Quality Assessment in Text Analysis Pipelines

Von der Graduate School of Excellence advanced Manufacturing Engineering
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Cornelia Kiefer

aus Villingen-Schwenningen

Hauptberichter:	Prof. Dr.-Ing. habil. Bernhard Mitschang
Mitberichter:	Prof. Dr.-Ing. Stefan Deßloch
Tag der mündlichen Prüfung:	29.07.2020

Institut für Parallele und Verteilte Systeme (IPVS)
der Universität Stuttgart
2020

Acknowledgements

This thesis is dedicated to several people who have supported my work on it in various ways. First of all, I would especially like to thank my doctoral advisor, Prof. Dr.-Ing. habil. Bernhard Mitschang, who gave me the opportunity to work on this challenging topic in his research group. His constructive support, valuable advice, and many interesting discussions have significantly helped me to conduct my scientific research and to grow both as a researcher and as a person.

Furthermore, my sincere thanks go to the co-reviewer, Prof. Dr.-Ing. Stefan Deßloch, for spending time reading this thesis and for giving helpful suggestions and comments. I would like to acknowledge Prof. Dr.-Ing. Thomas Bauernhansl for filling in the function of the chairman of the examination committee and his interesting advice with respect to possible future work.

Moreover, I would like to thank Dr. rer. nat. Peter Reimann, my junior research group leader at the Graduate School of Excellence, for his continuous and patient effort to improve the quality of my research. He is very motivating and a solid source for good advise.

I have been working on this thesis while being a PhD student at the Graduate School of excellence Advanced Manufacturing Engineering (GSaME) at the University of Stuttgart. I would like to thank the German Research Foundation (DFG), who gave financial support to the project.

I moreover was integrated into the department Applications of Parallel and Distributed Systems at the University of Stuttgart (IPVS-AS). It was a pleasure and a great time working together and having stimulating discussions with many colleagues at the department. Special thanks go to Dr.-Ing. Laura Kassner for her extensive and reliable collaboration, as well as her precise and helpful feedback. Furthermore, I would like to thank several persons who have published papers with me. Besides Prof. Dr.-Ing. habil. Bernhard Mitschang and Dr. rer. nat. Peter Reimann, this mainly includes Dr.-Ing. Laura Kassner, Dr.-Ing. Christoph Gröger, Dr.-Ing. Eva Hoos, Dr.-Ing. Jan Königsberger, Dr.-Ing. Stefan Silcher and Christian Weber. Likewise, many other current and former colleagues of the department Applications of Parallel and Distributed Systems helped me via miscellaneous advises and discussions, in particular Dr. rer. nat. habil. Holger Schwarz, Dr. rer. nat. Pascal Hirmer, Alejandro Gabriel Villanueva Zacarias, Manuel Fritz, Corinna Giebler, Michael Behringer, Vitali Hirsch, Marco Spiess and Julian Ziegler.

Moreover, I like to thank Dr.-Ing. Dominik Herr, Dr.-Ing. Lena Wagner and Abdullah Demir for their support in applying the concepts developed in this thesis to real use cases.

Finally, particular thanks go to Dr. rer. nat. Peter Reimann, Dr. rer. nat. habil. Holger Schwarz, Dr. rer. nat. Pascal Hirmer, Sarah Oppold and Christian Weber for spending their precious time proof-reading this document and for their valuable comments.

I would also like to acknowledge the work of several students of the University of Stuttgart. In particular, my thanks go to the following students for developing prototypes of the concepts and methods proposed by this thesis: Shreyas Bettadapura Raghavendra, Shalini Chellathurai Saroja, Paul Dieterich, Robert Golda, Raoul Graumann, Andreas Laukart, Narottam Panday, Tobias Renz and Marco Link.

In addition, I thank the administrations of the Institute of Parallel and Distributed Systems and of the Graduate School of Excellence advanced Manufacturing Engineering for their support in organizational and technical issues. Special thanks go to Ralf Aumüller, Eva Strähle, Corinna Noltenius, Dr. rer. nat. Matthias Saric, Carsten Graser, Dr.-Ing. Thomas Ackermann and Dr. Gabriele Erhardt who always helped me in case of questions or problems. Moreover, I like to especially thank Prof. Dr.-Ing. Sylvia Rohr who encouraged and supported me in holding my PhD project on track, also while I was partially at parental leave.

Furthermore I like to especially thank Prof. Dr. Ulrike Pado from the Hochschule für Technik (HFT) in Stuttgart, who encouraged me in starting a PhD project at all. Also, I want to thank Dr. Christian Scheible, who introduced me to Prof. Dr.-Ing. habil. Bernhard Mitschang.

Finally, I want to express my sincere gratitude to my husband Jascha, our two wonderful children Kim and Zoe, to my sister Stephanie and to my parents Waltraud and Christian. They always supported and encouraged me during my work on this thesis. I would also like to thank the rest of my family and all my friends who were by my side during all the years. Each of them helped me a lot, even if they are not aware of it.

Cornelia Kiefer

Magstadt, August 03, 2020

Contents

List of Abbreviations

df	Document frequency
fn	False negatives
fp	False positives
GSaME	Graduate School of Excellence advanced Manufacturing Engineering
LI	Language identifier
LSA	Latent semantic analysis
NHTSA	National Highway Traffic Security Administration
NLTK	Natural Language Processing Tool Kit
NLP	Natural language processing
OCR	Optical character recognition
POS	Part of speech
POS-tagger	Part-of-speech tagger
QUALM	**QUAL**ity **M**ining
REST	Representational State Transfer
SITAM	Stuttgart IT Architecture for Manufacturing
SVD	Singular value decomposition
SVM	Support vector machines
TDM	Term-document matrix
tf	Term frequency
tfidf	Term frequency-inverse document frequency
tp	True positives

Abstract

High quality data and data analysis results are a precondition for future concepts such as the data-driven factory of the future. The quality of business decisions is directly influenced by the quality of data and analysis results. Current data quality concepts and tools only consider the raw input data of data analysis pipelines. They fail to regard specifics of analysis tools as well as data for each step of analysis pipelines. To fill this research gap, the *QUALM concept for continuous and holistic data quality measurement and improvement within data analysis pipelines* is presented in this thesis. In QUALM, data characteristics as well as specifics of analysis tools such as training data, features and semantic resources are regarded in each step of analysis pipelines.

Existing data quality metrics measure the data quality of structured data, e.g., by counting null values, duplicates or invalid values. Equivalent approaches for textual data are missing. Additionally, most domain-specific text data sets are unlabeled. Thus, in addition to missing data quality metrics, also evaluation metrics are not calculable for these data sets and the thereupon derived analysis results. This leads to a *high uncertainty of the analysts with respect to the quality of data and analysis results*. QUALM conquers this challenge with a set of concrete text data quality methods. QUALM data quality indicators quantify text characteristics and give hints with respect to the expected quality of analysis results. Just as existing metrics for structured data determine, e.g., the number of null values and invalid fields, the QUALM indicators characterize texts with respect to, e.g., the number of abbreviations, spelling mistakes and ungrammatical sentences. Moreover, as demanded by the QUALM concept, these methods do not only consider the raw data, but also respect the specifics of the analysis tools. For example, QUALM has indicators which measure the confidence of standard analysis tools or the fit of semantic resources employed by analysis tools. Each indicator comes with a corresponding modifier. For example, the amount of abbreviations or spelling mistakes may be measured by a QUALM indicator. A corresponding QUALM modifier, e.g., modifies the data by means of resolving abbreviations or by a correction of spelling mistakes.

Moreover, the *selection of appropriate training data* is especially difficult for analysts such as domain experts with little IT and/or data science knowledge. Yet, the appropriate selection of training data has a high impact on the quality of analysis results. Therefore, QUALM addresses this issue through a concrete

method. The corresponding QUALM indicator measures data quality by means of the similarity between input and training data. In the case of textual data, text similarity metrics such as Latent Semantic Analysis and Cosine Similarity are employed. The counterpart QUALM modifier automatically selects the best-fitting training data and thus impedes low-quality results of domain-specific analysis.

Finally, QUALM has another method which addresses data quality issues that arise from information extraction approaches that only consider either structured or unstructured text data in isolation. These isolated approaches may lead to a loss in terms of the amount of new information that may be presented to the analyst. In this thesis, this issue is addressed by a hybrid approach, which *exploits structured and unstructured information sources in information extraction*. To this end, especially structured data is considered which is enriched by unstructured free text fields. In the suggested approach, structured data is used to guide and improve the text analysis process. To this end, structured data is employed as a basis for a first grouping of free text fields and for removing information from the free texts which is already present in the structured fields. Thus, the hybrid approach yields more new and relevant information.

The QUALM concept and methods are evaluated with respect to several industry-near application scenarios and corresponding concrete data sets. For example, the analysis of downtimes on a production line is considered. To this end a confidential industry data set comprising structured data enriched with free-text fields is employed. In further application scenarios, sample citizen data scientists are considered, i.e., domain experts with little IT and data science knowledge, who want to build analysis pipelines from scratch. E.g., they want to know customer opinions on a product. The evaluation results are very promising. The QUALM indicators and analysis result quality, measured as accuracy, correlate. Thus QUALM indicators are valid means to indicate the expected analysis result quality to the analyst. Moreover, the investigated QUALM modifiers lead to an increase in accuracy, e.g., of part-of-speech tagger and language identifier tools. In a qualitative discussion in this thesis, the positive effect of QUALM on a whole chain of analysis tools, i.e., an analysis pipeline, is shown.

Deutsche Zusammenfassung

Voraussetzung für Zukunftskonzepte wie die datengetriebene Fabrik der Zukunft sind qualitativ hochwertige Daten und Datenanalyseergebnisse. Die Güte von Geschäftsentscheidungen wird direkt durch die Qualität der Daten und Analyseergebnisse beeinflusst. Aktuelle Datenqualitätskonzepte und -werkzeuge betrachten lediglich die rohen Eingabedaten zu Datenanalysepipelines. Sie versäumen es, die Daten und die Spezifika von Analysetools für jeden Schritt in Analysepipelines zu betrachten. Um diese Forschungslücke zu adressieren, wird in dieser Arbeit das *QUALM-Konzept für kontinuierliche und holistische Datenqualitätsmessung und -verbesserung innerhalb von Datenanalysepipelines* vorgeschlagen. In QUALM werden die Charakteristika sowohl der Daten als auch der Ressourcen berücksichtigt.

Existierende Datenqualitätsmetriken messen die Datenqualität von strukturierten Daten z.B. indem Nullwerte, Duplikate oder ungültige Werte gezählt werden. Equivalente Ansätze für Textdaten fehlen. Hinzu kommt, dass die meisten domänenspezifischen Textdatensätze ungelabelt sind. Somit sind zusätzlich zu fehlenden Datenqualitätsmetriken außerdem auch Evaluationsmetriken für diese Datensätze und die abgeleiteten Analyseergebnisse nicht berechenbar. Dies führt zu einer *großen Unsicherheit der Analysten in Bezug auf die Qualität der Daten und Analyseergebnisse*. An diese Herausforderung wird mit Hilfe einer Liste konkreter Textdatenqualitätsmethoden in QUALM herangegangen. QUALM-Datenqualitätsindikatoren quantifizieren Textcharakteristika und geben Hinweise auf die zu erwartende Qualität von Analyseergebnissen. So wie die existierenden Metriken für strukturierte Daten zum Beispiel die Anzahl an Nullwerten und ungültigen Werten bestimmen, charakterisieren die QUALM-Indikatoren Texte in Bezug auf beispielsweise die Anzahl an Abkürzungen, Rechtschreibfehlern und ungrammatischen Sätzen. Wie vom QUALM-Konzept gefordert betrachten diese Methoden außerdem nicht nur die Rohdaten, sondern auch die Spezifika der Analysetools. Deshalb gibt es zusätzlich weitere Indikatoren, die beispielsweise die Konfidenz von Standardanalysetools messen oder prüfen wie gut die semantischen Ressourcen passen, die von den Analysetools genutzt werden. Zu jedem Indikator gibt es einen passenden Modifikator. Zum Beispiel kann die Menge an Abkürzungen oder Rechtschreibfehlern durch einen QUALM-Indikator gemessen werden. Ein entsprechender QUALM-Modifikator verändert die Daten beispielsweise indem Abkürzungen aufgelöst oder Rechtschreibfehler verbessert werden.

Weiterhin ist die *Auswahl geeigneter Trainingsdaten* besonders schwierig für Analysten wie etwa Domänenexperten mit wenig Wissen in den Bereichen IT und/oder 'Data Science'. Die Auswahl geeigneter Trainingsdaten hat jedoch einen großen Einfluss auf die Qualität von Analyseergebnissen. Deshalb wird diese Auswahl von einer der konkreten QUALM-Methoden adressiert. Der entsprechende QUALM-Indikator misst Datenqualität mittels der Ähnlichkeit zwischen Eingabe- und Trainingsdaten. Für Textdaten können Textähnlichkeitsmetriken wie beispielsweise die Kosinusähnlichkeit und die Latente Semantische Analyse genutzt werden. Der entsprechende Modifikator wählt automatisch die am besten passenden Trainingsdaten aus und verhindert so qualitativ schlechte Ergebnisse von domänenspezifischen Datenanalysen.

Zum Schluss adressiert eine weitere QUALM-Methode Datenqualitätsprobleme, die bei Informationsextraktionsansätzen entstehen, die entweder nur strukturierte oder nur unstrukturierte Daten isoliert betrachten. Diese isolierten Ansätze können zu einem Verlust in Bezug auf die Menge an neuen Informationen führen, die dem Analysten präsentiert werden können. In der vorliegenden Arbeit wird diese Problematik mit einem hybriden Ansatz adressiert, der sowohl *strukturierte als auch unstrukturierte Informationsquellen* bei der Informationsextraktion nutzt. Hierzu werden insbesondere mit Freitextfeldern angereicherte strukturierte Daten betrachtet. Im vorgeschlagenen Ansatz wird der Textanalyseprozess mittels der strukturierten Daten angeleitet und verbessert. Dabei werden strukturierte Daten als Basis für eine erste Gruppierung der Freitexte genutzt. Weiterhin werden auf Basis der strukturierten Datenfelder bereits in den strukturierten Daten enthaltene Informationen aus den Freitextfeldern gelöscht. Der hybride Ansatz führt zu mehr neuen und relevanten Informationen.

Das QUALM-Konzept und die QUALM-Methoden werden im Hinblick auf industrienahe Anwendungsszenarien und die entsprechenden konkreten Datensätze evaluiert. Zum Beispiel wird die Analyse von Stillständen auf einer Produktionslinie betrachtet. Hierzu wird ein vertraulicher Datensatz aus der Industrie genutzt, der mit Freitextfeldern angereichterte strukturierte Daten enthält. In weiteren Anwendungsszenarien werden 'Citizen Data Scientist' in den Fokus gerückt, das heißt Domänenexperten mit wenig Wissen in den Bereichen IT und 'Data Science'. Diese wollen Analysepipelines zügig und ohne große Hürden ganz von vorne aufbauen und zum Beispiel die Kundenmeinungen zu einem Produkt analysieren. Die Evaluationsergebnisse sind sehr vielversprechend. Die QUALM-Indikatoren und die Analyseergebnisqualität, gemessen als Accuracy,

korrelieren. QUALM-Indikatoren sind somit ein valides Mittel um dem Analysten die zu erwartende Analyseergebnisqualität anzuzeigen. Weiterhin führen die untersuchten QUALM-Modifikatoren zu einer Erhöhung der Accuracy, zum Beispiel von Werkzeugen zur Wortartenannotation und Sprachidentifikation. In einer qualitativen Diskussion wird abschließend der positive Effekt von QUALM auf eine Verkettung von Analysetools, also auf eine Analysepipeline, gezeigt.

Chapter 1

Introduction

The value which decision-makers in industry ascribe to unstructured data and text analysis results increases [HJ15, KM16]. For example, in the context of Industry 4.0 and the data-driven factory, data drives crucial decisions and processes [KGK+17]. Therefore, various data types and concrete data sets are being analyzed by means of advanced data analytics methods such as machine learning. For instance, structured data such as tables in databases may be analyzed by means of data mining. Unstructured data such as free text entries in a spreadsheet may be analyzed by means of text mining. Big and/or stream data such as sensor data, time series databases and social media data may be analyzed with scalable and distributed computing infrastructures. Many IT infrastructures and toolkits for mining structured data, text and big and/or stream data exist.

In general, profound data analytics is said to base on IT, data analytics as well as domain expertise [GSD07, WFH11]. Especially in industry, domain expertise is needed to interpret the data and to build reasonable data analysis pipelines. For example, the domain experts may provide context information, such as on normal and critical value ranges, units or contextual factors such as working habits and knowledge on peculiarities, e.g., of machines. New courses of studies try to bring all of these three competences, i.e., IT, data analytics as well as domain expertise, together [Unib, Unia]. Another approach is to simplify data analytics. Then, also IT-inept domain experts can build data analysis pipelines. Nowadays, many simplified toolkits for data processing and data analytics exist, e. g., RapidMiner[1], SPSS[2] and FlexMash [HB16] (also see Section 2.2.3).

[1] `https://rapidminer.com/`
[2] `http://www.ibm.com/analytics/us/en/technology/spss/`

Moreover, the quality of the input data to analysis crucially influences the quality of analysis results and the quality of oftentimes thereupon derived business decisions [GKH+16, MV18]. Low quality data can lead to low-quality analysis results and wrong or missing decisions. Huge losses for organizations are due to bad quality data [DK10]. Thus, also toolkits to ensure high data quality exist (cf. Section 2.3.5). These toolkits mainly work on structured data. They provide methods to measure and to improve data quality. For example, they determine the percentage of duplicates and null values in a database table and merge duplicates or filter out null values. Yet, the most important information sources in organizations, such as the workers, managers and customers produce unstructured data such as texts. Moreover, especially people make mistakes in data entry which leads to low data quality of such unstructured data. About 50% of the data in organizations are estimated to be unstructured [Rus07]. Thus, high quality of unstructured, e. g., text data, needs to be ensured as well. Yet, while plenty research approaches in the field of data quality address structured data, only little research addresses the quality of unstructured data, such as text data. Therefore, data quality dimensions and metrics for unstructured, especially text data, are needed [BS16, SIG+12, Son04].

In summary, the quality of data analysis results in industry highly depends on the (1) data analytics method, (2) computing infrastructure, (3) domain expertise and (4) data quality. The concepts proposed in this thesis do not aim at improving the quality of analysis results by enhancing the data analytics methods and computing infrastructures. This thesis rather focuses on data quality. Yet, in difference to existing frameworks and toolkits for data quality, the concept suggested does not merely focus on structured data and considers it in isolation. Rather, the quality of unstructured textual data in the context of data and text analysis is considered. The concepts suggested in this thesis moreover focus on data analysis performed by IT-inept domain experts without data analytics skills. Since data quality problems may add up within the analysis pipeline, not only the quality of the raw input data to analysis but also the quality of data at all intermediate steps from raw input data over preprocessing steps till analysis results need to be ensured.

Many existing definitions of data quality emphasize the subjective character of data quality. For example, Woodall and Wainman [WW15] discuss that data quality needs to be "fit for purpose". Wang and Strong define data quality as *"fitness for use by data consumers"* [WSF95]. In their definition, the data

consumers are described as the human end consumers of data. In a data analysis pipeline many algorithmic data consumers, i. e., analysis tools, need to be considered with respect to data quality (see Section 2.3.2). Thus, the approach suggested in this thesis, considers not only the data, but also the analysis tools, i.e., the data consumers, in quality assessment.

The core contribution of this thesis is a concept and prototypical implementation of a new data quality approach for high QUALity Mining (QUALM) (cf. Section 4). QUALM specifically enables high quality analysis pipelines built by domain experts without IT/data analytics competences. In QUALM, besides the data, also the specifics of analysis tools are considered in quality assessment. These specifics comprise the training data employed within supervised machine learning tools, semantic resources, such as dictionaries and ontologies, and information on features, i. e., data characteristics, that are highly weighted by a certain analysis tool. The QUALM concept and prototypical implementation enable a step-wise measurement and improvement of the fit of each analysis tool, i.e., data consumer, and the respective input data within the whole analysis process. By this, the fitness for use by data consumers, i. e., the data quality, of the data is assessed and improved for each individual step in the analysis pipeline. Within the QUALM framework, several concrete QUALM data quality methods for data quality assessment (QUALM indicators) and data quality improvement (QUALM modifiers) are exploited (for an overview, see Table 5.1 in Section 5.2). Since especially methods for quality assessment of textual data are missing in existing research works (cf. Section 2.3.4), the methods suggested focus on text data quality. They come with prototypical implementations and are evaluated with respect to several text data sets with different characteristics, such as news articles, prose, chat posts, tweets and free texts collected by workers on a production line. The evaluation results show that the QUALM indicators correlate with the accuracy of analysis tools used frequently in the preprocessing phase of text analysis pipelines, namely part-of-speech tagger (POS-tagger) and language identifiers. Thus, with the suggested quality indicators the quality of analysis results may be estimated, before the analysis tools are actually carried out. Moreover, as expected, the corresponding QUALM modifiers improve the quality of analysis results in terms of accuracy.

In Section 1.1, the major motivation of this thesis is illustrated via a running example. Section 1.2 comprises a discussion of more detailed challenges that have to be considered by concepts and methods for data quality in the context

of domain-specific, simplified data and text analysis. Furthermore, the state of current work is summarized and how this current work meets the discussed challenges. Concrete contributions of this thesis and of the proposed concepts and methods are illustrated in Section 1.3. Section 1.4 depicts the outline of this thesis.

1.1 Running Example and Major Motivation

As a motivating use case, a team leader who is responsible for a production line is considered (also see Sections 3.1 and 3.3.1). He wants to get information on frequent reasons of downtimes on the production line. To this end, he has access to a data set with a free text field on causes and corrective actions related to machine downtimes. He may use a simplified data analysis toolkit such as RapidMiner[3] or FlexMash[4]. He starts to build an analysis pipeline from scratch. In doing so, he employs many simplified "out-of-the-box" analysis tools.

In Figure 1.1, a sample analysis pipeline built by the domain expert for his use case, is illustrated. The industry data with free-text information on causes and actions related to *machine downtimes* is the operational input data to the analysis pipeline. The texts are analyzed by consecutive analysis modules. The domain expert decides to label the language of each text in a first step by the *Language Identifier*. He filters the data and only uses free texts with the language label "German" in the next steps. Then a part of speech, such as verb, noun or adjective is assigned to each word by the *part-of-speech tagger (POS-tagger)*. Finally, named entities, such as persons, errors and machines are recognized automatically by the *Named Entity Recognizer (NER)*. The analysis pipeline built by the domain expert, prepares the data and enriches it with labels. The resulting labeled data contains information, e.g., on persons and machines and is the basis for a subsequent interpretation by the domain expert.

For each step in the pipeline, various "out-of-the-box" implementations and tools exist. For example, Tika[5], Language-detector[6] and Language Identifier[7] may be employed in the automatic identification of the language. Various "out-of-the-

[3] `https://rapidminer.com/`
[4] `https://github.com/hirmerpl/FlexMash`
[5] `https://tika.apache.org/`
[6] `https://github.com/optimaize/language-detector`
[7] Available from the DKPro Core library: `https://dkpro.github.io/dkpro-core/`

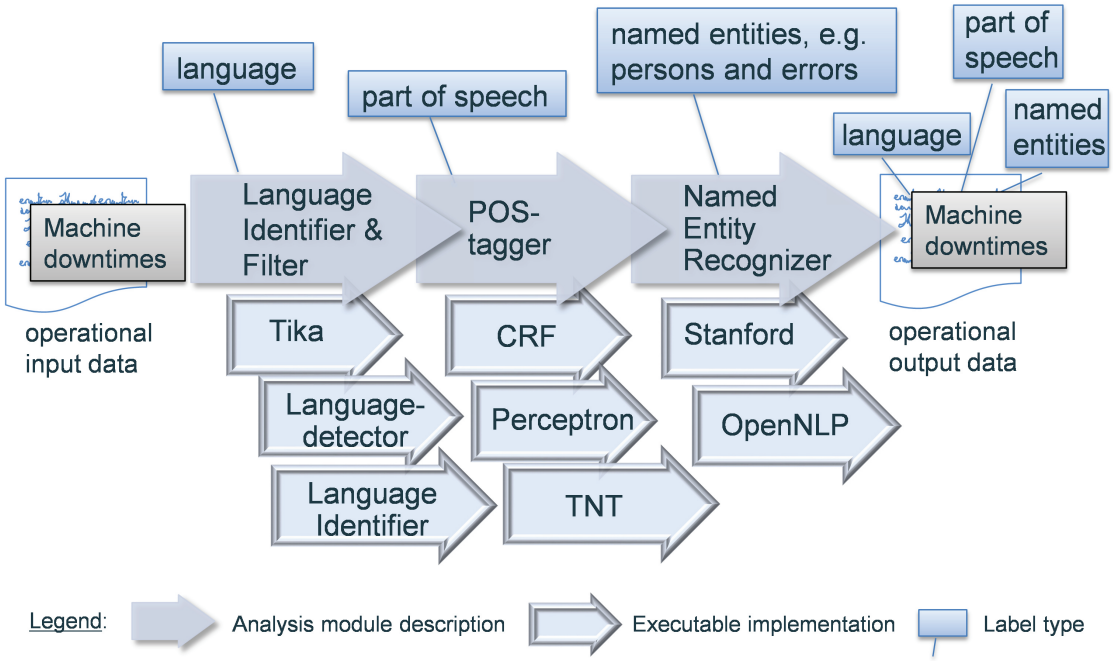


Figure 1.1: Sample analysis pipeline for textual industry data with concrete examples for "out-of-the-box" analysis tools such as Tika.

box" implementations for the POS-tagger module exist, e. g., CRF[8], Perceptron[9] and TNT[10]. Tools for NER exist in the Stanford tool collection[11] as well as in OpenNLP[12]. To the domain expert moreover several pre-trained models for NER are available. The domain expert applies default analysis tools "out-of-the-box" with default training data.

The concrete analysis tools in the analysis pipeline are typically trained on default training data, e.g., clean news texts. In the example illustrated in Figure 1.1, these tools are now applied to data with very different characteristics, such as industry data with many spelling mistakes and abbreviations. This leads to low-quality analysis results, since the quality of labels added by analysis tools, such as parts of speech or named entities, highly depends on the training data.

[8] http://www.nltk.org/_modules/nltk/tag/crf.html
[9] http://www.nltk.org/_modules/nltk/tag/perceptron.html
[10] https://www.nltk.org/_modules/nltk/tag/tnt.html
[11] https://nlp.stanford.edu/software/CRF-NER.shtml
[12] https://opennlp.apache.org/

Without the QUALM concept suggested in this thesis, the domain expert, e. g., employs default and oftentimes non-fitting analysis tools to messy text data. The specifics of the analysis tools and the characteristics of the data sets, e. g., many abbreviations and spelling mistakes in free texts collected by workers on a production line, do not fit together. For example, non-fitting news texts are employed as default training data by Tika (Language Identifier) and CRF (POS-tagger). This may lead to low-quality analysis results such as wrong language labels and missing named entity identifications, e.g., missing labels with respect to persons and errors. Furthermore, the analyst may not even know about the quality issue. The quality of analysis results in domain-specific analysis projects can usually not be assessed by means of existing evaluation metrics, since the necessary gold labels are missing. For example, no gold labels are available for the evaluation of analysis steps such as language identification and POS-tagging. To alleviate this situation, this thesis presents methods which can estimate the analysis result quality which might be expected.

1.2 Challenges and State of Current Work

The identification of detailed challenges has been backed up by a thorough literature review on data mining, text mining, natural language processing and data quality (cf. Chapter 2). Moreover, it is based on the investigation of several application scenarios and corresponding data analysis pipelines (cf. Chapter 3). A further basis has been an analysis of different data analysis and data quality toolkits (cf. Sections 2.2.3 and 2.3.5). The following subsections discuss the resulting challenges, as well as the corresponding state of current work.

1.2.1 Continuous and Holistic Data Quality Measurement and Improvement within Data Analysis Pipelines

In a data analysis pipeline, the output of a foregoing analysis tool is input to the subsequent analysis tool and so forth. Thus, data quality problems may add up within the analysis pipeline. For example, in the analysis pipeline illustrated in Section 1.1, errors in POS-tagging lead to subsequent errors in named entity recognition (NER), since the NER analysis tool relies on the part of speech

labels. For this reason, not only the quality of the raw input data to analysis, but also the quality of data at each intermediate step from raw input data over preprocessing steps till analysis results need to be considered. Moreover, data quality should not be assessed in isolation. Rather, data and data consumers such as concrete analysis tools need to be considered in quality assessment. Finally, tailored data quality improvement methods are needed.

Existing data quality frameworks do not consider data at each intermediate step in quality assessment (e.g., [SC13, WS96]). Moreover, analysis tools and their specifics such as training data, semantic resources and features are not considered by existing quality assessment approaches, although they have a crucial impact on the quality of analysis results (cf. [AHG11, PY10, HBB13, FP18, PCF+12]). Thus, also no tailored data quality improvement which considers data as well as analysis tools in each step of analysis pipelines is possible on the basis of existing data quality frameworks.

1.2.2 Uncertainty with Respect to the Quality of Analysis Results for Unlabeled Text Data

Evaluation methods to assess the quality of analysis results exist, but they depend on manually added labels. For example, the accuracy of an analysis tool may be calculated by comparing the predictions of the system with manually added labels, i.e., "gold labels". Yet, domain-specific text analysis usually is performed on unlabeled data, i. e., labels such as on topics and sentiments contained in the data are not manually added to the data. For example, in the analysis described in Section 1.1, no gold labels with respect to languages, parts of speech and named entities exist.

In some cases, labels for later analysis steps such as sentiment analysis are added to the domain-specific data. But then, still no labels for earlier steps in text mining are available. Usually, at least for preprocessing steps on textual data, no "gold labels" are available. Thus, evaluation metrics such as accuracy are not calculable. Moreover, only little research addresses the assessment of text data quality. While data quality indicators for structured data exist which for example measure the percentage of null and out-of-domain values, equivalent methods with respect to text data are missing [BS16, SIG+12, Son04]. Thus, an analyst who builds an analysis pipeline for such unlabeled data, cannot know data and/or analysis result quality. Moreover, the existing evaluation methods

do not give hints for improvement but rather only state, e. g., the percentage of correctly made decisions of a classifier.

1.2.3 Selection of Appropriate Training Data

In a data analysis pipeline the input data is processed by a sequence of various analysis tools. As described in the previous section, domain-specific input data sets usually come without gold labels, so that evaluation metrics such as accuracy are not calculable. Since domain-specific data is usually unlabeled, for many domain-specific analysis pipelines furthermore no perfectly fitting training data sets are available. Yet, analysis tools which base on supervised machine learning techniques and which are part of many analysis pipelines, rely on training data. Training data is compiled by manually labeling a part of the domain-specific data with gold labels. This is time-consuming, expensive and demands expert knowledge (e. g., see Ide et al. [IP17]). Therefore, it is not realistic to have a "perfect" training data set for each analysis tool and domain-specific input data set. Oftentimes, default training data sets are used instead. Besides these default training data sets, usually further training data sets are available, which, while not fitting perfectly to the domain-specific input data, still are more similar to the input data than default training data. Such better-fitting training data also leads to higher quality analysis results. Yet, most of the times, domain experts who construct analysis pipelines neglect the task of selecting appropriate training data. They rely on default training data sets, e. g., since they do not know which other training data sets exist and what they are used for. Since the default training data sets may be very different from the domain-specific input data that is to be analyzed, default training data may lead to low-quality analysis results.

Existing research in the machine learning community heavily discusses how to improve an already selected training data set. For example, instance selection, domain-adaptation and the selection of subsets of a given training data set are investigated (cf. [WNC05, OLAMTK10, AHG11, BM98, PY10]). None of these research works addresses the initial selection task. The approaches rather build up on already selected training data, e. g., in terms of improving performance or coverage.

1.2.4 Exploitation of Structured and Unstructured Information Sources in Information Extraction

Many data sets encompass structured data fields with embedded free text fields, e.g., a tabular data structure where one or more of the columns store textual content. The text fields allow to input information which cannot be encoded in structured fields. For example, in the use case described above, the free text fields contain information on causes and actions related to machine downtimes. Several information extraction approaches use either structured or unstructured data in isolated analyses, e.g. on textual data only (cf. [GACOR05, FKS06, GML14]). The result of isolated mining of structured data fields misses crucial information encoded in free text. The result of isolated text mining often mainly repeats information already available from structured data (cf. [GML14]). The actual information gain of isolated text mining is thus limited. For example, on the basis of an isolated analysis of structured data in the data set employed in the running example, many coarse-grained and well-known reasons for downtimes on the production line are identified. Based on the free texts only, information which is already known from the structured data, such as concrete error codes, is identified.

Approaches exist which consider both, structured as well as unstructured data, in information extraction. These approaches, e.g., first bring structure to the unstructured textual data by means of relation extraction approaches, and then combine the resulting structured data with other structured data sets, before actually applying the information extraction approaches (cf. [Zha15, GSB14]). Yet, within the application of the information extraction process, all existing approaches only consider either structured or unstructured text data in isolation.

1.3 Contributions of this Thesis

As illustrated in the following subsections, the contributions offered by this thesis address the challenges discussed in Section 1.2. The following subsections illustrate the respective concrete contributions proposed by this thesis. Most parts of this thesis correspond to revised and composite versions of previous author publications [Kie16, Kie17, Kie19, KRM19a, KRM19b, KRM20] All concepts in

these publications were developed exclusively by the author of this thesis. These publications are also respectively cited at relevant occasions in this section.

1.3.1 QUALity Mining (QUALM) Approach

As discussed in Section 1.2.1, related work considers only the raw input data to analysis. Since existing approaches do not assess quality for each individual intermediary step of analysis pipelines, and moreover, specifics of data analysis tools are not considered, no continuous and holistic data quality assessment is possible.

In this thesis, the QUALity Mining (QUALM) approach to continuous and holistic data quality measurement and improvement within data analysis pipelines is suggested (cf. [KRM19b] and Chapter 4). The contribution comprises a conceptual description of the QUALM approach and a prototypical implementation of QUALM. Moreover, concrete methods which may be employed in QUALM and prototypical implementations of these methods are presented. The concrete methods focus on text data quality, since only little approaches with respect to text data quality yet exist [BS16, SIG+12, Son04]. The QUALM methods comprise data quality "indicators" which assess data quality. Moreover, also corresponding methods for quality improvement exist, these are called "modifiers" in QUALM. The methods are applied to each individual step in an analysis pipeline. Firstly, all applicable indicators are selected and calculated. On the basis of indicators, for which a value beyond a pre-defined threshold is calculated, corresponding counterpart modifiers are suggested. For example, the data may be modified by, e. g., correcting spelling mistakes or by dissolving abbreviations. Moreover, the specifics of the analysis tool may be modified by, e. g., the employment of well-fitting semantic resources, such as a relevant dictionary of abbreviations or technical terms. By this, QUALM assesses and improves data quality for each step in the analysis pipeline. The concrete methods not only consider the respective data but also the specifics of data analysis tools in each step. Thus, especially also specifics of data analysis tools such as training data and semantic resources are taken into account when measuring and improving data quality. To this end, information with respect to the specifics of analysis tools are stored and provided in QUALM by means of repositories for training data, semantic resources and analysis tools. In the evaluation, several text data sets are employed and QUALM is evaluated in the context of text analysis pipelines.

Especially in the case of messy text data with many abbreviations and spelling mistakes, QUALM improves the accuracy of analysis results, e.g., of language identification, part-of-speech tagging, clustering and sentiment analysis tools. Besides the evaluation of single QUALM methods, moreover the positive effect of QUALM on a whole chain of analysis tools is discussed.

1.3.2 Quality Indicators for Text Data

In Section 1.2.2, an omnipresent challenge of domain-specific text analysis was pointed out: the textual data sets usually come without any labels. Since no labels are available for each step in the analysis pipeline, the analyst has no information on the quality of analysis results. Moreover, especially domain-specific data sets and analysis results are oftentimes of low quality (cf. Kiefer [Kie17]). Thus, many domain-specific data analysis results may be of low quality without the analyst even knowing about the problem.

We conquer this challenge by means of a set of concrete quality indicators for text data (cf. Kiefer [Kie19] and Section 5). These quality indicators assess text data quality for each step in the analysis pipeline. They reflect characteristics of the text data sets such as the proportion of data items which have a certain characteristic, e.g., the percentage of words with spelling mistakes in a text document. In difference to existing data quality metrics for structured data, the suggested indicators focus on unstructured text data. Moreover, the indicators do not depend on manual labels. We discuss design decisions in the implementation of QUALM indicators and evaluate them. In the evaluation, free texts from production, news, prose, tweets and chat data are investigated. The evaluation results show that the suggested indicators and the evaluation metric accuracy correlate. Thus, the indicators may be employed in predicting the expected analysis result quality.

Furthermore, the QUALM data quality indicators give hints towards starting points for data quality improvement. To this end, each QUALM indicator has a counterpart modifier (cf. Kiefer et al. [Kie16, KRM19b]). For example, an indicator to measure the percentage of abbreviations in text document exists. The corresponding modifier automatically dissolves abbreviations in texts and serves as method for data quality improvement.

1.3.3 Methods for the Automatic Measurement of the Fit of Training Data (FiT) and for the Automatic Selection of Training Data (SeT)

We explained in Section 1.2.3 that citizen data scientists neglect the task of selecting appropriate training data. A proper selection is crucial and has high impact on the analysis result quality.

In the third contribution of this thesis, the impact of various training data on the accuracy of analysis tools is considered (cf. [KRM20] and Chapter 6). If data quality is perceived as "fitness for use by data consumers", the supervised machine learning tool, and the training data employed by the tool, is considered as "data consumer" (cf. Section 2.3.2). Thus, operational input data need to be fit for use with respect to the analysis tools consuming that data, and thus, also with respect to the training data employed by those tools. We measure this fitness for use by means of similarity metrics. The similarity metric can be employed as a data quality indicator and furthermore is the basis for an automatic selection of the best-fitting training data. To this end, this thesis presents two new conceptual methods for the automatic measurement of the fit of training data (FiT) and for the automatic selection of training data (SeT). Moreover, a prototypical implementation and an extensive evaluation with respect to text data is presented. In the evaluation, three text similarity metrics are considered and a text analysis module which is present in almost any text analysis pipeline is focused: the POS-tagger. We employ three concrete implementations of POS-taggers and moreover test the methods suggested based on 18 different text data sets reaching from humorous texts over governmental texts and reviews to news and chat data. The evaluation results show a correlation between textual similarity and the result quality of the part-of-speech taggers and moreover show the benefits of an automatic selection of training data when compared to the employment of default training data.

1.3.4 A Hybrid Approach to the Exploitation of Structured Data Within the Information Extraction Process on Text

The last contribution addresses the challenge presented in Section 1.2.4. Information extraction is oftentimes performed on either structured or unstructured

data types in isolation. Yet, only an intelligent exploitation of both data sources guarantees high quality information.

In the last contribution, a hybrid approach to the exploitation of structured data within the information extraction process on text is suggested (cf. [KRM19a] and Chapter 7). It is based on two additional processing steps. Firstly, a new (1) *grouping step* is presented, which groups free text fields by means of groups which are already present in the structured data. Secondly, in a (2) *removal step*, information which is also available from structured data is removed from the free text fields. Thus, previously unseen information re-emerges in the analysis results. We present the concept for hybrid information extraction as well as a prototypical implementation and a thorough evaluation with respect to two use cases and concrete data sets from production and aftersales. The evaluation results show the benefits of the suggested approach when compared to isolated approaches which either only consider structured data or which focus exclusively on text data.

In summary, the proposed solution exploits results of analyzing structured data within a text mining process i. e., structured information guides and improves the information extraction process on textual data.

1.4 Outline of this Thesis

The next *Chapter 2* provides background information about terms, concepts, and technologies that are relevant for this thesis. This mainly encompasses information about three separate topics: (1) Industry 4.0 and the vision of a data-driven factory, (2) data mining and text mining, as well as (3) data quality. With respect to the latter topic, especially the definition of data quality used within this thesis is stated.

Chapter 3 circumscribes the major application scenarios considered in this thesis. Moreover, based on one of these application scenarios, an initial assessment of quality issues within domain-specific analysis pipelines built by citizen data scientists is presented.

The subsequent chapters describe the contributions of this thesis. These chapters illustrate the respective approaches and their design considerations. In addition, each chapter covers comprehensive discussions of related work.

Chapter 4 describes the QUALM concept for high quality data mining.

Chapter 5 details the concrete data quality methods within QUALM, i.e., QUALM indicators as well as QUALM modifiers. Furthermore, it presents evaluation results with respect to QUALM methods and with respect to their combined effect on a whole chain of analysis tools. Especially, the results presented show that QUALM data quality indicators may help in reducing the uncertainty with respect to the quality of data and analysis results for unlabeled text data.

The remaining two chapters detail two more QUALM methods and their prototypical implementations and evaluations.

Chapter 6 deals with the challenge of selecting appropriate training data within analysis pipelines and presents two new methods for the automatic measurement of the fit of training data and for an automatic selection of the best-fitting training data.

Chapter 7 addresses quality issues which arise due to information extraction on either structured data or text data in isolation. To maximize the amount of new information gained from structured data enriched by free text fields, a hybrid information extraction approach which exploits structured data within a text analysis process is suggested.

Finally, *Chapter 8* concludes this thesis with a summary of its major contributions. Furthermore, it lists promising opportunities for future research.

Chapter 2

Background

This chapter provides information about terms, concepts, and technologies that are important to understand the content of this thesis. Since the research is conducted in the context of data analytics in industry, a brief introduction to Industry 4.0 is given in Section 2.1 and a discussion on how the research work presented in this thesis fits into this future vision for industry is added. Then, the data-driven factory of the future is described, a concrete application scenario for Industry 4.0. Here, a conceptual application scenario and an IT architecture were developed at the University of Stuttgart by Kassner, Gröger, Königsberger, Hoos, Kiefer et al. [KGK+17]. Herein, we address requirements for a data quality layer, which are described in Section 2.1.2. The QUALM approach suggested in this thesis respects these requirements. In Section 2.2, relevant background information on data mining and text mining are presented. Especially, also data analysis pipelines are defined which are the basis on which QUALM operates. In Section 2.2.3, an overview on toolkits for text mining and data mining is given. Here, the toolkits which are suitable for domain experts without IT and data mining/text mining expertise are focused. In the last section, data quality is addressed (Section 2.3). Firstly, definitions for basic terms with respect to data quality are given (Section 2.3.1) and the definition employed throughout this thesis is presented (Section 2.3.2). Then, an overview on the main research fields in data quality is given (Section 2.3.3). More specifically, also existing approaches to the data quality of unstructured and textual data are considered in Section 2.3.4. Finally, an overview of data quality toolkits is presented in Section 2.3.5.

Parts of this chapter are revised versions of excerpts of previous author publications that are cited at affected locations [Kie16, GKH+16, KGK+17, KRM20]. All concepts in the publications [Kie16, KRM20] were developed exclusively by the author of this thesis. In Gröger et al. and Kassner et al. [GKH+16, KGK+17], the data-driven factory and the new IT architecture SITAM are described. These concepts were not developed by the author of this thesis. Yet, in [GKH+16, KGK+17] requirements for a data quality layer of the SITAM architecture were defined by the author of this thesis, which are summarized in Section 2.1.2.

2.1 Industry 4.0 and the Data-Driven Factory of the Future

In the following Section 2.1.1, Industry 4.0 and the earlier industrial revolutions are addressed for three representative countries: Germany, the USA and China (for a more detailed consideration, see [KKMR15]). All of these countries have very high percentages of manufacturing in their gross domestic products. Moreover, they communicate similar visions of the future of their industries. Several terms such as 'Advanced Manufacturing' and 'Made in China 2025' name these visions respectively. In Germany, new concepts for industry are bundled under the term 'Industry 4.0' and thus this term is also used throughout this thesis [BMB13]. After giving an overview on the development steps of industry in these countries till the fourth industrial revolution (Section 2.1.1), a more concrete vision for advanced data analytics within production in the fourth industrial revolution is detailed and the concept for a data-driven factory is described in Section 2.1.2.

2.1.1 Industry 4.0

In Germany and the USA, industrialization began approximately 1750 [ECD+14]. Firstly, mechanization, steam power and weaving looms were invented and enabled a first industrialization. In the following second industrial revolution, electrical energy, assemly lines and mass production revolutionized industry in Germany and the USA between 1870 and the begin of the first world war in 1914 [ECD+14]. The third industrial revolution added automation, computers and electronics and was started from early 1960 on in Germany and the USA

[BHVh14]. China started introducing the industrialization with the first five-year plans in the 1950s [Sta90]. Moreover, China missed the development steps which came with the third industrial revolution, but currently catches up [Shi14].

Today, the fourth industrial revolution is enabled based on new technological advancements, such as cyber-physical systems, advanced analytics and the internet of things [BHVh14]. Nowadays, all three countries leverage these new technologies to stay competitive [BHVh14, ZH19]. This thesis addresses the topic of advanced analytics and focuses on applications and quality considerations relevant with respect to advanced analytics within the factories of the future. In the next section, a more concrete concept is detailed.

