AIM
A Social Media Monitoring System
for Quality Engineering

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig angenommene

D I S S E R T A T I O N
zur Erlangung des akademischen Grades

DOCTOR rerum naturalium
(Dr. rer. nat.)

im Fachgebiet
Informatik

vorgelegt
von Dipl.-Inf. Mathias Christoph Bank
geboren am 09.12.1981 in Sigmaringen

Rodgau, den 18. Juni 2013

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. Gerhard Heyer, Universität Leipzig
2. Prof. Dr. Alexander Mehler, Universität Frankfurt am Main

Die Verleihung des akademischen Grades erfolgt mit Bestehen der
Verteidigung am 14.06.2013 mit dem Gesamtprädikat magna cum laude.
Abstract

In the last few years the World Wide Web has dramatically changed. It is no longer an information infrastructure where only a small number of people is able to publish content. Rather it has become a communication platform, in which every user can actively participate. This causes a huge amount of data dealing with every aspect of life, including quality topics of products and services. Analyzing Social Media promises to improve quality assurance processes, covering topics that are difficult to measure with currently available quality sensors. The systematic and reproducible processing of user generated content enforces, however, the adaption of already available tools as well as the creation of new Social Media specific analysis algorithms. This thesis presents a Social Media monitoring system which was developed to support an analysis protagonist investigating thousands of user comments with minimal time efforts. Its usage has shown unique advantages enabling an analysis protagonist to discover the customers’ definition of “quality”.

Kurzzusammenfassung

Acknowledgments

The most inspiring part in writing a scientific contribution is the discussion with other researchers. The expressed ideas and thoughts at conferences, team meetings and in discussions with colleagues within the Daimler AG have been nearly an endless information source which finally influenced my work significantly. So I am really thankful to many, many people. In the following, I want to highlight some of them.

First of all, I would like to express my sincere gratitude to Prof. Dr. Gerhard Heyer. His constructive criticism significantly improved this thesis while his continuous encouragement ensured the final quality of this work. He also brought me into contact with very impressive colleagues at the University of Leipzig. The discussions at the University have always been inspiring and I am always enjoying being there.

Additionally, I would like to thank my former team at the Daimler Research institute in Ulm. Of particular note here is my supervisor at the department for Quality and Reliability, Jürgen Franke. Without his tireless efforts, the inspiring discussions and finally his critical questionings this thesis would not have been possible at all. I learned a lot during that time and I am very happy to have been part in his team. In consequence I also want to thank all of the team members, namely Ulrich Bohnacker, Dr. Martin Schierle, Dr. Christian Hänig, Daniel Trabold, Michael Mattes and Robert Remus for their ideas, arguments and their time.

The ideas and results of this work have been investigated in many meetings with other departments within the company. Therefore, I want to thank further Daimler colleagues who provided me with valuable background information, requirements and feedback for the created Social Media monitoring system, namely Wolfgang Clar, Dr. Matthias Grabert and Dr. Axel Sprenger.

During the creation of this work, I have also been in intensive contact with researchers from other universities. Of particular note are the discussions with Dr. Thomas Gottron (formerly University of Mainz, now University of Koblenz-Landau) with regard to crawler technologies and Dr. Friedhelm Schwenker (University of Ulm) in context of machine learning challenges. I would also like to thank Prof. Dr. Franz Schweiggert from the University of Ulm for his support in context of the complete thesis.

Of course, I do not want to forget my friends who have never been tired to review the content of my work. With their help, writing this thesis was – overall – a lot of fun.
Last but not least, the greatest thanks belong to my wonderful wife Elena and to my family. They always supported my work and ensured that I have enough free-time to fulfill this thesis. Without their love and their understanding, I would have never been able to finalize this step in my life.
# Contents

**Contents**  

List of Tables iii  

List of Figures v  

1** Introduction**  

1.1 Chapter Overview ................................. 5  

2** Problem Definition and Data Environment**  

2.1 Commonly Applied Quality Sensors ............... 11  
2.2 The Growing Importance of Social Media .......... 20  
2.3 Social Media based Quality Experience .......... 21  
2.4 Change to the Holistic Concept of Quality ....... 24  
2.5 Definition of User Generated Content and Social Media ... 31  
2.6 Social Media Software Architectures .............. 33  

3** Data Collection**  

3.1 Related Work ..................................... 39  
3.2 Requirement Analysis .............................. 54  
3.3 A Blackboard Crawler Architecture ............... 59  
3.4 Semi-supervised Wrapper Generation ............... 74  
3.5 Structure Modification Detection ................. 88  
3.6 Conclusion ........................................ 89  

4** Hierarchical Fuzzy Clustering**  

4.1 Related Work ..................................... 96  
4.2 Generalization of Agglomerative Crisp Clustering Algorithms ...... 119  
4.3 Topic Groups Generation ........................... 130  
4.4 Evaluation ......................................... 134  
4.5 Conclusion ........................................ 152
## Contents

5 A Social Media Monitoring System for Quality Analyses 155
  5.1 Related Work ................................................. 158
  5.2 Pre-Processing Workflow ................................. 160
  5.3 Quality Indices .............................................. 171
  5.4 AIM Architecture .......................................... 178
  5.5 Evaluation .................................................. 186
  5.6 Conclusion ................................................... 194

6 Conclusion and Perspectives 197
  6.1 Contributions and Conclusions ......................... 197
  6.2 Perspectives ................................................ 200

Bibliography 203
List of Tables

2.1 Kano questionnaire questions ........................................ 16
3.1 Statistical distribution of the length of comments ............... 78
3.2 Wrapper Quality for Discussion and Posting Detection ........ 86
3.3 Engine based Quality Analysis ....................................... 87

4.1 Overview for the most common term weighting components .... 99
4.2 Classical Similarity Measures ........................................ 100
4.3 Table of Lance-Williams coefficients ............................... 113
4.4 Lance-William coefficients for hierarchical fuzzy clustering .. 128
4.5 Cosine Quality Evaluation ........................................... 145
4.6 Fuzziness Evaluation .................................................. 146
4.7 Cosine Quality for multiply assigned documents ............... 147
4.8 Cluster Size Evaluation .............................................. 147
4.9 Number of clusters per hierarchy in the complete cluster graph 148
4.10 Cluster Sizes in the complete cluster graph ................. 148
4.11 Cosine Quality in the complete cluster graph .................. 149

5.1 Sentiment Word Categories .......................................... 169
5.2 Correlation analysis of the satisfaction index and J.D. Power’s I.Q.S. study ......................................................... 188
List of Figures

2.1 Product Perception .......................................................... 10
2.2 Shortcoming categories and their monitoring ..................... 14
2.3 Review based recommendation system ................................. 22
2.4 Topic Distribution in benzworld.org ................................. 26
2.5 Explicitly provided production year distribution in benzworld.org 27
2.6 Implicitly provided production year distribution in benzworld.org 28
2.7 Top 20 product features for Mercedes GLK before product launch 29
2.8 Perception Discussions ................................................... 29
2.9 Top 20 product features for Mercedes GLK after market launch 30
2.10 Overview page structure ................................................ 34
2.11 Detail page structure ..................................................... 35

3.1 Site Style Tree calculation .............................................. 48
3.2 UML class diagram UGC ................................................ 65
3.3 UML activity diagram for Weblogs .................................... 67
3.4 UML activity diagram for Internet fora ............................... 69
3.5 Discussion URL Prediction ............................................... 72
3.6 XPath expressions for content detection ............................ 76
3.7 Post candidate search .................................................... 79
3.8 Multiple text nodes in one user posting ............................. 81
3.9 Wrapper Generation ...................................................... 85

4.1 Multiple Topics per document .......................................... 94
4.2 Hierarchical Cluster Example .......................................... 102
4.3 Crisp and Pyramidal Dendrogram .................................... 115
4.4 Fuzzy membership calculation ........................................ 124
4.5 Crisp clustering vs. fuzzy clustering ................................. 126
4.6 Topic Generation via Graph Pruning ................................. 132
4.7 Crisp clustering vs. fuzzy clustering with \( h_{\text{max}} = 5 \) ........ 133
4.8 Treemap visualization shows topical groups ....................... 135
4.9 Label propagation for quality evaluation ......................... 142
List of Figures

4.10 Data set creation for stability evaluation ....... 149
4.11 Cluster Stability Evaluation ....... 151

5.1 Pre-processing pipeline to extract relevant information from textual data ....... 161
5.2 A taxonomy structure for supervised topic detection in headed contexts ....... 166
5.3 Multiple sentiment expressions within one sentence ....... 167
5.4 AIM Analysis architecture ....... 179
5.5 AIM Analysis Frontend ....... 183
5.6 Pivot Analysis Scheme ....... 184
5.7 Deep-Dive Analyses ....... 185
5.8 J.D. Power's I.Q.S. results vs. the market satisfaction index ....... 189
5.9 Market satisfaction vs. product satisfaction – a real world example ....... 191
5.10 Detailed problem analysis example ....... 191
5.11 Detailed user comments provide problem circumstances ....... 192
5.12 Market satisfaction vs product satisfaction – another real world example ....... 193
5.13 Detailed problem analysis example ....... 193
5.14 Original user comments provide problem solutions ....... 194
Introduction

In the last few years the World Wide Web has dramatically changed the way people are communicating with each other. In the beginning, an active participation was limited to technically experienced users and companies. The majority of Internet users has only been able to watch content which was provided by a minority. With the growing availability of Social Media Systems like Internet fora, weblogs and social networks like facebook, XING or StudiVZ, the Web has changed. It is today, what it was originally designed for: A technical platform in which all users are able to interact with each other. Nowadays, there are billions of user comments available discussing all aspects of life and the data source is still growing. In 2011, the German newspaper *Süddeutsche Zeitung* described the Internet as follows:\(^1\):

\[\text{http://sz-magazin.sueddeutsche.de/texte/anzeigen/36648} \] accessed on 11/27/2011, freely translated by the author of this work.

The Internet is the birthplace and living environment of the communication society and thus the cipher of a new era. One day, it will be considered as groundbreaking as the invention of trains, cars, and airplanes. One step further to overcome limitations of time and space.

\(^1\)
Chapter 1. Introduction

The increasing amount of User Generated Content (UGC) becomes more and more relevant for our daily life. Recent studies have shown that this data could provide high quality content which is equivalent to editorially created data [87]. Also, user driven data have gained a lot of trust in the last years. Nielsen’s consumer report for example states in 2007 that 78% of online users trust recommendations of other customers [125]. This is much more than any other company communication is able to reach and thus demonstrates why UGC is so important to marketing research today. By measuring the customers’ feedback companies learn about their public perception and can adjust their communication strategy accordingly.

At the same time, manufacturers are dealing with an increasingly competitive environment in which there are two main product strategies: Either a product implements unique selling propositions, or the product fulfills a high quality standard as a source of competitive advantage (cf. [114]). This is also true for the automotive domain, in which this thesis was written. With regard to so-called premium car manufacturers, the product strategy is more and more defined in terms of quality. It is mainly based on customer satisfaction (cf. [105]) and can be distinguished in objective and subjective quality (cf. [151]). Numerous managing and engineering efforts are made to ensure that all product requirements will be fulfilled. The derived tasks, that are generally considered as Quality Engineering tasks, depend, however, on the experience and knowledge of the manufacturers and market research companies involved. With regard to technical aspects, many standards and methods have been established in the last years supporting a very high technical quality standard for many manufacturers today. However, the customers can disapprove of a product, even if it works as designed. Handling problems, for example, occur when a customer expects a different product behavior than the implemented one (e.g. an easier way to update the navigation map material). While perception shortcomings are no hard failures, they might prevent a customer to buy a product (e.g. a small-looking trunk). In the premium market segment, service quality is another important product strategy to satisfy customers. Controlling all these as-


pects of subjective quality is a challenging task, because it is very difficult to measure. Manufacturers and market researchers typically try to identify customers’ expectations and needs by using surveys. However, applying such a survey is a non-trivial task. On the one hand, it is important to receive information as detailed and complete as possible. On the other hand it is important to avoid overestimating any third-person’s preference as the customer’s opinion. Recent studies show that it is nearly impossible not to influence the survey participants. Distinct questions might already induce the customer to think about topics which have been completely irrelevant before. Comparable survey results are thus only possible by reasking the same questions. This limits the survey results concerning new market expectations (e.g. green technology) or new product features. In addition, questionnaires are often limited to yes/no questions to ensure a reproducible evaluation. However, subjective quality often enforces trade-off decisions. For example all customers prefer cars which need less fuel, but no-one wants to pay a lot of money for that.

