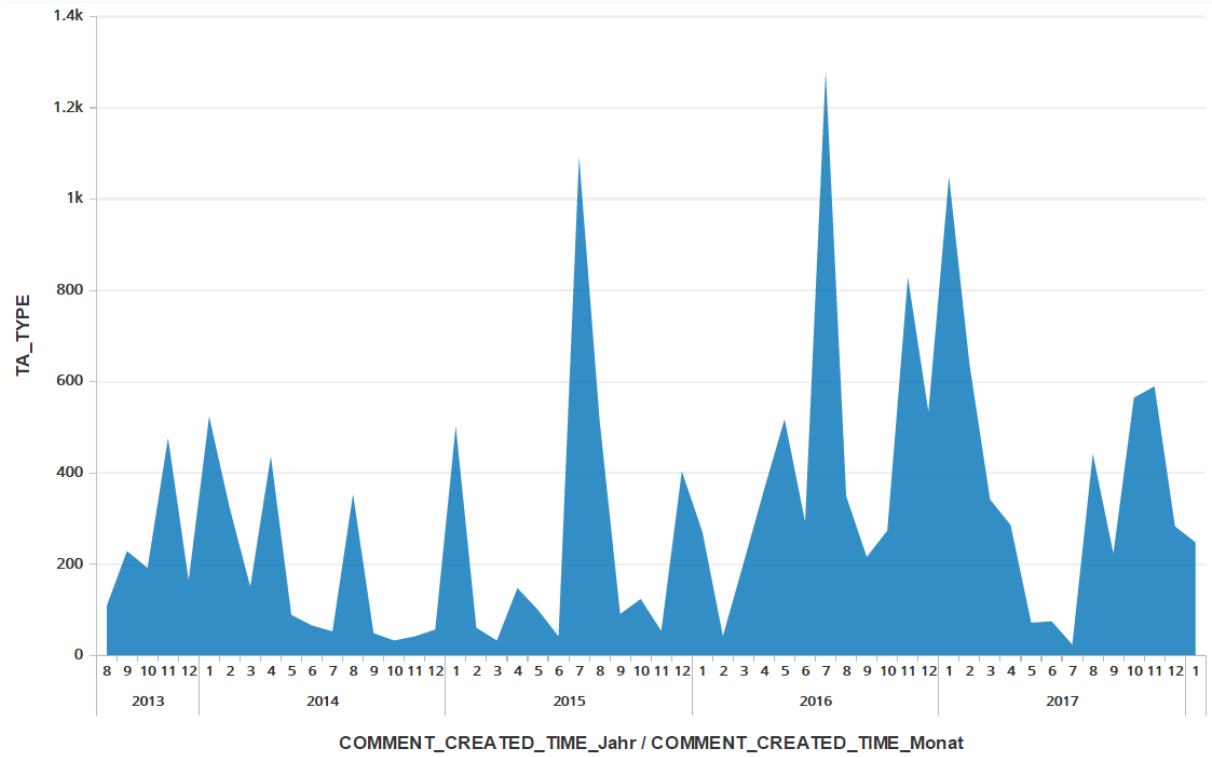
Anhang 24: Calculation View CV_TOPICS_PER_DOCUMENT



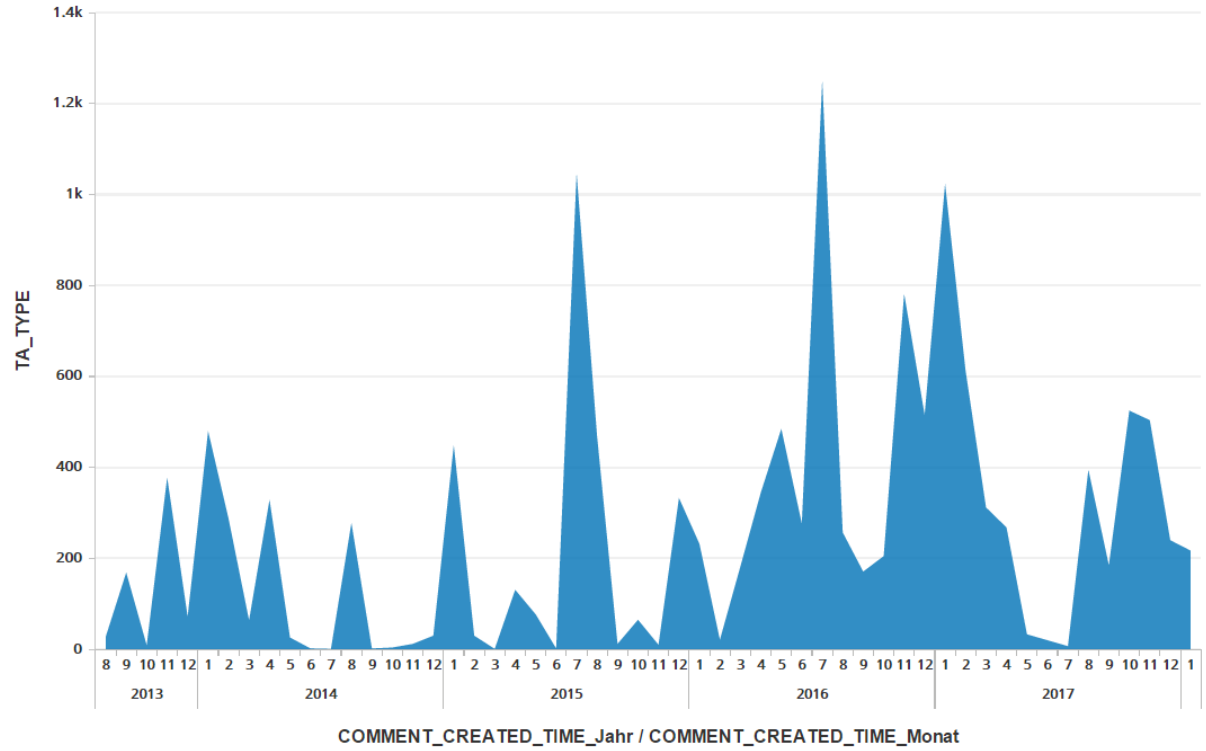
Anhang 25: Calculation View CV_TOP_WORDS_PER_TOPIC