2.1.2 The Data-Driven Factory of the Future

Industry 4.0 comprises many concepts, architectures and application scenarios. Here a concrete example is given: the data-driven factory. In the data-driven factory, all data generated across the entire product life cycle is considered. Structured as well as unstructured data is processed in real-time and by means of advanced analytics. The goal is not full automatization of processes. Rather, human domain experts such as the workers on the shop floor are a central element within the concept. The human domain experts decide, but are supported by means of analysis results. Moreover, their domain knowledge is collected and exploited in the data-driven factory. The main goal of the data-driven factory is to employ all data and knowledge existing within all life cycle phases of the manufactured products. This goal is reached by means of the new IT architecture SITAM (Stuttgart IT Architecture for Manufacturing) [KGK+17], which consists of information layers reaching from data sources till the user. Within SITAM, data quality is considered as overarching topic and as a layer in the architecture and requirements are defined in the course of our research work (cf. Kassner et al. [KGK+17] and Gröger et al. [GKH+16]):

Based on the characterization of the data-driven factory, a corresponding data quality framework needs to especially enable data quality measurement and improvement (1) for all types of data accumulating in the product life cycle, especially also unstructured, e. g., textual data. Moreover, the (2) knowledge of the domain experts must be considered within analytics, (3) data quality methods must be executable implementations which can be calculated concurrently or even before actually carrying data analysis tools out. Finally, high quality needs

to be ensured at (4) all steps from data source to users. Existing data quality frameworks fail to satisfy these requirements (e. g., cf. [WS96, SC13]). In this thesis, they are translated into the QUALM concept (Chapter 4) and concrete data quality methods are presented (Chapters 5-7), which may form the basis to measure and improve data quality within the data-driven factory of the future.

2.2 Data Mining and Text Mining

In this section, terms and basic concepts are described which are relevant to both data and text mining. In Section 2.2.1, background with respect to data mining is presented. These concepts and algorithms are relevant to both text and data mining. In Section 2.2.2, the specialities of text mining are illustrated. Finally, in Section 2.2.3, an overview of data analysis toolkits is presented. In the following, the two terms "advanced analytics" and "structured and unstructured data" are defined which are relevant with respect to data and text mining.

Advanced Analytics

In comparison to classical business intelligence, one difference to advanced analytics is that the latter comprises structured as well as unstructured data and thus comprises data as well as text mining. Moreover, the techniques are oftentimes based on machine learning. Not only questions such as "What happened?", but also questions with respect to the future and actions which can be taken, are answered by advanced analytics [HLK15].

Structured and Unstructured Data

In accordance with Batini et al. [BBCG11], in this thesis all data which comes with an explicitly described data model is defined as structured data (e. g., data in relational databases), and all data that comes without a data model is defined as unstructured data (e. g., texts, videos, pictures and speech).

2.2.1 Data Mining

Besides classic approaches such as association rule learning and regression analysis, a focus of advanced analytics methods lies on algorithms which are able to classify or cluster data instances. While the data and data instances differ in text versus data mining, the algorithms applied are the same. In the following two paragraphs, the characteristics of classification versus clustering algorithms is illustrated and

the descriptions furthermore address how algorithms may deduce rules from data. Moreover, concrete algorithms and sample applications are listed.

Classification and Supervised Machine Learning Approaches

In classification, data instances are assigned to a class out of a list of previously defined possible classes (cf. Witten et al. [WFH11]):

For example, structured data instances such as rows in a data-base table or unstructured text units such as sentences are assigned to one of three topics on a pre-defined list. For example, data instances may represent meals in a restaurant and the pre-defined list of classes may consist of a list of main food types, such as 'pasta', 'salad' or 'sweets'. Sample algorithms which are applied to classification tasks, e. g., comprise Naive Bayes, Maximum Entropy, decision tree learner such as the C4.5 algorithm, Support Vector Machines (SVM) and Neural Networks. At the core, all of these algorithms base on learning from labeled data sets. The list of possible classes to assign is given. This type of machine learning techniques are called 'supervised', since classification decisions are supervised by means of pre-defined classes and labeled data instances, i. e., *training data* (cf. the description of 'Gold Annotated Data Sets' on page 38). In the training phase, a function is learned that can be used to predict the class label for unseen data instances.

In the application phase, the algorithms exploit this function to calculate for each data instance the probability for each class. Finally, the class with maximal probability is assigned, e. g., in the example above the class 'salad' could be assigned. The classification decisions are mainly based on *'machine learning features'*. With respect to the food categorization example employed above, meal ingredients may be employed as features. In a structured database table, features may comprise single columns, i. e., attributes, of the table as well as appended aggregated information. For example, an appended categorical age attribute may be used in addition to an existing date of birth field. In the case of text data, e. g., single words ('unigrams'), pairs of two adjacent words ('bigrams') and information on parts of speech may be used as features in machine learning. Besides features such as the words in a sentence and the attributes in a database table, moreover rules or constraints defined by domain experts may be added to the feature set. For example, rules which employ syntactical characteristics and information such as on critical and normal value ranges, e. g., for measured temperatures, may be added. Sample applications of classifications comprise, e. g., preventative maintenance of electromechanical

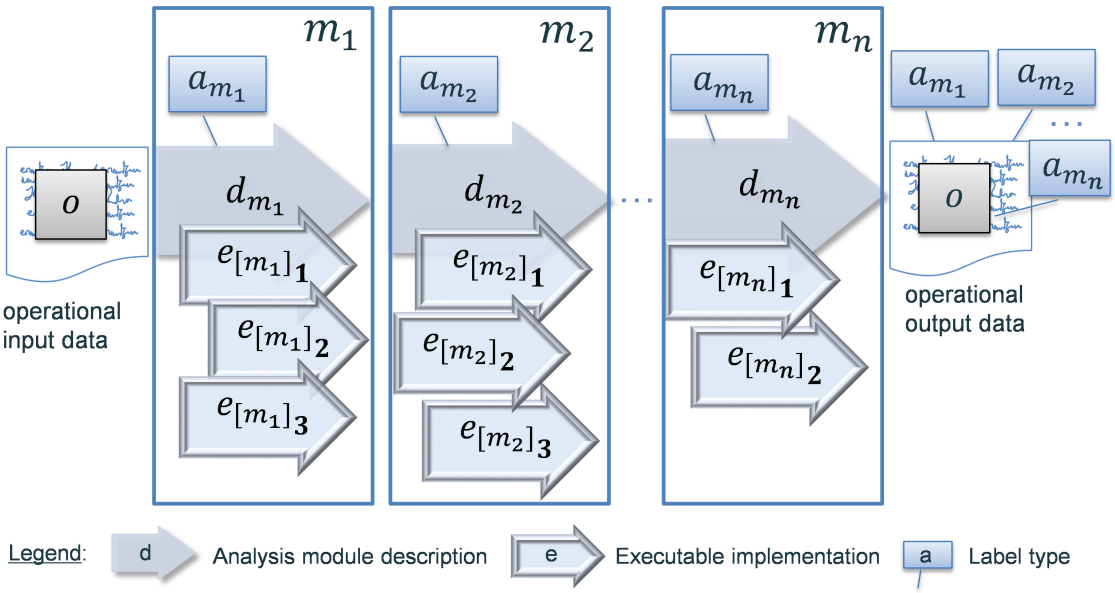


Figure 2.1: Sample formalized excerpt of an analysis process/pipeline based on supervised machine learning modules (cf. Kiefer et al. [KRM20]).

devices such as motors and generators, sentiment analysis/opinion mining and personalized marketing [WFH11, Liu11].

Clustering and Unsupervised Machine Learning Approaches

Besides these supervised techniques, unsupervised methods do not make use of a pre-defined lists of possible classes. Moreover, no labeled training data is employed. Interesting patterns and topics in the data are merely extracted from the data by means of statistics. Here, the algorithms group items that fall naturally together or that are somehow similar with respect to their feature values. For example, with respect to the restaurant example above, many database table items may contain a certain ingredient listed as an attribute, whereas all other instances do not contain this ingredient. The items which contain this certain ingredient naturally fall together and naturally build a group or 'cluster'. Also, sentences describing meals may contain a certain word very frequently. Thus they naturally fall together as a group/cluster.

Analysis Modules, Tools and Pipelines

In this paragraph, a formalization of analysis pipelines is given that is based on supervised analysis modules (cf. Kiefer et al. [KRM20]). In Figure 2.1, a sample formalization of an excerpt of an analysis pipeline is presented. A concrete

example for these formalizations is given in Figure 1.1. The operational input data o is read and then analyzed by consecutive *analysis modules* m_k. Each analysis module has a corresponding description, an annotation or label type, and several executable implementations. To indicate the affiliation of pipeline elements to the analysis module, a subscript m_k is added to each corresponding element. The module description d_{mk} is the name of the analysis module m_k, which are added for easy comprehensibility. For example, module descriptions, such as "Optical Character Recognition" (OCR), "speech-to-text", "POS-tagger" and "Named Entity Recognizer" exist. A module m_k adds labels of type a_{mk} to the operational data, where a_{mk} denotes one label type. Further examples of label types are "character", "language", "part of speech" and "named entity". Moreover, several executable implementations/analysis tools $e_{[m_k]_g}$ are possible for each module m_k. For example, various concrete implementations of POS-taggers exist (cf. Section 5.3.2). The result of the analysis pipeline is the operational data set now enriched with annotations/labels.

Semantic Resources

In text mining, semantic resources, such as dictionaries, taxonomies, ontologies and abbreviation lexica are employed frequently. For example, the lexical database WordNet[1] and the German pendant GermaNet[2] store words and their senses and make this information available in computer-readable form. Moreover, many domain-specific semantic resources exist, e. g., in the automotive sector [ST10] or on movie reviews [HL04]. But also in mining structured data, additional knowledge is added by means of semantic resources [Nig07]. For example, domain knowledge is integrated into data mining processes by means of ontologies in "semantic data mining" approaches [DWL15].

Storage of Structured and Unstructured Data, Training Data, Semantic Resources and (Intermediate) Analysis Results

An evaluation of various technologies to store and link structured and unstructured data has been performed in the course of a student project that has accompanied the work on this thesis [Die17]. Here, Alfresco[3], Cloudera[4]

[1] `https://wordnet.princeton.edu/`
[2] `http://www.sfs.uni-tuebingen.de/GermaNet/`
[3] `https://www.alfresco.com/de/`
[4] `https://de.cloudera.com/products/data-warehouse.html`

and Apache Marmotta[5] have been compared with respect to criteria such as "interfaces", "integrated search" and "support of analysis".

Recent research conceptually suggests a "data lake" for storing data of all types (cf. Terrizzano et al. [TSRC15]). Especially, structured as well as unstructured data, knowledge resources such as dictionaries and taxonomies, analysis results as well as intermediate analysis results are stored at the same place and are easily accessible [MGS+19, CSN+14, GGH+19]. Concretizations of this concept employ and combine various technologies, such as combining databases with content management systems and with storages for big and streaming data [Grö18].

Evaluation Metrics

Since the algorithms for classification in data and text mining are the same, also similar evaluation methods are applied to both data and text mining. The main difference yet lies in the nature of the gold-labeled "test" data sets and the labels needed, also see the next paragraph and Section 2.2.2.

Gold-Annotated Data Sets

Annotating means to tag data instances, such as rows in a database table, images, videos or text documents with a choice from a pre-defined set of possible classes, i.e., labels are added manually to the data instances [HHLRP12]. Such gold-annotated data sets are the basis for training and evaluation of algorithms which use annotations [WAMP14].

For example, to be able to evaluate the quality of predictions made by a classifier, the predicted classed are compared to the labels as stated by a human annotator. Usually this annotator is a domain expert, who manually assigns classes to parts of the data. For example, the owner of the restaurant in the example employed above, may manually add labels such as "sweets" and "pasta" to the data instances. Such distinct gold, i. e., manually annotated/labeled data sets, are needed both as training data for many machine learning algorithms, as already mentioned above, and furthermore as basis for the evaluation of the predictions made by these algorithms, i.e., as test data.

Also, high quality gold labels need to be ensured. To this end, for example annotation guidelines may help. The manual annotation of gold labels is often time-consuming and difficult [SPI08], e.g., consider the elaborate annotation guidelines developed for the labeling of entities and their relations in industry texts in a student work accompanying our project [Gra17].

[5] `https://marmotta.apache.org/`

Table 2.1: Illustration of the confusion matrix

		Predicted class	
		+	-
Actual class	+	tp	fn
	-	fp	tn

Precision and Recall

Precision and recall are two of a whole list of possible evaluation metrics which may be employed to evaluate classifiers (cf. [PA11] for an overview on evaluation metrics). The metrics may be employed in the context of a binary classification problem. If multiple classes are predicted, they may still be employed in the evaluation of each individual class. As basis for the calculation of these evaluation metrics, gold labels are needed, as just described. For each data instance in the test set the classifier needs to predict its class. Then it can be examined for each single class how good a classifier is in predicting it by comparing the predictions with the actual labels in the test data.

Here, *tp* denote the true positives, i. e., the data instances which are predicted to be in the class and indeed also are in that class in the gold data set. False positives (*fp*) denote those data instances which the classifier said to belong to the class but really, as judged based on the gold data, do not. Finally, the false negatives, *fn*, denote those data items that are judged to be in the class as based on the gold data, but were not predicted so and thus are missed by the classifier. The values for *tp, fp, tn* and *fn* are oftentimes presented in a two-dimensional contingency table, which is called confusion matrix in the machine learning field. For instance, in the rows of such a table the actual class is noted and in the columns of the table the predicted class is noted, as illustrated in Table 2.1 for the two labels + (positive) and - (negative) as is used by convention (cf. [PA11]). For a generalization of the confusion matrix for multiple classes, see [Man16].

In the following, the two evaluation metrics, precision and recall, are stated in Equations 2.1 and 2.2. As shown in Equation 2.1, the precision, e.g., reflects the proportion of true positives *(tp)* with respect to all instances predicted to be of class 'positive' by the classifier *(tp + fp)*. The recall, as shown in Equation 2.2, rather makes a statement on how many of those instances with actual class positive, i.e. *(tp+fn)*, are correctly classified as being positive by the classifier *(tp)*.

$$Precision = \frac{(tp)}{(tp + fp)} \tag{2.1}$$

$$Recall = \frac{(tp)}{(tp + fn)} \tag{2.2}$$

Accuracy

As shown in Equation 2.3, the accuracy of a classifier is calculated by dividing the number of all true results (correctly predicted) by the number of all data instances. This evaluation metric is used as a standard evaluation metric for the automatic part-of-speech tagger (POS-tagger) and language identifier which are investigated in this thesis. The quality of a POS-tagger for example is determined by comparing the tags predicted by the system (by the tagger) with manually added labels. The equation to calculate the Accuracy of a POS-tagger is given here as Equation 2.4.

$$ACC = \frac{(tp + tn)}{(tp + tn + fp + fn)} \tag{2.3}$$

$$ACC = \frac{(\#\ correct\ POS\ tags\ in\ tagged\ data)}{(\#\ total\ POS\ tags\ in\ tagged\ data)} \tag{2.4}$$

Correlation Metrics

With a correlation metric, two data rows are investigated with respect to their correlation. It is applied to two data rows X and Y with n values: $X = x_1, ...x_n$ and $Y = y_1, ...y_n$. A positive correlation value implies that if x rises, y rises as well. A negative value means that x and y correlate in the opposite direction, i. e., when x rises, y falls.

In Formula 2.5, it is shown how Pearson correlation is calculated for two rows $X = x_1, ...x_n$ and $Y = y_1, ...y_n$, where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{2.5}$$

Spearman's rho is a further correlation metric. It is based on the Pearson correlation metric r and is defined as the Pearson correlation coefficient between the ranks of ordinal scaled data. To calculate the Spearman correlation metric,

the raw scores in X and Y are converted to ranks rg_{x_i} and rg_{y_i}. Then, Spearman's rho,r_{SP}, is calculated as $r_{SP} = r_{rg_X, rg_Y}$ and thus as specified in Formula 2.6.

$$r_{SP} = \frac{\sum_{i=1}^{n}(rg_{x_i} - \overline{rg_x})(rg_{y_i} - \overline{rg_y})}{\sqrt{\sum_{i=1}^{n}(rg_{x_i} - \overline{rg_x})^2}\sqrt{\sum_{i=1}^{n}(rg_{y_i} - \overline{rg_y})^2}} \tag{2.6}$$

The Bravais-Pearson correlation [Kal12] assesses linear relationships and a normal distribution of X and Y is assumed. Spearman's rho rather assesses monotonic relationships and this metric may also be applied to data rows X and Y for which normal distribution does not hold.

In addition to correlation metrics, oftentimes the *p-value* (probability value) is used for hypothesis testing and is reported. It hints at the probability of an erroneous conclusion. As stated in Equation 2.7, the p-value is associated with the null hypothesis, i.e., with the hypothesis that the two data rows X and Y do not correlate (cf. [Mit16]). The p-value reflects how well the data rows, e.g.,X and Y, reject this null hypothesis. The p-value is defined as the probability of obtaining results as extreme or more extreme, given that the null hypothesis is true. Thus, it states the probability with which a non-correlating system produces data rows X and Y (cf. Fisher et al. [Fis35]).

The given significance level is denoted with α in Equation 2.7. Alpha is oftentimes set to 0.05, which means that the probability of achieving the same or more extreme results assuming the null hypothesis is 5%. It is compared to the empirical significance value, i.e., with the p-value α'. If α' does not exceed the value α, the null hypothesis is rejected.

$$Rejection\ of\ the\ null\ hypothesis\ H_0 \Leftrightarrow p{-}value\ \alpha' fulfills\ the\ condition\ \alpha' \leq \alpha \tag{2.7}$$

2.2.2 Text Mining

In the following sections background topics relevant for this thesis will be described which are special to text mining. Especially, the extensive preprocessing phase and corresponding methods are unique to text mining. In preprocessing of structured data, e. g., data sets need to be transformed and attributes are re-engineered. The preprocessing phase needed in text mining is different. Here, the

large string of text and all information hidden therein needs to be made usable by machine learning algorithms. Firstly, a description of relevant preprocessing steps special to text mining and explanations on how texts may be represented in a form that is usable by machine learning algorithms are given. Then, named entity recognition, sentiment analysis and text similarity metrics are described.

The Preprocessing Phase in Text Mining

While preprocessing in text mining encompasses a huge list of possible methods, e. g., tokenizer and syntax parsers, here the methods presented are restricted to those relevant as background knowledge to this thesis (cf. Jurafsky and Martin [JM09]):

Language Identification

Automatic language identification is based on classifiers that use short character subsequences as features. Most languages have distinctive signature patterns which may be employed in correctly recognizing the language of a text. A first work towards automatic language identification comes from cryptography and is based on a character-level n-gram language identification algorithm (Konheim [Kon81]). While other methods such as looking for distinctive function words and letter combinations have also been used, most methods rely on character n-gram techniques [JLZ+19].

Tokenization

In tokenization the large input text string is segmented into *tokens*, e. g., into the smallest meaningful text units such as the words in a text [JM09]. Existing approaches mainly rely on supervised machine learning. While at first glance the task might seem straightforward, many issues such as determining word boundaries for compounds, hyphenated words and contractions or determining word boundaries in languages without whitespaces such as Chinese arise. Many issues in tokenization are language-specific. It thus requires the language of the document to be known. In other words, the langauge of the text must be determined beforehand using a language identifier (see above) and is then used as decisive feature for the selection of an appropriate tokenizer.

Part-of-speech tagging (POS-tagging)

Bußmann defines a part of speech of a word, such as verb or noun, as “the result of a classification of all words in a language” [Buß08]. Different classification systems for each language exist, but they all try to group words according to their characteristics. Therefore part-of-speech tagger differ not only in terms

of the machine learning algorithm, but also in the tagset they use. A tagset denotes the set of all parts of speech that the tagger may assign. The English version of the Stanford tagger, e. g., uses the Penn Treebank tag set. In this tagset approximately 45 POS tags exist [TMS03]. For example, the output of the Stanford part-of-speech tagger for “the best of everything” is: the (DT) best (JJS) of (IN) everything (NN). The short identifiers DT, JJS, IN and NN indicate the parts of speech "determiner", "superlative adjective", "preposition or subordinating conjunction" and "noun". The general approach to part-of-speech tagging is to use a sequence model such as Hidden Markov Models. Here, previous and/or following tags are used to predict the current part of speech tag. For a detailed overview on part-of-speech tagging methods (cf. Jurafsky and Martin [JM09]).

Dependency Parsing Dependency parser add annotations with respect to the grammatical structure of the sentences in a text. The annotated dependencies indicate relationships between words such as on compounds and subject or object relationships in the sentence (cf. [JM09]).

Stemming and Lemmatization

For many applications in natural language processing and text mining, crucial preprocessing steps aim at grouping and/or "normalizing" words which have different word forms but the same meaning [JM09]. Words for example indicate grammatical functions such as singular and plural or case, e. g., in their suffixes. In stemming, all suffixes of the words in a text are eliminated/cut off. In lemmatization, rather all words are transformed to the base form as can be found in a dictionary. For example, the German word "Freiheiten" is reduced to the base form "Freiheit" by a lemmatizer and to "frei" by a stemmer. The algorithms for stemming and lemmatization are usually rule- and dictionary-based. A prominent stemming algorithm was developed by Porter [Por80].

Grouping Synonyms

Similarly to stemming and lemmatization as described above, also a grouping of synonymous words, i. e., of different terms which have the same meaning, is oftentimes used as preprocessing step in text mining applications. For example, the German words "Auto", "Personenwagen" and "KFZ" all denote a car and thus have the same meaning. Thus, all three may be substituted by one of the words, e. g., "Auto", so that subsequent algorithms can employ the information that all three words denote the same thing in the real-world. Most existing

approaches rely on dictionaries, co-occurrences and word embeddings (cf. Kubek [Kub19, LVDv16]).

Word Sense Disambiguation

The classic word sense disambiguation task is on selecting one of several possible senses of a word in a text. Disambiguation is based on a lexicon which provides word senses for the different words and on context and grammatical information. For example, without context, the term "smart" may have at least two senses denoted in a lexicon. Firstly, the word could be denoting a car manufactured by the Daimler AG, secondly it as well may denote a characteristic of a person. As context, surrounding words as well as grammar may be used to decide on the sense of the word "smart" in a certain sentence. For example, in "he is smart", the word describes a person, while in "will all of these shopping bags fit into your smart?" the word "smart" denotes a car. Besides the words occurring in the context, moreover information such as parts of speech and named entities, which were annotated in foregoing analysis steps of the text analysis pipeline, may be employed as features of word sense disambiguation systems. For instance, "smart" is correctly annotated as part of speech "adjective" in the first example and as part of speech "proper noun" in the second example. State-of-the-art approaches to automatic word sense disambiguation rely on semantic resources and classification algorithms from machine learning [Pop18].

Representations of Texts: TDM, DTM and LSA

The basis where, e. g., classification and clustering algorithms operate, are representations of texts (cf. Jurafsky and Martin [JM09]). Text documents may be represented by means of vectors. A vector may consist of a list of identifiers such as the tokens, i. e., terms, in the document. With respect to each identifier, e. g., a term, moreover a weight is noted. For example, a binary weight system uses 0 to indicate that the term is not present in the text document and 1 if it is. Other possible weighting schemes are based on the term frequency (*tf*) or term frequency times inverse document frequency (*tf-idf*). The term frequency *tf* is calculated by counting the number of occurrences of a term in a text document. The *idf* is calculated by dividing the number of all documents in the considered text document collection by the number of documents that contain a certain term. Thus, *idf* downweights terms which occur frequently in all documents and thus are not specific and may furthermore be less helpful in text mining and information retrieval. On the other hand, rare terms are upweighted if *tf-idf* is applied in comparison to the plain *tf* weight. The resulting vectors may then

be compared based on cosine similarity (cf. Jurafsky and Martin [JM09]). In a document-term-matrix (DTM) moreover, the words within the whole document collection are identifiers. In this matrix, rows correspond to documents in the collection and columns correspond to terms. A term-document matrix (TDM) denotes the terms in the rows and documents in the columns of the matrix. These matrices can get very large, consisting of tens of thousands of rows and columns.

These matrix representations of text document collections capture their content with respect to the words in the document collection. The approach described next, LSA, reduces the dimensionality of the matrix, e.g. TDM, with the goal of changing it so that it rather represents "meanings" or "concepts" than "words" in a text document (cf. Anandarajan et al. [AHN19]).

The standard text representations have two prominent problems: They do not address 1) synonymy and 2) polysemy in language [DDF$^+$90]. For example, *car* and *automobile* are synonyms, i.e., two words which have the same meaning. Still, these are listed as two different dimensions in the text representation, e.g., in the TDM. In difference, a polysemous word has multiple meanings but only one term. For example, the term *bank* may denote a building, a seating facility in a park, and a financial institute. In the standard text representation, e.g., in a TDM, these multiple meanings are not captured but fall together in one dimension of the matrix. These problems can be addressed by means of the Latent Semantic Analysis (LSA) method suggested by Deerwester et al. [DDF$^+$90]. LSA reduces the number of terms in the matrix A, e.g., in the TDM, based on Singular Value Decomposition (SVD). By means of Singular Value Decomposition (SVD), a low-rank approximation A_k of the TDM A is constructed. The number of dimensions in the new matrix, i.e. the value of k, is a lot smaller than the original rank of A. All values in a TDM A are non-negative values which represent tf, $tf - idf$ or presence and absence of a term in a document by means of the values 0 and 1. The rank of a $M * N$ matrix is determined as $rank(A) \leq min\{M, N\}$. By means of SVD, the TDM may be mapped into a 'semantic space' with lowered rank. To this end, firstly the original TDM is decomposed into the three matrices U, Σ and V^T, as shown in Equation 2.8.

$$A = U\Sigma V^T \tag{2.8}$$

The first of these matrices is the SVD *term* matrix U. This is a $M * M$ matrix whose columns are the orthogonal eigenvectors of AA^T. The transposed matrix A^T is the same as A but the row and column indices are switched. For a square matrix C with M rows and M columns and a non-zero vector $\vec{x}$, the eigenvalues of C are the values of λ satisfying Equation 2.9. The right eigenvector is the N-vector $\vec{x}$ satisfying Equation 2.9 for an eigenvalue λ.

$$C\vec{x} = \lambda\vec{x} \tag{2.9}$$

In the context of term-document matrices, AA^T is a square matrix with a row and a column corresponding to each of the M terms. The entry (i, j) in the matrix indicates the degree of co-occurrence of the terms, based on the number of documents in which the ith and jth terms co-occur. This matrix contains new semantic information on co-occurrences of terms. The values along the diagonal matrix Σ are called *singular values*. They are the square roots of the eigenvalues of A^TA and are noted in descending order on the diagonal of Σ. Finally, V^T is the SVD *document* matrix. The columns in V are the orthogonal eigenvectors of A^TA. This matrix contains information on correlations of documents over terms. Based on this decomposition, the TDM can furthermore be lowered in rank. In a first step (1) all but the largest values in Σ are zeroed out, in a second step (2) these zero-values are truncated. By means of forcing the TDM into a lower rank in a new 'semantic space', many synonymous terms are collapsed together. Thus, LSA helps to address the problem occurring due to synonyms in language as present in TDM text representations. LSA also partially addresses the problem with polysemy, since for a given polysemous term, values for co-occurring and rather similar-meaning terms/documents tend to add up and values with respect to terms/documents with very different meanings tend to decrease or cancel out. For concrete examples and the mathematical background we refer the reader to Deerwester et al. [DDF$^+$90].

Text Clustering

Specifics in text clustering are mainly due to the text preprocessing steps described in this section. For example, not only tokens, but also lemmas or stems may build the basic data instances on which clustering algorithms operate. Also, e. g., word senses and synonyms may have been dissolved, before clustering algorithms start to operate. In text clustering, clusters may be named straightforwardly, by means of the most frequent term in the cluster. Well-known

algorithms for clustering include k-means, DBSCAN, Expectation–Maximization and Agglomerative Hierarchical Clustering (cf. Witten et al. [WFH11]). Applications comprise preprocessing for classifiers such as spam filters and basic pattern recognition for market research and image processing.

Named Entity Recognition (NER)

A named entity may be, e. g., a person, a location, company or a machine. In text mining, named entities can be recognized automatically, based on machine learning approaches for classification. Most traditional approaches employ "sequence models" (cf. Nadeau and Sekine [NS07]). Herein, the named entity of a current word in a text is predicted based on knowledge about that word, as well as on all preceding and subsequent words and their grammatical and other information. More recent approaches rely on deep learning models (cf. Yadav and Bethard [YB18]).

Sentiment Analysis

Sentiment analysis, also called opinion mining (cf. Liu [Liu15]), is a classification task in text mining. Here, the possible text units considered in this classification task reach from sentences over paragraphs to articles and whole text data sets. The list of pre-defined classes usually comprises at least the three sentiments "positive", "negative" and "neutral". All machine learning algorithms may be applied to this classification task and moreover also semantic resources such as dictionaries with positive and negative words are employed frequently.

Text Similarity Metrics

A text similarity metric is a number in the interval [0,1] which states how similar two text strings are. If the texts are the same, the text similarity value is 1, if the texts are very different from each other, a text similarity metric of 0 will be calculated. Many possible metrics exist, e.g., cosine similarity, Jaccard and Levenshtein. Some metrics consider phonetic or corpus-based similarity [GF13]. Some metrics mainly base on a comparison of n-grams of characters in the texts or on n-grams of whole strings such as words. For example, the Jaccard metric operates on two vectors of characters or strings and is displayed in Equation 2.10 for vectors $\vec{t}$ and $\vec{e}$ (cf. [JM09]):

$$Jaccard(\vec{t}, \vec{e}) = \frac{\sum_{i=1}^{N} min(t_i, e_i)}{\sum_{i=1}^{N} max(t_i, e_i)} \tag{2.10}$$

The cosine similarity measure for comparing two vectors t and e, e.g., representing two text documents, is defined as shown in Equation 2.11 (cf. [JM09]):

$$\cos(\vec{t}, \vec{e}) = \frac{\vec{t}\vec{e}}{\|\vec{t}\|\|\vec{e}\|} = \frac{\sum_{i=1}^{N} t_i e_i}{\sqrt{\sum_{i=1}^{N} (t_i)^2}\sqrt{\sum_{i=1}^{N} (e_i)^2}} \tag{2.11}$$

2.2.3 Toolkits for Data and Text Mining

Several toolkits for the analysis of structured data as well as unstructured text data exist. In this thesis, toolkits are focused which enable easy usability also by domain experts without IT and data/text analysis skills. Usually toolkits with graphical user interfaces, which do not presume knowledge of a programming language, are preferred by these analysts. Therein, analysis pipelines (cf. Section 2.2.1) can be built easily from scratch by dragging and dropping graphical tools on the screen and by employing pre-defined tools/rules and schemata. Example toolkits include RapidMiner[6], IBM SPSS[7], the Konstanz Information Miner (KNIME)[8], the Waikato Environment for Knowledge Analysis (WEKA) GUI[9], FlexMash[10] (cf. [HB16]) and the Leipzig Corpus Miner (LCM)[11] (cf. [NWH17]). In the IBM Watson Analytic Toolkit[12], which was recently replaced by IBM Cognos Analytics[13], moreover, natural language questions can be asked to the data and ease the usability of the toolkit. For example, questions such as "What is the relationship between age and salary?" are understood and automatically transferred to SQL-queries. Also, many programming libraries make it easy for analysts to build data and text analysis pipelines "out-of-the-box", by means of default analysis tools with default specifics such as default training data. Among others, DKPro Core[14] and scikit-learn[15] need to be mentioned. Moreover, especially for natural language processing and text mining, oftentimes

[6] https://rapidminer.com/
[7] http://www.ibm.com/analytics/us/en/technology/spss/
[8] https://www.knime.com/
[9] https://www.cs.waikato.ac.nz/ml/weka/index.html
[10] https://github.com/hirmerpl/FlexMash
[11] http://www.epol-projekt.de/tools-nlp/leipzig-corpus-miner-lcm/
[12] https://www.ibm.com/watson-analytics
[13] https://www.ibm.com/de-de/products/cognos-analytics
[14] https://dkpro.github.io/dkpro-core/
[15] https://scikit-learn.org/stable/

NLTK[16], OpenNLP[17] Stanford Core NLP[18] and Stanford NER[19] are employed. Finally, TensorFlow is used in building neural networks for machine learning applications[20].

While some of the toolkits provide standard data quality metrics for structured data, such as assessing the percentage of null and out-of-domain values, data quality methods that operate within text analysis pipelines are missing. Moreover, unstructured data quality is not addressed by the toolkits.

In the course of a student project that has accompanied the work on this thesis [KLS16], standard toolkits for natural language processing were investigated and compared with respect to their robustness towards messy text data. Here, NLTK, Stanford CoreNLP, OpenNLP and DKPro were compared. Based on a comparison of the result quality for POS-tagging tools, all of these standard toolkits were found to have problems with messy text data such as chat posts and tweets. The results of this study are extended and presented in Section 3.4.

2.3 Data Quality

The research presented in this thesis is already placed in the context of Industry 4.0 and the data-driven factory in Section 2.1. Relevant background in data mining and text mining is described in Section 2.2. Given this context, this section comes closer to the core field touched by the research presented in this thesis, namely data quality. This section is started by giving definitions for basic terms with respect to data quality in Section 2.3.1. Then, the definition of data quality used in this thesis is detailed (Section 2.3.2, based on [Kie16]). Next, an overview on the main research fields in data quality is given in Section 2.3.3, before approaches to data quality for unstructured text data is focused in Section 2.3.4. This section is concluded with an overview on data quality toolkits and their characteristics (Section 2.3.5).

[16] https://www.nltk.org
[17] https://opennlp.apache.org/
[18] https://stanfordnlp.github.io/CoreNLP/
[19] https://nlp.stanford.edu/software/CRF-NER.shtml
[20] https://www.tensorflow.org/

2.3.1 Definitions for Basic Terms With Respect to Data Quality

In this section, definitions for basic terms with respect to data quality are given, such as "data consumer", "data quality dimension" and "data quality indicator".

Information Quality and Data Quality

Besides data quality, also the term **information quality** is used in plenty research works [Hil11, BS16]. This term was already employed early to describe the quality of the content of information systems [GH07]. Others draw a distinction between data and information quality in terms of elementary and aggregated data [MLV+03] or in terms of raw component data items and information products [SWZ00, Wan98]. Further research works use the terms data and information quality interchangeably, though [MWLZ09]. In this thesis, this latter view is adopted and furthermore mainly the term "data quality" is used in this thesis.

Data Consumer

Data/information is consumed (= processed and used) by humans and machines (algorithms). Sample machine consumers present in the preprocessing phase in text mining are listed in Section 2.2.2.

Data Quality Dimension

The multi-faceted nature of data quality is often expressed using dimensions, such as accuracy, timeliness and consistency (c.f. [WS96]).

Data Quality Method

In this thesis, concrete methods to assess and improve data quality are subsumed under the term "data quality method". In cases where the methods are considered separately, the terms "data quality indicator" and "data quality modifier" are used respectively to denote methods to assess and improve data quality.

Data Quality Metric

Data quality metrics may transfer data characteristics to a number in [0,1] where 0 indicates low data quality and 1 indicates high data quality. This definition is similar to standard descriptions of data quality metrics, such as the one given in [BBCG11].

Data Quality Indicator

A data quality dimension can be estimated using data quality indicators (also see Section 2.3.2). Data quality indicators must be transferable to a data

quality metric. Indicators can, e. g., be represented by confidence and similarity measures, yes/no-questions, or by proportions of data items which have a certain characteristic. If a data quality metric is not directly calculable, e.g., since information needed for the calculation is not available, a data quality indicator may be used, whose value may be determined without the missing information needed.

Data Quality Modifier

Besides assessing data quality by means of data quality indicators, data characteristics may moreover be changed/modified with the goal of improving quality. For example, a data quality modifier might dissolve duplicates or correct spelling mistakes.

2.3.2 Definition of Data Quality used in this Thesis

In this section, the definition of data quality employed in this thesis is illustrated (cf. [Kie16]). Data quality is a multi-faceted concept. Thus, plenty definitions of data quality exist: subjective [WS96] as well as objective ones [PLW02], definitions which base on lists of data quality characteristics or dimensions [iso], definitions which focus on the purpose [Jur88] or which compare the data to accurate "gold" data sets [FLR94]. The definition of data quality employed in this thesis is based on the definitions by [SC13] and [WS96]. Here, the data quality of a data set D is described by its similarity to the data set D' which is expected by the data consumer [SC13]. [WS96] defines data quality via the fitness for use by the data consumer. While [WS96] has a human consumer in mind, non-human/machine consumers of unstructured data are focused in this thesis. Furthermore, data quality needs to be defined in terms of accuracy. The accuracy dimension describes the similarity between the input data and the data which would be representing the real world. This definition of the data quality dimension accuracy is equal to existing ones, e. g., [FLR94].

All of the dimensions that were found to be relevant in the literature, such as completeness, timeliness and accuracy are relevant to structured as well as unstructured data. From these dimensions three dimensions which are relevant to text analysis pipelines and with respect to analysis pipelines for unstructured data are selected.

The dimensions are deduced from the elements involved in analytics: The input data, the real world, data consumers, analysis tools, a task and the knowledge extracted. Based on these elements, the quality of data D can be determined by comparing it to three classes of ideal data sets. Firstly, the data is compared to the data as expected by the current data consumer or analysis tool D_C (which will be called the Interpretability dimension). Secondly, it is compared to the data as it would be optimal for the task D_T (Relevancy). Finally, the data moreover needs to agree with reality and thus with the data set which is representing the real world D_W (Data Quality Dimension Accuracy). The deduced dimensions are also in line with the data quality definitions stated above. In Figure 2.2, the three classes of ideal data sets are illustrated in the context of a text analysis pipeline. Ideally, D would match the real world D_W and would be exactly the same as the data expected by the first analysis tool, i.e., by the first data consumer. Since unstructured data is analyzed in a pipeline, the output of the first data consumer is input to the second and should therefore match the data expected by the second data consumer and so on. For example, the result of the text analysis pipeline in Figure 2.2 is subsequently consumed by a human analyst, who, e.g., prepares a presentation or a report based on the analysis results. Ideally this presentation or report perfectly fits with D_T and is moreover also accurate. By basing the data quality dimensions on the elements involved in analytics, especially analytics for unstructured data, the quality of data, which is analyzed automatically in analysis pipelines, is focused.

In the following, the deduced data quality dimensions are described in more detail:

Accuracy is an important data quality dimension for both structured as well as unstructured, e.g., text data. D_W may be represented by a so-called gold standard data set with the accurate values labeled manually by human experts (cf. Section 2.2). Thus accuracy is hard to be measured, because the data set D_W, which represents the real world, is often not known and creating it as a gold standard involves the work of human experts, is time-consuming, costly or even impossible. The solution is usually to abstract away from details, e. g., by using rules to check general conformance of data points with expected patterns (e. g., e-mail addresses containing an @ sign) or to built D_W manually for a part of the data set only (cf. [VHD+14, WLB+16]). For example, statistical classifiers are evaluated by comparing the prediction of the statistical classifier

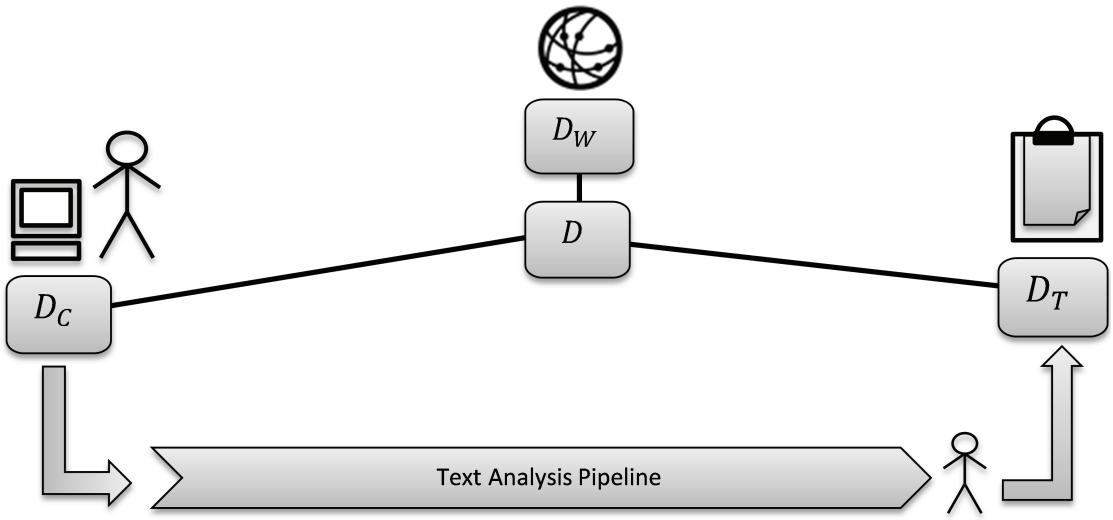


Figure 2.2: The three ideal data sets D_C, D_T and D_W in the context of the elements involved in analytics.

with those in a gold standard with manually labeled classes (cf. Formula 2.4). Since D_W is not known for all data sets D, many statistical classifiers can not be evaluated and the number of problems with accuracy in big databases can only be approximated.

Interpretability can be assessed as the degree of similarity between the data sets D and D_C. For example, consider a processor of an analysis pipeline which is used to segment a text into sentences. The processor is trained on Chinese texts and thus D_C consists of Chinese texts. The input data D to the processor consists of English texts. Since D and D_C are not similar, data quality is low. In interpreting unstructured data many different processors and corresponding data sets D_C are involved. Therefore, the interpretability dimension is especially crucial for unstructured data.

Relevancy can be assessed as the similarity between D and D_T. As an example for relevancy, consider a worker on the shop floor who is searching in a knowledge base for a solution for an urgent problem with a machine. If he only finds information on the price of the machine, the data quality of the result is low because it does not help him with his task of solving the problem.

The *Interpretability and Relevancy* of a data set D are assessed by its similarity to the data set D_C which is expected by the data consumers. Expectations differ

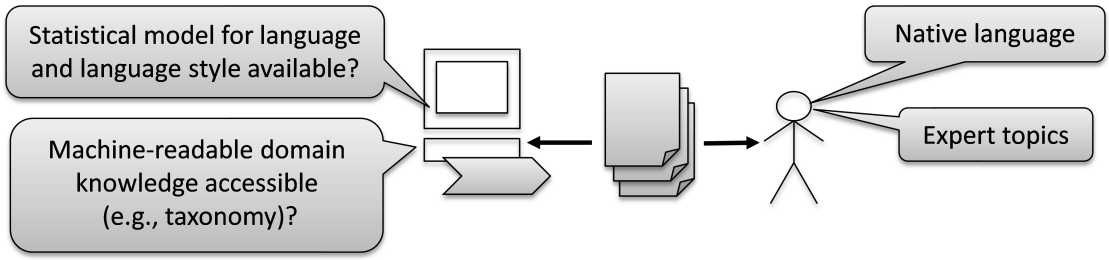


Figure 2.3: Machine and human data consumer and factors that influence the data expected.

from human to machine consumers. What a human data consumer expects, depends on subjective factors, such as his knowledge, experiences and goals. Expectations of machine consumers are very precise and depend on the algorithms, training data, statistical models, rules and knowledge resources available. This holds for all types of unstructured data. As illustrated in Figure 2.3, unstructured data such as textual documents may be consumed by machines or humans and the data set D_C depends on factors such as the native language of the human and the statistical language models available to the machine. For example, a human data consumer expects a manual for a machine to be in his native language or in a language he knows. He also expects the manual to explain the machine in a way he understands with his technical expertise. When a machine consumes unstructured data, factors such as statistical models available influence the interpretability and more precisely the similarity of the input data and the data expected. The knowledge of a machine consumer can be represented by machine-readable domain knowledge encoded in semantic resources (such as taxonomies), by training data, statistical models or by rules. As an example, imagine a machine consumer that uses a simple rule-based approach to the extraction of proper names from German text data, where all uppercased words are classified as nouns. This machine consumer expects a data set D_C with correct upper and lowercased words. If D is all lower-cased, D_C and D are not similar and the data is not fit for use by that data consumer.

Unstructured data is usually consumed by many different data consumers with many different data sets D_C, which are expected by human data consumers and analysis tools in analytics. In an analysis pipeline, the raw data is consumed and processed by several consumers in a sequence and the output of the previous

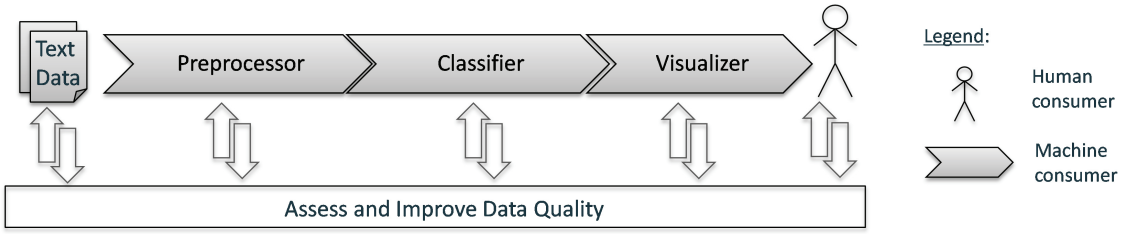


Figure 2.4: Assessing and improving data quality for each data consumer on the way from raw text documents to the final human consumer.

consumer is the input to the next consumer and so on. Data quality problems at intermediate consumers may be automatically propagated to following consumers. By considering all intermediate (machine and/or human) consumers, the exact points for data quality improvement can be determined. In Figure 2.4 an analysis pipeline is presented involving three machine consumers and one human end consumer of the data. Machine consumers are represented in this illustration by three high-level machine consumers which are present in many analysis pipelines of unstructured data: preprocessors, classifiers and visualizers. For example, the output of the preprocessor is input to automatic classification and the results are then visualized. The visualizations are finally the input to a human consumer of the data, who, e. g., derives decisions from it.

The concrete consumers of data differ from task to task and depend on factors such as the resources and the data available. For each of these categories, many different concrete consumers exist. They may differ in terms of the algorithm, training data, statistical model and machine-readable domain knowledge available and therefore also differ in terms of the data D_C they expect. Thus, all of these elements need to be considered by a holistic approach to data quality assessment in text analysis pipelines.