This thesis tries to overcome the limitations of conventional surveys. The idea is to use the freely provided data in Social Media systems instead of asking questions. This allows a deeper just-in-time insight into customers’ expectations and behaviors. The concept is based on the observation that customers are not only posting marketing relevant information. They also publish product oriented content including positive and negative experiences. It is assumed that this information represents a valuable data source for quality analyses: The original voices of the customers promise to specify a more exact and more concrete definition of “quality” than the one that is available to manufacturers or market researchers today.

However, the huge amount of unstructured user comments makes their evaluation very complex. It is impossible for an analysis protagonist to manually investigate the provided customer feedback. Therefore, Social Media specific algorithms have to be developed to collect, pre-process and finally analyze the data. This has been done by the Social Media monitoring system AIM (Automotive Internet Mining) that is the subject of this thesis.
Chapter 1. Introduction

It investigates how manufacturers, products, product features and related opinions are discussed in order to estimate the overall product quality from the customers’ point of view. AIM is able to track different types of data sources (e.g. Internet fora and weblogs) using a flexible multi-agent based crawler architecture. In contrast to classical web crawlers, the multi-agent based crawler supports individual crawling policies to minimize the download of irrelevant web pages. In addition, an unsupervised wrapper induction algorithm is introduced to automatically generate content extraction parameters which are specific for the crawled Social Media systems. These parameters are used in a special XPath based content extraction algorithm. It extracts individual user comments and their relation to each other thus enabling highly reliable analyses in further steps. Moreover, the algorithm is able to neglect navigation sections, advertisement and other irrelevant page elements. The extracted user comments are analyzed by different content analysis algorithms to gain a deeper insight into the discussed topics and opinions. Hereby, three different topic types are supported depending on the analysis needs.

1. The creation of highly reliable analysis results is realized by using a special context-aware taxonomy-based classification system.

2. Fast ad-hoc analyses are applied on top of classical fulltext search capabilities.

3. Finally, AIM supports the detection of blind-spots by using a new fuzzified hierarchical clustering algorithm. It is able to detect multiple topics per user comment and thus highlights previously unknown topics.

All three topic types are treated in a unified way to enable an analysis protagonist to apply all methods simultaneously and in exchange. The systematically processed user comments are visualized within an easy and flexible interactive analysis frontend. Special abstraction techniques support the investigation of thousands of user comments with minimal time ef-
forts. Hereby, specifically created indices show the relevancy and customer satisfaction of a given topic. The abstraction layers are complemented by multiple deep-dive techniques to filter any data set to a small but relevant subset of user comments. By reading the original voice of the customer, the analysis protagonist can get further background information to improve customer satisfaction.

The analysis of UGC with the help of the AIM system offers unique advantages for quality analyses. Some real-world results are discussed in detail at the end of this work. The applied techniques are, however, not limited to extract quality related information from Social Media systems. Moreover, each algorithm was developed to be transferable to other use cases, especially in the context of Social Media. Therefore, the different chapters of this thesis are dealing separately with specific requirements and use cases. Each chapter concludes with an evaluation that shows the unique advantages of the proposed algorithm. Thus, this work not only describes a Social Media monitoring system to gather quality related information. It also presents techniques required to investigate UGC in general, the collection of UGC from different Social Media systems, the analysis of user comments, and finally methods to create a deterministic and interactive analysis system.

1.1 Chapter Overview

This thesis deals with the challenging task of creating a Social Media monitoring system to gather quality related information. The gained insights into user comments provide a more complete definition of “quality” and thus have an important impact on Quality Engineering tasks. Chapter 2 describes the environment in which this thesis was created. It describes the current state of applied quality analyses including shortcomings based on findings of related work. It then investigates in which way Social Media could overcome such limitations. As “Social Media” is a very divers concept that is used for different data types in different publications, the chapter further provides an exact definition of the data that is used in this work.
including its typical representation.

The requirements to a suitable Social Media monitoring system are discussed in chapter 3, 4, and 5. All three chapters first describe requirements for an appropriate algorithm to be usable in the context of this work. Then, related work is investigated in detail. The scientific contributions of this thesis are presented separately to point out the new scientific findings.

The first step to analyze UGC is to gather user comments from different Social Media systems, which is described in chapter 3. This chapter first provides an overview of current state-of-the-art crawler architectures and concepts. Based on this, the requirements for an appropriate Social Media crawler architecture are discussed in detail in section 3.2. They are addressed in a new harvesting architecture that supports an individual optimization for different data sources while minimizing reimplementation efforts. The architecture is based on a special multi-agent concept and discussed in section 3.3. The implementations of two Social Media specific crawlers, namely Internet fora and weblog crawlers, are presented in section 3.3.1. Both use special content extraction algorithms – so-called wrappers – to extract only the information that is relevant for quality analysis aspects. However, the manual creation of a wrapper may be very time consuming. Therefore, section 3.4 describes an unsupervised wrapper induction algorithm that supports the user in adding new data sources. The wrapper application is continuously monitored as shown in section 3.5.

The large amount of textual data of Social Media systems has to be further analyzed to support an analysis protagonist. One important task is to extract the discussed topics. Chapter 4 deals with the analysis of previously unseen topics. After specifying the requirements of a Social Media based topic analysis, section 4.1 discusses related work dealing with this challenging task. The available approaches are however unsatisfying. Therefore, a new agglomerative hierarchical clustering algorithm is presented in section 4.2. It fuzzifies well-known crisp clustering algorithms to support multiple topics within one user comment. Section 4.3 extends the algorithm with a special pruning process that creates topical groups instead of binary clus-
1.1. Chapter Overview

These groups are evaluated in section 4.4 using two special evaluation techniques. They analyze the quality and stability of the topical cluster. It is shown that the algorithm creates stable and deterministic topical clusters with multiple abstraction levels and high quality. Multiple topics within one user comment can be flexibly considered. The algorithm thus realizes unique analysis properties that are absolutely necessary in a Social Media monitoring system.

Chapter 5 presents the complete Social Media analysis system AIM. Its three-tier architecture provides several analysis algorithms and visualizations enabling an analysis protagonist to gain a fast and profound quality insight. After defining the specific requirements for an appropriate Social Media monitoring system, related work dealing with such systems is discussed in section 5.1. The available approaches neglect important requirements for a reliable analysis system. Therefore, the new monitoring system AIM (Automotive Internet Mining) is proposed. Section 5.2 discusses the applied pre-processing which structures the a-priori unstructured textual data. The extracted information is used to provide the analysis protagonist with different analysis possibilities. Section 5.3 introduces new quality indices enabling an analyst to investigate thousands of user comments with minimal time effort. In combination with further abstraction algorithms, a very flexible and scalable architecture is created, which is discussed in detail in section 5.4. Its unique features are evaluated with a state-of-the-art quality index within the automotive domain (section 5.5). Further real-world use cases illustrate the unique advantages applying the proposed analysis system. The possibility to flexibly analyze UGC on different abstraction levels not only supports the identification of problems. The deep-dive analyses also provide relevant background-information. It can be used to accelerate problem solving and to improve the overall customers’ experience.

Finally, chapter 6 summarizes this thesis and shows further potential development.
Problem Definition and Data Environment

In a world of mature markets and intense competition, there are mainly two product strategies: Either the product provides unique features or it fulfills a high quality standard as a source of competitive advantage [114]. As more and more product features are created by suppliers, it is very difficult today to base the competitive advantage on unique selling propositions. Most manufacturers make therefore big efforts to ensure a high product quality. Quality however is not precisely specified. Shewart [151] distinguished objective and subjective quality aspects which means that quality is not limited to technical issues. Studies have shown in addition that a superior service and a fast and unbureaucratic problem solution can compensate technical shortcomings (cf. [174]). Today, quality is more or less defined in terms of customer satisfaction [105]. It is however not clearly stated which specific product features have to be realized and in which way they have to be implemented to increase customer satisfaction. Marketing and customer loyalty additionally influence buying decisions and are thus directly related to customer satisfaction (fig. 2.1). Customer loyalty for example decreases, if the customer has not been satisfied for a long time. A pre-eminent customer satisfaction on the other side increases the marketing
Chapter 2. Problem Definition and Data Environment

Figure 2.1: Customer satisfaction is not the only factor that influences buying decisions. The overall product perception also includes marketing and product loyalty. Each factor influences the others.

potential especially for mouth-to-mouth marketing. An inadequate product web page could further decrease customer satisfaction because the customer might not be able to get required information. “A quality engineer must understand all aspects of quality” [135].

The customer requirements, his expectations and needs highly depend on the market segment and the industry-specific standards. A low priced product for example can be perceived as high quality although it may provide substandard features. “What matters is what your customers think about your quality.” [135]. Therefore, a product manufacturer has to perform different operational, managerial, and engineering activities to define individual quality characteristics. This process is known as “Quality Engineering” (QE, cf. [135]) for which some basic requirements are defined in ISO standards (e.g. EN ISO 9000, EN ISO 90001, EN ISO 90004, EN ISO 19011). The activities are based on monitoring, analysis and improvement tasks (EN ISO 90001, chapter 8). These are typically realized using customer surveys, market analyses, continuous product and material tests, etc.. Analyzing commonly applied quality sensors in the automotive domain, it can be seen however that there are some analysis gaps that cannot be investigated using classical monitoring techniques. This is exemplarily shown by discussing one survey technique in detail. It is further discussed in this chapter whether and to which extent Social Media can fill these
2.1. Commonly Applied Quality Sensors

The life time of a product can be divided into four phases: the development phase, the market introduction phase, the production phase, and the post-production phase. Quality measurements have to be applied in all four phases right from its beginning in the development phase. They have to be taken during the market introduction to ensure that customers’ needs are fulfilled and have their peak during the production phase to react fast upon unexpected functioning. The quality measurements should even be performed in the post-production phase as the findings could improve the long-live behavior for future products.

During the last two phases, each individual product has two individual stages: the time, in which warranty and goodwill (W&G) are offered and the time until the product is not able to fulfill its tasks anymore (post-W&G). Quality information for the first stage is very important to limit future costs while the second stage provides relevant data especially for long-run products. Continuously gathering quality related information is the precondition to provide a product with superior product experience.

In the following the major quality sensors applied in the automotive domain are categorically discussed. This discussion is limited to quality sensors directly related to customers. It is obvious that there are many more quality sensors.

---

1 Due to confidentiality, this section is limited to a very abstract discussion about commonly applied quality sensors in which customers might be involved. It is no complete overview about different steps of Quality Engineering nor about customer-independent monitoring systems.
Chapter 2. Problem Definition and Data Environment

analysis methods especially during the development phase (e.g. crash tests, material tests, etc.). They are out of focus for this work. Quality management aspects are also neglected.

2.1.1 Survey on Applied Quality Sensors

A superior product experience is only possible when a manufacturer is able to recognize customers’ expectations and requirements. Using different QE sensors for quality monitoring, a systematic evaluation is possible. The sensors applied in the automotive domain can be categorized into four different groups:

1. **Payroll Sensors:** While warranty and goodwill are offered, each garage has to clear replaced parts to get paid for the part and the corresponding working time. The repair protocol has to be transferred into a clearing system in conjunction with several product details (e.g. model name, mileage, build year, etc.). The highly structured data source is used for many different kinds of statistics and data mining algorithms, e.g. early warnings and correlation analysis methods (cf. [23]).

2. **Garage Feedback Sensors:** Technical problems are generally solved in garages. In case of uncommon misbehaviors, a typical mechanic however needs individual advice to solve the problem. Therefore, database based question answering system have been established in which the mechanic is able to search for related problems and suggested problem solutions. If there is no answer for a given question, a quality expert can be contacted via hotline. If the problem cannot be solved remotely, specially trained quality experts are sent to the garage or the defective part will be sent to the manufacturer. New findings are protocoded and will be available in the question answering system. By continuously monitoring this system, different quality reports and indices are available.
3. **Customer Service Sensor**: Each customer can contact the company in case of problems or questions via a customer hotline. The discussed issues are protocoted and manually structured according to predefined classes for statistical analyses.

4. **Market Observation**: To get a more complete insight into customers’ needs, expectations and product perception, different survey techniques are applied during the whole life time of a product. In the last years, several market research companies have gained much attraction in this section. Best known in the automotive domain is the market research company J.D. Power\(^2\). It regularly carries out several (phone) surveys to gain customers’ feedback on different aspects of a bought product\(^3\):

   - The Initial Quality Study (IQS) is applied 90 days after purchase and provides problems reported by customers.
   - The Vehicle Dependability Study (VDS) investigates the long-term quality after three years of ownership.
   - The Automotive Performance, Execution and Layout Study (Appeal) measures what customers like about their product after 90 days of ownership.
   - The Customer Service Index Study (CSI) analyses the satisfaction of vehicle owners after maintenance or repair issues.
   - The Sales Satisfaction Index Study (SSI) examines the dealership’s ability to manage the sales process from product presentation, price negotiation and vehicle purchase to delivery and the finance and insurance process.

