

# Multidimensional Knowledge Representation through Integrative Text Mining

A knowledge base framework for extracted information from text

genehmigte DISSERTATION  
zur Erlangung des Grades eines Doktors  
der Ingenieurwissenschaften

vorgelegt von  
Johannes Zenkert (M.Sc.)

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät  
der Universität Siegen  
Siegen 2022

Betreuer und 1. Gutachter: Prof. Dr.-Ing. Madjid Fathi  
2. Gutachter: Prof. Dr.-Ing. Roman Obermaisser  
Vorsitzender: Prof. Dr.-Ing. Kai Daniel

Tag der mündlichen Prüfung: 10. Oktober 2022

*gedruckt auf alterungsbeständigem holz- und säurefreiem Papier*

# Danksagung

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. Madjid Fathi für die Betreuung meines Promotionsvorhabens, alle wertvollen Ratschläge, besonderes Verständnis sowie die großartige Unterstützung während meiner Arbeit am Institut für Wissensbasierte Systeme und Wissensmanagement an der Universität Siegen.

Herrn Prof. Dr. Udo Kelter von der Fachgruppe Praktische Informatik danke ich für eine langjährige Zusammenarbeit und Unterstützung in verschiedenen Projekten, die mir die Anfertigung dieser Arbeit ermöglicht haben.

Herrn Prof. Dr.-Ing. Roman Obermaisser, Chair for Embedded Systems, gilt ebenso mein Dank für die gemeinsame Zusammenarbeit. In vielen hilfreichen Diskussionen konnte ich Ergebnisse dieser Dissertation in Projektvorhaben der beiden Lehrstühle einbringen und somit die wissenschaftliche Relevanz meines Forschungsansatzes evaluieren. Bedanken möchte ich mich in diesem Zuge auch für die Übernahme des Zweitgutachtens.

Bei Herrn Prof. Dr.-Ing. Kai Daniel bedanke ich mich für die Leitung der Promotionskommission als Vorsitzender.

Ein Dank gilt außerdem Herrn Dr. André Klahold und Herrn Prof. Dr. Alexander Holland für die Zusammenarbeit, die vielen hilfreichen Diskussionen, wissenschaftliche Anregungen und Ideen, die zur Anfertigung dieser Dissertation beigetragen haben.

Ich möchte mich ebenfalls bedanken bei allen meinen aktuellen und ehemaligen Kolleginnen und Kollegen am Institut für Wissensbasierte Systeme und Wissensmanagement für die gute Zusammenarbeit im Team. Hierbei möchte ich auch ein Danke sagen an unsere aktuellen und ehemaligen studentischen und wissenschaftlichen Hilfskräfte, mit denen ich zusammen viele unserer Projekte umsetzen konnte.

Von ganzem Herzen danke ich meiner Familie, die mich stets in allen Lebenslagen unterstützt hat, nach Schicksalsschlägen besonders zusammenhält, und somit auch großen Anteil am Gelingen dieser Arbeit hatte.

Siegen, im August 2022

## Abstract

Natural language processing and text mining methods can be used to identify and extract valuable information from unstructured texts. Methodically extracted data provide helpful results but can be difficult to interpret in their individuality and cannot be used directly as knowledge. Cognitively, we as humans are able to process unstructured data, such as natural language in text form, filter out extracted information, classify it semantically, or interpret it. Computer systems cannot do this without help because it requires meaningful processing and combination of the data and information. Knowledge-based approaches attempt to solve this problem by providing appropriate representations for data and information and, by implementing them as expert systems, offer the possibility of reaching conclusions through inference using the knowledge base.

A methodology for structuring and representing acquired information, which can lead to the transformation of data and information from text to knowledge, is conceptualized, implemented, and evaluated in case studies in this dissertation.

The developed approach is called Multidimensional Knowledge Representation (MKR), since the results of different analysis dimensions are combined into a common representation structure by applying individual single text mining approaches, so-called pipelines. The results of text analysis and facets of knowledge acquisition are stored multi-dimensional in a document-oriented database, which can serve as the basis for a knowledge base in knowledge-based applications.

Current systems and tools for text mining are mostly one-dimensional in their application and focus on a specific evaluation in the analysis. They usually provide insights for a previously defined question, which is methodically investigated within the text data as a linear process. In this context, the various perspectives and interpretations of the pipelines can be described as individual analysis dimensions. From the text information can be extracted, for example, after the pre-processing of the text, the named entities, the present topic, contained semantic relations or the sentiment.

The methods of knowledge extraction, such as named entity recognition, topic detection or sentiment analysis are mostly applied individualized by trained methods and deliver a result that is finally interpreted. If the respective analysis question changes, the modified pipeline is often executed again in current state-of-the-art approaches. The core idea of MKR in contrast to current approaches is the support of multi-perspective questions by providing dimensional analysis results in the knowledge base. For example, complex questions such as the sentiment over time about a selected entity in a topic area can be answered efficiently by providing and accessing relevant data in the knowledge base.

In addition to the theoretical foundations of the dissertation project, which lead to the conceptualization and modeling of MKR, the implementation as KB:mkr Knowledge Base Maker is presented. Using specially created text corpora in German and English language, the representation structure is evaluated in an exploratory and case-based manner in various application and project examples in academic and industrial contexts.

# Kurzfassung

Mit Natural Language Processing und Text Mining Methoden lassen sich wertvolle Informationen aus unstrukturierten Texten identifizieren und extrahieren. Methodisch gewonnene Daten liefern hilfreiche Ergebnisse, können jedoch in ihrer Individualität schwer interpretiert und nicht unmittelbar als Wissen eingesetzt werden. Kognitiv sind wir als Menschen in der Lage, unstrukturierte Daten, wie natürliche Sprache in Textform, zu verarbeiten, gewonnene Informationen herauszufiltern, semantisch einzuordnen oder zu interpretieren. Computersysteme können dies nicht ohne Hilfe, denn es bedarf einer sinnvollen Verarbeitung und Kombination der Daten und Informationen. Wissensbasierte Ansätze versuchen dieses Problem durch geeignete Repräsentationsformen für Daten und Informationen zu lösen und bieten durch die Implementierung als Expertensysteme die Möglichkeit, Schlussfolgerungen mithilfe der Wissensbasis zu erzielen. Eine Methodik zur Strukturierung und Repräsentation von gewonnenen Informationen, die zu der Transformation von Daten und Informationen aus Text hin zu Wissen führen kann, wird im Rahmen dieser Dissertation konzeptualisiert, implementiert und in Fallbeispielen evaluiert.

Der entwickelte Ansatz wird als Multidimensional Knowledge Representation (MKR) bezeichnet, da die Ergebnisse verschiedener Analysedimensionen durch die Anwendung individueller einzelner Text Mining Ansätze, sogenannte Pipelines, zu einer gemeinsamen Repräsentationsstruktur zusammengeführt werden. Gespeichert werden die Ergebnisse der Textanalyse und Facetten der Wissensakquisition mehrdimensional in einer Dokumenten-orientierten Datenbank, die als Grundlage für eine Wissensbasis in wissensbasierten Anwendungen dienen kann.

Aktuelle Text Mining Werkzeuge und Tools sind meistens eindimensional in der Anwendung und fokussieren sich auf eine bestimmte Auswertung eines Sachverhalts. Sie liefern meist Erkenntnisse zu einer vorher definierten Fragestellung, die methodisch innerhalb der Textdaten als linear ablaufender Prozess untersucht wird. Als individuelle Analysedimensionen können in diesem Zusammenhang die verschiedenen Perspektiven und Interpretationen der Pipelines bezeichnet werden. Aus den Textinformationen können beispielsweise nach der Vorverarbeitung des Texts, die genannten Entitäten, das vorliegende Thema, enthaltene semantische Relationen oder das Sentiment extrahiert werden.

Die Methoden der Wissensextraktion, wie beispielsweise die Named Entity Recognition, Topic Detection oder Sentiment Analysis werden meistens individualisiert durch trainierte Methoden angewendet und liefern ein Ergebnis, das schließlich interpretiert wird. Verändert sich die jeweilige Fragestellung, findet im Stand der Technik eine erneute Durchführung der modifizierten Pipeline statt. Kernidee der MKR, im Gegensatz zu aktuellen Ansätzen, ist hierbei die Unterstützung von mehrperspektivischen Fragestellungen durch Bereitstellung dimensionaler Analyseergebnisse in der Wissensbasis. So können beispielsweise komplexe Fragestellungen wie des Sentiments über einen Zeitverlauf zu einer ausgewählten Entität in einem Themengebiet durch Bereitstellung und Zugriff auf relevante Daten in der Wissensbasis effizient beantwortet werden.

Neben den theoretischen Grundlagen des Dissertationsvorhabens, die zur Konzeptualisierung und Modellierung der MKR führen, wird die Implementierung als KB:mkr Knowledge Base Maker vorgestellt. Anhand von eigens erstellten Textkorpora in deutscher und englischer Sprache wird die Repräsentationsstruktur explorativ und fallbasiert in verschiedenen Anwendungs- und Projektbeispielen im akademischen und industriellen Kontext evaluiert.

# Contents

<b>Abstract</b>	<b>III</b>
<b>Kurzfassung</b>	<b>IV</b>
<b>Contents</b>	<b>IV</b>
<b>List of Tables</b>	<b>IX</b>
<b>List of Figures</b>	<b>X</b>
<b>List of Abbreviations</b>	<b>XII</b>
<b>List of Equations</b>	<b>XIV</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Problem Statement and Objectives . . . . .	3
1.3 Contributions . . . . .	4
1.4 Thesis Outline . . . . .	5
<b>I Theoretical Background</b>	<b>7</b>
<b>2 Data Mining, Text Mining, Web Mining</b>	<b>8</b>
2.1 Data Mining . . . . .	8
2.1.1 Introduction and Definition . . . . .	8
2.1.2 Process of Data Mining . . . . .	8
2.2 Text Mining . . . . .	9
2.2.1 Introduction and Definition . . . . .	10
2.2.2 Process of Text Mining . . . . .	11
2.3 Web Mining . . . . .	12
2.3.1 Introduction and Definition . . . . .	12
2.3.2 Process of Web Mining . . . . .	14
<b>3 Natural Language Processing</b>	<b>15</b>
3.1 Pre-Processing of Documents . . . . .	15
3.1.1 Tokenization . . . . .	16
3.1.2 Stop Word Removal . . . . .	17
3.1.3 Lemmatization . . . . .	17
3.1.4 Stemming . . . . .	17
3.2 Part-of-Speech (POS) Tagging . . . . .	17

3.2.1	Sentence Parsing and Chunking . . . . .	18
<b>4</b>	<b>Knowledge Extraction From Text</b>	<b>19</b>
4.1	Information Retrieval . . . . .	19
4.1.1	Term Frequency - Inverse Document Frequency . . . . .	19
4.1.2	Word Association . . . . .	19
4.2	Named Entity Recognition . . . . .	21
4.2.1	Methods . . . . .	22
4.3	Sentiment Analysis . . . . .	24
4.3.1	Methods . . . . .	24
4.3.2	Challenges and problems . . . . .	27
4.4	Topic Detection . . . . .	27
4.4.1	Methods . . . . .	28
4.4.2	Challenges and problems . . . . .	31
4.5	Semantic Relationship Analysis . . . . .	32
4.5.1	Wanderlust . . . . .	32
4.5.2	Extraction of RDF statements from text . . . . .	34
4.5.3	Resource Description Framework . . . . .	35
4.5.4	SPARQL Protocol And RDF Query Language . . . . .	36
4.5.5	Semantic search . . . . .	37
4.5.6	Linked Open Data . . . . .	38
4.6	Text summarization . . . . .	39
4.6.1	Abstractive text summarization . . . . .	40
4.6.2	Extractive text summarization . . . . .	40
4.6.3	Hybrid approach . . . . .	40
<b>5</b>	<b>Knowledge Representation and Visualization</b>	<b>41</b>
5.1	Declarative and Procedural Representation of Knowledge . . . . .	41
5.2	Knowledge Representation Development Techniques . . . . .	42
5.3	Knowledge Graph . . . . .	43
5.3.1	Definition of knowledge graph . . . . .	43
5.3.2	RDF Knowledge graphs and SPARQL . . . . .	44
5.3.3	Graph databases . . . . .	45
5.4	Visualization . . . . .	45
5.4.1	Semantic Network Visualization . . . . .	46
<b>II</b>	<b>MKR Framework: Conceptualization and Modeling</b>	<b>51</b>
<b>6</b>	<b>Multidimensional Knowledge Representation</b>	<b>52</b>
6.1	Extraction of meaningful information . . . . .	52
6.1.1	Pre-processing . . . . .	53
6.1.2	Part-of-Speech Tagging . . . . .	54
6.1.3	Named Entity Recognition . . . . .	54
6.1.4	Sentiment Analysis . . . . .	54
6.1.5	Word Association . . . . .	54
6.1.6	Topic Detection . . . . .	55
6.2	Conceptualization of Dimensional Representation . . . . .	55
6.2.1	Metadata information . . . . .	56
6.2.2	Textual Analysis Results . . . . .	58

6.3	Multidimensional knowledge base design . . . . .	59
6.3.1	Design characteristics . . . . .	59
6.3.2	Continuous update cycle for the knowledge base . . . . .	60
6.4	Representation Benefits . . . . .	61
6.4.1	Processing and Representation . . . . .	64
6.4.2	Adaptation of State-of-the-Art Visualization . . . . .	65
6.4.3	Transformation . . . . .	66
<b>7</b>	<b>Extraction, Operations and Pipeline Integration</b>	<b>70</b>
7.1	Pre-Processing Pipeline . . . . .	70
7.1.1	Language Detection . . . . .	70
7.1.2	Sentence Splitting . . . . .	70
7.1.3	Part-of-Speech Tagging . . . . .	70
7.2	Text Mining Methods and Integration of Results . . . . .	71
<b>8</b>	<b>Related Work</b>	<b>73</b>
8.1	Representation Approaches and Frameworks . . . . .	73
8.1.1	Big Data and Big Data Analytics . . . . .	73
8.1.2	Integrative Text Mining . . . . .	74
<b>III</b>	<b>Implementation</b>	<b>76</b>
<b>9</b>	<b>External Data, Lexical Resources and Linked Open Data</b>	<b>77</b>
9.1	External Data . . . . .	77
9.1.1	Stop Word Lists and Language-specific Resources . . . . .	77
9.2	Integration of Lexical Resources for Integrative Text Mining Methods . . . . .	77
9.2.1	SentiWS: Sentiment Evaluation . . . . .	77
9.2.2	Dornseiff: Topic thesaurus . . . . .	78
9.3	Validation through Linked Open Data . . . . .	79
9.3.1	Entity Validation and Auto Classification of Entities . . . . .	79
<b>10</b>	<b>Text Corpora: Data Collection and Pre-Processing</b>	<b>82</b>
10.1	Web Crawling and Web Data Collection . . . . .	82
10.2	Extraction of Web Structures . . . . .	82
10.3	Data Storage Architecture . . . . .	83
10.3.1	Web Crawling with R - Dynamic Web Crawling . . . . .	83
10.3.2	Crawling Procedure . . . . .	83
<b>11</b>	<b>KB:mkr: Knowledge Base Maker</b>	<b>85</b>
11.1	Technology Overview . . . . .	85
11.2	Software Implementation . . . . .	86
11.2.1	Data Analysis . . . . .	89
11.2.2	Named Entity Recognition . . . . .	89
11.2.3	Sentiment analysis . . . . .	89
11.2.4	Topic detection . . . . .	90
11.2.5	Semantic triple extraction . . . . .	90
11.3	Implementation of Representation Format . . . . .	91



<b>IV</b>	<b>Experimental Results, Case Studies and Evaluation</b>	<b>93</b>
<b>12</b>	<b>Exploratory Analysis and Evaluation of Text Corpora</b>	<b>94</b>
12.1	Knowledge Discovery Interface . . . . .	94
12.2	Exploratory Analysis: Topic Development . . . . .	96
12.3	Exploratory Analysis: Sentiment Analysis . . . . .	98
12.3.1	Temporal and Entity-related Sentiment Analysis . . . . .	98
12.4	Exploratory Analysis: Named Entity Recognition . . . . .	99
12.4.1	Entity Overview . . . . .	100
12.5	Exploratory Analysis: Word Associations . . . . .	100
12.6	Full-Text Associative Search . . . . .	101
<b>13</b>	<b>Multidimensional Knowledge Visualization</b>	<b>103</b>
13.1	Visualization Transformation and Dynamic Knowledge Maps . . . . .	103
13.2	Transformation of MKR into a Knowledge Graph . . . . .	104
<b>14</b>	<b>Use Cases and Applications</b>	<b>107</b>
14.1	Text Summarization via MKR . . . . .	107
14.1.1	Dimensional Text Summarization . . . . .	107
14.1.2	Experimental Result . . . . .	108
14.1.3	Limitations . . . . .	109
14.2	Semantic Relationship Analysis via MKR . . . . .	111
14.2.1	Matching with Wikidata Properties . . . . .	111
14.2.2	Building an Ontology Structure and SPARQL Queries . . . . .	111
14.2.3	Semantic search with SPARQL . . . . .	112
14.3	Knowledge Extraction from Forums . . . . .	113
14.4	Knowledge Graph Development with Neo4j based on MKR . . . . .	114
14.4.1	Neo4j knowledge graph . . . . .	115
<b>V</b>	<b>Discussion</b>	<b>116</b>
<b>15</b>	<b>Summary and Conclusion</b>	<b>117</b>
15.1	Summary of Contributions . . . . .	117
15.2	Discussion of Results . . . . .	118
15.3	Future Work . . . . .	118
	<b>Publications</b>	<b>119</b>
	<b>Bibliography</b>	<b>130</b>

# List of Tables

4.1	Contingency table for statistical topic detection (Whitney et al., 2009) . . . . .	31
5.1	Suitability of the design variables, (Pfeffer, 2010, p. 235) . . . . .	46
6.1	Typical Visualization Types for Text Mining Analysis Results (Zenkert et al., 2018) . .	63
6.2	Knowledge Graph Visualization Examples based on MKR (Zenkert et al., 2018) . . .	69
9.1	Topic thesaurus (Dornseiff, 2004) . . . . .	78

# List of Figures

1.1	The knowledge pyramid - DIKW chain . . . . .	2
1.2	Thesis Outline and Chapter Overview . . . . .	6
2.1	An Overview of the Steps That Compose the Knowledge Discovery in Databases Process (Fayyad et al., 1996) . . . . .	9
2.2	Text Mining Process (Hippner and Rentzmann, 2006) . . . . .	11
2.3	Web Mining Taxonomy, adapted from (Singh and Singh, 2010). . . . .	13
2.4	Web Mining Process . . . . .	14
3.1	Sequence of pre-processing of documents, adapted from (Vijayarani et al., 2015). . .	16
4.1	Sentiment Analysis Taxonomy, adapted from (Medhat et al., 2014). . . . .	25
4.2	Latent Dirichlet Allocation (LDA) Process, (Blei et al., 2003, p. 997) . . . . .	30
4.3	Wanderlust: Example sentence with relations (Akbik and Broß, 2009) . . . . .	33
4.4	Processing steps of Wanderlust (Akbik and Broß, 2009) . . . . .	34
4.5	Binary and n-ary representation of an example sentence (Martinez-Rodriguez et al., 2019). . . . .	35
4.6	RDF statements as graph (Hitzler et al., 2007) . . . . .	36
4.7	RDF triples in the RDF/XML syntax (Hitzler et al., 2007). . . . .	36
4.8	Representation of RDF triples in Turtle syntax (Hitzler et al., 2007) . . . . .	37
4.9	Example of a SPARQL query (Hitzler et al., 2007) . . . . .	37
4.10	Process of creation or deletion of new Wikidata properties, adapted from (Samuel, 2017). . . . .	39
5.1	A taxonomy of knowledge representations, adapted from (McNamara, 1994). . . . .	42
5.2	Example RDF knowledge graph in Neo4j graph database (Thorsten Liebig, 2018) . .	45
5.3	Size differences for quantitative and nominal attributes (Pfeffer, 2010, p. 235) . . . .	47
5.4	Munsell color system (Krempel, 2010, p. 549) . . . . .	48
5.5	Aesthetics in graphs (Bennett et al., 2007, p. 60) . . . . .	49
6.1	Extraction of entity information from textual resources with different analysis results (Zenkert and Fathi, 2016) . . . . .	53
6.2	Conceptual visualization of a three dimensional relation between entities, assigned documents and sentiment evaluation in corresponding topics (Zenkert and Fathi, 2016) . . . . .	56
6.3	Update process of the multidimensional knowledge base (Zenkert and Fathi, 2016) .	61
6.4	The process of enrichment in MKR (Zenkert et al., 2018) . . . . .	64
6.5	Knowledge graph types and different visualizations of MKR representation (Zenkert et al., 2018) . . . . .	68
7.1	Extended Text Mining Process and Pipeline Integration . . . . .	70

7.2 Architectural Overview of Process Analytics through Integrative Text Mining utilizing MKR (Zenkert et al., 2018) . . . . .	72
9.1 Training process of entities from current selected article in KB:mkr . . . . .	79
9.2 Training process of entities from current selected article in KB:mkr . . . . .	80
9.3 Entity validation and classification in KB:mkr - Classification of entity type “person” .	80
9.4 Knowledge Base Update in KB:mkr - Entity update . . . . .	81
10.1 Web Crawling Process, Data Storage and Pre-Processing Architecture (Zenkert et al., 2018) . . . . .	84
11.1 Model-View-ViewModel (MVVM) - Abstract View . . . . .	86
11.2 KB:mkr Knowledge Base Maker Logo . . . . .	87
11.3 KB:mkr Knowledge Base Maker User Interface - Text Analytics Module . . . . .	87
11.4 Implementation Overview and KB:mkr Knowledge Base Maker Architecture (Zenkert et al., 2018) . . . . .	88
11.5 Exemplary Sentiment Evaluation based on Document’s Emotion Classification (Normalized Scale from 0 to 1) (Zenkert et al., 2018) . . . . .	90
11.6 Java Script Object Notation (JSON) Format of MKR (Colored areas indicate integrative text mining results) (Zenkert et al., 2018) . . . . .	92
12.1 KB:mkr Knowledge Discovery Components . . . . .	95
12.2 KB:mkr Bar Chart on topic development . . . . .	96
12.3 KB:mkr Tile View on topics with subtopics . . . . .	97
12.4 KB:mkr Tile View drill-down on topic “society” - Overview of related entities . . . . .	98
12.5 KB:mkr Area Graph on Sentiment Development . . . . .	99
12.6 Sentiment analysis on topics on 01.10.2019 - Map View . . . . .	99
12.7 KB:mkr Entity Overview . . . . .	100
12.8 KB:mkr Entity Overview - Filter operation on entity type “Organization” . . . . .	100
12.9 KB:mkr Temporal Word Association Strength between Entities “Ukraine” (DE: Ukraine) and “Russia” (DE: Russland) . . . . .	101
12.10 Visualization of associative search with associative term extensions based on the CIMAWA (Zenkert et al., 2016) . . . . .	102
13.1 Conceptual overview of a dynamic knowledge map. Different entities (e.g., persons, places) are arranged by distances derived from Concept for the Imitation of the Mental Ability of Word Association (CIMAWA) word association strength. . . . .	104
13.2 KB:mkr Exemplary knowledge graph on documents . . . . .	105
13.3 KB:mkr Knowledge graph from appearing documents from 1st October 2019 . . . . .	106
14.1 Dimensional Text Summarization Algorithm in MKR knowledge base (Zenkert et al., 2018) . . . . .	110
14.2 KB:mkr - Free text analysis for extraction of facts . . . . .	112
14.3 KB:mkr - Table with extracted facts . . . . .	112
14.4 KB:mkr - Details view of an extracted semantic relationship . . . . .	112
14.5 Process of web crawling and natural language processing (Meckel et al., 2019) . . .	114
14.6 Process of the synthesis and subsequent optimization of the diagnostic graph (Meckel et al., 2019) . . . . .	114
14.7 Exemplary KIRETT Knowledge Graph - Base measurements . . . . .	115

## List of Abbreviations

<b>AI</b>	Artificial Intelligence
<b>AGF</b>	Associative Gravity Force
<b>BSON</b>	Binary Java Script Object Notation
<b>CIMAWA</b>	Concept for the Imitation of the Mental Ability of Word Association
<b>CSS</b>	Cascading Style Sheets
<b>GUI</b>	Graphical User Interface
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IE</b>	Information Extraction
<b>IR</b>	Information Retrieval
<b>IRI</b>	International Resource Identifier
<b>JS</b>	JavaScript
<b>JSON</b>	Java Script Object Notation
<b>KDD</b>	Knowledge Discovery in Databases
<b>KDT</b>	Knowledge Discovery from Text
<b>LDA</b>	Latent Dirichlet Allocation
<b>LOD</b>	Linked Open Data
<b>ML</b>	Machine Learning
<b>MKR</b>	Multidimensional Knowledge Representation
<b>MVVM</b>	Model-View-ViewModel
<b>NER</b>	Named Entity Recognition
<b>NLG</b>	Natural Language Generation
<b>NLP</b>	Natural Language Processing
<b>OWL</b>	Web Ontology Language
<b>POS</b>	Part-of-Speech
<b>RDF</b>	Resource Description Framework
<b>RNN</b>	Recurrent Neural Network

<b>ROUGE</b>	Recall-Oriented Understudy for Gisting Evaluation
<b>SPARQL</b>	SPARQL Protocol And RDF Query Language
<b>SQL</b>	Structured Query Language
<b>TDT</b>	Topic Detection and Tracking
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>WPF</b>	Windows Presentation Foundation
<b>WWW</b>	World Wide Web
<b>XML</b>	Extensible Markup Language

# List of Equations

4.1	Calculation of inverse document frequency (Manning et al., 2010)	19
4.2	Calculation rule for the TF-IDF measure of a word (Manning et al., 2010)	19
4.3	Tanimoto coefficient	20
4.4	Dice coefficient	20
4.5	Transinformation	20
4.6	Log-likelihood-Function	20
4.7	Poisson distribution	20
4.8	T-Score	20
4.9	Calculation of word association strength according to CIMAWA, (Uhr et al., 2013)	21
4.10	Associative Sentiment based on CIMAWA (Uhr et al., 2014)	26
4.11	Calculating the similarity of two documents using the cosine function (Muflikhah and Baharudin, 2009).	29
4.12	Calculation of Associative Gravity Force (AGF) values (Klahold et al., 2013)	29
4.13	Chi-square test for statistical topic detection (Whitney et al., 2009)	31
4.14	Deviation comparison for statistical topic detection (Whitney et al., 2009)	31
4.15	Gaussian deviation for statistical topic detection (Whitney et al., 2009)	31

# 1 Introduction

The individual epochs of mankind are often named after technical achievements that had an enormous influence on social development. Thus, earlier eras were already characterized by the ability to process new metals and materials. This technical progress was accompanied by social advancements and the need for people to adapt to new conditions. Today's 21st century is often referred to as the Information Age, in which the use of computers and the digitization of society are advancing and both influences are noticeably increasing.

Nowadays, information has an outstanding importance in our time. Comparable to the use of new raw materials and the influential technical achievements in earlier times, the impact on people through ubiquitous access to information in the vast majority of the world is a huge potential and yet one of the greatest challenges of our time.

Digitization is changing society, and people must rise to this challenge. People have access to ever-increasing amounts of information from an increasingly widespread supply of information. In this context, there is often the reference to an information overload or so-called information explosion. In this context, the quote *"We are drowning in information but starving for knowledge"* by John (1982) is well known and more relevant than ever. Nowadays, people have to deal with a flood of old and new information in all walks of life. Technological progress, in particular the increased mobility through modern end devices and the widespread use of the Internet in everyday life, are factors in the explosive growth of data. The increased use and growing integration of social media, news and streaming services into our lives are just a few reasons why.

Information is diverse and available in an unmanageable quantity. The extent to which the inappropriate handling of the flood of information has negative effects on health is discussed again and again. For this reason, people have to learn how to obtain information in an even more targeted manner and to question it critically. It is essential to filter the information in order to be able to distinguish between important and unimportant information.

The concepts of cognitive perception, collection and use of data, transformation into information, interpretation as knowledge and expert knowledge as wisdom are the main components of the human logical decision-making process. From a more technical standpoint, the relation between the same concepts of input data, information, knowledge, and wisdom is often expressed through the widely adopted concept of the so-called knowledge pyramid. Rowley (2007) summarizes also other similar concepts as *"the Data-Information-Knowledge-Wisdom hierarchy (DIKW), referred to variously as the knowledge hierarchy, the information hierarchy and the knowledge pyramid is one of the fundamental, widely recognized and taken-for-granted models in the information and knowledge literature"* (Rowley, 2007).

For the relationships in the knowledge pyramid between data, information, and knowledge, it can basically be stated in most approaches that knowledge can arise when information has been understood and can therefore be used to solve a problem. Information arises when data is used meaningfully and combined purposefully in a context. Here, software systems can interpret data as character strings and understand them as information by linking them to other data and information. Wisdom can be understood as expert knowledge. Generally, from bottom to the top of the pyramid, volume of available and created sources decreases and the complexity rises.

The widely-adopted concept of the knowledge pyramid, and described as DIKW chain (Rowley, 2007), is illustrated in Figure 1.1.



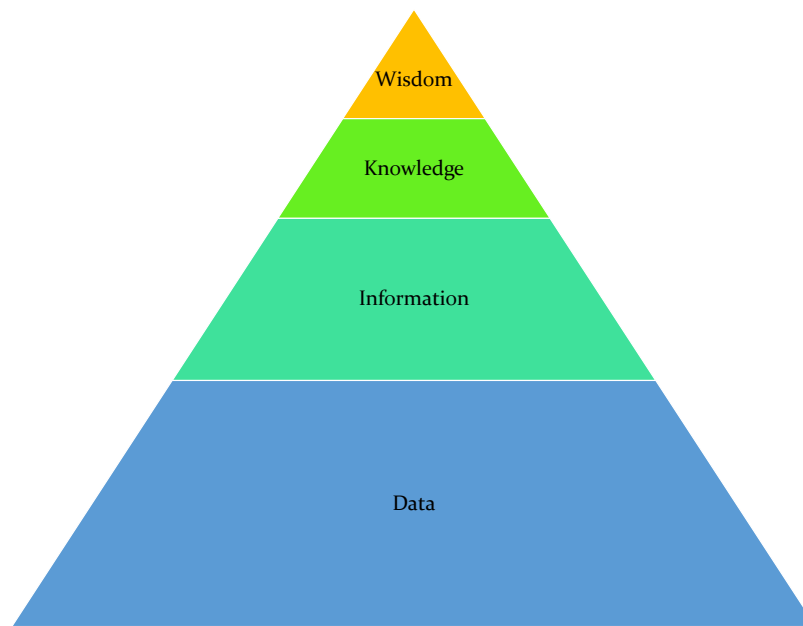


Figure 1.1: The knowledge pyramid - DIKW chain

A major challenge, however, is to process data in a way that identifies targeted information and extracts useful knowledge from it. The definition of “useful knowledge” has to be considered individually. In the textual context, it ranges, for example, from targeted topic detection in news articles, semantic relationship analysis in encyclopedias, to sentiment analysis of individual articles or analytical exploration of larger text corpora. The retrieval, recognition, and combination of relevant information by the intelligent application of knowledge extraction techniques is the focus of these tasks.

Even if finding out “useful knowledge” is a difficult task, all mentioned approaches for knowledge generation from text have in common that they can be applied computer-aided by text mining.

## 1.1 Motivation

Text mining is a very established and still expanding research field in the area of multifaceted processing of text, text analysis and visualization, and preparation of textual data for search engines. Text mining can be considered as a method to counteract the information overload by automatic processing of documents and to provide the results of automatic analyses for knowledge-based approaches. In this dissertation, text mining is used as a fundamental concept for knowledge acquisition.

The field of text mining was created at the beginning of the 1980s from the field of data mining in order to meet the requirements of textual data analysis and evaluation motivated by business applications to recognize information semantically, linguistically and grammatically by means of computer-assisted procedures (Sirmakessis, 2012). In contrast to data mining, which can build on a defined structure of a database for processing the data, texts turn out to be unstructured due to particularly complex syntax and contained semantics in various languages.

Since the 1990s, text data has dominated relational data elements in information systems, so there has been a need for techniques that are able to extract important knowledge from both text data and relational data (Jo, 2019). The objective of the analysis is often to identify concepts and topics from the data and to discover hidden relationships, patterns, and trends in the data.

In 2001, Berners-Lee et al. (2001) proposed the semantic web as an extension to the World Wide Web (WWW). The semantic web is supposed to be able not only to provide content, but also to understand its meaning, thus taking both cross-machine communication and communication between humans and computers to a new level. The steps in this direction are clearly visible. For example, search engines no longer function exclusively on the basis of character strings but attempt to determine the underlying terms by searching for inflected forms of the word forms entered and for possible synonyms.

The research areas of “data mining”, “text mining” and “web mining” have gained significantly in importance in recent years. Compared to a survey by Mehler and Wolff (2005) in the year 2005, the respective search results for the mining methods in the Google search engine have increased from initially 6,850,000, 301,000 and 136,000 to 31,000,000, 448,000, 441,000 search results in 2017 and 114,000,000, 11,100,000 and 1,190,000 search results in the year 2022.

A similar picture of the increase in search results shows when evaluating the topic-linked research papers via Google Scholar. In the scientific area, the increase from 122,000, 4,180 and 2,790 search results in 2005 (from the survey by Mehler and Wolff (2005)) to 1,550,00, 173,000 and 54,300 in 2017 and to 2,670,000, 352,000 and 84,600 search results in the year 2022 is even higher compared with the regular search results from the Google search engine. It can be seen that the aforementioned topics have become more and more important over the years.

In recent years, artificial neural networks have been increasingly used in the field of Artificial Intelligence (AI) to make machines intelligent by recognizing patterns and structures in given data. Nevertheless, natural language is very complicated and is not yet fully understood by machines. However, it has been possible to develop machines based on machine learning and the use of deep neural networks that come close to humans in the use of linguistic capabilities. GPT-3, a language processing model developed by the American non-profit organization OpenAI, can be mentioned as a well-known example and achievement from this progress (Dale, 2021). In this dissertation, language models, which are less used to extract knowledge and are often used to generate text, were not conceptually applied and implemented, but are necessary to be noted here.

A knowledge base that captures the storage of results obtained through text mining and their relationships to each other is a powerful tool to cope with an unmanageable amount of information. Accordingly, the short- and long-term storage of extracted information in a multidimensional structure in a knowledge base could be an approach to access and use existing knowledge for information retrieval and knowledge discovery.

## 1.2 Problem Statement and Objectives

Due to the rapidly growing number of electronic documents, it is becoming more and more time-consuming to continuously review and process all relevant sources. Without technical assistance, the task of extracting relevant knowledge from mostly unstructured information turns out to be an impossible challenge. A knowledge-based system for the extraction of information, and thus interpretation into knowledge, obtained with different text mining approaches could solve this problem in an application-oriented way and support the user. It can be used specifically for different analysis tasks, for example in the research area for the analysis of trends or developments, which would not be possible without the automation process.

Essential in this approach is the possibility of storing the extracted information efficiently in a representation method and using it as the basis for the knowledge base. The query of the knowledge base then enables different applications to access a common data basis to use the evaluation results in combination in more detailed analyzes without having to carry out individual text mining techniques and evaluation again.

For the extraction of knowledge, the mostly unstructured textual information must first be

transformed into a suitable format for further processing and finally into a suitable structure, which makes a higher level analysis possible in the first place (Manning and Schütze, 1999). Furthermore, a correct interpretation often requires background knowledge that is not part of the underlying text corpus and thus requires the combination of results from competing processing paths.

## 1.3 Contributions

The goal of the dissertation is to develop the MKR which is implemented in a knowledge-based system. In this context, the application shall take into account the theoretical background of the fields of data mining, text mining, web mining and integratively combine specialized text mining methods and pipelines into one representation. Topic detection, entity recognition or sentiment analysis are to be exemplified here, whose analysis results are to be provided as possible analysis dimensions to query the knowledge base contents accordingly.

The contributions of the dissertation are in detail:

1. **Development of a web crawling framework and generation of text corpora:** This part of the dissertation develops a concept and implementation for dynamic content recognition from web pages based on web structures and the subsequent extraction of potential knowledge from the documents obtained.
2. **Integration of text mining pipelines into a common representation:** As a second contribution, the prototype should be able to process a provided text corpus through different text mining pipelines to be implemented and integrate the results in the MKR structure into the knowledge base.
3. **Methods for dimensional filtering of knowledge representation:** By filtering different MKR properties, the courses and changes of the represented results of different time and analysis dimensions in the knowledge base can be determined. As a third contribution, a method set will be developed that enables the selection, transformation, and filtering of MKR results.
4. **Implementation of KB:mkr software:** On the base of aforementioned contributions, a modular software will be implemented that can process, store and query extracted information. For a targeted analysis of the knowledge base a MKR is aimed at. For the implementation, two forms of process flow are to be considered. First, it should be possible to retrieve specific web documents from the database, then select them, display, and save associated extracted information. In addition, automation can be used, so that documents are regularly read in, processed, and knowledge is extracted autonomously without the need for further user interaction. The knowledge base, the representation of the knowledge, as well as the program should be extendable by further components. The aim is therefore also to provide a Graphical User Interface (GUI) in such a way that it can be used for future work.
5. **Implementation of knowledge extraction methods and integrative text mining:** Possible queries are essentially performed using the proper names and the corresponding associated topic fields (e.g., associated entities such as person or location names, topics, sentiment values). By implementing different text mining methods, the knowledge structure of the knowledge base is enriched more and more, thus enabling complex queries. The merging of the methods as integrative text mining is the fifth contribution of this dissertation.
6. **Exploratory data analysis and transformative visualization:** The exploratory analysis and data visualization are the sixth contribution. The representation method is the basis for

different types of visualization, which can be transformed into each other, can represent different contexts, but at the same time consider the same data basis of the MKR structure. One of the visualization variants is the so-called knowledge graph, which can be created flexibly from the MKR structure according to requested analysis dimensions.

## 1.4 Thesis Outline

This dissertation consists of five main **Parts** as outlined in Figure 1.2. At the beginning, before the main parts, **Chapter 1** introduces the considered research direction of text mining, knowledge extraction, the proposed combination of results as MKR and its implementation in a knowledge-based system. The motivation, problem statement and objectives as well as scientific contributions of the thesis are given.

**Part I** provides the theoretical backgrounds of various subtopics. It starts with the introduction of the concepts data mining, text mining, and web mining with their related methods and processes in **Chapter 2**. Natural Language Processing (NLP), the basic procedure for preparing and pre-processing of mined textual content, is covered in **Chapter 3**. **Chapter 4** introduces several advanced knowledge extraction methods and text mining techniques used for the knowledge acquisition in this dissertation. In **Chapter 5**, knowledge representation and knowledge visualization of extracted results are discussed as relevant background and fields for MKR in relation to the concept of integrative text mining.

**Part II** covers the conceptualization and modeling of the MKR framework. For this purpose, **Chapter 6** introduces the core of the novel representation method for combining text mining results. It further provides an overview of the MKR enabled knowledge base design and its benefits. In **Chapter 7** the corresponding NLP methods, their results as pipelines and the integration into the MKR are discussed. **Chapter 8** compares MKR with existing methods from literature and provides a state-of-the-art overview. Furthermore, aspects of text mining and knowledge extraction to be combined as integrative text mining are covered.

**Part III** presents the implementation of MKR in the KB:mkr Knowledge Base Maker software. For this purpose, **Chapter 9** first introduces required external data and lexical resources from previously introduced text mining methods. The processes and implementation of data collection and pre-processing of data to prepare text corpora is described in **Chapter 10**. A flexible approach for web crawling which has been implemented is covered as well. The description of the core implementation of the KB:mkr Knowledge Base Maker is given in **Chapter 11**. Several screenshots illustrate the software and provide an overview of the knowledge-based system.

**Part IV** contains various experiments and results of different use cases and applications. It evaluates the MKR method through application in different project-related topics. In **Chapter 12** an exploratory data analysis of the web crawled text corpora is presented. **Chapter 13** discusses the adaptability and transformation of knowledge visualization through MKR. Additional use cases are covered in **Chapter 14**.

**Part V** deals with the final discussion of the results in form of a summary of contributions in **Chapter 15**. Furthermore, an outlook as future work and further developments are presented.

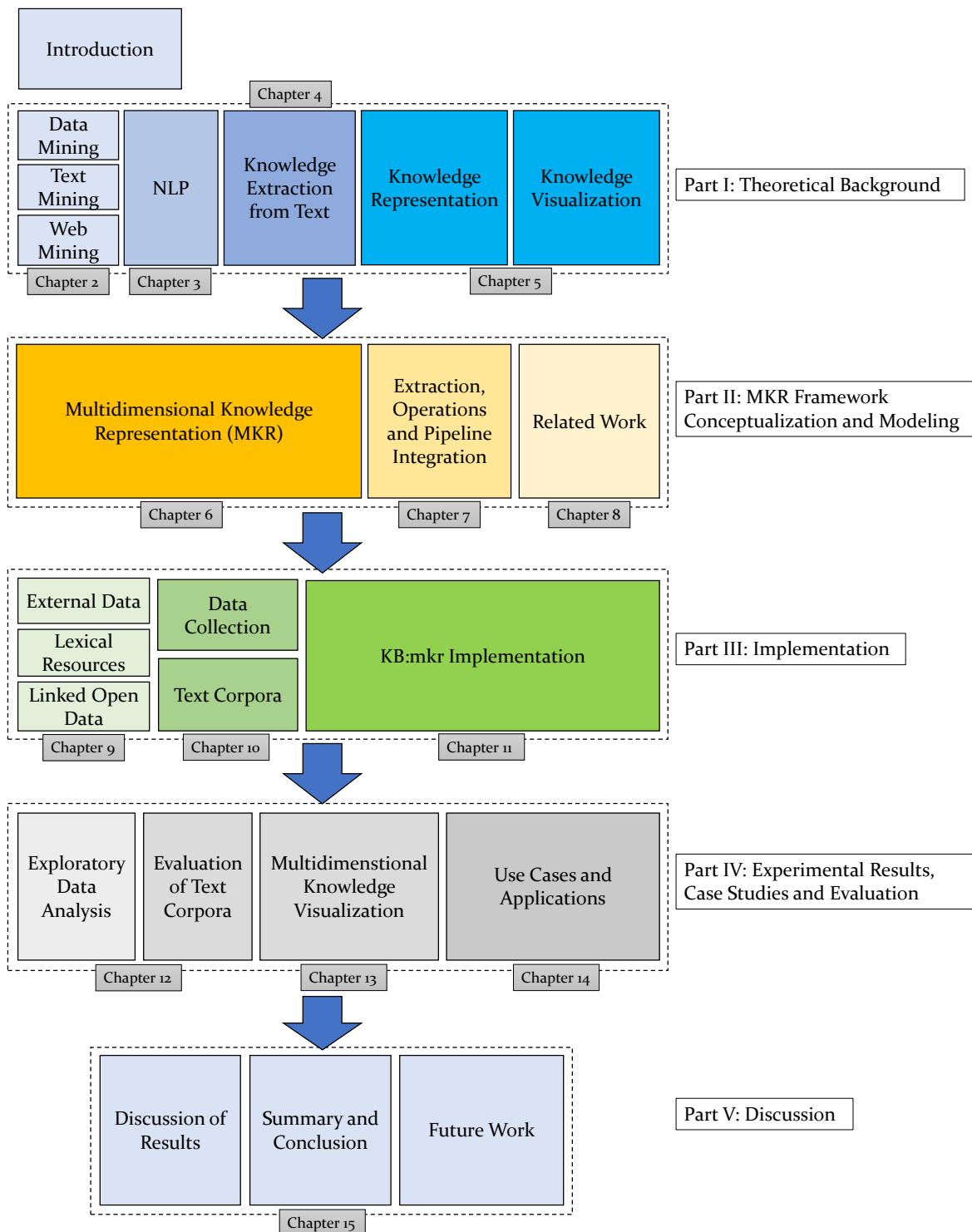


Figure 1.2: Thesis Outline and Chapter Overview

**Part I**

**Theoretical Background**

## 2 Data Mining, Text Mining, Web Mining

This chapter is dedicated to the theoretical background of different information mining concepts. Data mining, text mining and web mining techniques are relevant for the overall objective of this thesis. Here, mining methods support the applications of techniques for extracting, processing, storing, searching, and using meaningful information in a knowledge-based system. Therefore, the mentioned mining techniques are presented and introduced separately. In each section, a distinction is made between different definitions from the literature and an introduction to the historical developments as well as the current state-of-the-art is given.

### 2.1 Data Mining

First, data mining is introduced as the most popular way of obtaining information from an unknown data set. Data mining is the most common paraphrase for various methods of data analysis which are used in different areas. Typically, a database is involved as a starting point of data analysis.

#### 2.1.1 Introduction and Definition

The amount of data stored in structured and unstructured databases is constantly increasing. Historically, new ways had to be found to store increasing sizes of data in order to make appropriate use of available data and resources. From this necessity, the concept of Knowledge Discovery in Databases (KDD) was created (Kalavathy et al., 2007). (Sharafi, 2013, p.51) describes the KDD process as *“the totality of procedures [to identify] valid, previously unknown, useful and understandable patterns in large data sets”*.

Data mining itself is seen as a process step of the KDD. According to Sharafi (2013), it is the most important process step. However, in order for data mining to produce meaningful results, the other process steps of the KDD process, especially the pre-processing and interpretation of analysis results, are also very important to be considered and prepared carefully.