Data quality dimensions can be concretized by exploitation of data quality indicators, as will be shown in Chapter 5 for the interpretability dimension. This dimension is also the focus of the QUALM concept and concrete QUALM data quality methods, as suggested in this thesis.

2.3.3 Main Research Fields in Data Quality: An Overview

In this section, a broad overview on the main research fields in data quality is given. It is concluded with a first classification of the research presented in this thesis. In the concrete examples given in this section, structured and semi-structured data are focused. Approaches which consider unstructured and especially text data will be addressed in the subsequent Section 2.3.4.

Active research on data quality exists. The research topics and communities are as multi-faceted as the concept of "data quality" itself. Research is so numerous that even several classification schemes for data quality research exist [MWLZ09, GH07, Sad13]. Data quality research can for example be organized by topics and methods [MWLZ09], by the three main categories assessment, management and context [GH07], by dividing theoretical from practical approaches [LMV06] and by categorizing research into organizational, architectural and computational aspects of data quality [Sad13]. Research focused on organizational aspects, e. g., is conducted by business analysts and is about data quality management and the people, processes, policies and standards involved [Sad13]. Architectural research provides the technology landscape needed to deploy these processes and standards, and computational research provides appropriate tools and techniques [Sad13].

One big research area on data quality comes from the **business** side. Research from this business perspective for example investigates the impact of data quality on return of investment [Zod15]. Plenty research works suggest data quality management methodologies. For example, Woodall et al. survey existing data quality management techniques such as AIM Quality (AIMQ) and TDQM-a (Total Data Quality Management) [WBP13]. Also, research questions with respect to data quality in business processes are addressed in this line of research [CFCCP16].

Especially in this "business" research line, data quality is oftentimes **assessed manually**, by data users and other stakeholders, e. g., by means of a survey or questionnaire. For example, AIM Quality, presented by Lee et al. bases on a model of information quality and a questionnaire to measure information quality [LSKW02]. Another example is the metadata-based data quality evaluation of social media data presented by Immonen [IPO15]. Herein, quality attributes are calculated against hand-crafted data quality policies and are added as structured

metadata information. The quality metadata can then further be used to determine the quality of data in data extraction, processing, analysis and in decision-making.

Finally, some approaches are developed to especially **ensure high-quality data at the data source or at the point of data entry**. The use of mobile technologies for example may increase data quality [KK13]. For example, in cases such as data entry of measurement results by people working on a microscope, new data entry methods such as speech recognition may help to ensure high data quality [Sch13].

Various lists for **data quality dimensions and concrete metrics** were proposed [LR95, FLR94, WS96, SSMM12]. For example, in [WS96] *data quality dimensions*, such as believability, ease of understanding and completeness are defined from a consumer's point of view, based on a survey. [FLR94] suggests dimensions, such as consisteny, completeness and timeliness, [LR95] suggests 14 data quality dimensions that are crucial with respect to data model quality, e. g., unambiguous definitions, essentialness, domain precision and semantic consistency.

Plenty research especially focuses on the data quality dimension *accuracy*. Here, variances between the data and the real world, i. e., the truth, are considered. To this end, the correctness of the data values is checked. This data quality dimension is especially crucial, but at the same time also very hard to measure [WS96]. Dong et al. for example investigate the discovery of true values in data integration, where various conflicting data items are coming from multiple sources [DBES09].

Especially for intrinsic indicators of data quality that do not rely on context and subjective characteristics, *concrete metrics* expressed as numbers between 0 and 1 have been the goal of many researchers (e. g., [BM11, SC06, FH07]). Plenty such metrics exist for structured data. For example, information on data quality with respect to duplicates, out-of-domain/out-of-range, null, inconsistent and missing values is usually expressed by means of concrete data quality metrics (cf. Sebastian-Coleman [SC13]). These metrics are also provided within data quality toolkits which are offered as installable software packages as well as on the cloud. An overview on such toolkits is given in Section 2.3.5.

Also, many **frameworks** for data quality have been suggested yet. Most are dedicated to structured data in databases only (e. g., [SC13, WS96]), but also

special frameworks for social media data and Big Data were developed. In [SC13], a framework for measuring data quality is given. Data quality is assessed based on 48 generic measurement types which are based on five dimensions of data quality: completeness, timeliness, validity, consistency, and integrity. Measurement types check for things like the sufficiency of metadata, consistent formatting in one field, reasonable data processing duration and duplicates.

Plenty of research on data quality is on **database-related technical solutions**. This comprises concrete methods as already mentioned above, such as resolving entities, detecting null, out-of-domain and duplicate values in databases [SC13]. Furthermore, also research, e. g., on data provenance [BD10], uncertainty [Jen08, GS+06], data profiling [Nau14], cleansing, [MM00] and on checking data validity, e. g., by means of rules, are performed within this line of research [Hil11]. Uncertainty of data, for example, can be tracked from the structured source data to the query and analysis results in the database presented in [Jen08]. The confidence of single data values can be stored and then later also calculated for complex queries on the structured data. Similar to the *confidence* of statistical classifiers, these confidence values indicate the probability that a certain value is correct. In Griethe et al., methods to visualize such uncertainty in data are reviewed [GS+06]. Finally, holistic methodologies for data curation are elaborated [SBI+13]. Data curation comprises discovering data sources of interest, cleaning and transforming the data, semantically integrating it with other local data sources, and deduplicating the data.

Moreover, research on **data wrangling** focuses on data quality in the context of data analysis. Data Wrangling is, e. g., defined as a *"process of iterative data exploration and transformation that enables analysis"* [KHP+11]. The goal of data wrangling is to make data usable in data analysis, e. g., based on scripting in Python and R [McK12, Boe16], mainly based on data exploration and statistics [DJ03]. Endel et al. summarize various tasks belonging to data wrangling, such as data transformation, cleaning, data quality, merging of different data sources and managing data provenance [EP15]. The data wrangling process is oftentimes described as the most tedious component of the analysis process, some authors even estimate that up to 80% of the development time in analysis projects is on data wrangling [FGL+16, Wic14, DJ03].

A further field with respect to data quality, are research works which investigate the **impact of data quality on analysis results**. For example Pipino et al. define a cost model with respect to the costs associated with a data mining

initiative in terms of the quality of the dataset, and furthermore in terms of the effort spent in preprocessing [PK04]. For example, the percentage of missing and inconsistent data is considered in the cost model. Sessions et al. study the effects of data of varying levels of quality on an algorithm for learning Bayesian networks (the PC algorithm) and generated data sets of various known quality levels [SV06]. Furthermore, in the course of a student project that has accompanied the work on this thesis [Bet17], the relevance and inter-dependencies of data quality and algorithms have been investigated. Herein, a conceptual basis for a systematic investigation of the inter-dependencies between data quality and parameters as well as choice of machine learning algorithms was developed. Furthermore, the concept is evaluated by means of experiments with the NHTSA data set (cf. Section 7.5.1). The concept mainly bases on a *Data Quality Profile* which depicts quality indicators for the dataset and a *Classification Configuration Profile* which depicts the configuration parameters applied to the learning algorithm. These research efforts, with a focus on the choice of machine learning algorithm and parameters, are currently deepened in a parallel PhD project [VZRM18]. Moreover, in the course of a student seminar work that has accompanied the work on this thesis [Lin18], concrete examples for the impact of data quality on analysis quality are given for a structured data set (iris data set[21]) as well as a textual data set (comparative sentences data set [JL06]). To this end, the data sets were manipulated, i. e., data items were deleted or changed to alter the data quality. Then, the accuracy of classifiers was measured for the original as well as for the manipulated data sets. Based on this straightforward experiment, a high impact of data quality on analysis results could be shown for the structured data set. For the textual data set, the effect was less clearly.

With respect to these various research activities in data quality, the work presented in this thesis is classified as follows. The research presented falls in the broad *category* "computational research", since it provides tools and techniques with respect to data quality. *Business aspects* with respect to data quality are not addressed in this thesis. Furthermore, automatic methods which compute data quality or which assist analysts in data quality assessment are addressed rather than a *manual assessment of data quality*. The method for automatic disambiguation of ambiguous words "on write", which was developed in the course of a student project accompanying the work on this thesis [Pan19], addresses the need for data quality methods which *ensure high-quality data at the point of data*

[21] `https://gist.github.com/curran/a08a1080b88344b0c8a7`

entry. As in research on *indicators and metrics*, the research presented in this thesis also aims at developing data quality methods. Moreover, a comprising data quality *framework* is suggested in this thesis. While *database-related solutions* to data quality are generally not applicable to unstructured text data, concepts such as provenance and confidence are relevant with respect to unstructured text data also. As in *data wrangling*, the research presented in this thesis also addresses data quality in the context of analysis pipelines. Finally, the goal of the research presented in this thesis is to improve the quality of analysis results of analysis pipelines, thus this research also considers the *impact of data quality on analysis results*.

2.3.4 Approaches to Data Quality of Unstructured Data and Text

In this section, an overview on existing approaches which explicitly address data quality for unstructured and especially textual data is given. While only very little research explicitly investigates these topics, also other research areas are relevant and related to the concepts and methods presented in this work, e. g., methods from natural language processing, information retrieval, automatic essay assessment, machine learning, speech recognition and image recognition. These will be mentioned and discussed at the respective places in Chapters 4 - 7.

Most data quality management methodologies from the **business** research line, do not consider unstructured data [Sad13]. Yet, for example in the Prediction Markets (PM) assessment technique suggested by Pierce and Thomas, text data quality is **assessed manually** [PT07]. Prediction Markets are applied to the assessment of information quality of newspapers. In a Prediction Market, users need to bet on one of a range of possible outcomes. In the case considered in the PM technique, users were asked to bet on the number of corrections and retractions which will be reported for a certain newspaper in the future. The higher the betted number of corrections and retractions, the lower data quality of the newspapers is assessed using the PM technique.

In [CZ15] an assessment **framework** specially designed for Big Data is suggested. Here, for example a data quality element referring to the difficulty in transforming unstructured data to structured data is proposed. No details on how to measure this data quality element are given, though. In [AHJAJA15] a framework for quality analysis of social media data is presented. It is based on a

list of quality factors including Accessibility, Scalability, Performance and the data quality dimension Accuracy and describes data capturing and data analysis tools for Twitter, Facebook, Linkedin and Flickr data. The authors compare data capturing providers and analysis tools for social media data in terms of advantages and disadvantages such as how much data can be imported and what time frame of the data can be delivered.

Recently, much research moreover considers the **data quality of web data, linked data, Big Data and data lakes**. Most of the data types considered in these works are semi-structured, i. e., they contain unstructured parts such as texts. For example, web pages, linked data and streaming data such as sensor and social media data are investigated in these research works. Farid et al. consider data quality in data lakes by means of integrity constraints on data in data lakes, e. g., represented by rdf triplets [FRI+16]. Plenty research focuses on web, linked and Big Data. For instance, specialized lists of data quality dimensions are suggested [BS16]. Moreover, a data quality module for the Linked Data Integration Framework (LDIF) is proposed [MMB12] and concrete use cases and data quality problems are investigated [McC12].

Further research works in this field mainly consider **data quality dimensions and list conceptual first metrics** for unstructured and textual data. Schaal et al. suggest dimensions specially designed for the web [SSMM12]. They propose the two new dimensions enjoyability and user-conformability and present a list of 42 dimensions. The dimensions are evaluated on the basis of huge web sites, such as Youtube, Wikipedia and Facebook.

With respect to the data quality dimension *accuracy* for textual data, e. g., Jindal and Liu investigate the correctness/truth of written reviews of customers in research on "opinion spam" [JL08]. Opinion spam comprise, e. g., reviews which give undeserving opinions to some product or service in order to promote them or damage their reputations. The approach is based on a logistic regression model.

Several sources [BS16, SIG+12, Son04] address the need for data quality dimensions and metrics on unstructured, especially text data, but none of them gives executable methods. In Batini et al., for example data quality dimensions with respect to the quality of maps, linked open data and images are suggested [BS16]. Also, text data is addressed by Batini et al. Herein, quality with respect to humans reading the text documents is focused and the data quality dimensions Accuracy, Readability, Consistency and Accessibility are suggested. Within these

dimensions moreover conceptual measures are suggested, which consider (1) the closeness of words in a text to a reference vocabulary, (2) readability and text comprehension and (3) cohesion and coherence. All of these measures focus on human data consumers and mainly rely on counting sentences, words, complex words and syllables. At the DBKDA 2012, a panel session on data quality of non-structured data was held [SIG+12]. The published slides emphasize the need for quality metrics for non-structured data and list possible quality criteria for textual data such as the quality of technologies used and the author's expertise. Besides precision and recall, no concrete measurable data quality indicators were provided, though. At the GI Jahrestagung in 2004, Sonntag suggested four categories of data quality dimensions for unstructured natural language text data with focus on human as well as machine consumers of text data [Son04]. Herein, additional data quality dimensions are discussed, namely the dimensions "Contextual", "Intrinsic", "Representational" as well as the "Accessibility" of texts. Within the discussion of these broader dimensions, also concepts with respect to text data quality are mentioned. Especially, Sonntag suggests to consider (1) the reputation of the author of a text, (2) wrongly formulated data values, (3) typing errors, (4) different spellings of the same word and (5) lexical ambiguity.

In summary, in these works, interesting starting points for text quality indicators are defined, such as:

- The quality of technologies used to interpret unstructured data [SIG+12].
- Readability, text comprehension, cohesion, coherence [BS16], the author's expertise [SIG+12] and reputation [Son04].
- Spelling quality, lexical ambiguity [Son04] and the closeness of words in a text to a reference vocabulary [BS16].

While these research works provide first valuable starting points with respect to research on data quality of unstructured and especially textual data, no prototypical implementations are given and evaluations are missing. Finally, no holistic concept towards measuring and assessing unstructured, e. g., textual, data is provided in these existing approaches.

2.3.5 Data Quality Toolkits

In the course of a student project that has accompanied the work on this thesis [Che17], current data quality toolkits were inspected at the hand of criteria, such

as data quality dimensions considered, visualizations available and whether the toolkits are open source or not.

The following toolkits were categorized and studied in more detail: IBM Watson Analytics[22], Data Analyzer by Uniserv[23], Data Quality Dashboard and Reporting by Informatica[24], Data Quality Analysis Package by Salesforce[25], Data Quality Dashboard by Talend[26], DQ Dashboard by Attacama[27], Data Quality Dashboard by InsightSquared[28] and Data Governance Center 4.5 by Collibra[29].

Most importantly, all toolkits were searched for data quality methods for unstructured data. In this study, no data quality toolkit with methods for unstructured text data quality could be found. In the course of the student project, a standalone data quality toolkit was prototypically implemented, which provides methods to assess the quality of textual data also. The data quality methods are implemented as RESTful Web Services to allow for easy and flexible integration into various more toolkits, especially also into analysis toolkits (as described in Section 4.4).

[22] https://www.ibm.com/watson-analytics
[23] https://www.uniserv.com
[24] https://www.informatica.com/de/products/data-quality/informatica-data-quality.html
[25] https://appexchange.salesforce.com/
[26] https://de.talend.com/products/data-quality/data-quality-management/
[27] https://www.ataccama.com/product/
[28] https://www.insightsquared.com/
[29] https://www.collibra.com/data-governance

Chapter 3

Application Scenarios

For the evaluation of the concepts and concrete methods of the suggested data quality framework, three main application scenarios are employed: (1) the analysis of downtimes of a production line (Section 3.1), (2) the detection of safety-related defects in aftersales data (Section 3.2) and (3) two examples for domain-specific data analysis conducted by citizen data scientists (Section 3.3). In the latter scenario, a sample citizen data scientist from industry as well as a citizen data scientist from humanities are considered. Finally, in an initial assessment the quality of data and analysis results in domain-specific analysis pipelines are described based on the latter application scenario. Thus, holistic and continuous data quality assessment and improvement within domain-specific, oftentimes simplified, analysis pipelines is motivated (Section 3.4).

This chapter corresponds to revised and composite versions of excerpts of previous author publications that are cited at affected locations [Kie17, Kie19, KRM19a, KRM19b, KRM20]. All concepts in these publications were developed exclusively by the author of this thesis.

3.1 Application Scenario 1: Analysis of Downtimes of a Production Line

A downtime of a production line may cost millions of euros and may lead to delays and customer dissatisfaction. Thus, downtimes of production lines need to be prevented. If prevention is not possible, however, quick and good handling of downtimes is needed. The second problem is focused in this thesis. To this end, suggestions with respect to the handling of downtimes are given, based on the results of an analysis of structured and unstructured production line data.

In this application scenario, data analysis results may help in terms of reducing the duration of downtimes. To this end, analysis results are prepared for the shop floor workers, who actually handle the downtimes. The results are shown to them, e. g., as recommendation lists of the possibly best actions to handle a downtime. Alternatively, a list of the most probable underlying errors may be presented to the workers. The workers then have to decide on the proper action to handle the respective errors. As information medium, a tablet pc, which may be positioned at each machine in the production line, can be employed.

This application scenario is employed in Chapters 5, 6 and 7 (also see [KRM20, KRM19a]). It is concretized by means of a confidential data collection from an industry partner. This manufacturer, among other products, manufactures small electronic parts. The data comprises structured information on downtimes in a production line, and it moreover contains German free text information. The data set is described in Section 5.3.2.

3.2 Application Scenario 2: Detection of Safety-Related Defects in Aftersales Data

Whereas the first application scenario in Section 3.1 focuses on the production phase in the product life cycle, here, an application scenario is described with respect to data from the aftersales phase (also see [KRM19a]). Manufacturers need to ensure the safety of their products. Leastwise, where safety issues may cost lifes, they are legally bound to recall and exchange unsafe products or product parts. For manufacturers, this may result in a loss of money and reputation, e. g., consider the case of Firestone car tires of Ford Explorer vehicles, which were burst [Haw00]. Thus, manufacturers and national administrations collect information on safety-related issues, e. g., directly from customers or by intermediates, such as a repair shop or a garage. This collected data may be of structured as well as unstructured types, such as values in a database or customer complaint texts. The data may be analyzed by means of data analytics methods as described in Section 2.2. The result of such a data analysis may be presented, e. g., to product managers. For example, current topics such as problems with airbags and car tires may be illustrated by means of a bar chart, which denotes topics and adds information on their frequency. If this information is interpreted on a regular basis, it may help to detect safety issues earlier. Thus,

the manufacturer may react faster, which may safe lives, money and which may further reduce loss in reputation.

In Chapter 7, this application scenario is employed. It is concretized by focusing on the detection of safety-related defect trends in aftersales data from the automotive domain. Moreover, a concrete data collection from the National Highway Traffic Safety Administration (NHTSA) in the United States of America is employed[1], which will be described in more detail in Section 7.5.1.

3.3 Application Scenario 3: Domain-Specific Data Analyses Conducted by Citizen Data Scientists

The suggested concept and methods in this thesis mainly apply to domain-specific data analyses conducted by citizen data scientists. A citizen data scientist is a domain expert, oftentimes without IT or data analysis expertise (cf. Gröger et al. [Grö18]). For profound data analysis, besides IT and data analysis competences, also domain knowledge is essential [GSD07]. Therefore, also non-IT experts without data analysis expertise use analysis tools to build analysis pipelines from scratch. Nowadays, new courses of studies, such as the digital humanities and "computational and data science" try to bring competences from computer science, maths and statistics, data science, and from the respective application domains together [Unib, Unia, Gol12]. Another approach to this issue is to reduce IT and data analytics complexity and to provide simplified analysis tools and toolkits. Thus also IT-inept domain experts are enabled to do data analysis. While analytics need to be kept simple, still, high quality analysis results need to be ensured.

In the following two subsections, two sample citizen data scientists are described. Moreover, examples for data analysis pipelines they would want to build are given. In Section 3.4 and Chapters 4, 5 and 6, these application scenarios are employed in describing the problems that may arise in domain-specific data analyses conducted by citizen data scientists. Moreover, in these later chapters, the positive effect of QUALM on such analysis is illustrated.

[1] `https://www-odi.nhtsa.dot.gov/downloads/`

3.3.1 Citizen Data Scientists from Industry

In the following, a sample use case scenario is described, with a citizen data scientist from industry. The data scientist is a product manager and he wants to get an overview, e. g., on reasons for downtimes on the production line or on top topics and opinions with respect to the product he manages (cf. application scenarios 1 and 2). To this end, he might decide to select the respective data from the company-internal data lake (cf. Giebler et al. [GGH+19]) and to analyze it by himself. For instance, he constructs an analysis pipeline similar to the one depicted in the introduction to this work in Figure 1.1.

For example, the citizen data scientist from industry uses Tika[2] as analysis tool for automatic language identification and to be able to filter for a specific language of the texts (English, German, etc.). Subsequently, parts of speech, such as noun and adjective are annotated. Then, entities, such as persons and products, are recognized. Finally, also topics may be extracted, e. g., based on the implementation of the k-means clustering algorithm, which is available in scikit-learn[3]. Alternatively, in the last processing step, the domain expert might decide to extract opinions, e. g., based on a tool, such as the sentiment detector in NLTK[4]. Finally, analysis results are interpreted by the domain expert.

In Chapters 4, 5 and 6, this application scenario is employed (also see [KRM20, KRM19b]). It is concretized by, e.g., focusing on a concrete data collection from industry, which will be described in more detail in Section 5.3.2.

3.3.2 Citizen Data Scientists from Humanities

In the following, a sample use case scenario with a citizen data scientist from the humanities is described (cf. Kiefer [Kie17]). The data scientist is a linguist and he studies language and language change. He focuses on youth language and how the words used to express, e. g., opinions and emotions differ through time. To this end, he decides to analyze youth language in social networks. Textual data from chats as well as tweets are considered. The linguist employs an easy-to-use analysis toolkit and builds an analysis pipeline from scratch, as illustrated in Figure 3.1.

[2] `https://tika.apache.org/`

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

[4] `https://www.nltk.org/api/nltk.sentiment.html`

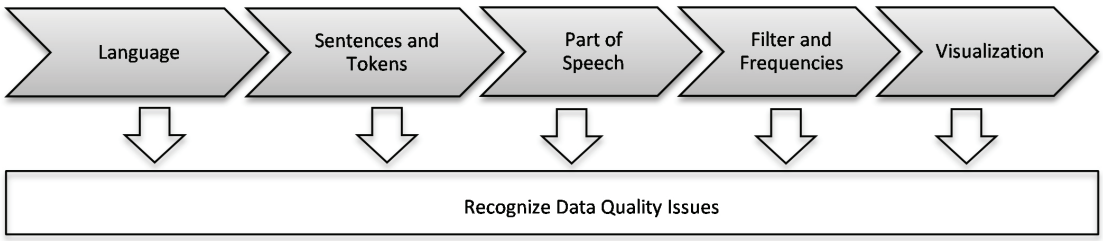


Figure 3.1: Data quality problems may occur in each analysis step of a whole text analysis pipeline.

In the first analysis step, the language is annotated and all English posts are filtered out. All following analysis steps are conducted on these posts. On these filtered posts, subsequently, various preprocessing steps are conducted by the citizen data scientist. For example, sentences are segmented (= sentence segmentation). Then, the sentences are split into the smallest meaning-bearing tokens (=tokenization). Finally, to each of these tokens, the corresponding part of speech is assigned (=POS-tagging). Based on this information, the tokens may be filtered, e. g., for adjectives and nouns only, and a frequency distribution may be calculated. Then, the citizen scientist visualizes and interprets the results.

In the following section, this application scenario is employed. It is concretized by focusing on a concrete data collection of tweets and chat posts, which is described in more detail in Section 5.3.2.

3.4 Initial Assessment

Low data quality is dangerous because it can lead to wrong or missing decisions, strategies and operations. It can slow down innovation processes, and losses for organizations caused by low data quality are immense [DK10]. Bad data is a huge problem: 60% of enterprises suffer from data quality issues, 10-30% of data in organizational databases is inaccurate, and individual reports of incomplete, inaccurate and ambiguous organizational data are numerous [NRC+11, HCW15].

Moreover, lots of crucial information exists, which is encoded in unstructured data [Rus07]. Organizations need to leverage the information hidden in unstructured data to stay competitive [HJ15]. Therefore, high quality of texts, pictures, videos and speech data needs to be ensured. However, while the need

for assessing and improving data quality of unstructured data was recognized (e. g., [SIG+12, BS16], also see Section 2.3.4), no concrete approach to assessing the quality of unstructured data was suggested yet. Similar to the concepts, frameworks and systems developed for structured data, approaches to assess and improve the data quality of unstructured text data are needed.

Many questions in the humanities, natural sciences and in industry can be answered by information that may be extracted from databases and text corpora [FS07]. For example, in the social sciences, biology, linguistics and in the automotive industry, data analysis are employed as the basis for answering questions from these domains. As already described in the introduction of this thesis, all of these domain-specific projects need to face the same challenge: the expert knowledge from IT, text analysis and of the domain, need to be brought together, so that analysts are able to answer emerging questions adequately. One possibility for bringing the needed competences together, is to provide simplified analysis toolkits. These allow domain experts to conduct data and text analyses on their own to answer their questions, e. g., in the humanities or in industry. For example, the Leipzig Corpus Miner (LCM) is a simplified text analysis toolkit. It was developed especially for social scientists without IT background [LW16]. Further text and data analysis toolkits are simplified so that domain experts without IT and text analysis knowledge can use them, e. g., RapidMiner, SPSS and WEKA (cf. Section 2.2.3). Simplifications, for example, comprise the construction of analysis pipelines in graphical user interfaces (GUI). Moreover, in simplified analytics toolkits, defaults are used oftentimes, e. g., especially default training data sets and default settings, e. g., with respect to semantic resources.

However, ease of data analytics should not be at cost of quality. The next section illustrates on the basis of various textual data sets from news data to tweets, chat as well as industry data that simplifications in text analytics may lead to severe quality issues of simplified domain-specific text analytics. The simplifications especially lead to quality problems in the analysis of messy text data, such as tweets or short free texts from industry, which, e. g., contain many spelling mistakes, abbreviations and technical terms.

The observations illustrated in the remaining parts of this section show the need for a comprising concept for data quality assessment and improvement of data quality within these simplified domain-specific data and especially text analyses. Thus, it moreover motivates the QUALM approach, which is presented in the next chapter and for which various aspects in the remaining chapters of

this thesis are detailed. The remaining part of this section corresponds to a revised version of excerpts of a previous author publication [Kie17].

In the following, concrete problems are illustrated that occur especially in the case of messy text data that is analyzed by simplified text analysis toolkits and by simplified standard tools for natural language processing. These tools are simplified and thus rely on defaults, such as default training data. Hence, these tools oftentimes expect clean data, such as newspaper texts and they have problems in handling lower-quality texts, such as tweets, chat posts and text entries fom industry, that come with many spelling mistakes, abbreviations and technical terms. To this end, the quality of labels for such analysis tools, namely language identifier and part-of-speech tagger implementations is shown for various text data sets in Tables 3.1 and 3.2.

In this chapter, possible application scenarios were described. In the last scenario, a linguist without IT and text analytics skills is interested in studying change of youth language over time (cf. Section 3.3.2). To this end, he wants to employ text analysis to textual entries from twitter. In Figure 3.1, the analysis pipeline which might be built by the linguist to answer his domain-specific question, is shown.

In this section, the quality of text analysis results for various data sets, namely, news texts (Penn Treebank[5], cf. [MMS93]), Prose (Brown, cf. [FK79]), Tweets (Twitter corpus, cf. [DRCB13]), Chat posts (NPS chat corpus[6]) and confidential industry data comprising free text entries by workers on a shop floor is reported (cf. Section 5.3.2 for detailed descriptions of the data sets).

In a whole analysis pipeline as, e.g., in the pipeline built by the citizen data scientist in the application scenario, low quality propagates and problems, such as a bad recognition of language and parts of speech, are added together. Thus, for each step in the analysis pipeline (cf. Figure 3.1), the quality of the data needs to be assessed, to prevent low-quality analysis results.

In the following, two of these steps are considered in more detail, namely the (1) automatic identification of the language ("Language" in Figure 3.1) and the (2) automatic annotation of part of speech ("Part of Speech" in Figure 3.1). Both concrete tools are based on supervised machine learning, and thus, employ huge amounts of manually annotated/labeled data.

[5]The freely available Penn Treebank excerpt as provided in NLTK was used.
[6]`http://faculty.nps.edu/cmartell/NPSChat.htm`

In the first preprocessing step, "Language", for each text entry, the correct language shall be recognized automatically. In many domain-specific text analysis pipelines, messy text data need to be processed, which leads to low rates of correct recognitions of the language. This is especially the case in text analysis pipelines that are built by domain-experts, since non-fitting default training data, such as news texts, are oftentimes the basis for the analysis tools applied by domain experts. In Table 3.1, concrete rates of correct recognitions for three such analysis tools for language identification are given, namely:

- Tika[7]
- Language-detector[8]
- LanguageIdentifier.[9]

The analysis tools are further employed in the evaluation in Chapter 5.

The accuracy reached by these analysis tools is shown and various data types are compared. While Apache Tika is a standard tool which was mainly trained on clean data, the language-detector was additionally trained on labeled tweets. The LanguageIdentifier is based on [CT94], here newsgroups were used as training data. The corpora are labeled sentence-wise and the correct label for each sentence is known, i. e., the gold labels for the separation of the corpora into sentences were used.

For clean data, such as news texts and prose, the percentage of correctly labeled sentences is above 0.8 for all three tools tested. For tweets, the portion of correctly recognized sentences is considerably low for the Tika language detector, which expects clean data (0.47). For the language-detector (0.72) and LanguageIdentifier (0.77), which both were also trained on tweets/newsgroups, the accuracy of the tools is higher. For the short texts from industry, the portion of correctly recognized sentences is low for all tools considered: Tika has an accuracy of 0.34, the language-detector results in an accuracy of 0.45 and the accuracy of the LanguageIdentifier is 0.56. In the industry data, many technical terms and abbreviations occur. These might be quantified and employed as data quality indicators.

For chat posts, the accuracy decreases immensely for all three analysis tools, to 0.2-0.33. In the chat posts and tweets, mainly short sentences with a high

[7] `https://tika.apache.org/`
[8] `https://github.com/optimaize/language-detector`
[9] Available from the DKPro Core library: `https://dkpro.github.io/dkpro-core/`

Table 3.1: Accuracy of language identifier analysis tools on various text collections.

Data set	Accuracy			
	Overall	Tika	Language-detector	LanguageIdentifier
News	0.93	0.86	0.96	0.96
Prose	0.88	0.84	0.89	0.91
Tweets	0.65	0.47	0.72	0.77
Industry	0.45	0.34	0.45	0.56
Chat	0.25	0.20	0.22	0.33

degree of noisy data are being misclassified. Among the problematic sentences are, e. g., "ah well", "where did everyone gooo ??", "RT @xochinadoll: I fel so blah today." Thus, measuring the degree of noisy data in these text entries might be a solid indicator for the accuracy that can be expected when analyzing text data. Moreover, also training data of these standard tools that are employed by the linguist and the input data differ tremendously in quality and style. These differences may be detected automatically by means of text similarity metrics, as will be examined in Chapter 6: Low textual similarity of training data and input data (such as news as training data and tweets as input data) may indicate low accuracy values. High textual similarity of training data and input data (such as news as training data and news as input data) might indicate high accuracy values.

As the next exemplary processing step, the automatic tagging of part of speech is considered. In the standard tools for part-of-speech tagging of textual data, usually a pre-defined training data set is employed, oftentimes labeled news texts such as the Penn Treebank. Tools based on such default training data are especially suitable for the analysis of clean data, such as news texts or prose. As with the processing tool for the automatic analysis of the language of text entries, employing such standard tools that rely on default training data may lead to quality issues when applied to messy data, such as tweets or chat posts. In Table 3.2, the problems which may occur in domain-specific text analysis projects are illustrated. To this end, concrete standard tools with default training data sets for POS-tagging are named and their accuracy calculated as the percentage of correctly labeled part of speech for various data sets is stated[10].

[10]The manually labeled sentence and token borders were employed.

Table 3.2: Accuracy of POS-tagger tools on various text collections.

Data set	Accuracy			
	Overall	CRF	Perceptron	TNT
News	0.90	0.94	0.90	0.85
Prose	0.70	0.79	0.82	0.64
Tweets	0.68	0.67	0.83	0.55
Chat	0.67	0.68	0.87	0.54

Three analysis tools are investigated:

- CRF POS-tagger[11]
- Perceptron tagger[12]
- TNT tagger.[13]

The POS-taggers base on the default Penn Treebank tagset and Treebank training data [MMS93]. The analysis tools will further be employed in the evaluations in Chapters 5 and 6. Note that the industry data set, which was listed in Table 3.1, cannot be added here, since, as it is usually the case for domain-specific data sets, no gold labels are available in the data set and, thus, no accuracies are calculable.

In comparison to the accuracies reached by the language identifiers, especially for messy data sets, such as chat posts and tweets, the accuracies of POS-taggers are higher. The POS-tagger analysis tools seem to be more robust with respect to messy text data sets, such as chat posts and tweets, than the language identifiers. Still, as for the language identifiers, the quality of POS-taggers also decreases for messy text data, when compared to clean texts such as news texts.

As shown in Table 3.2, for clean data, i.e., news texts, the standard tools work properly. For messy text data sets, such as chat posts and tweets, all tools have lower accuracies.

[11] `http://www.nltk.org/_modules/nltk/tag/crf.html`
[12] `http://www.nltk.org/_modules/nltk/tag/perceptron.html`
[13] `https://www.nltk.org/_modules/nltk/tag/tnt.html`

3.5 Summary

To enable the fusion of IT, text analysis and domain knowledge, simplified analysis toolkits with standard tools for natural language processing may be employed. Simplifications of text analysis tools, however, should not lead to low-quality analysis results. As shown in the previous section for two exemplary preprocessing steps in text analysis, problems may occur at any step in a text analysis pipeline. These are, e. g., due to simplifications and missing adaptations of the analysis tools by users without IT and text analysis expertise. For example, simplifications, such as standard tools which employ default training data, may lead to low accuracies if messy text data, such as tweets are being analyzed. Moreover, low quality propagates within the analysis pipeline and the problems are added together, so that finally the answers to the domain expert's questions may be distorted. In this thesis a concept for high-*qual*ity *m*ining (QUALM) is presented, which enables the detection of quality issues by measuring text data quality (1) before executing the analysis pipeline and (2) without need for gold labeled input data, which is most often not available in domain-specific projects (cf. Chapter 4).

Chapter 4

QUALity Mining (QUALM): Concept and Implementation

In this chapter, the first challenge is addressed, the continuous and holistic data quality measurement and improvement within data analysis pipelines (cf. Section 1.2.1). To this end, the QUALM concept for high **qual**ity **m**ining of data within analysis pipelines is presented. The goal of QUALM is to increase the quality of analysis results, e. g., with respect to the accuracy of a text classification. This goal is reached on the basis of a measurement and improvement of data quality for each step in the analysis pipeline. Moreover, data as well as specifics of analysis tools, such as training data, semantic resources and features are considered by QUALM. In this section, the QUALM concept is illustrated in detail, in the subsequent Chapter 5, the concept is further concretized by means of a description of QUALM data quality methods. The QUALM concept and methods are evaluated in Section 5.3.

Textual data, such as reports, tweets and customer complaints contain highly relevant information which is extracted by means of text analysis pipelines (cf. Kassner and Mitschang [KM16]). For the extraction of information from textual data, various analysis tools for natural language processing are applied within the text analysis pipeline. Figure 4.1 shows exemplary typical steps of text analysis pipelines and the therein applied analysis tools for processing free texts from industry. For example, Tika[1] is used as analysis tool for automatic language identification and to be able to filter for a specific language of the texts (English, German, etc.). Subsequently, parts of speech, such as Noun and Adjective, are added. Then, entities, such as persons and locations, are recognized and finally

[1] `https://tika.apache.org/`

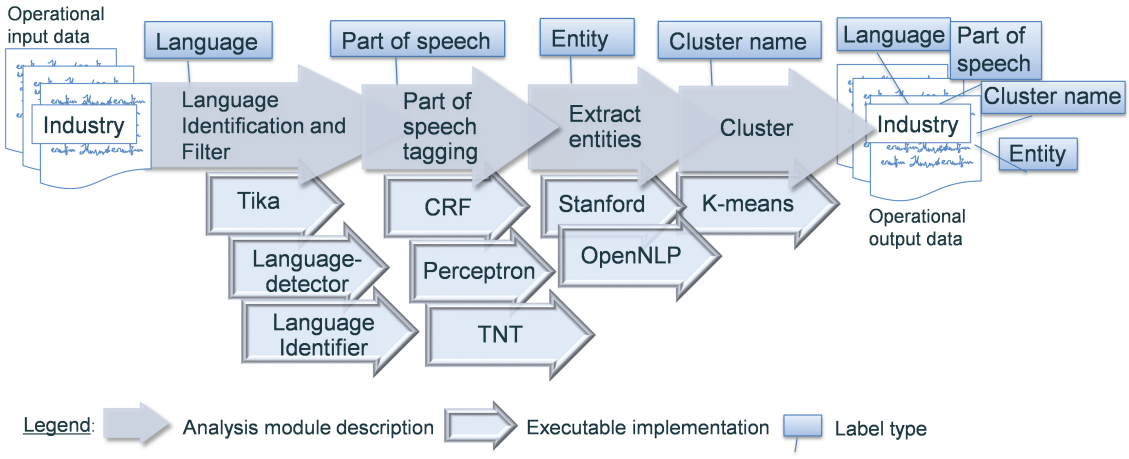


Figure 4.1: Simplified illustration of a text analysis pipeline with examples for analysis tools/implementations, such as Tika, CRF, Stanford NER and OpenNLP NER, mainly repeated from Figure 1.1.

topics are extracted, based on a cluster analysis, which, e. g., is based on the k-means implementation available in scikit-learn[2].

Decisions are made oftentimes based on such text analysis pipelines. For example, on processes in factories [GKH+16] or on the design of marketing campaigns [MV18]. A further typical application field of text analysis is the semi- or fully automatic enrichment of knowledge databases [GPM04]. To each of these, high quality results for all analysis tools employed in the analysis pipeline are crucial, since otherwise wrong decisions are being made.

The goal of the QUALM approach suggested in this chapter is to increase data quality and to thereby increase the quality of analysis results as well. This result quality is measurable, e. g., by means of evaluation metrics such as accuracy. Moreover, the approach considers all steps/analysis tools applied within an analysis pipeline.

In the course of a student project that has accompanied the work on this thesis [Bet17], the relevance and inter-dependencies of the two adjusting screws in data analysis: data quality and optimization of algorithms, has been investigated. Data quality and not an advancement of analytic algorithms or parameters is focused. These are addressed in a parallel PhD project [VZRM18].

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

In the literature, many evaluation metrics are described, which may be employed in measuring the quality of analysis results (cf. Hossin and Sulaiman [HS15]). Yet, to be able to measure the quality of data analysis based on such metrics, gold labeled data need to be available. The generation of gold labels, however, is costly, since every single data set needs to be manually enriched with the correct labels (cf. Marcus et al. [MMS93]). For single specific domains many such labeled text corpora are available by now (e. g., on kaggle[3]). Yet, for many other application domains and respective text analysis, especially for preprocessing tools in text analysis, no gold labeled data is available. A further disadvantage of evaluation metrics is that they do not give direct starting points for quality improvement. They just reflect the number of correct tagging decisions made by the analysis tool.

Data quality methods for measuring and improving the quality of textual data, can hardly be found and, if so, for restricted domains only (cf. Section 2.3.4). Yet, these data quality methods do not consider the specifics of the analysis tools within the analysis pipeline. In contrast to QUALM, related approaches do only consider single characteristics of the data or the influence of semantic resources. For example, spelling mistakes or single analysis modules, such as the language identifier, are investigated. All of these approaches miss to suggest a holistic concept for measuring and improving the quality of textual data within analysis pipelines.

Many definitions of data quality exist (cf. Sebastian-Coleman [SC13]). The concepts presented in this thesis base on the definition of Wang and Strong, in which data quality is defined as the data's **„degree of fitness for use by data consumers“** [WS96]. Beside human end consumers of data, in a text analysis pipeline, the employed analysis tools are also data consumers (cf. Section 2.3.2). A generally valid formal definition of this concept of the *degree of fitness for use by data consumers* can not be given. Yet, this is possible for single specifics of analysis tools as well as for single data characteristics and is captured in QUALM by the QUALM indicators. For example, an indicator in QUALM defines the degree of fitness for use for input data and data consumer, based on the text similarity of input and training data. Hence, in QUALM, not only the data but also specifics of the analysis tools, such as **training data and semantic resources** as well as information on the **features** employed by the analysis tools, are respected:

[3] `https://www.kaggle.com/datasets`

- The concrete language identifiers, part-of-speech taggers and named entity recognizers in Figure 4.1, for example, use different default **training data**. While Apache Tika was mainly trained on clean data (news texts), the language-detector was additionally trained on tweets. The LanguageIdentifier was trained on newsgroups [CT94].
- Many tools for natural language processing use **semantic resources**, such as the lexical database WordNet[4] or the German pendant GermaNet[5]. Moreover, many domain-specific semantic resources exist, e. g., in the automotive sector [ST10] or on movie reviews [HL04].
- Moreover, analysis tools may give higher weights to different features of data sets. Language identifiers may focus on word length and upper and lowercasing. Part-of-speech taggers may emphasize the antecedent and subsequent token of the currently classified word (such as the antecedent word or a sentence marker). Named Entity recognizers again may give high significance to special signal words (e. g., titles, such as Mr.).

To be able to respect all of these elements in analysis pipelines when measuring and improving data quality, the holistic QUALM concept is presented. It is applied to all analysis tools within the text analysis pipeline and, thus, measures the data quality for each step in the whole analysis pipeline. The concept is based on repositories for capturing the specifics of analysis tools as well as on concrete QUALM data quality methods. The latter comprise the indicators as well as corresponding modifiers for data quality improvement. The QUALM indicators reflect the raw measured values for characteristics of data and of specifics of analysis tools. For example, the average sentence length, the percentage of misspelled words and text similarity are QUALM indicators. These moreover shall be able to predict the quality of analysis results. For instance, they should be able to predict the accuracy of the annotations of analysis tools for certain data sets and thus, e. g., need to correlate with accuracy. In QUALM, each indicator has a corresponding modifier. Modifiers change the data and the specifics of analysis tools. For example, the percentage of abbreviations may be determined (indicator). Then, a corresponding method which can resolve abbreviations by means of a semantic resource may be suggested (modifier).

[4] `https://wordnet.princeton.edu/`
[5] `http://www.sfs.uni-tuebingen.de/GermaNet/`

With the QUALM data quality methods, three disadvantages of existing approaches are addressed: (1) In difference to evaluation metrics, QUALM indicators do not depend on gold labels, which are added manually and are costly. (2) Based on the mapping of indicators to modifiers, direct starting points for data quality improvement are given. (3) In difference to existing approaches, they comprise not only the data but, moreover, also consider the specifics of single analysis tools in text analysis pipelines.

In this chapter, the concept and a list of sample QUALM data quality methods is presented. Finally, a prototypical implementation of the QUALM concept is discussed. In the next Chapter 5, a detailed description of the concrete QUALM methods is given and evaluation results for concrete analysis tools and data sets are discussed. The experiments, for example, show an increased accuracy of language identifiers and part-of-speech taggers with QUALM in comparison to an analysis of the data without application of QUALM.

In the next section, related work is discussed. Then, the QUALM concept is presented, examples for QUALM data quality methods are given and QUALM repositories are discussed in Section 4.2. In Section 4.3, examples for the application of QUALM with respect to analysis tools and whole analysis pipelines are given. Then, design decisions in a prototypical implementation are discussed in Section 4.4. This chapter is concluded in Section 4.5 with a summary and outlook. Note that evaluation results for single QUALM methods as well as the effect of QUALM on a whole pipeline are discussed in Section 5.3. This chapter is a revised version of a previous author publication [KRM19b]. All concepts in this publication were developed exclusively by the author of this thesis.

4.1 Related Work

Already in Section 2.3.4, approaches to data quality of unstructured data and text were considered. In this section, related work and a discussion of their similarities as well as differences in comparison to the suggested QUALM concept are presented. To this end, discussions of research works are summarized, which are relevant with respect to the development of concrete methods. Next, works that consider single data characteristics as well as semantic resources and their influence on the quality of analysis results are mentioned. Finally, existing

approaches, that, as in QUALM, assess data quality not only for one step but all steps within the analysis pipeline are presented.

Many research works suggest single, mainly domain-specific methods for a measurement and improvement of unstructured and text data quality. These are discussed in Section 5.1. For example, automatic assessment methods for student essays and readability indices are suggested therein [MK00].

Yet, these research works are not integrated in an overall concept. In contrast, in this chapter a comprising concept is suggested. Within this concept, single methods, such as on abbreviations and spelling mistakes, may be applied. In QUALM, furthermore, new additional quality methods exist which moreover consider the fit of data and specifics of analysis tools (cf. Section 4.2.2).

Moreover, related works investigate the influence of single data characteristics and semantic resources on analysis result quality. For example, the effect of text size on the accuracy of language identifiers or the dependencies between data quality and search result quality are considered in these works [BB12, Fei13] (also see Section 5.1).

Still, all of these works only investigate the influence of single data characteristics or of semantic resources on the accuracy of single analysis tools. In these works, no comprising concept is suggested, in which several relevant data characteristics and all specifics of analysis tools in analysis processes/pipelines may be considered in quality measurement and improvement.