All these sensors try to provide an insight into the perceived quality. Each sensor focuses however only on special aspects. The first two sensors are based on data created in or by garages. This data covers situations in

---

\(^2\) [www.jdpower.com](http://www.jdpower.com)

which a previously specified behavior is not fulfilled. These technical aspects can be caused by specification or production shortcomings. Each of them is protocoded in detail and used in different data mining algorithms and statistical analyses to react fast on unexpected records. The extensive usage of this data is based on the fact that quality was defined in terms of technical quality for a long time. In conjunction with increasing standardization, certified development and production processes technical quality is expected to be at a very high level. Fast reactions to unexpected behaviors are systematically supported. The consideration of subjective quality is not as established and still causes discussion on how subjective quality is exactly defined. Especially in cases of design decisions, the influence on quality aspects is not clearly stated. With increasing competition, subjective quality becomes more and more important, which is why market researchers like J.D. Power are such important today. The applied surveys investigate perception, handling, and service shortcomings.

Fig. 2.2 assigns each sensor category to specific shortcoming types, which are grouped to conceptual shortcomings and shortcomings in the execution of work. It can be seen that not all possible shortcomings are continu-
Commonly Applied Quality Sensors

While technical aspects are documented in a very detailed way until warranty and goodwill (W&G) ends, the available information decreases very fast after this period. This is because the garages are not clearing broken material with the manufacturer anymore and many customers will not repair broken material in contract dealerships. The measurement of perception, handling and service shortcomings is further limited. The punctual execution of predefined questionnaires creates a-priori limited analysis results while warranty and goodwill are offered. The next section will discuss this problem in detail by exemplarily investigating a special survey technique: the Kano survey. The discussed issues are also available in other survey forms. The Kano survey is investigated as the conceptual idea is relevant for this work, which will be seen later.

2.1.2 Kano’s Attractive Quality as an Example for Customer Surveys

For the analysis and prioritization of customer requirements, Kano et al. have recognized that requirements can be classified into different priority levels (cf. [89]). They have proposed to use a special questionnaire to classify each product feature: For each question a functional and a dysfunctional part depict the customer’s reaction in case of presence or absence of a feature. The respondent has five possible choices for each part of the question (Table 2.1).

Each pair of answers has to be classified into one of six quality categories by using a special evaluation table given by Kano et al. [89]. The categories are defined as follows:

1. Attractive (A): The absence of this feature is not critical. The existence however would increase customer satisfaction.

2. Must-Be (M): These features are must-be requirements. An absence is very critical and is probably causing loss of sellings.
Table 2.1: A pair of customer requirement questions in a Kano questionnaire: For each part, the customer has five possible choices. These are the translations proposed by Matzler and Hinterhuber [114] for the original answers in Japanese.

<table>
<thead>
<tr>
<th>Functional</th>
<th>Dysfunctional</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the feature is present, how do you feel?</td>
<td>If the feature is not available, how do you feel?</td>
</tr>
<tr>
<td>1. I like it that way.</td>
<td>1. I like it that way.</td>
</tr>
<tr>
<td>2. It must be that way.</td>
<td>2. It must be that way.</td>
</tr>
<tr>
<td>3. I’m neutral.</td>
<td>3. I’m neutral.</td>
</tr>
<tr>
<td>4. I can live with it that way.</td>
<td>4. I can live with it that way.</td>
</tr>
<tr>
<td>5. I dislike it that way.</td>
<td>5. I dislike it that way.</td>
</tr>
</tbody>
</table>

3. One-Dimensional (O): These features correlate to customers’ satisfaction. With increasing functionality, the satisfaction increases, too.

4. Indifferent (I): The customer has not decided whether he needs this feature.

5. Reversal (R): The definition of functional and dysfunctional was inverse to the customers’ expectations. It is proposed in this situation to switch the dysfunctional and functional question [166].

6. Questionable (Q): There is a contradiction in the customer’s answer.

The basic concept of Kano et al. is to assign each product feature to one category to know the overall satisfaction influence. If respondents cause different classifications, it is proposed to choose the majority.

Kano’s model is dealing with a very important topic: How can a manufacturer learn what the customer needs in order to be satisfied? The approach of categorizing each feature into different quality levels is a widely accepted
method to rank each feature. A feature prioritization is absolutely necessary in a highly competitive industry with limited budgets. In the following, each step of administrating a Kano survey is analyzed to show up critical situations.

Feature Definition  The first step of creating a Kano survey is to define a list of potential customer requirements. The quality of this list defines the viability of the survey. The elements are typically identified by previous customer visits or surveys. These have to be administrated very carefully because on the one hand it is very important to get as much information as possible about customer needs, on the other hand it is important not to put requirements in the customer’s mouth. Assumed “must-be” requirements are the most difficult because they are exactly the ones which the customer expects but not necessarily talks about. Mistakes in the requirements list cannot be compensated by the Kano survey.

After defining all relevant requirements, the survey designer has to decide which product feature is queried. He has to keep in mind that designing a product or a new release of a product is often a trade-off process in which the product designer has to satisfy as many customers as possible. The Kano model however only allows either-or questions. If for example the question is: “How would you feel, if your car only needs three liters per 100 miles?”, no one will say that he dislikes this feature. A more sophisticated answer would be “I would like it, if it costs no more than 2.000$.”. Conditional answers are not possible in a Kano survey. This “black-white” called problem becomes more critical for conceptual features that have a high impact on subjective quality.

Having a list of potential requirements, the detail level of each feature has to be chosen. Too general features will not give any information for product designers, too technical ones are too specific for customers which can increase the “noise level” to a point where all requirements are considered indifferent (cf. [166]). The customer is not interested in how but which of his problems will be solved (cf. [114]). The Kano survey itself does not
distinguish between different levels of detail. A survey designer has to be aware of this fact and has to weight large features stronger than small ones, especially if they are mutually exclusive.

**Wording**  For each feature there is at least one pair of questions. Each pair has to deal only with one thought because otherwise it is not possible to recognize what the customer wants to say. This enforces the already discussed “black-white” problem of the Kano survey. Dependencies cannot be addressed in a Kano survey.

Additionally, the questions must be phrased very carefully. It is important to avoid polarizing wordings to not influence the answer. Technical terms have to be replaced by customer terms. If the customer needs more explanation for a feature, it is important to give this information to all respondents because that could be an important answer criterion. Depending on the survey method (postal, telephone, Internet), this is quite difficult.

The standardized answers are not less critical. There has been a lot of work dealing with this topic, beginning with different translations of Kano’s five-answers questionnaires (e.g. [150], [114]) or completely different three-answers questionnaires [39]. It has been shown that even small changes can influence the categorization. Nevertheless, Walden [166] argues that it could be necessary to completely modify the answers because the standardized answers could be misleading. In an international environment, at least linguistic adjustments are essential because of different interpretations (cf. [166]). This makes the survey results less comparable to each other.

Last but not least, practical usage has shown that respondents have to get an explanation of the way the Kano survey works and what the expected results are. Otherwise, the customers use the questions as ranking method (cf. [166]).

**Categorization**  Each feature is classified using a special evaluation table. The quality category is defined by all given answers. Kano et al. propose to use the majority which is often no adequate representation for the quality
level. Matzler and Hinterhuber [114] argue for example that the classification distribution should be transparent to decision makers: If respondents classify a feature as A with 90% and I with 10% while another feature is classified as A with 60% and I with 40% completely different decisions could be necessary. Matzler and Hinterhuber suggest to use an evaluation based on frequencies to take all responses into account. In their publication, they also mention that it may be helpful to include an importance survey to determine the relative importance of each feature.

Planning the next release of an already available product makes it additionally necessary to find out whether the quality of a given feature already satisfies the customer. In this case, the interpretation of the results is even more difficult [166]. For example, if a currently unavailable feature is rated as “must-be”, what does this mean? Have customer needs changed over time? Why are customers still buying this product? Do they just expect features that they have seen from competitors?

2.1.3 Conclusion

The discussion of the well-known Kano survey technique has shown that there is a long list of critical elements. Most of them are also seen in other survey forms. Creating reproducible and reliable surveys that take care of every important aspect is nearly impossible. This is why currently applied surveys by J.D. Power are well structured: In each survey, the same questions are asked. Thus, the analysis results are not complete, but reproducible and derived reactions are well-defined. This is important because otherwise a comparison over time would be impossible. The analysis however is only done for those aspects a customer is explicitly asked for. The structured survey form is not able to track new aspects that are relevant to the customers. Therefore, quality is defined in the view of the manufacturer or market researcher instead of the customer.

An additional shortcoming of structured surveys is the predefined list of answers. In many cases, it is not possible to categorize a design decision or a feature realization as wrong. In contrast to technical issues these short-
coming categories are neither black nor white. It might be necessary for customers to write free text. This is possible in some surveys today. The provided information is however not analyzed systematically because most answers are very short and don’t provide any background information. An engineer typically has no access to this additionally provided information although it might be necessary for him to reask, to reanalyze, and to interpret results on his own.

This discussion shows that subjective quality aspects are investigated in a very limited way. Only technical shortcomings are continuously protocoled until warranty and goodwill ends.

2.2 The Growing Importance of Social Media

During the last years, the World Wide Web has strongly changed. It is not any more an information infrastructure in which only few persons are able to publish content. It is furthermore a participation and communication infrastructure in which every user can actively take part. Today, an average Internet user spends more than 39% of his online time for communication issues. This time is consumed not only for instant messaging and emails but also for participation in different kinds of Social Media systems (cf. [32]). These systems are commonly summarized as so-called “Web 2.0” platforms. But participation systems have already existed before: Since the beginning of the World Wide Web there have been Newsgroups and Internet fora in which users have been able to interact with each other. With the increasing availability of Social Networks online participation becomes a mainstream component of the Internet.

Several studies have shown that today an increasing number of people using the Internet wants to take actively part starting their own weblog, commenting entries, managing profiles in Social Networks, etc. (cf. [163]). In Germany, this topic was analyzed by ARD and ZDF. They recognized that
in 2008 and 2009 nearly 25% of the Internet users have been highly interested in taking actively part (cf. [30, 52]). After intensive political criticism with regard to data protection in 2010, this active participation was reduced to 7%. Private communities however have not been affected and are used even more than in the years before [33]. This large number of users causes a large amount of user generated content: The largest German automotive forum – MotorTalk\(^4\) – for example has an average of 240,000 comments per month.\(^5\)

All these studies also show that user comments are not representative. The majority of authors is not older than 40 to 50 years and the author is likely male (cf. [151]). Nevertheless, user comments have a high impact on user decisions: 78% of Internet users rely on recommendations from consumers and 61% trust customer opinions posted online [125]. The impact on user decisions is the main reason why UGC is such important today. Even small web pages can cause marketing disasters which was shown by several examples during the last years\(^6\). It is therefore necessary to analyze not only the mainstream pages but also small ones. A slow reaction on critical issues can cause expensive reputational damages (e.g. the “Elchtest” problem of Daimler’s A class in 1997). Due to the very large amount of data, it is not possible to trace all comments manually. It is necessary to analyze the data with automated procedures. This is typically done using Social Media monitoring systems.

### 2.3 Social Media based Quality Experience

In recent years, a very special form of UGC has gained much attention in terms of quality analysis. This is based on the extreme success of e-commerce platforms like amazon.com. They have shown that providing recommendation systems support potential customers in finding the product that fits best to individual quality needs. In this way, it is possible to

\(^4\) http://www.motortalk.de
\(^5\) The average number of comments was calculated using the data available in 2009.
increase product sales (cf. [145, 121]). There are different kinds of recommendation systems available, based on customers’ buying behavior, their browsing behavior, provided reviews including review based rankings and finally product ratings. The last two recommendation types are the most used recommendation techniques. In most e-commerce systems, they are realized using free-text fields in addition to star-ratings (cf. fig. 2.3). While shop owners can increase their revenue offering this comment functionality [80, 62], customers are able to read product experiences of other customers and to extract individual quality aspects on their own [46]. This win-win situation leads to a general acceptance of product reviews as a cultural expression [46].