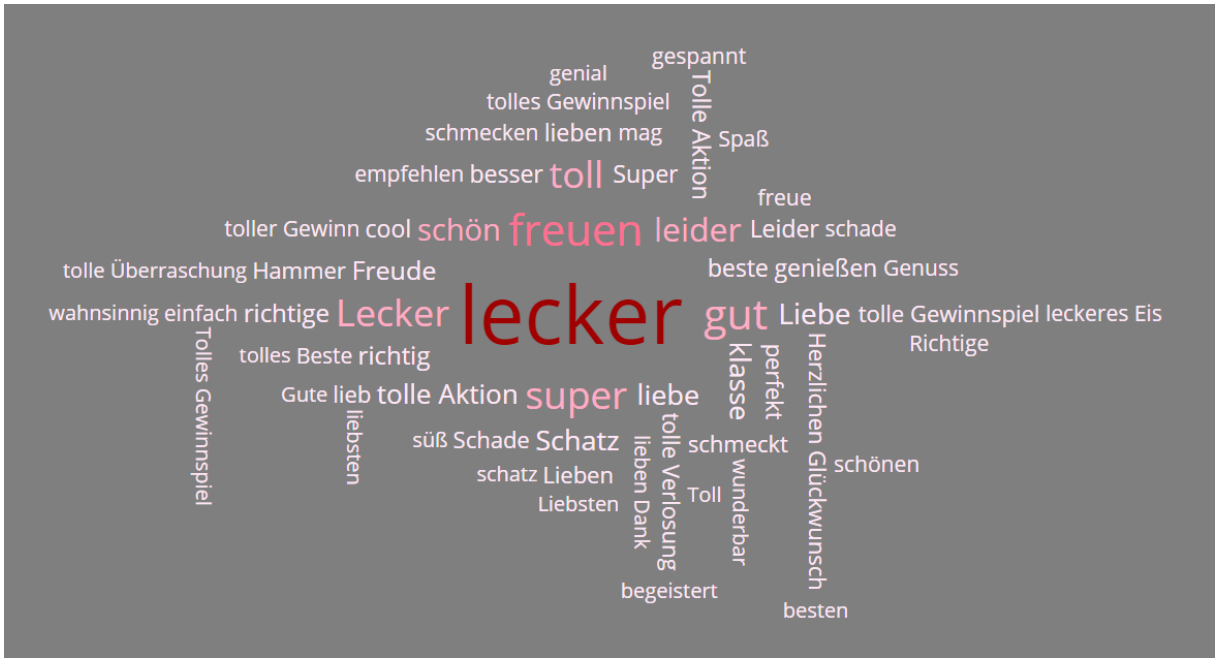
Anhang 26: Calculation View CV_VOC



Anhang 27: Alle extrahierten positiven Sentiments der Kunden-Kommentare von 2013-2018 (alle Themen zusammengefasst) (Quelle: Eigene Darstellung)



Anhang 28: Alle extrahierten positiven Sentiments der Kunden-Kommentare von 2013-2018 über das Thema Gewinnspiele (Quelle: Eigene Darstellung)



Anhang 29: Wort-Wolke über alle Arten der Sentiments der Kunden-Kommentare (Quelle: Eigene Darstellung)

Erklärung über die selbstständige Abfassung der Arbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Ort, Datum

Rechtsverbindliche Unterschrift