The concept of KDD has existed since a conference in 1989 (Fayyad et al., 1996). The need for a procedural model for the KDD process was already discussed at this conference (Sharafi, 2013). Different process models for the KDD process were developed. An overview of the process models can be found in (Kurgan and Musilek, 2006). All models use a different number of individual selection, pre-processing, and analysis process steps.

#### 2.1.2 Process of Data Mining

According to Sharafi (2013), the KDD process can be reduced to four essential process steps considering only the main steps. The most frequently used model, however, is the step-by-step model according to Fayyad et al. (1996). Figure 2.1 shows the essential process steps of the model.

The model according to Fayyad et al. (1996) consists of nine process steps. The individual process steps are as follows:

1. **Domain understanding and target definition:** The application domain must be understood and the goals to be achieved must be defined.
2. **Data selection:** The data to be analyzed in order to achieve the defined goal is selected.

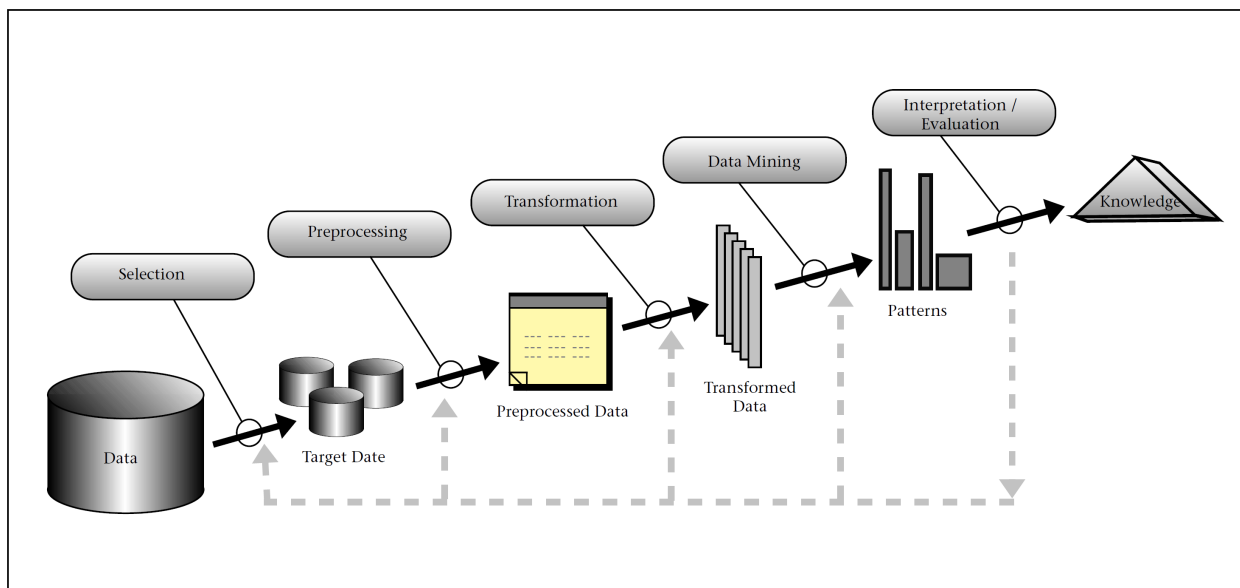


Figure 2.1: An Overview of the Steps That Compose the Knowledge Discovery in Databases Process (Fayyad et al., 1996)

3. **Data pre-processing and data cleansing:** Data pre-processing includes the detection of noise, as well as errors and inconsistencies in the database. The elimination of these defects is one of the most important tasks of data pre-processing.
4. **Data transformation:** By reducing the data dimensions and transforming the data, the number of variables under consideration is reduced.
5. **Coordination of goals and methods:** It has to be coordinated with which data mining methods (e.g., cluster analysis) the objectives of the KDD process from step 1 can be achieved.
6. **Model selection:** The data mining algorithms and models to be used for pattern discovery must be selected.
7. **Data mining:** Identification and detection of patterns in the data.
8. **Interpretation and evaluation:** The recognized patterns are interpreted and visualized.
9. **Knowledge processing:** The acquired knowledge must be documented properly and applied meaningful.

An essential aspect of data mining is that the data is available in a structured form (Singh and Singh, 2010). Instead of data mining, the terms knowledge extraction, information discovery, information harvesting, data archaeology and data pattern processing can also be found in the literature (Fayyad et al., 1996).

## 2.2 Text Mining

This section introduces the text mining. After the term is defined, an introduction to the topic is given. Then, the process of text mining is examined in more detail. Thereby, not only the basics are discussed, but also a delimitation to the superordinate research area of “data mining” is included. Finally, a possible procedure for text mining is described with the individual steps which are run through.



### 2.2.1 Introduction and Definition

Text mining is of particular importance, as text is the most natural way of storing information. It has been recognized more than two decades ago that 80% of information from companies is available in text form (Tan, 1999). Text mining deals with finding information in textual data. They are usually available in a large collection of text and are only weakly structured or have no structure at all. Within the data, the goal is to find patterns that yield new and useful information in order to generate new knowledge. Often, there is also great interest in relating documents to each other (Gupta and Lehal, 2009).

Text mining mainly uses techniques from the fields of information retrieval, machine learning, statistics, data mining, and computational linguistics. In many cases, the aforementioned KDD process model is followed by defining various steps that then enable the processing of extensive text data. However, one of the main challenges in this area is the lack of structure in the textual content. Therefore, pre-processing becomes more important in text mining. Software that applies text mining methods need to consider extensive possibilities for pre-processing and typically focuses on the extraction of information. Furthermore, the processing of the found information and the presentation for the user plays an important role (Hotho et al., 2005).

The goal of text mining is to find new, previously unknown knowledge in texts. It can therefore be regarded as a special form of data mining. The main difference between data mining and text mining is that the data to be analyzed are available in a structured form. Data are stored in databases in a normalized form. Texts, available as collection of text or text corpus, however, are considered unstructured (Hippner and Rentzmann, 2006). A closer look, nevertheless, reveals that texts are not completely without structure. The grammar used thus forms an implicit structure of the text (Feldman and Sanger, 2007; Hippner and Rentzmann, 2006). Feldman and Sanger (2007) further differentiate this into a semantic and a syntactic structure. Furthermore, there is also an explicit structure in texts (Hippner and Rentzmann, 2006). It is formed by punctuation marks, paragraphs, and headings (Feldman and Sanger, 2007; Hippner and Rentzmann, 2006).

In addition to the term text mining, there are a number of other terms used in literature. An overview can be found in (Mehler and Wolff, 2005). Different terms determine the respective view on text mining (Sharafi, 2013). Feldman and Dagan (1995) speak of knowledge discovery in textual databases. The term is strongly based on the process model of the KDD. It focuses on the extraction of information, and the categorization of the texts according to their topics. Hahn and Schnattinger (1998) use the term text knowledge engineering to generate new, domain-specific knowledge. Kodratoff (1999) also bases his definition of Knowledge Discovery from Text (KDT) on that of the KDD. Losiewicz et al. (2000) use the term textual data mining. One focus of his work is the use of metadata. The similar term of text data mining is used by Merkl (1998). It defines its focus on the classification of texts. The term is also used by Hearst (1999).

Text mining uses methods from the NLP, information retrieval, information extraction and AI (Hippner and Rentzmann, 2006). Depending on the specific application area, similar terms are used (Mehler and Wolff, 2005). As mentioned before, due to the influences of different research directions, it is difficult to formulate a universally valid definition. From the information extraction point of view, text mining is described as extracting facts from text (Hotho et al., 2005).

In text mining, less structured text documents are initially present (Hippner and Rentzmann, 2006). For this reason, a pre-processing with procedures of NLP is necessary in the first step. Subsequently, data mining methods are often applied on the results of text mining. The necessary steps to implement an application with text mining are shown in Figure 2.2.

### 2.2.1.1 Differentiation from data mining

The term data mining generally refers to the extraction of knowledge from data. Strictly speaking, text mining is a subordinate field of data mining, but it differs in particular with respect to the data basis. These are often large volumes of data with numerous data records, which can differ in their respective structure, but are basically structured. As in data mining, the goal from text mining is to find specific patterns within the underlying data that can be extracted as usable knowledge.

As a result, techniques from various scientific disciplines are also used in this area, ranging from pattern recognition and statistics to AI. Over time, the term “data analytics” and “data science” have become established, especially with regard to the computer-aided processing of large amounts of data. In addition, data mining as a term has been expanded over time to include the KDD process (see Section 2.2.1), which also occurs in text mining. In addition to the actual data mining, which is the analysis, it also includes processes that realize a selection, pre-processing, transformation, interpretation, and evaluation of the corresponding data.

### 2.2.2 Process of Text Mining

Text mining can be understood as an iterative process. Hippner and Rentzmann (2006) describe the process of text mining in six steps. It is similar to the data mining process in Section 2.1.2. However, data pre-processing is of particular importance. The process is shown in Figure 2.2.

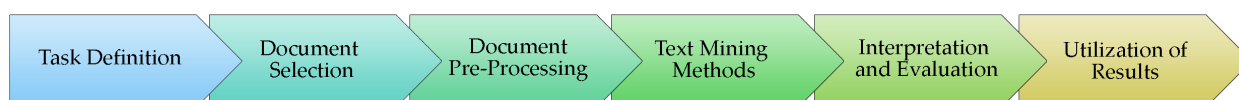


Figure 2.2: Text Mining Process (Hippner and Rentzmann, 2006)

The individual process steps according to Hippner and Rentzmann (2006) are

1. **Task definition:** The task to be completed must be described. Goals must be defined how the set task is to be solved.
2. **Document selection:** It is necessary to identify the documents that may be relevant. It is important to ensure that the documents are aligned with the previously defined objectives.
3. **Document preparation:** This process step is of particular importance for the text mining process. Hippner and Rentzmann (2006) specifically mention the splitting of documents into tokens as a task. These tokens give the text a structure and are required for further processing of the texts.
4. **(Text) Mining Methods:** Data mining methods can be applied on the tokens. In this way, a classification is possible in which the texts are assigned to previously defined classes. Within the segmentation it is possible to group similar texts. In a dependency analysis, words that occur together are examined more closely.
5. **Interpretation and evaluation of the results:** Relevant results must be analyzed.
6. **Application of the results:** The knowledge gained must be applied.

Basically, the process steps 1-6 are used in text mining, which behave similarly to data mining, but have differences, especially in the overall phase of data preparation and pre-processing. At the beginning, it must be clarified which goals are to be achieved at all and what the problem definition looks like. This analysis is recorded in a task definition and clarifies which text mining

approaches are used to solve the problem. The definition of the goals developed in the first step, serves thereupon for the selection of the documents. All possibly relevant documents are first collected in a corpus. Since at the end often very extensive collections of texts result, a certain similarity regarding language, size and topic can serve as a criterion.

The documents are then prepared for transfer into a standardized format. Using NLP methods, the document is broken down into its most important components, the so-called tokens (see Section 3.1.1). These can be individual words, word stems or selected word phrases, as this depends largely on which further processes are to be used.

After the preparation, the analysis follows as one of the main processes. For this step, there are different text mining methods, many of which are similar in their approaches. Frequently encountered are, for example, similarity analysis, classifications, and segmentation. The results from these methods are often vectors or clusters. The results of the text mining method have to be interpreted or evaluated afterwards. On the one hand, the existing knowledge in the texts has to be made explicit, and on the other hand, if possible, relationships between the individual documents should be made visible.

In the end, the results are presented in an application, which takes up a not insignificant part of text mining. Depending on the problem, the user must be provided with tools that enable visualization and search in the results.

## 2.3 Web Mining

This section deals with Web Mining. First, a definition and an introduction to the topic is given. The second part of the section explains the process of Web Mining.

### 2.3.1 Introduction and Definition

The term “web mining” was first mentioned in 1996 by Etzioni (1996) (Chen and Wei, 2010). There are also different terms for web mining. Besides the term “web mining”, the term “web data mining” is also common. Both terms are used synonymously (Singh and Singh, 2010). Furthermore, the terms “Internet data mining”, “web knowledge discovery” and “web information mining” can be found (Chen and Wei, 2010). Singh and Singh (2010) see web data mining as a special form of data mining. They define web data mining as the application of data mining techniques to find interesting and potentially useful knowledge on the WWW (Singh and Singh, 2010). According to Kosala and Blockeel (2000) the following problems exist in the use case of web mining from the WWW:

- **Finding relevant information:** Relevant data and information must be found on the WWW, which is made difficult by the large amount of available information. Even simple search engine queries return large amounts of data.
- **Generate new knowledge from available information:** This is a sub-problem of the above. In contrast to scraping and storage of data, this is a matter of data quality.
- **Personalization of the information:** When people use the WWW, they prefer different forms of content and presentation.
- **Learning from consumers and individual users:** Modern websites adapt the information output to the behavior of the user. Which leads to additional challenges for mining because of changing structures.

### 2.3.1.1 Taxonomy of Web Mining

Web mining can be subdivided into three sub-sectors: *Web Content Mining*, *Web Usages Mining* and *Web Structure Mining* (Singh and Singh, 2010). An overview is given in Figure 2.3.

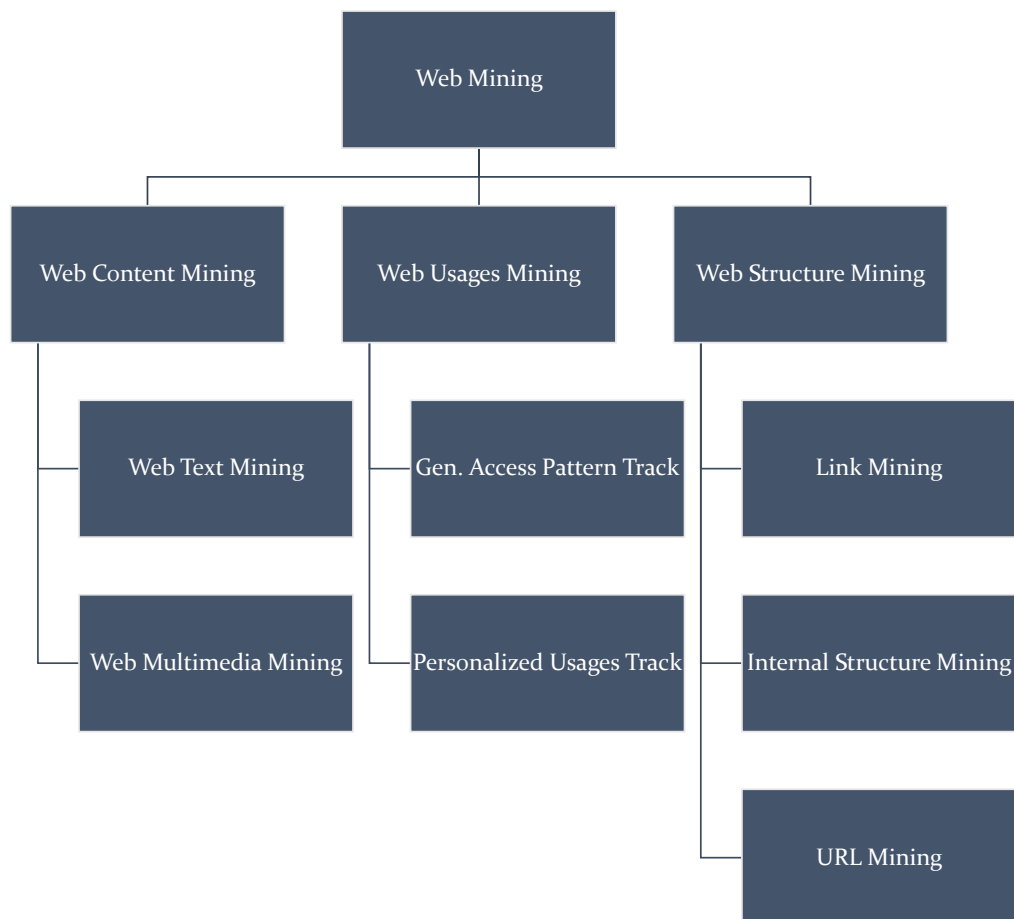


Figure 2.3: Web Mining Taxonomy, adapted from (Singh and Singh, 2010).

Web Content Mining focuses on the extraction of information and knowledge from the WWW. It can in turn be subdivided into Web Text Mining and Web Multimedia Mining. Web Text Mining is used to process and extract data in text form, whereas Web Multimedia Mining focuses on other WWW content such as images and videos (Kosala and Blockeel, 2000; Singh and Singh, 2010). The second major sub-area of web mining is web usages mining. This area is about the behavior of users in the WWW. To analyze the behavior of users of websites, log files on the servers are evaluated. The results should help to make the websites more user-friendly and lead to clearer structures (Kosala and Blockeel, 2000; Singh and Singh, 2010). Web Structure Mining is the third area of Web Mining. In this field, the focus is on the inner and outer structure of a website. In order to analyze it, the hyperlinks contained in the source code are considered. Among other things it is looked at whether web pages are only linked in one direction or mutually refer to other web pages. This area is of high interest in social network analysis (Kosala and Blockeel, 2000; Singh and Singh, 2010).

### 2.3.2 Process of Web Mining

The process of web mining is shown in Figure 2.4. It can be divided into five main steps and can be described as follows (Kosala and Blockeel, 2000; Singh and Singh, 2010; Zhang and Segall, 2008):

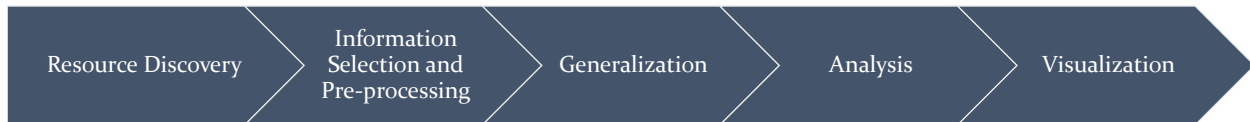


Figure 2.4: Web Mining Process

1. **Resource discovery:** The resources must be accessed and processed. They can be available online as well as offline. In addition to websites, resources can include, for example, e-mail and text databases.
2. **Information selection and pre-processing:** Kosala and Blockeel (2000) describe this step as an understanding of any transformation of the data. This includes in particular the methods of NLP.
3. **Generalization:** An attempt is made to find patterns on and between websites and resources.
4. **Analysis:** The patterns found must be validated and interpreted.
5. **Visualization:** Visualization is recommended for easy communication of the analysis results.

## 3 Natural Language Processing

NLP is a field of computer science, AI and computational linguistics that deals with the machine processing of natural (human spoken and written) languages (Carstensen et al., 2009). An essential task of this research area is to make linguistic patterns explicit. Thus, they form the basis for computing systems whose goal is to understand and produce language. Machine language processing is of great economic importance and is used in numerous different fields of application, for example in communication with machines via voice commands or in computer-aided translation and word processing (Jurafsky and Martin, 2014). For this dissertation, only the written variant of natural language is of interest.

NLP offers many possibilities how computers can facilitate the work with natural language texts and repetitive and labor-intensive processes can be automated. Computers can thus, for example, check texts for spelling or translate them into another language (Carstensen et al., 2009).

Natural language text data has the disadvantage that it is unstructured, or at least, lacks a structure that a computer can process automatically and efficiently. Thus, in order for computers to recognize more than just a string of characters, structures such as individual words and phrases, as well as entire sentences, must be transformed into a more understandable form by appropriate methods, so that the components of text data can be used in subsequent procedures.

It was already shown in Section 2.2.2 that document pre-processing is of outstanding importance for the text mining process. Among others, NLP operations are performed among this document pre-processing. The NLP has developed many methods to process texts. Which methods should be used depends on the specific task. Applying NLP methods lead to an increase in quality when using text mining (Kaur and Gupta, 2013).

According to Hippner and Rentzmann (2006), the methods of NLP can be divided into three groups. These are morphological, syntactic, and semantic analysis. Morphology deals with the structure of words. Every word is based on a root word. However, a root does not always have to represent a valid word. Moreover, the meaning of a word depends on the affixes used. To manipulate grammar, time and number, there are inflection forms (Witte et al., 2006). Tokenizing, removal of stop words, POS tagging, and also lemmatization are among the frequently used methods of NLP (Sharafi, 2013). Methods from the field of syntax include the formation of sentence structures. In order to be able to recognize words, phrases or sentences in text data algorithmically, various segmentation methods are used (Carstensen et al., 2009).

### 3.1 Pre-Processing of Documents

Since text mining is mainly based on unstructured data with little or no structure, pre-processing is an essential factor at the beginning of the analysis. Depending on the analysis method used later on, the individual steps can vary, however, they must always be selected appropriately, because pre-processing breaks down document structures and breaks the document down into its individual components.

Figure 3.1 shows a possible workflow for pre-processing of documents.



Figure 3.1: Sequence of pre-processing of documents, adapted from (Vijayarani et al., 2015).

After selection of relevant documents, normally the document structure is analyzed. Headlines, paragraphs, and formatting are valuable to distinguish between major parts in a document. The operation of sentence splitting handles the separation of the documents into smaller chunks. Typically, punctuation is used to perform the operation. In the identified sentences, it is advisable to start with the recognition of proper names as early as possible since these have the greatest significance for a document. If the proper names are not recognized in an early step of pre-processing, they could be distorted (Landmann and Züll, 2004). As the structure of the document is not yet fully broken down, and sentences are preserved, a POS tagger that assigns each word to a grammatical category is suitable for this purpose (Kübler and Maier, 2013). In the course of this, tokenization also takes place, in which the document is broken down into its individual words. Then, the so-called stop words can be removed. These are words that have little or no meaning and are therefore not important for further analysis. Stop words include articles, prepositions, conjunctions and other frequently occurring words of the respective language (Weiss et al., 2010). Often one removes the stop words with the help of a list, which is suitable for the respective language. Likewise, one can also work with a list when stemming (stem form reduction) the words. Here, each word is to be traced back to the root word. For example, if one finds the word “ran”, one translates it back into “run” (Jivani et al., 2011). However, also in this step specialties of the analyzed language have to be considered, which is why the results depend significantly on the method used. Stemming can also mean a great loss of information, since information about the tenses of the verbs is lost, for example.

Each step of the pre-processing of documents will be introduced in the following.

### 3.1.1 Tokenization

In tokenization, the text is divided into individual units, the so-called tokens. Manning et al. (2010) describe a token as a sequence of characters as a useful semantic unit. A simple form of tokenization consists of removing punctuation marks and whitespace between words and separating the text at these locations. Thus, the resulting tokens often correspond to words. However, tokens can also consist of several words or combination of words. This is important if the meaning of phrases should be captured correctly (Manning et al., 2010). For example, “text mining” is a possible token. The meaning of the 2-gram would be lost by a further splitting into “text” and “mining”. It should be further noted that tokenization always depends on the language used (Manning et al., 2010). In addition, the simple removal of punctuation creates many potential errors. Thus, punctuation is often used to increase the readability of numerical values. This is the case, for example, with monetary amounts (1,234.56 EUR) or IP addresses (127.0.0.1). Splitting the text at the points of the punctuation destroys the semantics (Manning et al., 2010; Sharafi, 2013). Problems can further arise here, for example, if abbreviations or dates are included in the text and are not treated differently by the tokenization algorithm (Carstensen et al., 2009).

### 3.1.2 Stop Word Removal

Stop words are words that have little significance for the analysis of a text (Sharafi, 2013). Words that occur frequently in a text can be counted among them. To determine these, the term frequency is often considered. This metric is created by counting the number of appearances of the individual words in a text. The stop words are removed from the text after the determination. This significantly reduces the number of remaining words. The stop words determined in this way are often articles, conjunctions, etc. (Manning et al., 2010). However, words that occur particularly rarely in a text can also count as stop words (Sharafi, 2013). Pre-defined stop word lists can also be used to remove stop words (Manning et al., 2010). The removal of stop words is not unproblematic. Stop words often take over an important function in phrases (Manning et al., 2010).

### 3.1.3 Lemmatization

Lemmatization is about reducing the number of word forms (Manning et al., 2010). Lemmatization attempts to bring nouns into the singular form and verbs into the basic form. The resulting form is called a lemma. The lemmatization of a text can be a very time-consuming task (Sharafi, 2013). Lemma does not necessarily mean the root of the word, but e.g., the infinitive form of a verb. In German language, lemmatization is problematic because of many irregularities, which is why lexicon-based approaches are more often preferred. In such an approach, for example, one can look up from a list of possible corresponding word forms. However, the list can become very long (Witte et al., 2006). Lemmatization has not been applied in the implementation of this dissertation.

### 3.1.4 Stemming

In stemming, the words are reduced to a common stem. The word stem is a group of words that have comparable meanings. Often this word stem does not exist as a used word in the respective language (Sharafi, 2013). In stemming, words are often reduced to their common beginning of word forms. How the root word actually looks like is determined by the algorithm used. Stemming has not been applied in the implementation of this dissertation.

## 3.2 POS Tagging

In POS tagging, the respective words are marked with their POS (noun, verb, adjective, etc.). To achieve this, POS taggers can use two sources of information. On the one hand, they evaluate lexicons where words and their possible word types are stored. On the other hand, syntagmatic information is used. The most probable word type is determined from a sequence of common word types. For this, the respective context has to be considered. Since words can have more than one POS, Hippner and Rentzmann (2006) describe POS tagging as non-trivial.

The POS tagging is the third step in the NLP process after tokenization as shown in Figure 3.1. However, POS tagging is not a mandatory text mining operation (Sharafi, 2013). Not every text analysis needs precise information about the distinctions of word forms. Often the simple occurrence of words is enough.

In POS tagging, each token of a sentence is marked with a POS tag to determine the word type without information about the content of the text, i.e., to decide whether a word is a noun, verb or adjective, for example. For a suggested implementation, the appropriate POS tags are appended to the respective words and separated from the word by visual indication, e.g., a slash ("/"). However, ambiguities can cause errors, e.g., if it is not clear whether a word should be interpreted as a noun or a verb.



The set of tags which can be distinguished by the POS tagger is called a tagset. Examples of tagsets for the English language are the Penn Treebank (Marcus et al., 1993). It has a corpus of about seven million words that have been assigned with the appropriate POS (Taylor et al., 2003). 36 tags plus special characters are included in the tagset. For the German language, the Stuttgart-Tübingen Tagset (STTS) (Schiller et al., 1995) is widely used. It is a merger of two POS tagsets developed at the University of Stuttgart and the University of Tübingen. In total, the STTS has 54 tags.

### 3.2.1 Sentence Parsing and Chunking

Sentence parsing is a grammatical analysis method with the goal of mapping the grammatical structure of the sentence into a tree structure. Typically, the root node of the tree represents the analyzed sentence. Starting from the root node, grammatical structures are mapped in the form of leaves. Leaves at the lowest level represent the individual words of the sentence.

Chunking of groups from grammatically related words into phrases are called constituents. The most important constituents are the nominal phrase (NP) and the verbal phrase (VP). An NP contains a head, namely a noun, and optional modifiers and determiners. Modifiers usually represent adjectives and determiners are articles (Witte et al., 2006). Accordingly, the head of a VP is a verb.

## 4 Knowledge Extraction From Text

In this chapter, the relevant theoretical background of the extraction methods for knowledge acquisition will be presented. In different sections, methods are explained individually and possible variants which are used in the course of the dissertation are introduced.

### 4.1 Information Retrieval

#### 4.1.1 Term Frequency - Inverse Document Frequency

One of the most commonly used approaches for information retrieval is the TF-IDF measure (Manning et al., 2010). It is a combination of the Term Frequency (TF) and the Inverse Document Frequency (IDF). The TF  $tf_{t,d}$  indicates the frequency of the term  $t$  in document  $d$ . The order of the terms in a document is not relevant in this approach. It is therefore also called a bag-of-words model. The IDF formula (4.1) does not consider a single document, but a corpus of documents with total number of  $N$  documents (Manning et al., 2010).

$$idf_t = \log \frac{N}{df_t} \quad (4.1)$$

In the IDF formula (4.1), the document frequency  $df_t$  is defined as the number of documents in the corpus containing term  $t$ .

The TF-IDF measure (4.2) weights a term  $t$  in document  $d$  highly, if it occurs frequently in a small number of documents. If a term occurs less frequently in the document or appears in many documents, it is given a lower weighting (Manning et al., 2010). The terms are therefore less characterizing for the respective documents.

$$tf-idf_{t,d} = tf_{t,d} \cdot idf_t \quad (4.2)$$

#### 4.1.2 Word Association

Detecting semantic relationship between individual word forms that occur in a text is a key challenge to text mining (Heyer et al., 2006). Two word forms that occur in close textual proximity due to frequent co-occurrence provide an important clue that such a semantic relationship between the two word forms is present. A necessary condition for a semantic relationship between two word forms to be recognized is that these word forms stand in syntagmatic relation to each other. These two word forms are only in a syntagmatic relation to each other if there is at least one local context that contains both word forms. The local context of a word form is the set of all word forms which appear together with this word form in a common sentence. Since sentences form semantic units, it makes sense to examine them for the co-occurrence of word forms. These word forms are also called sentence co-occurrences. In addition to co-occurrence in a sentence, it is necessary for sentence co-occurrences to co-occur in a sufficient number of local contexts, which ensures that these word forms are not merely coincidentally related syntagmatically but are recognized as significant co-occurrences by a statistically relevant frequency of co-occurrence. Significant co-occurrences are in a statistically syntagmatic relation to each other, which means that their co-occurrence with respect to a significance measure is not random (Heyer et al., 2006).

### 4.1.2.1 Significance measures

If one wants to determine how significant a syntagmatic co-occurrence of word forms is, it is necessary to be able to evaluate it mathematically and to assign it a number that provides information about its statistical relevance. This statistical quantity is called significance. Evert and Krenn (2001) use significance as a measure of lexical association, with the significance measure reflecting someone's intuition about how much the two terms belong together.

Co-occurrences are considered significant if their significance value exceeds a certain threshold. It is not easy to choose the right threshold, because it can vary depending on the corpus. If the threshold is set too low, even random co-occurrences will be falsely recognized as statistically syntagmatically related word forms. If the threshold is set too high, there is a risk of overlooking good quality co-occurrences and not recognizing them as such. Another way to identify significant co-occurrences is to select a set of the best  $n$  co-occurrences at a time. However, there is also a risk of setting the  $n$  too high or too low. Particularly in the case of fully automated procedures, it is unavoidable to make compromises. To determine the significance value of a co-occurrence of word forms  $A$  and  $B$ , four quantities are available:  $a$ : number of sentences in which the word  $A$  appears,  $b$ : number of sentences in which the word  $B$  appears,  $k$ : number of sentences in which both word forms appear together,  $n$ : Number of all sentences of the corpus.

Using these values, the significance value can be calculated with the help of a weighting function. Various measures are available in (Heyer et al., 2006):

Tanimoto coefficient:

$$sig(A, B) = \frac{k}{(a + b - k)} \quad (4.3)$$

Dice coefficient:

$$sig(A, B) = \frac{2k}{(a + b)} \quad (4.4)$$

Transinformation:

$$sig(A, B) = \log \frac{n \cdot k}{a \cdot b} \quad (4.5)$$

Log-likelihood-Function:

$$\begin{aligned} sig(A, B) = & n \cdot \log(n) - a \cdot \log(a) - b \cdot \log(b) - k \cdot \log(k) + (n - a - b + k) \\ & \cdot \log(n - a - b + k) + (a - k) \cdot \log(a - k) + (b - k) \cdot \log(b - k) \\ & - (n - a) \cdot \log(n - a) - (n - b) \cdot \log(n - b) \end{aligned} \quad (4.6)$$

Poisson distribution:

$$sig(A, B) = \frac{\log(k!) - k \cdot \log\left(\frac{a \cdot b}{n}\right) + \frac{a \cdot b}{n}}{\log(k)} \quad (4.7)$$

T-Score:

$$sig(A, B) = \frac{-a \cdot b}{\sqrt{\frac{k^2}{n}}} \quad (4.8)$$

These equations all provide some measure of significance for the co-occurrence of the word forms  $A$  and  $B$ . Higher significance values represent a stronger correlation between the two word forms.

### 4.1.2.2 CIMAWA

In association analysis, co-occurrences, e.g., the occurrence of two words in a text window, or sentence, are considered. Word associations are to be determined on the basis of this common occurrence of words. A well-known example of such a word association is “bread” and “butter”. Such word associations can be used to identify topics in texts (Klahold et al., 2013; Uhr et al., 2013). A method for recognizing word associations and to imitate the human ability of association is the CIMAWA approach, developed by Uhr et al. (2013).

Classical approaches to the determination of word associations are either symmetrical or asymmetrical. The special feature of CIMAWA is that both methods are combined. A symmetrical value and an asymmetrical value are determined. The result of CIMAWA is the sum of both, whereby the symmetrical value is attenuated by a factor. CIMAWA can therefore be seen as a hybrid method (Uhr et al., 2013).

The association strength between two words can be calculated via CIMAWA as follows (Uhr et al., 2013):

$$CIMAWA_{ws}^{\zeta}(x(y)) = \frac{Cooc_{ws}(x,y)}{(frequency(y))^{\alpha}} + \zeta \cdot \frac{Cooc_{ws}(x,y)}{(frequency(x))^{\alpha}} \quad (4.9)$$

For the calculation, one first uses the two frequencies of the words  $x$  and  $y$ , which are specified with  $frequency(x)$  and  $frequency(y)$ . In addition, however, one also needs the co-occurrence, i.e., the value that indicates how often the two words occur together in a section of a certain size. For this purpose, a “window size” is defined, which is denoted by  $ws$ . If the window size is four, for example, it means for the algorithm, that within the two words to the left and right of the word  $x$ , the word  $y$  is searched for. A window size of ten words, would mean five words to the left and right side accordingly.

Depending on the application and the desired analysis, this value can vary. If CIMAWA is applied to single sentences, a smaller window size is helpful, while for long texts or paragraphs, higher values also yield good results (Uhr et al., 2013). The damping factor  $\zeta$  takes values from 0 to 1 and is equally dependent on the goal of the analysis. If you set it to 1, CIMAWA behaves symmetrically and you can swap  $x$  and  $y$  without consequences. If, on the other hand, the value is set to 0, CIMAWA works completely asymmetrically as the second part of the formula no longer has any meaning.

The best results for  $\zeta$  are obtained with a value of 0.4 to 0.6, as studies with a corpus of 2.8 billion words have shown (Uhr et al., 2013). The result of  $CIMAWA_{ws}^{\zeta}(x(y))$  ultimately indicates how strong the meaning of the word  $x$  is in association with the word  $y$  based on the selected text window  $ws$  and the dampening factor  $\zeta$ . Of particular interest is the word  $y$  that reaches the highest value in association with  $x$  as it has the strongest association strength.

CIMAWA was also combined with a case-based reasoning approach and used for textual information recommendation (Nasiri et al., 2015), (Nasiri et al., 2017). In a learning assistant system, various keywords from medical literature and patient info were evaluated by CIMAWA and used as word association profiles.

## 4.2 Named Entity Recognition

Named Entity Recognition (NER) is a method for recognizing entities with proper names such as persons, places, locations, or organizations (Hotho et al., 2005). NER is especially important for semantic relationship extraction since proper names are an important part of entity relations. The simplest implementation for NER is using a list of known names of persons, places, or organizations (Carstensen et al., 2009).

In contrast to this very much simplified approach, there are procedures that use Machine Learning (ML) techniques. The goal of such NER systems is to learn the ability to recognize previously unknown entities. A distinction is made between supervised, semi-supervised and unsupervised learning methods, which differ, among other aspects, in the completeness of the available training data sets.

According to Nadeau and Sekine (2007), supervised methods require a large number of prepared documents giving positive and negative examples of expressions of the distinguishing characteristics, to automatically derive rules and recognize entities of a type.

Nothman et al. (2013) propose an approach to learn multilingual named entities from Wikipedia articles.

Semi-supervised methods require the user to enter a few examples to start the learning process.

In unsupervised procedures, there are accordingly no training examples available. Clustering is a frequently used approach (Nadeau and Sekine, 2007).

NER is a subfield of information extraction. The goal is the automatic entity recognition from unstructured text documents (Feldman and Sanger, 2007; Klahold, 2009).

The term NER was first mentioned at the Message Understanding Conference 6 (MUC-6). The task was to identify proper names from texts and assign them to the following entity types: Person, Organization, Place, Date, Time, Money, and Percent (Grishman and Sundheim, 1996). The identification of these entity references is also called Named Entity Recognition and Classification (NERC) because it identifies entities in the first step and classifies them in the second step.

At MUC-7 conference, the task definition of the NER was formulated in more detail and divided into the following sub-parts (Chinchor and Robinson, 1997; Nadeau and Sekine, 2007): Named entities include the components Organization, Person, and Location. Organization includes all businesses, government entities, or other organizational entities. All proper or family names are in the Person class and Location includes cities, provinces, countries, and mountains, i.e., names of a politically or geographically defined place.

According to Klahold (2009), two other variants for the realization of NER can be found in the literature. On the one hand there is the rule-based variant and on the other hand the list-based variant. Both variants have the goal to recognize named entities. While the rule-based variant uses grammatical, statistical, or other information to compare whether a word or word tuple is a named entity, the list-based variant uses directories or lexicons of named entities to compare words and word tuples. There are also mixed forms of both variants (Klahold, 2009).

### 4.2.1 Methods

Previous studies show that three different learning methods can be distinguished to recognize named entities. At the beginning of the research in the field of NER, the focus was on entity recognition using manually generated rules. In the course of time, the supervised machine learning method has become widely accepted and established. Using large annotated document collections, also known as corpus, positive and negative examples are used to study the properties of entities and to establish rules for recognition. If there is no corresponding corpus to base on, semi-supervised learning or unsupervised learning are used (Nadeau and Sekine, 2007).

#### 4.2.1.1 Supervised learning

The most commonly used method is the supervised learning approach. Methods have the following common features (Nadeau and Sekine, 2007): (1) reading in a large annotated corpus, (2) storing a list of entities, (3) creating disambiguation rules based on exclusionary properties.

Therefore, the often-suggested supervised learning method is tagging the entities and entity-related words in a corpus as training corpus. The performance depends on the transfer of the

vocabulary words that occur in both corpora. This vocabulary transfer, which indicates the ratio of words without repetitions, also serves as an indicator of the basic method used. The ratio of the number of recognized entities to the total number of entities is calculated (Nadeau and Sekine, 2007). Using the MUC-6 training data, for example, Palmer and Day (1997) calculated a transfer of 21%. This reflected 42% of place names, 17% of organizations, and 13% of proper and family names. Mikheev et al. (1999) calculated 76% for place names, 49% for organizations, and 26% for proper and family names on the MUC-7 corpus with a precision of 70% to 90%.

Statistical machine learning is the foundation in entity recognition based on supervised learning. It is considered as a sequence labeling problem like other NLP methods (e.g., POS tagging), and is a common problem in machine learning. Jiang (2012) formulates it as follows: The following observations are given, denoted as  $x = (x_1, x_2, \dots, x_n)$ . Each observation is represented by a feature vector. Each observation  $x_i$  is assigned a label  $y_i$ . Using the classification for named entities, the label for  $y_i$  in the sequence tag can be predicted based on  $x_i$ . It is assumed that the label  $y_i$  may depend not only on  $x_i$ , but other observations and labels. However, this refers only to the neighboring circle from position  $i$  (Jiang, 2012).

In order for the NER to be mapped as a sequence labeling problem, each word in a sentence is considered as an observation. It is important to note that the class labels uniquely specify not only the named entity boundaries, but also the types (Jiang, 2012; Ramshaw and Marcus, 1999).

**Hidden Markov Modell:** The Hidden Markov Model (HMM) owes its name to Andrey Andreyevich Markov, whose theory of the Markov process forms the basis for the model (Blunsom, 2004). Blunsom (2004) describes the model as a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process that generates an observable sequence. Jiang (2012) reasons that the Markov process, working with transition probabilities, fits the task of NER very well. This is because the best fitting designation sequence  $y = (y_1, y_2, \dots, y_n)$  for an observation sequence  $x = (x_1, x_2, \dots, x_n)$  is the one that maximizes the conditional probability  $p(y|x)$  or equivalently, the one that maximizes the joint probability of  $p(x, y)$ . This is done using the Markov process, where the generation of a label or an observation depends only on one or a few previous labels and/or observations (Jiang, 2012).

To obtain a hidden Markov model, all  $y$  have to be considered as hidden states (Rabiner, 1989). There are several systems that use the Hidden Markov Model and differ in their performance (Zhou and Su, 2002). One of the first systems is the Nymble system (Jiang, 2012). Developed in 1997, the system has been a joint development for the US-based BBN Technologies company (Bikel et al., 1998). Bikel et al. (1998) were able to obtain results with an accuracy level of 90%, which is close to human performance.

Exemplified by this system, the Hidden Markov Model can be described as follows. The system proceeds according to the following three steps:

1. By conditioning with the previous label  $y_{i-1}$  and the previous word  $x_{i-1}$ , each  $y_i$  is generated.
2. If the first word of a named entity is  $x_i$ , then it is generated by conditioning with the current  $y_i$  and the previous  $y_{i-1}$ .
3. However, if  $x_i$  is inside a named entity, then it is generated by conditioning with the previous  $x_{i-1}$  observation.

The process is repeated until the sequence under consideration is completely generated (Bikel et al., 1998). Feldman and Sanger (2007) point out that most systems and applications aim to solve these three problems: For the analysis of a given observation, first the probability distribution induced by the model is computed. Then, the most probable state sequence for a given observation sequence

is determined. Finally, to maximize the probability of the given observation the model is optimized (Feldman and Sanger, 2007).

#### 4.2.1.2 Semi-supervised learning

The semi-supervised or weakly supervised learning method requires very little supervision. The dominant technique is called “bootstrapping” (Nadeau and Sekine, 2007). Nadeau and Sekine (2007) compare it to “*a set of seeds that start a learning process*”. As an example, they state an approach which is specialized in “disease names” which prompts the user to provide example names. Sentences are then examined based on these examples, identifying contextual relations. This is followed by a learning process that is applied to the new examples to identify new and relevant contextual knowledge. Through repetition, more new names and context can be identified (Nadeau and Sekine, 2007).

#### 4.2.1.3 Unsupervised learning

Clustering is the typical basis for the unsupervised learning method. In this method, named entities are grouped together depending on the similarity of contexts. In addition, there are other non-supervised learning methods based on lexical resources, lexical patterns, and statistics.

### 4.3 Sentiment Analysis

This section discusses the theoretical foundations of sentiment analysis. It is first defined and an introduction to the topic is given. This is followed by an overview of the methods of sentiment analysis. Then the current challenges and problems of sentiment analysis are stated.

Sentiment analysis is a subject area of text mining (Medhat et al., 2014). Its purpose is to capture the sentiment, i.e., the feelings, opinions, emotions, or polarity contained in a text. This is mainly done by polarity detection (Kaur and Gupta, 2013). Therefore, sentiment analysis and polarity detection are partly equated (Cambria et al., 2013). Polarity can be divided into positive and negative polarity (Medhat et al., 2014). It is therefore a binary sentiment classification. An example of a further binary sentiment classification is the agreement detection. Cambria et al. (2013); Kaur and Gupta (2013), on the other hand, list a neutral polarity as the third possibility of polarity detection. In the literature, besides the term sentiment analysis, opinion mining is also common. Both terms are often used synonymously (Cambria et al., 2013). However, they are different approaches. Sentiment analysis determines the sentiment of a text and analyzes it. The core of opinion mining, on the other hand, is to capture opinions on entities (Cambria et al., 2013; Medhat et al., 2014).

Since Web 2.0, the importance of sentiment analysis has increased. Forums, blogs, social media, etc. are becoming more and more popular. Here, users write about their opinions and feelings on different topics that move or interest them (Bohlouli et al., 2015). The high value of the information contained is one of the main drivers for the development of sentiment analysis.

#### 4.3.1 Methods

Sentiment analysis can be applied on three different levels. Varghese and Jayasree (2013) distinguish between the document, sentence, and phrase level. With the document level the text of a document is considered as a whole. The complete document is thus assigned exactly one sentiment value. With the sentiment analysis on sentence level, each sentence of a text is assigned a sentiment value. This finer granularity makes it easier to detect changes in a text. The phrase level is

mainly about recognizing entities. The entities are then assigned a sentiment value. This level of sentiment analysis is therefore the most complex. However, it also has the greatest usefulness. For example, companies do not want to know the sentiments in any phrase, but rather the sentiments of customers about their own products (Varghese and Jayasree, 2013).

In order to reduce the amount of analyzed text, it is possible to classify the individual sentences as subjective or objective before the analysis. Objective sentences are irrelevant for the analysis and can be removed from the texts. Only the subjective sentences will be considered further (Varghese and Jayasree, 2013).

Other approaches in sentiment analysis are emotion detection, transfer learning through AI and building resources. The emotion detection is a task of sentiment analysis, which in contrast to the polarity detection, is not only divided into positive and negative. The aim is to detect concrete emotions. Medhat et al. (2014) name the eight emotions joy, sadness, anger, fear, trust, disgust, surprise, and anticipation.

The domain used plays a role in transfer learning. Here, the knowledge gained in one domain is used to improve the learning process in another domain. Building resources means the creation of resources that support sentiment analysis. For example, dictionaries and corpora are created in which polarities are already assigned to the individual terms for training (Medhat et al., 2014).

There are a number of methods that are used in the field of sentiment analysis. Basically, the methods of sentiment analysis can be divided into two broad areas. There are the machine learning approaches and the lexicon-based approaches (Medhat et al., 2014). Figure 4.1 provides an overview of these methods.