Unfortunately, customer reviews and the structured rating forms are not objective at all. In an empirical study, Hu et. al [81] have shown that
most of the reviews posted in online markets are bimodal. They tend to be ranked extremely high or extremely low. The additionally available numerical star-ratings do not convey a lot of information in this situation so that the users have to read the numerous available free-text comments. David and Pinch [46] have shown however that the majority of these comments is positively biased: Using a bug at the Canadian amazon website in 2004, the authors have identified many anonymous comments written by friends, husbands, wives, colleges or paid professionals. In addition, they have shown that many reviews are often misused for advertisement, promotion and communication issues or they are simply duplicated. According to David and Pinch, “it becomes very hard to distinguish the ‘objective’ quality of the reviews. [...] Many reviews are not authentic.” [46].

For many products there is an extremely large amount of reviews available\(^7\). Next to the qualitative problem, users have thus to handle a large amount of textual data. To address these problems, a measure is needed to distinguish high quality reviews from low quality reviews. Different shopping systems have introduced a peer-reviewed measure, in which each customer can specify whether a review was helpful (e.g. amazon.com). This rating is used as ranking method so that potential customers can focus on the most relevant user comments. It has been shown however that these assessments are influenced by a number of non-quality related factors. Ghose and Ipeirotis [62] state that reviews with more subjective elements get more helpful-votes. Liu et al. [110] have shown that “users tend to value others’ opinions positively rather than negatively”. This observation was confirmed by Danescu-Niculescu-Mizil et al. [45]. They also noticed that non-U.S. reviews at the international amazon.com web page are rated worse than U.S.-reviews and that the comment length is correlated to the number of votes.

In different publications, some further structural problems in peer-reviewed recommendation systems have been detected. Due to the implied ranking

---

\(^7\) Especially for popular or long-running products, there are often more than thousand reviews, e.g. [http://www.holidaycheck.com](http://www.holidaycheck.com)
mechanism, reviews that have already been rated as useful are more likely to be rated again. Kim et al. \cite{91} recognized that new reviews get generally less votes than older reviews. In consequence qualitative changes in a product are difficult to be recognized by potential costumers. This is a problem especially for long-running products like cars. But the problem not only occurs for new reviews, also reviews of low-traffic items have less ratings which makes it difficult to estimate the usefulness of a review \cite{91}.

Danescu-Niculescu-Mizil et al. \cite{45} showed that votes significantly depend on already available votes: Votes similar to the average rating seem to be more helpful than outliers. Votes slightly above the average rating get more helpful votes than votes with slightly worse ratings. However, the greater the variance of the ratings for a given product the more outliers get rewarded. These observations influence the complete community based concept as the raters themselves get rated \cite{45}. The integration of the author’s behavior into the helpfulness calculation, suggested for example by Huang et al. \cite{82}, has to be regarded critically in consequence.

Although user reviews are supposed to provide a valuable and reliable quality insight, it can be seen that the recommendation systems are not reliable at all. Due to monetary influence in e-commerce systems, the data is abused for marketing and thus biased.

### 2.4 Change to the Holistic Concept of Quality

In the previous sections currently available monitoring shortcomings have been discussed in detail: Especially the analysis of subjective quality aspects and the information from the post-W&G phase is incomplete. As these aspects are more and more important, alternative data sources are required. The growing number of UGC could be such an alternative: Customers already trust this data that covers many different aspects of life. Although it has to be assumed that the data cannot replace currently applied surveys,
2.4. Change to the Holistic Concept of Quality

it can at least complement them.

User comments can be created in many different Social Media systems. It has been explained that not every data source can provide reliable quality information. This is why this work is focusing on a more general type of UGC that is available in weblogs and web fora. The idea is to use data sources that are not motivated by monetary aspects. Recent work has shown that in weblogs the majority wants to share their experience or participants want to connect themselves to like-minded people (cf. [153]). Social Networks like facebook are mainly used for communication (cf. [32]). Internet fora have been analyzed in the Wave5 study [163]: The users want to be connected, to get help, to provide help and perhaps to improve their knowledge. In the following it is analyzed whether these findings can be confirmed by investigating available discussion genres in an automotive forum.

Genre Analysis The Wave5 study states that discussions in Internet fora are dealing with topics in which the authors need help or in which the authors provide information relevant to other community members. It is argued that most users want to be connected to similar minded people. This statement is now verified by investigating user comments in the automotive forum benzworld.org. For each discussion (thread) in January, 2007 the first user comment (post) was manually categorized into six abstract discussion genres (Fig. [2.4]). From 1,879 user posts more than 50% are dealing with problem descriptions. 25% of the entries are off-topic (e.g. dealing with politics) and are not relevant to automobiles at all. Only 10% of the user comments have a monetary background. All other entries are dealing with car modifications, service requests (e.g. missing manual) or the user just wants to discuss about his or her car.

These findings confirm the survey results of Wave5 [163]. The majority of user generated content in fora is obviously motivated by the authors themselves. This supports the idea of using UGC for quality analysis because

8 http://www.benzworld.org
Figure 2.4: Abstract topic analysis on 1,879 entries of the automotive forum benzworld.org. More than 50% of the posts deal with technical or conceptual problems in which the author asks for advice. 25% of the entries do not deal with any car-related topics at all and can be disregarded for quality analysis.

the data generation is mostly unbiased. In the following, it is analyzed which types of quality analyses are supported. Therefore the previously analyzed user comments from the benzworld.org community are investigated for further product characteristics.

Product Characteristics  In 73% of all user comments (77% in case of problem descriptions) the authors are mentioning the product model\(^9\) which is the elementary information for an analysis infrastructure to investigate product related topics. 75% of the textual data additionally contain the discussed components. Thus, a product analysis on component level is possible. In the automotive domain, there are however many different product improvements during the live time of a product. Without knowing the exact build year it is quite difficult to know which product version is

\(^9\) A product family (e.g. Mercedes E-class) has several product models (e.g. W210 from 1995 till 2003, W211 from 2003 till 2009). Each model can have several variants, e.g. a sedan version, coupe, etc..
2.4. Change to the Holistic Concept of Quality

Figure 2.5: The mentioned build years are distributed over 50 years while there is less data for the last four years, in which warranty and goodwill typically is offered. The number of comments decreases with increasing age.

discussed. This information is only provided by 21% of all comments (29% in case of problem descriptions). Analysis results are in consequence less detailed than it is possible with current quality sensors.

Post-Warranty Analysis  Investigating all provided build years, it can be stated that they are distributed over nearly 50 years. In consequence, UGC provides the possibility to get quality related information for products after warranty and goodwill ends. This is exactly the time for which currently used quality sensors provide less data even for technical issues. During the first 4 years the build year is not mentioned as often as for products after this age (fig. 2.5). These results are not only valid for the comments mentioning a build year. They can be confirmed investigating the mentioned product models and the corresponding build year ranges in all user comments (fig. 2.6).

Perception Discussions  Although the majority of user generated content is dealing with products for which warranty and goodwill is not offered any more, there is a noticeable amount of comments that deal with current
products. In the following it will be analyzed in which way this data can be used to gather quality related information. The analysis is based on several automotive Internet fora, which have been collected since January, 2007.

In October, 2008 Daimler started selling the SUV\textsuperscript{10} Mercedes GLK. This event is used to analyze whether users are discussing the product in the market introduction stage which started in January, 2008 with the product presentation at the North American International Auto Show in Detroit. The analysis shows that the product is moderately but continuously discussed. As there is only a very small number of customers that have been able to look at the current product, most of them are discussing although knowing only the product out of newspapers or TV shows. Thus, the discussed topics are obviously limited to perception issues. This is confirmed analyzing the 20 most discussed product features (Fig. 2.7): In addition to design issues the found comments are mainly dealing with topics that are controversial discussed in society, e.g. fuel consumption. By comparing the number of comments to discussions for Mercedes cars in general it is

\textsuperscript{10} An SUV (sport utility vehicle) is a mixture of a light-truck and a station wagon.
2.4. Change to the Holistic Concept of Quality

...and that design issues are more often discussed than it is usually done for Mercedes cars (green).

75% of the discussions compare the product to products of competitors. The comparison with the largest gap is found in discussions dealing with the trunk of the GLK that is compared to trunks of other SUVs, especially Volvo and VW (cf. fig. 2.8). The findings are not only relevant for quality engineering to find topics that are relevant to customers but also could improve marketing which might for example disprove alleged weaknesses.

But the narrow and short trunk is too tiny for an SUV. That is not even the level of the Tiguan which at least has a sliding rear bench.

Figure 2.8: An example of a received perception discussion: The product was discussed since the first presentation in January, 2008. Although the product has not been available at the time this comment was written, it was already compared to competitors.
Figure 2.9: The list of the top 20 discussed product features has changed after the market launch. While there are still perception discussions, the ratio has decreased and there are more topics related to product features customers have to deal with in daily usage.

Handling Discussions  Using the data from the previous section, the top 20 product features have been analyzed two months after the market launch of the Mercedes GLK (fig. 2.9). The previously seen perception discussions are still available. This is expected as the product had quite been new. Nevertheless, the relative frequency is strongly reduced. In addition, there are new topics related to product features customers have to deal with when driving the car.

The analysis results lead to the assumption that these handling discussions provide information about possible usability problems and unexpected behaviors. This is verified by investigating all user comments dealing with the Command System\textsuperscript{11}. The top 10 topics concern the integration of sub-components into the Command System: the DVD and audio system, the navigation system, control units at the steering wheel, handsfree units and Bluetooth, etc.. All these sub-components are known to be related to hand-

\textsuperscript{11} The central control unit that includes a navigation system and different entertainment systems.
ling shortcomings.

**Conclusion** The analysis has confirmed the basic idea that UGC is useful for quality analysis. It was shown that after warranty and goodwill is offered, the available data in Internet fora increases and could complement available data – especially for technical issues. It was also shown that perception issues are heavily discussed during the market introduction stage. From a quality engineering point of view the most interesting group of discussions is concerned with handling issues as these quality related topics are covered only by partial surveys today.

The majority of UGC is motivated by the authors themselves. There is no authority which compels to participate and asks questions. Therefore an Internet based quality analysis could systematically circumvent the problems surveys are suffering from. As UGC is generally unstructured, the users have the possibility to provide further information, which is commonly not available in surveys. The whole discussion surrounding a problem description could provide additional information about customers’ expectations but also circumstances and possible problem solutions. Thus, Social Media systems, especially fora and weblogs, are an impressive opportunity for a holistic concept of quality.

### 2.5 Definition of User Generated Content and Social Media

The term “User Generated Content” (UGC) was used until now to describe data which is available in the Internet and which was created by users. The term achieved broad popularity in 2005 when the journal Nature published a comparison between the English Wikipedia and the Encyclopædia Britannica: It was shown that UGC can provide as reliable data as it is

---

12 In some surveys, this aspect is considered using open questions. The provided answers are however not systematically analyzed as the provided text is typically very short and less informative.
Chapter 2. Problem Definition and Data Environment

done by established lexica. Although the term gained much attention in the following years, there was a lot of confusion about the exact definition. The first satisfactory definition was provided by Kaplan and Haenlein [90, p.61] in 2010:

“UGC needs to fulfill three basic requirements in order to be considered as such: first, it needs to be published either on a publicly accessible website or on a social networking site accessible to a selected group of people; second, it needs to show a certain amount of creative effort; and finally, it needs to have been created outside of professional routines and practices. The first condition excludes content exchanged in e-mails or instant messages; the second, mere replications of already existing content (e.g., posting a copy of an existing newspaper article on a personal blog without any modifications or commenting); and the third, all content that has been created with a commercial market context in mind.”

UGC based on this definition is as old as the Internet itself. In 1979, Tom Truscott, Steven Bellovin and Jim Ellis created the oldest discussion system that allowed Internet users to write user comments publicly: The Usenet system (Unix USer NETwork). The electronic network is divided into several newsgroups and is still used by Google Groups. 20 years later, in 1997, Bruce and Susan Abelson founded a community platform for online diary writers. The term “weblog” was first used in this time and was truncated to “blog” a year later. Many different systems followed until the term “Web 2.0” was firstly mentioned by Tim O’Reilly in 2004. It describes a new way in which software developers and end-users utilize the Internet to create and publish content and applications in a participatory and collaborative way. It can “be seen as an evolution back to the Internet’s root, since it re-transforms the World Wide Web to what it was initially created for: a platform to facilitate information exchange between users.” [90 p.60]. UGC is however not limited to collaborative data. With the increasing growth
of the Internet, the number of personal web pages increased dramatically since the 1990th. The majority is used for informative or entertainment purposes including biographical information, resumes and information about the author’s hobbies. According to the O’Reilly’s definition, this concept of content publishing without collaboration is called “Web 1.0”. It is out of focus for this thesis.