Finally, research works that consider data quality in multiple steps of the analysis pipeline, are related to the QUALM approach suggested in this chapter. In the QUALM approach, the relevance of a measurement of data quality for each step in the analysis pipeline is emphasized - from data creation or data source till usage and maybe even re-usage. Also, Immonen presents an approach for the stepwise measurement of quality [IPO15]. Todoran describes, how local data quality of the single modules which make up an information system, contribute to the global information quality of the whole system [TLKL15]. Ranjit and Kawaljeet [RK10] summarize the reasons for data quality problems in data warehousing. In this context, they address data quality on different levels of the whole process, e. g., the raw data level, data integration and staging. Also, concepts and research works on data provenance aim at collecting and exploiting information on all sources and transformation steps of data [HDB17, BD10, BEP15]. Data wrangling is described as an iterative process of data exploration and data

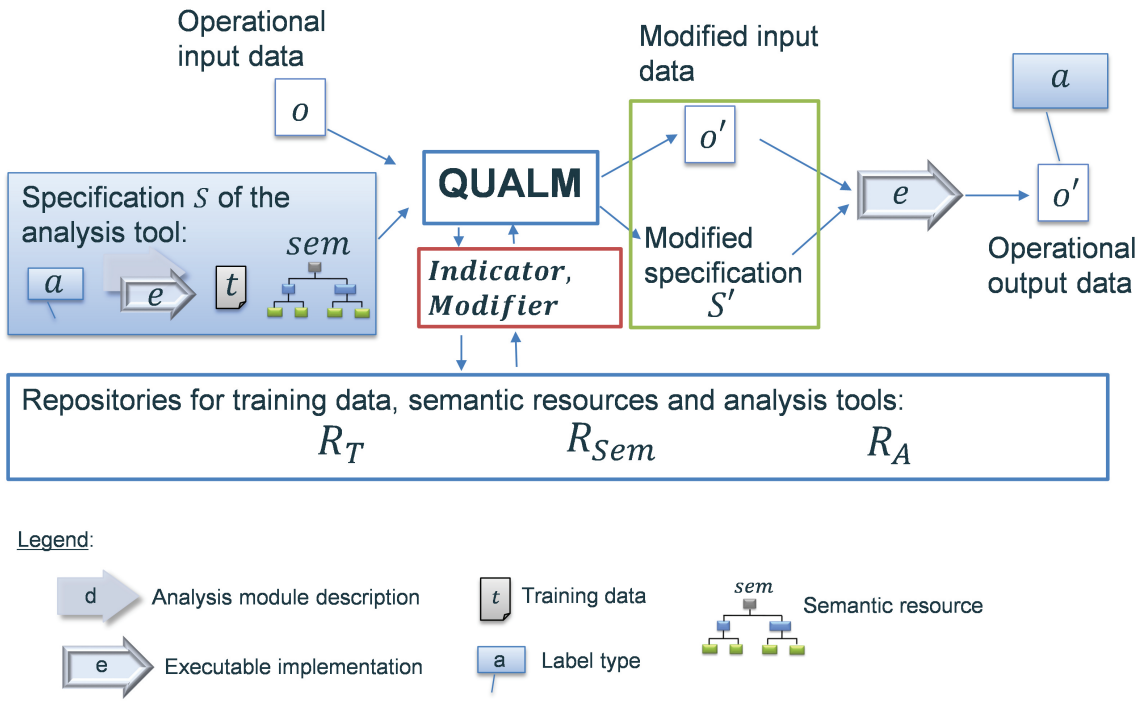


Figure 4.2: Formalized illustration of the application of the QUALM concept on one analysis tool.

transformations which aim at bringing the data to a format and condition which allows for analyzing it [KHP+11]. In works on data wrangling, data quality is discussed with respect to a subsequent data analysis [EP15].

Yet, textual data are not addressed in all of these research works. In difference to QUALM, also no targeted examination of the specifics of analysis tools is executed. Thus, moreover no targeted and tailored quality improvement for each step is possible in these approaches.

4.2 The QUALM Concept

In the subsequent sections, the fundamental ideas of QUALM are described (Section 4.2.1) and concrete QUALM data quality methods are outlined (Section 4.2.2). Then, the repositories for training data, semantic resources and analysis tools, are described which are needed as basis for QUALM (Sections 4.2.3 and 4.2.4).

4.2.1 Basic Ideas of QUALM

In Figure 4.2, the application of QUALM on one analysis tool is illustrated. The input data o as well as the specifications of the analysis tool S are being used in QUALM. The specification S comprises information on the annotation/label type a, such as *language* or *part of speech*. Furthermore, in S, information on the analysis module d and concrete tool e, the training data t and the semantic resources *sem* are contained. For each specific analysis tool, moreover, further information may be retrieved from R_A that is a repository of QUALM. For example, in R_A information on especially relevant indicators as well as on possibly pre-defined and thus available thresholds for QUALM indicators may be requested, which are specific for each analysis tool (cf. Section 4.2.4).

The QUALM indicators quantify data characteristics. The QUALM modifiers are concrete methods for the modification of data and/or specifications of analysis tools. To each indicator, a modifier exists as corresponding counterpart. As basis for the QUALM data quality methods, the repositories R_T, R_{Sem} and R_A for training data, semantic resources and analysis tools are used.

The selection and application of fitting indicators and modifiers is executed in three steps. Firstly, all applicable indicators for the given input data and the given specification of the analysis tool are calculated. Secondly, those indicators are selected, whose values are beyond a pre-defined threshold (for the thresholds, see Section 4.2.4). In the last step, the counterpart modifiers to the selected indicators are applied to the data and/or specifics of the analysis tools. If applicable, the modified operational input data o' as well as the changed specification of the analysis tool S' are used by the analysis tool e to output the data o', now enriched with annotations/labels of type a.

4.2.2 QUALM Data Quality Methods

An overview on all QUALM data quality methods is given in Table 5.1 in Chapter 5. In Chapter 5, a detailed overview over all QUALM data quality methods and their implementations are presented. Moreover, results for the evaluation with different data sets are presented, reaching from news texts over tweets to free texts from industry. Especially, the evaluation investigates whether the measured values for indicators may estimate the accuracy of part-of-speech taggers as well as language identifiers or not. Many of the investigated

indicators correlate with the accuracy of the analysis tools, e. g., the percentage of uppercased words, the percentage of abbreviations and the fit of training data and input data and thus may be applied in QUALM as concrete methods for measuring data quality (cf. Section 5.3).

The list of developed and implemented **QUALM indicators** comprises many straight-forward implementations and also several complex indicators, which, e. g., are based on machine learning. At the heart, the implementations base on existing libraries for natural language processing, such as NLTK[6] and libraries of Stanford[7]. The implementations additionally are available as RESTful webservices, so that an easy integration into existing analytics toolkits is ensured. The methods are exemplary integrated into the freely available data processing toolkit FlexMash (Hirmer and Behringer [HB16]). The methods may also be flexibly integrated into further data analysis toolkits, such as SPSS, Weka or RapidMiner (cf. Section 2.2.3).

Two types of indicators exist. The first group consists of indicators which are only applicable to the data. The second group comprises those indicators which consider the fit of data and analysis tools. For example, QUALM indicators measure the percentage of abbreviation and spelling mistakes and determine the confidence of a standard tool for natural language processing or the percentage of unknown words.

The **QUALM modifiers** represent the corresponding counterparts to QUALM indicators. They dissolve abbreviations, correct spelling mistakes, decrease lexical diversity by merging synonymous terms to one term or employ texts with correctly upper and lowercased words. The implementations of the modifiers in part strongly resemble the implementations of the indicators (cf. Chapter 5 for more details).

In the following, an overview description of two central data quality methods in QUALM is given, more details will be given in subsequent chapters (cf. Chapters 5 and 6): (1) the measurement of the *degree of the fit of input data and training data* as well as the compilation of a ranking of the best-fitting training data sets from the repository R_T and (2) the measurement of the *degree of the fit of input data and available semantic resources* as well as the generation of reasonable suggestions of well-fitting semantic resources from the repository

[6] `https://www.nltk.org`

[7] `https://stanfordnlp.github.io/CoreNLP/`,`https://nlp.stanford.edu/software/CRF-NER.shtml`

R_{Sem}. The central idea is to measure the similarity between the input data on the one hand and the training data sets available from R_T as well as the semantic resources available from R_{Sem} on the other hand. Then, this similarity-based quality indicator may be used as decisive criterion in the selection of better-fitting training data and semantic resources.

The *fitness for use with respect to training data* can be determined for analysis tools which base on supervised machine learning techniques. The indicator bases on text similarity metrics, such as the *Latent Semantic Analysis* (Landauer et al. [LFL98]) and *Cosine Similarity* (cf. Bär [DTI13] for an overview on text similarity metrics). The latter is often used in information retrieval. To this end, all words are listed in a vector and they are weighted according to term frequency (tf) or based on the product of term frequency and inverse document frequency (tf-idf) (cf. Section 2.2.2). Then, the cosine between the text vectors is calculated (also see [MRS08]). With these text similarity metrics, the degree of textual similarity between the vectorized input text and the vectorized training data may be output as a number in the interval [0,1]. A value of 0 means that both texts have no similarities at all, a value of 1 means that both texts are identical.

The result of a text similarity metric is thus a number in [0,1], which may be employed as a QUALM data quality indicator. Moreover, the metric serves as basis for a ranked list of the recommended training data and semantic resources from the repositories R_T and R_{Sem}. In QUALM, either the first entry from the list is automatically selected and applied, or the analyst chooses manually, e. g., the training data set which shall be applied, from the list presented in QUALM.

To determine the degree of fitness for use of operational input data and available semantic resources, very similar metrics which base on the relevance metrics from information retrieval, may be applied (cf. Chapter 5). With limitation to ontologies only, already many such metrics were suggested [TA07, YLCC07]. Yet, approaches that can be applied over the range of all different types of semantic resources usually available, are still missing and are addressed in Chapter 5.

In this chapter, the presentation of the overall QUALM concept is focused, in subsequent chapters details on single data quality methods will be given (cf. Table 5.1 for an overview).

4.2.3 Repositories for Training Data, Mining Models and Semantic Resources

In the concept suggested, new or better fitting training data and semantic resources may be suggested automatically. These suggestions are based on repositories for training data and semantic resources. In this section, the conceptual description of these repositories is given.

The **training data repository** R_T contains training data which were manually enriched with labels of various label types. For example, label types, such as *language*, *part of speech* and *named entity* exist. A certain supervised analysis tool needs training data with annotated labels of a corresponding label type. For example, a sentiment analysis tool needs training data which was enriched with labels of type *opinion*. Thus, the label types need to be stored in the metadata. Several different training data sets may have labels of the same type and may be stored as alternative options in the repository. In a concrete repository, training data sets may be ordered and stored per label type, thus, enabling an efficient lookup.

Fast processing could, furthermore, be ensured by linking pre-trained mining models to the training data sets. These may further be stored in a **mining model repository**. This repository would need to contain the training data sets, information on the label types and available mining models. A similar repository was already suggested in a patent by Sas Institute Inc[8] and may serve as a more detailed conceptual basis.

As basis to QUALM, furthermore, all semantic resources, such as general dictionaries, abbreviation dictionaries, taxonomies and ontologies, need to be stored at a central **repository for semantic resources** R_{Sem}. While they may be stored as a simple set, all resources need to be in a uniform format. Also, pre-trained domain-specific neural networks, which contain universal as well as domain-specific language representations (cf., e. g., Devlin et al. [DCLT18]), may be stored at this central place. This repository is independent from label type and the concrete analysis tool used. Thus, the repository for semantic resources R_{Sem} in the QUALM concept makes up a uniformly formatted simple set of semantic resources (cf. Section 5.2.2 for the QUALM indicator 13 'fit of semantic resources' and the corresponding QUALM modifier which base on this repository).

[8] `https://patents.google.com/patent/US6920458B1/en`

4.2.4 Repository for Analysis Tools

In a **repository for analysis tools** R_A, for each analysis tool, additionally it may be determined, which indicators are especially relevant to that tool. This might be the case, since corresponding data characteristics, i. e., certain features of those analysis tools are being weighted very high. Having this additional information is especially useful for concrete analysis tools, which are provided within an analysis toolkit, such as RapidMiner. Also, in the repository R_A, for each analysis tool, the thresholds for single indicators are captured. For the prototype, firstly default threshold values are defined based on the evaluation results presented in Chapter 5. Examples for threshold values are given in Section 4.3.1. In general, within the QUALM concept, the definition of default thresholds is done manually by experts, such as data scientists, or by developers of single analysis tools. Thereby, the knowledge of these experts is collected in the repository. This new information may then be offered and is available to domain experts as well and it can guide them in building a useful and high-quality analysis pipeline. In future, an automatic adaptation of threshold values, which works analog to approaches in the new research field on Auto-ML, may be investigated. In Auto-ML, for example, the best-fitting algorithm and the optimal parameters for a certain given data set may be determined automatically [FKE+15]. Also, for all analysis tools available in a company-internal *pool of available analysis tools*, as compiled and determined by experts, such as data scientists, information on relevant indicators may be added by the experts.

A repository for analysis tools R_A is formalized as basis for the QUALM concept, as a set of tuples. These represent information as 4-tupel existing of label type, analysis tool, relevant indicators and the thresholds. The label type is additionally stored, so that the training data with the fit label type for a certain analysis tool, may be pre-selected. In QUALM, the applicable QUALM data quality indicators may be selected from R_A via the label type and the concrete analysis tool. These may further be highlighted or may get higher weight. As an alternative, the whole list of QUALM data quality indicators (cf. Section 4.2.2) may be calculated as well.

4.3 Examples for the Application of QUALM

In this section, the application of QUALM on a single analysis tool (Section 4.3.1) and a whole analysis pipeline is illustrated (Section 4.3.2).

4.3.1 Application of QUALM on a Single Analysis Tool

In Section 4.2.1, the formalized application of QUALM was illustrated. In Figure 4.3, the application of QUALM is explained with a concrete example. Here, the operational data are tweets taken from the NLTK data collection[9] (cf. the application scenario described in Section 3.3.1). The analyst wants to get a realistic picture of the opinions in the texts. The applied analysis tool is a supervised classifier from NLTK, which adds labels of the type *opinion*[10]. Three concrete labels are possible: positive, negative and neutral. Without the application of QUALM, the analysis tool is based on reviews as training data[11]. As a default, it initially does not exploit semantic resources.

For the application of QUALM on the example, in the first step, all applicable indicators are selected and calculated. In the example, for the following indicators, which are applied to the data, a value beyond the thresholds defined in R_A is measured. The percentage of spelling mistakes is at 27% (threshold 20%), the percentage of uppercased words is 7% (threshold 2%) and thus comparably high, and the tweets contain many abbreviations (4.6% with a threshold of 2%) (cf. Section 5.3.3). The fit of input data and training data is with a cosine similarity of 0.08 very bad. For the fit of input data and semantic resources a value of 0 is determined, since in the example scenario initially no resource is employed.

On the basis of these indicators, corresponding counterpart modifiers are suggested. In the example, the data are modified by, e. g., correcting spelling mistakes or by dissolving abbreviations. Moreover, the QUALM modifier for an automatic selection of better fitting training data suggests to employ gold labeled tweets as training data instead of the default reviews[12]. In the concrete example, this leads to an increase of accuracy from 0.63 with reviews as training

[9] `http://www.nltk.org/nltk_data/`
[10] `https://www.nltk.org/api/nltk.sentiment.html`
[11] `https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset`
[12] The tweets from the NLTK data collection were splitted into disjunct training and test data sets for this example.

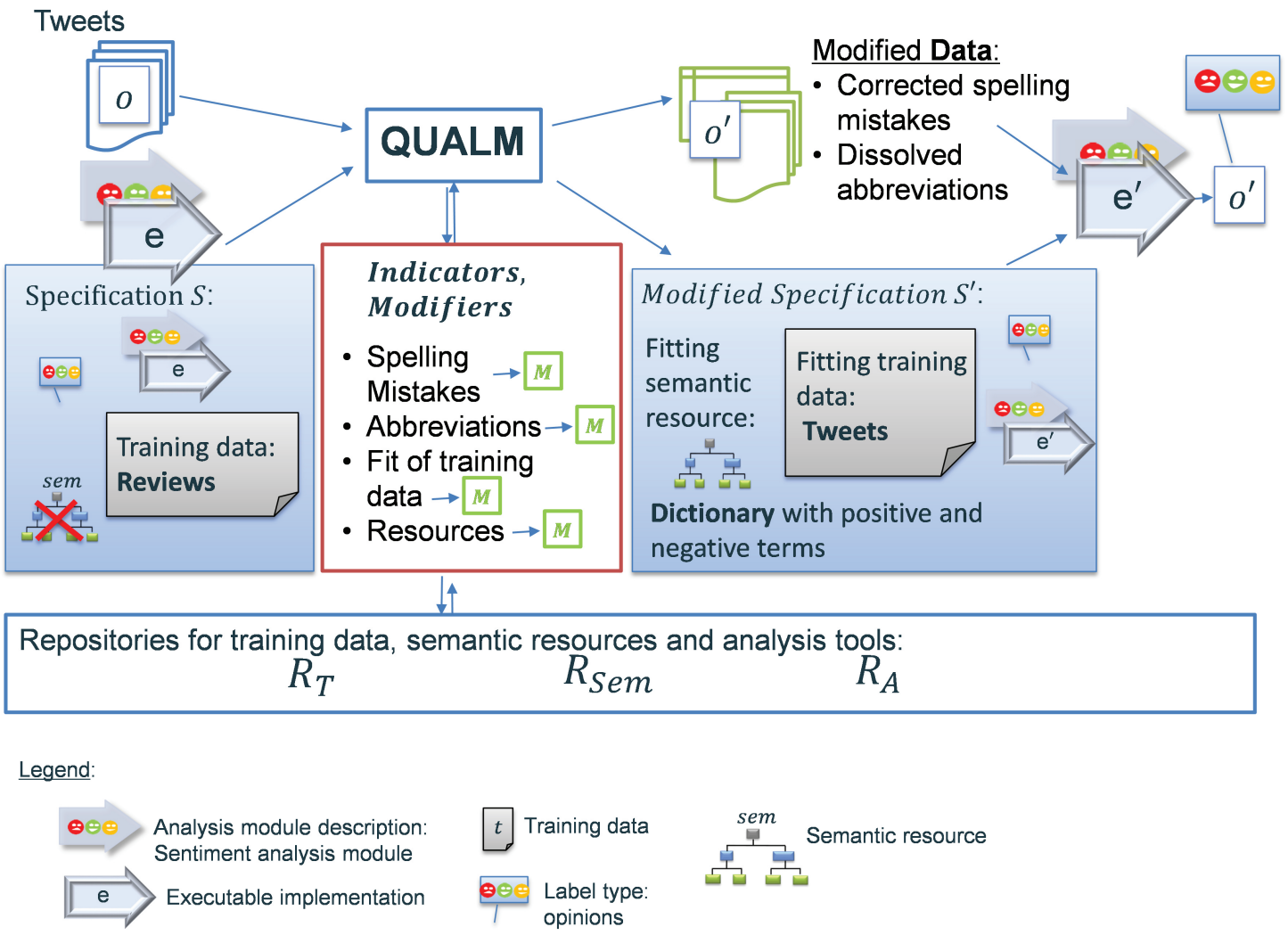


Figure 4.3: Example for the application of QUALM on tweets.

data (=without QUALM) to 0.95 with the labeled tweets as training data as suggested by QUALM (= with QUALM).

Moreover, as result of the application of QUALM, a list of the best-fitting resources in descending order is presented to the analyst. The list is based on cosine similarity between the resources in R_{Sem} and the input data. In the example scenario, thereupon an additional semantic resource is employed within analysis: A dictionary with positive and negative terms[13]. The exploitation of this semantic resource was suggested to the analyst by QUALM. It was selected by QUALM from a list of 15 non-fitting, mid and well-fitting resources automatically and increases the accuracy (cf. Formula 2.3) in addition, for a concrete sentiment analysis pipeline from kaggle[14], by 0.01 (or equivalently by 1%-point if accuracy is stated in percent). While this is only a small rise, an accuracy rise of 0.01 is still relevant in the context of sentiment analysis classification (cf. [DLY08, Liu15]). Detailed evaluation results over several data sets are described in Section 5.3.

4.3.2 Application of QUALM on a Whole Analysis Process

The measurement and improvement of data quality on the basis of the QUALM concept can take place for each analysis tool within an analysis pipeline separately. Within the application of the concept to a whole analysis pipeline, a further strength of quality measurement for each step is revealed. For one analysis tool, certain data characteristics might be helpful, such as transferring all texts to lowercased lettering. For a subsequent analysis tool, which as input data receives the output data of the precedent analysis tool, and thus the lowercased texts, yet a correct upper and lowercasing of the texts may be crucial. This means that both analysis tools regard different data characteristics as good or bad data quality. Therefore, a respective modification of the data could increase the quality of the analysis result of the first analysis tool, but at the same time decreases result quality for the second tool.

With QUALM, this problem is indicated to the analyst, since the quality is measured for each step in the analysis pipeline. On the basis of this information, the analyst can exchange one of the analysis tools, if this is possible, and he may

[13] https://github.com/felipebravom/StaticTwitterSent/tree/master/extra/Sentiment140-Lexicon-v0.1

[14] https://www.kaggle.com/paoloripamonti/twitter-sentiment-analysis

use a less sensible analysis tool instead, which has no quality issues on data with certain characteristics, such as wrong upper and lowercased words. Tika, for example, is a robust language identifier with respect to these characteristics (also see Section 5.3.3). Also, the analyst may decide to employ a modifier suggested by QUALM, which, e. g., transforms the texts back to the original upper and lowercasing. Hence, for each analysis tool and thus for each analysis step, tailored modifiers may be selected.

4.4 Prototypical Implementation

In the course of two student projects that accompanied the work on this thesis, it was investigated how QUALM data quality methods may be provided in Jupyter Notebooks [Lau17] as well as in a web-based standalone data quality toolkit [Che17]. Besides these possibilities of providing single QUALM methods, for a prototypical implementation of the whole QUALM concept, the methods need to be integrated into an analytics toolkit. In difference to existing approaches (cf. Section 2.3.4), the QUALM concept addresses the quality of data, especially also text data, within analysis pipelines. In each analysis step, data as well as analysis tools and their resources, such as dictionaries and training data are examined. Thus, QUALM is not implemented as a standalone data quality toolkit. Rather, an integration of QUALM into existing data analysis toolkits yields maximum value for the data analyst. Most existing data quality toolkits are standalone in that they do not consider the analysis pipeline in measuring and improving data quality (cf. Section 2.3.5). Additionally, existing analysis toolkits do not provide holistic methods with respect to data quality (cf. Section 2.2.3).

For the integration of QUALM into analysis toolkits, FlexMash[15] [HB16] was selected as a sample toolkit. FlexMash allows an easy and fast building of data processing and analysis pipelines from scratch. It especially also assists domain experts who want to build analysis and data processing pipelines in a graphical user interface. Moreover, since it is open source, it is a good choice for exemplifying the integration of the QUALM concept into an analysis toolkit.

First of all, the QUALM repositories for training data, analysis tools and semantic resources need to be provided as basis. For evaluation purposes and an easy handling in evaluation, these repositories described in Sections 4.2.3

[15] `https://github.com/hirmerpl/FlexMash`

and 4.2.4 are represented by folder structures in the file system. In future, NoSQL databases, such as MongoDB[16], may replace these. Also, concepts such as data lakes may be employed in future work. These aim at storing all types of data as well as knowledge resources, such as dictionaries and taxonomies, analysis results and analysis intermediate results at the same place (cf. Section 2.2.1).

The QUALM concept has been prototypically implemented and integrated into the processing toolkit FlexMash in the course of a student project that has accompanied the work on this thesis [Gol19]. Furthermore, concrete data quality methods were prototypically implemented in foregoing research work (cf. Section 5). The concrete lists of QUALM data quality methods, i. e., of indicators as well as modifiers, may be developed in various programming languages, such as Python and Java. Yet, all methods can be transformed into web services. In the prototypical implementation, RESTful web services are employed (cf. Richardson and Ruby [RR08]).

Several variants for the integration of QUALM into an analysis toolkit such as FlexMash are possible. Especially with respect to the communication and interaction with the user these concepts differ. In [Gol19], a first concept as well as a corresponding architecture and prototypical implementation are given. The concept suggested herein, aims at giving users all relevant information and it further lets the user decide on detailed settings, such as the training data sets finally selected. Further possibilities with respect to the interaction with the user will be investigated in future work.

The prototypical implementation in [Gol19] is based on the existing FlexMash system and two new Java projects. All data is exchanged in JSON format. The first Java project is the 'offline project', which, based on OpenNLP[17], implements new functionalities with respect to the analysis pipelines in FlexMash, e.g., new pipeline annotators such as tokenizer, POS-tagger and named entity recognizer (NER) and new pipeline filter elements which, e.g., determine the average sentence length, the percentage of unknown words, lexical diversity and the fit of training data (QUALM data quality methods, cf. Table 5.1). In a second project the respective RESTful web services are located. These are easily integrated into FlexMash via an entry in the FlexMash registry and a mapping of its name to an Internet Protocol (IP) address.

[16] `https://www.mongodb.com/`
[17] https://opennlp.apache.org/

4.5 Summary and Future Work

In this chapter, a holistic concept for the measurement and improvement of the quality of text data and analysis results was presented. Our concept bases on a definition of data quality which does not look at data in isolation, but rather aims at a better fit of both components, data consumer (i. e., analysis tool) and data. In the context of a text analysis pipeline, therefore, many non-human data consumers need to be considered, such as preprocessors and classifiers. These tools oftentimes employ training data and/or semantic resources and presume certain characteristics of the input data. The latter are captured in the machine learning tools with weighted features. The presented concept enables a stepwise measurement and goal-centered improvement of the quality of data within analysis pipelines. It is based on information on both, the specifics of analysis tools as well as on the characteristics of the input data. In QUALM, newly developed data quality indicators and modifiers are employed. For example, the *degree of the fitness for use* of the employed training data with respect to the input data may be measured. If the repository R_T contains better fitting training data, a list of fitting training data sets may be presented to the analyst. Details on QUALM data quality methods as well as evaluation results for concrete data sets and analysis tools are presented in the next chapter. These show the effectiveness of the QUALM data quality methods with respect to an increase of the accuracy of textual classifiers and a positive effect of QUALM on a whole analysis pipeline.

In upcoming research work, new practical fields of application for QUALM are investigated, such as the use in teaching. Furthermore, in the course of a student project that has accompanied the work on this thesis [Beh16], possible options for visualizing information on data quality to the users are evaluated. In this project, Behringer investigated how visual analytics could assist the data analysis toolkit user with respect to the quality of data and analysis results. In a PhD project following the student thesis, the research on potential benefits of visual analytics within data analysis toolkits is currently deepened [BHM17].

Chapter 5

QUALM Data Quality Methods

In this chapter, the challenge arising from the uncertainty with respect to the quality of analysis results for unlabeled text data is addressed (cf. Section 1.2.2).

A central element within QUALM are the concrete quality methods. The **QUALM data quality methods** comprise both quality indicators as well as quality modifiers. Methods, which determine the percentage of abbreviations, spelling mistakes and unknown words in texts are examples for **QUALM indicators**. These estimate the quality of the input texts and the fit of the data with analysis tools. The data quality indicators are not necessarily measures for data quality. Rather, the indicators give clues on when a text is of bad or good quality. Moreover, they may build the basis for a coarse assessment of the degree of a potentially low quality. The corresponding methods for data quality improvement are the **QUALM modifiers**. For instance, they dissolve abbreviations, correct spelling mistakes and add information from semantic resources such as dictionaries to dissolve unknown words. In this chapter, an overview on all QUALM data quality methods is given. In the two subsequent chapters (Chapters 6 and 7), details regarding two of the QUALM data quality methods are added, which need more space for explanation and evaluation.

The remainder of this chapter is organized as follows: Firstly, related work with respect to concrete data quality methods is described in Section 5.1. Then, the QUALM data quality indicators and modifiers are presented in Section 5.2. In the remaining section, the evaluation of QUALM and the methods therein is detailed (Section 5.3). The presented evaluation results show how well QUALM data quality indicators may predict the quality of analysis results and further investigate how well the QUALM modifiers may increase it. Finally this chapter is concluded in Section 5.4.

Parts of this chapter correspond to revised and composite versions of excerpts of previous author publications that are cited at affected locations [Kie19, KRM19b]. All concepts in these publications were developed exclusively by the author of this thesis.

5.1 Related Work

Plenty research on the quality of structured data tries to capture the quality of a data set as a number (e. g., see [SC13], [CKK17]). For example, the percentage of null, out-of-domain and duplicate values indicate the quality of structured data sets and can be expressed as a number in [0,1]. These methods are based on a comparison of the structured data to a "perfect" version of the data (or parts of it) that represent the real world, or to a rule that describes characteristics of such perfect versions of the data set. Yet, for unstructured text data oftentimes neither "perfect" versions of the data nor rules exist. Unstructured textual data moreover needs to be processed in natural language processing pipelines. Thus, additional means to capture how text characteristics influence the quality of text analysis modules in such pipelines are needed. Many works present first conceptual ideas for data quality dimensions and first conceptual metrics for text (cf. Section 2.3.4). Especially, Sonntag suggests the conceptual data quality indicators lexical ambiguity and spelling quality, which are added to the list of QUALM methods presented in this chapter [Son04]. While some first approaches and concepts yet exist, however executable methods for texts are missing [BS16]. Moreover, methods to improve unstructured data quality are not mentioned in this group of existing research works.

Thus, data quality research on text is still in its beginnings. But the **quality of textual documents is already considered in other research areas and applications**. For instance, many isolated methods useful in determining unstructured data quality exist in natural language processing, information retrieval, automatic essay assessment, machine learning, speech recognition and image recognition (for overviews on these topics, see [Bis06, MRS08, JM09, ZOM12, CV15]).

For example, the quality of written student essays [MK00, ZOM12] and of posts in online discussions [WGM07] can be assessed automatically. Also, many companies provide guidelines in writing texts such as error reports. The guide-

lines ensure that the texts can be processed automatically in high quality, e. g., by machine translation systems [Kuh13]. For example, very long and nested sentences should be avoided with respect to the quality of an automatically generated translation of a text. Researchers on text simplification develop automated methods to simplify texts [Sha14]. Genova et al. suggest a framework to measure and improve the quality of textual specifications for software [GFM+13]. While these works give interesting starting points for the development of concrete data quality indicators, no implemented data quality indicators are provided and modifiers are not addressed at all.

Readability metrics such as the Flesch readability index capture how easy and fast a human may read and understand a text. Flesch's formula is based on the number of words per sentence and the number of syllables per word. For an overview on readability indices, see Klare [Kla74]. Many automatic readability checkers exist[1]. For these tools, no implementation details are given, and the code is closed-source. These readability metrics only capture a very limited set of text characteristics, namely the number of words, syllables and sentences.

Particularly in the medical domain, much work is done in automatically detecting and resolving abbreviations (e. g., see [LGM+18]). In these works, the focus lies on resolving abbreviations, but the percentage of abbreviations is not used as quality indicator.

Also, existing research works on word sense disambiguation [Pop18], spelling correction [WLB08] and correcting upper and lowercasing in texts [NLDS03] may serve as a basis for the development of concrete methods. With respect to an automatic measurement of the fit of training data (cf. Chapter 6) as well as an automatic selection of the best-fitting training data, research works in areas such as instance selection and semi-supervised text classification need to be considered as related work [OLAMTK10, LY18]. More details with respect to these methods as well as related work are given in the next chapter. When considering related work with respect to an automatic selection of the best-fitting semantic resources, the reader is referred to existing works for ontology ranking [JMS10, TA07].

Moreover, related research works investigate the **influence of single data characteristics and semantic resources on analysis result quality**. For example, Botha et al. [BB12] investigate the effect of text size on the accuracy

[1] e. g., hemingwayapp.com and readable.io

of language identifiers. They found that the smaller the text size, the lower accuracy is. In the evaluation in Section 5.3, this result is confirmed. Additionally, here more indicators are added besides text size, and the accuracies of a language identifier as well as of a part-of-speech tagger are investigated. In [PCF+12] a method to predict the quality of rule-based segmentation processes on unstructured text data is presented. In the context of search engines, the quality of the search results and its correlation to the quality of the data basis is discussed as well [Fei13]. Also, the effect of employed semantic resources on the quality of results of text analytics is discussed oftentimes [HBB13, HSS03, FP18]. So, semantic resources such as WordNet improve the quality of analysis results of sentiment analysis tools [BJB12]. Also, language representations based on neural networks, as suggested by Devlin et al., improve the accuracy of named entity recognizer tools as well as question-answer systems [DCLT18]. Furthermore, research on developing robust natural language processing tools for messy data falls into this category (e. g., [BLR+10, DRCB13]). These existing approaches look at single characteristics of data or analytics methods only and methods to improve data quality are not addressed. In difference, in this thesis comprising lists of data quality indicators are given and corresponding methods to improve data quality are considered as well.

While many valuable first reference points for quality indicators for text data exist, they do not cover all necessary aspects. They are oftentimes not executable or closed-source and come from fields different than data quality research and thus have a limited perspective on text quality. Moreover, in none of these related works text data quality methods are applied to various data sets of varying quality as characterized by text analysis modules.

5.2 QUALM Data Quality Indicators and Modifiers

In this section, concrete QUALM data quality indicators and modifiers are listed. In QUALM, the focus lies on the quality of data with respect to algorithmic consumers of data in a data analysis pipeline, and thus on the interpretability dimension of data quality (cf. Section 2.3.2).

Data quality indicators must be transferable to a number in the interval [0,1] where 0 indicates low data quality and 1 indicates high data quality (cf.

Section 2.3.3). Therefore, indicators can, e. g., be represented by yes/no-questions or by proportions of data items which have a certain characteristic. The standard approaches to more concrete indicators for the quality of structured data involve counting the number of missing values, wrong values or the number of outliers. For the case of unstructured data, different indicators are needed.

The definition of indicators is based on the data quality definition discussed in Section 2.3.2 and on related work in natural language processing, information retrieval, automated assessment and machine learning (as described in Section 5.1). The methods presented are applicable to unstructured text data and executable implementations are given.

The **data quality modifiers** are the corresponding counterparts to the quality indicators. They modify data and resources with the goal of improving their fit and thus to increase data quality.

Firstly, an overview on the data quality indicators and modifiers developed within QUALM is presented (cf. Table 5.1). Then, each method is described in more detail. Moreover, a conceptual description for all methods is given. Some methods are conceptual-only, these are italicized in Table 5.1. For all other methods, details on design decisions in their implementation are given. All prototypically implemented methods are freely available on GitHub[2].

While the concepts behind the methods "confidence" and "fit of training data" (cf. indicators/modifiers 8 and 10 in Table 5.1), are applicable to all types of unstructured data which are processed by statistical machine learning components, the remaining indicators are text-specific.

Evaluation results for prototypically implemented indicators (cf. indicators 1-10 in Table 5.1), are given in Section 5.3.3 (cf. Kiefer [Kie19]). Moreover, corresponding evaluation results for the three modifiers, which are corresponding to the indicators "abbreviations", "uppercased terms" and "fit of default training data" (cf. modifiers 1, 2 and 10 in Table 5.1), are shown in Section 5.3.4 (cf. Kiefer et al. [KRM19b]). In Section 5.3.5, moreover the effect of QUALM on a whole chain of analysis tools, i. e., on an analysis pipeline, is discussed (cf. Kiefer et al. [KRM19b]). These evaluations are based on news, prose, chat, tweets and industry text data (cf. Section 5.3.2).

Further data quality methods are evaluated with respect to different data sets and analysis tools or whole analysis pipelines. These are described in Section

[2] `https://github.com/kieferca/quality-indicators-for-text`

Table 5.1: QUALM Data Quality Indicators and Modifiers.
- indicates low quality, + high quality, * very low and very high values indicate quality problems

Type	Id	Indicator	Scale	Modifier
Data	1	Percentage of **abbreviations**	+[0,100]-	Dissolve abbreviations
	2	Percentage of **uppercased terms**	[0,100]*	Transform all terms to lowercase, *correct* or restore correct casing
	3	Percentage of **spelling** mistakes	+[0,100]-	Correct spelling mistakes
	4	**Lexical diversity**	[0,1]*	Apply stemming, lemmatization and dissolution of synonyms to decrease lexical diversity
	5	Percentage of **ungrammatical** sentences	+[0,100]-	*Simplify sentences*
	6	Average sentence **length**	In theory $R+$, in practice in [3,50]*	*Shorten and split long sentences*
	7	Percentage of **noisy text elements such as special characters**	+[0,100]-	*Remove noisy text elements*
Tool	8	**Confidence** of standard analysis modules	-[0,1]+	Increase confidence by means of better-fitting training data and resources
	9	Percentage of **unknown words**	+[0,100]-	*Decrease the percentage of unknown words, e. g., by means of semantic resources and by dissolving abbreviations*
	10	**Fit of training data**	-[0,1]+	Automatic selection of the best-fitting training data
	11	*Percentage of unused **free text fields***	*+[0,100]-*	Increase the portion of new and relevant information extracted by means of hybrid information extraction
	12	*Percentage of **ambiguous terms***	*+[0,100]-*	*Disambiguation on write: disambiguate ambiguous terms at point of data entry*
	13	Fit of **semantic resources**	-[0,1]+	Automatic selection of well-fitting semantic resources from a repository, increase coverage of semantic resources

5.3.6. Here, results with respect to the following methods are presented: Firstly, the *indicator*, "semantic resources" (cf. indicator 13 in Table 5.1) is investigated. Then, the *modifiers* which may lead to possible benefits by means of "spelling" correction (cf. modifier 3 in Table 5.1), stemming, lemmatization and dissolution of synonyms to decrease "lexical diversity" (cf. modifier 4 in Table 5.1) are evaluated. Also, first results with respect to increasing "confidence" by means of better fitting training data and resources (cf. modifier 8 in Table 5.1) are described. Finally, the modifiers which automatically select well-fitting "semantic resources" from a repository and which increase the coverage of semantic resources (cf. modifier 13 in Table 5.1) are investigated.

The data quality methods with respect to "fit of training data" and "free text fields" in information extraction approaches (cf. indicators and corresponding modifiers 10 and 11 in Table 5.1), are discussed and evaluated in detail in Chapters 6 and 7.

The QUALM data quality methods involve two main components of data analysis pipelines in measuring and improving data quality (cf. Sections 5.2.1 and 5.2.2 and the "Type" column in Table 5.1): (1) the data and (2) the analysis tools. For example, the percentage of abbreviations is measured on the input data (cf. *data* type indicator/modifier 1 in Table 5.1). To measure the fit of training data, the analysis tool and more specifically the training data employed, need to be considered as well (cf. *tool* type indicator/modifier 10 in Table 5.1)

In the following, data quality methods are discussed with respect to the data (Section 5.2.1) and tool (Section 5.2.2). For each method, the QUALM data quality indicator, modifier and their implementation are detailed in separate paragraphs.

5.2.1 Data Quality Methods with Respect to Data

In many texts, **abbreviations** are used heavily. To this end, the *indicator* "percentage of abbreviations" is suggested. The counterpart *modifier* resolves abbreviations. This can be done based on suitable resources such as lists with domain-specific abbreviations and their meaning.

The *implementation* of the indicator, which automatically measures the percentage of abbreviations, is based on the Stanford Named Entity Recognizer[3]

[3] `https://nlp.stanford.edu/software/CRF-NER.shtml`

```
# Define the CRF model for Stanford NER.
stanford_ner_model = os.path.join(os.path.join(current_dir_path, '''StanfordNER'''),
    'ner-model-abbr-detection.ser.gz')

# The command line argument for running Stanford CoreNLP.
stanford_core_nlp_command = ["java", "-Xmx45g", "-jar", stanford_core_nlp_jar,
    "-props", props_file,
"-file", temp_file, "-outputDirectory", temp_dir, "-encoding", "UTF-8"]

# The command line argument for running Stanford NER.
stanford_ner_command = ["java", "-jar", stanford_ner_jar, "-Xmx45g", "-cp", '"*;
    lib/*"', "-loadClassifier",
stanford_ner_model, "-outputFormat", "tabbedEntities",
"-testFile", depparse_file, ">", temp_file, "-encoding", "UTF-8"]

# Firstly the corpora will be dependency parsed with Stanford CoreNLP.
subprocess.call(stanford_core_nlp_command, shell=True)

# Then, the actual Stanford NER tagging is performed and the label ABBR is added
    to abbreviations.
subprocess.call(stanford_ner_command, shell=True)
```

Listing 5.1: Abbreviation detection based on Stanford CoreNLP and Stanford Named Entity Recognizer

and the Stanford CoreNLP[4] Java libraries. As shown in Listing 5.1, Stanford CoreNLP is used to dependency parse the texts (cf. Section 2.2.2). Stanford NER is instantiated with a new CRF model 'ner-model-abbr-detection.ser.gz' and finally applied to the text data to identify abbreviations. Stanford NER is a classifier which automatically recognizes named entities, such as persons, cities and companies. Therefore, it uses information gained via natural language processing, such as the part of speech tags and syntax. Also, it uses training data manually labeled with named entities. It is based on conditional random fields (CRF), a supervised machine learning algorithm for sequential classifications. The CRF sequence models used are described in [FGM05]. In the case at hand, the sequence to classify is a sequence of words. Given the sequence of words, the method classifies each word as abbreviation or non-abbreviation. As a basis for an adaptation of the Stanford NER classifier to the task of determining if a word is an abbreviation or not, it was trained on a new training data set. In this training data set all individual words in a text collection are annotated each with one of the two labels "abbreviation" or "non-abbreviation". The compiled training data set is based on excerpts of the data sets listed in Section 5.3.2. Furthermore, the Stanford Named Entity Recognizer was adapted to the task of detecting abbreviations by an implementation of additional features, which are based on natural language processing methods from Stanford CoreNLP:

- The length of the word.
- A boolean value, which indicates whether the word contains symbols or not.
- A boolean value, which indicates whether the word contains a period or not.
- Information on the dependencies, i. e., syntax, of the whole sentence the word occurs in (cf. [dMDS+14]).
- A string value, which represents the word in a simplified form which indicates the sequence of vowels (v) and consonants (c) representing the current word (such as "cvcvc" for the word "Vowel").
- A string value, which represents the word in a simplified form which indicates the wordform, i. e., the sequence of upper (u) and lowercased characters (l) representing the current word) (such as "ullll" for the word "Vowel").

[4] `https://stanfordnlp.github.io/CoreNLP/`

The classifier prototype was evaluated on unseen data resulting in a precision of 0.85 and a recall of 0.72. Thus, for the tested data sets it works reliably enough for the purpose of measuring the percentage of abbreviations as data quality indicator. For the calculation of precision and recall, the data sets as described in Section 5.3.2 were used and split into separate training and testing slices.

The prototypical implementation of the modifier is based on Python dictionaries. For the evaluation of this modifier in Section 5.3.4, manually compiled lists for various data sets are employed and abbreviations are dissolved by means of a straightforward matching algorithm.

Correct casing of textual data, i. e., ensuring correctly lowercased and **uppercased terms** in a text, also crucially helps in quickly capturing the meaning encoded in textual data. This is not only the case for human readers of the text (cf. [Kla74]), but also for analysis tools such as the language identifier (cf. Section 5.3.4). Several variants for measuring and improving casing exist. The easiest way to realize the *indicator*, is to measure the percentage of uppercased terms in a text. In the evaluation, this straightforward approach is chosen. As a variant, instead of terms, single characters in the texts may be investigated. Then, the percentage of uppercased characters in the text may be determined. For the *modifier* with respect to casing all characters may be lowercased. Moreover, correct casing might be recuperated. If a version of the texts with correct casing is available, it may simply be restored. This is straightforward, but can oftentimes be applied to text data sets in industry which were transformed to all-uppercase by some processing tool in the analysis pipeline. If no correct version is available, methods which base on machine learning may be applied to textual data to correct casing. For example, Cheng et al. base their approach on a Hidden Markov Model trained on a large raw corpus of case-sensitive documents [NLDS03].

The *implementation* is straightforward, the indicator can be implemented using standard string manipulation methods in NLTK which, e. g., determine if a character is upper or lowercased or transform text strings to lowercase.

The number of **spelling** mistakes in a text corpus may be an *indicator* for text data quality. Moreover, with the counterpart *modifier*, spelling mistakes can be corrected.

The methods may be both *implemented* using the Python implementation PyEnchant[5] or any other spelling correction module. An excerpt of the corresponding implementation of the indicator is shown in Listing 5.2.

```
counter = 0
d = enchant.Dict(language)
for word in data_set:
    if(d.check(word)== False):
        counter++
```

Listing 5.2: Counting 'spelling mistakes' based on PyEnchant

The modifier can also be implemented based on the PyEnchant library. The spelling correction in PyEnchant is performed on the word level and is based on internal dictionaries. These are available for German, British/American English and French. Alternatively, LanguageTool[6] may be applied to the texts for correcting spelling mistakes. In comparison to PyEnchant, in LanguageTool whole sentences are checked on the basis of grammatical rules. Pre-defined rule sets are available for various languages such as English and German.

The *indicator* **lexical diversity** reflects the variety of words used in a text. It is calculated based on a comparison of the size of the set of all words employed in the text with the total number of all word occurrences in the text (cf. Listing 5.3). As an illustrating example, consider the following text, which consists of two sentences and has a lexical diversity of 14/17: "*He* is going to a *pizza* shop. There *he* meets friends and *he* eats a salami *pizza*."[7] For the calculation of the size of the set of all words employed in the text, repetitions of words, such as *he* and *pizza* in the example, are not counted. The calculation of lexical diversity is based on a formula suggested in the NLTK book [BKL09]. Lexical diversity gives a clue with respect to text style. For example, in a novel, the words employed may repeat oftentimes and the vocabulary of all different words used in the text is rather low, i. e., the set of all words employed in the text is small. In difference, a collection of articles across many domains such as automotive, finance and health may employ a much larger vocabulary and words may be repeated less

[5] `http://pythonhosted.org/pyenchant/`
[6] `https://languagetool.org/`
[7] Not that upper and lowercasing as well as punctuation is disregarded in this illustrating example.

frequently. Thus, lexical diversity of the latter is probably higher. As counterpart *modifier* methods, stemming, lemmatization and dissolution of synonyms may be applied to decrease lexical diversity, as explained in Section 2.2.2.