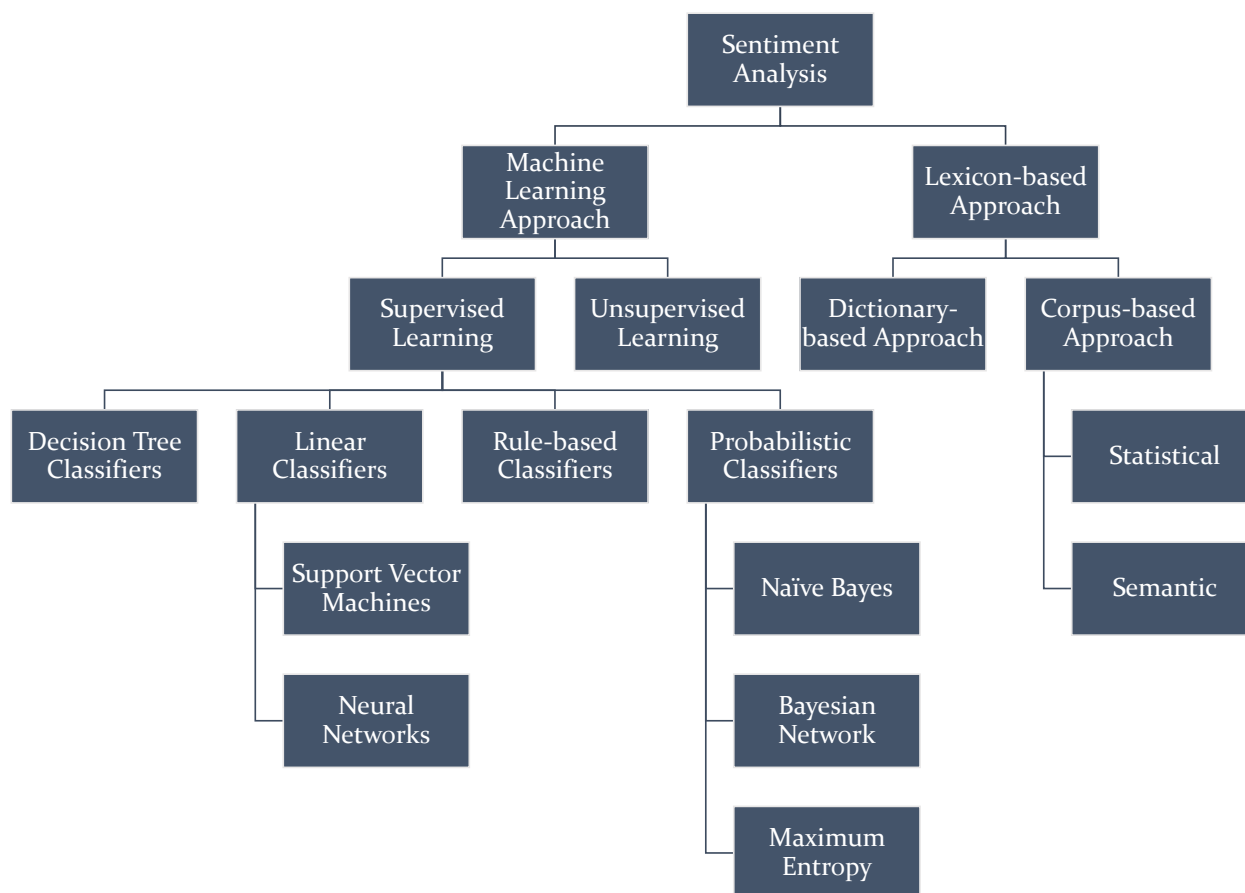


Figure 4.1: Sentiment Analysis Taxonomy, adapted from (Medhat et al., 2014).



The lexicon-based approach can be divided into the dictionary-based approach and the corpus-based approach (Medhat et al., 2014; Varghese and Jayasree, 2013).

The dictionary-based approach starts with words whose polarity is known. Synonyms and antonyms for these words are searched in word databases such as WordNet. The words found are added to the original list. The process is iteratively repeated until no new words can be added to the list (Medhat et al., 2014).

One strength of the corpus-based approach is the ability to find opinion words with contextual dependency. The corpus-based approach can in turn be divided into two areas, the statistical and the semantic approach (Medhat et al., 2014).

The statistical approach considers on the one hand the frequency of occurrences. If a word is used more often in positive texts, its polarity is positive. If it is used more often in negative texts, its polarity is correspondingly negative. If the word is used consistently, its polarity is neutral. On the other hand, similar opinion words are often used together in a text. If these words are identified, they can be assigned the same polarity (Medhat et al., 2014).

The semantic approach considers the semantic similarity between words. If two words have a strong semantic similarity, they are assigned similar polarities in this approach. The semantic approach can be combined with the statistical approach (Medhat et al., 2014).

With the machine learning approach, a distinction can be made between supervised and unsupervised learning. Supervised machine learning relies on the existence of evaluated test data to train the methods. Since it is often difficult to obtain suitable test data, unsupervised machine learning can be used. Here, for example, keyword lists and similarity measurements are used to classify the texts (Medhat et al., 2014). A detailed overview of the methods can be found in Medhat et al. (2014) and Varghese and Jayasree (2013).

#### 4.3.1.1 Associative Sentiment Analysis

An innovative method by Uhr et al. (2014) is to create a sentiment evaluation which combines the two concepts keyword recognition and word association. CIMAWA measures the association between words (Uhr et al., 2013). It was designed to simulate the Human Word Association (HWA), based on large collections of text documents (Uhr et al., 2014).

The CIMAWA describes the strength of association between two words  $x$  and  $y$  by taking into account the co-occurrences ( $CIMAWA_{ws}(x(y))$ ) in a predefined text window ( $ws$ ). The damping factor  $\zeta$  is used to balance the first asymmetric and second symmetric part of the hybrid association formula (Uhr et al., 2013). By solving the equation for different word combinations all associations can be created. Ideally, all associations are calculated, represented, scaled, and ordered by the numeric CIMAWA values from a selected document corpus. As a measure for sentiment analysis in a predefined time period, the concept and equation of CIMAWA, originally developed by Uhr et al. (2013), was adapted (Uhr et al., 2014) to be more flexible to changes of word associations and sentiment over the time:

$$CIMAWA_{T,ws}^{\zeta}(x(y)) = \frac{\sum_{t=1}^T [(\frac{Cooc_{ws}(x,y_i)}{(frequency(y_i))^{\alpha}} + \zeta \cdot \frac{Cooc_{ws}(x,y_i)}{(frequency(x))^{\alpha}}) \cdot \theta_{y_i}]}{n} \quad (4.10)$$

All co-occurrences ( $Cooc_{ws}(x, y_i)$ ) have a subindex  $i = 1, \dots, n$  at keyword  $y_i$  for an increment operation. Word  $x$  is a fixed term in the calculation process and will not be changed. The increment operation only counts if  $y_i$  (inside a text window size ( $ws$ )) is present in a resource ( $\Theta$ ) (e.g., a word list), with evaluated entries ( $\theta$ ) and finishes at  $n$ . For the damping factor values in the interval  $[0.4 - 0.6]$  achieved best results in previous studies (Klahold et al., 2013). Therefore, a damping factor  $\zeta$  of 0.5 is often used.

According to Uhr et al. (2014), the formula 4.10 is applied in temporary changing content. With timestamp information ( $t$ ), all sub periods (starting with  $t = 1$ ), combined in a maximum time window ( $T$ ) are observed. All calculated CIMAWA associations are multiplied by the valuated wordlist scale  $\Theta$ , with value entry  $\theta_{y_i}$  if keyword  $y_i$  is included. Finally, the sum of all multiplication results is divided by the last increment value  $n$ .

To create a dynamic sentiment score, this calculation is performed for each document in a document corpus with timestamp information. The time window ( $T$ ) moves forward and examines positive and negative associations in small portions of the entire text corpus as only historically relevant documents (based on timestamp) are considered. The advantage of this extension is the detailed historical review over time and the determination of the changing strength of word associations combined with the fast-processing concept of detecting relevant keywords or entities to filter the results. Fast or slow changing sentiment scores can be easily created by adjusting the time window, similar to a moving average. (Uhr et al., 2014)

In the dissertation a time window of three months is applied for the short-term associative sentiment. The long-term associative sentiment value is calculated considering all keyword occurrences in the corpus.

### 4.3.2 Challenges and problems

Varghese and Jayasree (2013) describe a number of challenges in sentiment analysis. Among these challenges is the recognition of named entities through NER. These are special noun phrases that refer to entities, such as persons or products. The goal is to find all mentions of named entities in the text. The named entities should then be sorted out of the text (Varghese and Jayasree, 2013). By Varghese and Jayasree (2013), the example is well-known: *“The Canon Power Shot is a great camera for beginners. It is easy to use and of high quality”*. In this example the “Canon Power Shot” is a named entity. The actual entity is the camera.

The resolution of co-references is a challenge, which is especially important when analyzing texts on the phrase level. Co-references are linguistic expressions that reference the same entity. Finding co-references is important to find all text passages that refer to the same entity. Otherwise, the result of the sentiment analysis would be distorted. Co-references can occur in one sentence as well as in different sentences.

In the example, “Canon Power Shot” and “It” are co-references (Varghese and Jayasree, 2013). Furthermore, aspect-oriented sentiment analysis is often used to distinguish between different properties, attributes or characteristics of objects and entities (Broß, 2013), (Schulz et al., 2018). Varghese and Jayasree (2013) describe the determination of relations as a further challenge. The syntactic relations between the individual words in a sentence are to be analyzed first. The semantics of a sentence can then be determined based on the syntactic relations. Thus, finding the relations between parts of the sentence is also a research area of NLP.

The domain dependency is another challenge of sentiment analysis. Sentiment classifiers that produce good results in one domain through training data can still produce bad results in another domain. This is because people use different terms to express opinions in different domains (Varghese and Jayasree, 2013).

## 4.4 Topic Detection

This section provides information dedicated to the methodology of topic detection. In a first section, the topic detection is defined and an introduction to the topic is given. Then, methods of topic detection are discussed in detail. Finally, problems and challenges of topic detection are pointed out.

Topic detection is another major sub-field of text mining. The aim of topic detection is to automatically determine the topic of a text. The texts should be assigned to the respective topic. Therefore, Amayri and Bouguila (2013) speak of a clustering problem when it comes to topic detection.

Topic detection is a field of research of the so-called Topic Detection and Tracking (TDT) (He et al., 2010). TDT goes back to research projects in the early 1990s as a sub-area of information retrieval (Allan et al., 1998a). An essential difference to classical information retrieval is that there is no previously defined need for information. A topic in the sense of TDT must have a concrete reference to place and time. A generic term is not considered a topic (Stock, 2007). According to Stock (2007) a topic can also be more general or abstract. Since the topics are events in a narrower view, the terms topic and event are used synonymously (Sayyadi and Raschid, 2013).

Following the pilot TDT project, further evaluation improvements and additions followed. For the second phase TDT2, primarily a larger corpus was used, containing approximately 40,000 messages and 1,000 hours of audio material (Wayne, 1997). During the third phase, on the other hand, an expansion to include an additional language occurred as three sources of messages in Mandarin (Chinese) were added (Graff et al., 1999). The most recent version is TDT5 and dates from 2004 (Glenn et al., 2006).

Nowadays, however, the focus is increasingly on social media, where one possible question could be which topics are of particular concern to users and are therefore frequently discussed.

The TDT can be subdivided into the subject areas Story Segmentation, Topic Detection, Topic Tracking, First Story Detection and Link Detection (Fiscus and Doddington, 2002). The aim of story segmentation is to divide the texts into individual stories. A story always deals with exactly one topic. A story can also be a complete document. The task of Topic Tracking is to assign new texts to an already known topic. The First Story Detection determines for each story whether the topic of the story is already known. Otherwise, the story is to be considered the first of a new topic. The task of Link Detection is to determine the similarity between two stories. This can be seen as an auxiliary task since the results can be used for the other tasks (Lavrenko et al., 2002).

According to Allan et al. (1998a), topic detection can be divided into Retrospective Event Detection and Online New Event Detection. Besides Retrospective Event Detection, the terms Batch Topic Detection and Offline Topic Detection are also commonly used (Amayri and Bouguila, 2013; He et al., 2010). In this approach, texts are analyzed from a given database. Online New Event Detection is also known as Online Topic Detection (Amayri and Bouguila, 2013; He et al., 2010). In this method, texts from a news stream are analyzed in real time. Here, the focus is on the detection of new emerging topics (Allan et al., 1998b).

#### 4.4.1 Methods

In the literature, a number of methods can be found that are dedicated to topic detection. The individual methods follow different approaches and can be classified into application-oriented categories. These categories include cluster analysis, topic models and statistical approaches and will be discussed in the following.

##### 4.4.1.1 Cluster analysis

Cluster analysis is a classic approach to finding topics in texts. The procedure was already proposed by Allan et al. (1998a). The aim is to assign documents to a topic. For this purpose, similar documents should be grouped together. To ensure this, methods of information retrieval are used (Allan et al., 1998a).

**Term Frequency - Inverse Document Frequency:** The method has been introduced in Section 4.1.1. TF-IDF is often used for classification and clustering in topic detection through matching of extracted keywords to a topic dictionary or other keywords in a recognized cluster. It uses a lexical resource to map different keywords, expressions, or phrases to a pre-trained topic thesaurus.

**Vector Space Model:** The Vector Space Model (VSM) is a popular method for representing documents in clusters. A document is represented here as a vector. This indicates whether a term exists or not. Instead of the presence, the term frequency can also be specified (Manning et al., 2010). The individual vectors are then grouped using cluster analysis. According to Allan et al. (1998a), the cosine similarity is used to calculate the similarity. Thus, similar documents are sorted by topic, but without a concrete definition of the topic (Allan et al., 1998a). VSM is not further applied for topic detection in the course of this dissertation.

**Single-Pass:** Single-pass clustering is one of the best-known techniques and requires only one pass to divide a complete collection of documents into different clusters (Klampanos et al., 2006). One of the major advantages of this method is its comparatively simple implementation. The currently considered document is only compared with the already processed ones. For the comparison, one represents the document as a vector, for which the TF-IDF (4.2) measure is used. The cosine function, among others, can then be used to calculate the similarity (Huang et al., 2008). Of course, there are many other ways to calculate the similarity between two documents. Therefore, the scalar product or the Euclidean distance would also be conceivable.

$$S(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 \cdot \vec{d}_2}{\|\vec{d}_1\| \times \|\vec{d}_2\|} \quad (4.11)$$

The two vectors  $\vec{d}_1$  and  $\vec{d}_2$  consist of the words of the respective document and thus represent it. The smaller the distance between the vectors, the more likely these documents will be grouped into a cluster. In some cases, instead of  $\vec{d}_2$ , an average vector from a cluster is also used for calculation. However, a minimum value for the similarity must still be selected. If a document does not reach this value, a new cluster is created.

**AGF:** Another method for clustering is AGF developed by Klahold et al. (2013), which differs from other algorithms in one important point. As a result, clusters are also obtained, but these only contain individual keywords and not complete documents. The algorithm uses TF-IDF (4.2) measure and CIMAWA values from a document to represent a focus in the text with a cluster (Klahold et al., 2013). Thus, in principle, a so-called Multi Topic Detection (MTD) is possible, and the method learns if possibly more than one focus is contained in a text. At the beginning, according to AGF formula (4.12), a value is calculated for each word pair for which a CIMAWA value already exists.

$$AGF(word_1(word_2)) = \frac{CIMAWA_{ws}^c(word_1(word_2)) \cdot kr(word_1)}{kr(word_2)} \quad (4.12)$$

The two variables  $kr(word_1)$  and  $kr(word_2)$  are respectively the TF-IDF values of the two words. However, it is equally possible to calculate the meaning of a word in another way and use it in the formula. Subsequently, each keyword's highest AGF value and the respective associated word are stored in an AGF table. After that, the creation of the topic clusters can be started, which according to Klahold et al. (2013) proceeds as follows:

1. Create a new cluster and add the word pair with the highest AGF value. The entry for the word pair is deleted from the AGF table.
2. Add more words to the new cluster. a) Store both words from (1.) in a temporary queue. b) Search the AGF table for the first word in the queue. c) If the word was found, delete it from the queue, add the associated word to the cluster and queue, and delete the entry from the AGF table. If the word was not found, delete it from the queue and continue. d) As long as the queue is not empty go to a).
3. As long as the AGF table is not empty, go to (1).

Finally, according to Klahold et al. (2013), the created clusters can be optimized. To do this, each word is checked to see if the association strength to words in another cluster might be higher than to the words in the current cluster.

1. For each word, calculate the inner association strength in the respective cluster by summing the AGF values to every other word in the cluster.
2. For each word, calculate the outer association strength to every other cluster by summing the AGF values to every other word in the respective cluster.
3. Compare the inner and outer association strength for each word and move it to the cluster with the highest association strength.
4. If a word has been moved, start again from (1), otherwise finish the optimization.

To conclude, a cluster represents exactly one focus of the document.

#### 4.4.1.2 Topic Models

There are a number of topic models for the recognition of topics. A widely used model is the LDA (Alghamdi and Alfalqi, 2015). The LDA was introduced by Blei et al. (2003) as a generative process. It tries to imitate the writing process. The LDA understands each document as a mixture of topics. The topic can be given as a discrete probability distribution, which indicates how likely a word is to belong to a given topic (Blei et al., 2003). Figure 4.2 shows a graphical illustration of the LDA approach.

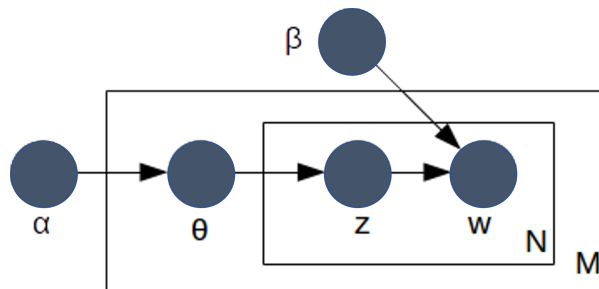


Figure 4.2: LDA Process, (Blei et al., 2003, p. 997)

A special feature of the LDA is that three levels are considered. These are the word, document, and corpus level. A word is the smallest unit considered. A document is considered as a sequence of  $N$  words. It is described as  $w = (w_1, w_1, \dots, w_N)$ . Where  $w_n$  is the  $n$ th word in the sequence. A corpus consists of  $M$  documents and is represented as  $D = (w_1, w_1, \dots, w_M)$ . In Figure 4.2,  $\alpha$  and  $\beta$  are corpus-level variables and will be defined once for the corpus.  $\theta_d$  is a document-level variable

and is defined per document. The word-level variables  $z_{dn}$  and  $w_{dn}$  are defined for each word (Blei et al., 2003). The model is based on Latent Semantic Indexing (LSI) developed by Deerwester et al. (1990) and the probabilistic LSI (pLSI) by Hofmann (1999).

#### 4.4.1.3 Statistical methods

Statistical methods involve examining a corpus of documents. In these procedures, the individual documents require a time stamp denoted as  $t$ . The corpus is divided into equal intervals. These can be hours, days, or weeks, for example. The system now determines which words are statistically significant for the respective intervals. Chi-square tests and Gaussian deviations, for example, are considered (Whitney et al., 2009). Considering contingency table 4.1 for word  $w_i$ ,  $x_t$  is the number of documents containing  $w_i$  in period  $t$ .  $N_t$  is the total number of documents in period  $t$ . The average number of documents containing  $w_i$  before period  $t$  is denoted by  $\bar{x}_{-t}$ .  $N_{-t}$  is the total number of documents before time period  $t$  that do not contain  $w_i$ .  $m$  denotes the number of time slots (Whitney et al., 2009).

	contains $w_i$	$w_i$ not contained
Time $t$	$n_{11} = x_t$	$n_{12} = N_t - x_t$
before Time $t$	$n_{21} = \bar{m}x_{-t}$	$n_{22} = N_{-t} - \bar{m}x_{-t}$

Table 4.1: Contingency table for statistical topic detection (Whitney et al., 2009)

$$\chi^2 = \frac{n_{..}(|n_{11}n_{22} - n_{12}n_{21}| - Y \frac{n_{..}}{2})^2}{n_{1.}n_{2.}n_{.1}n_{.2}} \quad (4.13)$$

The chi-square test from formula 4.13 indicates how the deviation of  $x_t$  in  $N_t$  compares to previous periods. Here, according to (Whitney et al., 2009):

$$\begin{aligned} n_{1.} &= n_{11} + n_{12} \\ n_{2.} &= n_{21} + n_{22} \\ n_{.1} &= n_{11} + n_{21} \\ n_{.2} &= n_{12} + n_{22} \\ n_{..} &= n_{11} + n_{12} + n_{21} + n_{22} \end{aligned} \quad (4.14)$$

$$G = \frac{x_t - \bar{x}_{-t}}{s \cdot (1 + \frac{1}{n})} \quad (4.15)$$

In Gaussian deviation (4.15),  $s$  is the standard deviation. The number of time windows considered is denoted by  $n$ . The formula thus compares  $x_t$  and  $\bar{x}_{-t}$ , normalized with respect to the standard deviation (Whitney et al., 2009).

#### 4.4.2 Challenges and problems

When recognizing topics or events in stories, different challenges and problems can arise. The difficulties of the correct recognition and detection of topics should be explained using the criteria time and place as examples. Similar stories can occur, but they differ in their spatial or temporal reference. To meet this challenge, the temporal and spatial similarity of the stories can be considered in addition to the general similarity (Stock, 2007). To make this possible, it is of particular

importance to determine the exact time of the event. For this purpose, it is not sufficient to identify the publication date of the stories. Text passages with a time reference such as “the day before yesterday” or “last week” must be recognized. The exact date of the event must then be calculated based on these text passages (Stock, 2007). The spatial reference of a story poses a similar challenge. Stories may have a high general similarity but refer to events at different locations. A graph is typically used to determine the spatial similarity via distance. It contains locations in a hierarchical structure equivalent to their geopolitical position.

Stock (2007) further describes another problem by using proper names. Two stories can be classified as similar if they often use the same proper names. In this case, the proper names have a high term frequency (TF). However, both stories can refer to different events. To determine this, the other words must be considered in addition to the proper names to distinguish between the events. To solve this problem, separate similarity values must be calculated between the proper names and the other words. Two stories are considered similar if both values exceed a certain threshold which has to be specified based on the application of topic detection (Stock, 2007).

## 4.5 Semantic Relationship Analysis

For a basic understanding of semantic relationship extraction from textual data, some important terms that occur in the context of relationship extraction or fact extraction will be further explained. Afterwards, two approaches of semantic relationship extraction are presented.

Methods of text-based Information Extraction (IE) are used to extract specific information from text documents and make it available in a knowledge base for further use (Hotho et al., 2005). Typically, text documents are first prepared in the context of pre-processing. This consists of processes such as sentence splitting, tokenization, POS tagging (see Section 3.1), and NER (see Section 4.2). After pre-processing, phrases and sentences can be parsed, semantically interpreted, and finally stored as usable information in a database (Hotho et al., 2005). Structured information such as entities and their properties, as well as relationships between entities, should be extracted from unstructured sources. In this way, more complex query forms than those of mere keyword searches are made possible (Sarawagi, 2008). A look at examples of semantic relationship extraction shows that it often involves determining semantic relations from plain text into a subject-predicate-object triple form, where subjects and objects represent entities and predicates describe relations between entities using a sequence of words (Akbik and Broß, 2009). Such triples can also be extracted in the form of Resource Description Framework (RDF) statements (see Section 4.5.3 for RDF), for which entities with proper names are particularly considered (Martinez-Rodriguez et al., 2019).

In the following, two examples for the extraction of facts are presented. These two are “Wanderlust” by Akbik and Broß (2009) and an approach to extract RDF statements from text in Linked Open Data (LOD) by Martinez-Rodriguez et al. (2019).

### 4.5.1 Wanderlust

Wanderlust is a method and system for automatically extracting semantic relations from grammatically correct English texts. The extracted semantic relations between two entities are represented in a subject-predicate-object triple form. This representation is typically used for statements in RDF (Akbik and Broß, 2009).

The extraction is based on grammatical dependencies within sentences, which are determined by a deep linguistic analysis and the information of the so-called Link Grammar (Sleator and Temperley, 1993). The Link Grammar uses a dictionary that defines all possible connections for each word, which means it determines which word types of the respective word can be connected with (Sleator and Temperley, 1993). Thus, links are formed between grammatically dependent words

within a sentence. The links are given labels according to the type of grammatical relationship they represent (Akbik and Broß, 2009). The following Figure 4.3 serves as an illustration.

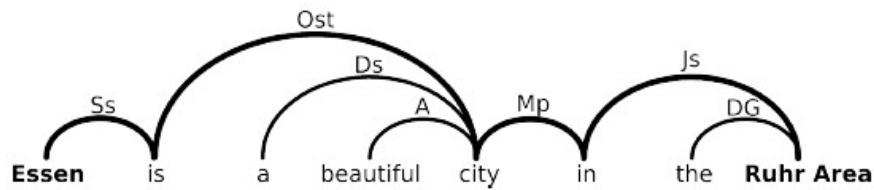


Figure 4.3: Wanderlust: Example sentence with relations (Akbik and Broß, 2009)

According to Akbik and Broß (2009), a linkpath is the path between two words, the so-called start and end term. In Figure 4.3 they are highlighted in bold font. Between them is a wordpath, which consists of several words. If certain grammatical conditions are met in a textual phrase, a wordpath describes the semantic relationship between the two terms. So, if the indirect connections are ignored, the example leads to the wordpath “is” - “city” - “in”. Altogether, the subject-predicate-object triple “Essen” - “isCityIn” - “RuhrArea” can then be extracted from the text data. (Akbik and Broß, 2009)

Obviously, the example makes clear that for the extraction of the semantic relationship between the words in a sentence only a valid linkpath can be used. For the decision whether a valid linkpath is useful, in the approach of Wanderlust by Akbik and Broß (2009) on Wikipedia articles, a manually created training set was used to train linkpaths that deliver useful results most often (Akbik and Broß, 2009). Figure 4.4 shows the process of extracting the semantic triples.



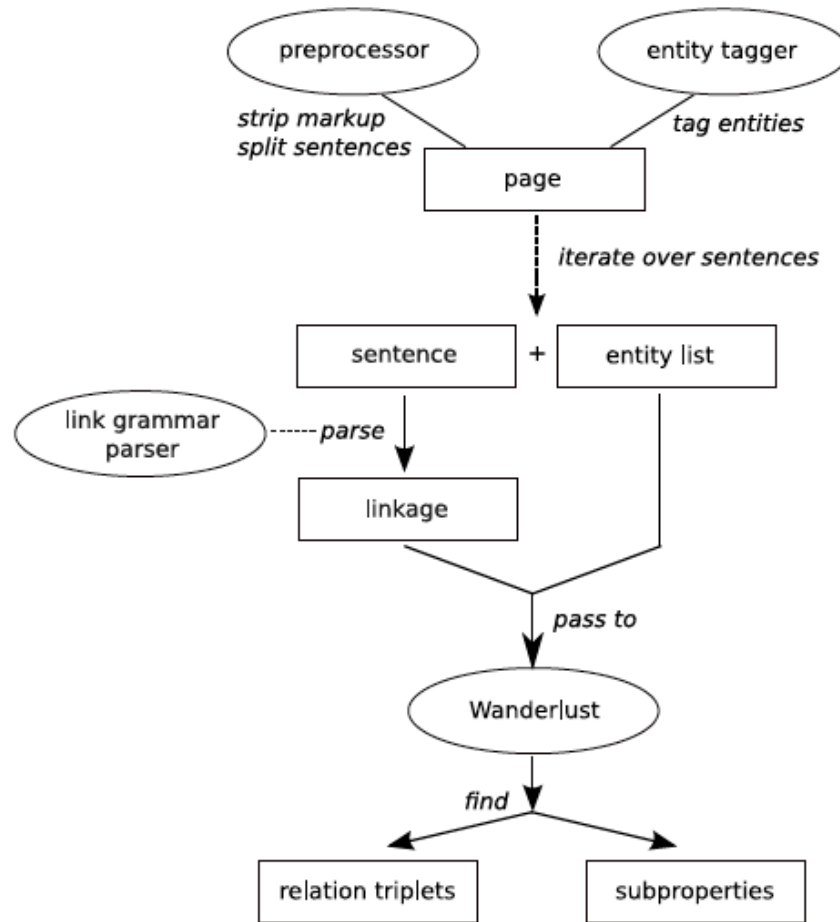


Figure 4.4: Processing steps of Wanderlust (Akbik and Broß, 2009)

In the approach of (Akbik and Broß, 2009), the Wikipedia articles are prepared and divided into sentences. Page links and highlighted terms are then passed to the entity tagger. Words occurring in the sentences are marked as entities if they are contained in the output list. A sentence is passed to the link grammar parser if at least two entities are included. From the set of all links of a sentence Wanderlust extracts a triple and stores it in the Semantic MediaWiki (SMW) database (Völkel et al., 2006). One of its advantages is that it supports subproperties. Therefore, subtypes can be extracted and stored. Subtypes are more specific variants of a wordpath. In addition to the abstract wordpath “isCityIn”, the subtype “isBeautifulCityIn” can also be extracted from the above example. However, since the number of different predicates can become very large, its use in semantic applications is difficult to be implemented and maintained (Akbik and Broß, 2009). A limited number of predicates is therefore considered to be more practical.

#### 4.5.2 Extraction of RDF statements from text

Martinez-Rodriguez et al. (2019) describe in their work a strategy for the extraction and representation of RDF statements (see Section 4.5.3 for RDF) from plain text. They consider two forms of representation, binary and n-ary statements. The binary representation corresponds to the usual triple form consisting of two resources connected by a property or a relation. N-ary statements allow a resource to connect to more than one other resource. They can be useful, for example, to describe specific situations or events where there are entities with different roles.

Following Hernández et al. (2015), a relation is modelled as a resource, so that it can be

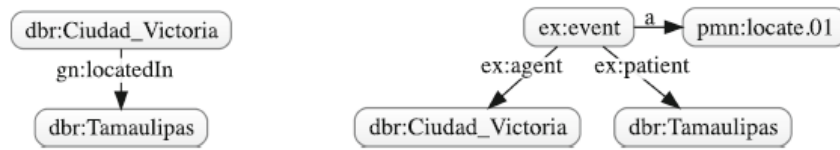


Figure 4.5: Binary and n-ary representation of an example sentence (Martinez-Rodriguez et al., 2019).

enriched with meta-information (Martinez-Rodriguez et al., 2019). The approach consists of three layers, the data layer, the knowledge extraction layer, and the representation layer. In the data layer, the input data is converted into plain text if required, since different input formats are possible. The plain text is then prepared for the next layer using NLP methods such as POS tagging. In the knowledge extraction layer, first mentions of entities found with a NER are linked with resources from knowledge bases such as Wikidata (see Section 4.5.6.1).

As a result of this sub-step, a set of entity Uniform Resource Identifier (URI) pairs is the output. During the subsequent extraction of the semantic relations, a distinction can be made between binary and n-ary representations, depending on the context and the available details. Here, according to Martinez-Rodriguez et al. (2019), n-ary relations require a more complex analysis to obtain all resources involved and to determine their roles (semantic role labeling). Accordingly, the output of this sub-step consists of the relations found between the entities and, if applicable, the roles found. In the representation layer, the correct arrangement of the entities in the RDF statement is determined, and the relationship types are linked to properties from a knowledge base (Property Selection). Properties can be obtained from a knowledge base in different ways, such as by an entity matching comparison (Dutta et al., 2015) or by direct string mappings (Gangemi et al., 2017). Often the degree of matching is also measured, which opens up possibilities to filter less relevant statements. Linking the components of extracted semantic relations with resources and properties of a knowledge base is summarized as Relation Extraction and Linking (Martinez-Rodriguez et al., 2018). Finally, in the last step, the statements are converted into the desired format.

### 4.5.3 Resource Description Framework

The RDF is a format for representing structured information and an important standard for the development of the Semantic Web (Hitzler et al., 2007). RDF enables the representation of information about any entities, such as people, locations, or objects, so that it can be processed by machines (Dengel, 2011). In addition to the RDF data model, two different notations are presented below. This is followed by an introduction to the RDF query language SPARQL Protocol And RDF Query Language (SPARQL) using an example.

#### 4.5.3.1 RDF Model and Syntax

In RDF, entities are represented by resources. RDF resources can be described by properties. Resources can be linked by properties. In this way, statements can be formed in the form of subject-predicate-object triples. Such triples are also called statements. In RDF, statements are modeled as directed graphs with resources as nodes and properties as edges (Dengel, 2011).

Resources and properties are identified by an International Resource Identifier (IRI). An IRI is to be understood as a generalization of URI and does not necessarily also have to represent a web address (Uniform Resource Locator (URL)) (Schreiber and Raimond, 2014).

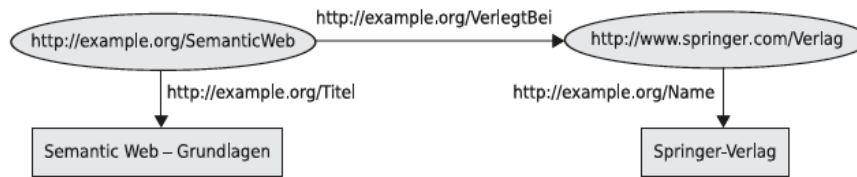


Figure 4.6: RDF statements as graph (Hitzler et al., 2007)

In Figure 4.6, the triple “SemanticWeb” - “PublishedBy” (*German Syntax: VerlegtBei*) - “Publisher” (*German Syntax: Verlag*) is formulated as a graph. It can be seen that the nodes and edges are each described by their URIs. At the end of an edge, there can also be a data value, as in the example. In this way, a resource can be enriched with additional information. Such data values are called literals. As seen in Figure 4.6 these can also occur without an URI, but they can only be the object in a triple (Hitzler et al., 2007). To better process and store RDF graphs, they must be represented in strings. Several notations are available for serialization. One syntax is based on Extensible Markup Language (XML) (Hitzler et al., 2007).

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/">

  <rdf:Description rdf:about="http://example.org/SemanticWeb">
    <ex:VerlegtBei>
      <rdf:Description rdf:about="http://springer.com/Verlag">
        </rdf:Description>
      </ex:VerlegtBei>
    </rdf:Description>

  </rdf:RDF>
```

Figure 4.7: RDF triples in the RDF/XML syntax (Hitzler et al., 2007).

A document in RDF/XML starts with a node of type “rdf:RDF”. Here, the namespaces (in the example, “rdf:” and “ex:”) are declared first. Here, URIs are abbreviated by the identifiers of the namespaces so that the colons they contain do not lead to errors within XML tags. Since XML defines the syntactical structure of the document, RDF triples must be hierarchically coded. So, the triple is located in the element “rdf:RDF”. Subjects and objects are represented by elements of the type “rdf:Description”. The identifiers are appended with the XML attribute “rdf:about”. Predicates, on the other hand, can simply be specified as elements. In the example, the resource “Publisher” (*German Syntax: Verlag*) is contained in the “PublishedBy” (*German Syntax: VerlegtBei*) element, which is again represented as an element. In contrast, literals are specified as data values within a predicate element (Hitzler et al., 2007).

Turtle syntax offers a more compact and easier to read syntax (see Figure 4.8). In Turtle, namespaces can also be declared. However, these are optional and only serve to abbreviate URIs. If URIs are not abbreviated, they must be enclosed in pointed brackets. Statements and declarations end with a period. Furthermore, literals are marked with quotation marks. Finally, it is noticeable that RDF graphs are not hierarchically nested as in RDF/XML but translated directly as triples (Hitzler et al., 2007).

#### 4.5.4 SPARQL Protocol And RDF Query Language

So-called triple stores are used to store larger quantities of RDF triples. These additionally offer the possibility to retrieve the stored information via queries. The query language SPARQL is usually

```

@prefix ex: <http://example.org/> .
@prefix springer: <http://springer.com/> .

ex:SemanticWeb    ex:VerlegtBei    springer:Verlag .
ex:SemanticWeb    ex:Titel        "Semantic Web - Grundlagen" .
springer:Verlag    ex:Name        "Springer-Verlag" .

```

Figure 4.8: Representation of RDF triples in Turtle syntax (Hitzler et al., 2007)

used for it (Hitzler et al., 2007). SPARQL has similarities to Structured Query Language (SQL), in particular, the same keywords are used. However, unlike SQL, the `WHERE` clause formulates RDF graph patterns in the Turtle syntax on which the query is based. Figure 4.9 first declares a namespace for better readability. Unlike the Turtle syntax, the declaration starts with the keyword `PREFIX` and does not end with a period. The `SELECT` clause defines how the output of the query should look like. SPARQL uses query variables that are marked with a question mark. In the `SELECT` clause, they represent the columns in the result table. In the example, all values determined for the two specified variables are thus output in a table with the two columns “title” (*German Syntax: titel*) and “author” (*German Syntax: autor*) (Hitzler et al., 2007).