The term “Social Media” describes UGC created in modern software systems. It is based on the ideological concept of Web 2.0 and the social change of the generation of “digital natives” (cf. [90]). In the literature it is not clearly specified whether data from older systems like weblogs and fora is also summarized under this term. In the context of this work, both systems are explicitly included. Furthermore the analysis will be limited to these two system types as the available data can be freely downloaded using some harvesting algorithms. Additionally, the long-run discussions in these systems are expected to provide useful background-information.

2.6 Social Media Software Architectures

Since the beginning of the Internet the list of different Social Media systems increased continuously. Each system provides the possibility to publish content in a participative or collaborative way. There is no system in which the end-user is able to directly manipulate the web pages. Instead, all Social Media systems provide interfaces in which the user is able to simply enter his or her comment. Each comment is presented by showing the data on predefined page sections using templates. For this thesis, there are two relevant template types:

1. Overview Pages: This template type is used to provide an overview about currently available discussions (cf. fig. 2.10). Each discussion consists of several comments with – typically – identical topic\(^\text{13}\). The presentation is limited to a small number of comments per discussion.

\(^{13}\) Discussion pages are also known as threads.
In addition to the comments themselves, some meta data is provided, including the initial author, the publication date and the number of responders. The comments may be truncated or only the title is presented to users. Each discussion is linked to special detail pages.

2. **Detail Pages:** In contrast to overview pages, detail pages are focused on single discussions which may be distributed to several pages (cf. fig. 2.11). There are two different discussion structures possible: tree structures and list structures. In a tree structure, each comment is related to one comment following a question-answer strategy. A list view is more discussion centered and all comments are (chronologically) ordered. Modern systems use hybrid presentations (cf. fig. 2.11 (b)).

A comment is displayed in a continuous region using block level elements [1, chap. 7]. It is surrounded by comment meta data, especially the authors (nick-)name and the publication time. The comment’s length highly differs depending on the Social Media system. Analyzing more than 3,500 comments in Internet fora, there is an average length of 320 characters. The 0.25 quantile has 80.3 characters and
2.6. Social Media Software Architectures

Figure 2.11: Detail pages are used to list all comments within one discussion. In addition to the comments themselves, the templates typically provide for each comment further meta data. Similar to overview pages, there are navigation elements, advertisement areas and title sections available. The discussion can be realized as tree or list (a). Modern systems (b) use hybrid representations in which branches are sorted by user rankings.

the 0.75 quantile is 387 characters long. Within each comment, there are inline block elements (e.g. images, links) and further block elements to structure additional sub-components such as cites, tables or code blocks.

In addition to the comments, both template types include different additional page structure elements. Most common is a title section, in which the name of the Social Media system is presented, and a navigation section to jump to another discussion. To finance the operation of the Social Media systems, some type of advertisement may be additionally available. As these page elements could negatively influence the analysis results, a Social Media monitoring system should only use the comments.

Following the definition of UGC, the available data is not limited to textual information. In contrast, a very large amount of UGC in fora and weblogs
consists of videos and pictures (cf. [163, p.17]). This thesis will be limited to textual information. Within both templates types, this data is presented using an (X)HTML structure. Although the structure is well-defined (cf. [1, 2]), most generated overview and detail pages are not valid. The correct handling of invalid pages is very complicated requiring many heuristics and much experience. It is not in the scope of this work. It is assumed that the (X)HTML file can be converted into a DOM tree for further processing steps.\footnote{The open source library htmlcleaner (http://htmlcleaner.sourceforge.net/) is used within the proposed analysis architecture.}
Data Collection

UGC from Social Media systems like Internet fora and weblogs are promising many possibilities in the context of Quality Engineering (cf. chapter 2). However, the a-priori unstructured data must be analyzed in detail to extract quality related information. Several publications in the scientific world propose to use established search engines like google or bing to investigate public data available from the Internet. Such an approach is however not appropriate for this work:

- Search engines are black boxes. It is neither known if the results are complete nor which data sources are used at all. Furthermore, it is unknown if and to which extend synonyms are taken into account.

- It is not known whether search engines are deterministic. Reproducible analysis results cannot be ensured.

- Search engines are not focused on UGC. This causes noisy search results due to findings in navigation sections, advertisement areas, etc.. For optimal analysis results, irrelevant textual data has to be removed by further data cleaning algorithms.

These problems are avoidable by crawling web pages independently. In this thesis a new autonomous multi-agent based crawler architecture is pre-
sented. It is able to continuously monitor different kinds of Social Media systems and provides the possibility to control the crawling workflow depending on the focused Social Media system. This avoids the downloading and processing of unnecessary web pages which is important due to the extremely large amount of data, the high creation frequency and the limited hardware resources.

An optimized crawler architecture for Social Media crawling is not sufficient to support a high quality data analysis. The downloaded pages have to be further analyzed to extract UGC. Elements like advertisement, navigation or other design elements should be neglected to avoid low-quality results. This is a well-known problem for search engines when the term of interest is not part of the content but of the navigation section. The detection of the main content was already addressed in related work. These findings are used to automatically label potential discussion sections for which a special discussion extraction algorithm can be induced. However, there were no attempts to extract UGC until now. Therefore, a completely new approach is presented. It first applies a heuristics-based analysis in which possible comment candidates are automatically labeled. These candidates are used in a new fault-tolerant process to induce Social Media specific extraction algorithms. Both extraction algorithms can be directly integrated in the proposed multi-agent based crawler architecture.

This chapter is structured as follows. In section 3.1 related work dealing with efficient and flexible crawler architectures, content detection, content extraction and change detection are reviewed in detail. Knowing these approaches it is possible to specify the requirements for an optimized Social Media crawler system (section 3.2). These requirements are considered in a specialized multi-agent based crawler architecture that is presented in section 3.3. It allows to crawl different kinds of systems efficiently which is shown for weblogs and Internet fora in section 3.3.1. Section 3.4 presents a powerful approach to automatically detect user comments and to create extraction algorithms that can be integrated in the multi-agent based crawler architecture. As Social Media systems are actively developed and template
3.1. Related Work

3.1.1 Crawler Architectures

According to the HTTP specification [17] each web page in the World Wide Web has to be transferred to the client to be readable. This transfer is done automatically by each browser downloading exactly the data the user specifies. A crawler – also known as spider, robot or harvester – is an autonomous system that specifies the data on its own. The behavior is based on several policies (cp. [36]):

**Selection Policy** The World Wide Web is evolving fast. The very large amount of data, its rate of change as well as the limited time and bandwidth of a crawler system implies that only a fraction of all currently available web pages can be harvested [36]. In the year 2000, a study by Lawrence and Giles [103] has shown for example that no search engine indexes more than 16% of the available data. An autonomous system is forced to specify which page has to be visited next to focus on the most relevant data sources. The used relevance metric can be realized in many different ways. The most popular metric is Google’s Pagerank [29]. Hereby, the relevance of a web page is calculated using the linking of all crawled web pages. Najork and Wiener [124] suggested to use a less costly method by applying a breath-first order approach: They have shown that the most important web pages can be detected early in the crawling process.

**Re-Visit Policy** The Web continuously changes which may make the already crawled data immediately outdated [36]. To take account of relevant changes a crawler system has to revisit already analyzed pages. This can be done in two ways: Either the system keeps the average
freshness of its collection as high as possible or it keeps the average age of the pages as low as possible. Both is not equivalent because pages change irregularly. In the first case, the system has to estimate how many pages are outdated, in the second case only the age is of interest \[36\].

There have been different approaches to calculate the re-visit frequency. Besides fixed revisit rates, a broadly applied approach is to define the revisit intervals on base of domain importance and average update intervals. Akamine et al. \[3\] for example estimate the importance of a page based on internal usage statistics and the update intervals are calculated using the Last-Modified HTTP header information. The authors mention however that the header is not reliable at all. Motor-Talk – the largest German automotive community – for example does not provide any update information via HTTP headers. Additionally, not every change is relevant for further processes (e.g. advertisement). To calculate an adequate revisit frequency it is therefore necessary not only to check whether there was an update but also whether the update was relevant. Small differences have to be tolerated or the change detection algorithms have to be limited to relevant subsequences of the (X)HTML page (cf. \[111, 57, 6\]). Castillo \[36\] mentioned that it is neither optimal to use a fixed frequency for all web pages nor to calculate the re-visit frequency proportional to the change rate. Especially for pages, that change very often, a uniform policy is superior so that a mixture of both methods should be chosen.

**Politeness Policy** Crawler systems are used for many different analysis targets which makes active systems quite numerous. There are however several practical, fundamental, and ethical problems using robots \[94\]. These problems mainly concern unnecessary traffic or server overloads caused by too frequently downloaded pages, poorly written crawlers or too many different crawlers which penetrate the same host. The robots exclusion protocol \[95\] tries to solve this problem by indicating, which crawlers are allowed to crawl which part on each
3.1. Related Work

The specification does not support any information about crawling frequencies although some crawler architectures support non-standardized parameters: e.g. Yahoo supports a “Crawl-delay” parameter\(^1\).

Based on the idea of crawler friendly web servers [28], Google published in 2005 the sitemap protocol\(^2\), which is today considered by all major search engines. In contrast to the robots exclusion protocol it specifies relevant URLs including their meta data like the last change time. This information is used to flexibly adjust the re-visit frequency and therefore to reduce unnecessary crawler activities. The sitemap protocol is supported by all major fora and weblog platforms. By empirically analyzing real world Internet fora and weblogs, it has to be stated however that it is not often used.

In the literature, there are further approaches proposed to reduce server penetrations. Castillo [36] suggests to use predefined intervals for each host access. He uses 15 seconds in the WIRE project. In the context of Social Media crawling, this approach is critical as there are many communities with a higher update and creation frequency. Thus, there is more new data than can be crawled. Heydon and Najork [76] proposed a flexibly calculated frequency. They argument that web sites with a high number of relevant content typically has higher traffic. This requires a better hardware infrastructure, which could be detected by measuring the transfer rate. It is defined as the time \(t\) a page needs to be downloaded. In their system Mercator, they specified the download frequency as \(t \ast 10\).

**Parallellization Policy** Due to the very large amount of data, it might be necessary to parallelize the download process. This not only reduces the time needed to download a fraction of the web. It also maximizes the download rate because most of the time a simple crawler awaits

---

data from remote computers as part of requesting and receiving a web page \[157\]. Cho and Garcia-Molina \[40\] identified three different possibilities to realize parallelized crawler architectures. In independent crawler architectures no crawler pipeline knows anything about other ones. It is hoped that there is only a very small overlap of web pages which would be downloaded twice. An optimal politeness and selection policy cannot be guaranteed as communication between each crawler pipeline would be necessary to avoid duplicate downloads and to ensure that all relevant pages are downloaded. In a dynamic assignment there is a central control unit that specifies which crawler has to download which page. This potential bottleneck can be avoided using a static assignment in which it is predefined which crawler has to download which page, e.g. by special hashing algorithms.

These different policies can be realized in various ways. Castillo \[36, pp.35\] provided in his PhD thesis an overview about major crawling architectures until 2004. Therefore only newer approaches will be discussed in the following.

**MultiCrawler** \[73\] Harth et al. proposed a pipelined crawler architecture specialized on crawling semantic data. In contrast to classical crawler architectures that are focused on the performance, MultiCrawler focuses “on detecting Semantic Web data, the transformation […] to RDF and the indexing” \[73\]. This is done in five consecutively arranged steps: The **fetch phase** downloads the data, the **detect phase** detects the format the semantic data is provided using the MIME type or the file extension. The **transform phase** transforms the data to a standard RDF representation that is indexed in a specialized **index phase**. In the last phase, URLs are extracted and passed to the fetch phase.

**IRLBot** \[104\] Lee et al. focused on a non-parallelized crawler approach. They identified three different aspects to increase the overall crawler
3.1. Related Work

performance. The first aspect is the process of checking an URL for its uniqueness. Due to the increasing number of spam pages available today it is additionally necessary to detect their uselessness to prioritize important and relevant pages. Last but not least a dynamic politeness policy is required not only for a given domain but even for the complete host server, because today there is normally more than one domain per server. Lee et al. propose to include the domain importance into the politeness policy based on the idea that popular pages provide better server infrastructures. The main approach provided by Lee et al. is the new hashing algorithm DRUM (Disk Repository with Update Management) that is able to handle the previously mentioned aspects in a very efficient way, so it is possible to download billions of web pages with a single crawler system.