The indicator is *implemented* using standard methods in NLTK for counting words. The relevant code is displayed in Listing 5.3, where the length (len) of the set of all tokens and words in the text (set) is divided by the length of all tokens and words in the text. The quotient may further be scaled up to enhance readability. The modifier bases on existing implementations for lemmatization, stemming and dissolution of synonyms in NLTK.

```
1 def lexical_diversity(text):
2 return (len(set(text)) / len(text))
```

Listing 5.3: 'Lexical diversity' implementation based on standard Python tools

Also, the percentage of **ungrammatical** sentences may be a good *indicator* for the result quality of analysis tools. Especially, very long and nested sentences are hard to process and potentially ungrammatical. Thus, simplifying and splitting long sentences into two may help as counterpart *modifier* method. Approaches to the simplification of textual data and to split long and nested sentences yet exist (cf. Section 5.1) and are adopted in QUALM.

For a prototypical *implementation* of the indicator, non-parsable sentences can be identified using an automatic syntax parser, e. g., parsers implemented in natural language processing libraries such as OpenNLP or the Natural Language Processing Tool Kit NLTK (cf. Section 2.2.3). In the prototype, the percentage of ungrammatical sentences, i. e., non-parsable sentences using an automatic syntax parser implemented in the natural language processing library OpenNLP, is identified. An excerpt of the respective method is displayed in Listing 5.4.

The **length** of sentences characterize text data and may *indicate* quality. They may moreover influence ease of processing. For example, the average sentence length of undergraduate texts is smaller than that of, e. g., poetic novels, and tweets are very short in comparison to news texts. As counterpart *modifier* method, long sentences may be shortened or split, as already described above for the indicator "ungrammatical".

```
numUncompleteParses = 0;
for(Parse p : topParses){
    if(!p.complete()){
        numUncompleteParses++;
    }
}
```

Listing 5.4: Counting 'ungrammatical sentences' based on OpenNLP

The indicator which assesses the average length of sentences can be *implemented*, e. g., using the tokenizer and sentence segmenter from the natural language processing library NLTK and by counting the determined tokens for each sentence.

Potential elements for the **noisy text elements** *indicator* are special characters, such as in urls, mail addresses, emoticons and punctuations. These may reduce the quality of analysis results, if analysis tools are sensitive with respect to these text elements. As counterpart *modifier* method, "noisy text elements" may simply be removed from the texts.

For a prototypical *implementation* of the concept, the number of punctuation may be, e. g., calculated using the standard part-of-speech tagger implemented in NLTK (which has an individual class for punctuation). The *isalpha()* method in Python can be used to automatically identify the percentage of special characters. Regular expressions can be used to automatically identify the percentage of URLs, mail addresses, emoticons and pause-filling words in texts. If identified, all "noisy text elements" may easily be removed from the texts by means of the very same regular expressions and methods.

5.2.2 Data Quality Methods with Respect to Analysis Tools

In the previous section we listed QUALM methods which consider the data. In this section, methods are presented which focus on the role of the analysis tools and their specifics such as training data and semantic resources (cf. Chapter 4).

The *indicator* **confidence** also focuses on data quality of text data as perceived from the point of view of a statistical classifier. A statistical classifier estimates the probabilities for each class from a fixed list of classes. These probabilities are also called confidence values (for more details, see [GFL06]). If the probability of a classification decision is very high, confidence of the statistical classifier is said to be high. Confidence is expressed as a number in the interval [0,1] and may be used for measuring data quality. As *modifiers*, QUALM methods may be selected, which increase the confidence, e. g., by means of better fitting training data and resources (see below and Chapter 6).

For the *implementation* of the indicator, the confidence of standard processing modules can be calculated for standard text classifiers, e. g., for the part-of-speech tagger. For example, confidence measures are available and can be retrieved for the natural language processing tools in OpenNLP[8] (such as the tokenizer and part-of-speech tagger), a Java library for natural language processing which is heavily used in industry applications because it has an Apache license. For instance, to get these confidence values for the part-of-speech tagger, the *probs* method is called which returns an array of the probabilities for all tagging decisions. The method is shown in Listing 5.5. Then, the mean over all sentences is calculated and returned.

```
1 POSTaggerME tagger = new POSTaggerME(model);
2 tagger.tag(sentence);
3 double probs[] = tagger.probs();
```

Listing 5.5: Excerpt of 'confidence of standard processing modules' implementation based on OpenNLP

The number of **unknown words** indicator may be defined as the number of words not known to a standard preprocessing tool in natural language processing, such as a part-of-speech tagger. The corresponding *modifier* methods decrease the percentage of unknown words, e. g., by means of semantic resources and by dissolving abbreviations as explained in this section for the indicator "abbreviations".

The percentage of unknown words may be *implemented* by applying the standard part-of-speech tagger implemented in NLTK to the texts, which has

[8] `https://opennlp.apache.org/`

an individual class for unknown words, i. e., 'X'. A code excerpt which shows tagging a text corpus and counting all words which were tagged as 'X', i. e., as an "unknown word" is given in Listing 5.6.

```
1 tagged_data = nltk.pos_tag(data, tagset='universal')
2 result = 0
3 for w in tagged_data:
4 if w[1] == 'X':
5 result += 1
```

Listing 5.6: Counting 'unknown words' based on NLTK

The *indicator* **fit of training data** directly follows from the definition for interpretability given in Section 2.3.2, when considering statistical classifiers as data consumers. The quality of text data with respect to a machine consumer can be measured by calculating the similarity of the input text data and the data expected by the data consumer D_C. In the case of statistical classifiers such as a part-of-speech tagger (which automatically assigns parts of speech to each token such as a word in a text) or sentiment classifiers (which automatically detects opinions in texts and assigns e. g., the classes positive, negative and neutral to texts), D_C may be represented by the training data. For the case of unstructured text data the similarity can be measured using text similarity metrics. For example, one may consider the situation where Twitter data is consumed by a statistical classifier such as a part-of-speech tagger that was trained on newspaper texts. By the definition of interpretability used in this work, data quality is lower than for another tagger that was trained on text data from Twitter as well. Examples for metrics for this indicator are text similarity metrics such as Cosine Similarity and Greedy String Tiling (cf. [DTI13]).

One option to improve the fit of input and training data is it to adapt the input data. To this end, modifiers as suggested in the previous section may be applied to the textual input data. Moreover, also the analysis tool and its specifics such as the training data may be modified. To this end, we suggest a *modifier* which selects the best-fitting training data for a given input data set and analysis tool, based on the training data repository described in Section 4.2.3. The indicator and the counterpart modifier are described in more detail in Chapter 6.

For a prototypical *implementation* of the simplified indicator "fit of Treebank training data", the Cosine Similarity metric from the DKPro Similarity library is employed [DTI13]. While this QUALM data quality method is investigated in more detail in Chapter 6, in the evaluation of the QUALM data quality indicators presented in Section 5.3.3, also results for the fit of Treebank training data are shown. This is a simplified version of the indicator, which does not consider the concrete training data set employed by the analysis tool, but only considers one sample default training data set. Since it is most often used as default in many processing modules in natural language processing, the Treebank data set is used as default (cf. Section 5.3.2). For measuring the fit of Treebank training data, the text similarity of the operational text data set that is actually being analyzed and this default training data set is calculated. The whole concept, design decisions in implementation and a thorough evaluation with various text similarity metrics will be presented in Chapter 6.

If isolated information extraction approaches are applied to structured data enriched by **free text fields**, relevant information might be missing from the analysis results. Isolated approaches only work on one data type, while the input data set indeed contains two-typed data, namely structured data fields as well as one or more free text field(s). Thus, data of either one of the types is not considered in the analysis.

This may be the basis for a further QUALM data quality *indicator* which may, e. g., be based on the number of unstructured free text fields not employed during analysis. As corresponding *modifier*, a hybrid information extraction approach is suggested, which structures German and English unstructured texts. The goal of this approach is to especially increase the portion of new and relevant information extracted by means of information extraction from mixed-typed input data sets, which consists of structured data fields enriched by free text fields. Details on the concept, a prototypical *implementation* and an evaluation are given in Chapter 7.

Ambiguous terms are words which have more than one possible meaning. For example, in the sentence "The jaguar is fast." at least three possible meanings of jaguar exist: (1) the sports car (2) the animal and (3) the operating system. As a second example consider the word "bank" which either denotes (1) a financial institution (2) the building where a financial institution offers services or (3) may be employed as a synonym for "do business with a bank or keep an account at a bank". Also, many ambiguous words are used in industry texts. These texts

describe, e. g., machines, errors, projects, persons and processes with ambiguous terms. The descriptions may lead to problems, when, e. g., the person responsible for a product may not be identified unambiguously from the textual description or if a defective machine or error type cannot be identified clearly. Plenty approaches that address the disambiguation of terms in text documents exist (cf. Section 2.2.2). Yet, all of these approaches start long after the point of text entry. The ambiguous texts were written hours, days or even years ago and only then disambiguation starts. In difference to these existing approaches, the possibility to disambiguate texts "on write", i. e., at the point in time of data entry, is investigated in a student work [Pan19]. This approach goes along with the common belief of many data quality experts, who state that the optimal point for improving data quality is the point of data entry [McC12, Red01, Sad13].

The *indicator* may be implemented based upon existing word sense disambiguation tools (cf. Section 2.2.2). The percentage of identified candidate terms may then be returned as a data quality indicator. A corresponding *modifier* could base on existing works for disambiguating the terms (cf. Section 2.2.2). Moreover, the approach described above may help to decrease the number of ambiguous terms "on write" and thus improve data quality. An advanced *implementation* and an evaluation of the "disambiguation on write" approach, which integrates search engines such as Lucene[9] and existing techniques to identify ambiguous terms, is up to future work.

With respect to **semantic resources**, which may be part of an analysis pipeline, an *indicator* which measures the fitness of semantic resources with respect to the input data is suggested. Within QUALM, it can be decided if resources are fit for use, based on the repository for semantic resources (cf. Section 4.2.3). Fitness may be measured by counting all terms in the input text which are also present in the resource. For example, this coverage is determined for a sample automotive taxonomy and error reports with respect to warranty issues in [KK15]. Also, input text as well as semantic resources may be vectorized, and the resulting vectors may then be compared with each other (cf. Section 2.2.2). To this end, all words are listed in a vector and they are weighted according to term frequency (tf) or based on the product of term frequency and inverse document frequency (tf-idf) (cf. Section 2.2.2). Then, the cosine between the text vectors is calculated (also see [MRS08]). The counterpart QUALM *modifiers*

[9] `https://lucene.apache.org/`

automatically select well-fitting resources from a repository and furthermore increase the coverage of semantic resources.

A study on measuring the fit of semantic resources and the automatic suggestion of best-fitting semantic resources has been performed in the course of a student project that has accompanied the work on this thesis [Lin19]. The prototypical *implementation* of the concept is built upon the Pachyderm analysis infrastructure[10]. Pachyderm especially offers version control for data. Moreover, repositories may be built and stored within the infrastructure, and data analysis pipelines together with the respective input and output data can be managed. In Pachyderm, analysis pipelines run in Docker containers, thus the implemented prototype is independent from the programming languages and analysis tools used within single analysis pipeline steps. The measurement of the fit of semantic resources is based on a Python implementation and employs the Gensim library[11]. Gensim is a framework for natural language processing which furthermore implements methods from information retrieval. The two standard weighting schemata, tf and tf-idf are used (cf. Section 2.2.2). Based on these weightings, a vector for the input data set as well as for all semantic resources available from the repository are calculated (cf. Section 4.2.3). Optionally, the vectors may be reduced in terms of dimensions by means of a matrix reduction based on Latent Semantic Analysis (LSA) (cf. Section 2.2.2). Then, the cosine between each pair of an input data vector and a semantic resource vector from the repository is calculated. Moreover, these calculations are performed for each step in the analysis pipeline. Based on this, for each step in an analysis pipeline and furthermore based on the corresponding input data, the fit of employed and/or all available semantic resources can be indicated to the user as data quality indicator. Moreover, based on the calculated similarities, the resources which fit best may be automatically selected. In the prototypical implementation, all resources available from the semantic resource repository are presented as a ranked list to the analyst. The items in the list are in descending order based on their similarity to the input data vector. Then, the analyst may decide whether to integrate a suggested resource or not. If he selects a resource, it is automatically integrated into the analysis pipeline by means of an automatic change in the metadata which describes the pipeline and by an automatic deployment with Python-Pachyderm after storing the changed metadata.

[10] `https://www.pachyderm.com/`
[11] `https://pypi.org/project/gensim/`

A second counterpart QUALM modifier with respect to semantic resources increases the coverage of semantic resources. In Kassner and Kiefer [KK15], we suggested to increase the coverage of a taxonomy by means of an adaption of these knowledge-representing resources to new domains and tasks. In this work, two data analysis projects A and B in the automotive domain are considered. A taxonomy and a corresponding matching algorithm were developed for a specific project A. If this taxonomy and the matcher are applied to a new project B, quality is low: only little concepts are recognized by the matcher and the coverage of the semantic resource, i. e., of the taxonomy is low. To improve the re-usability of resources from one project to another, in Kassner and Kiefer [KK15], we suggest to increase coverage by means of (1) improvements in data structure and matching algorithm and (2) self-learning coverage improvement. With respect to (1), e. g., the original annotator, which uses token-based nodes (cf. Schierle and Trabold [ST10]) was changed so that it uses single-character nodes to enable a slightly faster searchability. With respect to (2), synonym suggestions are given to human experts in a taxonomy maintenance app, based on well-established distributional measures such as context windows. These methods, e. g., consider context windows surrounding the candidate spans or the document vectors of the candidate spans for the concept synonym detection [Len18].

5.3 Evaluation

In this section, the evaluation of the QUALM concept over several concrete data sets, analysis tools and QUALM data quality methods is described. As a basis, the prototypical implementations of QUALM described in Section 4.4 and concrete QUALM methods as described in the previous section are employed.

In the next section, the evaluation method for QUALM is described (Section 5.3.1). Then, the data sets used within evaluation as well as the analysis tools employed are listed (Section 5.3.2). Next, evaluation results for QUALM indicators and modifiers are presented (Sections 5.3.3 and 5.3.4) and the effect of QUALM on a chain of analysis tools is discussed (Section 5.3.5). In the last section, additional evaluation results with respect to the QUALM methods 3 "spelling", 4 "lexical diversity", 8 "confidence" and 13 "semantic resources" are presented.

5.3.1 Evaluation Method

As explained in Section 2.2, the quality of text mining results is judged by comparing the predictions of the analysis tools with the gold labels annotated by human experts. For example, to determine the quality of a part-of-speech tagger, its accuracy is calculated as shown in Equation 5.1, here repeated from Equation 2.4.

$$ACC = \frac{(\#\ correct\ POS\ tags\ in\ tagged\ data)}{(\#\ total\ POS\ tags\ in\ tagged\ data)} \tag{5.1}$$

The accuracy of language identifiers is calculated by comparing the gold language labels with the labels added by the tool. As already mentioned in the introduction of this chapter, accuracies can only be calculated if manually added labels are available. This is oftentimes not the case. The calculation of the suggested quality indicators does not need such manually added labels, though. In Table 5.3, first results with respect to whether they are able to predict the quality of such tools are shown. To this end, the indicator values and the accuracy values calculated for various analysis tools are investigated with respect to a possible correlation. If the indicator values correlate with accuracy values, they may be employed as a reliable means to predict the quality of analysis results.

Another goal of QUALM is to improve the quality of analysis results. For the evaluation of the QUALM modifiers, single analysis tools with and without application of QUALM may be compared with respect to evaluation metrics such as accuracy. This method is applied for the evaluation of QUALM in Section 5.3.4.

Finally, besides a correlation with and an improvement of the accuracy of single analysis tools, moreover the effect of QUALM on a whole chain of analysis tools is discussed in the evaluation (Section 5.3.5). A systematic investigation of the effect of QUALM on a complete analysis process, cf. Section 4.3.2, presupposes gold labels for each single analysis tool in the process. For the analysis process excerpt as illustrated in the introduction to Chapter 4, see Figure 4.1, e. g., gold labels for language, part of speech, named entities and cluster categories would be needed. Most domain-specific data sets in industry do not come with gold labels, though. Freely available data sets usually are enriched with labels of one type only. For example, these contain gold labels for language, but not for part of speech, named entities or cluster categories. Thus the effect of QUALM on a

Table 5.2: Data sets used in the experiments.

Data	Text Samples with Abbreviations	# tokens
News	Lorillard **Inc.**, the unit of New York-based Loews **Corp.** that makes Kent cigarettes, (. . .)	40k
Prose	Recently, **WWRL** won praise for its expose of particular cases of employment agency deceit (. . .)	1.15M
Tweets	**RT** eye: **LMBO**! This man filed an EMERGENCY Motion (. . .); **LOL** I know, Lemme (...) a quick **s/o** !!!	35k
Chat	**r u** serious?; there **ya** go (...)	45k
Industry	Produkt **n.i.o**, **Ktk** kaputt **vd** verklemmt.; Verhaken sich in Schiene **T22**; **VR** nach **UI**, SPÄNE AM Bauteil.	153k

concatenation of, e. g., the analysis tools depicted in Figure 4.1, is not possible with evaluation metrics such as accuracy (cf. Section 5.3.1). Alternatively, a qualitative description of the effect of QUALM on a whole chain of analysis tools is given in Section 5.3.5.

5.3.2 Data and Analysis Tools Used in the Experiments

In the evaluation of QUALM, various data sets are applied. Firstly, the brown corpora collection (which consists of 14 different data sets, e.g., reviews and humourous texts) is employed as an example for **prose**[12]. Moreover, a part of the Penn Treebank is employed as concrete example for **news** texts and corpora with **tweets**, **chat** and **industry** data are used. While the remaining corpora are publicly available, the industry data set is non-public. It comprises information on downtimes in a production line and contains German free text information. The data set contains information on the reasons for downtimes and the actions that were taken to put the production line running again. The workers on the shop floor can fill the free text field via text entry into a tablet. The Tweet corpus comes from Gimpel et al. [GSO+11], all other data sets are from the corpus collection in NLTK[13]. All but the industry data set have gold labels for

[12] In the evaluation results reported for modifier 10, "training data", these subcorpora are employed

[13] `http://www.nltk.org/nltk_data/`

part of speech. Moreover, for all data sets, language is known, so that accuracy of part-of-speech taggers as well as language identifiers may be calculated following Equation 5.1. In Table 5.2, the main characteristics of the data sets are listed and concrete text samples are given, which illustrate the usage of abbreviations (boldfaced) and the style of the texts in the data sets. The text samples for the industry data set are anonymized.

Moreover, two different analysis modules are investigated, the language identifier and the part-of-speech tagger, and several different implementations, i. e., concrete **analysis tools** for each module. In the evaluation results in this section, accuracies averaged over all of these analysis tools are employed. While the accuracy levels of the analysis tools differ, all tools are equally sensible with respect to messy text data such as chat and industry data. Single accuracy values for each tool were furthermore given in the initial assessment in Section 3.4, in Tables 3.1 and 3.2.

As concrete examples for the language identifier, the following implementations are employed:

- Tika[14]
- Language-detector[15]
- LanguageIdentifier.[16]

Accuracies for three implementations of part-of-speech taggers are reported, based on the default Penn Treebank tagset and Treebank training data [MMS93]. These analysis tools are used within the evaluation in the subsequent Sections 5.3.3, 5.3.4 and 5.3.5 and will further be employed within the evaluation in Section 6.3. The following implementations of part-of-speech tagger were investigated:

- CRF POS-tagger[17]
- Perceptron tagger[18]
- TNT tagger.[19]

[14] `https://tika.apache.org/`
[15] `https://github.com/optimaize/language-detector`
[16] Available from the DKPro Core library: `https://dkpro.github.io/dkpro-core/`
[17] `http://www.nltk.org/_modules/nltk/tag/crf.html`
[18] `http://www.nltk.org/_modules/nltk/tag/perceptron.html`
[19] `https://www.nltk.org/_modules/nltk/tag/tnt.html`

5.3.3 Evaluation Results for QUALM Indicators

In the columns in Table 5.3, the data sets are noted. In the last column, moreover, the scale for accuracy and indicator values is given (repeated from Table 5.3). In the rows, the overall accuracies for the two text analysis modules are presented (cf. Section 5.3.2). Compiling manually added labels/annotations costs time and expert knowledge. Therefore, for the industry data no manually added labels of part of speech are available. Thus, part of speech (POS) accuracy can't be calculated for this data set. Nevertheless, the accuracy of the language identifier (LI) and the indicators give insights on the textual characteristics. In the following rows, the results for the suggested quality indicators for text data 1-10 are presented in Table 5.3. Note that the remaining indicators 11 and 12 are not implemented. Moreover, results with respect to indicator 13 are presented in Section 5.3.6.

The raw numbers gained for data quality indicators as described in Section 5.3.3, are reported. Thus, most indicators are measured in percent and some are plain numbers such as the average sentence length. In future work, these raw measurement results need to be transferred to uniform data quality metrics as already mentioned in Section 5.2. Also, further analysis modules, implementations and data sets need to be addressed in future work.

From first to last column, the overall accuracy of the two text mining modules decreases. Treebank and Brown (news and prose) can be processed in a reliable quality by these text mining tools (=overall accuracy is high). Tweets, chat posts and industry data can only be processed in low quality (=overall accuracy is low). A similar classification is made by the data quality indicators: Treebank and Brown contain less abbreviations and spelling mistakes and have a low lexical diversity. The amount of uppercased characters is low. They hardly contain ungrammatical sentences. The sentences are longer when compared to tweets and especially when compared to chat and industry data. The fit of training data and confidence are high, and the amount of unknown words is low.

Low quality of tweets, chat posts and industry data is indicated by many abbreviations, spelling mistakes and unknown words as well as a low fit of training data and low confidence values. Tweets contain a significantly higher lexical diversity than the other data sets. Chat posts contain particularly many abbreviations and lexical diversity is high. In both, tweets and chat posts, more words than in the other data sets are uppercased and they contain more

Table 5.3: Evaluation results for QUALM indicators.
- indicates low quality, + high quality, * very low and very high values indicate quality problems

	Data					
	Treebank	Brown	Twitter	Chat	Industry	Scale
Overall Accuracy	0.92	0.82	0.67	0.48	n.a.	-[0,1]+
Overall Accuracy LI	0.93	0.88	0.65	0.25	0.45	-[0,1]+
Overall Accuracy POS	0.90	0.75	0.68	0.70	n.a.	-[0,1]+
Percentage of Abbreviations (1)	2.0	0.6	4.6	11.0	7.1	+[0,100]-
Percentage of Uppercased Terms (2)	1.6	0.9	7.0	15.8	0.1	[0,100]*
Percentage of Spelling Mistakes (3)	19.0	12.0	27.0	34.0	23.0	+[0,100]-
Lexical Diversity (4)	0.015	0.001	0.106	0.046	0.004	[0,1]*
Percentage of Ungrammatical Sentences (5)	0.1	0.4	1.5	1.0	0.0	+[0,100]-
Average Sentence Length (6)	24.0	20.3	14.5	4.3	4.8	[3,50]*
Percentage of Noisy Text Elements (7)	0.20	0.16	0.28	0.24	0.41	+[0,100]-
Confidence (8)	0.9	0.9	0.8	0.6	0.7	-[0,1]+
Percentage of Unknown Words (9)	0.0	0.1	0.2	0.3	0.5	+[0,100]-
Fit of Treebank Training Data (10)	1.0	0.9	0.5	0.5	0.5	-[0,1]+

ungrammatical sentences. In chat and industry data the sentences are very short. The industry data is full of domain-specific abbreviations, unknown words, noisy text elements such as special characters and spelling mistakes. Lexical diversity is rather low and the sentences are very short and parseable, i. e., the number of ungrammatical sentences is low.

Both text analysis modules selected are high quality modules oftentimes employed in text mining projects in industry (cf. Section 5.3.2). While the language identifiers seem to be very sensible with respect to some text characteristics, the part-of-speech taggers are more robust. From Table 5.3 it can be seen that the suggested indicators are good starting points that may characterize the data and indicate analysis result quality. The indicators may be exploited in predicting the quality of analysis results. Thus, in a real analysis situation in humanities, science or industry, where accuracies are not calculable and thus not known, the suggested data quality indicators give a hint on how good processing modules may be able to cope with the data set(s). Low analysis result quality may especially be indicated by means of a high percentage of abbreviations, spelling mistakes, noisy text elements and unknown words. Also, a comparably short sentence length may indicate quality problems. The percentage of uppercased terms, the lexical diversity and the percentage of ungrammatical sentences correlate less clearly with the accuracies. Finally, the indicators confidence and fit of treebank training data are good indicators for the analysis result quality.

5.3.4 Evaluation Results for QUALM Modifiers

In this section, the results for the suggested quality modifiers for text data 1, 2 and a simplified version of modifier 10, i. e., with respect to abbreviations, uppercased terms and fit of training data are presented (cf. Table 5.1). Note that the remaining modifiers are not implemented (cf. modifiers 5, 6, 7, 9, 12 in Table 5.1) or are evaluated in Chapters 6 and 7 (cf. modifiers 10 and 11 in Table 5.1). Additional evaluation results with respect to modifiers 3, 4, 8, 13 are reported in Section 5.3.6 (cf. modifiers 3, 4, 8, 13 in Table 5.1).

In this section, the evaluation results with respect to the three QUALM modifiers are presented. Here, M_1 "abbreviations" and M_2 "uppercased terms", are used for language identifier and M_{10} "Treebank training data" for part-of-speech tagger (cf. Section 5.3.2):

- M_1: For the values *before application of QUALM*, the original data set is employed. For the values *after application of QUALM* abbreviations are dissolved on the basis of an abbreviation dictionary.
- For M_2 as starting point, the textual data set is presumed to be completely transferred to uppercase (for the values *before application of QUALM*). Oftentimes textual data sets in industry are only available in all-uppercased format. This may lead to crucial quality issues of language identifiers, which yet may be discovered and also prevented by QUALM. After application of M_2, the text as originally available is used, which mostly is correctly upper and lowercased (for the values *after application of QUALM*). In future work, additional methods may be employed, which automatically correct upper and lowercasing in texts (e. g., see Niu et al. [NLDS03]).
- M_{10}: The best-fitting training data set here is exemplary selected based on cosine similarity. For the values *before application of QUALM*, the treebank data set (news) is selected as training data in each case. This is a heavily employed default training data set of part-of-speech taggers and thus a training data set which is frequently employed by domain experts (cf. Marcus et al. [MMS93]). For the values *after application of QUALM*, the training data set is selected from R_T which has maximum cosine similarity to the input data. More detailed evaluation results, which also consider several possible variants of text similarity metrics such as Latent Semantic Analysis are investigated in Section 6.3.

In Table 5.4, for each modifier the average (avg) accuracy over all examined data sets and the investigated implementations for the language identifier module (for M_1 and M_2) as well as the part-of-speech tagger module (for M_{10}) is shown before and after application of the respective QUALM modifier. Additionally, the range of yielded accuracy values is illustrated by giving the minimal (min) as well as the maximal (max) value. All detailed result tables and prototypical implementations employed are provided on GitHub[20].

M_1 in average leads to an improvement of the accuracy from 0.53 to 0.54 (by 0.01), furthermore accuracy does not deteriorate in any of the cases. Most tested data sets only contain little abbreviations (cf. Section 5.3.3). For these data sets, it makes no difference whether abbreviations are resolved or not. The concrete analysis tools Tika, Language-detector and 'LanguageIdentifier' are

[20] `https://github.com/kieferca/qualm`

Table 5.4: Evaluation results for QUALM modifiers assessed based on accuracy (cf. Section 5.3.1) for the data and three implementations for the language identifier module (Tika, Language-detector, 'LanguageIdentifier') and the part-of-speech tagger module (CRF, Perceptron, TNT) (cf. Section 5.3.2).

Modifier, Analysis Module	Accuracy **before** the Application of the QUALM Modifier			Accuracy **after** the Application of the QUALM Modifier		
	min	max	avg	min	max	avg
M_1 'Abbreviations', Language identifier	0.20	0.96	0.53	0.21	0.96	0.54
M_2 'Uppercased terms', Language identifier	0.00	0.86	0.21	0.20	0.96	0.53
M_{10} 'Treebank training data', Part-of-speech tagger	0.52	0.89	0.71	0.54	0.96	0.82

differently robust with respect to the percentage of abbreviations contained in the data (cf. Section 5.3.2). Tika is very robust and accuracy only increases minimally for 2 of the data sets. For the 'LanguageIdentifier' and the 'Language-detector' and tweets, chat and especially for the industry data which contain many abbreviations, the difference is higher. So, for example for industry data and an analysis with the 'Language-detector', the accuracy increases from 0.45 to 0.59 (by 0.14). For the industry data set, with M_1, also the best accuracy value over all three analysis tools tested is reached for the 'Language-detector'.

Without application of M_2, the accuracy of the language identifiers in average is at 0.21. The value is very low. While 'Tika' is robust with respect to different upper and lowercasing in the data, the 'Language-detector' usually delivers analysis results with insufficient quality, if no correct upper and lowercased text is provided. The 'LanguageIdentifier' actually recognizes all sentences wrongly in the English data sets. With application of M_2, accuracy could be improved in average to 0.53 (by 0.32).

The automatic selection of training data as based on cosine similarity between input data and all training data available in R_T (M_{10}), leads in the test cases and for the considered analysis tools (cf. Section 5.3.2) in average to an improvement of accuracy from 0.71 to 0.82 (by 0.11). While for many clean data sets, such as most texts from the brown corpus collection, an employment of default news training data is no problem and leads to good accuracy values, M_{10} especially leads to an improvement of the accuracy for special data sets such as tweets and reviews (part of brown).

Beside these evaluation results, several single values also show a significantly higher accuracy of sentiment analysis tools when fitting resources are employed (cf. Section 4.3.1 and Balamurali et al. [BJB12]). Also a higher accuracy of language identifiers was found for longer texts when compared to shorter ones [BB12]. These quality improvements accumulate in the text analysis pipeline. In Kassner and Kiefer [KK15], we have shown for example how an improved recognition of the language may also lead to an optimized extraction of concepts. In the following section, the results for the application of QUALM on a whole analysis pipeline for cluster analysis of industry data is discussed.

5.3.5 Effect of QUALM on a Chain of Analysis Tools

In this section QUALM is evaluated with respect to a whole analysis pipeline, which is built with respect to application scenario 1 (cf. Section 3.1).

As already described in Section 5.3.1, a systematic investigation of the effect of QUALM on a complete analysis pipeline, is not possible with evaluation metrics such as accuracy. Alternatively a qualitative description of the effect of QUALM on the analysis pipeline depicted in Figure 4.1 is given, in which a cluster analysis of industry data is conducted.

In the first analysis step, the *language is identified*. On the basis of the language labels added, only texts with language label "German" are used in subsequent processing steps. The accuracy of language identifiers without the application of QUALM is very low (worst case accuracy of 0.21, see Table 5.4). Thus, more than three fourths of the text entries are not used in subsequent analysis steps. The application of QUALM on the analysis pipeline improves the average accuracy of language identifiers to 0.54 (cf. modifiers M_1 and M_2 in Table 5.4). If only the text entries that are correctly recognized as German without QUALM are employed as input to subsequent analysis steps, the analysis result of all subsequent analysis tools is qualitatively as well as quantitatively incomplete. With QUALM, yet more texts and thus also more information is being annotated/labeled by the subsequent analysis tools such as part-of-speech tagger and named entity recognizer. This also leads to a more complete text basis for cluster analysis with QUALM.

The positive effects of M_1 and M_2 on the subsequent *part-of-speech tagger* lead to a more comprehensive automatic *recognition of entities* in the subsequent analysis step. This effect is even strengthened by the selection of better fitting training data (M_{10}) for the part-of-speech tagger. For example, now the industry data also contains entities of type "error type", whereas without QUALM no instances of this entity type were being recognized. These new entities represent crucial information with respect to application scenario 1 (cf. Section 3.1).

Also, in the subsequent *cluster analysis*, this effect of QUALM on preceding analysis tools continues. Without QUALM, 9 out of the 10 biggest clusters are abbreviations. With QUALM, the resolved cluster names are determined instead of abbreviations. The final result of the analysis pipeline with QUALM in comparison to the result without QUALM were put into bar charts and shown

to experts for the industry data. The domain experts valued the result with QUALM as higher quality than the analysis result without QUALM.

5.3.6 Additional Evaluation Results

In this section, first evaluation results for further data quality methods are presented, namely for the **modifiers** 3 "spelling", 4 "lexical diversity", 8 "confidence" and 13 "semantic resources" (cf. Table 5.1). Moreover, for **indicator** 13 "semantic resources" evaluation results are discussed (cf. Table 5.1). The foregoing sections presented thorough evaluation results with respect to the language identifier and POS-tagger module and news, prose, chat, tweet and industry data, as described in Section 5.3.2. In difference, the evaluation of the QUALM modifiers 3, 4, 8, 13 and the QUALM indicator 13 as listed above base on different analysis modules such as sentiment analysis modules and named entity recognizer, or on whole analysis pipelines which are publicly available, e.g., on kaggle. Moreover, also different data sets are employed in this section, which are applicable with respect to the evaluation method (cf. 5.3.1) and the additional analysis tools and pipelines, i.e., which have appropriate gold annotations. Thus, the description of these additional evaluation results is isolated from the foregoing sections.

With respect to a first evaluation of the **spelling** correction modifier (cf. modifier 3 in Table 5.1), a study on the effects of spelling quality on the quality of analysis results has been performed in the course of a student project that has accompanied the work on this thesis [Gra16]. While differences over several data sets can be indicated by the percentage of spelling mistakes (as shown in Section 5.3.3), the automatic correction of spelling mistakes did not help in terms of accuracy in the experiments conducted.

For an evaluation of the modifiers stemming, lemmatization and dissolution of synonyms to decrease **lexical diversity** (cf. modifier 4 in Table 5.1), a study with respect to clustering quality has been performed in the course of a student project that has accompanied the work on this thesis [Ren19]. The core concept of this work relies on the fact that normalization procedures in natural language processing need to go along with a decrease in lexical diversity. For instance, stemming, lemmatization and synonym resolution lead to less various terms in the resulting texts than in the original input text. In Renz [Ren19], this effect is confirmed for 8 sample data sets and 6 different normalization steps. Lexical diversity thus may be a criterion in deciding whether a given text is yet normalized

or not. Moreover, normalization is known to help in terms of cluster quality [HSWL12, MRS08]. Therefore, further experiments in [Ren19] investigated if it is also the case that lexical diversity, which indicates the degree of normalization of the texts, and cluster quality correlate. Then, of course, lexical diversity would be a beneficial indicator for the expected result quality of clusterings. To this end, this effect was tested for three different clustering algorithms and numerous different parameter settings, especially with respect to the number of clusters and the length of input text units, but could not be confirmed for all test cases. Clustering quality was measured by means of extrinsic as well as intrinsic evaluation metrics such as silhouette score and purity [Rou87, MRS08]. As a summary of the results, factors such as the algorithm chosen and the parameter set are most relevant in terms of cluster quality. The effect of normalization as measurable via lexical diversity is comparably low and not consistent over all test cases. For some data sets and algorithms, normalization does not help, but even decreases the quality of clustering results. The effects found in this study need to be investigated in more detail, and for classification algorithms as well as clustering algorithms. This will be studied in future work.

In the following, first evaluation results regarding the suggested modifier for increasing **confidence** by means of better fitting training data and resources (cf. modifier 8 in Table 5.1) are presented. To this end, the impact of employing better-fitting training data on confidence was investigated in the course of a student project that has accompanied the work on this thesis [Gol19]. For a sample pipeline for sentiment classification of tweets[21], confidence and accuracy values for classification based on two different training data sets are compared. In the first variant, the classifier is trained on passably fitting movie reviews[22]. In the second variant, it is trained on the best-fitting training data, i. e., tweets[23].

- For the first variant, with passably fitting training data, confidence is 0.5 and accuracy is 0.63.
- For the second variant, with best-fitting training data, confidence is 0.9 and accuracy is 0.95.

In future work, these first evaluation results will be expanded to more analysis pipelines and data sets.

[21] Contained in the NLTK data collection available at: `http://www.nltk.org/nltk_data/`
[22] `https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset`
[23] To this end, the tweets contained in the NLTK data collection are split into training and test sets.

Finally, evaluation results for QUALM methods with respect to **semantic resources** are presented (cf. indicator and modifier 13 in Table 5.1). Two modifiers with respect to semantic resources are suggested in this work, the first modifier automatically selects well-fitting "semantic resources" from a repository, the second modifier increases the coverage of semantic resources. For a first evaluation of the concept and prototypical implementation of the first modifier and the corresponding indicator (cf. Section 5.2.2), 16 different semantic resources of three categories were employed: gazetteers, sentiment lexica and abbreviation resources. The impact of an automatic integration of semantic resources is tested on two different data sets for three sample analysis pipelines, one for named entity recognition, the second and third for sentiment analysis. The evaluation results are very promising. All experiments show a positive effect of the integration of semantic resources into analysis pipelines. In the following, the evaluation results obtained for the three concrete analysis pipelines are listed:

- A dictionary with positive and negative terms[24] is suggested for integration into a sentiment analysis pipeline. For a concrete sentiment analysis pipeline from kaggle[25], the integration of the resource increases precision by 0.02 and recall by 0.03.
- For a sample Stanford-based named entity recognition template/pipeline[26] and the CoNLL2003 data set, the highly ranked automatically suggested semantic resources are gazetteers. They improve precision by up to 0.29 and recall by up to 0.33.
- For a concrete sentiment analysis pipeline on tweets[27], the integration of the suggested semantic resources improve accuracy by up to 0.02.

The second modifier is suggested in Kassner and Kiefer [KK15] and increases the coverage of semantic resources. First evaluation results for this modifier are promising and show a significant increase in terms of the number of concept annotations made on two sample data sets, the NHTSA data set described in Section 7.5.1 and a confidential mixed-language industrial data set. The latter data set is different from the industry data set described in Section 5.3.2, which is employed for the evaluation throughout this thesis.

[24] `https://github.com/felipebravom/StaticTwitterSent/tree/master/extra/Sentiment140-Lexicon-v0.1`

[25] `https://www.kaggle.com/paoloripamonti/twitter-sentiment-analysis`

[26] `https://nlp.stanford.edu/software/CRF-NER.shtml`, `https://github.com/synalp/NER/blob/master/templates/scrf.props`

[27] `https://www.kaggle.com/jiashenliu/how-can-we-predict-the-sentiment-by-tweets`

5.4 Summary and Future Work

Concrete methods within QUALM which may especially be applied to text data sets were presented. Operational text data sets usually do not come with manual gold annotations/labels for text processing steps. Thus, the quality of many text analysis results is not known in text mining projects in the humanities, science and industry. To this end, data quality indicators were suggested which help in deciding if default text mining modules will deal easily with the textual data or not, i. e., if quality improvements are needed or not.

Moreover, each data quality indicator comes together with a corresponding modifier. If data quality problems are indicated, improvement methods may be applied to the data. In the experiments, an application of three concrete modifiers which consider abbreviations, casing and training data, leads to an increase of the accuracy of language identifier and part-of-speech tagger analysis tools. Based on these results, moreover the positive effect of QUALM on a whole chain of analysis tools was discussed.

In future work, measured percentages and raw numbers such as average sentence length need to be transferred into consistent data quality metrics in [0,1] where 0 means low and 1 high quality, e. g., based on adequate step functions (cf. Section 2.3). For example, the first indicator measures the percentage of abbreviations. A high percentage of abbreviations should result in a low quality metric and a low percentage of abbreviations in a high quality metric. This can be achieved by means of a step function, where, e. g., 0-1% abbreviations are transferred to the quality metric 1 and >10% to 0, etc. Furthermore how indicators may be combined to measure data quality will be explored in future work. Also, studies which investigate how the improvement of data quality as perceived by intermediate consumers influences data quality from a rather end consumer viewpoint, need to be considered in future work. Moreover, the concept will be evaluated with additional data sets, analysis tools and text analysis pipelines.

Chapter 6

Automatic Selection of Training Data in QUALM

In this chapter the challenging task of selecting appropriate training data is addressed (cf. Section 1.2.3).

In difference to data scientists, who oftentimes write program code, domain experts use simplified analysis toolkits such as RapidMiner to construct analysis pipelines from scratch. In an analysis pipeline, various analysis tools are applied to domain-specific data consecutively. For example, an analysis pipeline for the extraction of opinions from texts may consist of a first analysis tool that annotates where a least meaningful text unit such as a word begins and where it ends ("tokenizer"). A subsequent analysis tool in this pipeline then adds information on the part of speech of these tokens, such as adjective, noun or verb ("part-of-speech tagger"). Afterwards, a third analysis tool adds information on opinions, such as positive, negative and neutral ("sentiment analysis tool").

Most of the times, the domain expert who constructs such an analysis pipeline does not know for each step in the analysis pipeline which training data sets are available and what these training data may be used for. Thus, in many domain-specific analysis pipelines, "out-of-the-box" tools with default training data sets are used. For example, news texts are used as default training data for the tokenizer and part-of-speech tagger analysis tools mentioned above. However, these default training data sets are very different from the operational input data sets of real industry and domain-specific use cases. Thus, **the use of default training data sets can lead to low-quality analysis results when applied in practice**.

Of course, it would be optimal to compile a new labeled data set for each analysis tool and domain-specific input data set. Yet, preparing labeled training data sets is very time-consuming, expensive and demands expert knowledge (e. g., see Ide et al. [IP17]). Therefore, it is not realistic to have a "perfect" training data set for each analysis tool and domain-specific input data set and oftentimes default training data sets are used instead.

Moreover, the domain expert has no information on the impact of selected training data on the quality of analysis results. Traditional evaluation metrics for supervised analysis tools such as accuracy rely on labeled domain-specific input data [HS15]. Since labels are oftentimes not available for domain-specific data, information on analysis result quality is not measurable with evaluation metrics. Thus, no information on the quality of analysis results is available to the domain expert who constructs a domain-specific analysis pipeline. So, the domain expert usually cannot verify and thus cannot know, that the employment of default training data may lead to low-quality analysis results.

In this chapter, a method is suggested which addresses this quality issue arising from non-fitting default training data. In contrast to analytics as carried out by data scientists, the methods only apply to analytics carried out with simplified analysis toolkits such as Rapid Miner and by domain experts who are not IT or data analytics experts. Besides the focus on domain experts, the method suggested is crucial to many use cases, since domain experts and "citizen data scientists" make up the majority of the users of simplified analysis toolkits. Furthermore it is assumed that the domain expert who constructs an analysis pipeline based on simplified analysis tools neglects the initial task of selecting the best-fitting training data available. The major reason is that s/he often does not know which training data sets exist and what they are used for.

The main contributions described in this chapter comprise (1) a concept for an **assessment of the expected quality of analysis results** via a measurement of the similarity between operational input data and training data. To this end, a quality indicator based on similarity metrics is suggested. Depending on the data type, similarity metrics for images, speech, structured data or texts may be employed (cf. Section 6.1). Moreover, based on these similarity measurements and based on a repository of available training data sets, a method for (2) automatic **selection of the best-fitting training data** for a given analysis tool and operational input data set is suggested. Then, (3) a **prototypical implementation** of these concepts is presented. The method is exemplary

integrated into the FlexMash toolkit [HB16], which is a data processing and analysis toolkit suited for domain experts with little IT skills (cf. Section 4.4). In future, it may be easily integrated into other simplified analysis toolkits as well, e. g., RapidMiner. Finally, an (4) **evaluation** of the concepts for textual data and the part-of-speech tagger analysis tool is presented. Several text similarity metrics and findings are presented, which show that the accuracy of part-of-speech taggers can be crucially increased by an automatic selection of the best-fitting training data when compared to the accuracy of "out-of-the-box" tools which employ default training data.

As a sample use case, a use case described in the introduction to this work is considered (cf. Sections 3.1 and 3.3.1). A team leader in industry wants to get information on frequent reasons for downtimes on a production line and builds an analysis pipeline from scratch, within an easy-to-use data analysis toolkit such as RapidMiner[1] or SPSS[2]. To this end, s/he employs analysis tools which base on default training data such as gold-annotated, i.e., labeled, news texts, which are very different from the operational industry data s/he wants to analyze. Thus, this citizen data scientist may end up employing non-fitting training data within analytics, leading to low-quality analysis results. To prevent quality issues arising from non-fitting training data, in this chapter two more QUALM methods are suggested. The respective QUALM indicator automatically measures the fit of training data and the corresponding modifier automatically selects the best-fitting training data.

Section 6.1 outlines related work, before the concept is presented in Section 6.2. Then, evaluation results for textual data sets and the part-of-speech tagger (POS-tagger) analysis module are described in Section 6.3. Finally, this chapter is concluded (Section 6.4).

This chapter is a revised version of a previous author publication [KRM20]. All concepts in this publication were developed exclusively by the author of this thesis.

[1] `https://rapidminer.com/`
[2] `http://www.ibm.com/analytics/us/en/technology/spss/`

6.1 Related Work

An employment of the similarity of operational and training data as a quality indicator is suggested in this chapter. Wang and Strong [WS96] define data quality as "fitness for use by data consumers". This definition is extended and it is explained that data also needs to be fit for use by analytical processing tools (cf. Section 2.3.2). Many data quality frameworks and quality indicators for structured data exist (cf. Section 2.3.3). However, data quality frameworks and quality indicators for unstructured data are demanded, and executable quality indicators for unstructured data are missing (cf. Section 2.3.4). In Section 6.3.4, a possible correlation of text similarity with the expected accuracy is investigated, and thus text similarity as a new quality indicator for textual data sets.