```

PREFIX ex: <http://example.org/>
SELECT ?titel ?autor
WHERE
{
  ?buch    ex:VerlegtBei    <http://springer.com/Verlag> .
  ?buch    ex:Titel        ?titel .
  ?buch    ex:Autor        ?autor . }

```

Figure 4.9: Example of a SPARQL query (Hitzler et al., 2007)

According to Hitzler et al. (2007), in the subsequent `WHERE` clause, the actual request is formulated in curly brackets. The example in Figure 4.9 shows a graph pattern with three triples. It can be seen that query variables are also used here. They serve as placeholders and are filled with suitable values by the query. The query in the example by Hitzler et al. (2007) therefore searches for subjects of the specified predicate “`ex:PublishedBy`” (*German Syntax: VerlegtBei*) and the specified object “`http://springer.com/Verlag`”. Likewise, the subjects found must then also fulfill the two following triples, because they also contain the variable “`?buch`”. The objects or the values of the two properties “`ex:Titel`” and “`ex:Author`” are then output in the result table because they represent values for variables that are also specified in the `SELECT` clause (Hitzler et al., 2007).

#### 4.5.5 Semantic search

A keyword-based search or full-text search on a text corpus typically returns documents which match to a specific input pattern of characters. The output documents are sorted according to various criteria, e.g., the number of occurrences of patterns in the entire document or in the title. However, information is often searched for without keywords, and various limitations or restrictions. Furthermore, the information is often spread over many documents and can only be found with a more comprehensive search approach. The semantic search should solve such a problem (Bast, 2013; Bast et al., 2016).

There are different definitions for a semantic search. These are often general and focus on a specific area, such as searching for documents. The search can be generally divided into three phases: query, search process and result representation (Dengel, 2011). According to Hildebrand et al. (2007), a semantic search is involved if semantics are used during one of the three phases.

Often NLP, ML and semantic technologies are used to improve the traditional search for documents or the search in knowledge bases. Therefore, semantic search can also be seen as a collective term for approaches that benefit from such techniques.

For example, to search intelligently in a knowledge base, it must be possible to determine during the search process whether an entity is from a special type, so that only matching entities are provided as output. Ontology-like structures, such as Wikidata (see Section 4.5.6.1) or other manually created ontologies allow such search processes. An ontology can be described in a simplified way as a collection of subject-predicate-object triples. Identical entities and relations must have the same identifiers in different triples. Ontologies or knowledge graphs are usually represented by RDF, the Web Ontology Language (OWL) or derivatives thereof. Such knowledge bases usually support structured queries with SPARQL or natural language queries using keywords (Bast, 2013; Bast et al., 2016).

### 4.5.6 Linked Open Data

LOD is a model for publishing and linking structured information on the WWW in compliance with defined rules and best practices. The entirety of the data connected in this way forms a global network of linked data, the so-called LOD Cloud. LOD should enable data providers to publish structured information more easily and users to benefit from improved search options and query results (Auer et al., 2014).

One of the first knowledge bases that contributed to the LOD Cloud is DBpedia (<https://www.dbpedia.org/>). It provides structured RDF data from extracted Wikipedia content and is often the first point of connection for new data in the LOD Cloud. It is also considered a model for the successful implementation of LOD (Dengel, 2011). It is based on the so-called Linked Data Principles, which were introduced by Berners-Lee (2006). These are the following four principles: (1) use URIs as names for things, (2) use Hypertext Transfer Protocol (HTTP) URIs so that people can look up the names, (3) when someone looks up a URI, use the standards (RDF, SPARQL) to provide useful information, (4) include links to other URIs so that more things can be discovered.

The first rule ensures that entities can be identified, for example Tim Berners-Lee has his own URI ([http://dbpedia.org/resource/Tim\\_Berners-Lee](http://dbpedia.org/resource/Tim_Berners-Lee)). The second rule allows the URI to be accessed with a web browser, making it useful for people to get information. The use of standards ensures a uniform and machine-readable presentation of relevant semantic relationships such as “*Tim Berners-Lee is the director of the World Wide Web Consortium (W3C)*”. According to the fourth rule, the W3C also has a URI, which can be looked up (Blumauer, 2014b). Knowledge bases of the LOD Cloud offer many application possibilities, such as linking found proper names with resources from Wikidata when extracting facts (Martinez-Rodriguez et al., 2019) (see Section 4.5.2). Also noteworthy for this work is the free multilingual dictionary Wiktionary, which can be used as a useful resource for computational linguistic applications. For example, it was used in combination with Wikipedia to improve natural language Information Retrieval (IR) (Müller and Gurevych, 2008). For a systematic analysis of Wiktionary see also (Zesch et al., 2008).

#### 4.5.6.1 Wikidata

Wikidata (<https://www.wikidata.org>) is a publicly accessible knowledge base that is extended and controlled by voluntary users. It manages the information contained in Wikipedia with the aim of making it consistent, simple, and available in many languages (Vrandečić and Krötzsch, 2014). The structure of Wikidata has strong similarities to RDF graphs. For example, the statements in Wikidata consist of so-called items and data values, which are linked by properties. Properties have an assigned data type that determines which values are possible, e.g., numbers, strings, URLs or items. Unlike RDF, statements and their components can be enriched with additional information.

For example, the item “Germany” has the property “Speed Limit” with the value “100 km/h”, which consists of a number and a unit of measurement that in turn is an item. Furthermore, statements can have additional property-data-value pairs (qualifiers) that describe the context (in the example: “applies to location” - “out-of-town street”) (Malyshev et al., 2018). Items and properties have unique identifiers with which they are identified. These are numbers that begin with the letter “Q” for items and with the letter “P” for properties. Each item and property have their own pages in Wikidata, in which they are described by statements and further information such as label, description and alias. The creation and deletion of Wikidata items is not arbitrary, but under the control of the community. Therefore, there are fixed guidelines that must be followed. For example, to create a new property, it must first be proposed by a user. Afterwards, a discussion and a vote must take place before the property can be created and translated into all supported languages (Samuel, 2017). Figure 4.10 illustrates the proposition, creation, or deletion of new Wikidata properties.

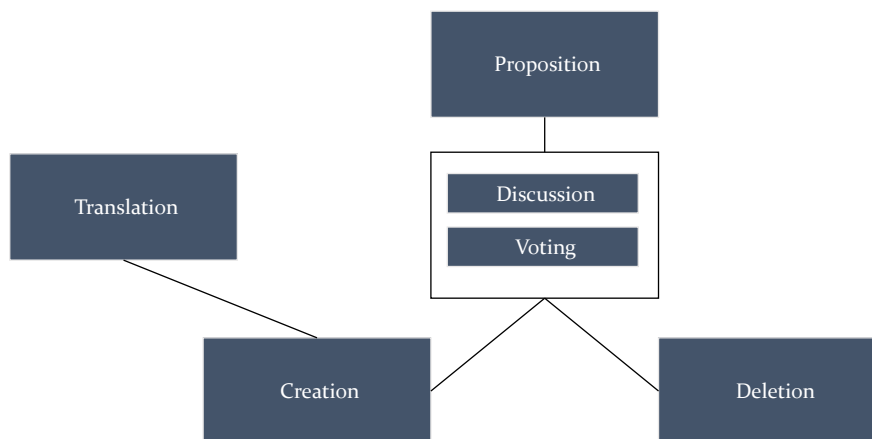


Figure 4.10: Process of creation or deletion of new Wikidata properties, adapted from (Samuel, 2017).

Compared to similar knowledge bases like DBpedia or YAGO (Fabian et al., 2007), there are areas of application for which Wikidata is a more suitable source. For example, Wikidata contains the most people entries. On the other hand, DBpedia contains more places (Ringler and Paulheim, 2017).

## 4.6 Text summarization

Several surveys have been published related to text summarization over the last decades with focus on information extraction, summarization methods and systems (Allahyari et al., 2017; Gupta and Lehal, 2010; Lloret, 2008; Nenkova and McKeown, 2012; Zechner, 1997). Text summarization concepts are generally divided into two related approaches - the *abstractive* and the *extractive text summarization*. While abstractive text summarization focuses on re-writing text based on core information, extractive text summarization identifies the most important parts from a text, a document, or a collection of documents to use them in the summarization process. (Zenkert et al., 2018)

### 4.6.1 Abstractive text summarization

As stated in previous work (Zenkert et al., 2018), in the abstractive text summarization, a summarization system generally tries to understand all textual information given in a document or a collection of documents. After a general knowledge has been gained or trained through machine learning, an abstract or text summary is created using pieces of content re-phrased through trained vocabulary or frequent sequences of n-grams, words or even only characters. Particularly in the field of Natural Language Generation (NLG), this approach requires a profound understanding and good modelling through machine learning and related disciplines. The abstractive text summarization task is more difficult than the extractive summarization since human readers will recognize errors in the text generated by machines and subconsciously evaluate the texts for writing style and its readability. Those difficulties make automatic abstractive text summarization a very complex and non-trivial NLP/NLG task. Through the advances in deep learning, abstractive text summarization gained more attention. For example, research in this area uses Sequence-to-Sequence learning with Recurrent Neural Network (RNN) for the task of abstractive summarization (Nallapati et al., 2016).

### 4.6.2 Extractive text summarization

Extractive text summarization typically uses features created from the content of the text such as term frequency (Luhn, 1958), inverse document frequency (Nobata et al., 2001), contained named entities or word co-occurrences. Furthermore, the title similarity (Nobata et al., 2001) and the utilization of cue words have been proposed in the literature by Edmundson (1969); Fattah and Ren (2009). Sentences are weighted with terms which are similar to the words in the headline (Edmundson, 1969; Nobata et al., 2001). This method assumes that the title or subtitle is the shortest possible form of the text summary. Moreover, the layout of an article has also been considered as relevant for summarization. Therefore, the sentence location (Baxendale, 1958; Edmundson, 1969; Fattah and Ren, 2009; Nobata et al., 2001) and the font styles (Kupiec et al., 1995) have been proposed as useful text summarization indicators. (Zenkert et al., 2018)

### 4.6.3 Hybrid approach

The combination of extractive and abstractive text summarization has been shown in a two-stage extractive-abstractive framework by Liu et al. (2018). Here, extractive summarization has been used to identify relevant information and a neural abstractive model has been applied to generate Wikipedia lead sections and full articles (Liu et al., 2018). (Zenkert et al., 2018)

## 5 Knowledge Representation and Visualization

When it comes to knowledge-based systems, after the phase of knowledge acquisition, one of the key concepts to consider is knowledge representation. Knowledge representation and the inference mechanism are most important for making the system intelligent. In the implementation of AI systems, efficiency, speed, and maintenance are the major characteristics which are affected by knowledge representation (Tanwar et al., 2012).

Knowledge describes the world and gives it meaning but meaning cannot be stored in a machine directly. In order to approach this, knowledge representation is used. The meaning of the data is transformed in such a way that the machine can be used to store and retrieve knowledge (Pfeiffer and Pfeiffer, 2007). Knowledge representation enables an entity to determine consequences by thinking and reasoning rather than acting. The two basic components are reasoning and inference (Tanwar et al., 2010).

Davis et al. (1993) defined the five roles of knowledge representation as follows:

- **A knowledge representation is a surrogate:** It is a substitute of the thing; it enables an existent to determine consequences by reasoning.
- **A knowledge representation is a set of ontological commitments:** Representations are imperfect approximations to reality. Commitments determine how and what to see in the world, in other words what to focus on.
- **A knowledge representation is a fragmentary theory of intelligent reasoning:** It can be stated in term of three components: (1) the representation's fundamental conception of intelligent reasoning, (2) the set of inferences that the representation sanctions which determines what can be inferred, and (3) the set of inferences that it recommends, which is concerned with what should be inferred.
- **A knowledge representation is a medium for efficient computation:** It is a computational environment in which thinking takes place. The representation provides guidance for organizing information to facilitate making the recommended inferences.
- **A knowledge representation is a medium of human expression:** It is the means by which we describe the world, the means by which we communicate with the machine to explain the world to it.

Pfeiffer and Pfeiffer (2007) states that, for the computer the description of a problem which is to be solved is called knowledge representation.

### 5.1 Declarative and Procedural Representation of Knowledge

There are two categories of knowledge representation, declarative and procedural representation (McNamara, 1994; Tanwar et al., 2010). Generally, declarative knowledge is considered the knowledge of facts, and the procedural knowledge is the knowledge of skills. Declarative knowledge representation can be divided into analogical and symbolic representations (McNamara, 1994). According to McNamara (1994), representations retain information intrinsically, are tied to a specific sensory modality, and play a crucial role in many tasks. Especially, in tasks whose solution requires the repetition of previous experiences. McNamara (1994) further states, that symbolic



representations store information extrinsically, are abstract, and form the main basis for logical reasoning.

Unlike declarative knowledge, procedural knowledge is difficult or impossible to teach; it is usually learned gradually, requires much practice, and is used in narrow, well-defined situations (McNamara, 1994). The procedural knowledge used in cognitive skills can be formally represented as production rules. These condition-action rules provide the link between thought and action (McNamara, 1994).

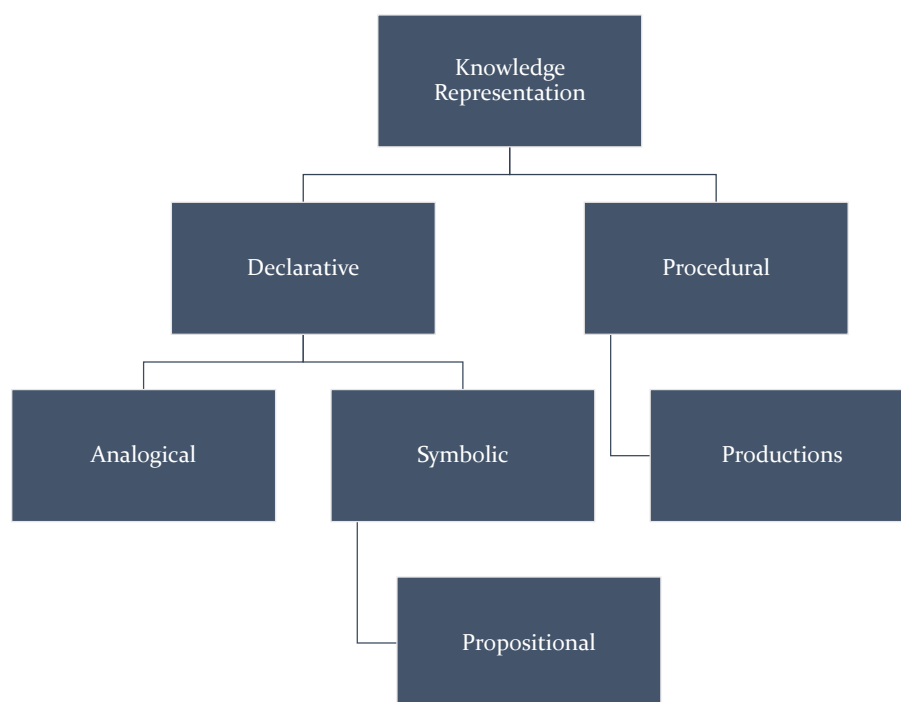


Figure 5.1: A taxonomy of knowledge representations, adapted from (McNamara, 1994).

Objects, facts, relationships can be represented using the declarative representation techniques, and the procedural representation are used for representing the action performed by the objects (Tanwar et al., 2012).

## 5.2 Knowledge Representation Development Techniques

Many techniques are developed for knowledge representation such as semantic networks in which nodes and links represent propositions, schemas such as frames and scripts to represent common-sense knowledge, rule-based and logic-based knowledge representations (Tanwar et al., 2010), all of them have their own semantics, structure, level of power and also their own advantages and disadvantages (Tanwar et al., 2012). Tanwar et al. (2012) stated that “*the propositional logic, predicate logic and semantic net are the declarative knowledge representation techniques and script, conceptual dependency are procedural knowledge representation techniques*”.

One of the common techniques in representing knowledge is the semantic network, which is a directed graph in which the nodes represent the generic or particular class or an instance of a class (object), and the links represent semantic relations between objects. Semantic nets are also known as associative nets because there is an association between two or more nodes of them and these associations are useful for inferring new knowledge. This technique can be used for both representing the knowledge and also for supporting automated systems for reasoning about

knowledge (Tanwar et al., 2012). Knowledge graphs can be considered as a further development of semantic networks.

## 5.3 Knowledge Graph

After the description of knowledge representation and its different forms, the modern representation technique of a knowledge graph will be explained in more detail. This section provides a comprehensive introduction to knowledge graphs, which have recently gained significant attention in scenarios that require exploiting diverse, dynamic, large-scale collections of data.

A knowledge graph is utilized for linking information about the objects from the real world. The object could vary greatly (a person, a movie, a book, and many other types of things). The term knowledge graph refers to large network of entities, their semantic types, properties, and relation between entities. Knowledge graph has been commonly used with semantic web technologies, linked data and large-scale data analytics. Google coined the term knowledge graph in 2012, since then it has been in the focus of research (Auer et al., 2018; Ehrlinger and Wöß, 2016).

### 5.3.1 Definition of knowledge graph

Various descriptions and definitions of a knowledge graph have been published (Ehrlinger and Wöß, 2016). Knowledge graph is generally based on a semantic network approach. Since semantic net is a declarative knowledge representation technique, the knowledge graph is also a similar declarative knowledge representation technique which can be used to represent objects, facts, and relationships. In the following, this concept will be addressed in detail.

The Google Knowledge Graph is defined as “[a graph] that understands real-world entities and their relationships to one another” (Amit Singhal, 2022). The main components are nodes, edges, and labels, which represents a network of entities and illustrates the relationship between them.

Ehrlinger and Wöß (2016) summarize selected state-of-the-art definitions of a knowledge graph:

- “A knowledge graph (i) mainly describes real world entities and their interrelations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains.” (Paulheim, 2017)
- “Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities.” (Kroetsch and Weikum, 2016)
- “Knowledge graphs could be envisaged as a network of all kind things which are relevant to a specific domain or to an organization. They are not limited to abstract concepts and relations but can also contain instances of things like documents and datasets.” (Blumauer, 2014a)
- “We define a Knowledge Graph as an RDF graph. An RDF graph consists of a set of RDF triples where each RDF triple  $(s, p, o)$  is an ordered set of the following RDF terms: a subject  $s \in U \cup B$ , a predicate  $p \in U$ , and an object  $U \cup B \cup L$ . An RDF term is either a URI  $u \in U$ , a blank node  $b \in B$ , or a literal  $l \in L$ .” (Färber et al., 2018)
- “[...] systems exist, [...], which use a variety of techniques to extract new knowledge, in the form of facts, from the web. These facts are interrelated, and hence, recently this extracted knowledge has been referred to as a knowledge graph.” (Pujara et al., 2013)

Ehrlinger and Wöß (2016) further define knowledge graph with the acquisition and integration of information into an ontology. Additionally, it applies a reasoner to derive new knowledge.

They also claim that a knowledge graph is different from the ontology and is better and more complex than a knowledge base, because it uses a reasoning engine to generate new knowledge and integrates one or more sources of information.

Hogan et al. (2021) define a knowledge graph as “*a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities*”.

The knowledge may be acquired from external sources, or from the knowledge graph itself, and knowledge may be a simple statement (Hogan et al., 2021). Hogan et al. (2021) further noted that statements can be added as edges in the knowledge graph. If the graph is to be complemented with quantified statements, a more expressive type of knowledge representation - such as ontologies or rules - are required. Deductive methods can then be used to derive and accumulate additional knowledge. Additional knowledge - based on simple or quantified statements - can also be extracted and accumulated from the knowledge graph using inductive methods (Hogan et al., 2021).

Statements can be similarly interpreted as the extracted statements from text by methods of semantic relationship analysis (see Section 4.5).

### 5.3.2 RDF Knowledge graphs and SPARQL

The most common framework to build Knowledge graph is RDF (see Section 4.5.3), which is used to represent semantic knowledge. RDF was officially adopted by the W3C (World Wide Web consortium) in 1999 as a data exchange model on the web. The use of RDF has been increasingly influenced by the use of semantic web (Zenkert et al., 2018).

The Semantic Web is a common framework which describes the connection and relationship between the resources available in the web, by forming a knowledge-based framework to help users to discover desired knowledge. A typical representation and visualization for large semantic networks are knowledge graphs, which are a graphical representation of a knowledge base (Zenkert et al., 2018).

The knowledge graph representation of RDF or a collection of RDF triples is based on the idea of a directed graph consisting of nodes connected by directed edges. Both elements are assigned unique identifiers. Unlike XML, RDF was not designed for hierarchical structures, but for the general description of relationships between resources. In addition, RDF was designed for describing data on the WWW and other electronic networks with decentrally stored information. Combining these resources is not a problem in RDF, while tree structures like XML are hardly suitable for this.

RDF is regarded as a data model of triples as each statement in RDF is a collection of three parts, subject, predicate, and object and each statement is referred as an RDF triple (Lassila and Swick, 1999). The representation of a knowledge graph for semantic relationships is realized by a collection of triples (Zenkert et al., 2018).

The semantic query language for RDF is known as SPARQL (Prud’hommeaux et al., 2017). It allows the composition of structured queries which consist triple patterns. A triple pattern is an RDF triple with one or more variables (see Section 4.5.4).

Online documents in the WWW can be uniquely identified by the URL, while offline resources don’t have a URL, which are then referred to by Uniform Resource Names (URN). URNs and URLs complement each other. The subject and predicates are often represented as URIs. The object or third part of the triple can also be a URI. Following this approach, the resource can be the object of some triples and at the same time the subject of others, which connect triples into a network of data. The concept can be transferred into a RDF knowledge graph, as shown in Figure 5.2.



Figure 5.2: Example RDF knowledge graph in Neo4j graph database (Thorsten Liebig, 2018)

Although the graph-based representation of RDF is descriptive and comprehensible in visualization, it is not directly suitable for processing in computer systems. Therefore, a serialization of the graph is needed which is the conversion of complex data objects into character strings (Hitzler et al., 2007). This converts the RDF description into a syntactic form which is machine-readable. Each edge of the RDF graph is defined by its start and end point, as well as its label. In this way, an RDF statement is composed by these three components, the subject, predicate, and object (Broekstra et al., 2002), and form an RDF triple (Hitzler et al., 2007). The representation of a graph for semantic relationships is realized by a collection of triples. (Zenkert et al., 2018)

### 5.3.3 Graph databases

Graph databases are also called graphically oriented databases. As NoSQL databases, they utilize a graph scheme to store, map, and query relationships. A graph database is basically a collection of nodes and edges as the relations among nodes in the graph (Ehrlinger and Wöß, 2016). Nodes typically represent entities and edges represent connections or relationships between two nodes. A node has a unique identifier, outgoing and incoming edges, and a set of properties, often expressed as key value pairs.

Graph databases have gained enormous interest in the last years, due to their applications in areas such as the semantic web and social network analysis. Graph databases provide an effective and efficient solution to data storage and querying data in these scenarios, where data is rich in relationships (Ehrlinger and Wöß, 2016).

Neo4j is one the graph databases that is well-known and in wide usage (Thorsten Liebig, 2018). It offers the Cypher query language and also provides translation for SPARQL to Cypher.

## 5.4 Visualization

Visualization is an important aspect of the descriptive representation of a knowledge context. Especially in network and graph visualization, nodes and edges can be used to illustrate relationships. For the visualization of network and knowledge graphs, various properties of semantic network visualization will be discussed in the following.

### 5.4.1 Semantic Network Visualization

A semantic network consists of nodes and edges. Through proper visualization, the relationships between the nodes can be shown vividly. First, an introduction into network visualization is given. Afterwards, various design options and an overview of the aesthetics are presented.

#### 5.4.1.1 Definition and introduction

Networks can be represented as graphs and formally defined as  $G = (V, E)$ . A graph  $G$  has a node set  $V$  and an edge set  $E$ . The nodes represent the units of the graph and are often represented by circles. The edges represent the relations of a graph. If the relations do not have a direction, they are represented by lines. If there is a direction, the edges are represented by arrows. Accordingly, graphs are referred to as undirected or directed (Pfeffer, 2010).

The goal of network visualization is effective communication. Communication is considered effective when it can be decoded particularly quickly by the viewer. The great advantage of visual representation lies in the fact that it can be quickly grasped by an observer. Writing and mathematical notations can be grasped by humans only sequentially. In contrast, individual parts of visual representations can be grasped in parallel. Certain visual stimuli can be processed in parallel. Therefore, communication via visualization has a wider bandwidth in the transmission of information (Krempel, 2005).

A distinction can be made between planar and retinal variables in the design options. The planar variables include the positions on a two-dimensional surface. With the  $x$  and  $y$  values, two pieces of information are already communicated. With the retinal variables, further information can be communicated in the image. Retinal variables include, for example, shape, color, and size. Retinal variables can be distinguished into grouping and separating variables (Krempel, 2005).

According to Krempel (2005), to achieve effective communication, pre-attentive elements are used. These are elements that can be perceived particularly quickly and in parallel by the human brain. It could be shown that retinal variables belong to the pre-attentive elements. Combinations of pre-attentive variables cannot be perceived pre-attentively. If information is communicated by such a combination, all possible combinations must be considered by the human brain. Therefore, information should always be communicated via pre-attentive variables, but never via their combination (Krempel, 2005).

A visualization is intended to make the structure underlying the data visible. However, every visualization always represents a distortion of reality since it is converted into a two- or three-dimensional image (Pfeffer, 2010).

#### 5.4.1.2 Design options

This section takes a closer look at selected design options. Table 5.1 shows which design options are particularly appropriate for each variable.

Rank	Quantitative	Ordinal	Nominal
1	Color	Saturation	Hue
2	Saturation	Hue	Texture
3	Hue	Texture	Saturation
4	Texture	Size	Shapes
5	Shapes	Shapes	Size

Table 5.1: Suitability of the design variables, (Pfeffer, 2010, p. 235)

### 5.4.1.3 Position

Positioning and generation of the layout are among the core problems of network visualization. The position of the units should be based on the inner structure of the data. Even small networks can become very complex (Krempel, 2005; Stegbauer and Häußling, 2010). There are infinite possibilities for the positioning of the elements in a graph (Stegbauer and Häußling, 2010). The layout can be computed from the relationships of the nodes of the graph. There are different algorithms for these calculations (Krempel, 2005). The most commonly used algorithms are the spring embedders. They are also called force directed layout algorithms.

These base the representation on a force model. Figuratively, one can imagine that the edges are replaced by springs. Due to the structure of the network, attractive and repulsive forces act on the springs. By balancing the forces, the optimal edge length is determined. Spring embedders have the property that central nodes are centered in a network. Less central nodes are often arranged in a circular shape outside. An advantage of spring embedders is that the structure of the network is preserved. Thus, adjacent nodes are arranged next to each other.

Spring embedder algorithms have a tendency to utilize the entire available surface area. The nodes arrange themselves as close together as the repulsive forces allow. While this leads to an even distribution of nodes, it distorts the structure. Individual spring embedder algorithms differ in how the forces are implemented in detail (Pfeffer, 2010). In addition to spring embedders, multivariate statistical methods, as well as methods of correspondence analysis, can be used to compute the layout (Krempel, 2010).

### 5.4.1.4 Size

Size as a design element is particularly suitable for quantitative data. An example visualization is shown in Figure 5.3. In the case of visual representation, two properties of human perception in particular must be taken into account. These are known as Weber's law and Stevens' law (Krempel, 2005).



Figure 5.3: Size differences for quantitative and nominal attributes (Pfeffer, 2010, p. 235)

For humans to perceive two objects as being of different sizes, the difference in size must exceed a certain threshold. Otherwise, both objects are perceived as the same size. Weber's law states that the threshold is proportional to the size of the object. Thus, a given size difference is more easily perceived between two short lines than between two long lines. It follows from Weber's law that only a limited number of different sizes can be communicated in an image (Krempel, 2005). Stevens published the power function  $p = ka^a$  in 1975. Through this, the effect of the stimulus on human perception can be described for different media. Here  $p$  stands for the magnitude which is subjectively perceived. The  $a$  corresponds to the physical size of the stimulus. The exponent  $a$  corresponds to the relationship between the stimulus and the perceived impression.  $k$  is a multiplicative constant (Krempel, 2005). For the size perception of lines, an exponent of 1 applies. Thus, the size impressions are linear. This means, a size change is perceived according to its actual physical change. In all other cases, a nonlinear relationship applies. It could be shown that for areas an exponent of 0.7 applies (Krempel, 2010). Areas twice as large are thus not perceived as twice as large (Pfeffer, 2010).

### 5.4.1.5 Color

Color is fundamentally well suited to communicating information. Color can be distinguished between hue, color saturation and brightness. Thus, colors can be used to communicate several pieces of information at the same time. For example, a blue hue in cartography indicates water. Color intensity also provides information about the depth of the water. There are a number of aspects that must be considered in color perception. These include physiological, psychological, and cultural aspects (Krempel, 2010). There are hues for which names exist in almost every culture. These include violet, green, yellow, orange, blue, red, brown, pink, white, gray, and black. By using these colors, cross-cultural international communication is possible. However, colors can have different meanings in cultures.

Colors are not good for communicating information in every situation. The visual representation depends on the light. A change in the lighting situation can change the perception for the viewer. In addition, there is a great influence on perception through contrast. For example, the same color tones can be perceived as different with different backgrounds (Krempel, 2005, 2010). Perception-oriented color systems are used for communication with colors. These color systems are based on human color perception. They distinguish between hue, color saturation and brightness. To communicate qualitative attributes by color, the perception-oriented color systems should also be perceptually uniform. This is the case when the individual color gradations are perceived as equal in size by the human observer. One such widely used perception-oriented color system is the Munsell color system visualized in Figure 5.4 (Krempel, 2005).

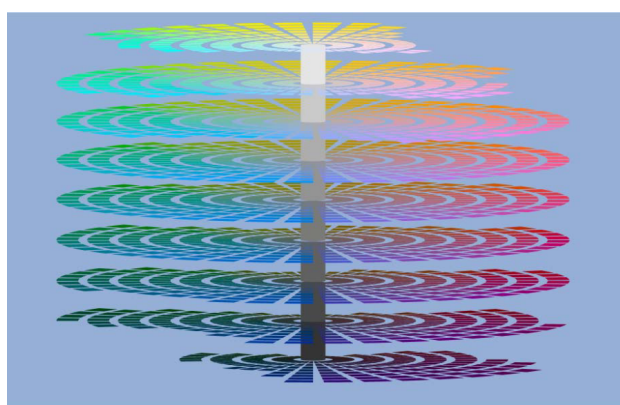


Figure 5.4: Munsell color system (Krempel, 2010, p. 549)

The Munsell color system distinguishes between hue, saturation, and lightness. The color system uses the five hues red, yellow, green, blue, and violet as well as five mixed hues. The hues are arranged around a light-dark axis. This consists of ten gradations from white to black and represents the brightness. The color saturation from colored to neutral gray has 32 gradations in Munsell (Krempel, 2005). The requirements for communication with colors depend on the type of variable. Perceptually oriented color systems are useful for quantitative data. For nominal data, for example, it is sufficient that the hues are distinguishable (Krempel, 2005).

### 5.4.1.6 Aesthetics

Aesthetic criteria increase the readability and comprehensibility of graphic illustrations. They are applied because visual illustrations are made for the human eye (Bennett et al., 2007; Pfeffer, 2010). Bennett et al. (2007) describe four categories of heuristics to ensure an aesthetic visualization of a graph. These are node placement, edge placement, and overall layout. In addition, there

are domain-specific heuristics, which are beyond the scope of this dissertation. Figure 5.5 shows the benefits of using aesthetic criteria. Both visualizations represent the same graph. However, in the right visualization, the structure of the network is more intuitively recognizable through the application of aesthetic criteria (Bennett et al., 2007).

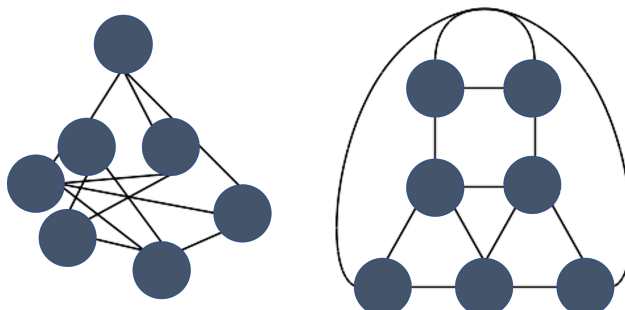


Figure 5.5: Aesthetics in graphs (Bennett et al., 2007, p. 60)

#### 5.4.1.7 Node Placement Heuristics

According to Bennett et al. (2007), node placement heuristics include the following:

- Uniform distribution of nodes: Improves the appearance of the graph.
- No overlap of nodes: There should always be some distance between two nodes.
- Group adjacent nodes: Lets the structure of the network be clearly seen. Conflicts with the requirement of uniform distribution.
- Maintain distance from nodes to edges: This prevents the viewer from misperceiving the structure.
- Maximize orthogonality: Also leads to separation of nodes and improves perception.

#### 5.4.1.8 Edge Placement Heuristics

Bennett et al. (2007) describe the following edge placement heuristics:

- Minimize number of edge intersections: Enable more performant perception by the viewer.
- Minimize number of edge curvatures: Curved edges in a visualization are harder to follow.
- Unify edge curvatures: Different edge curvatures make it more difficult to perceive.
- Minimize edge length: Reduces the area used.
- Minimize maximum edge length: A specialization of the previous requirement.
- Uniform edge lengths: Makes the graph more uniform.
- Maximizing minimum angles: the angles of edges between adjacent nodes should be maximized. Also leads to a more central placement of nodes with high degree.
- Maximizing the orthogonality of edges: Leads to a reduction of edge crossings and maximizes the angles between edges.



#### 5.4.1.9 Overall layout heuristics

According to Bennett et al. (2007), the following heuristics should be considered to ensure an aesthetic overall layout:

- Maximizing global symmetry: for an aesthetic perception, graphs should be symmetric.
- Maximizing local symmetry of subgraphs: Symmetry should also be achieved for the subgraphs of a graph.
- Minimize the area used: this improves the readability of the graph.
- Match aspect ratio to medium: Medium here is for example a screen or a sheet of paper.
- Maximize convex faces: If nodes form a convex surface, this improves perception.
- Consistent overall flow direction: This is a requirement for directed graphs.

## **Part II**

# **MKR Framework: Conceptualization and Modeling**

## 6 Multidimensional Knowledge Representation

This chapter, as one of the main pillars of the dissertation, is a summary of previously published work in several publications. The concept of MKR has been outlined conceptually in (Zenkert et al., 2016), (Zenkert and Fathi, 2016; Zenkert et al., 2018). A detailed description of the MKR methodology, as well as the implementation of the method has been introduced in (Zenkert et al., 2018). The representation method has been further discussed as a method for knowledge integration in smart factories (Zenkert et al., 2021) and for multidimensional decision support (Fathi et al., 2020).

According to previous research (Zenkert and Fathi, 2016), one of the biggest challenges in knowledge discovery is the interpretation of data. Data itself can be meaningless, but through interpretation it can be considered information or even knowledge. Consequently, unstructured data must be interpreted and disambiguated. Following this hypothesis, knowledge creation and information extraction from data can only be achieved by considering the context and metadata of unstructured data.

Intelligent systems with provided knowledge bases are seen as a possible solution to the disambiguation and interpretation task. An innovative approach to using unstructured text resources in a knowledge base is to identify information for multidimensional structuring in each text resource itself. Dimensions can be viewed as flexible categories into which parts of the data are inserted and mapped to identified relationships. (Zenkert and Fathi, 2016)