Zhu et al.’s Ring Crawler [178] Zhu et al. proposed a decentralized parallelized crawler approach that uses a special network infrastructure to reduce communication overhead. All crawlers are arranged in a ring in which each crawler agent has a previous and a following crawler. If a crawler gets an URL it is not responsible for, it will use a special hashing algorithm that indicates in which direction the URL has to be communicated. If the receiver is also not responsible, the procedure continues.

Zhu et al. are not handling crawler failures. Therefore, it could be possible that no crawler is responsible for a given URL and the communication would never end. Inserting new crawler agents will cause shifted responsibilities which again could cause duplicate downloads because the new crawlers are not aware of already downloaded pages. This problem was already discussed in the UbiCrawler architecture [24] that uses a consistent hashing algorithm: The assignment of an URL to the crawler agent is fixed in their approach.

Akamine et al.’s Large Scale Web Crawler [3] Akamine et al. proposed a crawler architecture that downloads data from different types of web pages, namely an RSS crawler, a news crawler and an universal
crawler. Each crawler is applied parallel to the others without any communication between the systems. The politeness policy is realized using an additional proxy server that blocks too many downloads for each host. Akamine et al. propose to combine the probability that the page is important and the probability that there was an update for an improved re-visit policy. The first probability is calculated using internal usage information. The second one is based on the average number of pages that have been updated until now. This information is based on the Last-Changed HTTP Header. The authors mention that this method is not perfect because there could be smaller changes (e.g. advertisement) although the main content is equal. Therefore the content has to be compared additionally.

**Hsieh et al.’s Extensible Web Crawler** [78] Hsieh et al. propose a crawler as a service architecture. In contrast to classical crawler architectures that download different pages and store the content for further processes, the proposed architecture applies different predefined filters that notify registered processes about newly downloaded pages. The filtering is applied on the fly so that it is not necessary to store downloaded pages. Each registered process is responsible on its own to handle new or updated pages.

In addition to these general architectures, there have been specialized crawler approaches based on the idea of focused crawling [38]. To handle the huge amount of data available in the World Wide Web, these crawlers download pages only related to a predefined topic. This strategy is only possible if the relevance of a document can be a priori estimated. Chakrabarti et al. [38] proposed to use a classification algorithm based on manually labeled data. A different approach is to apply different meta data extraction algorithms. The available information is used to estimate the relevance of an URL (cf. [50, 49]). With the increasing availability of semantic structured data, some promising approaches have been published to estimate the relevance of a page using ontology structures (cf. [51]).
3.1.2 Content Extraction

A modern web page not only consists of relevant textual data but provides further layout elements like a title bar, navigation elements, link lists or advertisement sections. Applying any kind of analysis on this data produces suboptimal analysis results due to the noisy nature of the data. An example is the creation of a search index, in which the navigation section is irrelevant as the user wants to get page hits within the main content. It is thus necessary to further process the data to separate the central information of a document [63]. This process is called **Content Extraction** (CE). Gottron [67] defines CE as “the task to identify the main content in a document”. He mentions that most publications are not defining which elements exactly belong to the main content. Using some examples he shows that indeed this definition cannot be provided in general. It is a task dependent definition:

> “In most cases [the main content] will be text as text still is the predominant method to transmit information in the WWW. [...] A photo community will have pictures as main content for most documents, video portals deal with movies or animated images. Software collections will provide source code, while digital online libraries will feature electronic documents, more likely in a PDF format than in a pure HTML format.” [67, pp.83].

By limiting CE to textual content in documents or web pages, the main content is nothing else than a subsequence of the textual data. Gottron proposes to use offline annotations to mark the positions of the substring [67, p.87]. He defines CE formally as a function $f_{CE}$ with:

$$f_{CE} : D \to \{(s_i, e_i) : 1 \leq s_i \leq e_i \leq |D|, i \in I\}$$  \hspace{1cm} (3.1)

where $D$ is the source code of a document represented by $|D|$ characters, $s_i$ and $e_i$ are the start and end position of the main content. The index $i \in I$ allows the main content to be split up in several parts.
Liu [106] distinguishes three approaches to extract the main content out of textual data [106, p.323]: manual wrapper creation, wrapper induction and automatic extraction.

**Manual Wrapper Creation** The most appropriate extraction algorithm can be created by manually analyzing a number of different documents and by creating patterns that denote the relevant content that should be extracted. An extraction algorithm that applies these patterns is called a wrapper algorithm. Many different specialized wrapper programming languages have been proposed to support this approach like TSIMMIS [70], Jedi [85], Minerva [43] or the Lixto Project [66]. Limiting the wrapper generation to HTML pages, even tree-aware approaches like XPath\(^3\) can be applied (cf. [5, 31]). The manual approach causes however high efforts not only in wrapper creation but also in wrapper maintenance. It is therefore known to be not applicable for a large set of different analysis targets.

**Wrapper Induction** A more automated approach is the use of labeled documents to automatically learn extraction patterns with super- or semi-supervised algorithms. The labeling process is manually done using *landmarks* which delimit relevant fragments. Different systems have been proposed for an integrated labeling and induction process, e.g. WIEN [97], Stalker [86] or SoftMealy [79]. The labeled data set is typically larger than the data set which is manually investigated in a manual wrapper generation. This causes much more robust wrapper algorithms (cf. [98]).

**Automatic Extraction** Due to the very fast increasing amount of data in the Internet, automatic extraction algorithms have gained much attraction in the last years. Gibson et al. [63] have grouped the available techniques into two families. *Local* techniques are applied only

---

\(^3\) XPath (XML Path Language) is a W3C specified language for selecting nodes from an XML document. [http://www.w3.org/TR/xpath20/](http://www.w3.org/TR/xpath20/)
3.1. Related Work

on single pages without any reference to other pages. To be applicable, these techniques need some heuristics to detect relevant and irrelevant page elements. *Global* techniques on the other side consider a collection of web pages to identify reoccurring page elements. The approach benefits of the increasing use of template based web applications during the last decades.

Gottron [67, chap.3] has surveyed local and global techniques to automatically identify the main content in different types of web pages. The most relevant approaches for this thesis are discussed in detail in the following. They will be extended to an unsupervised content identification and comment segmentation algorithm for Social Media systems so that it is possibly to provide high quality analysis results.

**Site Style Tree**

Yi et al. [171] proposed a Template Detection (TD) algorithm that tries to identify redundant page elements by comparing visual elements of a set of web pages from the same origin. The global analysis technique generates a relevancy information for each text and (X)HTML node which is used to eliminate irrelevant page elements. Thus it is possible to focus on the main content. This *Site Style Tree* (SST) named algorithm is applied to improve classification results for product descriptions.

The analyzed elements are limited to the body content of each (X)HTML page because the header does not contain any visual information. These tags – represented as a DOM\(^4\) – are transformed to a Style Tree. It contains two different kinds of nodes: style nodes and element nodes. Style nodes capture the entire sequence of sibling elements occurring at a given level of a sub-tree in the DOM. Element nodes correspond to DOM nodes as they contain the name of the element and its representation related attributes including their child style nodes.

\(^4\) A DOM (Document Object Model) is the standardized representation model for (X)HTML and XML documents.
Figure 3.1: The Site Style Tree is generated out of the DOM representation of different web pages of the same origin (based on [171]). Each DOM node is transformed into an element node and each sequence of nodes represents a style node. By collecting all style nodes from several pages, the Site Style Tree (SST) is created.

The SST algorithm first generates a Style Tree for a single document – so called Page Style Trees (PST). These are created by transforming each DOM node into an element node and each sequence of nodes into a style node. The SST is created by combining all PSTs: All possible child style nodes of an element node are collected and attached to the element node in conjunction to their occurrence frequency (Fig. 3.1). Two style nodes are equal if and only if they have the same sequence of element nodes. This mechanism makes the SST algorithm very efficient because it is not necessary to recursively compare sub-trees.

To identify template structures and therefore to reduce noisy data for further processes meaningful nodes in the SST have to be identified. That is done by calculating an importance value for each element node. Depending on the presence of child nodes, this importance is differently defined:

**Leaf Nodes** Element nodes with no further style nodes are called leaf nodes. Their importance is only based on their content. The more the
content differs and thus the more different contents are available, the more important it is. For each leaf node $E$, the importance can thus be calculated using Shannon’s entropy: $l$ is the number of different features, $m$ is the number of documents $E$ appears in. $p_{ij}$ denotes the probability a feature $i$ appears in the context of $E$ in page $j$.

$$Imp(E) = \begin{cases} 
1, & \text{if } m = 1 \\
1 + \frac{1}{l} \sum_{i=1}^{l} \sum_{j=1}^{m} p_{ij} \log_{m} p_{ij}, & \text{if } m > 1
\end{cases}$$ (3.2)

Features may be tokens, $n$-grams or phrases. Yi et al. have not specified which type of feature was used in their work. In this thesis, the analysis is done using tokens. A further analysis of these features (e.g. stemming, lemmatization, etc.) is not necessary as differences would be relevant for the node importance.

**Tree Nodes** The more different style nodes are available in an non-terminal element node, the more important it is. Yi et al. use an entropy formula to deduce a presentation importance of a node $E$ that appears in $m$ documents. $E$ has $l$ child style nodes, $p_i$ is the probability for the $i$-th of these nodes to appear with $E$:

$$PresImp(E) = \begin{cases} 
- \sum_{i=1}^{l} p_i \log_{m} p_i, & \text{if } m > 1 \\
1, & \text{if } m = 1
\end{cases}$$ (3.3)

The importance of a tree node however depends not only on the direct child elements. Therefore it is necessary to define the importance of a node $E$ recursively. Yi et al. propose to use a convex combination:

$$Imp(E) = (1 - \gamma^l)PresImp(E) + \gamma^l \sum_{i=1}^{l} (p_i \ast Imp(S_i))$$ (3.4)

$\gamma$ is stated to be an attenuating factor which is set to 0.9 in the original paper, $l$ denotes the number of child style nodes of node $E$. The probability $p_i$ denotes how likely the style $S_i$ appears in node $E$. 
Imp(S_i) is the importance of a style node S_i which is defined as the average composite importance of the contained element nodes:

\[ \text{Imp}(S) = \frac{1}{k} \sum_{j=1}^{k} \text{Imp}(E_k) \]  

(3.5)

The composite importance \( \text{Imp}(E) \) is calculated for each node beginning with the body node. If the importance of a node is lower than a predefined threshold \( t \), it is assumed to be non-relevant.

To eliminate the noise in a given web document, this document is transformed to a PST and the elements are matched to the elements in a SST. The noisy nodes (\( \text{Imp}(E) < t \)) are deleted while relevant nodes are kept.

**Largest Size Increase**

Han et al. \[71\] proposed a local analysis algorithm to locate the primary content. They developed three different heuristics: largest tag count, highest fanout and largest size increase. The last heuristic is intended to find a single node containing the textual main content by calculating for each node how much visible text is added to the document. The node that relatively adds the most content is assumed to be the node that contains the main content. Gottron \[67\] uses this heuristic to remove all text nodes that are not recognized as main content.

**3.1.3 Item Separation**

Web crawlers are often applied to crawl news articles which are common analysis subjects in the scientific community. These articles can be categorized by a title, a teaser and one large text. The main content in a Social Media system is different as it consists of several user comments (cf. section 2.6). A high quality analysis of UGC enforces to separate the main content to these comments. Otherwise, irrelevant user comments\(^5\) or topic changes

\(^5\) e.g. off-topic comments within one discussion
within one discussion involve the risk of a noisy data base and thus poor analysis results. It is however not only important to separate the comments but to keep the dependencies and relations as they support the analysis of a discussion, e.g. to detect quotes or discussed topics.

The separation of user comments has not been focused on in related work so far. The only work in the context of UGC was published by Cao et al. [35] who try to separate the post and the comment section in weblogs. It was however not analyzed in which way each comment can be separated.

The process of content segmentation is focused however by several researchers that try to extract product details in semistructured web pages (e.g. online shops). The concepts and observations are important for this work although the approaches are not directly applicable in the context of Social Media systems. This is due to very different structures of user comments in contrast to very similar structured product details in product lists.

Similar to the CE task (cf. section 3.1.2), there are two possible methods: supervised approaches to learn extraction rules (wrapper creation and wrapper induction) and unsupervised approaches. The latter can be separated into global and local analysis techniques [122]. Global analysis approaches are not applicable in the context of this work as the user comments highly differ in content and representation\(^6\). Global techniques rely however on similar records (cf. [175]). Supervised methods cannot be applied because manually labeled web data is insufficiently available. Thus, the following discussion is limited to local techniques.