The approach is based on similarity metrics. Several such metrics exist that are applicable to various types of data, such as database tables, images, videos and text (cf. Shirkhorshidi et al. [SAW15], Mielke [Mie12], Fuentes et al. [FBOGA12] and Section 6.3.3). In the evaluation, a focus is set on textual data and thus text similarity metrics are employed. Text similarity metrics play a crucial role in many research areas. For example, they are employed in automatic plagiarism detection [HB17] and automated assessment of student exams [ZOM12, PK15, KP15]. In Section 6.2.1 a quality assessment based on the automatic measurement of the similarity between input and training data in analysis pipelines is suggested. To the best of our knowledge, no previous work employs such similarity metrics as quality indicator.

Moreover, in this thesis, the similarity between operational and training data is employed for the automatic selection of the best-fitting training data. In the machine learning community, automatically creating, improving and enriching training data for a specific analysis purpose, is a heavily discussed topic. Many works termed as "training data selection" try to reduce the size of the training data without (drastically) affecting the quality in a negative way. For example, Wang et al. present an approach for support vector machines [WNC05]. They show that, while they reduce the size of the training data set, accuracy does not deteriorate. Work on "instance selection" tries to compile perfect training data with respect to runtime and accuracy on an instance level [OLAMTK10]. Traditional instance selection methods rely on labeled operational data, which in the setting presented is not available. Moore and Lewis select training data on an instance level for building a language model [ML10]. Here, the selection

is based on a pre-compiled in-domain model. Several approaches also extend labeled training data sets by means of unlabeled data. Axelrod et al. adapt training data sets for statistical machine translation systems to new domains [AHG11]. Li et al. address semi-supervised text classification [LY18]. Blum et al. suggest a co-training approach combining labeled and unlabeled data for the classification of web pages [BM98]. In "transfer learning", a predictive function is learnt from training data of one domain and then adjusted to a new domain by transferring knowledge from the source domain to the new domain [PY10]. All of these approaches are different from the presented method, since they assume that the initial training data set was already chosen or is composed, e. g., by means of "instance selection". They build up on initial training data or mining models, e. g., in terms of improving performance or coverage. The initial step of selecting these training data is often neglected by domain experts. Different from existing work, the approach addresses the first task of analysts to select the best available training data. Domain experts employ out-of-the-box analysis tools to build analysis pipelines from scratch and thus oftentimes also use default training data. They often neglect the task of selecting appropriate training data. For this reason, the method suggested in this chapter focuses on impeding failures by domain experts on an early stage.

6.2 FiT and SeT Methods to Prevent Low-Quality Analytics

The concept for the fit of training data as data quality indicator (FiT) is described in Section 6.2.1 and the corresponding modifier which automatically selects best-fitting training data (SeT) is illustrated in Section 6.2.2. The concepts are meant to be applied to all elements within analysis pipelines based on supervised machine learning tools. Analysis pipelines are described in Section 2.2. The data quality concept suggested in this chapter, is applied to each supervised-learning-based step separately, e. g., in an analysis pipeline as illustrated in Figure 2.1. Hence, it is applicable to more complex, also non-sequential workflows. Furthermore, the QUALM repository for training data and mining models builds the basis for the concept, as already described in Section 4.2.3.

6.2.1 Measuring the Similarity Between Input and Training Data: Fit of Training Data as Quality Indicator (FiT)

Operational input data sets from real use cases do not come with labels for text analysis modules. For these data sets, no accuracy values are calculable. Thus, the quality of many analysis steps in domain-specific text analysis pipelines is usually not known. For example, if a domain expert analyzes an industry data set without labels, he cannot determine the accuracy of an optical character recognizer (OCR), speech-to-text or POS-tagger module. Nevertheless, if a certain similarity metric and the accuracy correlate, the similarity value can be used as a quality indicator that informs the data analyst on how well the operational data and the training data used fit together. This goes along with the definition of data quality as "fitness for use by data consumers" as described by Wang and Strong [WS96]. Pointing at this definition, the "similarity between input and training data" is denoted by "fit of training data" and it is suggested as a new indicator for the quality of data within analysis pipelines. In Section 6.3 evaluation results are discussed, which show that this indicator correlates positively with accuracy.

The domain expert works with an analysis toolkit such as RapidMiner. Here, the domain expert can build analysis pipelines within a graphical user interface. In such tools, each analysis module is represented graphically, e. g., by a carat or a circle. This graphic representation may be highlighted with a green or red color, thus indicating high or low quality. With the FiT approach, this quality judgment may be based on the similarity of the operational and the training data. Each concrete analysis tool $e_{[m_k]_g}$, (cf. Figure 2.1), comes with metainformation on the training data set it employs as default. This default training data set can be retrieved from the repository described in Section 4.2.3. Then its similarity to the operational data set may be measured. Since similarity and accuracy correlate positively (cf. Section 6.3.4), the module may moreover be highlighted, e. g., with colors corresponding to quality levels. Thus, the analyst is informed and may react directly on quality issues, e. g., by changing the analysis tool or training data set employed. These steps may be performed for each (supervised learning) module in the analysis pipeline (1) before the modules are executed and (2) without need for labels in the operational data. This impedes failures and moreover saves time and resources in the construction of domain-specific analysis pipelines.

6.2.2 Automatic Selection of the Best-Fitting Training Data (SeT)

Besides giving feedback on the quality, the best-fitting training data available may be automatically selected from the training data repository, to improve analysis result quality. To enable access to the raw training data sets, the concept of a **training data/mining model repository** is introduced in Section 4.2.3. Here, various training data sets may be available with the same label type. These may be retrieved from the repository. The result is denoted with $R_{T_{m_k}}$, since this is a repository R for the set of all training data sets, T, which are applicable by module m_k (cf. Figure 2.1). Moreover, a training data set $t_{[a_{m_k}]_i}$ is denoted, which is in a certain set of available training data sets $R_{T_{m_k}}$, i. e., which has label a_{m_k}. Based on these notions and the formal description of analysis pipelines from Section 2.2, the selection method is illustrated for a sample module m_2 with label a_{m_2} in Figure 6.1. In the first step, all training data sets with the appropriate label for the module are selected from the training data repository, i. e., $R_{T_{m_2}}$ is calculated (see step(1) in Figure 6.1). In the example illustrated in Figure 6.1, six possible training data sets $t_{[a_{m_2}]_1}$ ... $t_{[a_{m_2}]_6}$ exist which have labels of type a_{m_2} and thus could be used by m_2. Then, the similarity is calculated for each pair of operational input data o and $t_{[a_{m_2}]_i}$ (see step(2) in Figure 6.1). In the last step, the training data set $t_{[a_{m_2}]_i}$ with the highest similarity to o is chosen. In the example given in Figure 6.1, $t_{[a_{m_2}]_6}$ has the highest similiarity 0.65 and is thus selected as the training data set.

In Formula 6.1, the selection function is shown. For each training data in the set of all applicable training data sets $R_{T_{m_k}}$, the score function $sim(t, o)$ is calculated, i. e., the similarity metric sim is applied to the training data and the operational data. Then, the t_i which maximizes this score function is selected.

$$\arg\max_{t \in R_{T_{m_k}}} sim(t, o) \tag{6.1}$$

After the most similar training data set is selected from the set of available training data sets, the concrete implementation $e_{[m_k]_g}$ maybe needs to be re-executed using the selected training data set. Based on a mining model repository (cf. Section 4.2.3), instead of generating a new mining model, which might take too long, analysis pipelines may be directly equipped with the corresponding pre-compiled mining models.

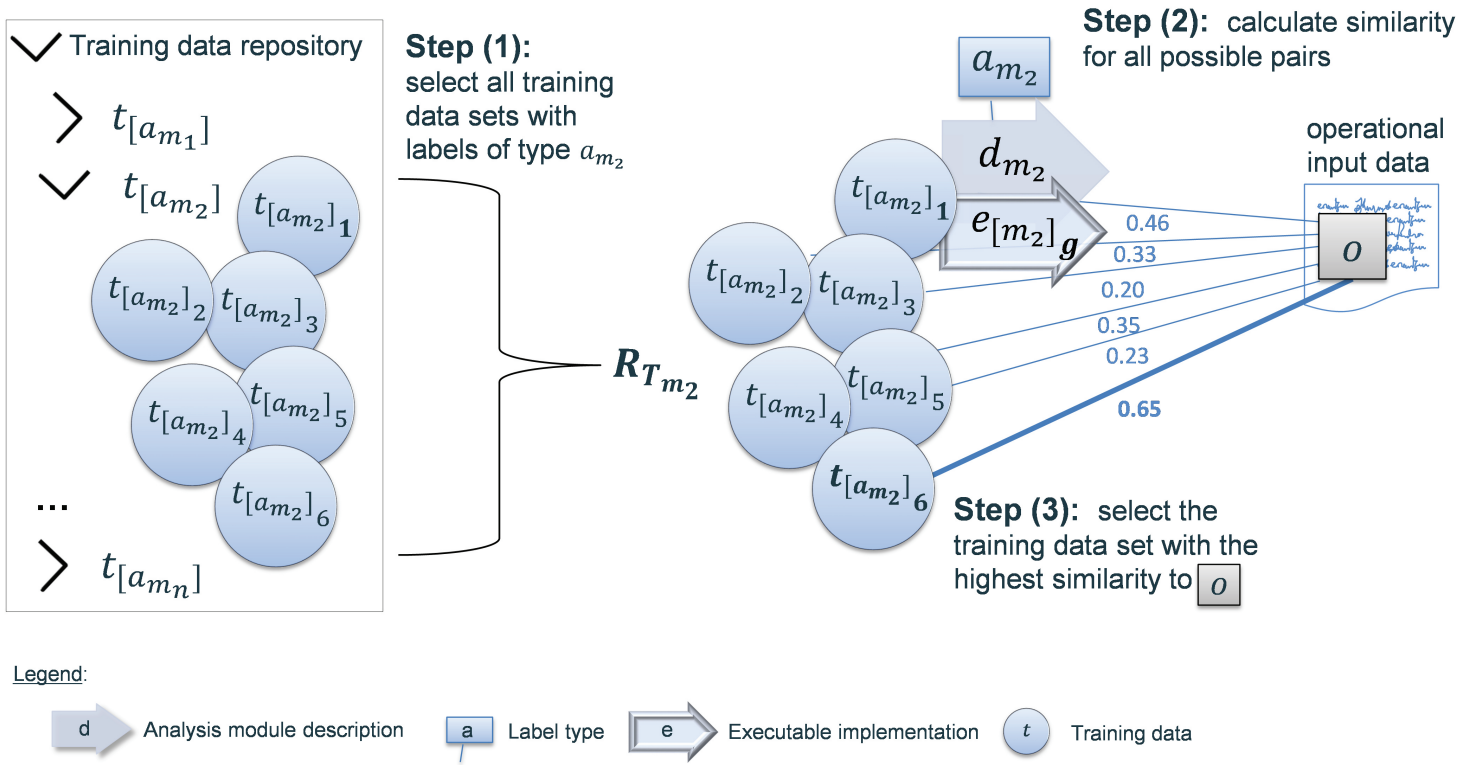


Figure 6.1: Illustration of the FiT and SeT methods for one module m_k, with a corresponding description d, an annotation type a, an executable implementation e and training data sets t: Step (1): Selecting training data sets and Step (2): Calculation of similarity metrics between training data sets and operational data by using FiT; Step (3): Selection of training data with the highest similarity to the operational data by using SeT.

6.3 Evaluation

While the general concept is applicable to various data types, in the remainder of this work textual data is focused. For textual data, it is especially hard to create perfectly fitting training data (cf. Ide et al. [IP17]) so that text analysis pipelines may benefit from selecting the most similar training data set available. In the following, the concept is evaluated for a text analysis module, which is present in almost any text analysis pipeline: the POS-tagger.

This section is started with a short description of the analysis tools and data sets used. Then, the prototypical implementation is outlined and the selection of relevant text similarity metrics is described. Finally, the evaluation results are detailed.

6.3.1 Data and Analysis Tools used in the Experiments

As described in the background to this thesis in Section 2.2.2, a part-of-speech tagger (POS-tagger) automatically assigns a parts of speech, such as *verb*, *adjective*, *noun*, to each word in a text. Consider the following sample sentence together with the part of speech tags as assigned by the NLTK POS-tagger (CRF): Can/*verb* we/*pronoun* automatically/*adverb* select/*verb* the/*article* optimal/*adjective* training/*noun* data/*noun* ?/*punctuation mark*

A POS-tagger module relies on training data. Several training data sets with manually assigned labels exist, see below. For the evaluation of the concept suggested in this chapter, the same three high quality, feature-rich and heavily used implementations of the POS-tagger module are employed as already used in the evaluation of the other QUALM methods in Section 5.3:

- CRF POS-tagger[3]
- Perceptron tagger[4]
- TNT tagger.[5]

The Python-based scripts used in the evaluation are available on GitHub, together with the prototype (cf. Section 6.3.2). The quality of a POS-tagger is determined by comparing the tags predicted by the system (by the tagger) with

[3] `http://www.nltk.org/_modules/nltk/tag/crf.html`
[4] `http://www.nltk.org/_modules/nltk/tag/perceptron.html`
[5] `https://www.nltk.org/_modules/nltk/tag/tnt.html`

manually annotated labels. The standard metric for measuring the quality of a tagger is its accuracy (ACC, cf. Section 2.2). The equation to calculate the accuracy is repeated here as Equation 6.2. Note that the concrete numbers in Equation 6.2 vary with the operational data set o, the executable implementation e chosen and with the training data set t which was used to train the POS-tagger.

$$ACC = \frac{(\#\ correct\ POS\ tags\ in\ tagged\ data)}{(\#\ total\ POS\ tags\ in\ tagged\ data)} \tag{6.2}$$

The experiments are conducted on **18 data sets from different domains**, such as news, prose, reviews, governmental texts, humorous texts, tweets and chat posts. The goal is a collection of as many different gold labeled data sets possible. To this end, firstly the same data sets as already described in Section 5.3.2 are employed: a subset of the Penn Treebank (news), the Brown corpora collection (prose), a Twitter corpus and a data set with chat posts. The Twitter corpus was taken from Gimpel et al. [GSO+11], all other data sets are taken from the collection of corpora which comes with NLTK[6]. Moreover, the CONLL 2000 data set is added as a further example for a news text. The huge "Brown corpus collection" contains 1.15M tokens and consists of 14 data sets of different genres, such as religion, mystery and reviews. These are employed as single data sets within the experiments in this chapter. Furthermore, all of the data sets come with manually annotated part of speech tags and are freely available. Thus, in the evaluational setting, the accuracies may be calculated and, furthermore, all evaluation results are reproducible.

6.3.2 Prototypical Implementation

In this section, a prototypical implementation of the concept explained in Section 6.2 is described, which was developed for the validation of the concept. In Figure 6.2, the processing pipeline is illustrated. The prototype was implemented in the course of a student project that has accompanied the work on this thesis [Lau16].

Several software frameworks implement text similarity measures. The DKPro Similarity library [DTI13] is chosen, since it is open-source, actively developed

[6] `http://www.nltk.org/nltk_data/`

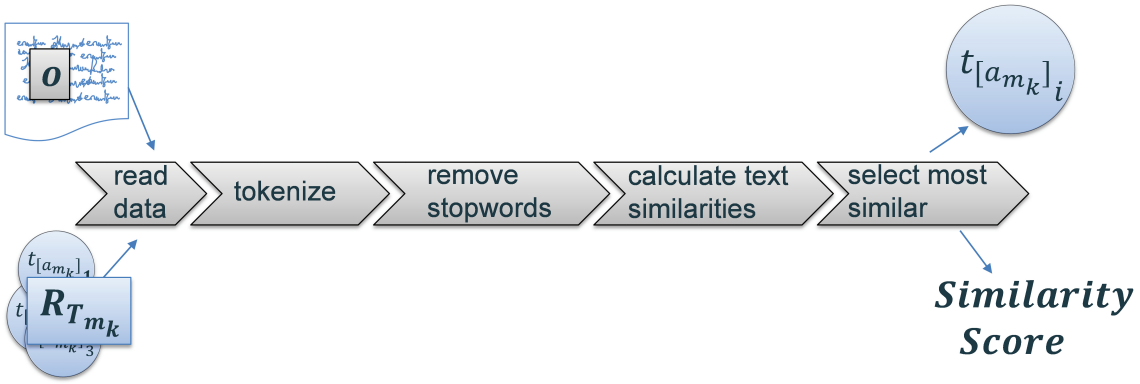


Figure 6.2: Processing pipeline for the prototypical implementation of the FiT and SeT methods.

and easy to use. It is part of DKPro, a collection of tools and programming libraries for Natural Language Processing developed at the technical university of Darmstadt in Germany. DKPro Similarity is based on the Apache UIMA framework[7], which ensures easy extensibility and scalability. A standalone prototypical implementation is provided on GitHub[8]. The prototype first **reads data** from two different folders in the file system in txt format. One folder contains the operational data set, e. g., in the use case described in Section 3.3.1, the industry data. The other folder contains all training data sets available, e. g., labeled news, tweets and chat data. Before measuring the text similarity, standard preprocessing is executed: the texts are **tokenized** (split into the least meaningful units such as words) and **stopwords may be removed** based on a stopword list. The standard stopword list in NLTK is employed[9] and all remaining words may additionally be normalized to its lemma (= base form of the word). The resulting lists of tokens are then compared by using **text similarity**, e. g., Cosine Similarity or LSA (cf. Section 2.2.2). In measuring the FiT (cf. Section 6.2.1), the similarity between the training data set and the operational data set is returned. A low value, e. g., caused by non-fitting news texts employed as training data for the analysis of operational industry data, is indicated to the domain expert. In SeT (cf. Section 6.2.2), for each possible pair of operational and training data, the text similarity is calculated, the training data sets are ordered by similarity and the **most similar training data set is**

[7] `https://uima.apache.org/`
[8] `https://github.com/kieferca/training-data-selection`
[9] `https://gist.github.com/sebleier/554280`

selected. As a result, the name of the best-fitting training corpus is returned along with the score. For example, a related industry data set or annotated tweets may be selected instead of news training data in the use case scenario (cf. Section 3.3.1). In addition, the method is packaged as web service and integrated into the flexible and easy-to-use data processing and analytics toolkit FlexMash [HB16]. Analogously, it may be integrated into further analysis toolkits such as RapidMiner. Thus, the method will be available to domain-experts who want to build analysis pipelines from scratch.

Table 6.1: Relevant metrics: Semantic and string-based text similarity metrics.

Type	Concrete metric	Time in seconds	Reference
Semantic	**Cosine Similarity**	16.7	[MRS08]
	Latent Semantic Analysis (LSA)	22.7	[LFL98]
String-based	**'WordNGramJaccard'**	47.1	[MRS08]
	GreedyStringTiling	out-of-memory	[Wis93]
	Levenshtein	out-of-memory	[Lev66]

6.3.3 Similarity Metrics for Textual Data Sets

There are several similarity metrics for textual data. They focus on different aspects such as string-based similarity or semantic similarity. Further metrics consider structural, phonetic or stylistic characteristics of the texts. For an overview, the reader is referred to Bär et al. [DTI13]. Most text analysis modules employ features which are semantic and string-based. Thus, these two metric types and 2-3 well-known concrete metrics for each type are considered (see Table 6.1). For the task of quickly supporting the domain expert in the construction of analysis pipelines, the selection must also be fast. The performance of SeT (cf. Section 6.2.2) is tested based on a prototypical implementation (cf. Section 6.3.2) on a PC with a 64-bit system, Intel(R) Core(TM) i7-6600U, 2.60 GHz, 2 cores and 16 GB RAM. Processing times are calculated for all 18 data sets (cf. Section 6.3.1) and the experimental settings as described in Section 6.3.5.

Two string-based metrics, GreedyStringTiling and Levenshtein are not applicable to big data sets with billions of characters. Time complexity is $O(n^3)$, hence they result in out-of-memory errors after considerably longer processing times of several 10-minutes. However, we found CosineSimilarity, LSA and

"WordNGramJaccard" to be applicable as they have processing times of 16.7, 22.7, and 47.1 seconds. Thus, for the evaluation of the concept, the bold-faced metrics listed in Table 6.1 are focused.

6.3.4 Evaluation Results regarding the Automatic Measurement of the Fit of Training Data (FiT)

In this section, the question is examined, whether similarity is a suitable quality indicator or not. Therefore, the evaluation considers how strong the similarity of the training data and operational data *sim* correlates with the respective accuracy *ACC*. In Equation 6.3 the spearman correlation metric for two data rows *sim* with similarity values and *ACC* with accuracy values is calculated (cf. Section 6.3.1 for a description of the accuracy metric (ACC), Sections 6.3.3 and 2.2.2 for an overview on similarity metrics and Section 2.2.1 for the background on spearman correlation).

$$Spearman_{rg_{sim},rg_{ACC}} = \frac{\sum_{i=1}^{n}(rg_{sim_i} - \overline{rg_{sim}})(rg_{ACC_i} - \overline{rg_{ACC}})}{\sqrt{\sum_{i=1}^{n}(rg_{sim_i} - \overline{rg_{sim}})^2}\sqrt{\sum_{i=1}^{n}(rg_{ACC_i} - \overline{rg_{ACC}})^2}} \quad (6.3)$$

All 18 data sets are employed in the experiments (cf. Section 6.3.1). In the evaluation, all possible pairs of operational and training data are compared, where operational and training data are not the same. Thus, 18*17=306 pairs of operational and training data can be generated. The values for all pairs build two data rows, one with the similarity metrics and one with the accuracies. For each pair of operational and training data, it is checked how good similarity and accuracy correlate.

Consider the following two concrete examples, where a domain expert wants to build a text analysis pipeline and immediately gets feedback on quality.

In the *first example*, the domain expert wants to analyze newspaper texts such as the texts in the CoNLL data set (cf. Section 6.3.1) with out-of-the-box text analysis modules. Operational news text data such as the CoNLL data set and out-of-the-box training data such as the Treebank data set (news) are very similar. A high Cosine Similarity of 0.95 of CoNLL and Treebank news texts indicates high quality. In the experimental setting, only data sets with manually annotated labels for parts of speech are employed. Thus, it may be

further assessed if this quality judgement is valid by comparing it to the accuracy values. The accuracies may be calculated as shown in Equation 6.2. For the CoNLL and Treebank data sets, Cosine Similarity is 0.95 and accuracy is 0.89. Thus, in this example the high similarity correctly indicates high analysis result quality, i.e., high accuracy. In an analysis toolkit, graphical representations of the modules may be colored with respect to the similarity values, and thus with respect to the expected analysis result quality. For example, an analysis module which turns green may indicate high quality and a red coloring may indicate low quality. Thus, the domain expert who builds an analysis pipeline in the analysis toolkit gets immediate feedback on the expected quality of analysis results.

In the *second example*, the domain expert wants to analyze reviews with a text analysis module that was trained on chat data. The graphic representation of the module may turn red, thus indicating low quality. The reason is that operational review data and chat training data are not similar. The Cosine Similarity of reviews and chat data is 0.51. Also, for reviews and chat data, a low accuracy of 0.56 is calculated.

The similarity and accuracy values in the two examples above indicate a high correlation between Cosine Similarity and accuracy. Beside looking at single value pairs as in these two examples, the two complete data rows of similarity and accuracy values as obtained for the experimental setting, which consists of 18*17=306 pairs of operational and training data, may be compared by calculating correlation metrics. Spearman's rho is calculated, see Kaltenbach [Kal12] using the implementation which is part of the scipy package in python[10]. It is applied to two data rows X and Y with n values: $X = x_1, ...x_n$ and $Y = y_1, ...y_n$. A positive value for spearman's rho implies that if x rises, y rises as well. A negative rho value means that x and y correlate in the opposite direction: when x rises, y falls.

In Table 6.2, spearman's rho is given together with the p-value (for the p-value, cf. Section 2.2.1). Spearman's rho indicates correlation of two data rows, where the *first data row* consists of the similarity values. Three variants are considered, i.e., the similarity rows generated by LSA, Cosine Similarity and "WordNGramJaccard". The *second data row* is made up by the average accuracy values of the POS-tagger, as listed in Section 6.3.1. For detailed result tables which compare the similarity metrics to the full distribution of training data

[10] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html`

Table 6.2: Correlation of Text Similarity and Accuracy.

Text Similarity Metric	Spearman's rho	p-value
LSA	0.84	$1.58e^{-83}$
Cosine Similarity	0.80	$1.52e^{-69}$
'WordNGramJaccard'	0.75	$4.20e^{-57}$

sets and corresponding accuracy values, see the GitHub repository. Here also separate values are listed for each executable implementation tested. While the POS-taggers have different overall accuracy levels, for all three taggers the accuracy values do correlate with text similarity.

The p-value represents the probability that a non-correlating system produces data sets X and Y with the specified spearman correlation (for details on the p-value cf. Fisher [Fis35]). The very low p-values for all similarity metrics indicate that the calculated correlations are valid and that it is very unlikely that data with such correlations could be generated by chance. The "WordNGramJaccard" metric and accuracy have a solid positive spearman correlation. The Cosine and LSA similarity metrics have a strong positive correlation with accuracy.

6.3.5 Evaluation Results regarding the Automatic Selection of Training Data (SeT)

In Section 6.2.2, a method to automatically select the best-fitting training data within analysis pipelines built by domain experts is suggested. Here, evaluation results with respect to this task are presented. To this end, again all 18 data sets are employed (cf. Section 6.3.1) and 18 experimental settings are compiled. Each experimental setting consists of an operational data set and a training data repository containing 17 disjunct available training data sets.

For the experiments, *for each* operational data set **o** and *for each* training data set **t** and *for each* text similarity metric **sim** and *for each* executable implementation **e** the following steps are performed:

- *step (1):* the text similarity $sim(t, o)$ is calculated.
- *step (2):* a POS-tagger implementation e is trained with t and tested on o (i. e., its accuracy is calculated, see Equation 6.2).

Table 6.3: Gain in Accuracy (ACC) with the suggested SeT method compared to a default and worst selection of training data.

Accuracy Gain by SeT compared to applying a default training corpus		Accuracy Gain by SeT compared to the worst selection of training data that could be made	
arithmetic mean	maximum value	arithmetic mean	maximum value
0.12	0.27	0.26	0.44

In Table 6.3, the gain in accuracy yielded by the SeT method is reported. To this end, the accuracies with SeT are compared to those with **default** training data, and with the **worst** selection the domain expert could make. The accuracy values for all 3 taggers and all 3 similarity metrics are considered and the arithmetic mean and maximum values are presented in Table 6.3. The full distribution of possible results across the different training data sets and taggers can be found on GitHub (cf. Section 6.3.2). The POS-tagger implementations used are listed in Section 6.3.1. While the POS-taggers have different overall accuracy levels, the accuracy of all three taggers is equally sensible with respect to employing default, non-fitting or well-fitting training data.

Compared to the worst selection that could be made, a selection based on text similarity improves accuracy by an average of 0.26 and by up to 0.44. When compared with the out-of-the-box versions of the taggers represented by the Treebank training data set (cf. Marcus et al. [MMS93]), improvements of 0.12 in average and up to 0.27 were found. The Treebank data set is an annotated gold corpus consisting of news texts, which is most heavily used as default training data in out-of-the-box POS-tagger modules. While default training data, as often employed in out-of-the-box modules, performs well compared to a bad selection of training data, the method presented crucially improves the quality of the POS-tagging text analysis modules in both cases.

6.4 Summary and Future Work

Domain experts without an IT/data analytics background build analysis pipelines from scratch. In this chapter, a concept that prevents low-quality analysis results within these analysis pipelines is provided. This concept employs the similarity between operational input data and training data as quality indicator

and automatically selects the best-fitting training data. Thus, it prevents the domain expert from employing non-fitting default training data or wrongly selected training data that lead to low accuracies. In the first part of this work, the concept is presented, which is based on a training data repository and similarity metrics. While the concept presented is applicable to various data types, unstructured textual data is focused in a prototypical implementation and evaluation. To this end, a choice of useful text similarity metrics was made and evaluation results for the POS-tagger module that is a part of most text analysis pipelines are presented. The results are very promising. First, it was shown that the similarity metrics Cosine Similarity and LSA correlate positively with the evaluation metric accuracy. Thus, Cosine Similarity and LSA may be used as quality indicators. Finally, evaluation results with respect to the automatic selection of best-fitting training data were presented. As a result, the method leads to higher accuracies of POS-tagger tools without any additional effort for the domain expert.

In future work, additional analysis modules will be considered and more experiments will be conducted with respect to the automatic selection of training data. Moreover, methods such as "instance selection" and "transfer learning" (cf. Section 6.1) should be investigated and made available to domain experts in easy-to-use analytics toolkits.

Chapter 7

Exploiting Structured Data Within a Text Mining Process

In this chapter, a hybrid information extraction approach is suggested, which exploits structured data within a text mining process (cf. the challenge described in Section 1.2.4).

Many data sets in research and industry capture information both in structured and unstructured data fields. Structured data fields are suitable if the data type and value domain fit to the perceived purpose. For example, structured data fields are appropriate to store the duration of a downtime in a production line in seconds. Unstructured data fields are better if no suitable structured type is available or if one needs to express certain issues in natural language to be readable and understandable by human users. For example, unstructured free text fields are adequate when explaining how to repair a machine, since this information is complex and cannot be captured in structured data. Especially, humans tend to provide more complete information using natural language texts than using structured information [HW96]. Thus, it is important to extract information not only from structured data, but also from unstructured, e. g., textual data as is mostly available in data sets from production, aftersales and research.

Standard approaches for information extraction from unstructured text data do not use structured data in text analysis (cf. Section 7.1). For example, information already available in a categorical field in structured data, such as information on car components, is usually not used in text analysis. The result of such text analysis may be information already available from structured data, e.g., information on car components mentioned in the texts. Isolated approaches miss

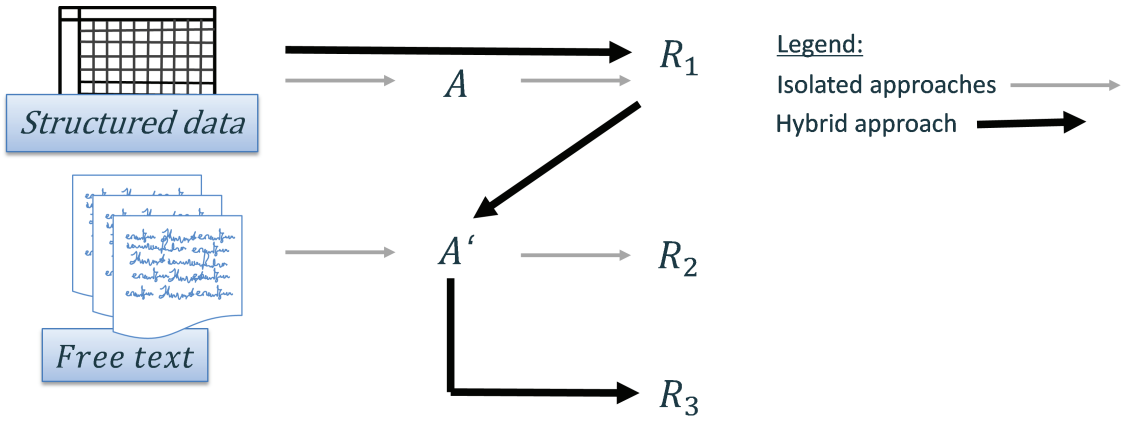


Figure 7.1: Isolated information extraction approaches A on structured data and A' on unstructured text data yield results R_1 and R_2 respectively. The hybrid information extraction approach uses analysis result R_1 in the text mining process and yields result R_3, thus extending results R_1 and R_2.

crucial information. As illustrated in Figure 7.1, in contrast to these approaches, the hybrid information extraction approach exploits analysis results obtained from structured data in the text analysis pipeline. Especially, information which is already known from the structured data, e.g. car components already known from a structured data field, are used for a first grouping of free texts and are further removed from the free texts. So, information with repect to issues with cars instead of information on car components as already known from structured data, re-emerges in the result of the hybrid approach to information extraction presented in this chapter.

The hybrid information extraction concept may be applied to all information extraction approaches on structured data with embedded or linked text data. In this chapter, the concept and a prototypical implementation are presented. In the evaluation, two real-world data sets are presented and the method is applied to them. For these data sets and use cases, no information on what to extract from the data is known beforehand, and no training data sets are available. Thus, a clustering algorithm is selected for the prototypical implementation (cf. Section 7.4). Clustering is a robust and unsupervised means to extract information from text.

The goal of the suggested hybrid approach (yielding R_3 in Figure 7.1) is to increase the amount of new information, when compared to the information

gained by the two isolated approaches (R_1 and R_2 in Figure 7.1). In the evaluation based on the two real-world data sets and the prototype, the **"degree of new information"** is denoted with i_{new} and it is defined as shown in Formula 7.1.

$$i_{new} = (c_{new}/N) \tag{7.1}$$

where c_{new} is the number of cluster names not already known from the structured column and N is the number of all clusters considered.

By employing this purely quantitative metric, a comparison of R_1, R_2 and R_3 is possible in a straightforward way. Additionally, we compare the results to the work of Ghazizadeh et al. [GML14]. In future work, more qualitative insights as well as additional quantitative metrics are considered, e. g., based on entropy. With an isolated approach on structured data (R_1), all information is extracted from the structured data fields only. Thus, based on the definition above, $i_{new} = 0$. Only, if information from free text is employed, i_{new} increases. R_2 and R_3 differ due to the exploitation of structured information available. By filtering out information already known from the structured data within the text mining process, i_{new} increases (for R_3 when compared to R_2). For the prototype and the two real-world data sets, for R_3, i_{new} is by 0.21 and 0.43 higher, than for R_2.

The main contributions presented in this chapter are:

- A description of the concept of hybrid information extraction.
- A discussion of design issues of a prototypical implementation of the approach for English as well as German free text fields.
- An evaluation of the hybrid information extraction approach. Here, the results of isolated approaches are compared (cf. R_1 and R_2 in Figure 7.1) with results of the hybrid approach (cf. R_3 in Figure 7.1) and it is shown that i_{new} is higher for the hybrid approach. For this purpose, the prototype is applied to an open dataset on problems with cars in aftersales (NHTSA data set[1]) and to a dataset on downtimes in a production line.

In the next two sections, an overview on work related to the approach is given and the hybrid information extraction is motivated with an example use case (Sections 7.1 and 7.2). In Section 7.3, the method used for hybrid information extraction is described, and implementation details are discussed in Section 7.4. The

[1] `https://www-odi.nhtsa.dot.gov/downloads/`

benefit of the approach suggested is illustrated with two data sets in Section 7.5 and this chapter is concluded in Section 7.6.

This chapter is a revised version of a previous author publication [KRM19a]. All concepts in this publications were developed exclusively by the author of this thesis.

7.1 Related Work

Plenty research works propose text mining approaches on free text. Compared to this work, these publications make no use of analytical results of structured data in the text mining process. Many approaches look at free text information in isolation. In the following paragraph, an excerpt of these approaches is presented which work with real data sets:

Carter et al. show a use case in the pharmaceutics domain where they mine the Pillreports.com database using the k-means algorithm [CH14]. Gamon et al. apply clustering to mine opinions on cars in the car reviews database[2]. The approach is based on a self-defined clustering algorithm [GACOR05]. Brooks focuses on preventing industrial accidents [Bro08]. In his approach, the SAS Text Miner Software is used to mine workers' compensation claims data. Clustering is based on the Expectation Maximization algorithm. Forman et al. assist technical support staff in a call center applying a self-developed clustering method on call logs [FKS06]. In many of the data sets used in these isolated approaches, also information in structured data fields is available. The main drawback of these approaches is however that they do not make use of this information source and thus crucial information may not be exploited.

Many approaches use both structured and unstructured information in information extraction in a parallel fashion. Yet, these approaches solely integrate the results of the isolated approaches. For many data sets like the data considered in this work such approaches are problematic, since valuable information may get lost. For example, Tan et al. mined data of a service center to get information on the expected processing times of service requests. Mining is based on structured data (the processing times) and case descriptions in free text fields [TBHG00]. In their approach, they build a classification model which uses features induced separately from structured and text data. Chougule et al. speed up repair

[2] `https://www.msn.com/en-us/autos/`

tasks of cars based on a framework which combines association rule mining, case-based reasoning and text mining [CRB11]. While the whole framework considers structured as well as unstructured data, the text mining component analyses the texts in isolation using hierarchical clustering algorithms.

Similarly, many approaches convert unstructured text data into structured data fields with the goal of merging structured and converted unstructured data and information. In contrast to the method suggested, the information extraction methods work on the texts in isolation. For example, the DeepDive system structures free texts using statistical inference and machine learning [Zha15]. Gubanov et al. present the data tamer system [GSB14], where the structuring of textual data is based on external tools that are not described in more detail. After the conversion of the unstructured text data, modules such as schema integration and entity consolidation in data tamer may be applied.

Various approaches to information extraction are called hybrid, since they combine two machine learning algorithms. Silva et al. combine naive bayes, the PART algorithm and the k-nearest-neighbour with hidden markov models [SBP06]. Xiao et al. combine maximum entropy and maximum entropy markov models [XZZ08]. These approaches still analyze one type of data in isolation. They are not hybrid in the sense of using analytical results on structured data in the text mining process.

The work most related to the research presented in this chapter is by Ghazizadeh et al. [GML14] and uses the same data set as employed in this chapter (NHTSA data set, cf. Section 7.2 and 7.5.1). Ghazizadeh et al. investigate reasons for fatal car accidents. They apply LSA and hierarchical clustering to the free text fields in isolation. In difference to the approach suggested, this work uses structured information in a first step only, before clustering takes place, to filter out the relevant part of the data. All structured information are ignored in the next steps of the information extraction process. Ghazizadeh et al. [GML14] present evaluation results which show that half of the cluster names correspond to vehicle components. These vehicle components represent information which is also available in a structured data field in the data set. In the result reported by Ghazizadeh et al. thus only half of the clusters represent information which is not already known from the structured data. In the hybrid approach suggested in this work this inconvenience is addressed.

While many approaches for the extraction of information from structured and free text data exist, the main drawback is that they are isolated: they do not

employ structured information that is available and helpful within the text mining process. In this chapter, this issue is addressed and results for two data sets from the product lifecycle show, that a hybrid approach to information extraction leads to new information that otherwise would be hidden behind information already known from the structured data.

7.2 Motivating Example

The department for National Highway Traffic Safety in the U.S. (NHTSA) wants to reduce the number of traffic crashes (cf. application scenario 2 in Section 3.2). For this purpose, they conduct recalls of unsafe vehicles and collect and analyze data on car crashes and problems with cars in a huge database since 1995. The data set contains structured information such as the car component affected. Customers filled this data field choosing the appropriate car component from a drop-down menu. Moreover, the NHTSA data set contains a free text field. The free text field describes the car crash or problem with the car. In Table 7.1, a small example data set containing information on the car component and a free text description is shown.

An isolated analysis on structured data for example yields an ordered list of the most frequent car components involved in car crashes (cf. R_1 in Figure 7.1). An isolated analysis on the free text field also lists primarily car components (cf. R_2 in Figure 7.1). The results of Ghazizadeh et al. showed that half of the information in R_2 is not interesting to the analyst since it contains too much information also contained in the structured field and the degree of new information i_{new} thus is comparably low ([GML14], cf. Section 7.1). They applied an isolated LSA and hierarchical clustering approach to the NHTSA data set.

The hybrid approach presented in this thesis, tries to tackle the problem by increasing the degree of new information. It yields a list of frequently mentioned terms that are not deducible from the structured part of the dataset. R_1 and R_2 are oriented on car components, but R_3 mostly is oriented on issues. For example, in R_3 (cf. R_3 in Figure 7.1) the analyst finds new valuable information among the 5 highest ranked clusters (which will be discussed in more detail in Section 7.5.2): Many customers report problems in getting new secure car parts that the manufacturers need to change due to a recall. In isolated approaches the analyst misses this information since it is not present in R_1, and in R_2 it is

Table 7.1: Example data set with structured (id, component) and unstructured information (description).

id	component	description
1	AIR BAG	AIR BAG FAILED DURING ACCIDENT (...)
2	AIR BAG	AIRBAG FAILED TWICE.
3	AIR BAG	AIR BAG LIGHT FAILED.
4	STEERING	VERY SENSITIVE STEERING AT HIGH VELOCITY.
5	STEERING	STEERING FAILED.
6	ENGINE	THE ENGINE SHUT OFF TWICE ON THAT DAY (...)
7	ENGINE	ALL ENGINE LIGHTS CAME ON (...)

ranked as place 175 only (cf. Section 7.5.2 for more details on the prototypical implementation yielding R_2). The information that customers have problems in getting new secure car parts may be crucial in preventing car crashes. Customers, while waiting for the secure car parts, might decide to drive the car anyway.

Furthermore, the information in R_3 can be used to improve and extend the categories available in structured data in a feedback loop. The analyst may decide to add the "unavailability of car parts" to the future structured data values available to the customers who file a complaint in the NHTSA database.

7.3 Hybrid Information Extraction Approach

The goal of the approach suggested is to extract more information from structured and free text data in terms of a higher degree of new information as defined in Formula 7.1 in the introduction to this work. In Figure 7.2, an isolated approach to information extraction is illustrated and the resulting table is shown in Figure 7.3. In Figures 7.4 and 7.5, an example illustrating the hybrid information extraction method and R_3 is given. Since the difference between standard isolated approaches and the hybrid approach are emphasized in this thesis, preprocessing steps (such as tokenization) and vectorization are not described in this section, which both approaches have in common. These steps are explained in detail in the next section. Here, the focus lies on the steps special to the hybrid approach: (1) grouping and (2) removal.

In Figure 7.2, an isolated approach is illustrated. Here, free text fields are considered in isolation and all free texts are clustered. Finally, the name of the cluster in which a free text falls is added to the overall data set in the additional column "cluster" as shown in Figure 7.3. The cluster name is based on the most frequent word in the cluster.

In Figure 7.4 the first processing step of the hybrid approach is illustrated, in which structured data is used to **group** free text fields. Here, the NHTSA data set is grouped by the structured data field on the car components into three groups (AIR BAG, STEERING and ENGINE) (step (1)). In Figure 7.5, the next step is illustrated, in which all information that is already available in the structured data field on car components is **removed** (step (2)). Only then, the free texts are **clustered** (step (3)). Finally, a new column is added to the table which contains the name of the cluster, the result is shown in the last step in Figure 7.5. The isolated approach adds cluster information such as "air bag" and "steering". This information is already available in the structured field "car component" (see columns "component" and "cluster" in Figure 7.3). The hybrid approach results in clusters, such as "light" and "fail". They represent new valuable information (see column "cluster" and compare to column "component" in Figure 7.5). The approach is based on three predominant characteristics of structured data sets with embedded free text fields, which are discussed in more detail in the remaining paragraphs of this section.

First of all, **free text fields store valuable information**. This was confirmed in many studies (cf. Section 7.1). Various methods, such as relation extraction, classification and clustering can extract valuable information stored in free text [ZM16]. Clustering is used since it is suited best for the use cases considered. Moreover, Ghazizadeh et al. [GML14] also used a clustering approach, and the results shall be comparable with their findings. However, the hybrid information extraction approach suggested in this chapter is independent from the concrete information extraction method chosen. Also relation extraction and classification approaches may benefit from applying the concept to them. For example, a classifier which uses structured fields as well as unstructured free text fields in the feature generation, may benefit from removing information already present in structured fields from the free texts. The concrete effects on further information extraction methods need to be investigated in future work. Here, the validation of the core concept is focused and a state-of-the-art clustering technique is employed.