For example, time information and major keywords extracted from documents can help organize a large collection of documents. Here, according to Zenkert and Fathi (2016) the time and keywords can be considered as dimensional information. Through analysis of the document collection and the textual information it contains, more dimensional information can be identified, associations can be provided, and relationships can be established, resulting in a well-structured document knowledge base. In this way, querying information in the knowledge base with multiple dimensions will provide more accurate search results.

In this example, text mining methods are applied to documents and their textual content. In general, NLP provides the methods to create interpretable data from unstructured data for further analysis. POS tagging and NER are used to provide sentiment analysis, topic detection, and CIMAWA to further extract dimensional information from a textual resource. With the results of text analysis, knowledge can be enriched by adding new relationships and associations to existing or new dimensions in the knowledge base (Zenkert and Fathi, 2016).

### 6.1 Extraction of meaningful information

To extract meaningful information from text resources, the text content must be pre-processed by NLP operations. These operations should be performed by an intelligent system after the text content is provided as data input. The language of the text content, which is extremely important in the analysis, is often provided as metadata, especially for web data. If it is not available, in many cases it can also be identified by a frequent use of stop words in the text. This approach is also important for identifying language changes within the text content. Depending on the identified language, language-specific models should be selected during text analysis by the application. The most commonly used pre-processing steps in NLP are sentence splitting and tokenization. Both are also applied as initial steps in the described approach (Zenkert and Fathi, 2016).

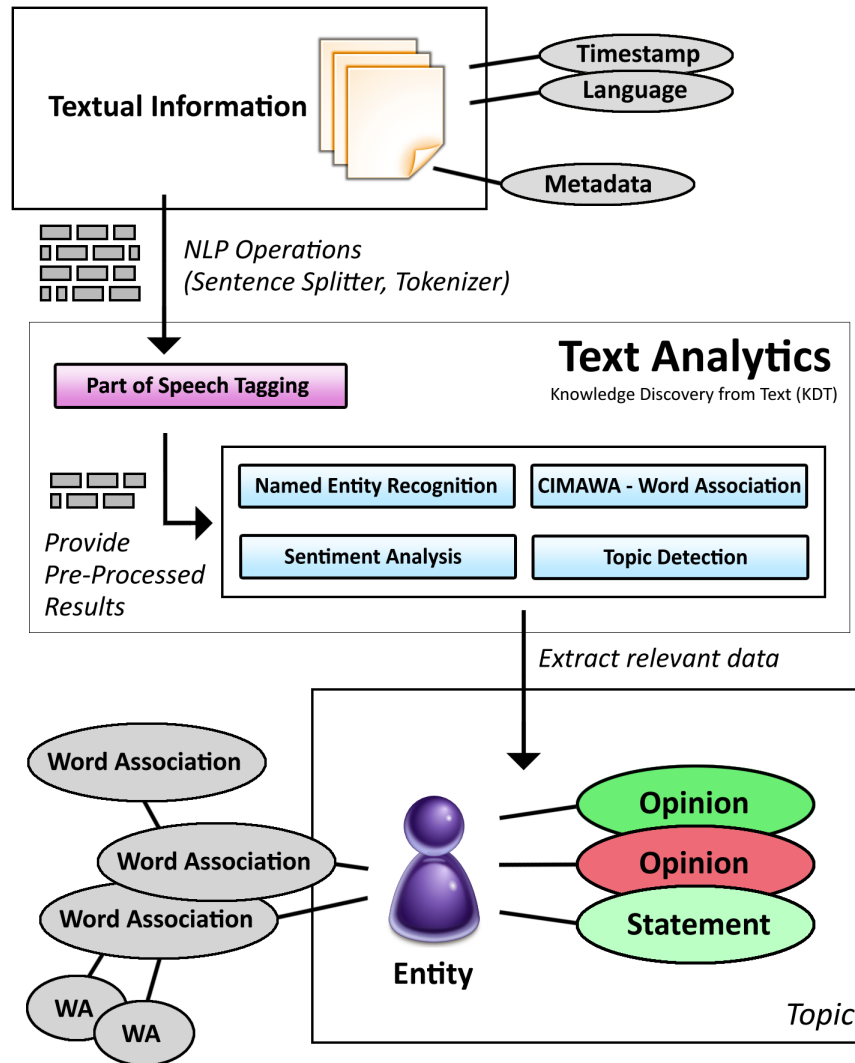


Figure 6.1: Extraction of entity information from textual resources with different analysis results (Zenkert and Fathi, 2016)

After pre-processing, various text mining methods can be used in the extraction of entity information. The whole process is shown in Figure 6.1, published by Zenkert and Fathi (2016).

### 6.1.1 Pre-processing

Zenkert and Fathi (2016) discuss the pre-processing as follows: Sentence splitting and tokenization are used to convert documents into a machine-readable form. The first step is to convert the document into a list of sentences. Based on the layout of a text, the heading and paragraphs are separated. Then, the text is broken into smaller parts. The actual content of a document, the body of the text, is further divided into sentences, taking into account punctuation. In the next step, the identified sentences are processed with a tokenizer to create a list of tokens. Depending on the implementation of the application and other analysis methods, the list of tokens can also be represented internally as a vector of words. The extracted tokens and sentences are used as input for POS tagging in the next step. Punctuation is also very important in this input, and the POS tagger must take it into account to assign the POS tag for each word in a sentence.

### 6.1.2 Part-of-Speech Tagging

With the annotation labeling of each word in a sentence, grammatical analysis shows the structure of the sentence and entities can be resolved in the next step. Many approaches to POS tagging have been developed in the past. Rules, maximum entropy models, and stochastic methods are proven methods for the POS tagging process (Brants, 2000; Brill, 1992; Ratnaparkhi, 1996). POS tagging utilizes lists of tokens from sentences and looks for a likely POS sequence. An option for POS tagging with stochastic methods is a Hidden Markov Model (HMM). Based on a trained model, the POS tagger is used to calculate the most likely POS sequence based on the probability of the next tag. With the tagging of each word in a sentence, the grammatical analysis shows the structure of the sentence and entities can be resolved in the next step. Identified word types such as adverbs and adjectives are also particularly useful for sentiment analysis, as they are thought to have a large influence on sentiment intensity (Benamara et al., 2007).

### 6.1.3 Named Entity Recognition

NER is used to identify entities of different types in textual information. Various approaches to NER have been developed in the past. Maximum entropy models, transformation-based learning, HMM, and robust risk minimization have been studied, and their performance was considered the best. Typical examples of entity types are people (e.g., person names), organizations (e.g., companies, organizations), places (e.g., cities, countries), and date or time expressions (e.g., months, years, time). Lists of person names or place names (gazetteers) can also be used in the application to identify entities by keywords. However, these lists are not always useful, especially when the entities are named in variations within the text content. In this case, NER focuses on the POS tags provided and makes an estimate based on the probabilistic value of the tag for a named entity (NE) in a sentence tag sequence.

### 6.1.4 Sentiment Analysis

Sentiment analysis is used to determine whether a positive, neutral, or negative opinion, statement, or subjectivity is expressed by one entity about another entity. Methods and developments towards opinion-based information retrieval systems have been studied in the past (Pang and Lee, 2008). Approaches in the literature mainly focus on using a corpus, a dictionary, keywords, or a lexicon to decide whether the available textual resources contain terms with a polarity (Cambria et al., 2013). A well-known lexical resource for opinion mining is SentiWordNet 3.0, which is used by more than 300 research groups (Baccianella et al., 2010). A semantic orientation based approach for sentiment analysis was developed by Agarwal and Mittal (2016).

Sentiment analysis as a method for extracting information for the multidimensional knowledge base focuses on the identified entities through NER and therefore searches for terms that highlight and express emotions about other entity information within the text. Classification of terms into emotion types can be directly used for dimensional information in the knowledge base design of this research. (Zenkert and Fathi, 2016)

### 6.1.5 Word Association

The hybrid word association measure CIMAWA, presented in (Uhr et al., 2013), is used to create a numerical value of the association strength between entity keywords and other directly related words. In this way, human association can be modeled for each entity and provides a list of descriptive features for the knowledge base. Entities can be compared to each other using these association profiles, and association strength can be analyzed frequently to detect temporal changes in

the knowledge base. In addition, word associations can be used in both knowledge representation and visualization to define distances between entities and dimensions based on a numerical value derived from the strength of the word association.

### 6.1.6 Topic Detection

Various approaches have been developed for topic detection in the past. Different levels have been considered for topic detection. Entire documents have been assigned to topics based on classifications, bag-of-words, or keywords. In Latent Dirichlet Allocation (LDA), each document or text resource is considered as a mixture of different topics. LDA is a generative probabilistic model for collections of discrete data and was introduced by Blei et al. (2003).

Latent Semantic Indexing (LSI) is an indexing and retrieval technique introduced by Dumais et al. (1988). LSI uses Singular Value Decomposition (SVD) to identify patterns in the relationships between terms and concepts contained in an unstructured text collection (Dumais et al., 1988). LSI has been used in many research papers and applications.

Based on the strengths of word associations obtained from CIMAWA computations, the concept of Associative Gravity (Klahold et al., 2013) can also be used to identify multi-topic structures in text resources. Associative gravity uses word associations to identify different topics in a text.

## 6.2 Conceptualization of Dimensional Representation

The conceptualization of dimensional representation has been first mentioned in (Zenkert and Fathi, 2016).

Similar to the Semantic Web and the RDF, information from different analysis results is to be kept in a multidimensional structure in the knowledge base. With the aforementioned information representation styles as an archetype, the principle of semantic relationship should also be considered and modeled in the knowledge base. Facts and extracted information normally stored in the knowledge base are provided in the form of a triple, and represented as a combination of subject, predicate, and object. However, these parts are further classified into their dimensional affiliation.

With the dimensional structuring approach (Zenkert and Fathi, 2016), a combination of different data can be extracted from the knowledge base as one piece of information since the dimensions are taken into account during extraction and provide the necessary context. In this way, queries that retrieve data from different dimensions provide more detailed results and smarter output from the knowledge base. For example, entity opinions on specific topics in text resources (e.g., news, social media) can be identified, analyzed, and stored in the knowledge base. By querying the knowledge base, the analysis returns opinions that can be compared from two different dimensional perspectives. Figure 6.2 shows the conceptual visualization of a possible scenario.

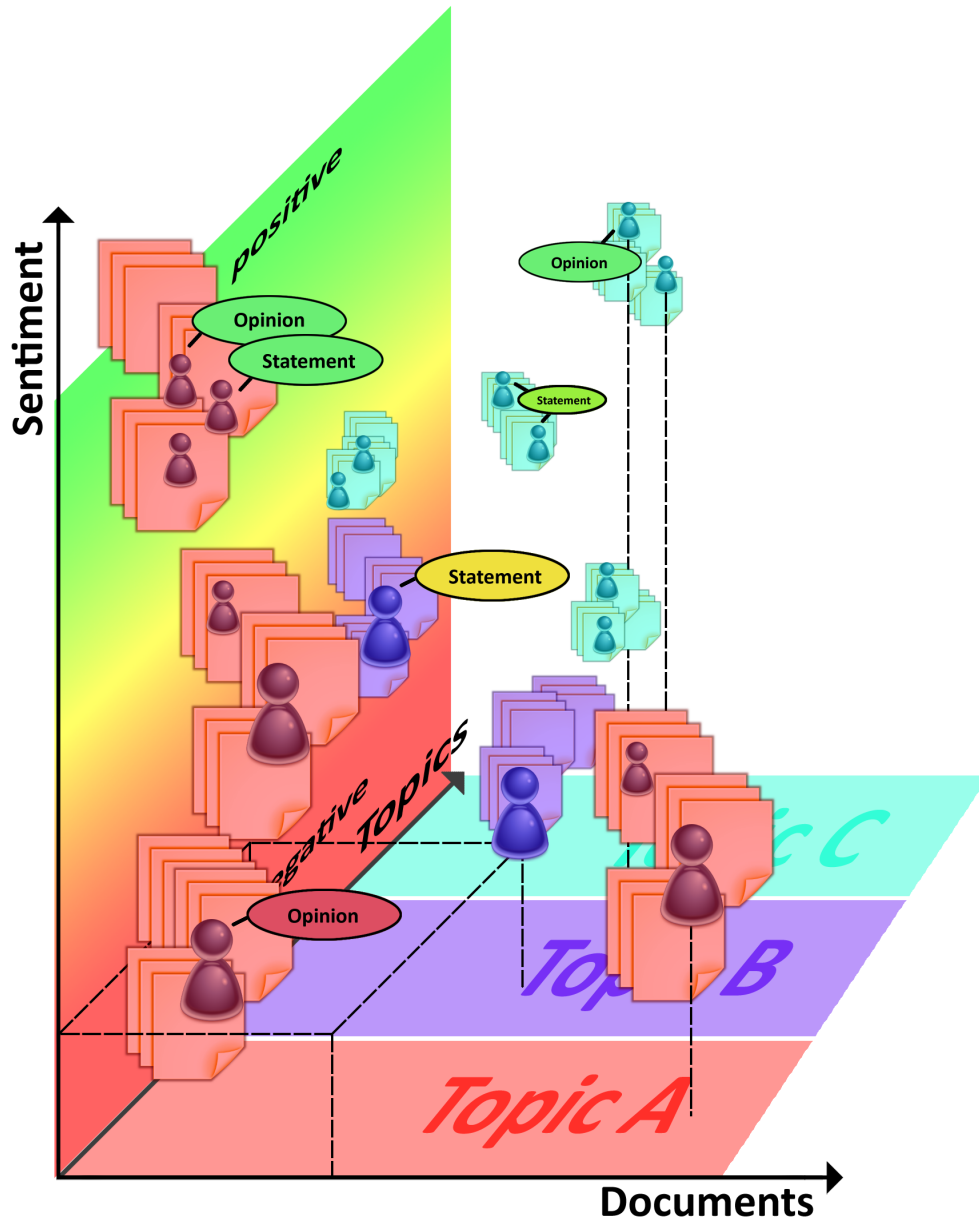


Figure 6.2: Conceptual visualization of a three dimensional relation between entities, assigned documents and sentiment evaluation in corresponding topics (Zenkert and Fathi, 2016)

Further dimensional information can represent a useful analysis potential. Usually, dimensions are derived from metadata or extracted from the textual content using the text analysis methods already mentioned. In the following sections, adapted from Zenkert and Fathi (2016), possible dimensions are briefly presented. In addition, interesting application scenarios are mentioned to further explain the practical benefits of the presented knowledge base design.

### 6.2.1 Metadata information

Metadata information can be used directly for specific dimensions in the multidimensional knowledge base. Therefore, unstructured data must be analyzed in search of this information. There are different document formats that provide different ways to use the metadata they contain. For web data, for example, a document created with Hypertext Markup Language (HTML) declares

the Document Type Definition (DTD). This metadata information is available in XML format and is often used to specify all the associated properties (e.g., description, author, language, keywords) of the document.

When comparing documents, metadata is a useful filtering tool that should be captured and associated with the document itself, the corresponding entities, the topics included, and other extracted information. Examples of valuable metadata information for use in dimensions are given in the following sections, first published by Zenkert and Fathi (2016).

### 6.2.1.1 Time

Timestamps are available in the description header of documents. This information can be used directly for the time dimension of the extracted information. In the temporal dimension, a distinction must be made between the date of creation, the date of last access, and the date of last modification. This separation allows the relevance of the knowledge to be derived from the last access or last modification, depending on the use case. Parts of the knowledge base content that are never used can be archived. Newer information and information with a high frequency of use must be kept directly in the knowledge base and prepared for fast access in order to avoid long loading times. (Zenkert and Fathi, 2016)

### 6.2.1.2 Language

Language is often a metadata property in many text resources. In the description header of documents, the language property is specified and can be used as dimensional information. In this way, the knowledge base can be partitioned by language-specific dimensions. Due to the availability of different text analysis methods trained on different languages, information can be extracted from documents and assigned to the appropriate language dimension. (Zenkert and Fathi, 2016)

### 6.2.1.3 Organization Structure

Knowledge bases in industrial context should cover all information about customers, suppliers, stakeholders, products, and business processes to support the overall corporate strategy (Zenkert and Fathi, 2016). Based on the different units or the organizational structure, the knowledge base can be customized to cover this specific information for each unit separately by dimensions. In this way, the knowledge base structure is directly related to the organizational structure. For example, any extracted information from text resources that is relevant only to the company's marketing department can be assigned to a marketing dimension.

### 6.2.1.4 Access Restriction

In corporate knowledge bases, dimensional information is essential for describing access restrictions. Information must be organized in a secure and reliable way. By using dimensions, a distinction between private and public information can be easily added. Public information should therefore be labeled as publicly available information in the knowledge base. In addition, different user rights can be directly assigned to the dimensions in the knowledge base.

### 6.2.1.5 Personal Information

One of the dimensions in a knowledge base should be usable for personal preferences and needs. Selected information that needs to be accessed frequently should be marked as particularly relevant



for personal work. In this way, a time-consuming search for information can be avoided since all relevant information is already prepared in the knowledge database. (Zenkert and Fathi, 2016)

### 6.2.2 Textual Analysis Results

Entity, sentiment, topic, and associative content are the introduced dimensions derived from the analysis results of the information extraction process mentioned earlier. Examples of analysis results for use in dimensions are given in the following sections, first published in (Zenkert and Fathi, 2016).

#### 6.2.2.1 Entity

Entities recognized by NER are considered as a central entity in the knowledge base. Associated and related information for each entity is separated in different dimensions. In this way, the entity dimension selection allows entities to be compared (e.g., different products, people, or other entity types). The entity dimension offers enormous analysis potential.

#### 6.2.2.2 Sentiment

Extracted text phrases may contain opinions and emotions that can be classified as positive, neutral, or negative. Each expressed opinion should have at least one related entity. Based on the relationships from one entity expressing an opinion about another entity, a relational link is established between the dimensions in the knowledge base. The results of sentiment analysis have tremendous analysis potential by using the multidimensional structure of the knowledge base. In the above organizational use case, the company's knowledge base can provide statistically sound answers to interesting market research questions by analyzing customer feedback on specific products.

#### 6.2.2.3 Topic

Identified topics can be assigned to various other textual content in the knowledge base. This makes it easy to query the knowledge base, which aggregates information on specific topics. Since topics in the knowledge base may change over time, temporal analysis can be used for trend analysis. In the enterprise knowledge base use case, customer feedback for each specific product is a multidimensional search result and can provide interesting insights for product improvement.

#### 6.2.2.4 Associative Content

Based on the associative content dimension, the textual information in the knowledge base can be mapped. For example, based on the strength of association between a word and another word, a numerical value from CIMAWA can be used to indicate the distances between words. As a result, a knowledge map of the associated content can be created. Adding more dimensions to the knowledge map provides further analysis potential. The associative content dimension can also be used as a basis for enterprise content management and innovative associative search methods. Based on the multi-dimensional mapping in the knowledge base, associative search can directly provide all related content if some keywords are specified in a search method. Additional dimensional filtering can identify the relevant content and improve algorithms for finding the right content in enterprise content management (ECM) systems.

## 6.3 Multidimensional knowledge base design

Different applications have their own databases and do not provide a universal interface to use them as a central or distributed knowledge base. There are disadvantages to pursuing this approach in IT strategy. Data is not updated consistently and may have critical or unnecessary gaps if organizations do not use a unified database schema. In contrast, if a centralized knowledge base is provided to the organization, the potential for analysis and decision support for management and decision makers increases with the volume of data. The knowledge base concept presented is suitable for organizational data, as explained using the use cases above, but is also capable of storing a wide range of other knowledge from other use cases.

The knowledge base uses the concept of relationships between dimensions. In this way, not only can the associations or relationships be represented, but also any strength of the relationship between the represented features in a given dimension can be described.

The properties of scalability, flexibility, dimensionality, and relevance can be considered beneficial to the knowledge base design. In the following sections, these properties, adapted from Zenkert and Fathi (2016), are briefly described.

### 6.3.1 Design characteristics

#### 6.3.1.1 Scalability

For the scalability of the multidimensional knowledge base, according to Zenkert and Fathi (2016), the key question is how knowledge can be represented and visualized. This feature influences the overall design of the knowledge base. The integrated data is represented in a relational way between different dimensions. With the relationships between data entries, scaling can be modeled directly in the knowledge base. The scaling of data requires different levels of representation to summarize or specify data in more detail. That is, in the knowledge representation, an entity can be represented internally as an entity. However, the entity is connected to various other entities through dimensions.

Similar to the main functionality of Online Analytical Processing (OLAP), the knowledge base design provides the ability to consolidate data (known as “roll-up” and “drill-down”) by displaying related entities and their information, extract dimensional information (“slicing”), and search for specific knowledge through multidimensional information (“dicing”). These capabilities are of great use for decision making. (Zenkert and Fathi, 2016)

#### 6.3.1.2 Flexibility

In designing the multidimensional knowledge base, special attention is paid to the different forms of data input. Based on the concept for representing entity aspects in dimensions, additional dimensions can be easily added to the knowledge base to further specify and describe the existing knowledge. In addition to the qualitative and quantitative characteristics of the data, various formats of unstructured data (e.g., image, audio, video formats) must also be considered for integration into the knowledge base. The design of the knowledge base enables the integration of different data formats and takes into account the increasing size of the data volume through dimensional structuring. (Zenkert and Fathi, 2016)

#### 6.3.1.3 Dimensionality

Dimensions are the basis for the structure of the knowledge base. Dimensions are considered fully customizable when designing the knowledge base. New dimensions can be added to the knowledge

base by creating new relationships between object representations in the knowledge base. In this way, the design takes advantage of a schema-less structure that automatically adapts to new data inputs. By using the text mining methods described earlier, unstructured textual information can be transformed into a dimensionally structured form. Adding and removing additional dimensions can filter the results and greatly increases the analysis potential.

#### 6.3.1.4 Relevance

The relevance of data can be determined by interpreting data in a context (e.g., querying the knowledge base for data across multiple dimensions). By considering multiple dimensions, contextual information helps to decide whether (new) data must be kept in the knowledge base, deleted, or even discarded as irrelevant. The relevance characteristic of the knowledge base is also influenced by the time dimension included. New data or recently changed data are stored for quick access, long-term data are archived accordingly. In addition, the relevance aspect is largely decisive for the quality assurance of the knowledge base and the data it contains. The relevance consideration is of high importance for companies, which are often faced with the problem of deciding whether their information is still up-to-date or reliable.

#### 6.3.2 Continuous update cycle for the knowledge base

The continuous updating process of the knowledge base is visualized in Figure 6.3. It was first published in (Zenkert and Fathi, 2016).

To create the knowledge base, information must first be created in a dimensional structure. Also, text resources must be provided for the applications to perform the analyses. The text corpus is used as input for training the text analysis procedures for information extraction. At the beginning and at the initialization of the knowledge base, a user has to distinguish between correct and incorrect suggestions of the system. The user's effort and distinction between correctly and incorrectly extracted information from textual resources can be measured by precision and recall calculations.

The first knowledge in the knowledge base can be a single piece of information. After inserting the data into a dimensional context, this relationship can be used and extended by the system. For example, if the knowledge base starts with the information "Berlin is the capital of Germany", other extracted information can be inserted into the knowledge base as relations to the entities "Berlin" and "Germany". In this example by Zenkert and Fathi (2016), these entities will be easily recognized by location-based gazetteers of NER. The relationships between the added entities do not need to be fully specified, since it is always possible to update the relationships and dimensional structure in the provided knowledge base. In this step, the two expressions "Berlin" and "Germany" are also considered as dimensions. Since the word association of "capital" and "city" pushes the system in different directions and adds unknown relations and dimensions to the existing knowledge base, the numerical word association strengths directly provide distance values between directions and help to differentiate the content.

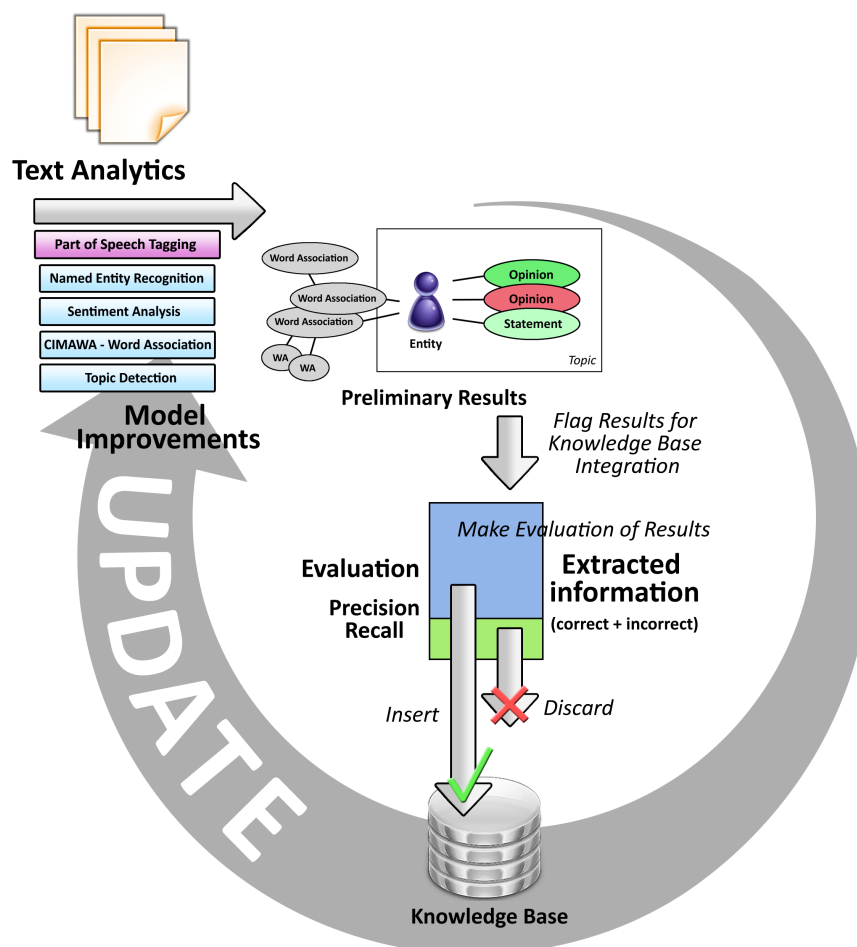


Figure 6.3: Update process of the multidimensional knowledge base (Zenkert and Fathi, 2016)

Updates and integration of new data into the knowledge base lead to a reduction in user effort, as more training data provides better newly trained models as input to the applied text mining methods. This hypothesis holds until better information extraction accuracy is no longer worth the decrease in system performance due to additional rules or exceptions. However, in this knowledge base design by Zenkert and Fathi (2016) with continuous integration of new data with associated relations, it is hard to avoid over-fitting and over-optimization of models. Nevertheless, the optimization process is considered as a specialization to one domain or subject area.

With tremendous flexibility, the knowledge base design can be used in different use cases and application domains. By using a dimensional structure, knowledge representation and knowledge discovery in the knowledge base can be facilitated by the central hypothesis, the interpretation of data-mediated contextual information. Furthermore, the potential for interpretation and analysis increases with the volume of data and the completion of the overall dimensional structure.

With the integration of newly extracted information from textual content, analysis methods can be improved and lead to less effort in decision making about integrating new information into the knowledge base. (Zenkert and Fathi, 2016)

## 6.4 Representation Benefits

According to Zenkert et al. (2018), MKR is a representation format that enables the combination of different analysis results from integrative text mining and related methods in the knowledge












base. For this purpose, the results obtained by synchronously or asynchronously computed text mining methods are considered as dimensional information and inserted into a shared representation structure. By combining different analysis perspectives and their results, the knowledge base is able to provide a broad overview of the information obtained and use it for further analyses and queries. Thus, integrating the analysis results into a holistic knowledge representation structure offers advantages in visualizing the data and inferring the knowledge base.

While normal representation structures are limited to and based on a specific, predefined analysis result, multidimensional representation structures are able to offer further unexpected analysis results, context, or additional information for the knowledge discovery process and methods. This is particularly reflected in knowledge visualization, as appropriate knowledge graphs can add further results from various other analysis dimensions or perspectives for adaptation or customization. In this way, results are used for contextualization, filtering, comparing, combining, or even discovering unexpected results in further analyses. (Zenkert et al., 2018)

For example, when querying a knowledge base for entities directly related to a specific topic in a given time period, MKR can directly offer further information about sentiment information or extracted facts related to both the entities and the time period and topics. Entity visualizations, e.g., entity type specific nodes with a color indicator from red (negative) to green (positive), are used in this scenario to show the sentiment relationship within the same visualization, while, if desired, edges could indicate a semantic relationship or other nodes could indicate a thematic relationship or influence the layout arrangement. (Zenkert et al., 2018)

As the example by Zenkert et al. (2018) shows, MKR is particularly important in knowledge visualization because the results of other analysis methods are inserted directly into the main visualization without changing the knowledge base query. Table 6.1 gives an overview of the typical visualization options for different text mining methods and the modifications made with MKR.

Table 6.1: Typical Visualization Types for Text Mining Analysis Results (Zenkert et al., 2018)

	 <b>Tile View</b>	 <b>Topic Map</b>	 <b>Area Chart</b>	 <b>Bar Chart</b>	 <b>Scatter Chart</b>	 <b>Entity Graph</b>	 <b>Map Chart</b>
 <b>Named Entity Recognition</b>	Tiles show corresponding entities. Size and color is variable on type or number of occurrence.	Topic rectangles show corresponding entities. Size and color is variable on type or number of occurrence.	Areas show the number of entity occurrence in text collection over time.	Bars (stacked) indicate the number of entity occurrence in text collection over time.	Markers show the number of entities which have been identified in text collection over time.	Documents are represented as node collection based on content and included entities.	Parts of the map chart (countries) are colored based on the number of occurrence of country names which have been identified.
 <b>Sentiment Analysis</b>	Tiles are colored in green-red scale based on sentiment evaluation.	Topic rectangles are colored in green-red scale based on overall sentiment evaluation.	Areas show the number of positive, neutral or negative evaluated texts from a collection over time.	Bars (stacked) show the number of positive, neutral or negative evaluated texts from a collection over time.	Markers show the calculated sentiment evaluation of texts from a collection over time.	Nodes visualize documents in different colors in green-red scale based on overall sentiment evaluation.	Parts of the map chart (countries) are colored based on their (document) related sentiment evaluation.
 <b>Topic Detection</b>	Tiles are in different colors which represent different topics.	Topic rectangles are colored in different colors that represent different topics.	Areas show the number of texts which have been assigned to different topics from a text collection over time.	Bars (stacked) show the number of texts which have been assigned to different topics from a text collection over time.	Markers show the number of texts which have been assigned to different topics from a text collection over time.	Nodes visualize documents in different colors that represent different topics. Nodes are connected with topic node.	Parts of the map chart (countries) are colored based on their related topic.
 <b>Semantic Triple Extraction</b>	Tiles visualize Subjects or Objects from extracted triples (Subject - Predicate - Object).	Topic rectangles visualize Subjects or Objects from extracted triples (Subject - Predicate - Object).	Areas show the number of texts over time in which Subjects or Objects have been identified.	Bars (stacked) show the number of texts over time in which Subjects or Objects have been identified.	Markers show the number of texts over time in which Subjects or Objects have been identified.	Nodes and edges visualize the extracted triples (Subject - Predicate - Object). Relationships are visualized.	If Subject or Object is a location, parts of the map chart (countries) are colored.

In the next section, the process of MKR to enrich basic knowledge representation is explained in detail. It has been first published in (Zenkert et al., 2018).

### 6.4.1 Processing and Representation

Typical tools for knowledge management are document management systems (DMS). DMS typically use indexing of various variables or characteristics (e.g., metadata) to search and find documents in the knowledge base. Documents are usually represented by a set of properties in a relational database. With schema-free structures in NoSQL databases, the representation methods can be more flexible. In this way, the basic document representation format of knowledge bases can be dynamically enriched with additional dimensional information or analysis results as they become available. Supporting this enrichment process is the main intention of MKR. (Zenkert et al., 2018)

MKR prevents multiple queries of the knowledge base by combining and storing different text mining analysis results in one representation format. The included additional information is provided by the methods of integrative text mining, namely NER, topic detection, sentiment analysis and semantic relations extraction. It is worth noting that other text mining methods can be inserted into the MKR process and into the same resulting representation structure. For example, long- and short-term word associations of entities within the document could be inserted into MKR and would be available for temporal analyses of documents over time without re-computation.

Pre-processed documents are extracted from the knowledge base and analyzed in essentially four steps, described by Zenkert et al. (2018): 1) extraction of named entities 2) detection of topics 3) determination of sentiments 4) extraction of semantic relations. Then, the analysis results are collected, stored, and presented in the database in JSON format. Figure 6.4 illustrates an exemplary MKR process from Zenkert et al. (2018).

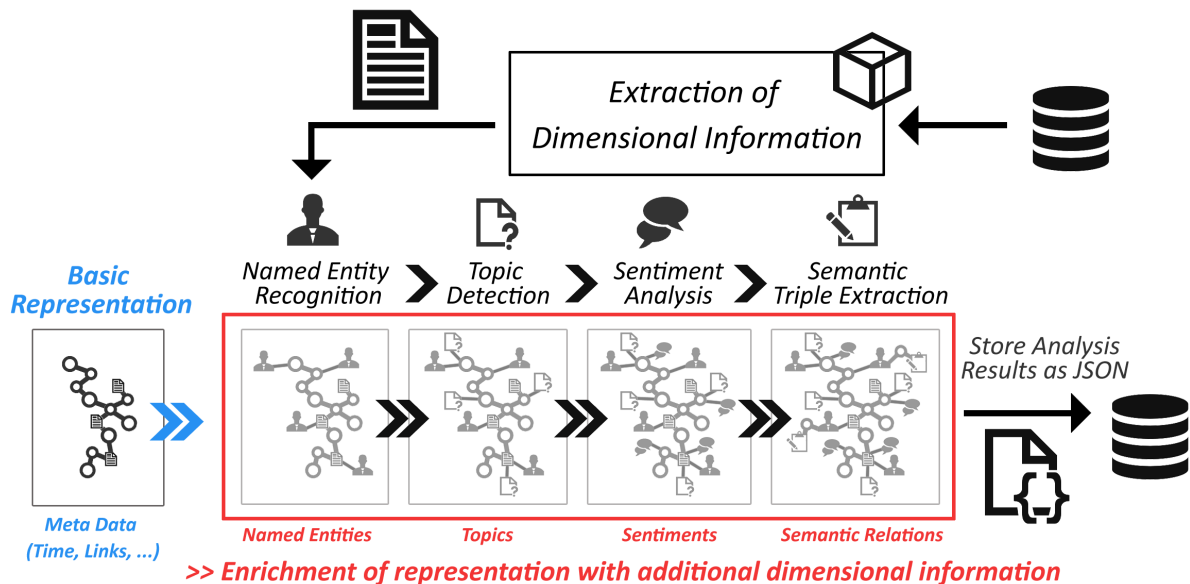


Figure 6.4: The process of enrichment in MKR (Zenkert et al., 2018)

It is also important to note that the MKR process does not necessarily have to follow a four-step procedure. Methods can be skipped, other methods can be included, or even executed independently - as long as there are no dependencies. Because of the schema-less representation of MKR in JSON format, results can be added to the knowledge base as they appear or as the computation is completed.

In the next section, the visualization capabilities of MKR are discussed in more detail. State-of-the-art visualization methods are used to explain the advantages of MKR and how it can be used as a multidimensional visualization tool.

## 6.4.2 Adaptation of State-of-the-Art Visualization

In the following, visualization types are explained, most of which can be considered state-of-the-art visualizations from related work in data mining and knowledge discovery (Fayyad et al., 2002; Stahl et al., 2013). By briefly explaining the main goal of the visualization types, the advantages of MKR should be presented by describing the visualization modification and customization options mentioned in Table 6.1. The graph types and their MKR relation have been published in (Zenkert et al., 2018).

### 6.4.2.1 Overview Graphs

Overview visualizations are used to provide a summary of the most important, recent, or unexpected information at a given point in time. Below the Tile View and Topic Map tools are presented as overview visualization tools that work very well in combination with MKR.

**Tile View:** Tile view is an overview visualization that displays a selected group of elements. The elements can usually vary in size and/or color. The number of elements is also variable. For textual data, a tile usually contains a summary of the text content or the first part of the text as a preview. Here, results from NER, sentiment analysis or topic detection, but also semantic triples can be used to change the colors and size of the tiles. For results from NER, images are also useful to show the entities contained in the text, and they can be used for visualization e.g., in the background of a tile. (Zenkert et al., 2018)

**Topic Map:** The Topic Map is another overview visualization that represents a selected group of elements. The elements can usually vary in size and/or color. Each part of the topic map represents a dimensional information at a certain hierarchy level. Figure 6.5 a) shows an example of a topic map visualization. Each rectangle represents the topic of a collection of documents where topics have been detected based on a topic detection method and the analysis results have been stored in MKR. Associated subtopics, also referenced to topics, and represented in MKR, are additionally visualized here - to see the distribution of the document collection based on the rectangle size. The topic map is interactive and allows navigating to different hierarchy or dimension levels. For example, in Figure 6.5 c), navigation from all topics to a specific topic is shown, with the topic “Society” selected and the layout of the topic map changed to visualize related entities of type “Person” shown in MKR (similar to an OLAP’s drill-down operation). As shown in Figure 6.5 c), the small collection of rectangles within the topic indicates the number of occurrences of the corresponding entity in the document collection. Drill down further into the document subset that matches the society topic, and a particular entity can be selected again, and so on. In general, the structure of MKR allows the topic map to be customized in desired dimensions. Another example is shown in Figure 12.1, published in (Zenkert et al., 2018), where the sentiment analysis results from the MKR representation are used as another filter.

### 6.4.2.2 Statistical Graphs

Statistical graphs are typically used to visualize information from a category (e.g., knowledge dimension) over a specific time dimension. Therefore, statistical charts typically represent the time



dimension with an adjustable scale. Area chart, bar chart and scatter chart are typical examples and will be explained in the following.

**Area Chart:** In the area chart, various time series are displayed in the form of stacked areas. All data values of the time series represent a summary of the analysis results at a given time (or period). The colors and appearance as well as the number of stacked areas are changed based on the selected dimensions.

**Bar Chart:** Typically, a set of data is plotted horizontally or vertically at various discrete values (e.g., timestamps) to create a bar chart. The bars can contain different categories and represent the analysis results in stacked or grouped form. Figure 6.5 b) shows an example where the topic map has been converted to a bar chart while the same results display is active. Here, different topics are visualized together with a time dimension, showing the number of documents related to the topics over time.

**Scatter Chart:** The scatter chart is a type of visualization that represents data observations in two dimensions. Each point in the chart (also called a marker) represents an observation in the data of one dimension at a particular value of another dimension. The points can vary in size and color to visualize additional information. Figure 6.5 d) shows an example in which documents from the topic area "Society" are ordered by the result of MKR's sentiment analysis. Each marker represents the number of documents (by changing the marker size) within a configurable interval over time. The color indicates the sentiment values on a scale from positive (green) to negative (red).

#### 6.4.2.3 Entity Graphs

The entity graph is a visualization type that uses nodes and edges to represent relationships between data. Each node usually represents an entity or a collection of entities. Nodes and edges can vary in size, spacing, and color to illustrate additional information. The layout of the entity graph can be customized based on various layout algorithms. Figure 6.5 e) shows an example where documents are visually linked to entities extracted from text content. The entity graph is the best option for visualizing the semantic relationship between extracted entities and RDF triples (e.g., nodes for subjects/objects and labels for predicates).

#### 6.4.2.4 Map Graphs

Documents containing named entities of type "location" can be used for abstract visualization in a map graph. Each named geographic location can be used to highlight different parts of the map. For example, countries are visualized on the map and can be used to represent analysis results. Figure 6.5 f) shows an example with countries mentioned as named entities in the content of a document collection. The color intensity is based, for example, on the frequency of occurrence of named entities or on other method results stored in MKR, such as sentiment or topic relation.