One of the first approaches to extract data records via local techniques has been published by Buttler et al. [34]. They locate relevant objects with a two-step algorithm by first identifying the minimal subtree that contains all relevant objects and than identifying possible object separators. In both steps different heuristics are applied without the usage of domain

\(^6\) In most Social Media systems, it is possible to add lists, images, URLs, tables, code and quote sections, etc.. Additionally, the length of a comment and the number of paragraphs is not limited.
knowledge, e.g. the content size, the tag count, etc.. Liu et al. \cite{107} extend this approach with improved heuristics. They proposed an often cited algorithm that surpasses the results of previous works. It is based on two observations:

1. Data records that contain similar objects are typically presented in particular regions of a web page using similar HTML tags. This region is called data region. In case of product lists, each data record is formatted using almost the same sequence of HTML tags.

2. A group of similar data records in a HTML page can be found under one parent node in the HTML tag tree. Within this parent node, it is very unlikely that a data record starts inside one child sub-tree and ends inside another one. Instead one data record commonly starts and ends at the same child sub-tree.

Liu et al. proposed the MDR algorithm (Mining Data Records) that takes advantage of these observations by using the HTML tag structure. The approach first introduces virtual nodes that contain a collection of adjacent HTML nodes with an identical parent node. These so called generalized nodes will be grouped to data regions if all generalized nodes in the region are adjacent and have the same number of HTML nodes. In this way, it is possible to capture data records that are split into several HTML tags. The generalized nodes are possible data records. It is assumed that they are presented similarly and thus, each record can be identified by comparing the HTML nodes on the same tree level recursively. Liu et al. suggested to use string matching algorithms. The presence of further product information (e.g. discounts) is tolerated by accepting small differences. Zhai and Liu \cite{175} improved the identification of data records by using additionally visual information and a partial tree alignment algorithm.

Mundluru et al. \cite{122} criticize the MDR algorithm as not effective, especially when the degree of regularity across the records is not very high or if there are additional data regions, e.g. for advertisement. They identified
one requirement of the algorithm that makes MDR not applicable for many problems: all generalized nodes which collectively comprise a region must have the same length. Mundluru et al. proposed a three-step algorithm that is more flexible. It is based on the observations published by Liu et al. extended by an additional observation noting that the sub-tree paths of \( n \) records under a parent node are identical. This observation is of course a logical consequence in HTML trees. In the first step, the algorithm tries to discover potential parent nodes by searching leaf node paths with at least \( K \geq 2 \) identical structures. After identifying possible data regions, a robust string matching algorithm based on an edit distance is applied to identify possible record paths. To support different lengths, a threshold is used. It is dynamically calculated using sub-tree characteristics. It is however not mentioned what characteristics are exactly used. It can be stated however that there is still a large similarity necessary. In the last step, there are some heuristics used to identify the target records, e.g. the number of characters.

Alvarez et al. [112] confirm the observations of Liu and additionally note that each attribute has the same path from the root node in several data records. In their proposed two-step algorithm, they first try to detect the main record list in a web page searching these text nodes with identical path to the root node. The identified data region is separated into individual records by applying an edit distance measure that compares potential record candidates.

Kim et al. [92] proposed a more flexible similarity measure by weighting node types differently to tolerate inline elements (links, images, etc.). This makes it possible to be more flexible for additionally provided information than in previous presented publications.

All discussed approaches are limited to detect data records that are presented in a list form. Liu and Zhou [109] focused on detecting nested structures by post-order traversing a tag tree. Algur and Hiremath [4] proposed a more visual analysis algorithms using gaps and spaces to detect nested record structures.
3.1.4 Detection of Template Changes

As discussed in the previous section that content detection and the resulting content extraction is a time consuming task which is why wrapper algorithms are typically used to improve scalability. The created wrapper algorithms have to be verified regularly to ensure that their application still extracts valid data items. Kushmerick [98] proposes an easy approach that is based on the standard software engineering regression testing paradigm (cf. [123]): Applying the wrapper algorithm on an already downloaded and processed document, the result can be compared to the stored data. The wrapper algorithms are still valid if the results are identical. An exact match however can not be expected because the content may have changed, too. Therefore wrapper verification has to deal with two problems simultaneously: changed content and changed representation. Kushmerick proposes the domain-independent, heuristic algorithm RAPTURE to handle this problem. It uses a limited set of predefined numeric features, namely “the numbers of characters, the fraction of punctuation and upper-case characters, etc.” [98] to tolerate content changes. Alternatively, it can be imagined to use already discussed text similarity methods that limit the similarity calculation to special HTML parts (cf. section 3.1.3). All these methods are however not able to tolerate changes in which complete new comments are available.

3.2 Requirement Analysis

An appropriate data collection process for Social Media analysis has to focus on several aspects that are not considered in related work. This section discusses these aspects in detail which have to be fulfilled in a specialized crawler system.

First of all, the analysis of Social Media systems is focused on UGC to extract quality related information in this work. Thus, the data collection process is not intended to provide complete web pages but user generated data. The most important requirement concerns in consequence the data
3.2. Requirement Analysis

The data collection process must be able to extract the relevant data to support optimal analysis results. Advertisement, navigation and other page structure elements are not of interest and could negatively influence any analysis due to word occurrences in these sections\textsuperscript{7}. The extraction highly depends on the Social Media type (e.g. weblog vs. forum) and of the Social Media system as each system has different designs and thus requires different extraction methods. To uniformly handle different kinds of Social Media systems, a universal data type definition is required that is able to represent user comments, the relation between comments and additional available meta data (e.g. author, date).

The second aspect is the immense amount of possibly relevant data and its high growth rate. Due to limited bandwidth and hardware resources, a crawler system has to exploit special properties that are available in Social Media systems that differ to approaches used in classical web crawler systems. The resulting requirements regarding the crawler architecture can be discussed using the crawler policies introduced in section \ref{sec:related-work}:

**Selection Policy** The selection policies presented in related work (cf. section \ref{sec:related-work}) are not applicable in Social Media systems because it is a-priori not possible to identify discussions and comments with relevant and important content. A paradigm like “good pages are seen early in the crawling process” \cite{36} or the importance definition using the linkage structure \cite{29} are definitely not applicable. On the contrary, each discussion and comment may be important for different types of analyses and it is therefore not possible to assign an importance factor to unseen data. Nevertheless it is possible to specify which page should be loaded next. Therefore, all Social Media systems provide overview pages or special syndication formats (RSS – Really Simple Syndication) which can be used to extract the URLs with the latest changes. The concrete technique depends however on the Social Media system itself. A universal crawler architecture must

\textsuperscript{7} This problem can often be seen in search engines in which the search term is not found in the content section but in the navigation area.
Chapter 3. Data Collection

thus provide a flexible approach in which the selection policy can be defined for each Social Media system separately.

Re-Visit Policy There are two different scenarios in which a re-visit policy has to be applied: On the one hand it is necessary to monitor overview pages (or syndication files) to identify new discussions, on the other hand it is necessary to detect changes in already visited discussions. While the re-visit rate for the first scenario can be dynamically calculated using the average discussion frequency, the second aspect is more critical as the huge amount of different discussions makes it difficult to re-crawl all discussion pages. Analyzing 2 million randomly selected discussions out of Internet fora shows that the average discussion takes 15 days.\(^8\) Thus it is possible to define the re-visit frequency depending on the age of the discussion. Discussions have to be re-visited more often during the first 15 days than older discussions.

To support the calculation of dynamic re-visit frequencies in both scenarios, a Social Media crawler must be able to extract date information on overview and detail pages. It is typically not available via a HTTP header information because the pages are dynamically created.\(^9\) In general, there is even no sitemap protocol used which makes it necessary to extract comment dates out of the web pages themselves. The way this can be achieved depends on the Social Media system again. A crawler architecture must provide a mechanism to extract and store relevant date information to support flexible re-visit calculations.

Politeness Policy A crawler system should never overload a host. Therefore different approaches have been proposed to flexibly specify the

---

\(^8\) This is the same duration that is typically used in weblogs to intensify spam analysis activities (e.g. in Spam Karma 2 – [http://unknowngenius.com/blog/wordpress/spam-karma/](http://unknowngenius.com/blog/wordpress/spam-karma/) – accessed on 01-13-2011)

\(^9\) This was tested for common community systems like phpBB, vBulletin and wordpress. For all systems, there are extensions available that provide additional date information via HTTP headers, but these extensions cannot be expected.
load balance per host like fixed load frequencies or dynamically calculated frequencies based on the average load time of a web page (cf. section 3.1.1). Social Media systems have to deal with an additional problem: spam. On the one side, there is a large number of specialized crawler systems that try to extract email addresses for spam e-mails, on the other side there is a large number of spam comments. To limit unnecessary computation times, many Social Media systems analyze visitors by investigating their user agent information and individual user behavior (e.g. BadBehavior\textsuperscript{10}). If a crawler agent acts like commonly available crawler systems or if the crawler loads too much data in too short time, the data provider block the access for a predefined delay except for white listed IP ranges. Spam protection algorithms have to be considered in consequence by an appropriate Politeness Policy.

Limiting the crawler process to pages that contain UGC, an optimized Politeness Policy is possible. Overview pages for example are commonly not very interesting for Social Media analyses except to detect discussion URLs. Unfortunately, it is not possible to a-priori specify whether and which type of URL can be neglected for further processing steps. This has to be defined for each Social Media system separately. Furthermore, the overall download flow within one Social Media system should be controllable to ensure a non-classical crawler behavior.

**Parallelization Policy** Downloading a web page, the largest part of the consumed time is spent for awaiting data from remote servers. This makes it necessary to parallelize a crawler system to allow parallel download processes. This distribution is not only limited to different hardware elements but also to different hardware locations to take advantage of additional bandwidth capabilities. Due to the very large amount of UGC available in Social Media systems it is however also necessary to distribute individual processes to take advantage of im-

\footnote{http://bad-behavior.ioerror.us/ – accessed on 01-13-2011}
proved computing time. An appropriate Social Media crawler system has to support the distribution of each analysis process – no matter whether the target selection is concerned, the harvesting process, the content detection or the content extraction.

Summarizing all these requirements it can be seen that the target selection, re-visit frequency calculation and data extraction highly depends on the target system that should be crawled. Due to the different structures and designs of the Social Media systems, it is impossible to use just one algorithm\textsuperscript{11}. An appropriate crawler system has to support the application of individual algorithms while at the same time the interaction of all components have to be controlled to guarantee an optimal Politeness Policy.

As example assume having the weblog found at \url{http://www.shopblogger.de/blog/}. At this domain, an appropriate crawler just has to download the RSS feed\textsuperscript{12} which can be found at \url{http://www.shopblogger.de/blog/feeds/index.rss2}. This feed provides the full content which can be extracted using a standard RSS parser. A crawler for the weblog \url{http://www.mathias-bank.de} can also use the RSS feed found at \url{http://www.mathias-bank.de/feed/}. This feed however only provides excerpts which make it necessary to additionally crawl each linked page and than to extract the content by defining the HTML tag that contains the content. It can be seen that a successful and optimal crawling of both Social Media systems needs an individual download flow and an individual extraction process although both systems are weblogs. This individual approach ensures an optimal Politeness Policy as only required pages are downloaded. The optimal re-visit frequency can be calculated using the information provided in the RSS feeds. An appropriate crawler architecture has to support these possibilities.

\textsuperscript{11} At least different parameters are necessary.

\textsuperscript{12} RSS (Really Simply Syndication) is a web standard to publish data. In addition to a textual description of the web page, the author, the publication time and the corresponding link is typically provided.
3.3 A Blackboard Crawler Architecture

The requirements proposed in section 3.2 enforce a pluginable architecture that can be easily extended for further Social Media systems. To support the extension without recompilation or even shutdown cycles, a special multi agent concept is used to develop an appropriate crawler system: the blackboard architecture. It can be described using the following metaphor:

“Imagine a group of human or agent specialists seated next to a large blackboard. The specialists are working cooperatively to solve a problem, using the blackboard as the workplace for developing the solution. Problem solving begins when the problem and initial data are written onto the blackboard. The specialists watch the blackboard, looking for an opportunity to apply their expertise to the developing solution. When a specialist finds sufficient information to make a contribution, he records the contribution on the blackboard. This additional information may enable other specialists to apply their expertise. This process of adding contributions to the blackboard continues until the problem has been solved.” [168, p.103]

The multi agent model allows to realize highly specialized agents that interact with each other by simultaneously reducing communication needs. At least in the classical model, each agent communicates just with the blackboard. It is however possible to extend this approach using additional negotiation agents or centralized managed agents. On the blackboard it is possible to store any object type.