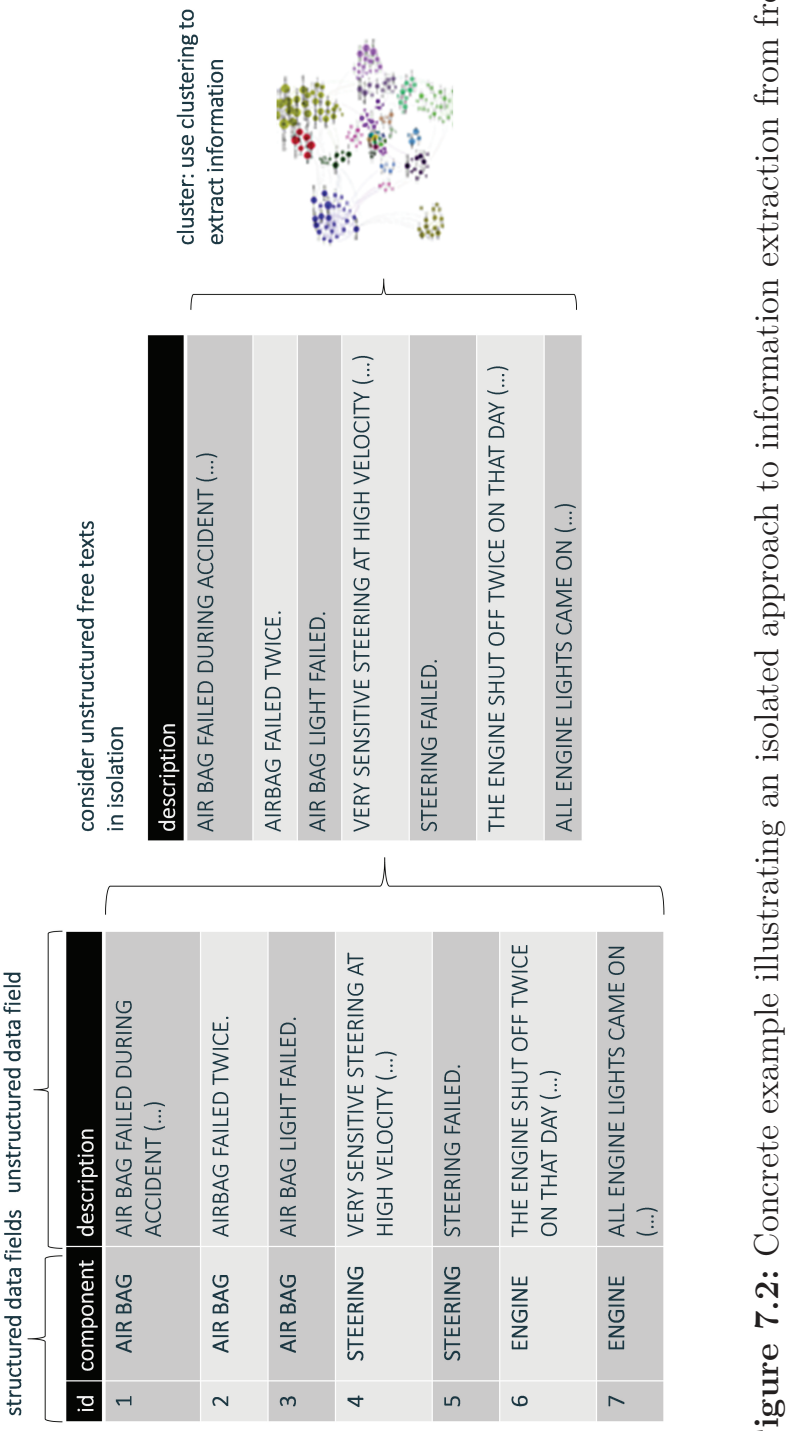


Figure 7.2: Concrete example illustrating an isolated approach to information extraction from free text fields.

id	component (R1)	description	cluster (R2)
1	AIR BAG	AIR BAG FAILED DURING ACCIDENT (...)	air bag
2	AIR BAG	AIRBAG FAILED TWICE.	air bag
3	AIR BAG	AIR BAG LIGHT FAILED.	air bag
4	STEERING	VERY SENSITIVE STEERING AT HIGH VELOCITY (...)	steering
5	STEERING	STEERING FAILED.	steering
6	ENGINE	THE ENGINE SHUT OFF TWICE ON THAT DAY (...)	engine
7	ENGINE	ALL ENGINE LIGHTS CAME ON (...)	engine

Figure 7.3: Concrete example illustrating the result R_2 of the isolated approach to information extraction. It does not bring new information but contains information already seen in R_1 such as "air bag", "steering" and "engine".

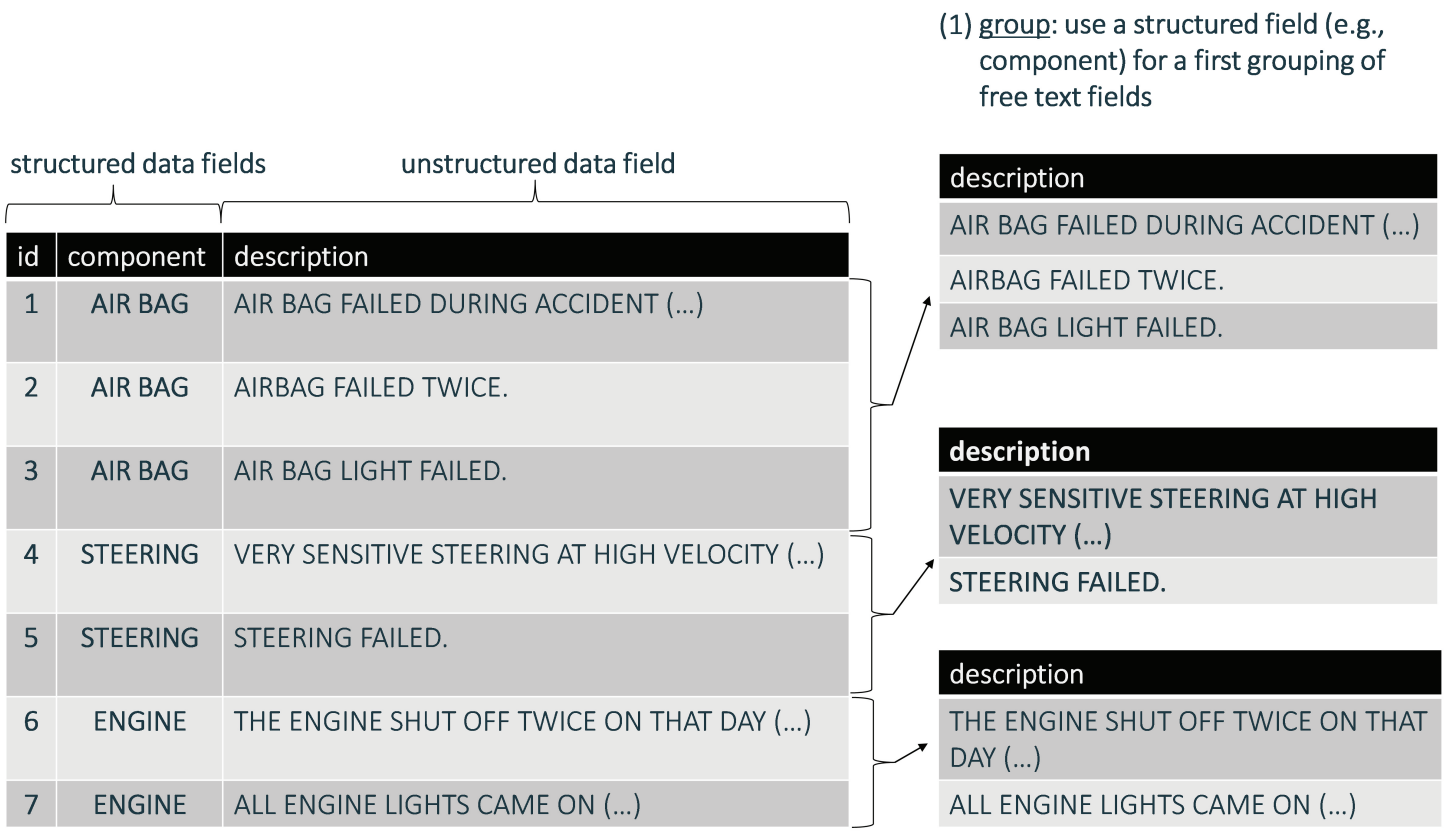


Figure 7.4: Concrete example illustrating the distinguishing step "group" of the hybrid approach.

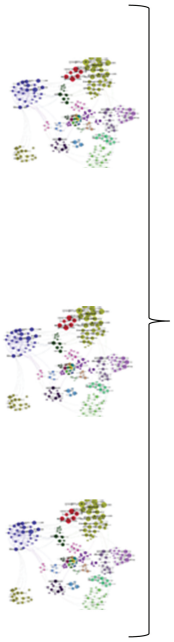
(2) remove: information from free texts which is already available in a structured field (e.g., component is removed)

description
~~AIR BAG~~ FAILED DURING ACCIDENT (...)
~~AIRBAG~~ FAILED TWICE.
~~AIR BAG~~ LIGHT FAILED.

description
VERY SENSITIVE ~~STEERING~~ AT HIGH VELOCITY (...)
~~STEERING~~ FAILED.

description
THE ~~ENGINE~~ SHUT OFF TWICE ON THAT DAY (...)
ALL ~~ENGINE~~ LIGHTS CAME ON (...)

(3) cluster for each group



add information on clusters to the data

id	component (R1)	description	cluster (R3)
1	**AIR BAG**	AIR BAG FAILED DURING ACCIDENT (...)	fail
2	**AIR BAG**	AIRBAG FAILED TWICE.	fail
3	**AIR BAG**	AIR BAG LIGHT FAILED.	light
4	**STEERING**	VERY SENSITIVE STEERING AT HIGH VELOCITY (...)	sensitive
5	**STEERING**	STEERING FAILED.	fail
6	**ENGINE**	THE ENGINE SHUT OFF TWICE ON THAT DAY (...)	shut off
7	**ENGINE**	ALL ENGINE LIGHTS CAME ON (...)	light

Figure 7.5: Concrete example illustrating the distinguishing steps "remove" and "cluster for each group" of the hybrid approach and the result R_3 containing new valuable information such as "fail", "light", "sensitive" and "shut off".

Second, in many data sets in industry and research, **unstructured free text fields can be grouped via information encoded in structured data fields**. For example, the NHTSA data set (cf. Section 7.5.1) may be divided into groups based on structured fields, such as car component, year and car make. The hybrid approach uses this information for grouping. Thus, the analyst does not end up extracting the same groups based on text mining free text fields. For a concrete example, see Figure 7.4 and Figure 7.5.

Lastly, **if the same information can be extracted from either structured data or from free text fields of a data set, usually structured data is preferred**. In most research and industry data sets, the quality of structured data fields is estimated to be quite high. Pre-defined value ranges and quality control at the point of data entry lead to high quality structured data. However, the entry of texts is free and usually no pre-defined value ranges and quality control exist. Thus, free texts are oftentimes full of spelling mistakes, grammatical errors and abbreviations (compare, e. g., [KM16] and [ZMZ16]). If an information is present in a structured field as well as in a free text field, the information from the structured field is used. Consequently, this information shall not be extracted from the free text field with text mining. Thus, during preprocessing, this information is removed. For example, in Figure 7.5, the word "steering" is removed from all free texts in the respective group.

As can be seen from Figure 7.5, in the hybrid approach, cluster names such as "fail" or "light" are added. After the grouping and removal step, these new cluster characteristics show up. Thus, compared to isolated approaches, the approach proposed in this chapter increases the amount of new information i_{new} available in the data set. More information on this can be found in the evaluation in Section 7.5.

7.4 Prototypical Implementation

The hybrid information extraction approach has been prototypically implemented in the course of a student project that has accompanied the work on this thesis [Lin17]. The prototype is open source and can be retrieved from GitHub[3]. It is implemented in Python, since many Python programming libraries for natural language processing exist. The implementation is straightforward and helpful

[3] `https://github.com/LinkMarco/PrototypeClustering`

documentations are available (e. g., [BKL09] and [Per14]). Furthermore, all tools and libraries chosen for the implementation of the prototype have industry-friendly licences. The prototypical implementation enables an easy integration of, e. g., new preprocessing methods, clustering algorithms and visualizations, as well as an easy adaptation to other languages. The prototype is designed in a flexible fashion, so that both use cases with English and German free text fields may be covered easily. Other design decisions, such as on data types the prototype can read and preprocessing performed, are founded on two use cases described in more detail in Section 7.5. In Figure 7.6, a schematic illustration of the prototype to the hybrid information extraction approach is shown. For the evaluation of the hybrid information extraction approach, a state-of-the-art clustering prototype is implemented as a baseline and a prototype for the hybrid approach. The two implementations are exactly the same, except for the two distinguishing processing steps "group" and "remove" (these two steps are bold-faced in Figure 7.6). In the following subsections the processing steps from Figure 7.6 are described in more detail and the implementation choices are stated.

For **reading** configurations, ConfigObj[4] is used. In the configuration, the user needs to state the column that contains the free texts and the column that contains the structured data that shall be used in the grouping and removal steps. If more than one structured categorical field is available and suitable, both may be applied to the "removal" step. However, the use cases only require to select exactly one structured field for the "grouping" step. Moreover, the processing steps can be freely defined by the user, or alternatively the default settings are used. With the default values, the prototype uses standard preprocessors, no synonyms in normalization, a tf-idf vectorizer and creates 12 clusters per group. The user may adapt these values if needed. The prototype can read ODBC databases as well as CSV-formatted data sets. Therefore, the library PyODBC[5] and a CSV-standard tool in Python[6] are employed. NumPy[7] arrays represent the incoming and outgoing data.

The **grouping** step is based on Python standard tools and SQL SELECT statements which are invoked from Python. All following steps are subsequently executed for each group. The free texts are grouped based on the structured

[4] `https://pypi.python.org/pypi/configobj/5.0.6`
[5] `https://pypi.python.org/pypi/pyodbc/4.0.3`
[6] `https://docs.python.org/3/library/csv.html`
[7] `http://www.numpy.org/`

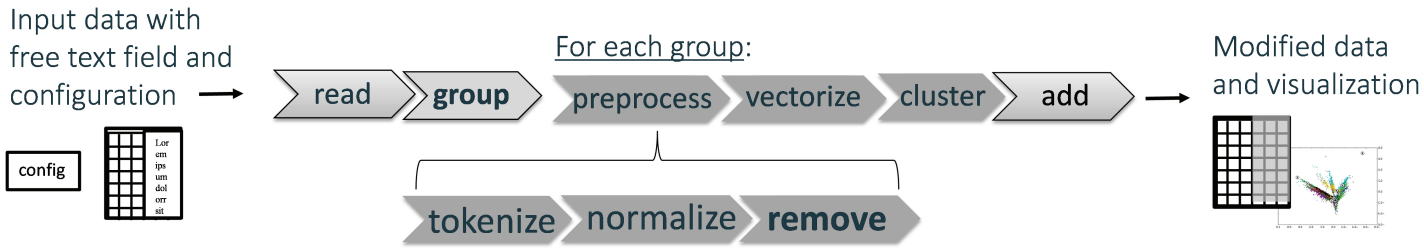


Figure 7.6: Schematic illustration of the prototype. In the two boldfaced preprocessing steps "group" and "remove", structured data is used, which makes the approach hybrid.

Table 7.2: Small example illustrating the preprocessing steps of the prototype: tokenization, normalization, and the removal of stopwords and information as determined via analysis of structured data.

Preprocessing step	Sample text
Before preprocessing	AIRBAGS FAILED DURING ACCIDENT, BUT CAR PARTS ARE NOT AVAILABLE.
Tokenize	[AIRBAGS] [FAILED] [DURING] [ACCIDENT] [,] [BUT] [CAR PARTS] [ARE] [NOT] [AVAILABLE] [.]
Normalize	[air bag] [fail] [during] [accident] [but] [car part] [are] [unavail]
Remove	[fail] [accident] [car part] [unavail]

data column as defined by the user in the configuration. For a concrete example of this processing step, see step (1) in Figure 7.4.

The **preprocessing phase** is a central part in text mining, since here the features are given by the words of the text (cf., e. g., [MRS08]). Table 7.2 shows the main steps.

The detailed settings are flexible and can be determined in the configuration. The main library used for natural language processing is the Natural Language Toolkit (NLTK)[8].

For **tokenization**, the Whitespace Tokenizer[9] or the Penn Treebank Tokenizer[10] may be applied. In tokenization, the text is split into the smallest meaningful units such as words and compounds. An example is given in Table 7.2. Note that here the compound "car parts", while being separated by a whitespace, was correctly selected as one token.

Then, all tokens are **normalized** based on plenty normalization methods which are provided in the prototypical implementation. These are optional and may be selected and adapted by the analyst for each use case, e. g., as described in Sections 7.5.2 and 7.5.3. In the normalization process, all texts may be

[8] `https://www.nltk.org`
[9] `http://www.nltk.org/_modules/nltk/tokenize/regexp.html#WhitespaceTokenizer`
[10] `http://www.nltk.org/_modules/nltk/tokenize/treebank.html#TreebankWordTokenizer`

lowercased. Spelling mistakes may be corrected using TextBlob[11]. Furthermore, contractions such as "didn't" may be extracted. The multi-word expression tokenizer from NLTK[12] is used for extraction. Then, white spaces may be normalized (two or more whitespaces are reduced to one). Moreover, URLs, mail adresses, telephone numbers, numbers, punctuation marks, currency symbols and accents may be removed using tools from Textacy[13]. Finally, the tokens may be stemmed, i. e., affixes are deleted with the goal of normalizing and thereby grouping the tokens. For stemming, the Porter Stemmer[14] from NLTK is applied. Additionally, different expressions which have the same meaning (=synonyms) may be consolidated. The synonym consolidation is implemented based on standard tools in Python. In the small example in Table 7.2, all words are lowercased. Then "NOT AVAILABLE" is normalized to its synonym "unavailable". Finally the tokens are stemmed, which, e. g., transforms "unavail*able*" to "unavail" and "fail*ed*" to "fail".

The **removal step** is based on the Word List Corpus Reader[15] in NLTK. In this step, stopwords are removed. Stopwords are words that do not bear interesting information, but merely are present in the texts for grammatical reasons. In Table 7.2, *during, but* and *are* were identified as stopwords. Additionally the information already present in the structured data column specified by the user is removed. In the prototypical implementation, the string from the categorical structured field is added to the stopword list. Since these structured values usually are words in their base form, they match with the corresponding word occurrences in the stemmed free text fields. Depending on the use case and data set, further resolutions of synonyms and abbreviations need to be added, which are also supported by the prototype. For a concrete example of the "remove" processing step, see Figure 7.5 step (2). In the small example in Table 7.2, the word "air bag" was additionally removed.

The machine learning library Scikit-learn[16] is used for **vectorizing and clustering** the free text fields. Vectorization means building a representation for each document that notes which words are present in the document and how these shall be weighted (cf. Section 2.2.2). The **vectorizer** can use two different

[11] `https://pypi.python.org/pypi/textblob`
[12] `http://www.nltk.org/_modules/nltk/tokenize/mwe.html#MWETokenizer`
[13] `https://pypi.python.org/pypi/textacy`
[14] `http://www.nltk.org/_modules/nltk/stem/porter.html#PorterStemmer`
[15] `http://www.nltk.org/_modules/nltk/corpus/reader/wordlist.html`
[16] `https://scikit-learn.org/stable/`

weighting schemes: Either plain term frequencies (tf) or term frequencies times inverse document frequency (tf-idf). Tf-idf is a weighting scheme often used in information retrieval [MRS08], which leads to a proper baseline clustering prototype. Here, terms which are very frequent in the complete free text collection are downweighted and rare ones are upweighted. Thus, much irrelevant information is already downweighted by means of the state-of-the-art weighting scheme. Thus, the state-of-the-art approach already extracts much new information and is a strong baseline. As will be shown in Section 7.5, i_{new} still is higher for the hybrid information extraction approach than for that baseline. Several **clustering** algorithms are implemented in Scikit-learn. For its popularity and robustness, the k-means algorithm is chosen for the prototype. This algorithm is a hard partitioning clustering algorithm, which means that each free text may only be put into exactly one of the clusters built. K-means is implemented following Lloyd's algorithm [Llo06]. NumPy is used for array representations and calculations in Scikit-learn. Thus, vectorization and clustering is fast. The prototype is built so that it is possible to calculate and compare i_{new} in the evaluation of the core concept. While a robust and state-of-the-art clustering algorithm is employed in the prototype, the concept is independent of the implementation chosen. Since the clustering step in the prototype is based on the Scikit-learn library, it is easy to add other clustering algorithms if needed.

Finally, **a new column is added** to the original data set. For each data instance it contains the name of the cluster to which the data instance belongs. The cluster name is based on the most frequent word in the cluster. This processing step uses NumPy arrays[17] and pyodbc[18].

Visualizations are optional. They were implemented using Matplotlib[19]. The clusters as well as cluster quality metrics such as the silhouette coefficient may be visualized.

[17] `https://numpy.org/`
[18] `https://pypi.org/project/pyodbc/`
[19] `http://matplotlib.org/`

7.5 Evaluation of the Hybrid Information Extraction Approach

In the following subsections, details on two data sets from the product life cycle are given and the application of the prototype to them is explained. Furthermore, the results of the hybrid approach are compared with the results from isolated approaches. In Figure 7.1 these results are illustrated as R_1, R_2 and R_3, where R_1 is the result of an isolated approach on structured data, R_2 is the result of an isolated approach on unstructured data, and R_3 is the result of the hybrid approach. For easy reference, the prototypes used in the experiments are denoted by R_1, R_2 and R_3 respectively. In the following, the three prototypes tested are defined:

- R_1: The **isolated approach on structured data** is based on a simple SQL-query which is run on the databases. It is exemplified for the categorical structured data field "component" in the NHTSA data set in the following SQL statement[20]. It simply groups the data values for the structured data field "component", counts the lines, and orders the result in descending order:

  ```
  SELECT component, count(*) FROM nhtsa_table
  GROUP BY component ORDER BY count(*) DESC;
  ```

- R_2: The implementation of the **isolated approach on unstructured data** base on the prototype described in Section 7.4. In fact, it is exactly the same, but the grouping as well as the removal of information from free texts based on the structured data field are omitted. Standard stopwords, such as *and, the, it* are still removed. This ensures that the effect of the steps special to the hybrid approach can be viewed in isolation.
- R_3: The implementation of the **hybrid information extraction prototype** is described in the previous section. It is adapted to the two use cases, i. e., as described below tailored configurations such as synonyms and preprocessors are defined.

By applying both R_2 and R_3 to the data sets, a state-of-the-art baseline clustering approach (R_2) can be compared with the hybrid approach, i. e., , the very same approach plus the two distinguishing steps "grouping" and "removing"

[20]Note: the "component" column is named "compdesc" in the original NHTSA data set.

(R_3). Both R_1 and R_2 are oriented on the components, i. e., air bag, steering, power train, whereas R_3 mostly is oriented on issues, i. e., fail, noise, (unintended) acceleration, unavailable (car parts). In the presentation of the detailed evaluation results, for comparison the results R_1 representing component-based and R_3 representing issue-based results are shown. Moreover, the difference between all three approaches R_1, R_2 and R_3 is reported in terms of the amount of new information i_{new} (as defined in Formula 7.1 in the introduction to this chapter) and the resulting clusters are discussed.

7.5.1 Data Sets

To evaluate the prototype, it is applied to two data sets. The first one is a freely accessible data set from the National Highway Traffic Safety Administration (NHTSA) in the United States of America. A respective application scenario is described in Section 3.2. Since the data set[21] as well as the prototype are freely accessible, all evaluation results with respect to the NHTSA data set are reproducible. The NHTSA complaint data set currently contains more than 1.3 million reports on incidents with cars.

As second data set the "industry" data set as already described in Section 5.3.2 is employed. It contains 153k entries, comprises information on downtimes in a production line and contains German free text information. On that line, smaller, but complex parts of a car are manufactured. This data set allows us to see how the prototype may be applied to a use case in production. It contains structured information on the downtimes, such as error codes and the duration of a downtime in seconds. Furthermore, information on the reasons for downtimes and the actions that were taken to put the production line running again are noted in a free text field. The workers can fill the free text field via text entry into a tablet, directly on the shop floor. The text entries are in German and quite short (4.1 words per entry in average), full of spelling mistakes and domain-specific abbreviations, which brings special challenges with respect to information extraction (cf. Section 5.3.3). A respective application scenario is described in Section 3.1.

[21] `https://www-odi.nhtsa.dot.gov/downloads/`, cf. Section 7.5.1

7.5.2 Data Analysis of the NHTSA Data Set

To apply the prototype resulting in R_3 to the NHTSA data set, at least the column that contains the structured field and the column which contains the free text field need to be given. The structured field with the affected car component is used in the grouping and removal steps. The default settings are applied (cf. Section 7.4), but some synonyms and context synonyms are added. For example, all occurrences of "not" or "failed" and "deploy" with no more than 3 words in between are normalized to "not deploy". Thereby, different ways of expressing the same concept are normalized to one term. The added synonym consolidations help in building semantically reasonable clusters. In this use case they have no influence on the "removal" step of the core method. From this hybrid prototype adapted to the use case, the isolated prototype for R_2 is deduced. Both approaches are the same but for the grouping and removing steps and generate the same number of clusters (12 per group). R_1 is computed as defined above.

In the first analysis, a focus is put on how an isolated approach on structured data only (resulting in R_1) and the hybrid approach (resulting in R_3) differ. In Figure 7.7, the five most frequent car components (R_1, left) and cluster terms (R_3, right) are shown. For easier readability and comprehensibility, clusters are named by the most frequent word in that cluster in the base form. From the structured data, the information on the most frequent car components affected is extracted. For instance, the electrical system, air bags, power trains of automatic transmissions, steering and power train. Using the hybrid approach (R_3), additional information can be extracted: It is already known from structured data that problems with transmission are frequent. But the analyst did not know that many complaints are about problems with noise, acceleration, unavailable (car parts) and problems where the car stalls. Accelerating the supply with car parts needed due to a recall (cluster "unavailable") and investigating problems, where the car stalls (cluster "stall") or accelerates (cluster "acceleration") might help in preventing car crashes. The result R_3 completes the information already available. The user can use both in his analysis: structured as well as generated cluster information. In Table 7.3 also both are noted, information from R_3 and R_1 (the latter is noted in brackets following the free text samples).

In a last experiment with the NHTSA data, it is checked how R_1, R_2 and R_3 differ in terms of i_{new} (see Formula 7.1). In R_1 no new information which

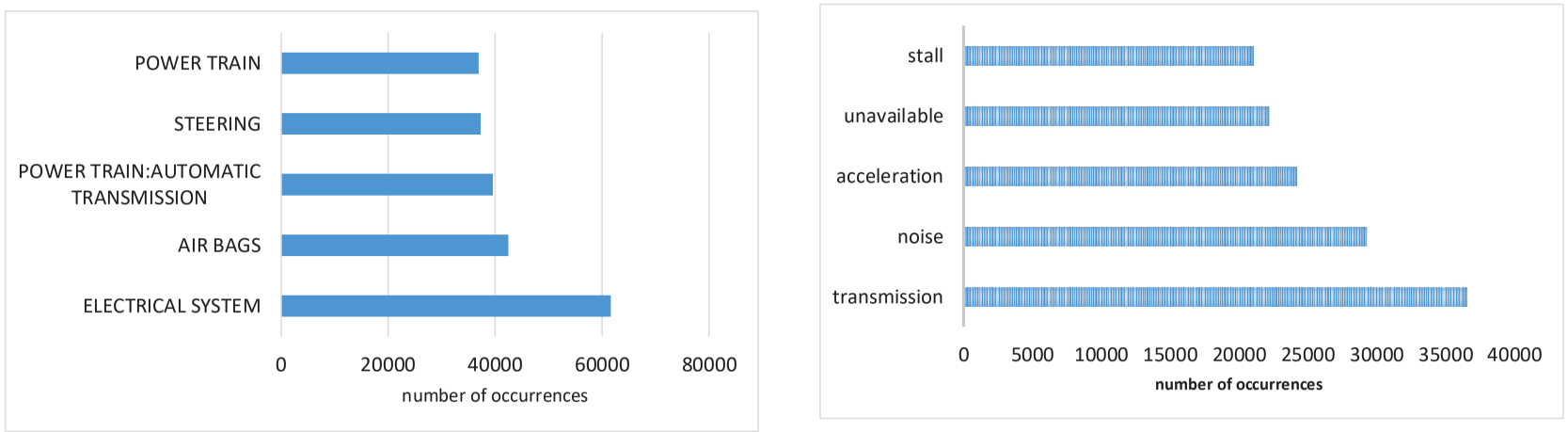


Figure 7.7: Most frequent car problems based on an isolated analysis of structured data only (R_1, left) and on the hybrid analysis (R_3, right), all based on the NHTSA data source.

Table 7.3: Concrete examples of NHTSA complaints for most frequent clusters together with structured information as noted in the categorical field 'component' given in parenthesis.

Cluster	Sample complaints
stall	VEHICLE STALLED DUE TO AN ELECTRICAL PROBLEM.('component': 'electrical system'); ENGINE STALLS WHEN APPROACHING A STOP. ('component': 'power train')
unavailable	THE PART TO DO THE REPAIR WAS UNAVAILABLE. (...) ('component': 'air bag'); (...) PARTS FOR RECALL IS NOT AVAILABLE SINCE REQUESTING (4) WEEKS AGO. (..) ('component': 'child seat')
acceleration	VEHICLE ACCELERATED BY ITSELF (...) ('component': 'vehicle speed control'); THE VEHICLE IS NOT ACCELERATING PROPERLY. (...) ('component': 'power train:automatic transmission')
noise	IT MAKES A LOUD NOISE AND NO 1 KNOW WHAT IT IS ('component': 'electrical system');WHINING NOISE WHEN TURNING STEERING WHEEL. (...) ('component': 'steering')
transmission	TRANSMISSION FAILURE AT 105,000 MILES (...)('component': 'power train:automatic transmission')

Table 7.4: Comparison of the degree of new information i_{new} for the three approaches R_1, R_2 and R_3 on the NHTSA data set.

Result set	i_{new}
R_1 (only structured information)	0
R_2 (baseline)	0.55
R_3 (hybrid approach)	0.98

goes beyond the structured information may be gathered, thus $i_{new} = 0$. The degree of new information within the 100 biggest clusters is compared in R_2 and R_3 respectively. The difference between R_1, R_2 and R_3 is illustrated with respect to this measure in Table 7.4. Here it can be seen that R_3 contains significantly more new information than R_2. The new information might be crucial for manufacturers, e. g., to get better customer satisfaction. Thus, the results of Ghazizadeh et al. [GML14] could be confirmed and improved, who report half of the cluster names to correspond to vehicle components, which corresponds to an i_{new} value of 0.5.

7.5.3 Data Analysis of the Industry Data Set

For a second evaluation, the prototype is applied to an industry data set with a German free text field. It contains information on causes and actions related to machine downtimes. A structured field with an error code description which indicates the group of errors of a downtime on the production line is used for grouping. The possible choices for filling the structured data field do not cover all types of errors and reasons for downtimes that may occur in reality. More choices are indicated in a free text field and may be deduced by the hybrid information extraction method, only.

To apply the prototype resulting in R_3 to the industry data set, details in preprocessing need to be changed in the configuration file due to German text: a German standard stopword list (from NLTK, cf. Section 7.4) and a German stemmer[22] need to be specified. Some spelling mistakes and verbalizations are normalized and a few synonyms and context synonyms are added. For example, if the German words "strom" and "leerlaufstrom" (both terms are on power/electricity) are mentioned near "hoch" (in English: "high") the main word is substituted by "strom_hoch"(in English: "power_high"). Some of the added synonyms help ensure a good recall of the "removal" step (cf. Section 2.2.2 for a definition of the recall metric). Also, encoded umlauts such as "ae" and "ue" are normalized to "ä" and "ü" respectively. After adapting the hybrid prototype to the use case, R_2 is deduced from it. Both approaches yield the same number of clusters and only differ in terms of the two distinguishing steps "group" and "remove". K in k-means is set as described above for the NHTSA data. For R_1,

[22] For instance, `http://www.nltk.org/_modules/nltk/stem/snowball.html#GermanStemmer`

the data is grouped by error code description with a SQL statement similar to the one described in the introduction of this section. For reasons of confidentiality, high-level abstractions of the German definitions are used. They furthermore are translated to English for the examples given in this work.

Finally, the results of an isolated approach on structured data only (R_1) are compared with the results of the hybrid approach (R_3). Using an isolated approach on structured data, the reasons for downtimes may be analyzed using the structured field with an error code and a table that defines these error codes. Following the hybrid approach, many fine-grained clusters are gained that represent new information. The results are illustrated in Figure 7.8. The structured information on errors on the production line are very coarse-grained: problems due to faulty bought-in parts, mounting processes in general, change of tools and change of calibrations often lead to downtimes. The most frequent error code hints at "miscellaneous" problems. This group is very big since many reasons for downtimes are not reflected in the given structured data values. Therefore, the workers on the shop floor oftentimes chose the structured value "miscellaneous" instead.

This is not helpful for the workers on the shop floor, who want to prevent or fix problems with downtimes of a production line. In contrast, the clusters resulting by the hybrid approach (R_3) are much more fine-grained. For example, in the big structured group "miscellaneous", clusters such as "problems with component parts" and "problems with out-of-commission machines" were found. These give more detailed information to workers on the shop floor and to managers of the production line. From R_3, more detailed and new information on reasons for downtimes may be extracted: From biggest to smaller clusters, detailed information on problems with checks, parts of the products which are produced, repeated checks and parameter deviations that lead to downtimes in the production line are reflected in the clusters. If this data was prepared for the shop floor workers, it might help in solving new downtimes of the production line faster. In a feedback loop, new reasons for downtimes in the production line may be added to the list of error codes in order to strengthen the significance of the structured fields.

Moreover, the degree of new information in Table 7.5 is compared with the same method as described for the NHTSA data set in the previous section (see Formula 7.1). In this case the baseline is even stronger than for the NHTSA data set, due to weaker structured information available in the industry data. Still, R_3 has a 0.21 higher i_{new} value than R_2. The grouping step helps to

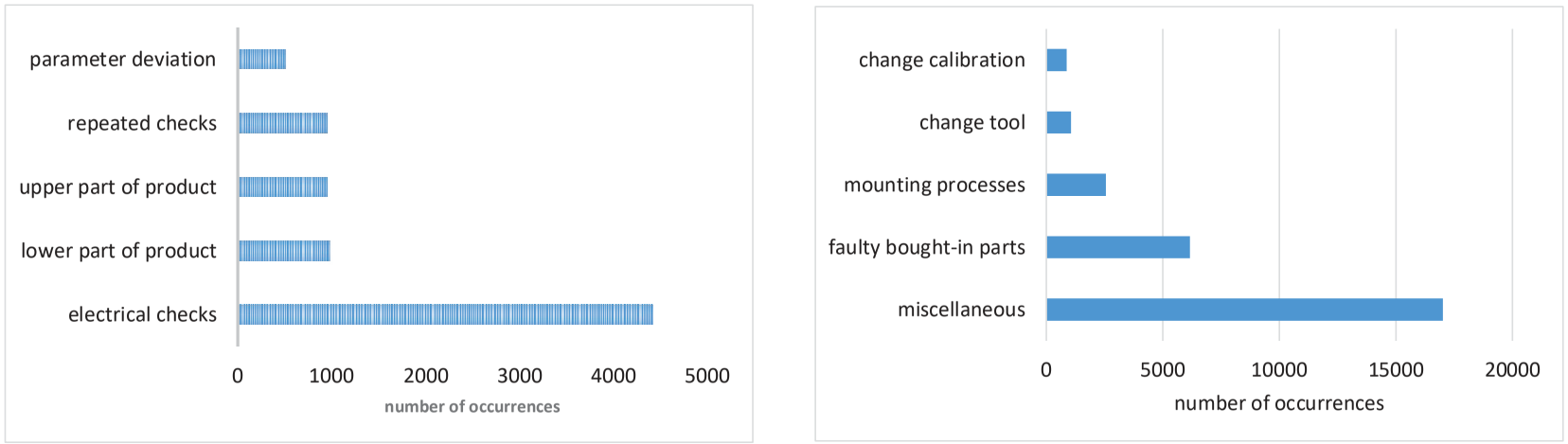


Figure 7.8: Most frequent reasons for downtimes based on the structured data field containing an error code (R_1, left) and based on the hybrid approach (R_3, right), all based on the industry partner data source.

Table 7.5: Comparison of the degree of new information i_{new} for the three approaches R_1, R_2 and R_3 and the industry data set.

Result set	i_{new}
R_1 (only structured information)	0
R_2 (baseline)	0.78
R_3 (hybrid approach)	0.99

reduce the number and size of big "miscellaneous" clusters. Special attention needs to be payed to synonyms, spelling mistakes and abbreviations. If not addressed properly, these issues may lead to less exploitation of the benefits of the removing step. The different results were discussed with domain experts and it was found that the additional information contained in R_3 is relevant to the task of optimizing the process. Also, from the point of view of domain experts, the hybrid information extraction method has future potential: Before a new shift on the production line begins, a summary of current insights may be presented to the shop floor workers. Moreover, new staff could be assisted by a presentation of aggregated insights.

7.6 Summary and Future Work

A hybrid approach to the extraction of information from free text fields was suggested, which yields more new information i_{new} from data with structured data fields and unstructured free text fields. The approach is based on natural language processing and k-means clustering and improves the results of two baseline isolated approaches by employing analytical results from structured data within the text analysis process. First, data is grouped based on a structured data field. Then, data is preprocessed, while also already seen information, as determined via a structured data field, is removed. Then, data is clustered and a new column containing the cluster name is added to the data set. In this chapter, the concept and a prototypical implementation of the approach for hybrid information extraction is presented. The prototype for the two baselines as well as the hybrid information extraction approach is based on free, open-source tools. The prototype is freely available on GitHub[23]. The prototype for R_2

[23] `https://github.com/LinkMarco/PrototypeClustering`

can be deviated from it and R_1 can be gathered by means of a SQL-query as shown in Section 7.5. Finally, the approach suggested is evaluated with two example data sets with German and English free text fields. While the data set from production is confidential, the NHTSA data set may be downloaded[24] and thus the results presented with respect to this use case are reproducible. The approach was compared to baseline isolated approaches on structured or unstructured data. It was shown that isolated approaches to free text yield much information already available in structured data. The hybrid approach impedes this and yields more new information. For the two use cases, the degree of new information in R_3 is significantly higher than in R_2.

In future work, the concept will be applied to further information extraction approaches. Moreover, an efficient handling of big data may be enabled by transferring the prototype into text databases (e. g., [KLA15]). Furthermore, additional evaluation metrics will be employed, e. g., based on entropy.

[24] `https://www-odi.nhtsa.dot.gov/downloads/`

Chapter 8

Conclusion and Future Work

The main contribution of this thesis is the presentation of the QUALM concept and concrete data quality methods therein, which assess and improve the quality of textual data in text analysis pipelines. Especially domain experts want to build analysis pipelines from scratch and oftentimes are not IT or data analysis experts, but solely experts in their domain. To ensure high quality analysis results of such analysis, a holistic approach to data quality measurement and improvement for each step in the analysis pipeline is needed. To this end, the QUALM concept is suggested. It is based on repositories to capture knowledge relevant for analysis, such as semantic resources, training data and features.

Moreover, concrete data quality methods are central elements of QUALM. The QUALM indicators assess the quality of the input data as well as the fit of input data and analysis tools. For example, they determine the percentage of abbreviations in a text and the fit of the input data and the training data employed by the analysis tool. To each indicator, corresponding methods exist which aim at improving data quality. These are called the QUALM modifiers in the approach suggested. For example, methods to automatically suggest well-fitting training data and semantic resources, to dissolve abbreviations and to yield new valuable information from structured data sets enriched by free text fields are provided in QUALM.

In the following, Section 8.1 summarizes concrete contributions offered by the proposed concepts and methods. Afterwards, Section 8.2 discusses possible future work.

8.1 Summary of the Contributions

Chapter 4 describes the QUALM concept for continuous and holistic data quality measurement and improvement within data analysis pipelines and discusses design decisions for a prototypical implementation (cf. Kiefer et al. [KRM19b]). Firstly, the QUALity Mining (QUALM) concept is presented. The goal of QUALM is it to prevent low-quality analysis results, as they oftentimes appear especially in domain-specific analysis pipelines (cf. Kiefer [Kie17]). Thus, it operates on single analysis tools as well as on whole chains of analysis tools, i. e., analysis pipelines. Within such pipelines many data quality problems may arise, especially when citizen data scientists build analysis pipelines from scratch. For example, non-fitting training data such as news are employed within analysis of industry data, relevant semantic resources such as dictionaries of abbreviations and domain-specific taxonomies are not exploited and data characteristics expected by the analysis tools such as correct upper and lowercasing of words are not met by the input data. Data quality assessment and improvement in QUALM base on an extension of the existing definition of data quality as "fitness for use by data consumers" [WS96].

Besides human end consumers, QUALM also considers algorithmic data consumers [Kie16]. These algorithmic consumers may especially also be analysis tools in analysis pipelines. For example, speech-to-text tools, optical character recognizers, gene classifiers, image classifiers and web analysis tools may be employed. Moreover, especially also analysis tools in text analysis pipelines may be such data consumers, e. g., language identifier, part-of-speech tagger, named entity recognizer and sentiment analyzer. The fitness for use, i. e., data quality, is assessed with respect to the input data and the specifics of analysis tools. To this end, new data quality methods were developed, which are addressed in the next paragraph. The specifics of analysis tools are stored in respective repositories and comprise the training data, semantic resources and features employed by those. For example, labeled news texts may be employed as training data by a named entity recognizer, it may employ semantic resources, such as gazetteer lists, dictionaries, taxonomies or ontologies and further relies on features and thus may, e.g., give higher weights to uppercased terms in comparison to lowercased ones. With QUALM, a concept which employs all this knowledge in assessing and improving data quality in each step of analysis pipelines is presented. Thus, based on QUALM, the challenging task of continuously measuring and improving data

quality within data analysis pipelines may be conquered and a holistic approach is suggested (cf. Section 1.2.1). Since QUALM operates within analysis pipelines, for a prototypical implementation it is integrated into the data processing and analysis toolkit FlexMash [HB16], which was developed at the University of Stuttgart. The concrete QUALM methods are implemented as web services and may thus be flexibly integrated into further analysis toolkits oftentimes employed by citizen data scientists, e.g. RapidMiner and SPSS (cf. Section 2.2.3).

Chapter 5 introduces concrete QUALM data quality methods. First, an overview of QUALM methods is presented (cf. Table 5.1). Two types of QUALM methods exist. QUALM indicators assess certain characteristics of input data and analysis tools. Each indicator comes with a corresponding modifier, which changes characteristics of data or analysis tools and thus aims at increasing the fitness for use of data by analysis tools, i. e., data quality. In difference to evaluation methods, such as accuracy, which require at least parts of the input data to be gold labeled manually, the data quality indicators can be assessed without need for gold labels. Moreover, evaluation results are presented, which show for various data sets reaching from news over prose to chat, tweets and industry data and for various implementations of POS-tagger and language identifier analysis tools, that QUALM data quality indicators and accuracy correlate (cf. Kiefer [Kie19]). Thus, QUALM indicators are a solid basis to conquer the challenge arising by the uncertainty with respect to analysis result quality for unlabeled text data (cf. Section 1.2.2). Moreover, the effect of QUALM modifiers with respect to the accuracy of such analysis tools is investigated and the effect of QUALM on a whole chain of analysis tools, i. e., on an analysis pipeline, is discussed. With QUALM, accuracy of analysis tools such as POS-tagger and language identifier was found to rise and moreover the quality of insights derived from analytics and the respective domain-specific analysis pipelines increases (cf. Kiefer et al. [KRM19b]).

In *Chapter 6*, two central data quality methods within QUALM are introduced and assessed in more detail, namely an indicator which assesses the fit of training data (FiT) and a corresponding modifier method, which automatically selects the best-fitting training data from a repository (SeT) (cf. Kiefer et al. [KRM20]). The FiT and SeT methods operate within analysis pipelines which are made up of supervised machine learning tools. A prototypical implementation of the concepts for the case of textual data is presented and they are evaluated with respect to several text data sets in the context of text analysis pipelines. The

evaluation results show that the challenge arising by a negligence of the selection of appropriate training data by citizen data scientists may be conquered by means of the suggested method (cf. Section 1.2.3). Citizen data scientist may build analysis pipelines from scratch and need not worry about training data - since problems are indicated to them by means of the FiT indicator and moreover, with SeT best-fitting training data may be selected automatically.

Finally, *Chapter 7* deals with the exploitation of structured and unstructured information sources in information extraction (cf. Kiefer et al. [KRM19a]). Oftentimes, either structured or unstructured data only is considered in isolated information extraction approaches. If both data types are addressed, information extraction is still performed only in parallel, isolated approaches. Yet, these isolated approaches may lead to valuable information get lost. This challenge (cf. Section 1.2.4) is addressed by means of a hybrid information extraction concept which exploits structured data within the text analysis process. The suggested concept considers structured data enriched by free text fields as basis. To exploit both of these information sources, two main changes within standard text clustering processes are suggested: (1) an employment of groups present in structured data fields for a first grouping of free text fields and (2) the deletion of information which is already known from the structured data fields from the free text fields. Based on a prototypical implementation of the concept it is evaluated for two industry-near application scenarios and the respective data sets (cf. Section 7.5). Qualitatively as well as assessed by means of a straightforward metric, which assesses the amount of new information within clustering results, huge benefits of hybrid information extraction can be shown in comparison to an isolated approach on structured data only and with respect to an isolated approach on unstructured text data only.

Altogether, the concepts and methods proposed by this thesis fill several gaps in current research regarding the assessment and improvement of data quality within data analysis pipelines. Especially, with respect to domain-specific analysis pipelines built by citizen data scientists the suggested QUALM concept and methods therein prevent low-quality analysis results. In particular, concrete QUALM methods with respect to text data and text analysis tools, help to improve the quality of domain-specific text analysis performed by IT-inept citizen data scientists without data science/text mining skills. The positive effect of QUALM and the methods therein has also been confirmed by the results of the profound evaluations of individual proposed concepts and methods.

8.2 Future Work

The presented thesis and its contributions summarized in Section 8.1 constitute a good basis for future research opportunities.

Firstly, by an application of QUALM and the methods therein to further domain-specific application scenarios, data sets, analysis tools and pipelines, more information may be collected with respect to the correlation of data quality indicators and accuracy. If corresponding gold annotations/labels are added to the data, further investigations with respect to the indicators and the accuracy of intermediate and end results of whole analysis pipelines could be an interesting field of research. For example, interesting experiments could focus on how the improvement of data quality as perceived by intermediate consumers influences data quality from a rather end consumer viewpoint.

These insights may furthermore be exploited and may be the basis to further improve the methods for data quality assessment and improvement. For example, thresholds with respect to certain data characteristics and analysis tools may be refined. To this end, an integration of QUALM into further analysis toolkits such as RapidMiner and SPSS would be a good basis for the collection of domain-, data- and tool-specific insights. Moreover, aspects in how to visualize data quality within the GUI of such a data analysis tool needs to be investigated further.

With respect to the concrete QUALM methods, the transfer of indicators to consistent data quality metrics and a combination and weighting of several indicators may be addressed.