### 6.4.3 Transformation

According to Zenkert et al. (2018), MKR enables easier transformation from one knowledge visualization to another due to its generic representation structure and combination of dimensional analysis results. Dimensional selection and dimensional filtering are two possible MKR operations

that can be used to adapt MKR analysis results for visualization. An overview of example visualization transformations is provided in Table 6.2. It further describes use cases of MKR by the possible combination of different dimensions in different visualizations.













Dimensional selection separates the MKR dataset of the knowledge base into a smaller subset by selecting one or more values for the data source, time, or language. In this way, a particular subset of the knowledge base can be dynamically prepared or customized.

Dimensional filtering produces a smaller subset by filtering the analysis results with specific values from one or more dimensions. The analysis results represented by MKR in the knowledge base and thus knowledge elements searched for can be greatly reduced by dimensional filtering. According to Zenkert et al. (2018), dimensional filtering produces a knowledge base query that performs a search for documents that relate to a particular topic (e.g., politics), have a particular sentiment range (e.g., positive), and contain at least one occurrence of a particular entity (e.g., White House).



Figure 6.5: Knowledge graph types and different visualizations of MKR representation (Zenkert et al., 2018): a) Topic map with subtopics, b) Bar chart which shows the number of documents in different topics over time, c) Topic map with selected topic “society” - visualizing all named entities from type “person”, d) Scatter chart with sentiment analysis results from documents in the “society” topic, e) Entity graph with visualization of documents from the topic “society” - connected with named entities, identified by named entity recognition (and represented in MKR), and other semantically related knowledge items, f) Map graph visualization which indicates the reference of named entities from type “location” within documents from the topic “society” - visualizing the result of document’s sentiment analysis.

Table 6.2: Knowledge Graph Visualization Examples based on MKR (Zenkert et al., 2018)

Dimensions					Visualization							Description / MKR
 Time	 Named Entity	 Sentiment	 Topic	 Facts	 Tile View	 Topic Map	 Area Chart	 Bar Chart	 Scatter Chart	 Entity Graph	 Map Chart	
Time Stamp	✓	✓	✓	✓							✓	Documents are represented as a node collection based on content and included entities. Topic and semantic relationships are inserted into the entity graph as nodes and edges. Sentiment is represented with colors or by modification of edges and/or nodes (size). Furthermore, layout algorithms consider dimensional information to arrange node distances.
Time Span		✓	✓				✓	✓	✓			Sentiment and topic information from selected documents is represented in separate areas, bars or markers within the visualization. The best indicators are sentiment or topic related colors and size (number of documents).
Time Stamp	✓	✓	✓								✓	Sentiment and topic information about selected documents is represented in separate parts (countries) of the map chart. The best indicator is color. Each country in the map is highlighted by its related topic or sentiment color.
Time Stamp	✓	✓	✓			✓						Entities, sentiments or topics are presented inside the topic map. For example, all entities which have a strong relationship to the corresponding topic are visualized in the topic rectangle as smaller rectangles. Rectangles can be further highlighted according to the assigned sentiment, named entity or topic.
Time Stamp		✓	✓		✓							Tiles typically consist of parts from the document's content / a text summarization. Sentiment or topic information is visualized in the tile overview by color modifications.

## 7 Extraction, Operations and Pipeline Integration

### 7.1 Pre-Processing Pipeline

The pre-processing steps which are implemented in C# for the KB:mkr software (see Section 11) are described in the following sections. The tasks which are applied on loaded documents from the MongoDB include language detection, sentence splitting and POS tagging. A detailed description has been published by Zenkert et al. (2018).

An own implementation of the pre-processing steps within the implementation of the prototype has been chosen over external tools or libraries because of lacking availability in C# language, customization possibilities, research purpose and especially for future work adaptation.

Figure 7.1 visualizes the applied pre-processing of documents and integration of results into MKR. Compared to the linear process of text mining (see Section 2.2.2), results from the extended text mining process are stored in MKR format and kept in the knowledge base for further analysis.

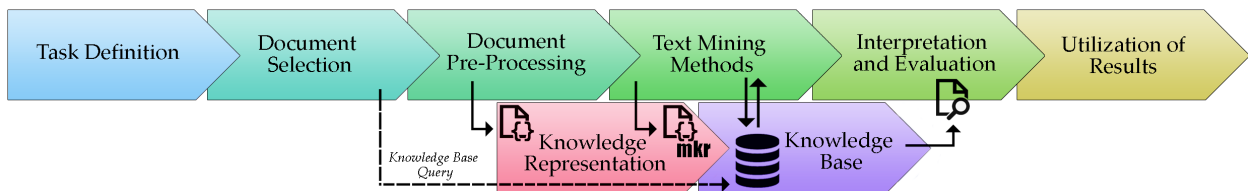


Figure 7.1: Extended Text Mining Process and Pipeline Integration

#### 7.1.1 Language Detection

The implemented KB:mkr software uses a language detection system that analyzes the textual content of documents and determines the stop words within the message text. Two different lists of stop words are available for the German and English language, which contain several language-specific words that are not meaningful. Each word from the document is compared with the entries in the stop word lists and the corresponding language with the most matches is finally recognized as the language of the document. This recognition approach works very well for large documents and considerably well for short texts. Documents containing very small text fragments or text phrases without stop words are flagged for manual detection.

#### 7.1.2 Sentence Splitting

In the KB:mkr software, an algorithm is implemented to split documents into sentences. A rule-based model was created from scratch to separate sentences based on a set of regular expressions (e.g., applied to punctuation with set of exceptions). Furthermore, additional language-specific rule sets are used to avoid incorrect sentence splitting in many different cases. These cases include, for example, titles, abbreviations, numbers, currencies, and further problematic cases.

#### 7.1.3 Part-of-Speech Tagging

The KB:mkr software implements a POS tagger based on a Hidden Markov Model (HMM). The POS tagger uses the STTS tagset (Schiller et al., 1995) for the German language and the Penn

Trebank tagset (Marcus et al., 1993) for the English language. Both implemented POS taggers use a separate corpus (training set) with POS annotated sentences and about 1 million words. Also, the Viterbi algorithm (Forney, 1973) has been implemented and is applied in both taggers.

## 7.2 Text Mining Methods and Integration of Results

In the text mining architecture described by Zenkert et al. (2018), the methods NER, topic detection, sentiment analysis, and semantic triple extraction are used. With NER, named entities in the text - and thus relevant semantic relationships within the design and process documentation - are detected, linked to corresponding documents, and made available in the knowledge base.

In the industrial context, applied by Zenkert et al. (2018), named entities are designations for machines, manufacturing systems, product names, process components, names of experts, and other bodies of knowledge referenced in the text. Topic detection distinguishes between document-oriented topic recognition and multi-topic recognition, which enables a more precise assignment of text components to subtopics. Sentiment analysis examines polarity and opinions within a text and focuses on positive or negative word choice within a text, at document, sentence, entity, or aspect level. Semantic information extraction further examines language use to generate facts from written text based on the grammar of the text. All methods use either frequency-based criteria in combination with dictionaries (e.g., term frequencies), co-occurrence analysis and evaluation (e.g., word association strength), or as classification models trained by machine learning.

An architecture for the integration of text mining results has been proposed by Zenkert et al. (2018). Figure 7.2 illustrates the overall approach. Input documents, in the form of industrial documents, logs, reports, etc. are pre-processed as previously described. Here, NLP extracts n-grams (individual tokens), keywords and co-occurrences for linking of text information and relevant machine data. After the creation of the initial data basis, the enrichment process of MKR is applied to enable intelligent search queries and knowledge inference. Furthermore, knowledge discovery and knowledge visualization could provide various insights into the analysis of the process documentation from the industrial use case. Those insights are also relevant for the business process analytics as the last step in the architecture (Zenkert et al., 2018), (Abu Rasheed et al., 2020). (Dornhöfer et al., 2020) propose integrative text mining in multi-agent system, (Fathi et al., 2020) applies it for decision-support.

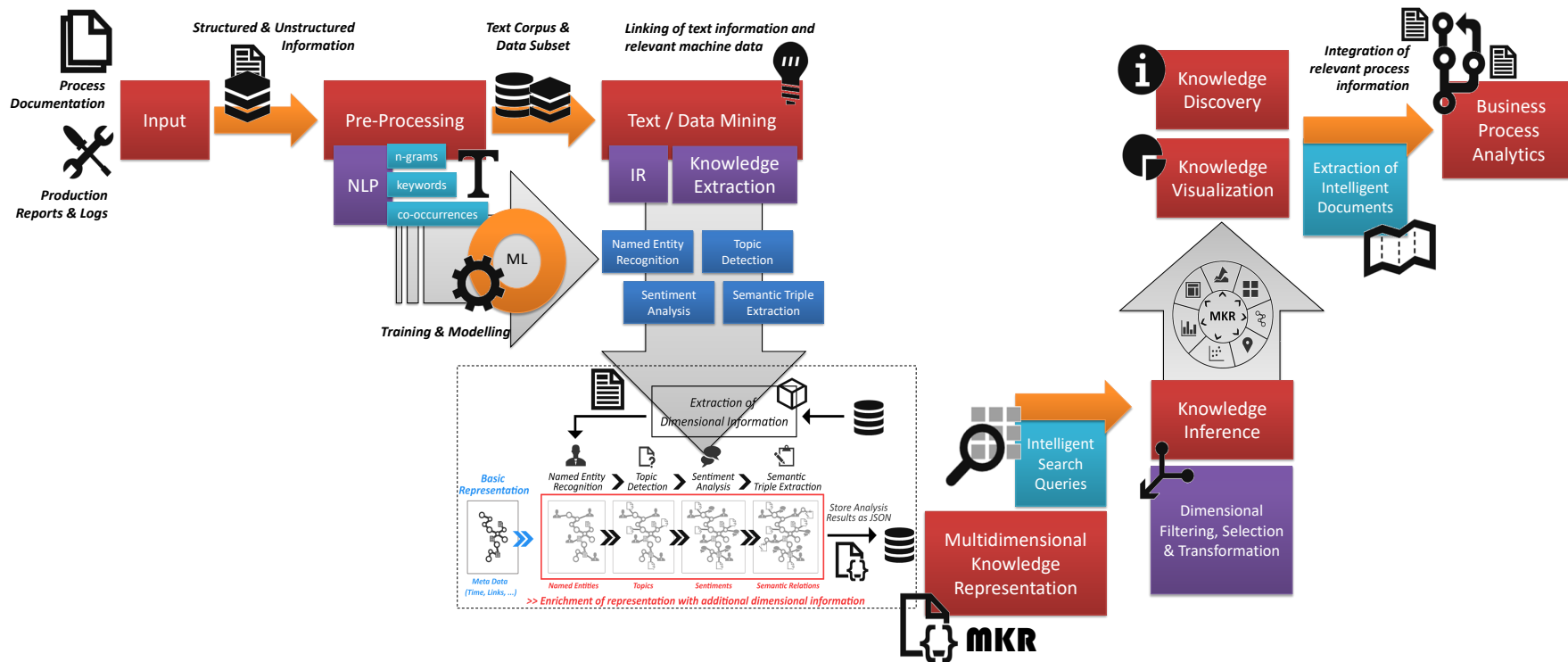


Figure 7.2: Architectural Overview of Process Analytics through Integrative Text Mining utilizing MKR (Zenkert et al., 2018)

## 8 Related Work

This chapter discusses related work for various aspects. First, additional frameworks for representation of knowledge are discussed. Afterwards, big data and big data analytics published in (Zenkert et al., 2018) is focused. In a third section, the concept of integrative text mining is further discussed.

### 8.1 Representation Approaches and Frameworks

According to Zenkert et al. (2018), semantic technologies facilitate the interconnection of web data sources and enable the structured representation of relationships between different pieces of information. In a semantic context, information is viewed as a set of objects and meaningful relationships that can show a larger, complex picture in a simple and structured representation.

To this end, the visualization of semantic structures is typically a graph that contains graphical mappings of topics, concepts, and or entities. In addition, following Zenkert et al. (2018), these structures, such as semantic networks, contain relationships between these objects. A typical representation and visualization for large semantic networks are knowledge graphs - a graphical representation of a knowledge base. The information in the knowledge base can be organized in different forms, although the typical form of knowledge base representation is an ontology, a collection of semantic triples or dimensional information stored in a multidimensional knowledge base. The analysis and representation of semantic relationships between entities, information extraction, or inference on knowledge graphs are just a few examples of the use and application of knowledge graphs. In recent years, large knowledge graphs have been created. Freebase (Bollacker et al., 2008), Wikidata (<https://www.wikidata.org/>), Yago (Fabian et al., 2007), NELL (<http://rtw.ml.cmu.edu/rtw/>), Microsoft Concept Graph (<https://concept.research.microsoft.com/>), and Google Knowledge Graph (<https://www.google.com/intl/es419/insidesearch/>) are typical examples.

Simitsis et al. (2008) uses a multidimensional structure for content analysis and exploration by combining keyword search with online analytical processing (OLAP) aggregation, navigation, and reporting.

Zhang (2013) proposed the use of topic modeling for OLAP on multidimensional text databases (MTD).

Lin et al. (2008) defined a text cube model for multidimensional text databases.

Similarly, Zenkert and Fathi (2016) conceived the proposed MKR as a framework for text analytics and extended it in (Zenkert et al., 2018).

#### 8.1.1 Big Data and Big Data Analytics

Big Data refers to very large amounts of data across a wide range of applications. It has become a popular term describing the exponential growth, availability, and use of information from both structured and unstructured data (Michalik et al., 2014). The application of advanced data analytics methods in real-time or near real-time is referred to as Big Data Analytics. Big Data analytics is the process of examining large amounts of data with heterogeneous data types to discover hidden patterns, unknown correlations, market trends, user preferences, and other useful information (Marjani et al., 2017). The results of Big Data analytics can provide important insights into areas such as customer behavior based on social media analytics related to a particular product (Bohlouli



et al., 2015). These results enable companies to improve productivity and efficiency and make efficient and informed decisions (Marjani et al., 2017). Fast Big Data technologies for Big Data analytics use in-memory technologies and apply distributed parallel processing to handle data from multiple sources (Chen and Zhang, 2014). In this way, data processing is significantly accelerated compared to traditional processing and storage techniques. For scenarios with large amounts of data, NoSQL databases are particularly suitable. These databases are not based on a fixed schema and are therefore compatible with many data types. In addition, they allow customization of data formats and structures without affecting the application landscape in use.

Big Data is often characterized primarily by the large volume of information. However, there are also other important aspects; typical characteristics are the volume, the variety of data (Variety) and a high speed of data generation (Velocity) (Zenkert et al., 2018).

**Volume:** The first characteristic of Big Data is the volume of data. Companies are managing increasingly large volumes of data. This is the case, for example, at Google, Yahoo or Facebook, where about 500 terabytes of new data have been stored every day (Zenkert et al., 2018). In total, there have been more than 4.4 zettabytes of data. The requests from the approximately 1.37 billion daily active users on Facebook (<https://newsroom.fb.com/company-info/>), average value for September 2017) had to be processed simultaneously (Zenkert et al., 2018). Relational databases are not very efficient for this amount of data, so database technologies with horizontal scalability are necessary. It becomes practically impossible for a server to process such data on its own because the data processing times are too long.

**Variety:** Another important feature of Big Data is the variety of data. Big Data refers to the storage of structured, semi-structured and unstructured multimedia data (text, images, audio, and video) (Fasel and Meier, 2016). In fact, most Big Data is unstructured data. Unstructured data is mostly information that cannot be stored in a relational database or traditional data structure. Statistics show that more than 80 percent of all available data is unstructured (Zenkert et al., 2018). This data usually comes from various social networks such as Facebook, Twitter, or YouTube. Such large, poly-structured data sets (consisting of numbers, texts, images, videos, relationship data, etc.) have a high potential for predictive and forecasting analytics (Gadatsch and Landrock, 2017).

**Velocity:** The third aspect of Big Data mentioned is Velocity. Velocity refers to the speed at which data is generated, stored, analyzed, and processed (Nandury and Begum, 2016). Big Data needs to be processed quickly to help companies make better decisions. Figuratively speaking, the speed of data is accelerating. For example, sensors placed on roads to analyze traffic capture thousands of data values per minute. This data must be processed in real time to predict traffic. Consequently, the speed of data processing is important for predicting our climate, financial markets, and many other areas where forecasting models and methods are used.

The analysis of unstructured textual data sources from different perspectives is called integrative text mining (Zenkert et al., 2018).

### 8.1.2 Integrative Text Mining

According to Zenkert et al. (2018), integrative text mining is the process of extracting information from text resources under different analysis perspectives and combining the results for further analysis potential. In general, text mining provides a number of different statistical methods for computers to understand unstructured text content. Typical applications include text classification or clustering, topic detection, entity recognition, sentiment analysis, automatic document summarization, or knowledge fact extraction. With the capabilities of the various text mining analyses,

topics, sentiments, entities, and facts can be identified and presented in separate visualizations. However, in the integrative text mining approach, different analysis perspectives are combined to gain additional information or to exploit further analysis potentials.

As mentioned in the Chapter 2 of this dissertation, the terms text mining, web mining and data mining are often used in a similar context, although they are different application areas. The term text mining is also often associated with data mining as parent category. Alternatively, text mining and web mining are referred to as a special form of data mining (Bohnacker et al., 2002). Data mining, also described as the KDD process, is concerned with searching and analyzing large amounts of data and analyzing recognizable patterns and rules (Abts and Mülder, 2009).

**Part III**

**Implementation**

## 9 External Data, Lexical Resources and Linked Open Data

This chapter describes an additional base for the implementation. In the first section, the external data used are discussed, giving an overview of the materials used as well as the lexical resources and linked open data.

### 9.1 External Data

This section describes external data that was used for implementation. Typically, word lists are considered as external data, which have a scientific background and have been developed especially for use in text mining and NLP.

#### 9.1.1 Stop Word Lists and Language-specific Resources

In the context of NLP, removing stop words from texts can be beneficial for improving the results of text mining. A number of freely available stop word lists can be found online. In this implementation, the following stopword list has been considered:

- NLTK: The Python implementation NLTK, Natural Language Toolkit (<https://www.nltk.org/>) provides stop word lists for fourteen different languages. In addition to German and English, stop word lists are also available for the languages Danish, Finnish, French, Italian, Dutch, Norwegian, Portuguese, Russian, Swedish, Spanish, Turkish and Hungarian. The NLTK stopword list can be selected to remove the stopwords from the retrieved texts. In addition, the lists are used in the implementation to detect the language of a text.

### 9.2 Integration of Lexical Resources for Integrative Text Mining Methods

Additional resources have been used for the integrative text mining approach. Two lexical resources should be mentioned specifically here for the algorithms of sentiment analysis, the associative sentiment analysis, and the topic detection, considering a topic thesaurus to classify documents into specific topics.

#### 9.2.1 SentiWS: Sentiment Evaluation

In this implementation, the Sentiment Wortschatz (SentiWS) (Remus et al., 2010) is used. This is a freely available resource for sentiment analysis in German language. The SentiWS was developed at the University of Leipzig. The SentiWS consists of two lists. One contains words with 1,650 positive polarities and one contains words with 1,818 negative polarities. In addition to polarities, the lists contain POS tags and alternative word forms. With these word forms, the SentiWS contains 15,649 positive and 15,632 negative words. The polarities are given in an interval from -1 to 1 (Remus et al., 2010). For the implementation, version 1.8c has been applied.

### 9.2.2 Dornseiff: Topic thesaurus

For the detection of topics two language-specific lexical resources were created, which contain words and n-grams on different subject areas. For the German language, the corpus is based on Dornseiff, a German thesaurus arranged by subject groups (Dornseiff, 2004). Topics and subtopics have been extracted in the scope of this research. An English version of the topic corpus has been created manually from the German corpus to also cover translated topics and subtopics. In the implementation, a document is assigned to a topic and subtopic area according to the word occurrences and correspondences of the n-grams with the annotated corpora. The classification process based on n-gram occurrence works very well for longer documents. In the case of shorter documents or documents which frequently change the topic focus, the implemented topic detection is susceptible to errors, according to Zenkert et al. (2018). To eliminate the errors, a function has been implemented which provides the user with a notification for disambiguation on frequent occurrence of words from several topic areas. The list of topics based on Dornseiff (2004) is given in Table 9.1. A dictionary with translations of topics and subtopics from English to German and vice versa has been created and is used by the application to sort the analysis results into a uniform topic structure.

Topic	Number of Subtopics
nature and environment	25
life	43
room, position, shape	46
size, quantity, number	52
entity, relationship, event	47
time	35
visibility, light, color, sound, temperature, weight, aggregate conditions	68
location and location change	46
will and action	83
feeling, affects, character traits	60
thinking	56
sign, message, speech	63
science	27
arts and culture	24
human living	80
food and drink	22
sport and leisure	28
society	33
devices and technology	27
economy and finance	50
law and ethics	35
religion and spiritual	20

Table 9.1: Topic thesaurus (Dornseiff, 2004)

The administration of the topic thesaurus via KB:mkr is illustrated in Figure 9.1. Translation of topics expressions into other languages, here the German “*Natur und Umwelt*”, are managed programmatically.

info text analytics visual analytics knowledge discovery computational linguistics artificial intelligence

OVERVIEW TEXT SUMMARIZATION NAMED ENTITY RECOGNITION SENTIMENT ANALYSIS **TOPIC DETECTION** WORD ASSOCIATION SEMANTIC TRIPLE EXTRACTION

**LIST OF TOPIC FEATURES**

deDE Natur und Umwelt Kosmos Search Load

Topic	Subtopic	n Gram
Natur und Umwelt	Kosmos	Kosmos
Natur und Umwelt	Kosmos	All
Natur und Umwelt	Kosmos	Äther
Natur und Umwelt	Kosmos	Makrokosmos
Natur und Umwelt	Kosmos	Natur
Natur und Umwelt	Kosmos	Universum
Natur und Umwelt	Kosmos	Welt
Natur und Umwelt	Kosmos	Weltall
Natur und Umwelt	Kosmos	Weltgebäude

**DETAILS** [edit](#)

**Makrokosmos**

Language: deDE  
Topic Value: Natur und Umwelt  
Subtopic Value: Kosmos

**TRANSLATIONS** [edit](#)

Language (From)	Value	Language (To)	Translation
deDE	Natur und Umwelt	enEN	Nature and environment
deDE	Natur und Umwelt	frFR	La nature et l'environnement
deDE	Kosmos	enEN	Cosmos

Figure 9.1: Training process of entities from current selected article in KB:mkR

## 9.3 Validation through Linked Open Data

Recognized entities by the implemented method of NER are validated through LOD. Here, a query on DBpedia has been implemented as part of the validation process. In this way, each recognized entity is checked for its availability in the open source knowledge base. If the recognized entity is available in the external knowledge base, it will be used in MKR directly. If it cannot be found, it is still stored via MKR and provided for manual validation or blacklisting.

### 9.3.1 Entity Validation and Auto Classification of Entities

Based on a classification model, an entity classifier was implemented. The algorithm is able to classify the entities recognized due to the sentence structure by POS and validated by LOD. The information of the full text of the linked DBpedia entry is used for this purpose. With a training process, further entities can thus also be trained at runtime using the classification model. Figure 9.2 shows an example of the automatic training process of entities from current selected article in the implemented software. Figure 9.3 illustrates a notification by the software for manual classification and validation of a recognized entity from type “person”. Figure 9.4 completes the entity classification and updates all occurrences of the classified entity in all MKR in the knowledge base.

### 9.3. Validation through Linked Open Data

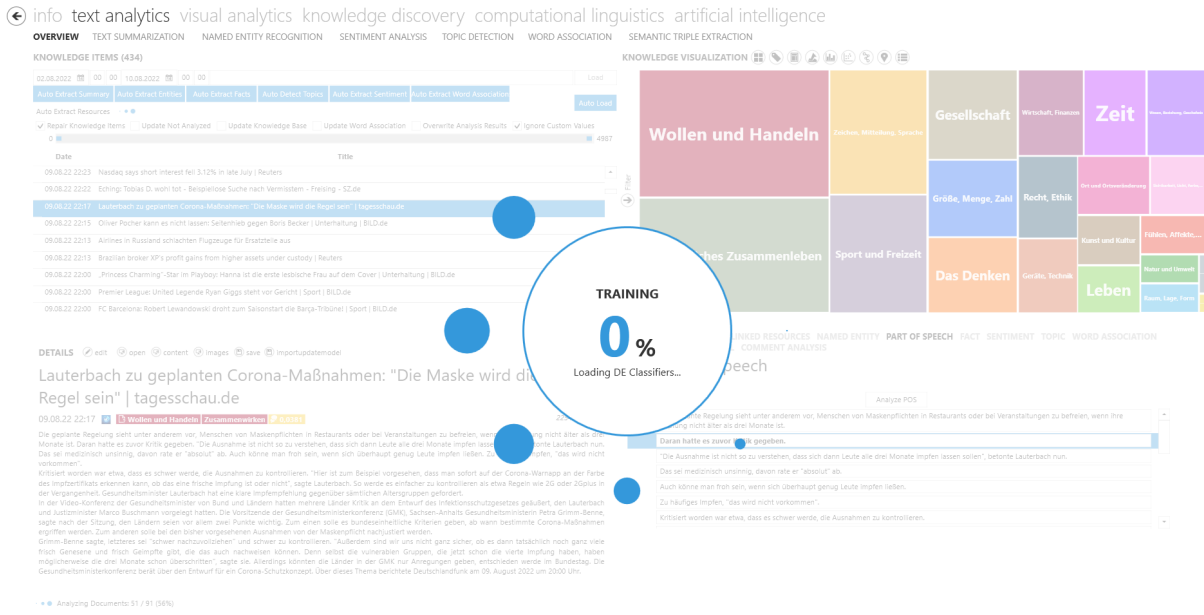


Figure 9.2: Training process of entities from current selected article in KB:mrk



Figure 9.3: Entity validation and classification in KB:mrk - Classification of entity type "person"

### 9.3. Validation through Linked Open Data

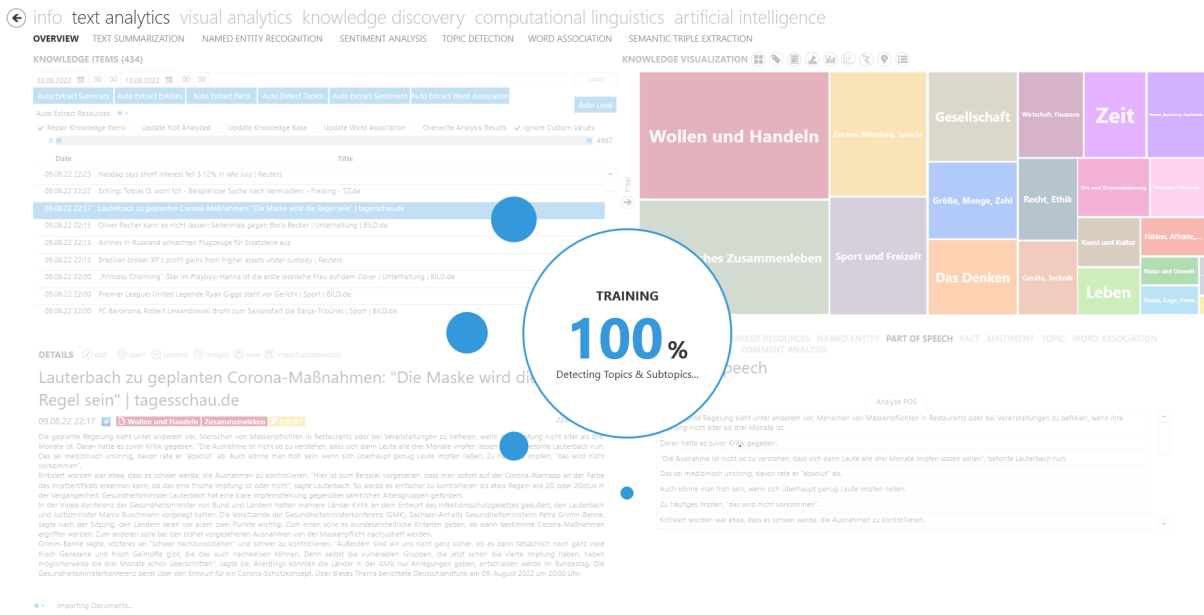


Figure 9.4: Knowledge Base Update in KB:mkr - Entity update



## 10 Text Corpora: Data Collection and Pre-Processing

For the purpose of evaluation of this dissertation, news articles have been crawled from various websites of news portals. These include for German, among others, the *Süddeutsche Zeitung*, *Zeit*, *tagesschau.de*, *Spiegel*. For English language, for example, articles from BBC, *reuters.com* and CNN have been crawled.

### 10.1 Web Crawling and Web Data Collection

Textual news articles from the WWW serve as the data basis for the KB:mkr software.

As Zenkert et al. (2018) defines, many news portals have a similar structure. Typically, news sites often have a menu bar with different categories. Articles are often assigned to individual categories. In addition, there is often an overview page with the current headlines, in which articles of all categories are entered. In addition, the respective articles are usually arranged chronologically. Although the structure of news portals is basically similar, websites can be described as unstructured overall. All news portals differ in their internal structure. As mentioned in Zenkert et al. (2018), the insights gained cannot be transferred to other news portals. This is true despite the fact that the respective HTML file of the website has a structure. The pages also have similarities with the respective articles. There is often a headline, a short summary and the actual text. Again, the internal structure is always different. To make matters worse, the actual text can also be considered unstructured. Texts, in turn, can be represented by various HTML elements.

In order to target the text of web pages, it is therefore necessary to take a closer look at the structure of the individual news pages. Then the parts of the web page that are useful for selection of text, etc. can be determined. This can be done, for example, using the XML Path Language (XPath). Also Cascading Style Sheets (CSS) selectors are useful to select specific website elements. If these options are not available, a reference to the used HTML elements is possible (Zenkert et al., 2018).

### 10.2 Extraction of Web Structures

For the development of a prototype in the field of web mining, it is mandatory to know and use the structures on the web. The basis of an ordinary web page is an HTML file. Such a file is a text file that is subject to a fixed structure. This structure is created by the HTML code contained in it. The HTML code is used to create static web pages. It defines the elements of a web page. The visual modification of the elements is often done by CSS. If a web page is to contain dynamic elements and respond to user input, for example, this can be added using JavaScript (JS) code. (Zenkert et al., 2018)

The structure of HTML files is determined by so-called tags. Usually there is an opening and a closing tag for each element. These are keywords enclosed in angle brackets. Closing tags also contain a slash. The HTML file starts or ends with an `<html>` or `</html>` tag accordingly. The HTML file is divided into a header and a body part. The header part usually contains data that can be used for machine evaluation, e.g., by search engines. The web pages can be retrieved from the server and displayed via a browser. A web crawler is a program that automatically retrieves web pages and evaluates and stores the data received. The program determines the contained links and retrieves the respective web pages in turn. The links are located in HTML files in `<a>` tags. The tags

contain an href attribute, which contains the URL of the link. The URL is the destination address that is actually required. URLs must distinguish how a target is addressed. There are absolute and relative paths as URL. The absolute paths contain a complete addressing of the target. The relative paths describe the target address relatively from the current web page. For a storage and a later access to the links it is advantageous to transform them directly into absolute paths. In addition to the type of a path, the destination must also be considered. Any resources can be addressed on the web by an URL. For a web crawler, however, only web pages are of importance as a target in the context of this work. All other links must therefore be filtered out. Links that point to a web page often end in corresponding file extensions such as .html, .htm, .xhtml or .php. However, there are also URLs that have no file extension or end with a slash. Such links also usually lead to web pages.

### 10.3 Data Storage Architecture

#### 10.3.1 Web Crawling with R - Dynamic Web Crawling

As mentioned in (Zenkert et al., 2018), due to the individual, dynamic structure of web pages in the WWW, it is hardly possible to know where the actual headline, body, date, author name or other metadata from a published (textual) news is located on a specific website. For example, some pages use simple <p> elements (text paragraphs), others use <div> elements (block of paragraphs and multiple elements, such as images, tables, etc.) to define parts of the HTML document. Elements normally also use class or id attributes in conjunction with CSS to apply a defined layout or style on the web page. Consequently, in order to extract text information from a website it must be a priori known, in which HTML tags or CSS elements the actual message text can be found. Moreover, the news text can also spread over several elements.

In general, there are two ways to solve the generic web crawling problem. Either a separate web crawler must be developed for each news portal, specifically adapted to the characteristics of the web page, or a generic crawler must be used, which stores all elements of a web page and, if available, all associated information such as CSS identifiers. The disadvantage of the first approach is that as the number of different news portals increases, many crawlers or crawler variants would also have to be written, updated, or even adapted when the structure of the portal or web pages changes. The second approach is a generic web crawling approach that can be applied to almost any CSS-designed web page. Crawling is based on the selection of CSS elements. Each available class or id attribute of the CSS formatted HTML elements on a web page are used as a selection parameter in XML Path Language (XPath) queries. This leads to the extraction of various larger meaningful pieces from web pages. Of course, a lot of unnecessary content is also extracted this way. However, the rule-based cleaning and pre-processing of documents resulting from the second crawling approach are simpler than manual adaptation of different web crawlers depending on the target news portal or even page-level adaptation.

#### 10.3.2 Crawling Procedure

The crawling algorithm applied by Zenkert et al. (2018) is explained as follows: The crawler first retrieves a summary of lately published news articles from the news portal's RSS feed. Secondly, articles which are missing in the MongoDB database are loaded (if crawling is allowed on the web page at all) with the headless browser PhantomJS (<https://phantomjs.org/>) in order to fully load the content. Thirdly, web page's metadata (e.g., the favicon of the page, the time of creation, the title) and all contained links are retrieved from the web page's source code by means of using regular expressions. All CSS style identifiers are determined in the next step, whereupon

the associated textual content is selected by an XPath query and stored in a temporary list. For the identifiers to be unambiguous, corresponding endings are added in case of multiple occurrences. Finally, all the extracted information is evaluated for length and a general heuristic to detect the main article of a web page is applied. Afterwards it is stored in Binary Java Script Object Notation (BSON) format in a MongoDB document collection. The crawling procedure is the first step from the overall workflow of the prototype visualized in Figure 10.1.

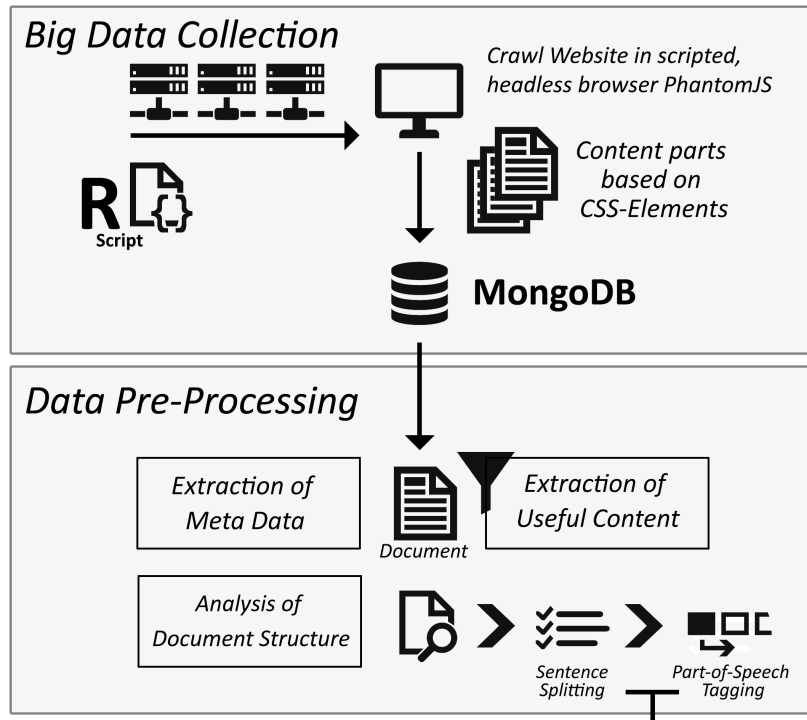


Figure 10.1: Web Crawling Process, Data Storage and Pre-Processing Architecture (Zenkert et al., 2018)

## 11 KB:mkr: Knowledge Base Maker

This chapter presents the main implementation of the dissertation. Previously mentioned methods of the text mining pipeline and all described knowledge extraction methods were implemented in the so-called KB:mkr software. All results are combined into the MKR format which is stored in a document-oriented database. For the interaction with the knowledge base, KB:mkr software provides all needed interfaces.

### 11.1 Technology Overview

This section provides a brief introduction to the technologies used, by which is meant primarily the programming language and environment, and the database technology. Additionally, information about the main programming design pattern of the software is provided.

#### The programming language C#

C# is a modern, object-oriented, and type-safe programming language, which originates from C. It is particularly well known because of the “.NET Framework” from Microsoft. Therefore C# is often used especially for applications for Windows. Many functions are already supported directly by the language, including exception handling and the release of unused objects, also known as “garbage collection” (Hejlsberg et al., 2006). A further advantage is the large similarity to other programming languages such as Java, whereby the familiarization is clearly easier, if one already has experience with these.

#### Microsoft Visual Studio 2022

Visual Studio 2022 from Microsoft is used as the programming environment. It offers various convenience functions, which include a debugger for troubleshooting and automatic correction. In addition, the GUI can be designed within Visual Studio with the help of Windows Presentation Foundation (WPF). This is an extensive class library, which represents a powerful tool. Despite many design freedoms, it includes all the familiar elements that users of Windows are already familiar with (MacDonald, 2010). The necessary components are already integrated in the .NET framework and therefore do not have to be installed separately.

#### MongoDB

MongoDB is a NoSQL database technology without relational database schema, as it is known from SQL databases. This results in several advantages, but the indexing of documents and processing of large amounts of data is one of the most important (Chodorow, 2013). Finally, larger documents can be stored and loaded without affecting performance, as would it be the case with relational databases due to the JOIN operations.

#### Model View ViewModel

The MVVM is a design pattern designed for use with WPF. It can be used to structure an application much better, as the presentation and the program logic are better separated, which also simplifies

team collaboration. While designers place the necessary elements on the graphical interface, developers can take care of creating the necessary classes for the program's functions (Sorensen and Mikalesc, 2010). Basically, MVVM distinguishes between three important parts, which are structured as follows according to Kühnel (2012):

- **Model:** It contains the data of the application, which in this context includes the messages and their contents.
- **View:** This part contains everything that is ultimately displayed on the screen. However, parts of the program logic may also be included. A mouse click for example is often handled directly in the view. Otherwise, however, the view deals exclusively with the representation of the contents for the user and the provision of the control elements. and the provision of the control elements.
- **ViewModel:** It is located between the model and the view and thus represents the connection between the two parts. The tasks include preparing the data from the model for the view. Likewise, however, data entered in the view must be received in order to store it in the model or to use it further.

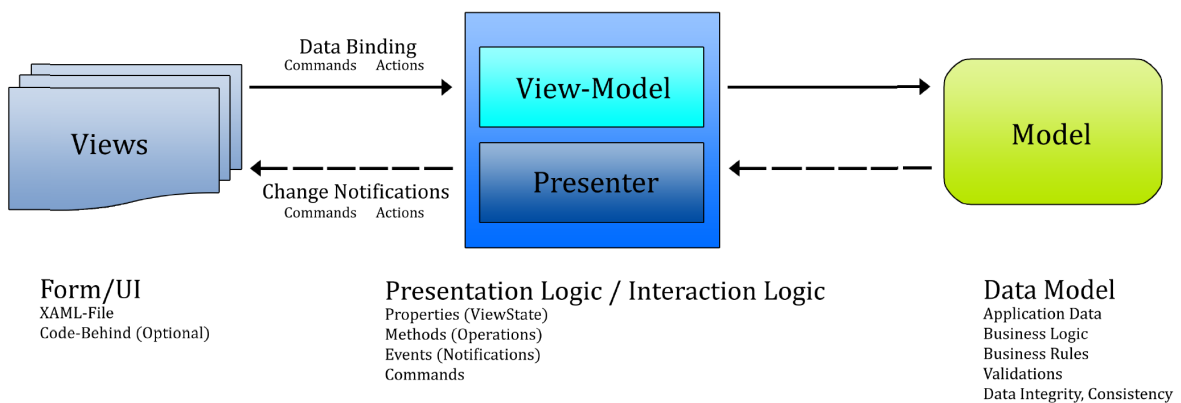


Figure 11.1: MVVM - Abstract View

Figure 11.1 shows the interaction of the three parts in the MVVM. This also illustrates the central task of the ViewModel and shows which data is exchanged at which point.

## 11.2 Software Implementation

The implementation KB:mkr of this research was created in C# and the WPF framework. For the data collection task, a script written in R has been used to externally crawl German and English news portals on distributed machines and storage in a NoSQL database (MongoDB) (see Section 10.3.1). PhantomJS (a headless browser used for automating web page interaction, <https://phantomjs.org/>) is executed by the R script to load and iteratively crawl the content of websites.

A test dataset was created for the period from September 2016 to July 2022. The dataset is about 39.7 GB in size and contains around 2.3 million articles, along with images and pre-processed content from various typical news domains (e.g., finance, business, politics, sports, lifestyle, culture, science, and others) with an average length of 5,227.33 characters and 562.33 words per

article. More than 70 million co-occurrences between 3.2 million entities have been analyzed. Additionally, over 200 million tokens from the analyzed text are individually stored to calculate changing association strength over time. The languages German and English are selected to analyze how the implemented methods of the integrative text mining approach can be adapted or directly applied by both languages (Zenkert et al., 2018).



Figure 11.2: KB:mkr Knowledge Base Maker Logo

The screenshot shows the KB:mkr Knowledge Base Maker interface in text analytics mode. At the top, there are navigation tabs for 'info', 'text analytics', 'visual analytics', 'knowledge discovery', 'computational linguistics', and 'artificial intelligence'. Below this, a 'KNOWLEDGE ITEMS (256)' table lists various news items with columns for 'Date' and 'Title'. One item is selected, showing its details in the bottom left. On the right, a 'KNOWLEDGE VISUALIZATION' bubble chart shows the distribution of items over time. Below the chart, a 'PART OF SPEECH' analysis tool is active, showing a selected sentence from the knowledge item and its corresponding POS tags.

Figure 11.3: KB:mkr Knowledge Base Maker User Interface - Text Analytics Module

Figure 11.2 depicts the created logo of the software. Figure 11.3 shows the interface of the implemented prototype in text analytics mode. The user interface contains multiple modules. On the left side, knowledge items can be loaded from the knowledge base. Details from selected item are shown in the bottom left area. Visual elements are for example, the title and body of a document from the knowledge base. On the bottom right side, additional functionality from the implemented analysis methods is provided. In Figure 11.3, the POS tagger is loaded and a selected sentence from the knowledge item is analyzed.

Figure 11.4 illustrates the whole implementation workflow for KB:mkr. More specific descriptions of the implementation parts are given in the following sections.

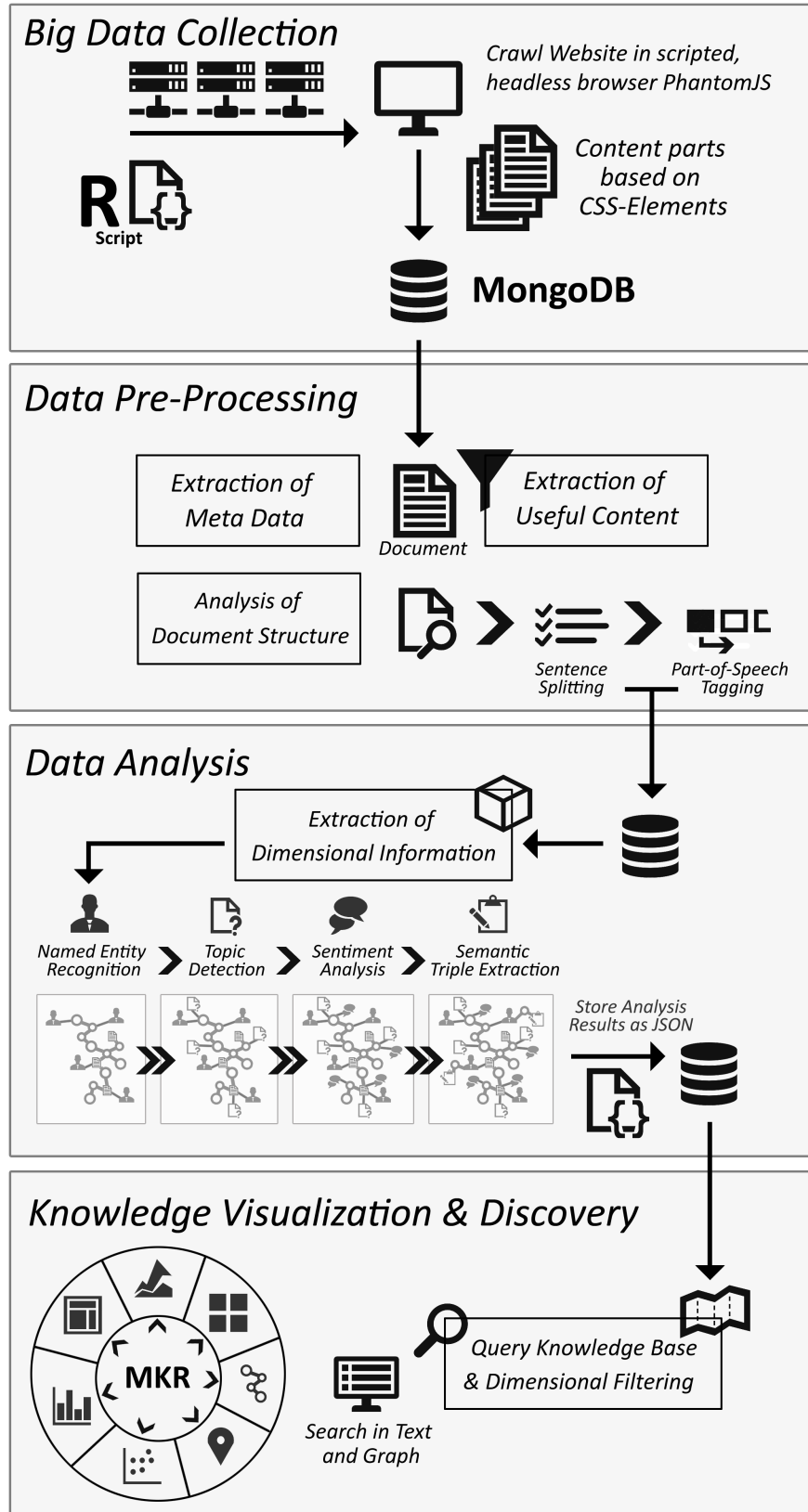


Figure 11.4: Implementation Overview and KB:mkr Knowledge Base Maker Architecture (Zenkert et al., 2018)

### 11.2.1 Data Analysis

In the data analysis step illustrated in 11.4, presented in (Zenkert et al., 2018), pre-processed documents are analyzed by the introduced NLP and text mining methods in order to extract dimensional information for the MKR structure. Here, the metadata and the document structure are considered carefully to split sentences and to perform the POS tagging. The data analysis in the implementation follows the steps of the MKR process which have been explained in Section 6.4.1 and visualized in Figure 6.4. The implementation details of NER, sentiment analysis, topic detection and semantic triple extraction are given in the following sections.

### 11.2.2 Named Entity Recognition

Named Entities are first recognized within documents based on the POS tags in splitted sentences. The POS tag sequences coming from the pre-processing are further analyzed in a second step to combine related words, all tagged as named entity candidates, into one entity. Then, a lexical resource and a reference to the German or English LOD DBpedia are used to evaluate and distinguish possible proposed named entities. The language-dependent resources contain the entity categories date, place, organization, person, and other. The identified entities are classified into the categories using a classification approach (see Section 9.3). Blacklisting of incorrectly tagged named entities is also possible within the application to improve the quality of the named entity collection. The entities of each document can be selectively edited, updated, or deleted to make manual adjustments.

### 11.2.3 Sentiment analysis

Sentiment analysis has been implemented in the application via Associative Sentiment approach (see Section 4.3.1.1) to identify co-occurrences, relevant words and n-grams which express a positive or negative polarity. Adjectives and adverbs are recognized based on POS tag information and language-specific lexical resources are used to evaluate them. In the English language, polarity shifters (“no”, “not” or the ending “n’t”) before adjectives or adverbs, are recognized within sentences and the corresponding sentiment values are inverted for negation handling. In the German language the token “nicht” is another typical example for a polarity shifter. Sentiment intensifiers are used in the prototype’s sentiment analysis to increase the sentiment value of several words or n-grams in the positive or negative sense. (Zenkert et al., 2018)

In addition to the regular sentiment analysis, an emotion classifier was included in the implementation to further distinguish the sentiment scores according to their specific emotions. Eight categories are used for emotions, namely joy, sadness, anxiety, fear, trust, disgust, surprise, and anticipation. Figure 11.5 illustrates an emotion classification using an example document.



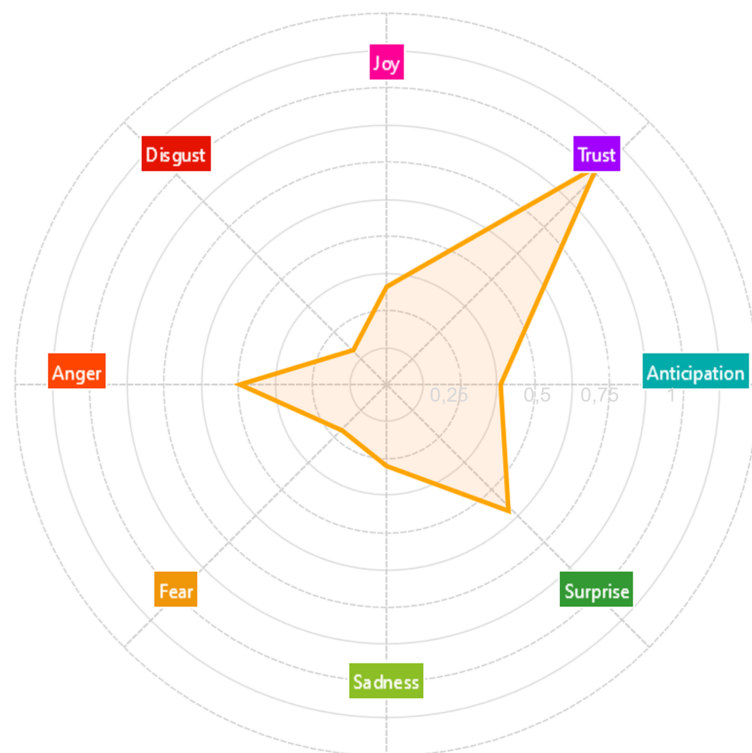


Figure 11.5: Exemplary Sentiment Evaluation based on Document's Emotion Classification (Normalized Scale from 0 to 1) (Zenkert et al., 2018)

#### 11.2.4 Topic detection

For topic detection, two language-specific lexical resources were created, containing words and n-grams on different topics. For German, the corpus is based on the Dornseiff (Dornseiff, 2004) (see Section 9.2.2), a German thesaurus organized by topic groups. The topics and subtopics were extracted as part of this thesis. An English version of the topic corpus was manually created from the German corpus to include translated topics and subtopics. In the implementation described in (Zenkert et al., 2018), a document is assigned to a topic and subtopic area based on the word occurrences and correspondences of the n-grams with the annotated corpora. The classification process based on the occurrence of n-grams works very well for longer documents. For shorter documents or documents that frequently change topic focus, the implemented topic detection is prone to errors. To eliminate the errors, a feature was implemented to provide the user with a disambiguation notification when words from multiple topics occur frequently.

#### 11.2.5 Semantic triple extraction

For semantic triple extraction, the approach from Akbik and Broß (2009) (see Section 4.5.1) is applied in the application to the pre-processed sentences. To support the POS tagger, a chunker was implemented to identify noun phrases (NP), proper noun phrases (PNP), and verb phrases (VP). For each phrase, it is checked whether two NP or PNP are included. In this step, the results of named entity recognition are used. If two such phrases are present, rules determine whether they are extracted as subject and object with an associated predicate.

The details of the extraction process mentioned in (Zenkert et al., 2018) are: First, cases where NP or PNP pairs have been detected in the subordinate clause are removed from the analysis set. The extraction is also interrupted for certain punctuation between phrases. This applies analo-

gously to text segments in parentheses. Subsequently, there must be at least one verb between the potential subject and the object, and there should be a maximum of five words between them, so that a direct semantic connection is likely. If a NP is followed by a PNP, this is chosen instead of the NP, for a maximum of five words. This number has been found to be satisfactory during development, as larger spacing sometimes led to false associations and smaller spacing led to missing information. When these decision criteria are met, a predicate is formed from the words between subject and object, omitting all articles. Another rule-based filter, similar to the one presented in (Akbik and Broß, 2009), is applied to the formed predicate in the final step to check whether it represents a semantic relation (e.g., relation of type “is-a”).

## 11.3 Implementation of Representation Format

After the text mining methods are applied and the results are collected, the MKR is created by the implementation and inserted into the knowledge base. Therefore, the MKR is stored in MongoDB using an JSON format. Figure 11.6 describes the MKR structure for a referenced document in the knowledge base, with all named relationships shown as examples. A timestamp is created for all relationships in order to use time as another dimension of analysis and necessary information for various visualizations.

In the structure shown in Figure 11.6, the results of the previously mentioned sentiment analysis, topic detection, entity recognition and semantic relation are presented. The sentiment n-grams are tagged with an `ObjectId` to further analyze the relationships between documents and entities with similar polarity. The relationship between the sentiment-n-gram and the entity information (both referenced by `ObjectId`) is further described with additional details about intensification or sentiment shift in negation cases.

The multi-topic relation is represented by referenced topic names and subtopic names.

The entity relation lists named entities from the document with their name, type, document language, and creation time to allow temporal analysis and possible analysis of entity occurrence over time.

Semantic relationships between entities of the document are presented in the last part of the MKR representation in Figure 11.6. Here, the subject, predicate, and object of an RDF triple are stored in the knowledge base along with the timestamp and associated identifiers to allow further analysis of the similarity and correspondence of entity references.

Figure 11.6 visualizes the integration of analysis results into MKR with colors to specific knowledge extraction methods (red color for sentiment analysis results, orange color for topic detection, green color for named entities and blue color for semantic relationships).

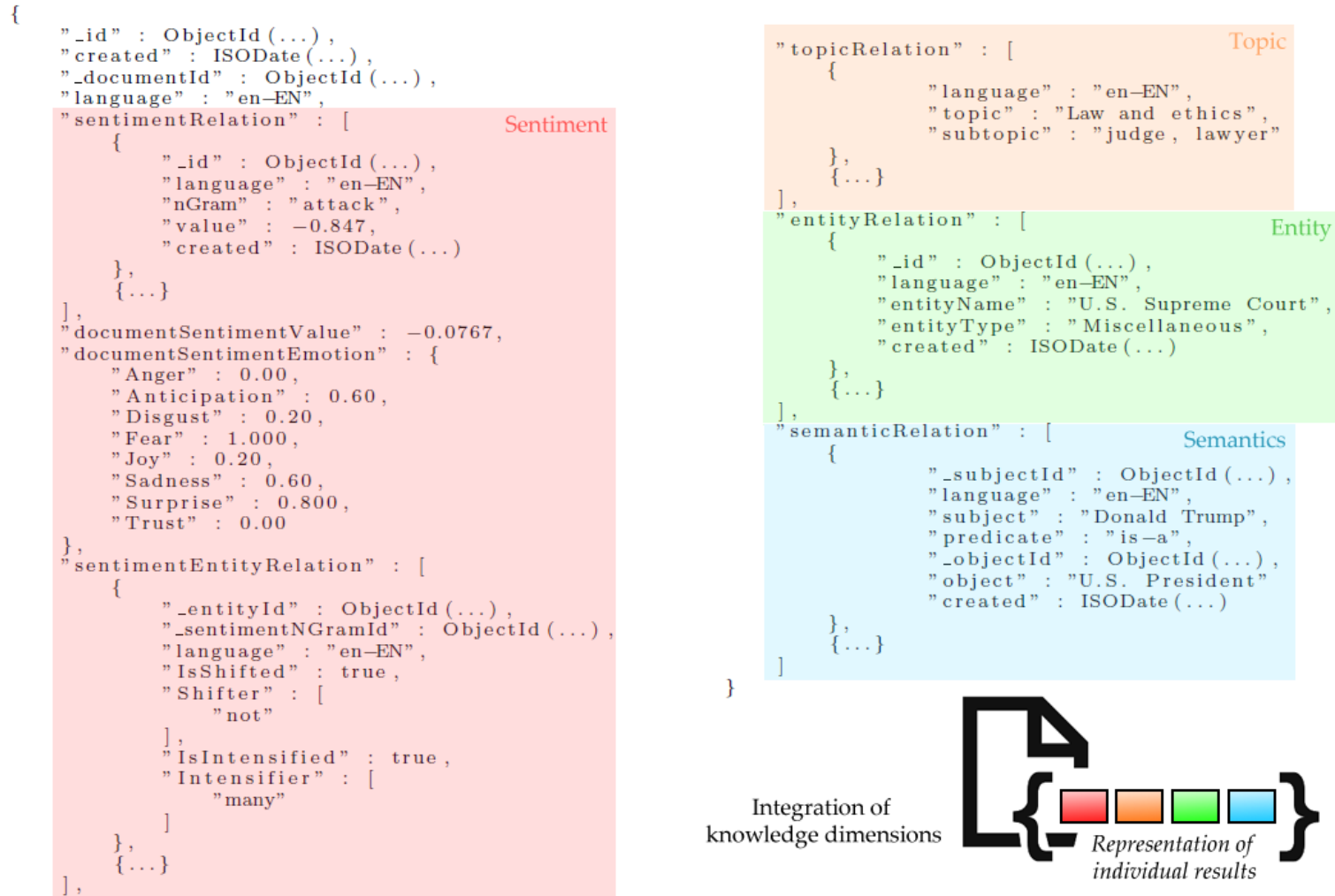


Figure 11.6: JSON Format of MKR (Colored areas indicate integrative text mining results) (Zenkert et al., 2018)

## **Part IV**

# **Experimental Results, Case Studies and Evaluation**

## 12 Exploratory Analysis and Evaluation of Text Corpora

This chapter deals with the evaluation of the prototype implementation. The web crawled data collected between September 2016 and July 2022 is evaluated in various research directions, topics, and use cases. The text corpus has been used to illustrate several visualizations and analysis options have been applied which were discussed as analysis dimensions in the previous chapters.

### 12.1 Knowledge Discovery Interface

MKR is particularly important for graphical search and knowledge discovery process in knowledge-based systems (Zenkert et al., 2018). By integrating different analysis results from text mining methods and making them available (pre-calculated results), MKR enables the visualization of different types of knowledge graphs in a multidimensional perspective. Importantly, additional analysis results from other dimensions can be integrated into the knowledge visualization if desired.

MKR operations and transformations, namely dimension selection and filtering, customize visualizations for specific queries. Furthermore, knowledge graphs can even be replaced by other visualizations and still integrate the same dimensions into the graph.

Figure 12.1 shows different components that can be used in knowledge discovery. The different types of knowledge visualizations (selectable in the red highlighted area) provided by the prototype support the user in finding and visualizing the desired information.

The previews of additional (not necessarily expected or desired) results from MKR next to the current visualization (purple) contextualize the information and show results from other dimensions or further possibilities for dimensional filtering or selection.

When such a filter is selected, the knowledge base query is automatically adjusted, and possible unexpected results are provided. In this way, the knowledge discovery process is exploratory and different information visualizations can be mapped into each other to adjust the knowledge graph to display suggested or desired results. (Zenkert et al., 2018)

Knowledge discovery results are always provided in the KB:mkr as a list of knowledge items (e.g., documents) and individual documents can also be selected to read, modify, or export details to support the user's knowledge discovery process or information search.

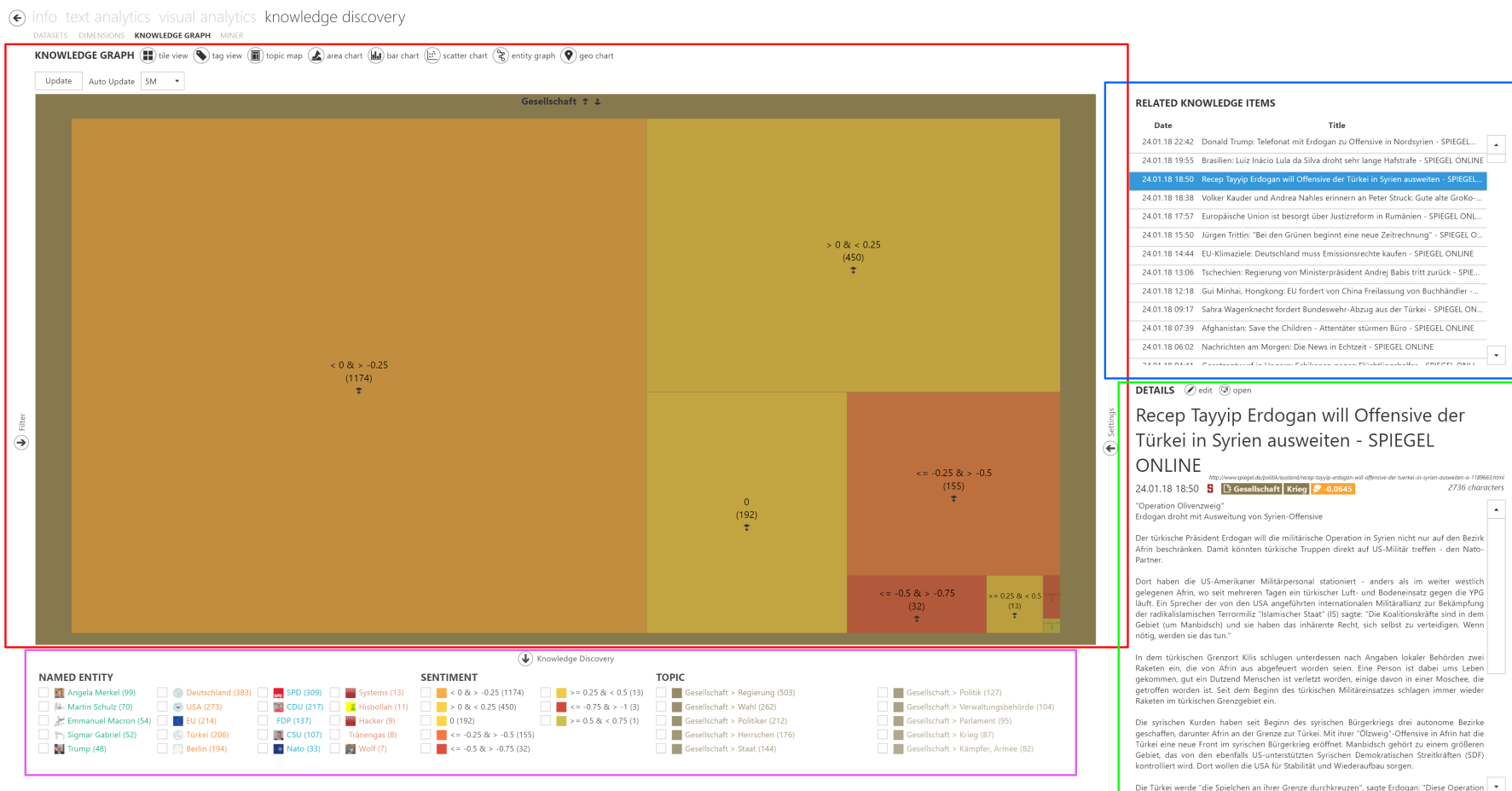


Figure 12.1: Knowledge Discovery Components: Knowledge Graph (red), Related Knowledge Items (blue), Details of Selected Knowledge Item (green) and MKR Results from Named Entity, Sentiment, Topic Dimension in Preview (purple). *The Screenshot of the prototype shows a subset of SPIEGEL Online articles related to the topic "society", organized in a topic map visualization using the document's sentiment dimension as layout arrangement.* (Zenkert et al., 2018)

The data set which contains articles from various news domains in German and English language has been analyzed by the presented text mining methods. The results have been integrated in the presented MKR structure for knowledge discovery and information visualization purpose. Each of the around 2.3 million articles has been analyzed for topics (and related subtopics), sentiment on different levels, named entities (with types of person, location, organization, and miscellaneous) and semantic relationships in RDF triple format. Additionally, the analysis of documents into the MKR also creates a text summary and stores it into the MKR.

## 12.2 Exploratory Analysis: Topic Development

A typical analysis option is to observe topic development over time. For example, a bar chart can be used for this purpose, which can display the assignment of documents to topic areas over a selected period of time. Figure 12.2 shows an example of the performed analysis. Colors indicate different topics, related entities and sentiment evaluation are shown below the bar chart in the knowledge discovery area.

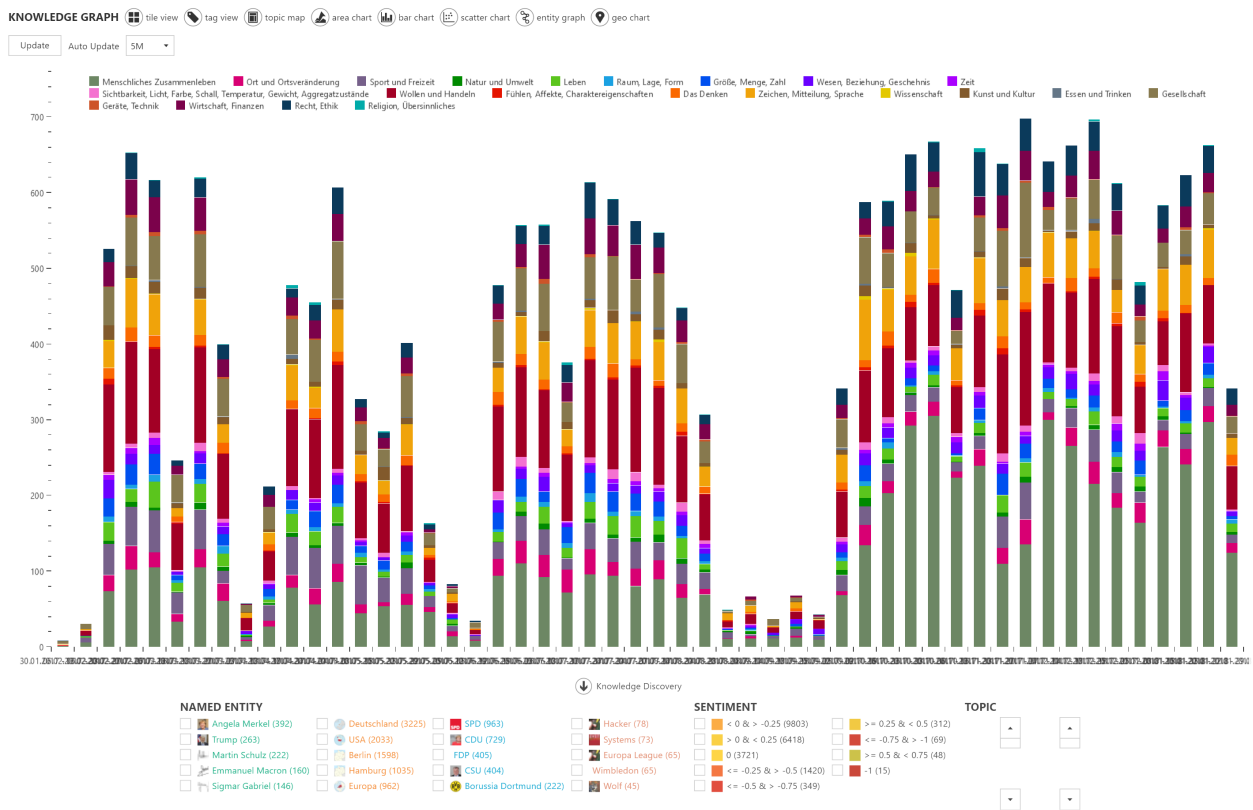


Figure 12.2: KB:mkr Bar Chart on topic development

The tile view from KB:mkr offers a different view. Here, in Figure 12.3, for example, the frequencies of related subtopics are displayed in the respective tiles of the topic. Size of tiles indicated the amount of contained documents.

## 12.2. Exploratory Analysis: Topic Development

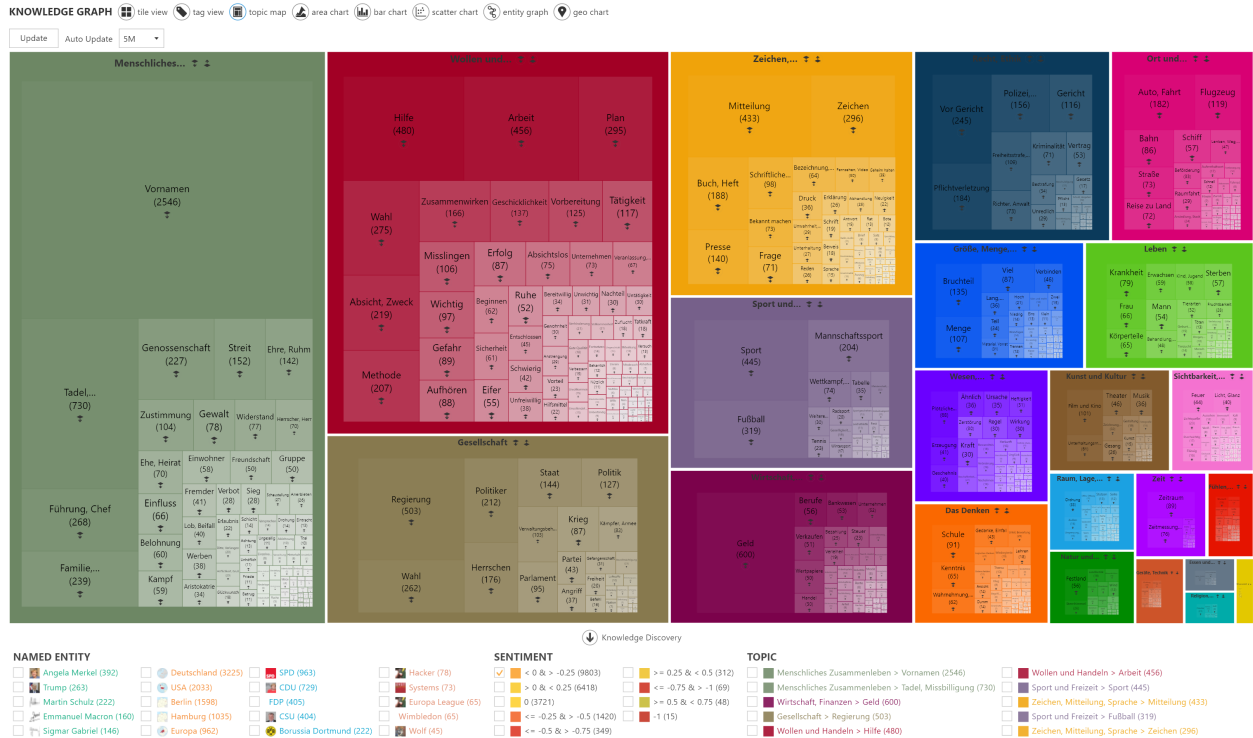


Figure 12.3: KB:mkr Tile View on topics with subtopics

Results from previous work (Zenkert et al., 2018) showed an average of 9.45 extracted named entities per document on a smaller text corpus. The most assigned topics were “human living” and “will and action”. According to the topic detection analysis results, the corpus contained the lowest amounts of documents in the domain of “science” and “food and drink”. A total amount of 3131 semantic relationships were extracted from the documents by further analysis of the named entities. The average sentiment from the overall analysis was slightly negative with a value of -0.002. Documents assigned to the topics “religion and spiritual” had the most positive sentiment values in average (+0.064), whereas “law and ethics” related documents had the most negative sentiment values (-0.072) (Zenkert et al., 2018).

By using MKR, it is possible to perform a “drill-down” operation to filter the document corpus based on the topic selection. The GUI in KB:mkr then presents an overview of the "society" area and thus focuses the analysis on entities contained in this topic area. The size of the individual tile indicates the frequency of entity occurrence. The Figure 12.4 illustrates the result of the operation. Here, entity “Angela Merkel” occurs 99 times, “Martin Schulz” occurs 70 times, and “Emmanuel Macron”, for example, occurs 54 times as an entity in related documents.

The “roll-up” operation in KB:mkr reverses the entity view and returns to the topic overview of MKR from related documents.



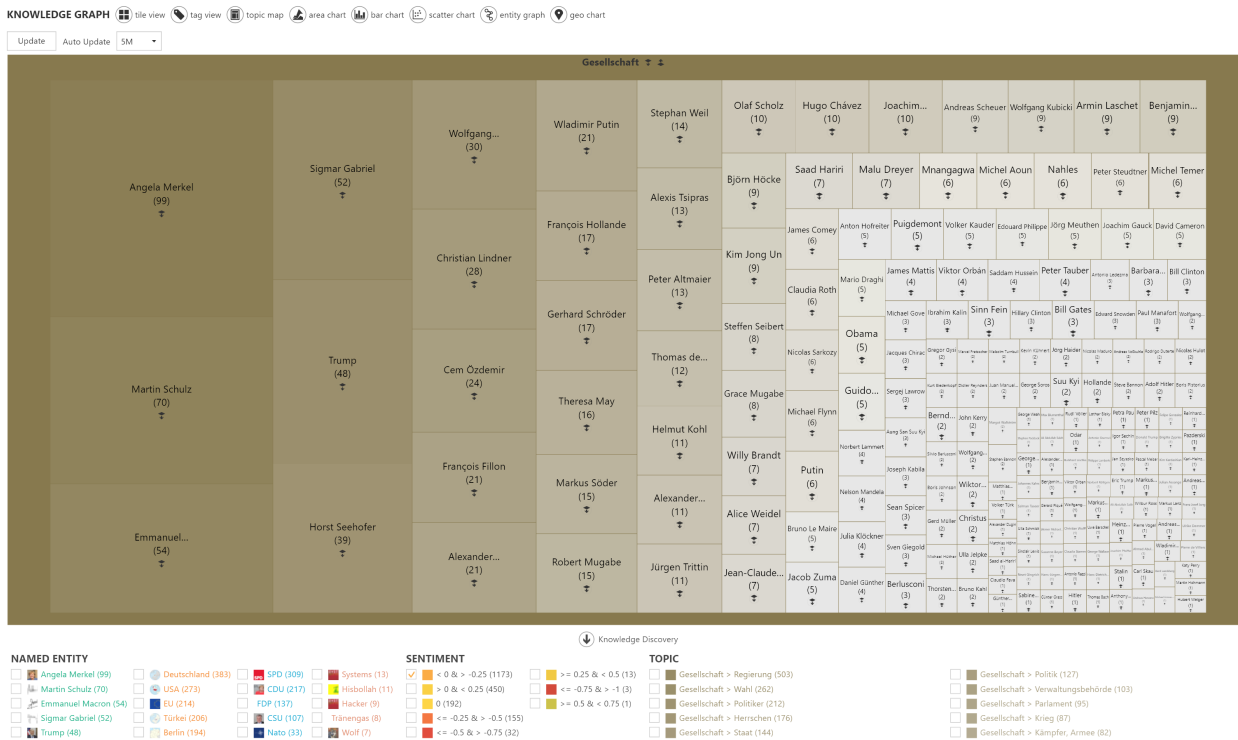


Figure 12.4: KB:mkr Tile View drill-down on topic “society” - Overview of related entities

## 12.3 Exploratory Analysis: Sentiment Analysis

Sentiment analysis can be applied to various other dimensions, such as to time as well as to topics. A color-oriented visualization allows insights into the sentiment analysis to be determined very quickly. In the following, the temporal sentiment analysis is presented first. Subsequently, an entity-related sentiment analysis is performed.

### 12.3.1 Temporal and Entity-related Sentiment Analysis

Two further examples are demonstrating the application of sentiment analysis in KB:mkr. Here, Figure 12.5 shows an analysis possibility that arranges different articles over time in an area diagram that shows the intensity of the sentiment analysis. Individual articles can be selected, and details are given via a tab menu in the analysis section in the lower right part of the KB:mkr software.

Additionally, an implemented Map View, shown in Figure 12.6, uses the same data for analysis of the entity information on entity type “locations” to colorize the mentioned countries in the world map according to their related sentiment value.

## 12.4. Exploratory Analysis: Named Entity Recognition

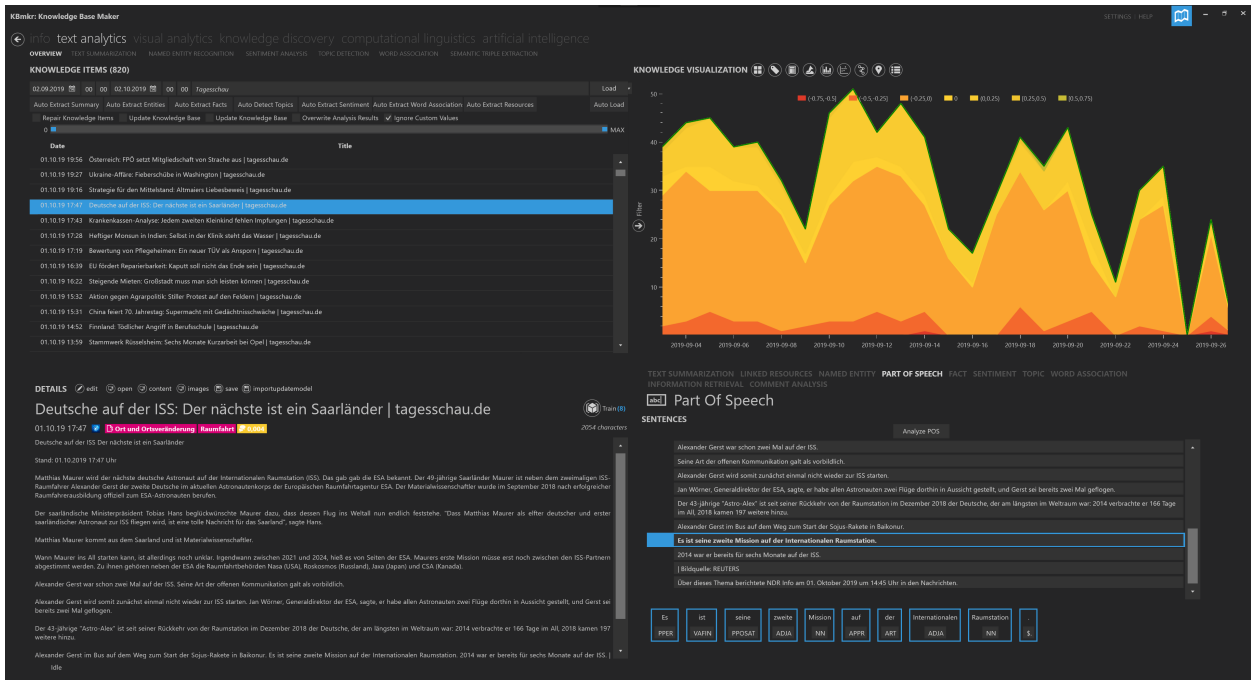


Figure 12.5: KB:mkr Area Graph on Sentiment Development

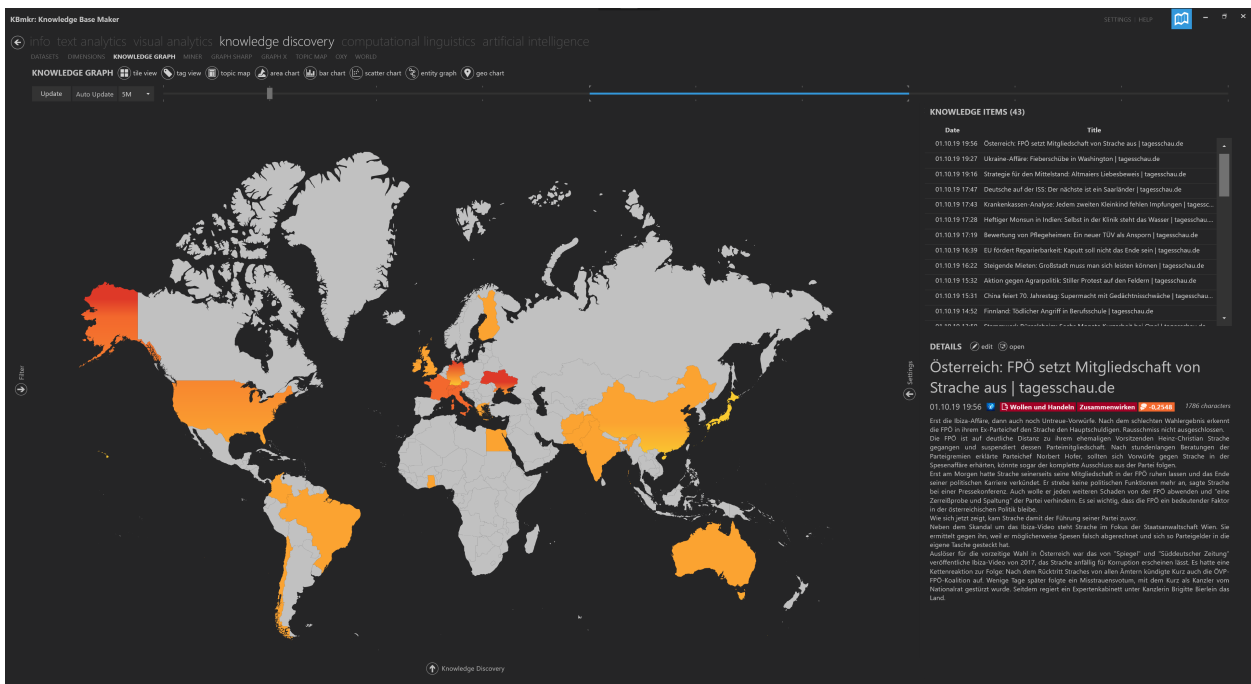


Figure 12.6: Sentiment analysis on topics on 01.10.2019 - Map View

## 12.4 Exploratory Analysis: Named Entity Recognition

In this section, the analysis of entities, resulting from the integration of NER into MKR is focused.

### 12.4.1 Entity Overview

A possible application of KB:mkr is the overview of named entities which are contained in the data set within a selected time period. Figure 12.7 exemplarily visualizes the entities contained in the selected data set. The size is depending on their frequency of occurrence and information (e.g., symbol or image) obtained from the LOD. Filtering can be performed as operation, for example to only select entities of the “organization” entity type. This filtered result is shown in Figure 12.8.

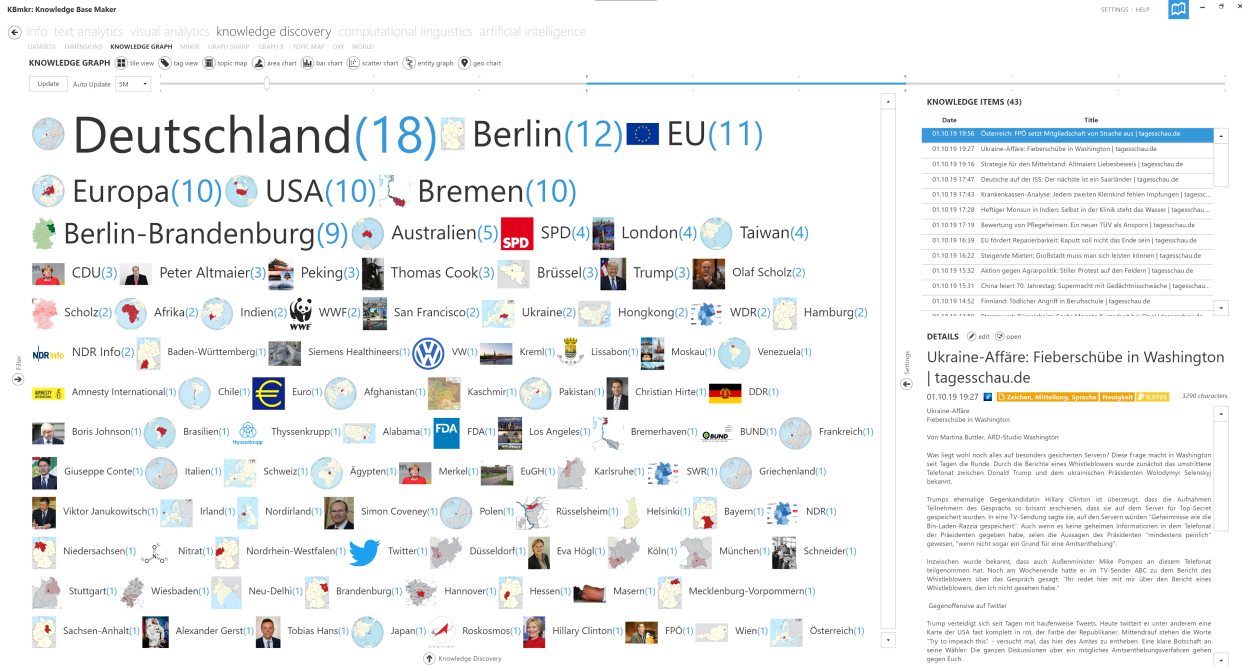


Figure 12.7: KB:mkr Entity Overview

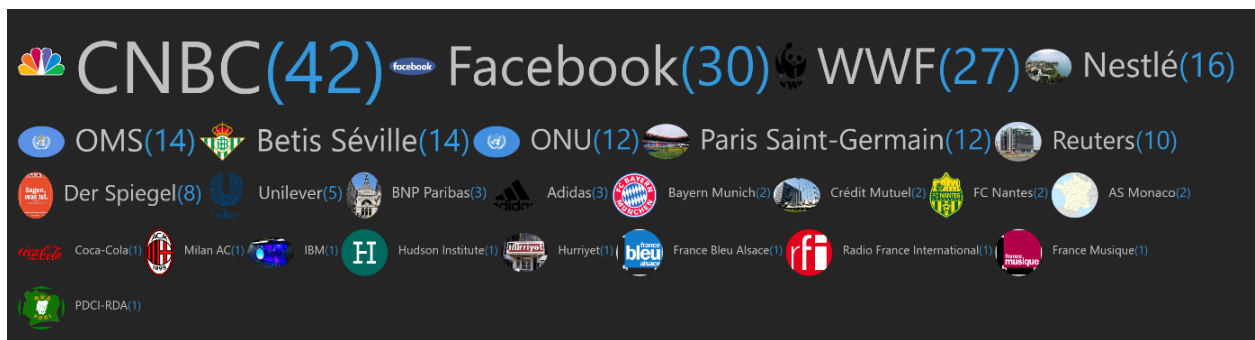


Figure 12.8: KB:mkr Entity Overview - Filter operation on entity type “Organization”

## 12.5 Exploratory Analysis: Word Associations

In this section, an example is given for the temporal word association, which is calculated continuously by the KB:mkr software for all extracted co-occurrences. Figure 12.9 illustrates the example of word association strength based on CIMAWA between “Ukraine” (DE: Ukraine) and “Russia” (DE: Russland). Of course, as there have been much more news reports about the war which started in February 2022, the word association strength based on CIMAWA has increased strongly.

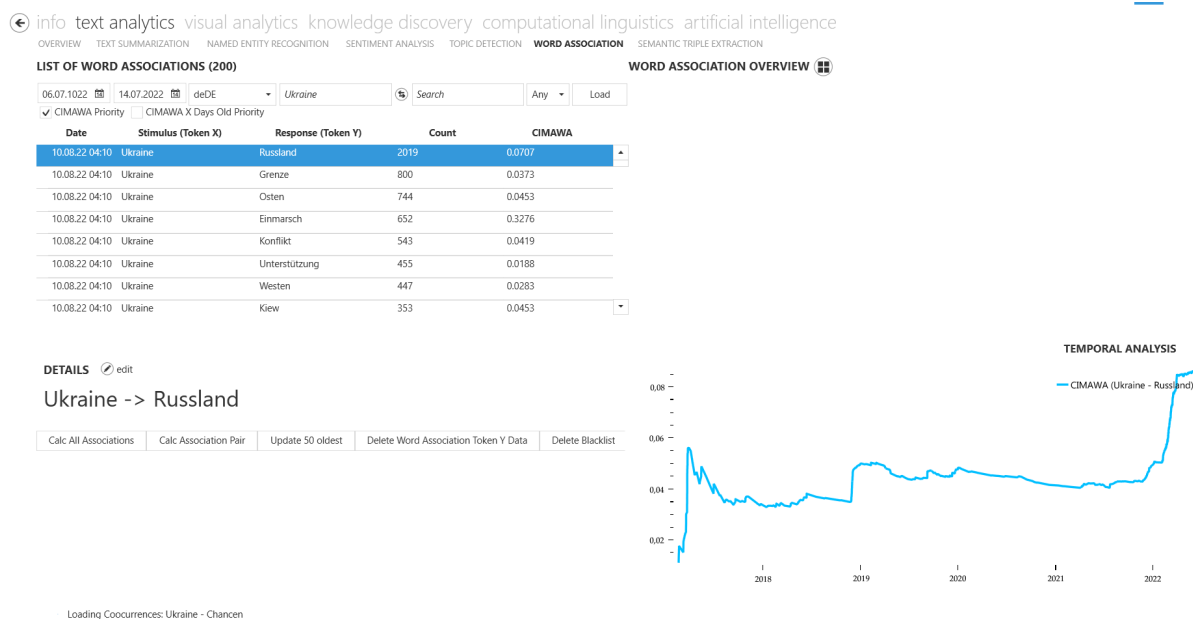


Figure 12.9: KB:mkr Temporal Word Association Strength between Entities “Ukraine” (DE: Ukraine) and “Russia” (DE: Russland)

## 12.6 Full-Text Associative Search

According to Zenkert et al. (2016), searching for information is often a very time-consuming task. Unstructured document-based repositories often offer only keyword-based content searches, which can lead to a large number of search results. It is even more difficult to search for the desired information within the initial filtered results. In addition, the growing number of documents and data produced makes it increasingly difficult to find the right and needed information. In companies, these problems can quickly turn into real cost drivers and the existing knowledge potential remains unused (Abu-Rasheed et al., 2022).

To be efficient, employees need a better search methodology than just specifying one or more keywords. An intelligent search approach via MKR is to provide contextual information within the search engine by using word associations to suggest (additional) search terms (Zenkert et al., 2016).

In many available search engines, suggestions for the next term are made based on frequently used search terms from (other or similar) users. Contextual search for associated information takes a different approach. By considering word association strengths within the content, more closely related search terms can be automatically found and suggested for further filtering of search results. After the first search term is entered into the search engine and the first results are returned, all associated terms are identified by utilizing the CIMAWA word association strength represented in MKR. The highest word association strength indicates the closest (semantic) relationship and therefore additional related words are suggested as possible additional terms. In this way, the search engine results are further expanded by directly considering additional associated knowledge.

Technically, all MKR from documents are searched through in the knowledge base as part of the full-text search for all entered terms. The obtained results are in the form of related documents and their semantically related keywords based on CIMAWA word association strength. In order to filter the number of results, the additional acquired tokens (contextual information) through MKR relation can be added to the search expression in order to filter or expand search results.

The full-text associative search is illustrated in Figure 12.10.

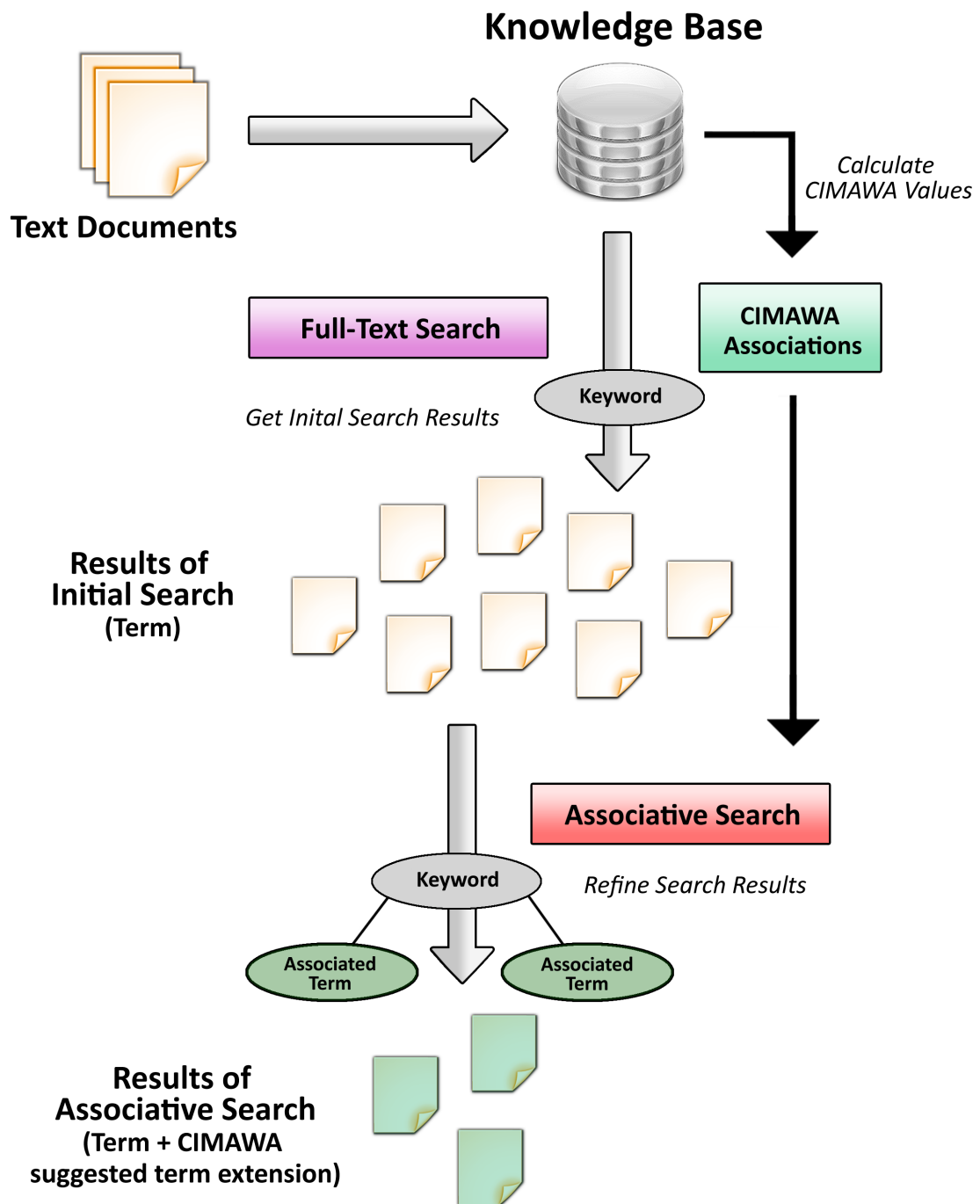


Figure 12.10: Visualization of associative search with associative term extensions based on the CIMAWA (Zenkert et al., 2016)

## 13 Multidimensional Knowledge Visualization

Knowledge visualization is a difficult task if a knowledge base does not provide an interface that enables easy browsing and searching for specific information. In enterprises, the knowledge base is often viewed as a database, distributed database, or cloud storage that supports various applications by providing the required data. However, if an organization's knowledge base is viewed only as a static repository of documents or knowledge assets, knowledge discovery is a very time-consuming task.

In contrast, building a dimensionally structured knowledge base provides opportunities for knowledge discovery. The main advantages and characteristics of multidimensional knowledge bases are scalability, flexibility, dimensionality, and relevance (Zenkert and Fathi, 2016). This means that entities such as people, brands, locations, etc. can be selected from the knowledge base based on dimensional relationships and used for analysis (Zenkert and Fathi, 2016).

For example, different expert information (named entities) can be selected from the knowledge base and all associated facts can be provided. In this scenario, an associative knowledge map or knowledge graph could visualize all facts based on word association strength. Due to the dimensional structuring of the knowledge base, this knowledge map can be further structured by additional dimensional selection. In the example, an associative knowledge map would visualize only the facts about experts related to specific topics (if topics are selected as additional dimensional information).

In the following, the dynamic knowledge map is briefly described. Then, a use case for the transformation of MKR into a knowledge graph is given.

### 13.1 Visualization Transformation and Dynamic Knowledge Maps

Knowledge maps can provide a visual summary of information and the relationships between each piece of information. Based on the graphical layout, the knowledge map can use various nodes, symbols, or graphical representations that relate to the knowledge assets or information. Entities identified by NER are used in the dynamic knowledge map to improve the graphical layout and visual comprehensibility. Depending on the type of named entity (e.g., product, person, place), different (standard) symbols can be used in the knowledge map to distinguish them (Zenkert and Fathi, 2016). Another option is to automatically search for related images from the Internet to visualize the entity information within the nodes in the knowledge map. Klahold et al. (2019) developed a geographically oriented concept for knowledge maps to analyze the inner-city distribution of interests over time.

Proximity calculations based on the CIMAWA measure, explained in Section 4.1.2.2, can be used to arrange the content. Associative information is visualized next to related knowledge objects, while unrelated information is hidden or visualized at a larger distance. Greater spacing implies non-existent or lower word association strength based on CIMAWA.

Other features such as zooming in and out of the map are one of the main goals of a dynamic knowledge map. For this feature, further breakdown of the summarized information is required. For textual resources, this means breaking down documents into their sections, pages, paragraphs, or even sentences. The associations between the contained keywords within the text fragments and the desired level of detail are able to provide the information for the graphical arrangement based on the word association strength. In this way, the dynamic knowledge map is able to restructure

itself and adapt to the selected depth of information (Zenkert and Fathi, 2016). A conceptual overview of the dynamic knowledge map is shown in Figure 13.1.

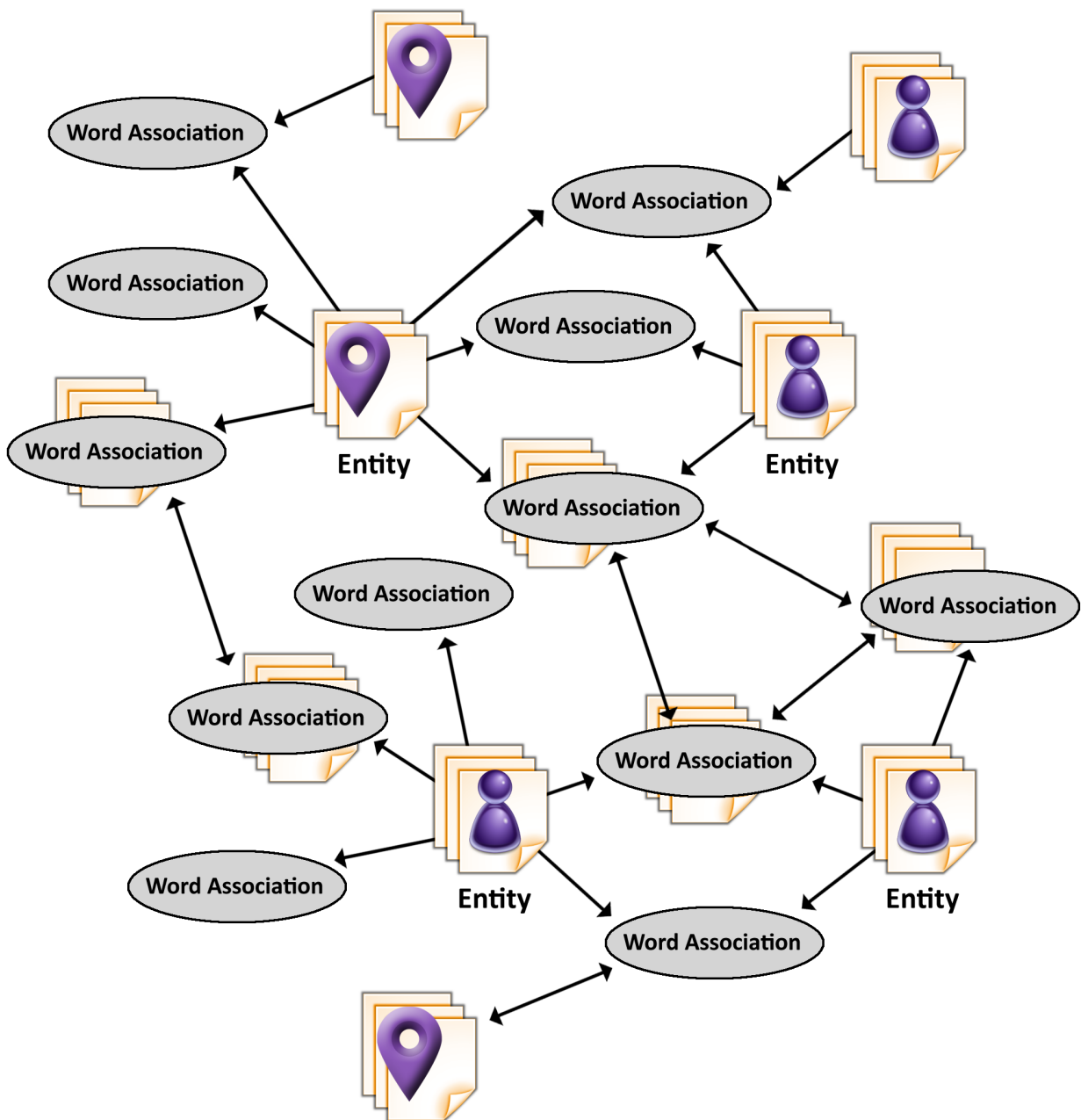


Figure 13.1: Conceptual overview of the dynamic knowledge map. Different entities (e.g., persons, places) are arranged by distances derived from CIMAWA word association strength. (Zenkert et al., 2016)

## 13.2 Transformation of MKR into a Knowledge Graph

MKR contains the results of NER as a representation format and integrates the individual results of the text mining analysis into a combined structure.

For the transformation of MKR into the data basis for a knowledge graph, it is necessary to

## 13.2. Transformation of MKR into a Knowledge Graph

use the corresponding components (entity relationships) of the representation structure to build the nodes and edges of the graph. An example is shown in Figure 13.2. Furthermore, Figure 13.3 shows a second evaluation on different documents from the database. Here, individual documents and related entities are linked to each other as nodes via edges and visualized according to the selected knowledge graph layout. Several layout algorithms have been implemented but will not be further discussed in this dissertation.

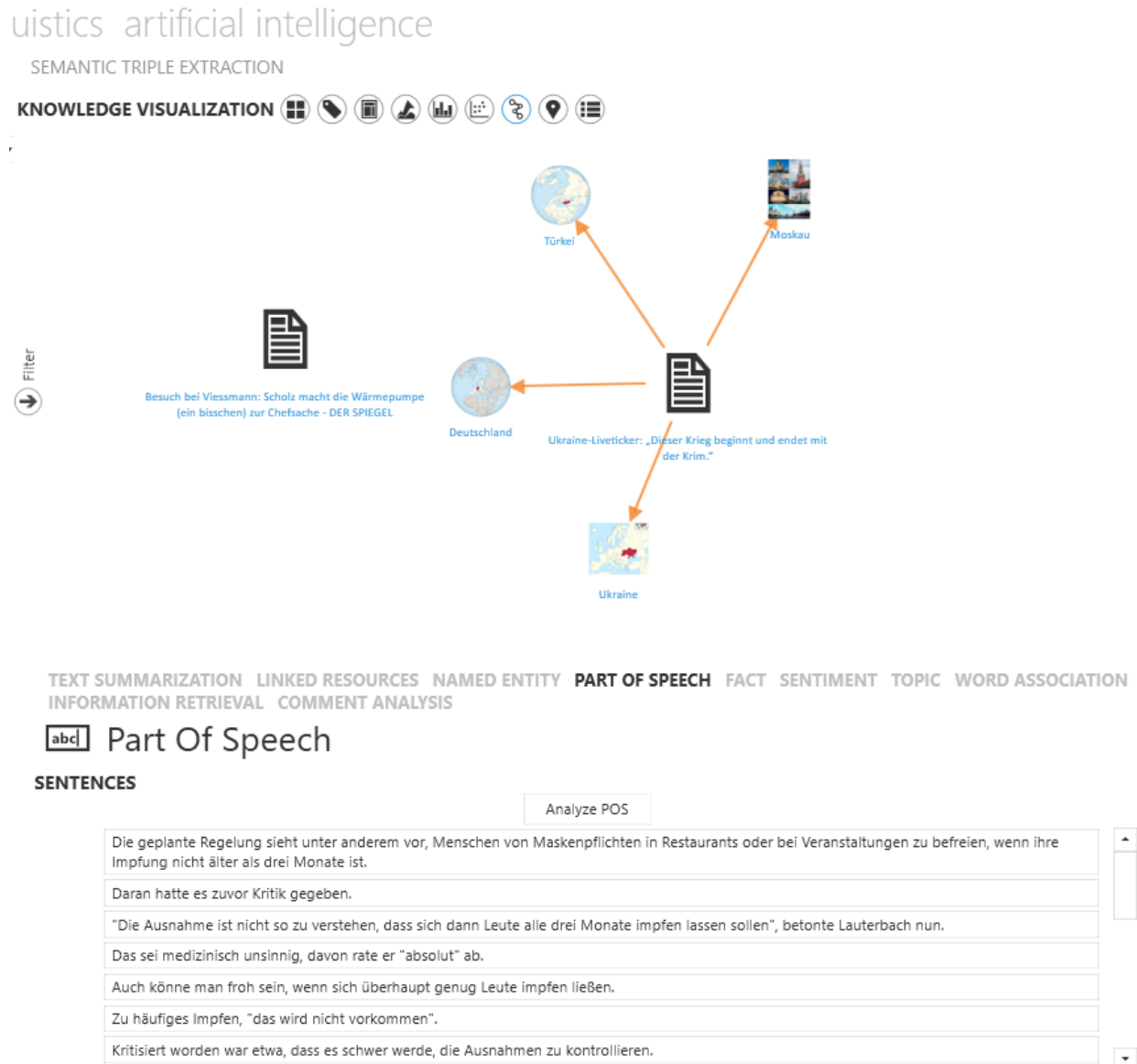


Figure 13.2: KB:mkr Exemplary knowledge graph on documents



## 13.2. Transformation of MKR into a Knowledge Graph

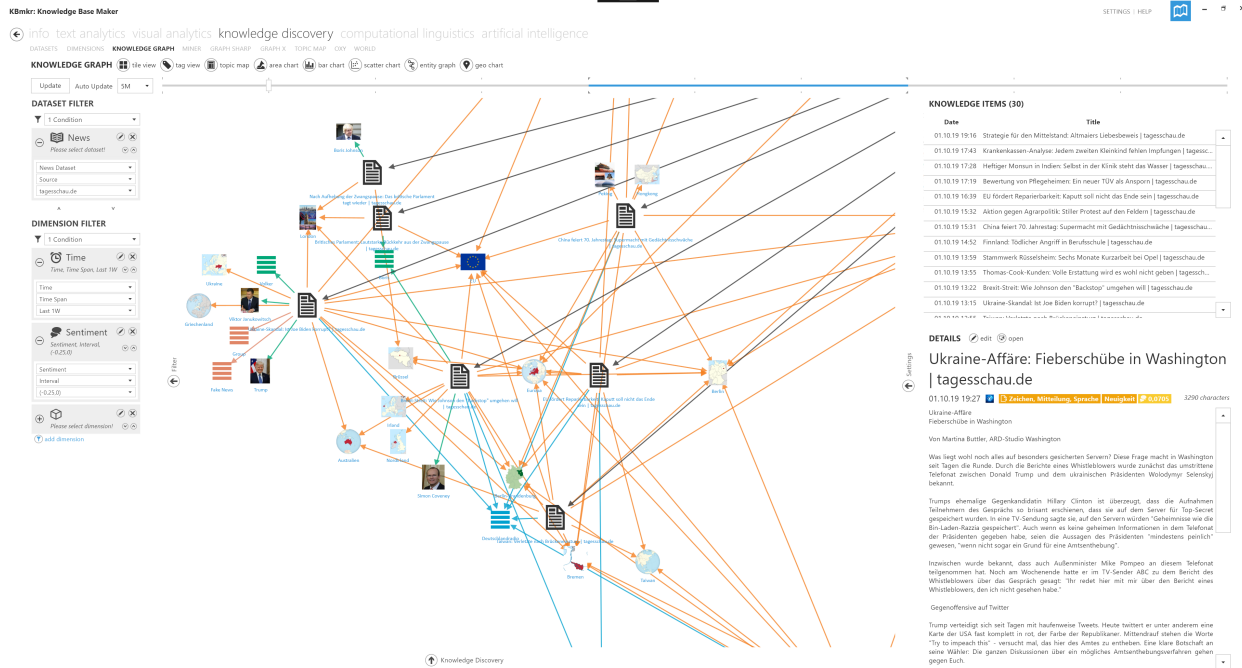


Figure 13.3: KB:mkr Knowledge graph from appearing documents from 1st October 2019

## 14 Use Cases and Applications

### 14.1 Text Summarization via MKR

In text summarization, writers typically select information from multiple relevant sources, extract the most important core information, and then write a topic-specific text or summary. This approach is performed manually by writers, reporters, and journalists. The need for text summaries is very high, because the information available usually exceeds the capacity and the ability of humans to summarize a text or even a collection of texts in a short time and in a meaningful way. (Zenkert et al., 2018)

However, text summarization is a difficult task for computers because of the complexity of dealing with natural language, identifying relationships between textual information, and especially recognizing the importance of individual topics in the big picture. This is the reason why there is currently no general, domain-independent solution for (automatic) text summarization. Approaches and methods for text summarization are typically adapted to the context, require a specific vocabulary, or use an ontology to understand complex contexts.

Integrative text mining is an approach that combines the results of individual NLP and text mining methods into a knowledge base to achieve a multi-perspective analysis of the input text. The insights obtained typically reflect the results of methods such as NER, sentiment analysis, or topic detection applied to the text. In the proposed MKR framework by Zenkert and Fathi (2016), Zenkert et al. (2018), the results are stored in the knowledge base to facilitate knowledge discovery processes, provide contextual (semantic) information in adaptive visualization, and support knowledge discovery or complex computational writing tasks. In the MKR approach, texts are analyzed using the above methods and the results are stored as dimensions. For example, the semantic relations between the extracted entities from a text and their entity-level sentiment given the information about multi-topics from each document are stored in the knowledge base. Accordingly, in a large document collection, related documents can be retrieved based on their entity information on a topic in a corresponding sentiment or even emotion classification.

In this use case published in (Zenkert et al., 2018), dimensional information is used as a selection and filtering tool for extractive text summarization to take advantage of combining the individual analysis results of the aforementioned methods. Furthermore, semantically related information and its meaning is used to aggregate large sets of documents by identifying relevant parts of documents for text summarization using MKR.

#### 14.1.1 Dimensional Text Summarization

Dimensional text summarization is based on the concept of MKR in knowledge bases (Zenkert and Fathi, 2016), (Zenkert et al., 2018). Since all potentially required information is already computed by individual text mining methods and stored in a JSON format (see Figure 11.6), the proposed extractive text summarization algorithm benefits from the operations of dimensional content selection or filtering and can be targeted to content which matches the user query conditions.

In the following, the proposed algorithm published in (Zenkert et al., 2018) is explained with a focus on entity-related text summaries. For example, if a user wants to summarize all documents that mention the German ex-chancellor “Angela Merkel” in the topic “politics”, all documents that contain information about the entity and match the searched topic “politics” usually have to be browsed. Also, documents containing only the term “Merkel” will not be found if the search is

similar to a full-text search. In the dimensionally structured knowledge base, this information is available once the document has been parsed since it stores the results of the NER and topic detection applied to the document in the corresponding *entityRelation* and *topicRelation* in the final MKR output. In this way, the search is backwards and starts with the filtered analysis results to process fewer documents and correct content. Especially for very complex multidimensional search queries, the dimension-oriented structure has a great advantage because the analysis results can be used for text summarization with selection or filtering operations.

The algorithm illustrated in Figure 14.1 is explained as follows (Zenkert et al., 2018):

1. The knowledge base ( $KB$ ) is queried based on different filters for MKR analysis results. The analysis results represent the combinations of analysis dimensions *NER*, *sentiment analysis*, *topic detection*.
  - (a) A *time* filter removes content which doesn't fit into user request (if *time* specified).
  - (b) Filtering or selection of dimensions *entity*, *sentiment* and *topic* (relations in MKR).
2. The analysis results from the subset are retrieved from  $KB$ .
3. While the list of  $MKR_{i...n}$  is not empty, perform following actions:
  - (a) Load the  $MKR_i$  related document  $D_i$ .
  - (b) Perform *NLP operations* (e.g., paragraph splitting, sentence splitting, tokenizing, POS tagging) in order to pre-process  $D_i$ .
  - (c) For each sentence  $S_{j...m}$  from  $D_i$ :
    - i. If  $S_j$  contains the requested named entity  $e$  (*entityRelation*),  $S_j$  is kept, otherwise it is skipped.
    - ii. If the user specified *entity-to-entity* relationship,  $S_j$  is further checked if it contains all entities  $e_{i...n}$ . If yes,  $S_j$  is kept, if no,  $S_j$  is skipped.
    - iii. If the user specified a *sentiment polarity*  $p = \{p \in \mathbb{R} \mid -1 \geq p \leq 1\}$  in *range* or *level*,  $S_j$  is checked if it matches requested  $p$  (*sentimentEntityRelation*). If yes,  $S_j$  is kept, if no,  $S_j$  is skipped.
    - iv.  $S_j$  is appended on the output text  $T$ . A *reference*  $R_i$  to the document  $D_i$  (metadata) is created in a *reference section* at the end of the text (if  $R_i$  is not yet in *reference section*).
    - v. If  $S_m$  is reached, remove  $MKR_i$  from list and select the next analysis result  $MKR_{i+1}$  and its document  $D_{i+1}$ . The algorithm continues at step 3.
4. The algorithm exits if  $n$  is reached in the MKR list and therefore, all related documents  $D_n$  have been analyzed and all related information has been extracted for summarization.

### 14.1.2 Experimental Result

The analysis results of the algorithm are presented in (Zenkert et al., 2018). Results are evaluated with the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric. The usability of the method is particularly seen in the field of chronological text summarization and generation of an automatic excerpt as MKR is already providing all metadata and algorithm input.

### 14.1.3 Limitations

The identification of relevant information is particularly important for the extraction of text content from documents. This process can be modeled algorithmically in a similar way to how human authors would search for information. However, pieces from the content of identified sources must be meaningfully combined to create a coherent text, particularly stylistically, but also chronologically.

The approach outlined has so far left these two problems unaddressed. Nevertheless, the approach is suitable for Wikipedia-like chronological summarization because the algorithm can process documents sequentially based on timestamp information. Thus, according to user input, a summary is created from a chronological sequence of referenced articles, which can even be further adjusted or modified in various criteria (entities, sentiment, topic) via the MKR approach.

This reveals further potential for abstract text summarization, as the generated text can be used as an input or training dataset to rewrite or rephrase selected sentences. (Zenkert et al., 2018)

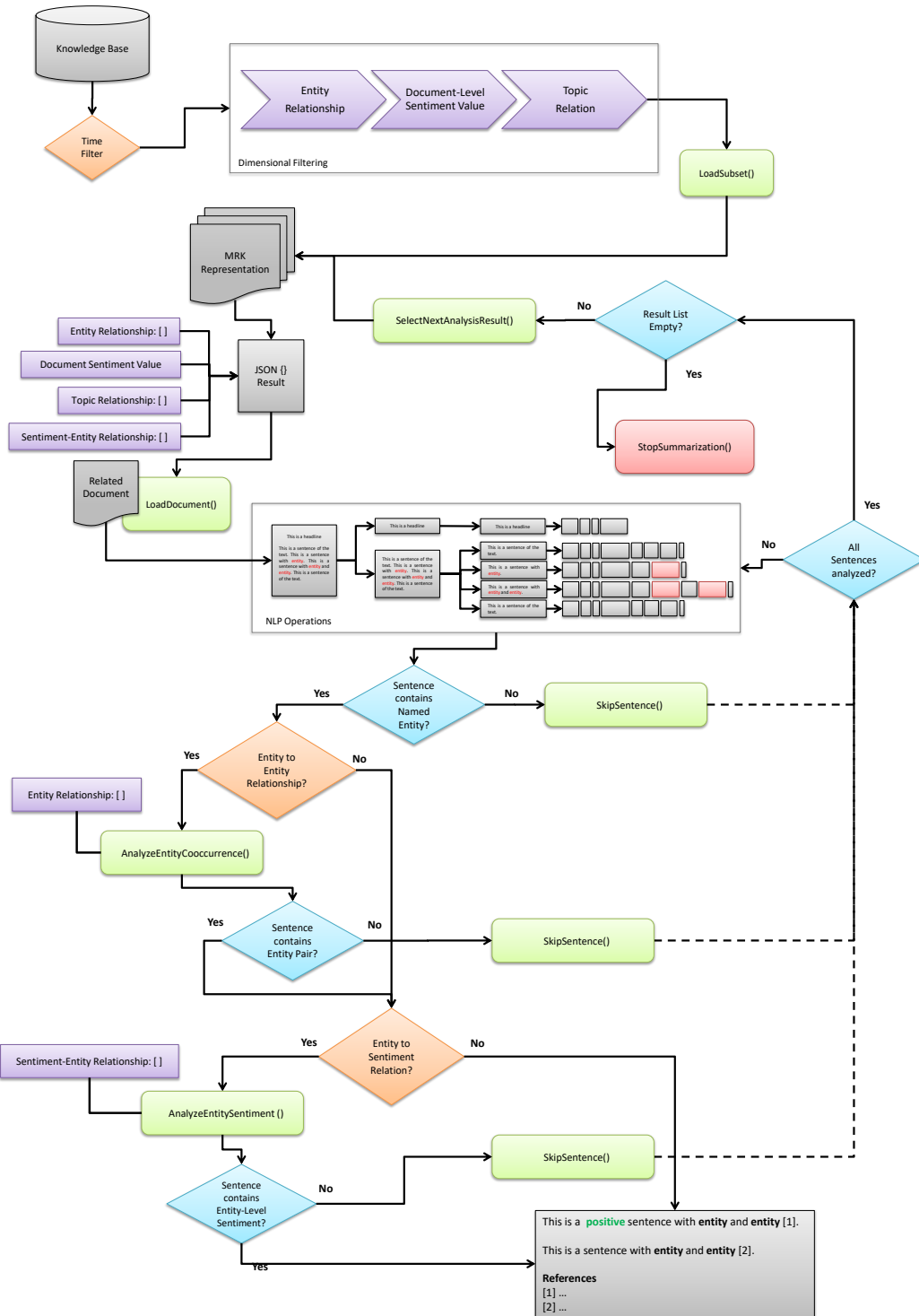


Figure 14.1: Dimensional Text Summarization Algorithm in MKR knowledge base (Zenkert et al., 2018)

## 14.2 Semantic Relationship Analysis via MKR

The process step of semantic relationship analysis in MKR recognizes and extracts semantic relationships, or here mentioned as facts, and stores them in a document-oriented database. Content that is recognized as useful is first divided into sentences and displayed in a list. The sentences are then tokenized. Afterwards the tokens are marked with POS tags, so that the implemented chunker can then use phrases to recognize relevant noun phrases (NP), proper noun phrases (PNP), verb phrases (VP), and other phrases (OP). The phrases are used in the semantic relationship analysis to extract RDF triples. The system searches for an NP (and/or PNP) that follows a VP after a PNP is found. These three components together form a subject-predicate-object triple. The process is repeated for each textual content and a list of triples or semantic relationships are obtained.

Finally, all recognized semantic triples are stored with related metadata in the MKR from corresponding document. Elements of each triple can be edited via the KB:mkr software. It is also possible to manually add or remove triples.

### 14.2.1 Matching with Wikidata Properties

The Wikidata properties are available in the knowledge base of KB:mkr. Each Wikidata property has an identifier, a label, a description, a list of aliases, a data type, and a counter. For matching, the label and the aliases are especially crucial. However, since the aliases are lists, they need to be treated individually, so that there is only one alias in each row. Thus, each property can be compared individually.

In the case for English language, prepared Wikidata properties can be directly compared against the predicate of extracted semantic relationship. In the case of German, the last character of the alias is omitted so that more matches can be found. However, if the aliases are too short, the entire alias is used. The returned values in KB:mkr are the property identifier, the matching alias or label, the identifier, and the value for the match.

### 14.2.2 Building an Ontology Structure and SPARQL Queries

After matching with the Wikidata properties, the returned semantic relationships are available as extracted facts in the MKR of the analyzed document. In the next step, a triple store for RDF triples can be created from the returned semantic relationships. The parent property of the predicate of a fact here is the label of the Wikidata property with which a match was found if the predicate is not identical to the label. If the parent property is not yet in the triple store, then the missing triples are added. Then the label is set as the parent property of the predicate. Only proper names and nouns or entities, classes or strings can appear as objects. To distinguish these cases, DBpedia is asked whether the object is a class or an entity. Finally, additional RDF triples are added to the triple store by calling the reasoning method from the dotNetRDF package (dotNetRDF Project, 2021). Based on the rules specified in a schema file in the form of a graph, new triples are generated from the existing data by applying inference.

The KB:mkr also supports the analysis of free text. The module is located in the lower right area in the tab menu. Figure 14.2 shows the relevant frame from the KB:mkr software.

During processing of entered text, intermediate results are reported via the output. Finally, the number of all records, extracted and filtered facts are provided in the output. In addition, the execution time is displayed. The extracted facts illustrated in Figure 14.3 are added to a table in the KB:mkr software.

For each triple in this table, the time of extraction is displayed. In addition, it is indicated to which list the respective triple was assigned. Below the table is the detailed view shown in Figure 14.4, with which a selected triple can be edited as desired. Changes can be made directly

WIKIDATA PROPERTY MATCHING FREE TEXT ANALYSIS

## Free Text Analysis

**Input**  extract\_facts  apply\_sparql  remote\_query  sparql\_help

Angela[1] Dorothea Merkel (\* 17. Juli 1954 in Hamb  
Merkel wuchs in der DDR auf und war dort als Physi

Fakt gefunden: Angela Dorothea Merkel ist eine deutsche Politikerin  
Kein Fakt.  
Kein Fakt.  
Kein Fakt.  
Fakt gefunden: Merkel wuchs auf in der DDR  
Kein Fakt.  
Kein Fakt.  
Kein Fakt.

-----  
Sätze: 8  
Fakten: 2  
Fakten in Blacklist: 0  
Ausführungszeit: 0:0:4

Figure 14.2: KB:mkr - Free text analysis for extraction of facts

**LIST OF SEMANTIC TRIPLES (2)**

Date	Subject	Predicate	Object	List
16.09.19 22:23	Angela Dorothea Merkel	ist eine deutsche	Politikerin	White
16.09.19 22:23	Merkel	wuchs auf in der	DDR	White

Figure 14.3: KB:mkr - Table with extracted facts

in the table and confirmed with the edit button. Selected triples can also be deleted and the whole table of extracted facts can be cleared. Corresponding MKR entries from selected documents in the knowledge base, or entries created upon free text, are updated accordingly.

**DETAILS**  edit  remove\_triple  clear\_table

## Angela Merkel wuchs auf in der DDR

Language: deDE

Semantic Triple Listing:

Figure 14.4: KB:mkr - Details view of an extracted semantic relationship

**14.2.3 Semantic search with SPARQL**

SPARQL queries are passed as a string to the implemented SparqlSELECT method.

dotNetRDF provides several options. In particular, queries can be made to a local triple store using the SPARQL implementation, and remote ontologies such as DBpedia can be queried using SPARQL endpoints (dotNetRDF Project, 2021). How the query is processed depends on which of the four keywords SELECT, ASK, CONSTRUCT or DESCRIBE is included in the query. SELECT and ASK queries return a table, whereas ASK queries return only one value. CONSTRUCT and DESCRIBE queries return a graph (dotNetRDF Project, 2021).