In this thesis, textual data is focused. Yet, also an application of the QUALM concept to further data types such as images and speech might be addressed in future work. Then, e. g., besides text similarity metrics as suggested in this thesis for the automatic training data selection within text analysis pipelines, analog metrics for determining, e. g., the similarity of images and speech may be employed. Finally, also the application to new practical fields such as in teaching may be investigated.

Moreover, the hybrid approach to exploiting structured data within text mining, may be transferred to further analysis tools besides text clusterer, e.g., the concept may be applied to text classification as well. Furthermore, additional evaluation metrics, e. g., based on entropy, may be investigated.

Thereby, all this offers a great potential to enhance text and data quality within data analysis pipelines and, based on this, also the quality of analysis results.

Author Publications

- Cornelia Kiefer, Peter Reimann, and Bernhard Mitschang. Prevent Low-Quality Analytics by Automatic Selection of the Best-Fitting Training Data. In *Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS)*, Maui, Hawaii, USA, 2020.
- Cornelia Kiefer, Peter Reimann, and Bernhard Mitschang. QUALM: Ganzheitliche Messung und Verbesserung der Datenqualität in der Textanalyse. *Datenbank-Spektrum*, 19(2):137–148, 2019.
- Cornelia Kiefer. Quality Indicators for Text Data. In Meyer, Holger, et al., editor, *Datenbanksysteme für Business, Technologie und Web (BTW 2019), 18. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme (DBIS), 4.-8. März 2019, Rostock, Germany, Workshopband*, volume P-290 of *LNI*, pages 145–154. Gesellschaft für Informatik, Bonn, 2019.
- Cornelia Kiefer, Peter Reimann, and Bernhard Mitschang. A Hybrid Information Extraction Approach Exploiting Structured Data Within a Text Mining Process. In Grust, Torsten, et al., editor, *Datenbanksysteme für Business, Technologie und Web (BTW 2019), 18. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme (DBIS), 4.-8. März 2019, Rostock, Germany, Proceedings*, volume P-289 of *LNI*, pages 149–168. Gesellschaft für Informatik, Bonn, 2019.
- Cornelia Kiefer. Die Gratwanderung zwischen qualitativ hochwertigen und einfach zu erstellenden domänenspezifischen Textanalysen. In B. Mitschang et al., editor, *GI-Edition Lecture Notes in Informatics Datenbanksysteme für Business, Technologie und Web (BTW 2017) Workshopband*, Stuttgart, Germany, 2017.
- Cornelia Kiefer. Assessing the Quality of Unstructured Data: An Initial Overview. In Ralf Krestel, Davide Mottin, and Emmanuel Müller, editors,

Proceedings of the LWDA, CEUR Workshop Proceedings, pages 62–73, Potsdam, Germany, 2016.

- Cornelia Kiefer and Ulrike Pado. Freitextaufgaben in Online-Tests – Bewertung und Bewertungsunterstützung. *HMD Praxis der Wirtschaftsinformatik*, 52(1):96–107, 2015.
- Laura Kassner, Christoph Gröger, Jan Königsberger, Eva Hoos, Cornelia Kiefer, Christian Weber, Stefan Silcher, and Bernhard Mitschang. The Stuttgart IT Architecture for Manufacturing. In *Lecture Notes in Business Information Processing*, volume 291, pages 53–80. Springer International Publishing, 2017.
- Christoph Gröger, Laura Kassner, Eva Hoos, Jan Königsberger, Cornelia Kiefer, Stefan Silcher, and Bernhard Mitschang. The Data-Driven Factory. Leveraging Big Industrial Data for Agile, Learning and Human-Centric Manufacturing. In Slimane Hammoudi, Leszek Maciaszek, Michele M. Missikoff, Olivier Camp, and Jose Cordeiro, editors, *Proceedings of the 18th International Conference on Enterprise Information Systems*, pages 40–52, Rome, Italy, 2016. SciTePress.
- Laura Kassner and Cornelia Kiefer. Taxonomy Transfer: Adapting a Knowledge Representing Resource to new Domains and Tasks. In *Proceedings of the 16th European Conference on Knowledge Management*, pages 399–407, Udine, Italy, 2015.
- Ulrike Pado and Cornelia Kiefer. Short Answer Grading: When Sorting Helps and When it Doesn't. In Linköpings universitet Linköping University Electronic Press, editor, *Proceedings of the 4th workshop on NLP for Computer Assisted Language Learning, NODALIDA 2015*, Linköping Electronic Conference Proceedings, pages 42–50, Wilna, Lithuania, 2015. LiU Electronic Press and ACL Anthology.

Bibliography

[AHG11] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Edinburgh, Scotland, UK, 2011. Association for Computational Linguistics.

[AHJAJA15] Dua'a Al-Hajjar, Nouf Jaafar, Manal Al-Jadaan, and Reem Alnutaifi. Framework for social media big data quality analysis. In Nick Bassiliades, Mirjana Ivanovic, Margita Kon-Popovska, Yannis Manolopoulos, Themis Palpanas, Goce Trajcevski, and Athena Vakali, editors, *New Trends in Database and Information Systems II*, volume 312 of *Advances in Intelligent Systems and Computing*, pages 301–314. Springer International Publishing, Cham, 2015.

[AHN19] Murugan Anandarajan, Chelsey Hill, and Thomas Nolan, editors. *Practical Text Analytics: Maximizing the Value of Text Data*. Springer International Publishing, Cham, 2019.

[BB12] Gerrit Reinier Botha and Etienne Barnard. Factors that Affect the Accuracy of Text-Based Language Identification. *Computer Speech & Language*, 26(5):307–320, 2012.

[BBCG11] Carlo Batini, Daniele Barone, Federico Cabitza, and Simone Grega. A data quality methodology for heterogeneous data. *International Journal of Database Management Systems (IJDMS)*, 3(1):60–79, 2011.

[BD10] Peter Buneman and Susan B. Davidson. *Data provenance – the foundation of data quality*, 2010. Available at: `https://pdfs.semanticscholar.org/9ec4/`

`275fed43df7145dec34cba9743a9186dc972.pdf`.

[Beh16] Michael Behringer. *Visual Analytics im Kontext der Daten- und Analysequalität am Beispiel von Data Mashups*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2016.

[BEP15] Chris Baillie, Peter Edwards, and Edoardo Pignotti. Qual: A provenance-aware quality model. *Journal of Data and Information Quality (JDIQ)*, 5(3):12, 2015.

[Bet17] Shreyas Bettadapura Raghavendra. *Relevance of the two adjusting screws in data analytics: data quality and optimization of algorithms*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[BHM17] Michael Behringer, Pascal Hirmer, and Bernhard Mitschang. Towards interactive data processing and analytics - putting the human in the center of the loop. In *International Conference on Enterprise Information Systems*, pages 87–96, Porto, Portugal, 2017.

[BHVh14] Thomas Bauernhansl, Michael ten Hompel, and Birgit Vogelheuser. *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien und Migration*. Springer Vieweg, Wiesbaden, 2014.

[Bis06] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

[BJB12] Aiswarya Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. Cost and Benefit of Using WordNet Senses for Sentiment Analysis. In *International conference on Language Resources and Evaluation (LREC)*, Turkey, Istanbul, 2012.

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. O'Reilly, Beijing and Cambridge [Mass.], 1st edition, 2009.

[BLR+10] Roberto Basili, Daniel Lopresti, Christoph Ringlestetter, Shourya Roy, Klaus U. Schulz, and L. Venkata Subramaniam. Summary of the 4th workshop on analytics for noisy unstructured text data (and). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages

1965–1966, Toronto, Candada, 2010. ACM.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, Madison, Wisconsin, USA, 1998. ACM.

[BM11] Roger Blake and Paul Mangiameli. The effects and interactions of data quality and problem complexity on classification. *J. Data and Information Quality*, 2(2):8:1–8:28, 2011.

[BMB13] BMBF. *Zukunftsbild Industrie 4.0: Broschüre, Bundesministerium für Bildung und Forschung (BMBF)*. Bonn, 2013 edition, 2013. Available at: `http://www.bmbf.de/pubRD/Zukunftsbild_Industrie_40.pdf`.

[Boe16] Bradley Boehmke. *Data wrangling with r*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[Bro08] Benjamin Brooks. Shifting the focus of strategic occupational injury prevention: Mining free-text, workers compensation claims data. *Safety Science*, 46(1):1–21, 2008.

[BS16] Carlo Batini and Monica Scannapieco. *Data and Information Quality*. Springer International Publishing, Cham, 2016.

[Buß08] Hadumod Bußmann. *Lexikon der Sprachwissenschaft*. Alfred Kröner Verlag, Stuttgart, Germany, 2008.

[CFCCP16] Ana G. Carretero, Alberto Freitas, Ricardo Cruz-Correia, and Mario Piattini. A case study on assessing the organizational maturity of data management, data quality management and data governance by means of mamd. In *ICIQ*, pages 75–84, Ciudad Real, Spain, 2016.

[CH14] B. Carter and M. Hofmann. An analysis into using unstructured non-expert text in the illicit drug domain. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 651–657, Gurgaon, India, 2014.

[Che17] Shalini Chellathurai Saroja. *Measurement of the quality of structured and unstructured data accumulating in the product life cycle in a data quality dashboard*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[CKK17] Yeounoh Chung, Sanjay Krishnan, and Tim Kraska. A data quality metric (dqm): How to estimate the number of undetected errors in data sets. *Proc. VLDB Endow.*, 10(10):1094–1105, 2017.

[CRB11] Rahul Chougule, Dnyanesh Rajpathak, and Pulak Bandyopadhyay. An integrated framework for effective service and repair in the automotive domain: An application of association mining and case-based-reasoning. *Computers in Industry*, 62(7):742–754, 2011.

[CSN+14] M. Chessell, F. Scheepers, N. Nguyen, R. van Kessel, and R. van der Starre. *Governing and Managing Big Data for Analytics and Decision Makers*, 2014. Available at: `http://www.redbooks.ibm.com/redpapers/pdfs/redp5120.pdf`.

[CT94] William B. Cavnar and John M. Trenkle. N-gram-based Text Categorization. *Ann Arbor MI*, pages 161–175, 1994.

[CV15] Francesco Camastra and Alessandro Vinciarelli. *Machine learning for audio, image and video analysis: Theory and applications*. Advanced Information and Knowledge Processing. Springer, London, second edition, 2015.

[CZ15] Li Cai and Yangyong Zhu. The challenges of data quality and data quality assessment in the big data era. *Data Science Journal*, 14(0):2, 2015.

[DBES09] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Integrating conflicting data. *Proceedings of the VLDB Endowment*, 2(1):550–561, 2009.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Computing Research Repository (CoRR)*, abs(1810.04805), 2018.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[Die17] Paul Dieterich. *Evaluierung verschiedener Technologien zur Speicherung und Verknüpfung von strukturierten und unstrukturierten*

Daten. Student thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[DJ03] Tamraparni Dasu and Theodore Johnson. *Exploratory data mining and data cleaning*. Wiley series in probability and statistics. Wiley-Interscience, New York, 2003.

[DK10] Debabrata Dey and Subodha Kumar. Reassessing data quality for information products. *Management Science*, 56(12):2316–2322, 2010.

[DLY08] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In Marc Najork, Andrei Z. Broder, and Soumen Chakrabarti, editors, *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pages 231–240. ACM, 2008.

[dMDS+14] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

[DRCB13] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, Hissar, Bulgaria, 2013. Association for Computational Linguistics.

[DTI13] Daniel Bär, Torsten Zesch, and Iryna Gurevych. DKPro Similarity: An Open Source Framework for Text Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 121–126, Sofia, Bulgaria, 2013.

[DWL15] Dejing Dou, Hao Wang, and Haishan Liu. Semantic data mining: A survey of ontology-based approaches. In *Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015)*, pages 244–251, Anaheim, CA, USA, 2015.

[ECD+14] Mark Gerald Easton, Geraldine Carrodus, Tim Delany, Bernie Howitt, Richard Smith, Helen Butler, and Kate Mcarthur. *Geography*. South Melbourne, Victoria : Oxford University Press, 2014.

[EP15] Florian Endel and Harald Piringer. Data Wrangling: Making Data Useful Again. *IFAC-PapersOnLine*, 48(1):111–112, 2015.

[FBOGA12] D. Fuentes, R. Bardeli, J. A. Ortega, and L. Gonzalez-Abril. A similarity measure between videos using alignment, graphical and speech features. *Expert Systems with Applications*, 39(11):10278–10282, 2012.

[Fei13] Christina Feilmayr. Decision guidance for optimizing web data quality - a recommendation model for completing information extraction results. *24th International Workshop on Database and Expert Systems Applications*, pages 113–117, 2013.

[FGL+16] Tim Furche, Georg Gottlob, Leonid Libkin, Giorgio Orsi, and Norman W. Paton. Data wrangling for big data: Challenges and opportunities. In *Advances in Database Technology — EDBT 2016*, Advances in Database Technology, pages 473–478, Bordeaux, France, 2016.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

[FH07] Owen Foley and Markus Helfert. The development of an objective metric for the accessibility dimension of data quality. In *Innovations in Information Technologies (IIT)*, pages 11–15, Dubai, United Arab Emirates, 2007.

[Fis35] R. A. Fisher. *The design of experiments*. Oliver & Boyd, Oxford, England, 1935.

[FK79] W. N. Francis and H. Kučera. *Manual of Information to Accompany A Standard Corpus of Present-day Edited American English, for Use with Digital Computers*. Brown University, Department of Lingustics, 1979.

[FKE+15] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2962–2970. Curran Associates, Inc, 2015.

[FKS06] George Forman, Evan Kirshenbaum, and Jaap Suermondt. Pragmatic text mining: Minimizing human effort to quantify many issues in call logs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 852–861, Philadelphia, USA, 2006. ACM.

[FLR94] Christopher Fox, Anany Levitin, and Thomas Redman. The notion of data and its quality dimensions. *Inf. Process. Manage.*, 30(1):9–19, 1994.

[FP18] Jernej Flisar and Vili Podgorelec. Document Enrichment Using DBPedia Ontology for Short Text Classification. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, WIMS '18, pages 8:1–8:9, Novi Sad, Serbia, 2018. ACM.

[FRI+16] Mina Farid, Alexandra Roatis, Ihab F. Ilyas, Hella-Franziska Hoffmann, and Xu Chu. Clams: Bringing quality to data lakes. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 2089–2092, San Francisco, USA, 2016. Association for Computing Machinery.

[FS07] Ronen Feldman and James Sanger. *The text mining handbook*. Cambridge University Press, New York, 2007.

[GACOR05] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Conference on Advances in Intelligent Data Analysis*, IDA'05, pages 121–132, Berlin, Heidelberg, 2005. Springer-Verlag.

[GF13] Wael H. Gomaa and Aly A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.

[GFL06] Simona Gandrabur, George Foster, and Guy Lapalme. Confidence estimation for nlp applications. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(3):1–29, 2006.

[GFM+13] Gonzalo Génova, José Miguel Fuentes, Juan Llorens Morillo, Omar Hurtado, and Valentin Moreno. A framework to measure and improve the quality of textual requirements. *Requir. Eng.*, 18(1):25–41, 2013.

[GGH+19] Corinna Giebler, Christoph Gröger, Eva Hoos, Holger Schwarz, and Bernhard Mitschang. Leveraging the data lake - current state and challenges. In *Proceedings of the 21st International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2019)*, Linz, Austria, 2019.

[GH07] Mouzhi Ge and Markus Helfert. A review of information quality research. In *The 12th international conference on information quality*, Cambridge, MA, USA, 2007.

[GKH+16] Christoph Gröger, Laura Kassner, Eva Hoos, Jan Königsberger, Cornelia Kiefer, Stefan Silcher, and Bernhard Mitschang. The Data-Driven Factory. Leveraging Big Industrial Data for Agile, Learning and Human-Centric Manufacturing. In Slimane Hammoudi, Leszek Maciaszek, Michele M. Missikoff, Olivier Camp, and Jose Cordeiro, editors, *Proceedings of the 18th International Conference on Enterprise Information Systems*, pages 40–52, Rome, Italy, 2016. SciTePress.

[GML14] Mahtab Ghazizadeh, Anthony D. McDonald, and John D. Lee. Text Mining to Decipher Free-Response Consumer Complaints: Insights From the NHTSA Vehicle Owner's Complaint Database. *Human Factors The Journal of the Human Factors and Ergonomics Society*, 6(56):1189–1203, 2014.

[Gol12] Matthew K. Gold. *Debates in the digital humanities*. U of Minnesota Press, 2012.

[Gol19] Robert Golda. *Integration von Datenqualitätsmethoden in das Datenflusswerkzeug FlexMash*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2019.

[GPM04] Asuncion Goméz-Perez and David Manzano Macho. An Overview of Methods and Tools for Ontology Learning from Texts. *The*

Knowledge Engineering Review, 19(3):187–212, 2004.

[Gra16] Raoul Graumann. *Passung von Trainingsdaten und Textdaten - Textdaten mit vielen Fehlern und Ungenauigkeiten anpassen.* Student thesis, University of Stuttgart, Stuttgart, Germany, 2016.

[Gra17] Raoul Graumann. *Untersuchungen zur Qualität und zur Strukturierung von Textdaten aus dem Produktlebenszyklus.* Student thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[Grö18] Christoph Gröger. Building an industry 4.0 analytics platform. practical challenges, approaches and future research directions. *Datenbank-Spektrum*, 2018.

[GS+06] Henning Griethe, Heidrun Schumann, et al. The visualization of uncertain data: Methods and problems. In *SimVis*, pages 143–156, 2006.

[GSB14] M. Gubanov, M. Stonebraker, and D. Bruckner. Text and structured data fusion in data tamer at scale. In *IEEE 30th International Conference on Data Engineering*, Chicago, USA, 2014.

[GSD07] Warwick Graco, Tatiana Semenova, and Eugene Dubossarsky. Toward knowledge-driven data mining. In *Proceedings of the 2007 International Workshop on Domain Driven Data Mining*, DDDM '07, pages 49–54, San Jose, California, USA, 2007. ACM.

[GSO+11] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 42–47, Portland, Oregon, 2011. Association for Computational Linguistics.

[Haw00] Dietmar Hawranek. *Grausame Todesfälle*, 37/2000 edition, 2000. Available at: `http://www.spiegel.de/spiegel/print/d-17322756.html`.

[HB16] Pascal Hirmer and Michael Behringer. FlexMash 2.0 - Flexible Modeling and Execution of Data Mashups. In *Rapid Mashup Challenge (RMC)*, Lugano, Switzerland, 2016.

[HB17] Oumaima Hourrane and El Habib Benlahmar. Survey of plagiarism detection approaches and big data techniques related to plagiarism candidate retrieval. In *Proceedings of the 2Nd International Conference on Big Data, Cloud and Applications*, BDCA'17, pages 15:1–15:6, Tetuan, Morocco, 2017. ACM.

[HBB13] Hussam Hamdan, Frederic Béchet, and Patrice Bellot. Experiments with DBpedia, WordNet and SentiWordNet as Resources for Sentiment Analysis in Micro-blogging. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 455–459, Atlanta, Georgia, USA, 2013. Association for Computational Linguistics.

[HCW15] Jingyu Han, Kejia Chen, and Jianing Wang. Web article quality ranking based on web community knowledge. *Computing*, 97(5):509–537, 2015.

[HDB17] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017.

[HHLRP12] Annika Hinze, Ralf Heese, Markus Luczak-Rösch, and Adrian Paschke. Semantic enrichment by non-experts: Usability of manual annotation tools. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2012*, pages 165–181, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Hil11] Knut Hildebrand, editor. *Daten- und Informationsqualität: Auf dem Weg zur Information Excellence*. Praxis. Vieweg + Teubner, Wiesbaden, 2., aktualisierte und erw. aufl. edition, 2011.

[HJ15] Karin Hartl and Olaf Jacob. Determing the business value of business intelligence with data mining methods. In *Data Analytics*, pages 87–91, Nice, France, 2015.

[HL04] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04,

pages 168–177, Seattle, WA, USA, 2004. ACM.

[HLK15] Gareth Herschel, Alexander Linden, and Lisa Kart. Magic quadrant for advanced analytics platforms. *Gartner Report G*, 270612, 2015.

[HS15] Mohammad Hossin and M. N. Sulaiman. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5:01–11, 2015.

[HSS03] A. Hotho, S. Staab, and G. Stumme. Ontologies Improve Text Document Clustering. In *Third IEEE International Conference on Data Mining*, pages 541–544, Melbourne, Florida, 2003.

[HSWL12] P. Han, S. Shen, D. Wang, and Y. Liu. The influence of word normalization in english document clustering. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 2, pages 116–120, Zhangjiajie, China, 2012.

[HW96] W. R. Hogan and M. M. Wagner. Free-text fields change the meaning of coded data. *Proceedings of the AMIA Annual Fall Symposium*, pages 517–521, 1996.

[IP17] Nancy Ide and James Pustejovsky. *Handbook of Linguistic Annotation*. Springer Netherlands, 2017.

[IPO15] Anne Immonen, Pekka Paakkonen, and Eila Ovaska. Evaluating the Quality of Social Media Data in Big Data Architecture. *IEEE Access*, 1(3), 2015.

[iso] iso. Iso number: 9000:2015, available at: `https://www.iso.org/standard/45481.html`.

[Jen08] Jennifer Widom. Trio: A system for data, uncertainty, and lineage. In *Managing and Mining Uncertain Data*. Springer, 2008.

[JL06] Nitin Jindal and Bing Liu. Identifying comparative sentences in text documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 244–251, Seattle, Washington, USA, 2006. ACM.

[JL08] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search*

and Data Mining, WSDM '08, pages 219–230, Stanford, USA, 2008. ACM.

[JLZ+19] Tommi Sakari Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782, 2019.

[JM09] Dan Jurafsky and James H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, Upper Saddle River, N.J. and London, 2nd edition, 2009.

[JMS10] Clement Jonquet, Mark A. Musen, and Nigam H. Shah. Building a Biomedical Ontology Recommender Web Service. *Journal of biomedical semantics*, 1 Suppl 1(Suppl 1):S1–S1, 2010.

[Jur88] J. M. Juran. *Juran on planning for quality*. Free Press and Collier Macmillan, New York, 1988.

[Kal12] Hans-Michael Kaltenbach, editor. *A Concise Guide to Statistics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[KGK+17] Laura Kassner, Christoph Gröger, Jan Königsberger, Eva Hoos, Cornelia Kiefer, Christian Weber, Stefan Silcher, and Bernhard Mitschang. The Stuttgart IT Architecture for Manufacturing. In *Lecture Notes in Business Information Processing*, volume 291, pages 53–80. Springer International Publishing, 2017.

[KHP+11] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research Directions in Data Wrangling: Visualizations and Transformations for Usable and Credible Data. *Information Visualization*, 10(4):271–288, 2011.

[Kie16] Cornelia Kiefer. Assessing the Quality of Unstructured Data: An Initial Overview. In Ralf Krestel, Davide Mottin, and Emmanuel Müller, editors, *Proceedings of the LWDA*, CEUR Workshop Proceedings, pages 62–73, Potsdam, Germany, 2016.

[Kie17] Cornelia Kiefer. Die Gratwanderung zwischen qualitativ hochwertigen und einfach zu erstellenden domänenspezifischen Textanal-

ysen. In B. Mitschang et al., editor, *GI-Edition Lecture Notes in Informatics Datenbanksysteme für Business, Technologie und Web (BTW 2017) Workshopband*, Stuttgart, Germany, 2017.

[Kie19] Cornelia Kiefer. Quality Indicators for Text Data. In Meyer, Holger, et al., editor, *Datenbanksysteme für Business, Technologie und Web (BTW 2019), 18. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme (DBIS), 4.-8. März 2019, Rostock, Germany, Workshopband*, volume P-290 of *LNI*, pages 145–154. Gesellschaft für Informatik, Bonn, 2019.

[KK13] Christian Kirsch and Oliver Krueger. Aspekte einer Mobil-Strategie. In *Digitalisierung und Innovation*, pages 61–80. Springer, 2013.

[KK15] Laura Kassner and Cornelia Kiefer. Taxonomy Transfer: Adapting a Knowledge Representing Resource to new Domains and Tasks. In *Proceedings of the 16th European Conference on Knowledge Management*, pages 399–407, Udine, Italy, 2015.

[KKMR15] Cornelia Kiefer, Matthias Krämer, Sarah Müller, and Christian Röhrer. *Evaluation der unterschiedlichen Entwicklungen und Ausprägungen von Industrie 4.0 im globalen Kontext.* Interdisziplinäre Arbeit, unveröffentlichter Bericht, Graduate School of Excellence advanced Manufacturing Engineering (GSaME), University of Stuttgart, Stuttgart, Germany, 2015.

[Kla74] George R. Klare. Assessing readability. *Reading Research Quarterly*, 10(1):62–102, 1974.

[KLA15] Torsten Kilias, Alexander Löser, and Periklis Andritsos. Indrex: In-database relation extraction. *Information Systems*, 53:124–144, 2015.

[KLS16] Nils Krieg, Marco Link, and Benedict Steuerlein. *Auswertung der Robustheit von NLP-Tools für 'messy data'.* Student survey, University of Stuttgart, Stuttgart, Germany, 2016.

[KM16] Laura Kassner and Bernhard Mitschang. Exploring Text Classification for Messy Data: An Industry Use Case for Domain-Specific Analytics. In *Advances in Database Technology - EDBT 2016, 19th International Conference on Extending Database Technology, Proceedings*, pages 491–502, Bordeaux, France, 2016. OpenPro-

ceedings.org.

[Kon81] Alan G. Konheim. *Cryptography, a Primer*. John Wiley & Sons, Inc, New York, NY, USA, 1st edition, 1981.

[KP15] Cornelia Kiefer and Ulrike Pado. Freitextaufgaben in Online-Tests – Bewertung und Bewertungsunterstützung. *HMD Praxis der Wirtschaftsinformatik*, 52(1):96–107, 2015.

[KRM19a] Cornelia Kiefer, Peter Reimann, and Bernhard Mitschang. A Hybrid Information Extraction Approach Exploiting Structured Data Within a Text Mining Process. In Grust, Torsten, et al., editor, *Datenbanksysteme für Business, Technologie und Web (BTW 2019), 18. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme (DBIS), 4.-8. März 2019, Rostock, Germany, Proceedings*, volume P-289 of *LNI*, pages 149–168. Gesellschaft für Informatik, Bonn, 2019.

[KRM19b] Cornelia Kiefer, Peter Reimann, and Bernhard Mitschang. QUALM: Ganzheitliche Messung und Verbesserung der Datenqualität in der Textanalyse. *Datenbank-Spektrum*, 19(2):137–148, 2019.

[KRM20] Cornelia Kiefer, Peter Reimann, and Bernhard Mitschang. Prevent Low-Quality Analytics by Automatic Selection of the Best-Fitting Training Data. In *Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS)*, Maui, Hawaii, USA, 2020.

[Kub19] Mario Kubek. Natural language processing and text mining. In *Concepts and Methods for a Librarian of the Web*, pages 35–52. Springer International Publishing, Cham, 2019.

[Kuh13] Tobias Kuhn. A survey and classification of controlled natural languages. *Computational Linguistics*, 40(1):121–170, 2013.

[Lau16] Andreas Laukart. *Fit of training data and text data – automatic identification of the best fitting training data.* Student thesis, University of Stuttgart, Stuttgart, Germany, 2016.

[Lau17] Andreas Laukart. *Untersuchung zur Qualität von Fertigungsdaten – Ein Beispiel für die Analyse großer Datenmengen.* Student thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[Len18] Alessandro Lenci. Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171, 2018.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707, 1966.

[LFL98] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

[LGM+18] Yue Liu, Tao Ge, Kusum S. Mathews, Heng Ji, and Deborah L. McGuinness. Exploiting Task-Oriented Resources to Learn Word Embeddings for Clinical Abbreviation Expansion. *CoRR*, 2018.

[Lin17] Marco Link. *Erschließen von Freitextfeldern mittels Text Mining und die Qualität der gewonnenen Informationen*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[Lin18] Marco Link. *Auswirkungen von schlechter Datenqualität auf Datenanalyseergebnisse*. Term paper, University of Stuttgart, Stuttgart, Germany, 2018.

[Lin19] Marco Link. *Automatische Ressourcenselektion in Datenanalysepipelines*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2019.

[Liu11] Bing Liu. *Web data mining: Exploring hyperlinks, contents, and usage data*. Data-Centric Systems and Applications. Springer, Berlin, 2nd ed. edition, 2011.

[Liu15] B. Liu. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 1 edition edition, 2015.

[Llo06] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, 2006.

[LMV06] Luis Francisco Ramos Lima, Antonio Carlos Gastaud Maçada, and Lilia Maria Vargas. Research into information quality: A study of the state of the art in iq and its consolidation. In John R. Talburt, Elizabeth M. Pierce, Ningning Wu, and Traci Campbell, editors, *Proceedings of the 11th International Conference on Information Quality*, pages 146–158, Cambridge, MA, USA, 2006. MIT.

[LR95] Anany Levitin and Thomas Redman. Quality dimensions of a conceptual view. *Inf. Process. Manage.*, 31(1):81–88, 1995.

[LSKW02] Yang W. Lee, Diane M. Strong, Beverly K. Kahn, and Richard Y. Wang. Aimq: A methodology for information quality assessment. *Inf. Manage.*, 40(2):133–146, 2002.

[LVDv16] Artuur Leeuwenberg, Mihaela Vela, Jon Dehdari, and Josef van Genabith. A minimally supervised approach for synonym extraction with word embeddings. *The Prague Bulletin of Mathematical Linguistics*, 105(111 - 142), 2016.

[LW16] Matthias Lemke and Gregor Wiedemann. *Text Mining in den Sozialwissenschaften*. Springer Fachmedien, Wiesbaden, 2016.

[LY18] Yan Li and Jieping Ye. Learning Adversarial Networks for Semi-Supervised Text Classification via Policy Gradient. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1715–1723, London, United Kingdom, 2018. ACM.

[Man16] Cinmayii Manliguez. *Generalized Confusion Matrix for Multiple Classes*, 2016. Available at: `https://www.researchgate.net/publication/310799885_Generalized_Confusion_Matrix_for_Multiple_Classes/link/5838088d08aef00f3bf9e407/download`.

[McC12] Q. Ethan McCallum. *Bad data handbook*. O'Reilly Media, Sebastopol, CA, 2012.

[McK12] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, Inc, 2012.

[MGS+19] Mohamed Nadjib Mami, Damien Graux, Simon Scerri, Hajira Jabeen, Sören Auer, and Jens Lehmann. Uniform access to multiform data lakes using semantic technologies. In *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services*, iiWAS2019, pages 313–322, Munich, Germany, 2019. Association for Computing Machinery.

[Mie12] Jeff Mielke. A phonetically based metric of sound similarity. *Lingua*, 122(2):145–163, 2012.

[Mit16] Hans-Joachim Mittag, editor. *Statistik: Eine Einführung mit interaktiven Elementen*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[MK00] Eleni Miltsakaki and Karen Kukichy. Automated Evaluation of Coherence in Student Essays. In *Proceedings of International Conference on Language Resources & Evaluation (LREC)*, pages 1–8, Athens, Greece, 2000.

[ML10] Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Uppsala, Sweden, 2010. Association for Computational Linguistics.

[MLV+03] Paolo Missier, G. Lalk, Vassilios Verykios, F. Grillo, T. Lorusso, and P. Angeletti. Improving data quality in practice: A case study in the italian public administration. *Distributed and Parallel Databases*, 13:135–160, 2003.

[MM00] Jonathan I. Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *Iq*, pages 200–209, 2000.

[MMB12] Pablo N. Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123, Berlin, Germany, 2012.

[MMS93] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, 19(2):313–330, 1993.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, 2008.

[MV18] Nikolaos Misirlis and Maro Vlachopoulou. Social Media Metrics and Analytics in Marketing – S3M: A Mapping Literature Review. *International Journal of Information Management*, 38(1):270–276, 2018.

[MWLZ09] Stuart E. Madnick, Richard Y. Wang, Yang W. Lee, and Hongwei Zhu. Overview and framework for data and information quality research. *Journal of Data and Information Quality (JDIQ)*, 1(1):2:1–2:22, 2009.

[Nau14] Felix Naumann. Data profiling revisited. *SIGMOD Rec*, 42(4):40–49, 2014.

[Nig07] Héctor Oscar Nigro. *Data Mining with Ontologies: Implementations, Findings, and Frameworks*. IGI Global, 2007.

[NLDS03] Cheng Niu, Wei Li, Jihong Ding, and Rohini K. Srihari. Orthographic Case Restoration Using Supervised Learning Without Manual Annotation. *International Journal on Artificial Intelligence Tools*, 2003.

[NRC+11] Jason R.C. Nurse, Syed S. Rahman, Sadie Creese, Michael Goldsmith, and Koen Lamberts. Information quality and trustworthiness: A topical state-of-the-art review. *International Conference on Computer Applications and Network Security (ICCANS 2011)*, 2011.

[NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[NWH17] Andreas Niekler, Gregor Wiedemann, and Gerhard Heyer. Leipzig corpus miner - a text mining infrastructure for qualitative data analysis. *CoRR*, abs/1707.03253, 2017.

[OLAMTK10] José Olvera-López, Jesús Ariel Carrasco-Ochoa, José Francisco Martínez-Trinidad, and Josef Kittler. A Review of Instance Selection Methods. *Artif. Intell. Rev.*, 34:133–143, 2010.

[PA11] David Powers and Ailab. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2:2229–3981, 2011.

[Pan19] Narottam Panday. *Disambiguation on Write*. Student thesis, University of Stuttgart, Stuttgart, Germany, 2019.

[PCF+12] K. H. Prasad, S. Chaturvedi, T. A. Faruquie, L. V. Subramaniam, and M. K. Mohania. Managing data quality by identifying the noisiest data samples. In *Service Operations and Logistics, and Informatics (SOLI), 2012 IEEE International Conference*, pages 90–95, Suzhou, China, 2012.

[Per14] Jacob Perkins. *Python 3 text processing with NLTK 3 cookbook: Over 80 practical recipes on natural language processing techniques using Python's NLTK 3.0*. Packt Pub., Birmingham, UK, 2014.

[PK04] Leo Pipino and David P. Kopcso. Data mining, dirty data, and costs. In InduShobha N. Chengalur-Smith, Louiqa Raschid, Jennifer Long, and Craig Seko, editors, *Ninth International Conference on Information Quality (ICIQ 2004)*, pages 164–169, Cambridge, Massachusetts, USA, 2004. MIT.

[PK15] Ulrike Pado and Cornelia Kiefer. Short Answer Grading: When Sorting Helps and When it Doesn't. In Linköpings universitet Linköping University Electronic Press, editor, *Proceedings of the 4th workshop on NLP for Computer Assisted Language Learning, NODALIDA 2015*, Linköping Electronic Conference Proceedings, pages 42–50, Wilna, Lithuania, 2015. LiU Electronic Press and ACL Anthology.

[PLW02] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. Data quality assessment. *Commun. ACM*, 45(4):211–218, 2002.

[Pop18] Alexander Popov. Neural network models for word sense disambiguation: an overview. *Cybernetics and information technologies*, 18(1):139–151, 2018.

[Por80] Martin F. Porter. An algorithm for suffix stripping. *program*, 14(3):130–137, 1980.

[PT07] Elizabeth Pierce and Londraies Thomas. Assessing information quality using prediction markets. In *Proceedings of the 12th International Conference on Information Quality*, Cambridge, MA, USA, 2007.

[PY10] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Red01] Thomas C. Redman. *Data quality: The field guide.* Digital Press, Boston, 2001.

[Ren19] Tobias Renz. *Auswirkungen von Textcharakteristika auf die Qualität von Clustern.* Student thesis, University of Stuttgart, Stuttgart, Germany, 2019.

[RK10] Singh Ranjit and Singh Kawaljeet. A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing. *International Journal of Computer Science Issues*, 7, 2010.

[Rou87] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[RR08] Leonard Richardson and Sam Ruby. *RESTful web services*. O'Reilly Media, Inc, 2008.

[Rus07] Philip Russom. *BI Search and Text Analytics: New Additions to the BI Technology Stack*, 2007. Available at: `http://download.101com.com/pub/tdwi/Files/TDWI_RRQ207_lo.pdf`.

[Sad13] Shazia Sadiq. *Handbook of data quality: Research and practice*. Springer-Verlag, Berlin and New York, 2013.

[SAW15] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, and Teh Ying Wah. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, 10 e0144059(12), 2015.

[SBI+13] Michael Stonebraker, Daniel Bruckner, Ihab F. Ilyas, George Beskales, Mitch Cherniack, Stanley B. Zdonik, Alexander Pagan, and Shan Xu. Data curation at scale: The data tamer system. In *6th biennial Conference on Innovative Data Systems Research (CIDR)*, Asilomar, California, 2013.

[SBP06] E. F. A. Silva, F. A. Barros, and R. B. C. Prudencio. A hybrid machine learning approach for information extraction. In *Sixth International Conference on Hybrid Intelligent Systems (HIS'06)*, Rio de Janeiro, Brazil, Brazil, 2006.

[SC06] G. Shankaranarayanan and Yu Cai. Supporting data quality management in decision-making. *Decision Support Systems*, 42(1):302–317, 2006.

[SC13] Laura Sebastian-Coleman. *Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework*. Elsevier Science, Burlington, 2013.

[Sch13] B. Scherff. *Spracheingabe zur Programmierung von Schweißrobotern*. fir+iaw Forschung für die Praxis. Springer Berlin Heidelberg, 2013.

[Sha14] Matthew Shardlow. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing*

2014, 4(1), 2014.

[Shi14] L. Shih. *Chinas Industriepolitik von 1978-2013: Programme, Prozesse und Beschränkungen*. Springer Fachmedien Wiesbaden, 2014.

[SIG+12] Andreas Schmidt, Chris Ireland, Eloy Gonzales, Maria Del Pilar Angeles, and Dumitru Dan Burdescu. *On the Quality of Non-structured Data*. St. Gilles, Réunion, 2012. Available at: `http://www.iaria.org/conferences2012/filesDBKDA12/DBKDA_2012_PANEL.pdf`.

[Son04] Daniel Sonntag. Assessing the Quality of Natural Language Text Data. In *GI Jahrestagung*, pages 259–263, Ulm, Germany, 2004.

[SPI08] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *the 14th ACM SIGKDD international conference*, page 614, Las Vegas, Nevada, USA, 2008.

[SSMM12] Markus Schaal, Barry Smyth, Roland M. Mueller, and Rutger MacLean. Information quality dimensions for the social web. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pages 53–58. ACM, 2012.

[ST10] Martin Schierle and Daniel Trabold. Multilingual Knowledge-Based Concept Recognition in Textual Data. In Andreas Fink, Berthold Lausen, Wilfried Seidel, and Alfred Ultsch, editors, *Advances in Data Analysis, Data Handling and Business Intelligence*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 327–336. Springer, Berlin, Heidelberg, 2010.

[Sta90] Brunhild Staiger. Wenn der Drache sich erhebt. China zwischen Gestern und Heute. By Thomas Heberer. [Baden-Baden: Signal-Verlag 1988. 224 pp. DM34.80]. *The China Quarterly*, 121:152–153, 1990.

[SV06] Valerie Sessions and Marco Valtorta. The effects of data quality on machine learning algorithms. In *Proceedings of the 11th International Conference on Information Quality*, pages 485–498, Cambridge, MA, USA, 2006.

[SWZ00] Ganesan Shankaranarayanan, Richard Wang, and Mostapha Ziad. Ip-map: Representing the manufacture of an information product. In *Proceedings of the 5th Conference on Information Quality (IQ)*, pages 1–16, Cambridge, MA, USA, 2000.

[TA07] S. Tartir and I. B. Arpinar. Ontology Evaluation and Ranking using OntoQA. In *International Conference on Semantic Computing (ICSC 2007)*, pages 185–192, Irvine, California, USA, 2007.

[TBHG00] Pang-Ning Tan, Hannah Blau, Steve Harp, and Robert Goldman. Textual data mining of service center call records. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 417–423, Boston, Massachusetts, USA, 2000. ACM.

[TLKL15] Ion-George Todoran, Laurent Lecornu, Ali Khenchaf, and Jean-Marc Le Caillec. A Methodology to Evaluate Important Dimensions of Information Quality in Systems. *Journal of Data and Information Quality*, 6(2-3):1–23, 2015.

[TMS03] Ann Taylor, Mitchell Marcus, and Beatrice Santorini. The penn treebank: An overview. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 5–22. Springer Netherlands, Dordrecht, 2003.

[TSRC15] Ignacio G. Terrizzano, Peter M. Schwarz, Mary Roth, and John E. Colino. Data wrangling: The challenging yourney from the wild to the lake research. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, USA, 2015. www.cidrdb.org.

[Unia] Universität Jena. *Computational and Data Science: Ein neuer Studiengang für eine neue Wissenschaftsdisziplin.* Available at `http://www.cds.uni-jena.de/`.

[Unib] Universität Stuttgart. *Abteilung Digital Humanities.* Available at `http://www.uni-stuttgart.de/dh/studium/`.

[VHD+14] Tobias Vogel, Arvid Heise, Uwe Draisbach, Dustin Lange, and Felix Naumann. Reach for gold. *Journal of Data and Information Quality*, 5(1-2):1–25, 2014.

[VZRM18] Alejandro Gabriel Villanueva-Zacarias, Peter Reimann, and Bernhard Mitschang. A framework to guide the selection and configuration of machine-learning-based data analytics solutions in manufacturing. In *Procedia CIRP*, volume 72, pages 153–158, Stockholm, Sweden, 2018.

[WAMP14] Lars Wissler, Mohammed Almashraee, Dagmar Monett, and Adrian Paschke. The gold standard in corpus annotation. In *IEEE GSC*, Passau, Germany, 2014.

[Wan98] Richard Y. Wang. A product perspective on total data quality management. *Commun. ACM*, 41(2):58–65, 1998.

[WBP13] Philip Woodall, Alexander Borek, and Ajith Kumar Parlikad. Data quality assessment: The hybrid approach. *Information & Management*, 50(7):369–382, 2013.

[WFH11] I. H. Witten, Eibe Frank, and Mark A. Hall. *Data mining: Practical machine learning tools and techniques*. [Morgan Kaufmann series in data management systems]. Morgan Kaufmann, Burlington, MA, 3rd ed. edition, 2011.

[WGM07] Markus Weimer, Iryna Gurevych, and Max Mühlhäuser. Automatically assessing the post quality in online discussions on software. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 125–128, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[Wic14] Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10):1–23, 2014.

[Wis93] Michael Wise. String similarity via greedy string tiling and running karp−rabin matching. *Unpublished Basser Department of Computer Science Report*, 1993.

[WLB08] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Enhanced Integrated Scoring for Cleaning Dirty Texts. *CoRR*, 2008.

[WLB+16] Hongzhi Wang, Mingda Li, Yingyi Bu, Jianzhong Li, Hong Gao, and Jiacheng Zhang. Cleanix. *ACM SIGMOD Record*, 44(4):35–40, 2016.

[WNC05] Jigang Wang, Predrag Neskovic, and Leon N. Cooper. Training data selection for support vector machines. In Lipo Wang, Ke Chen, and Yew Soon Ong, editors, *Advances in Natural Computation*, pages 554–564, Berlin, Heidelberg, 2005. Springer.

[WS96] Richard Y. Wang and Diane M. Strong. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manage. Inf. Syst.*, pages 5–33, 1996.

[WSF95] Richard Y. Wang, Veda C. Storey, and Christopher P. Firth. A framework for analysis of data quality research. *IEEE Trans. on Knowl. and Data Eng.*, 7(4):623–640, 1995.

[WW15] Philip Woodall and Anthony Wainman. Data quality in analytics: Key problems arising from the repurposing of manufacturing data. In *Proceedings of the 20th International Conference on Information Quality (ICIQ)*. MIT Information Quality Program, Cambridge, MA, USA, 2015.

[XZZ08] Ji-Yi Xiao, Dao-Hui Zhu, and La-Mei Zou, editors. *A hybrid approach for web information extraction: 2008 International Conference on Machine Learning and Cybernetics*, volume 3, 2008.

[YB18] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In *27th International Conference on Computational Linguistics (COLING 2018)*, Santa Fe, New-Mexico, USA, 2018.

[YLCC07] Wei Yu, Qing Li, Junpeng Chen, and Jiaheng Cao. OS-RANK: Structure Analysis for Ontology Ranking. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on Data Engineering*, pages 339–346, Istanbul, Turkey, 2007.

[ZH19] Max J. Zenglein and Anna Holzmann. *Evolving Made in China 2025: China's industrial policy in the quest for global tech leadership*, 2019. Available at: `https://www.merics.org/en/papers-on-china/evolving-made-in-china-2025`.

[Zha15] Ce Zhang. *DeepDive: A Data Management System for Automatic Knowledge Base Construction.* PhD thesis, University of Wisconsin-Madison, 2015.

[ZM16] ChengXiang Zhai and Sean Massung. *Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining*. ACM, New York, USA, 2016.

[ZMZ16] Yuhao Zhang, Wenji Mao, and Daniel Zeng. A non-parametric topic model for short texts incorporating word coherence knowledge. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 2017–2020, Indianapolis, Indiana, USA, 2016. ACM.

[Zod15] Stephan Zoder. Improving enterprise master data quality - what is the roi? a practitioner's analysis of the financial value organizations can expect from improving overall master data quality. In *21st Americas Conference on Information Systems (AMCIS)*, Puerto Rico, 2015. Association for Information Systems.

[ZOM12] Ramon Ziai, Niels Ott, and Detmar Meurers. Short answer assessment: Establishing links between research strands. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA7)*, Montreal, Canada, 2012. Association for Computational Linguistics.

All links in this bibliography and throughout the rest of this document have last been visited and found working on August 03, 2020.

List of Figures

List of Tables