## 14.3 Knowledge Extraction from Forums

Another application and use case of MKR can be presented for the extraction of forum information in the domain of maintenance and repair of modern vehicles. Within the project application “AUTODIREKT” (SME call for proposal, funded by the Federal Ministry of Education and Research, Germany), MKR was proposed for the modeling and creation of a knowledge base for representation of extracted forum information.

A preliminary study of the project proposal was published by Meckel et al. (2019) and suggests methods for extracting knowledge from unstructured and informal contributions in Internet forums with the goal of synthesizing diagnostic graphs from the existing knowledge base that are part of a maintenance software that supports repair shops in servicing vehicles by suggesting more efficient and targeted diagnostic and maintenance actions in real time (Meckel et al., 2019).

As stated by Meckel et al. (2019), MKR has been proposed in this approach for knowledge extraction and integrative text mining combined with diagnostic graph synthesis and optimization. The targeted solution aims to go beyond OEM service manual instructions and to improve diagnostic procedures for car repair garages. The approach provides accumulated knowledge from a large number of experienced contributors via Internet forums. Extracted information should be made available to garage technicians, enabling a structured, transparent, and thus standardized diagnostic testing procedure. In an example, Meckel et al. (2019) show that it is possible to create a MKR from relevant Internet forums and that the graph synthesis is able to automatically derive optimized diagnostic strategies.

The approach by Meckel et al. (2019) is briefly summarized into three steps:

1. **Webcrawling and Natural Language Processing:** A text mining pipeline is used to extract the forum information via web crawling. The process is shown in Figure 14.5.
2. **Extraction and Calculation of Word Associations with MKR:** Phrases that describe sequences, possibilities, or alternatives are extracted and the entity process descriptions are arranged in a form that resembles an undirected graph, as shown in Figure 14.6
3. **Diagnostic Graph Synthesis and Optimization:** Based on the extracted co-occurrence chains, the graph modeling algorithm detects phrases that indicate variants, alternatives or conditions and synthesizes the diagnostic graph, which is a directed acyclic graph. Root-cases are to be identified through graph synthesis and optimization. The final step in creating the final diagnostic test structure for automotive mechanics is a post-synthesis optimization of the multipath graph.



## 14.4. Knowledge Graph Development with Neo4j based on MKR

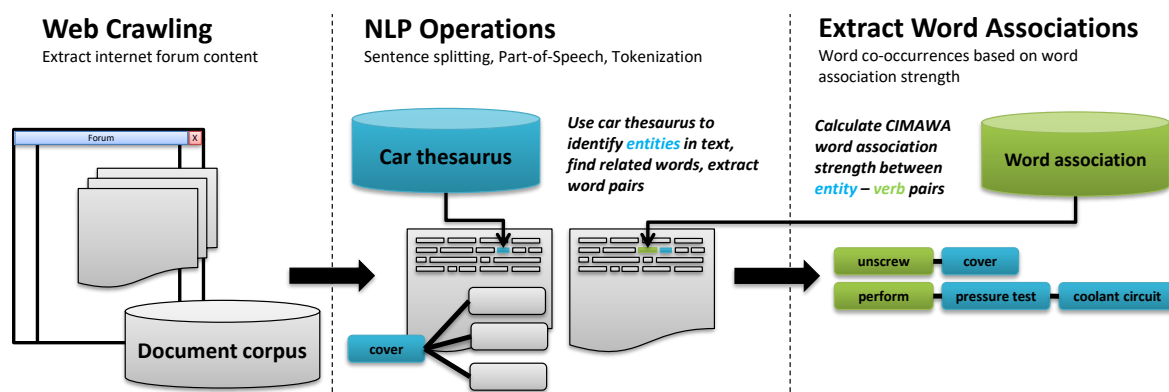


Figure 14.5: Process of web crawling and natural language processing (Meckel et al., 2019)

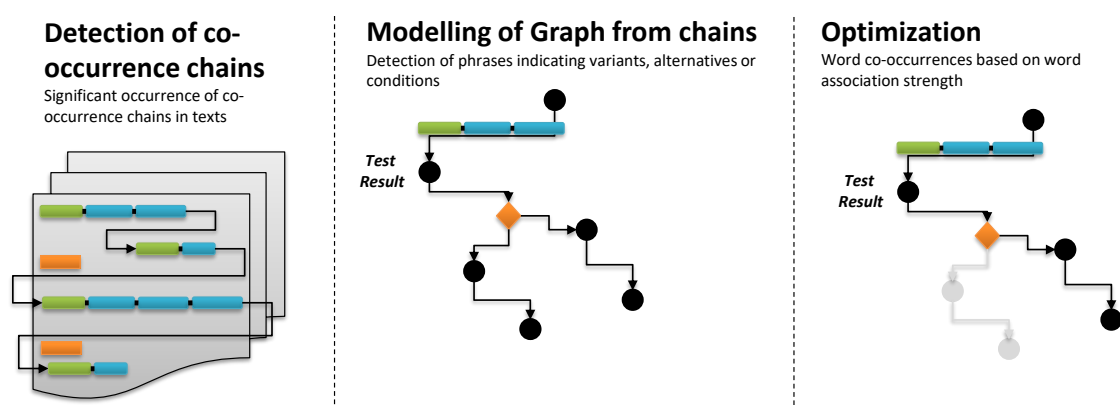


Figure 14.6: Process of the synthesis and subsequent optimization of the diagnostic graph (Meckel et al., 2019)

An example of a synthesized and optimized diagnostic graph for the fault symptom from engine coolant level sensor is further described in (Meckel et al., 2019).

## 14.4 Knowledge Graph Development with Neo4j based on MKR

In the project “Computerunterstützung durch künstliche Intelligenz bei Rettungseinsätzen zur Verbesserung der Erstversorgung” (*EN: Computer support through artificial intelligence in rescue missions to improve first aid treatment*), abbreviated with KIRETT, MKR has been applied as method to pre-process and organize textual data from the medical domain.

The KIRETT project (funded by the Federal Ministry of Education and Research (Germany)) aims to improve first aid during rescue operations using a wearable device. The wearable is used for computer-aided situation recognition by means of artificial intelligence. It provides contextual recommendations for actions and operations to rescue personnel and is intended to minimize damage to patients due to incorrect treatment, as well as increase the probability of survival (Zenkert et al., 2022).

In the first step of the project, the standard operating procedures (textual materials) of basic measures and specialized treatment paths have been provided by the City of Siegen rescue station. The materials were analyzed using the KB:mkr text mining framework. Textual content was extracted from a provided PDF document and transferred to a knowledge base (Zenkert et al., 2022). Named entities and semantic relations recognized by KB:mkr have been further analyzed and are

used in the next step, for the graph modeling and transfer of results into a knowledge graph in Neo4j.

### 14.4.1 Neo4j knowledge graph

The treatment paths and actions of the rescue operations are modeled in a Neo4j graph database. Based on the provided medical literature for rescue operations and guidelines, the modeling, in style of an event-driven process chain (EPC), was first analyzed then transferred into the Neo4j graph database.



Figure 14.7: Exemplary KIRETT Knowledge Graph - Base measurements

**Part V**

**Discussion**

## 15 Summary and Conclusion

This dissertation presents the method of MKR. Here, in the sense of integrative text mining, analysis results of different pipelines are provided for the representation as analysis dimensions for the knowledge base.

The knowledge representation was practically realized with the help of an implementation and tested and evaluated in several use cases.

At the beginning of this thesis, the topics of data mining, text mining and web mining were explained, which together with NLP form the basis of knowledge extraction and automated knowledge acquisition.

Methods based on these, such as NER, topic recognition, sentiment analysis, semantic relational analysis, and text summarization, were systematically introduced and their use as different analysis perspectives in MKR was explained.

Subsequently, the knowledge representation technique MKR, was compared with similar representation formats. Here, especially the knowledge graph as a concept for representation and visualization was emphasized and the possible transformation of MKR into a knowledge graph using RDF was discussed. The topic of visualizing represented results from MKR was also addressed.

A preliminary theoretical consideration of MKR with desired properties was then presented from previously published literature. Here, a knowledge base concept was presented, and properties obtained through MKR were further highlighted.

The pipeline integration via MKR is then further explored and the concept of integrative text mining is introduced. Operations which can be performed by MKR on the knowledge base were then presented. This part was complemented by related literature.

In the implementation part, the KB:mkr Knowledge Base Maker software was introduced and various technical principles were explained. External data required for the implementation, lexical resources, and the creation of text corpora were also discussed.

The implementation of different knowledge extraction processes in KB:mkr and the resulting technical implementation of MKR as a document-oriented representation format were also illustrated.

Finally, the implementation of MKR was used in various use cases and results were summarized.

### 15.1 Summary of Contributions

With respect to the scientific contributions mentioned in Section 1.3, the contributions can be recapitulated and summarized:

1. **Development of a web crawling framework and generation of text corpora:** In Section 10.3.1, a dynamic web crawling framework was presented that automatically retrieves the textual website content from various news portals and provides the textual content for evaluation in the context of the dissertation. A text corpus for the German and English language was created separately to discuss language-specific processing in the MKR framework.
2. **Integration of text mining pipelines into a common representation:** Text mining pipelines were explained through several parts of this dissertation. For this, text mining was distinguished from data mining and web mining, and NLP was introduced. As specialized text mining methods for knowledge acquisition, several extraction methods were explained, and

their individual pipelines were presented. Finally, in Section 11.3, all analysis results are integrated into a common structure, the MKR.

3. **Methods for dimensional filtering of knowledge representation:** Possibilities of selecting individual analysis dimensions and filtering by restricting the database with the help of MKR were discussed in section 6.2. MKR allows backward-oriented search by indexing and can use different analysis results as selection and filtering criteria (and operation).
4. **Implementation of KB:mkr software:** KB:mkr Knowledge Base Maker software was developed for the implementation of MKR in this dissertation. Using a document-oriented NoSQL database, different text corpora were provided and presented through different functionalities of the software such as a) topic development over time b) sentiment evaluation c) entity overview d) full text associative search.
5. **Implementation of knowledge extraction methods and integrative text mining:** Individual analysis methods such as NER, topic detection, sentiment analysis, semantic relational analysis and text summarization are presented in the work in Section 11.2. The merging of the methods in terms of integrative text mining is discussed in Section 11.3.
6. **Exploratory data analysis and transformative visualization:** MKR was used for the dissertation in different use cases and the results were illustrated in the Part IV of the thesis. Here, in addition to exploratory data analysis and visualization of results, use cases have been presented.

## 15.2 Discussion of Results

Through the outline of MKR, it could be shown that a combination of knowledge extraction methods, such as sentiment analysis and topic detection with a corresponding visualization and the processing of semantic web structures is well suited to reduce the information overload, considering the introduction of this work. The use of KB:mkr is simple and self-explanatory. All techniques offer the potential to be further explored. This is especially true in combination with the automation of the techniques.

The knowledge representation method MKR offers extensive application potential for text mining, since MKR is not limited to methods defined in this work. Different applications and as far as results transferable into a textual format are available, can use the methodology of the automated knowledge base. For this, the results must be put into a suitable format that can be represented in MKR. However, due to the JSON format of MKR, there is a high flexibility in the internal structure.

## 15.3 Future Work

As an outlook of the work, the further development of the representation method MKR in the KB:mkr software can be mentioned. It would be possible to use MKR as a tool for providing contextual information in the subject area of computer-aided writing (Klahold et al., 2017), (Klahold and Fathi, 2020). For example, suggestions about semantically related content can be proposed using the relations from MKR. By indexing in the document-oriented NoSQL database, queries can be made at runtime, and, for example, a text editor could be provided with suggestions from the knowledge base. MKR can be further applied in different projects for knowledge representation and knowledge base creation. The compatibility with knowledge graph concepts in RDF open further potential and research direction such as the educational domain (Abu-Rasheed et al., 2018, 2019), (Abu-Rasheed et al., 2022).

## Publications

Various publications have been made as part of this dissertation. Relevant publications are listed by publication type in the following sections:

### Journal Publications

- Abu-Rasheed, H., Weber, C., Zenkert, J., Dornhöfer, M., and Fathi, M. (2022). Transferrable framework based on knowledge graphs for generating explainable results in domain-specific, intelligent information retrieval. In *Informatics*, volume 9, page 6. MDPI.
- Bohlouli, M., Dalter, J., Dornhöfer, M., Zenkert, J., and Fathi, M. (2015). Knowledge discovery from social media using big data-provided sentiment analysis (SoMABiT). *Journal of Information Science*, 41(6):779–798.
- Dornhöfer, M., Sack, S., Zenkert, J., and Fathi, M. (2020). Simulation of smart factory processes applying multi-agent-systems—a knowledge management perspective. *Journal of Manufacturing and Materials Processing*, 4(3):89.
- Fathi, M., Dornhöfer, M., and Zenkert, J. (2020). Mehrdimensionales Wissensmanagement zur nachhaltigen Entscheidungsunterstützung. *Controlling*, 32(1):12–19.
- Nasiri, S., Zenkert, J., and Fathi, M. (2017). Improving CBR adaptation for recommendation of associated references in a knowledge-based learning assistant system. *Neurocomputing*, 250:5–17.
- Zenkert, J., Klahold, A., and Fathi, M. (2018). Knowledge discovery in multidimensional knowledge representation framework. *Iran Journal of Computer Science*, 1(4):199–216.
- Zenkert, J., Weber, C., Dornhöfer, M., Abu-Rasheed, H., and Fathi, M. (2021). Knowledge integration in smart factories. *Encyclopedia*, 1(3):792–811.

### Conference Publications

- Abu-Rasheed, H., Weber, C., Harrison, S., Zenkert, J., and Fathi, M. (2018). What to learn next: Incorporating student, teacher and domain preferences for a comparative educational recommender system. *EduLEARN18 proceedings*, pages 6790–6800.
- Abu Rasheed, H., Weber, C., Zenkert, J., Czerner, P., Krumm, R., and Fathi, M. (2020). A text extraction-based smart knowledge graph composition for integrating lessons learned during the microchip design. In *Proceedings of SAI Intelligent Systems Conference*, pages 594–610. Springer.
- Abu-Rasheed, H., Weber, C., Zenkert, J., Krumm, R., and Fathi, M. (2022). Explainable graph-based search for lessons-learned documents in the semiconductor industry. In *Intelligent Computing*, pages 1097–1106. Springer.

- Abu-Rasheed, H., Zenkert, J., Weber, C., Dornhöfer, M., and Fathi, M. (2019). Language learning tool based on augmented reality and the concept for imitating mental ability of word association (CIMAWA). *EduLEARN19 proceedings*, pages 4118–4125.
- Klahold, A., Fathi, M., and Zenkert, J. (2019). ICDOI-the use of "knowledge discovery from text" to analyze the inner-city distribution of interests over time. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 571–574. IEEE.
- Klahold, A., Zenkert, J., and Fathi, M. (2017). Computer aided writing—a prototype of an intelligent word processing system. In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 234–239. IEEE.
- Meckel, S., Zenkert, J., Weber, C., Obermaisser, R., Fathi, M., and Sadat, R. (2019). Optimized automotive fault-diagnosis based on knowledge extraction from web resources. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1261–1264. IEEE.
- Nasiri, S., Zenkert, J., and Fathi, M. (2015). A medical case-based reasoning approach using image classification and text information for recommendation. In Rojas, I., Joya, G., and Catala, A., editors, *Advances in Computational Intelligence*, pages 43–55, Cham. Springer International Publishing.
- Schulz, W., Zenkert, J., Weber, C., Klahold, A., and Fathi, M. (2018). Sentlyzer: Aspect-oriented sentiment analysis of product reviews. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 270–273. IEEE.
- Uhr, P., Zenkert, J., and Fathi, M. (2014). Sentiment analysis in financial markets - a framework to utilize the human ability of word association for analyzing stock market news reports. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 912–917.
- Zenkert, J. and Fathi, M. (2016). Multidimensional knowledge representation of text analytics results in knowledge bases. In *2016 IEEE International Conference on Electro Information Technology (EIT)*, pages 541–546.
- Zenkert, J., Holland, A., and Fathi, M. (2016). Discovering contextual knowledge with associated information in dimensional structured knowledge bases. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1923–1928.
- Zenkert, J., Klahold, A., and Fathi, M. (2018). Towards extractive text summarization using multidimensional knowledge representation. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0826–0831. IEEE.
- Zenkert, J., Weber, C., Klahold, A., Fathi, M., and Hahn, K. (2018). Knowledge-based production documentation analysis: An integrated text mining architecture. In *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 717–720.
- Zenkert, J., Weber, C., Nadeem, M., Bender, L., Fathi, M., Ahammed, A. S., Ezekiel, A. M., Obermaisser, R., and Bradford, M. (2022). KIRETT - a wearable device to support rescue operations using artificial intelligence to improve first aid. In *8th IEEE International Smart Cities Conference 2022*. IEEE (accepted).

## Bibliography

- Abts, D. and Müller, W. (2009). *Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung*. Springer-Verlag.
- Agarwal, B. and Mittal, N. (2016). Semantic orientation-based approach for sentiment analysis. In *Prominent feature extraction for sentiment analysis*, pages 77–88. Springer.
- Akbik, A. and Broß, J. (2009). Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *www workshop*, volume 48.
- Alghamdi, R. and Alfalqi, K. (2015). A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, 6(1).
- Allahyari, M., Pouriyeh, S. A., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. J. (2017). Text summarization techniques: A brief survey. *CoRR*, abs/1707.02268.
- Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., and Yang, Y. (1998a). Topic detection and tracking pilot study final report.
- Allan, J., Papka, R., and Lavrenko, V. (1998b). On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45.
- Amayri, O. and Bouguila, N. (2013). Online news topic detection and tracking via localized feature selection. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Amit Singhal (2022). Introducing the Knowledge Graph: things, not strings. <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>. [Online; accessed 15.06.22].
- Auer, S., Bryl, V., and Tramp, S. (2014). *Linked Open Data—Creating Knowledge Out of Interlinked Data: Results of the LOD2 Project*, volume 8661. Springer.
- Auer, S., Kovtun, V., Prinz, M., Kasprzik, A., Stocker, M., and Vidal, M. E. (2018). Towards a knowledge graph for science. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pages 1–6.
- Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.
- Bast, H. (2013). Semantische suche. *Informatik-Spektrum*, 36(2):136–143.
- Bast, H., Buchhold, B., Haussmann, E., et al. (2016). Semantic search on text and knowledge bases. *Foundations and Trends® in Information Retrieval*, 10(2-3):119–271.
- Baxendale, P. B. (1958). Machine-made index for technical literature—an experiment. *IBM Journal of Research and Development*, 2(4):354–361.
- Benamara, F., Cesarano, C., Picariello, A., Recupero, D. R., and Subrahmanian, V. S. (2007). Sentiment analysis: Adjectives and adverbs are better than adjectives alone. *ICWSM*, 7:203–206.
- Bennett, C., Ryall, J., Spalteholz, L., and Gooch, A. (2007). The aesthetics of graph visualization. *Computational aesthetics*, 2007:57–64.



- Berners-Lee, T. (2006). Linked data, 2006.
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, 284(5):34–43.
- Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1998). Nymble: a high-performance learning name-finder. *arXiv preprint cmp-lg/9803003*.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Blumauer, A. (2014a). From taxonomies over ontologies to knowledge graphs.
- Blumauer, A. (2014b). Linked Data in Unternehmen. Methodische Grundlagen und Einsatzszenarien. In *Linked enterprise data*, pages 3–20. Springer.
- Blunsom, P. (2004). Hidden markov models. *Lecture notes, August*, 15(18-19):48.
- Bohnacker, U., Dehning, L., Franke, J., and Renz, I. (2002). Textual analysis of customer statements for quality control and help desk support. In *Classification, Clustering, and Data Analysis*, pages 437–445. Springer.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.
- Brants, T. (2000). TnT-a statistical part-of-speech tagger. *arXiv preprint cs/0003055*.
- Brill, E. (1992). A simple rule-based part of speech tagger. Technical report, Pennsylvania Univ Philadelphia Dept Of Computer And Information Science.
- Broekstra, J., Klein, M., Decker, S., Fensel, D., Van Harmelen, F., and Horrocks, I. (2002). Enabling knowledge representation on the web by extending RDF schema. *Computer networks*, 39(5):609–634.
- Broß, J. (2013). *Aspect-oriented sentiment analysis of customer reviews using distant supervision techniques*. PhD thesis, Freie Universität Berlin.
- Cambria, E., Schuller, B., Xia, Y., and Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems*, 28(2):15–21.
- Carstensen, K., Ebert, C., Ebert, C., Jekat, S., Langer, H., and Klabunde, R. (2009). *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Spektrum Lehrbuch. Spektrum Akademischer Verlag.
- Chen, C. P. and Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347.
- Chen, L. and Wei, L. (2010). The hot research topics and the research fronts in the field of web data mining (WDM) based on web of science. In *2010 5th International Conference on Computer Science & Education*, pages 515–518. IEEE.
- Chinchor, N. and Robinson, P. (1997). MUC-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21.

- Chodorow, K. (2013). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. O'Reilly Media, Inc.
- Dale, R. (2021). GPT-3: What's it good for? *Natural Language Engineering*, 27(1):113–118.
- Davis, R., Shrobe, H., and Szolovits, P. (1993). What is a knowledge representation? *AI magazine*, 14(1):17–17.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.
- Dengel, A. (2011). *Semantische Technologien: Grundlagen. Konzepte. Anwendungen*. Springer-Verlag.
- Dornseiff, F. (2004). *Der deutsche Wortschatz nach Sachgruppen*. de Gruyter.
- dotNetRDF Project (2021). dotNetRDF Documentation. <https://www.dotnetrdf.org/docs/>. [Online; accessed 22.08.21].
- Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285.
- Dutta, A., Meilicke, C., and Stuckenschmidt, H. (2015). Enriching structured knowledge with open information. In *Proceedings of the 24th international conference on world wide web*, pages 267–277.
- Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.
- Ehrlinger, L. and Wöß, W. (2016). Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48:1–4.
- Etzioni, O. (1996). The world-wide web: quagmire or gold mine? *Communications of the ACM*, 39(11):65–68.
- Evert, S. and Krenn, B. (2001). Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th annual meeting of the association for computational linguistics*, pages 188–195.
- Fabian, M., Gjergji, K., Gerhard, W., et al. (2007). Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *16th International World Wide Web Conference, WWW*, pages 697–706.
- Färber, M., Bartscherer, F., Menne, C., and Rettinger, A. (2018). Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1):77–129.
- Fasel, D. and Meier, A. (2016). *Big Data: Grundlagen, Systeme und Nutzungspotenziale*. Springer-Verlag.
- Fattah, M. A. and Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech & Language*, 23(1):126–144.
- Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37.

- Fayyad, U. M., Wierse, A., and Grinstein, G. G. (2002). *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann.
- Feldman, R. and Dagan, I. (1995). Knowledge discovery in textual databases (KDT). In *KDD*, volume 95, pages 112–117.
- Feldman, R. and Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.
- Fiscus, J. G. and Doddington, G. R. (2002). Topic detection and tracking evaluation overview. In *Topic detection and tracking*, pages 17–31. Springer.
- Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.
- Gadatsch, A. and Landrock, H. (2017). Zielsetzung von Big-Data-Projekten. In *Big Data für Entscheider*, pages 11–16. Springer.
- Gangemi, A., Presutti, V., Reforgiato Recupero, D., Nuzzolese, A. G., Draicchio, F., and Mongiovi, M. (2017). Semantic web machine reading with FRED. *Semantic Web*, 8(6):873–893.
- Glenn, M., Strassel, S., Kong, J., and Maeda, K. (2006). TDT5 Topics and Annotations. *Linguistic Data Consortium*.
- Graff, D., Cieri, C., Strassel, S., and Martey, N. (1999). The TDT-3 text and speech corpus. In *Proceedings of DARPA Broadcast News Workshop*, pages 57–60.
- Grishman, R. and Sundheim, B. M. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Gupta, V. and Lehal, G. S. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, 1(1):60–76.
- Gupta, V. and Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268.
- Hahn, U. and Schnattinger, K. (1998). Towards text knowledge engineering. *Hypothesis*, 1(2).
- He, Q., Chang, K., Lim, E.-P., and Banerjee, A. (2010). Keep it simple with time: A reexamination of probabilistic topic detection models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1795–1808.
- Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10. Association for Computational Linguistics.
- Hejlsberg, A., Wiltamuth, S., and Golde, P. (2006). *The C# programming language*. Adobe Press.
- Hernández, D., Hogan, A., and Krötzsch, M. (2015). Reifying RDF: What works well with wikidata? *SSWS@ ISWC*, 1457:32–47.
- Heyer, G., Quasthoff, U., and Wittig, T. (2006). Text Mining: Wissensrohstoff Text–Konzepte. *Konzepte, Algorithmen, Ergebnisse*.
- Hildebrand, M., van Ossenbruggen, J., and Hardman, L. (2007). An analysis of search-based user interaction on the semantic web. *Information Systems [INS]*, (E0706).

- Hippner, H. and Rentzmann, R. (2006). Text mining. *Informatik-Spektrum*, 29(4):287–290.
- Hitzler, P., Krötzsch, M., Rudolph, S., and Sure, Y. (2007). *Semantic Web: Grundlagen*. Springer-Verlag.
- Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.
- Hogan, A., Blomqvist, E., Cochez, M., d’Amato, C., Melo, G. d., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., et al. (2021). Knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge*, 12(2):1–257.
- Hotho, A., Nürnberger, A., and Paaß, G. (2005). A brief survey of text mining. In *Ldv Forum*, volume 20, pages 19–62.
- Huang, A. et al. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand, volume 4, pages 9–56.
- Jiang, J. (2012). Information extraction from text. In *Mining text data*, pages 11–41. Springer.
- Jivani, A. G. et al. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938.
- Jo, T. (2019). Text mining. *Studies in Big Data*. Cham: Springer International Publishing.
- John, N. (1982). Megatrends: Ten new directions transforming our lives. New York: Warner.
- Jurafsky, D. and Martin, J. H. (2014). Speech and language processing. vol. 3. US: Prentice Hall.
- Kalavathy, R., Suresh, R., and Akhila, R. (2007). KDD and data mining. In *2007 IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES 2007)*, pages 1105–1110. IET.
- Kaur, A. and Gupta, V. (2013). A survey on sentiment analysis and opinion mining techniques. *Journal of Emerging Technologies in Web Intelligence*, 5(4):367–371.
- Klahold, A. (2009). *Empfehlungssysteme*. Springer.
- Klahold, A. and Fathi, M. (2020). *Computer Aided Writing*. Springer.
- Klahold, A., Uhr, P., Ansari, F., and Fathi, M. (2013). Using word association to detect multitopic structures in text documents. *IEEE Intelligent Systems*, 29(5):40–46.
- Klampanos, I. A., Jose, J. M., and van Rijsbergen, C. K. (2006). Single-pass clustering for peer-to-peer information retrieval: the effect of document ordering. In *Proceedings of the 1st international conference on Scalable information systems*, pages 36–es.
- Kodratoff, Y. (1999). Knowledge discovery in texts: a definition, and applications. In *International Symposium on Methodologies for Intelligent Systems*, pages 16–29. Springer.
- Kosala, R. and Blockeel, H. (2000). Web mining research: A survey. *ACM Sigkdd Explorations Newsletter*, 2(1):1–15.
- Krempel, L. (2005). *Visualisierung komplexer Strukturen: Grundlagen der Darstellung mehrdimensionaler Netzwerke*. Campus Verlag.

- Krempel, L. (2010). Netzwerkvisualisierung. In *Handbuch Netzwerkforschung*, pages 539–567. Springer.
- Kroetsch, M. and Weikum, G. (2016). Special issue on knowledge graphs. *Journal of Web Semantics*, 37(38):53–54.
- Kübler, S. and Maier, W. (2013). Über den Einfluss von Part-of-Speech-Tags auf Parsing-Ergebnisse. *J. Lang. Technol. Comput. Linguistics*, 28(1):17–44.
- Kühnel, A. (2012). *Visual C# 2012: Das umfassende Handbuch*. Galileo Press.
- Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM.
- Kurgan, L. A. and Musilek, P. (2006). A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, 21(1):1–24.
- Landmann, J. and Züll, C. (2004). Computerunterstützte Inhaltsanalyse ohne Diktionär? Ein Praxistest. *ZUMA Nachrichten*, 28(54):117–140.
- Lassila, O. and Swick, R. R. (1999). Resource description framework (RDF) model and syntax specification.
- Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., and Thomas, S. (2002). Relevance models for topic detection and tracking. In *Proceedings of the second international conference on Human Language Technology Research*, pages 115–121. Morgan Kaufmann Publishers Inc.
- Lin, C. X., Ding, B., Han, J., Zhu, F., and Zhao, B. (2008). Text Cube: Computing IR measures for multidimensional text database analysis. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 905–910. IEEE.
- Liu, P. J., Saleh, M. A., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences.
- Lloret, E. (2008). Text summarization: an overview. *Paper supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01)*.
- Losiewicz, P., Oard, D. W., and Kostoff, R. N. (2000). Textual data mining to support science and technology management. *Journal of Intelligent Information Systems*, 15(2):99–119.
- Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.
- MacDonald, M. (2010). *Pro WPF in VB 2010: Windows Presentation Foundation in .NET 4*. Apress.
- Malyshev, S., Krötzsch, M., González, L., Gonsior, J., and Bielefeldt, A. (2018). Getting the most out of wikidata: semantic technology usage in wikipedia’s knowledge graph. In *International Semantic Web Conference*, pages 376–394. Springer.
- Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.
- Manning, C. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

- Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.
- Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I., Siddiqa, A., and Yaqoob, I. (2017). Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access*.
- Martinez-Rodriguez, J. L., Hogan, A., and Lopez-Arevalo, I. (2018). Information extraction meets the semantic web: a survey. *Semantic Web*, (Preprint):1–81.
- Martinez-Rodriguez, J. L., Lopez-Arevalo, I., Rios-Alvarado, A. B., Hernandez, J., and Aldana-Bobadilla, E. (2019). Extraction of RDF statements from text. In *Iberoamerican Knowledge Graphs and Semantic Web Conference*, pages 87–101. Springer.
- McNamara, T. P. (1994). Knowledge representation. In *Thinking and problem solving*, pages 81–117. Elsevier.
- Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.
- Mehler, A. and Wolff, C. (2005). Einleitung: Perspektiven und Positionen des Text Mining. In *LDV-Forum*, volume 20.
- Merkl, D. (1998). Text data mining. *A handbook of natural language processing: techniques and applications for the processing of language as text*.
- Michalik, P., Stofa, J., and Zolotova, I. (2014). Concept definition for big data architecture in the education system. In *Applied Machine Intelligence and Informatics (SAMII), 2014 IEEE 12th International Symposium on*, pages 331–334. IEEE.
- Mikheev, A., Moens, M., and Grover, C. (1999). Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics.
- Muflikhah, L. and Baharudin, B. (2009). Document clustering using concept space and cosine similarity measurement. In *2009 International Conference on Computer Technology and Development*, volume 1, pages 58–62. IEEE.
- Müller, C. and Gurevych, I. (2008). Using wikipedia and wiktionary in domain-specific information retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 219–226. Springer.
- Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Nallapati, R., Zhou, B., Gulcehre, C., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.
- Nandury, S. V. and Begum, B. A. (2016). Strategies to handle big data for traffic management in smart cities. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, pages 356–364. IEEE.
- Nenkova, A. and McKeown, K. (2012). A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.

- Nobata, C., Sekine, S., Murata, M., Uchimoto, K., Utiyama, M., and Isahara, H. (2001). Sentence extraction system assembling multiple evidence. In *NTCIR*.
- Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2013). Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.
- Palmer, D. D. and Day, D. S. (1997). A statistical profile of the named entity task. In *Proceedings of the fifth conference on Applied natural language processing*, pages 190–193. Association for Computational Linguistics.
- Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 1-2(2):1–135.
- Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.
- Pfeffer, J. (2010). Visualisierung sozialer netzwerke. In *Netzwerkanalyse und Netzwerktheorie*, pages 227–238. Springer.
- Pfeiffer, H. D. and Pfeiffer, J. J. (2007). Representation levels within knowledge representation. In *International conference on conceptual structures*, pages 484–487. Springer.
- Prud'hommeaux, E., Seaborne, A., et al. (2017). Sparql query language for rdf. w3c recommendation (2008).
- Pujara, J., Miao, H., Getoor, L., and Cohen, W. (2013). Knowledge graph identification. In *International semantic web conference*, pages 542–557. Springer.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Ramshaw, L. A. and Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.
- Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Conference on empirical methods in natural language processing*.
- Remus, R., Quasthoff, U., and Heyer, G. (2010). SentiWS-A Publicly Available German-language Resource for Sentiment Analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.
- Ringler, D. and Paulheim, H. (2017). One knowledge graph to rule them all? Analyzing the differences between DBpedia, YAGO, Wikidata & co. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 366–372. Springer.
- Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of information science*, 33(2):163–180.
- Samuel, J. (2017). Collaborative approach to developing a multilingual ontology: a case study of wikidata. In *Research Conference on Metadata and Semantics Research*, pages 167–172. Springer.
- Sarawagi, S. (2008). *Information extraction*. Now Publishers Inc.
- Sayyadi, H. and Raschid, L. (2013). A graph analytical approach for topic detection. *ACM Transactions on Internet Technology (TOIT)*, 13(2):1–23.

- Schiller, A., Teufel, S., and Thielen, C. (1995). Guidelines für das tagging deutscher textcorpora mit stts. *Universitäten Stuttgart und Tübingen*.
- Schreiber, A. T. and Raimond, Y. (2014). RDF 1.1 Primer.
- Sharafi, A. (2013). Knowledge discovery in databases. In *Knowledge Discovery in Databases*, pages 51–108. Springer.
- Simitsis, A., Baid, A., Sismanis, Y., and Reinwald, B. (2008). Multidimensional content exploration. *Proceedings of the VLDB Endowment*, 1(1):660–671.
- Singh, B. and Singh, H. K. (2010). Web data mining research: a survey. In *2010 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–10. IEEE.
- Sirmakessis, S. (2012). *Text mining and its applications: results of the NEMIS Launch Conference*, volume 138. Springer.
- Sleator, D. D. and Temperley, D. (1993). Parsing english with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, pages 277–292.
- Sorensen, E. and Mikailesc, M. (2010). Model-View-ViewModel (MVVM) design pattern using Windows Presentation Foundation (WPF) technology. *MegaByte Journal*, 9(4):1–19.
- Stahl, F., Gabrys, B., Gaber, M. M., and Berendsen, M. (2013). An overview of interactive visual data mining techniques for knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(4):239–256.
- Stegbauer, C. and Häußling, R. (2010). Einleitung: Visualisierung von Netzwerken. In *Handbuch Netzwerkforschung*, pages 525–525. Springer.
- Stock, W. G. (2007). Themenentdeckung und -verfolgung und ihr Einsatz bei Informationsdiensten für Nachrichten. *Information–Wissenschaft und Praxis*, 58(1):41–46.
- Tan, A.-H. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, volume 8, pages 65–70. sn.
- Tanwar, P., Prasad, T., and Aswal, M. S. (2010). Comparative study of three declarative knowledge representation techniques. *International Journal on Computer Science and Engineering*, 2(07):2274–2281.
- Tanwar, P., Prasad, T., and Datta, K. (2012). Hybrid technique for knowledge representation & a comparative study. *International Journal of Computer Science and Engineering Survey*, 3(4):43.
- Taylor, A., Marcus, M., and Santorini, B. (2003). The penn treebank: an overview. *Treebanks*, pages 5–22.
- Thorsten Liebig (2018). Neo4j: A Reasonable RDF Graph Database & Reasoning Engine [Community Post]. <https://neo4j.com/blog/neo4j-rdf-graph-database-reasoning-engine/>. [Online; accessed 15.06.22].
- Uhr, P., Klahold, A., and Fathi, M. (2013). Imitation of the human ability of word association. *International Journal of Soft Computing and Software Engineering (JSCSE)*, 3(3):248–254.
- Varghese, R. and Jayasree, M. (2013). A survey on sentiment analysis and opinion mining. *International Journal of Research in Engineering and Technology*, 2(11):312–317.



- Vijayarani, S., Ilamathi, M. J., Nithya, M., et al. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16.
- Völkel, M., Kröttsch, M., Vrandečić, D., Haller, H., and Studer, R. (2006). Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pages 585–594.
- Vrandečić, D. and Kröttsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.
- Wayne, C. L. (1997). Topic detection and tracking (TDT). In *Workshop held at the University of Maryland on*, volume 27, page 28.
- Weiss, S. M., Indurkha, N., Zhang, T., and Damerau, F. (2010). *Text mining: predictive methods for analyzing unstructured information*. Springer Science & Business Media.
- Whitney, P., Engel, D., and Cramer, N. (2009). Mining for surprise events within text streams. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 617–627. SIAM.
- Witte, R., Mülle, J., Bestehorn, M., Gitzinger, T., Heitmann, B., Kappler, T., Krestel, R., Lang, T., Leitner, J., Siegmund, C., and Wild, F. (2006). *Text Mining: Wissensgewinnung aus natürlich-sprachigen Dokumenten*.
- Zechner, K. (1997). A literature survey on information extraction and text summarization. *Computational Linguistics Program*, 22.
- Zesch, T., Müller, C., and Gurevych, I. (2008). Extracting lexical semantic knowledge from wikipedia and wiktioary. In *LREC*, volume 8, pages 1646–1652.
- Zhang, D. (2013). *Integrative text mining and management in multidimensional text databases*. PhD thesis, University of Illinois at Urbana-Champaign.
- Zhang, Q. and Segall, R. S. (2008). Web mining: a survey of current research, techniques, and software. *International Journal of Information Technology & Decision Making*, 7(04):683–720.
- Zhou, G. and Su, J. (2002). Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 473–480.