A general crawler architecture based on the multi agent approach needs five different kinds of agents:

**Target Selection** Following the Politeness and Selection Policy, the first step for each crawler architecture is to specify the target URLs which have to be downloaded. In a multi agent system, this is done using
Chapter 3. Data Collection

specialized agents. They create new objects of type URL and transfer these objects to the blackboard. Each object consists of two attributes: the URL itself and the expected data type (e.g. RSS feed, forum thread, robots.txt file, etc.). Both attributes are needed by further agents to check their responsibility.

In the simplest case, one target selection agent creates URL objects based on internal lists while an additional target selection agent may extract URLs out of downloaded web pages. By using specialized target selection agents for each Social Media system, it is possible to limit the crawling process to relevant pages (e.g. the one with the latest changes). The agents can create corresponding URL objects according to the Re-visit Policy and Selection Policy. This supports an optimal Politeness Policy as only relevant pages are downloaded.

Duplicate downloads can be avoided using static or dynamic assignments (cf. [40]). In case of static assignment it is necessary for the blackboard architecture itself to assign each target selection agent a responsibility range. In case of dynamic assignment it is necessary to allow direct communication between the target selection agents. In both cases the blackboard approach is relaxed as there is an additional communication next to the data objects. The implemented crawler architecture realizes a static assignment, in which the blackboard architecture defines the ranges. Within each target selection agent, hash tables are used to avoid duplicate URL objects at the blackboard.

Legal Audit Downloading web pages affects legal rights – especially the copyright. It is therefore necessary to ensure that a given URL is allowed to be downloaded for further processing steps. Depending on the data type, different steps are necessary.

Whether an URL is allowed to be downloaded depends on the definition in the robots.txt file, specified by the robots protocol [95]. To ensure that an object \(O\) with a given URL is allowed to be downloaded, a special legal audit agent has to compare the URL with the definition provided in this file. If this file is not available or if this file
is not up-to-date, the agent has to store the URL for the `robots.txt` file on the blackboard for a separate download and analysis process by additional agents. The object $O$ cannot be processed until the `robots.txt` results are available. If the results are available and still valid, the URL of object $O$ can be compared. The legal audit agent has to put the URL object back to the blackboard so that the page can be downloaded by an harvesting agent. To prohibit an additional check by the legal `robots.txt` legal audit agent, an additional attribute is added to the object.

In case of (X)HTML files, there is an additional possibility that restricts a page for processing issues. A web page can provide special META tags that restrict a crawler architecture to index the page\(^\text{13}\) or to follow additional URLs on the page\(^\text{14}\). Following the multi agent approach, a second legal audit agent is necessary to check these tags. After an (X)HTML file was downloaded (using an harvesting agent), the page has to be taken from the blackboard. If the objects can be further processed, the agent has to put them back to the backboard with a corresponding attribute which indicates the further usage.

**Harvesting** The core algorithm in each crawler architecture is the process of downloading a given URL. Today, there are many different libraries available that support this process\(^\text{15}\) which is why it is not discussed in detail.

A harvesting agent needs an object of type URL that is allowed to be downloaded. After downloading the URL, the result is stored to the blackboard as object `Page`. The object has at least two attributes: the content and the data type which is based on the data type of the URL object.

According to the Politeness Policy, it is necessary to limit the host penetration to ensure that the crawler is not blocked. This topic has to

\(^{13}\) `<meta name="robots" content="noindex" />`

\(^{14}\) `<meta name="robots" content="nofollow" />`

\(^{15}\) In this thesis, the Apache Commons HttpClient was used \([\text{http://projects.apache.org/projects/commons_httpclient.html}]\)
be focused by the harvesting agents. Each agent is responsible for only one host \( h \). The average download time \( t \) per host is used to estimate whether additional agents for the same host \( h \) can be created. If \( t < \tau \) (\( \tau \) is a predefined limit) the blackboard infrastructure is informed that it is possible to initialize additional harvesting agents for \( h \). If \( t \geq \tau \) the infrastructure is informed that there are too many agents. This behavior allows to flexibly adjust the download rate per host. By first reducing agents which overload a host, the overall download rate is optimally adopted to the available bandwidth. If there are free capabilities, new harvesting agents can be created.

**Processing** The downloaded data has to be processed by additional algorithms. According to the multi agent concept, each algorithm is realized by a specialized processing agent. Each of them depends at least on the data type (e.g. weblog, forum, etc.).

A general usable processing agent is the robots.txt processing agent. It requires a Page of data type robots.txt. If such an object is available at the blackboard, the processing agent can take the object and analyze its content. The result might be stored to the blackboard which can be gathered by the legal audit agents. It is however also possible to allow direct communication between the processing agent and the legal audit agent, relaxing the classical blackboard architecture as the receiver of this data is known.

In case of (X)HTML pages the textual data has to be transformed into a DOM representation. This task is non-trivial because many pages are not realized according to the web standards. There are different algorithms available to support this task\(^{16}\). They can be integrated into the processing agent. The output of the processing agent is a standard format which can be easily analyzed by further XML based analysis algorithms. The agent is therefore applied to all objects based on (X)HTML structures.

\(^{16}\) We used the htmlcleaner project [http://htmlcleaner.sourceforge.net/](http://htmlcleaner.sourceforge.net/)
Content extraction and evaluation are further processing steps that have to be done by analysis agents. They highly depend on the concrete page type. Some examples are discussed in the next sections.

Storage After downloading a page and processing the data, there has to be a storage process. The most simple storage process stores the downloaded page to the file system. More sophisticated data structures include the discussion structure and comment relations which depend on the processing steps and on the concrete page type. The corresponding storage agents cannot be described in general.

For the proposed multi agent blackboard architecture, it is not necessary to define any workflow explicitly because it is implicitly given using pre- and post-conditions for each agent. This makes it quite easy to extend the workflow with additional specialized agents for new Social Media systems while general agents can be easily reused without the need to re-implement code or to create individual workflows. That makes the crawler architecture quite flexible for different types of data sources.

It is obvious that different agents need different computing time to fulfill their tasks. Some tasks may cause an analysis bottleneck in which it is necessary to first process the unfinished tasks before new tasks can be created. An example is the harvesting agent: Although the download process may be quite fast (if enough bandwidth is available) the task is much slower than the creation of URL objects. In a multi agent blackboard architecture, the problem can simply be solved by creating several agent instances. The blackboard architecture can directly control the agent creation and termination by analyzing the available tasks at the blackboard including their creation and end time. In addition to this agent specific parallelization and distribution, it is also possible to parallelize the complete crawler system as proposed in related work (cf. section 3.1.1). There could be multiple blackboards whose responsibility for different domains is defined using static assignment approaches (e.g. similar to the Ring Crawler [178] or the UbiCrawler [24]). The multi agent blackboard paradigm therefore provides
a granular scalable architecture. Due to the agent-based approach, it is even robust towards unexpected behaviors in which individual agents may crash. The overall crawler system would not be influenced, the blackboard just needs to instantiate new agents.

It can be summarized that the proposed architecture is able to fulfill the defined requirements if there are agents that are specialized to Social Media systems. In the next section, two special Social Media systems are discussed in detail: weblogs and Internet fora. It will be shown that the crawler architecture supports an optimal crawling process for both Social Media types with minimal implementation requirements as general agents can simply be reused.

### 3.3.1 Social Media Agents for Weblogs and Internet Fora

Using the multi agent crawler system proposed in the previous section, it is possible to specify specialized Social Media agents. This is done for weblogs and Internet fora in the following. In both cases, only UGC is of interest. The data must be stored in a generalized data structure that is applicable for a very large number of different Social Media types.

#### Comment and Discussion Representation

Figure 3.2 presents an UML class diagram that realizes a universal data structure for UGC. The core element is the user comment itself. It is typically associated with an author and a publication date. But there are also additional attributes possible, e.g. a signature, the number of edit times, tags or community votes. It is advisable to store this information although it is not used in the currently applied analysis process. A recrawling is more difficult as the huge amount of data needs a lot of time until all user comments will be updated. A general data representation for UGC requires a list of a-priori undefined name-value pairs to enrich comment information with additional attributes. Common attributes have to be mapped to iden-
tical names (or URIs) to be comparable. The author is represented as a separate object with a nickname and an author ID so that it is possible to get all comments of the same user.

In most Social Media systems, comments are grouped into discussions. There are two different realizations: linearly according to the publication date and nested according to answer relations. The comment relation should be stored as attribute of the comment as it provides relevant analysis information. It can be used for example to detect cites. The storage is done using a so-called parent model: Each comment links to the comment it is related to. If the discussion is linearly realized, each comment except of the first one is connected to the preceding comment. User comments without a discussion structure are realized as discussion with only one comment. In systems with articles (e.g. weblogs), the first comment represents the article.

In addition to comment relations, currently available software systems structure discussions using categories or tags. Both classification methods provide relevant information about the topic of the discussion. In the automo-

Figure 3.2: The UML diagram presents a universal class structure in which it is possible to store UGC from different Social Media types. The central element is the comment, which is linked to a discussion. Each discussion is linked to a Social Media system using at least one category.
tive domain, it is for example used to provide the product model, which is often not mentioned in the comments. As the tags and categories are used for multiple discussions, they are stored normalized\textsuperscript{17} and not as name-value attributes as it is done for comments.

**Weblog Agents**

A weblog is a type of webpage usually displaying date-stamped entries in reverse chronological order\textsuperscript{127}. Today, there are many different software systems\textsuperscript{18} and even hosting services\textsuperscript{19} available that make it easy to run a weblog. Each software system and all available hosting services provide RSS feeds. RSS is a well-documented XML format\textsuperscript{20} in which the latest weblog entries are listed. The weblog crawler agents are using these RSS feeds. The implicitly given workflow for this Social Media type is shown in figure 3.3. The workflow requires two Social Media specific agents to detect the RSS feed and to extract UGC out this RSS feed. All other workflow elements are realized using already available agents at the blackboard architecture.

In the following it is assumed that there is a list of weblog URLs that should be crawled. There is no other information than the list itself. In especially, there is no importance information used because all weblogs are a-priori rated equally.

By using a universal target selection agent that simply puts an URL object of data type \texttt{Weblog} to the blackboard, the root page of each weblog will be downloaded using both legal audit agents and the harvesting agent. At this page, the URL for the RSS feed has to be extracted. It is predefined in which way the URL is available: In case of RSS 2.0 (which is the current standard) it is provided as meta link of type \texttt{application/rss+xml} (listing 3.1).

The extraction is done by a weblog specific processing agent that requires

\textsuperscript{17} The process of organizing data to minimize redundancy is called normalization.
\textsuperscript{18} e.g. wordpress, Moveable Type
\textsuperscript{19} e.g. Blogger.com, wordpress.com
\textsuperscript{20} \url{http://www.rssboard.org/rss-specification} – accessed on 2011-03-17
3.3. A Blackboard Crawler Architecture

Figure 3.3: The UML activity diagram presents the initial handling of a new weblog URL that is selected via an universal target selection agent. After applying all legal audit agents and downloading the webpage, the RSS feed URL is extracted and used for further data processing. Only two Social Media specific agents have to be implemented (orange). General agents can be reused flexibly.

```
<head>
  ...
  <link ... type="application/rss+xml" ... href="http://example.org/feed/" />
  ...
</head>
```

Listing 3.1: RSS link in weblogs

a Page object of type Weblog. It extracts the RSS feed URL using simple regular expressions and then marks the page as extracted. The extracted URL is used by a storage agent that saves the RSS feed URL to the corresponding weblog. In further iterations it is therefore not necessary to extract the RSS feed URL anymore. The process can directly load the RSS feed. This improves the Politeness Policy. In addition, the processing agent creates a URL object of type RSS which is stored at the blackboard again. After applying the robots.txt agent, the RSS feed will be downloaded by the harvesting agent and a new Page object of type RSS is placed on the blackboard. Because RSS is well documented, the object can directly be
used by a weblog specific data extraction agent that parses the RSS feed and places each extracted comment as data type Comment on the blackboard if it was not already extracted in previous cycles (using a hashing mechanism). This Comment is used by the general discussion storage engine to store the data into a database.

The re-visit frequency can directly be extracted using the RSS feed that provides time information. It can be used to calculate the re-visit frequency dynamically based on the average time of updates. This is done by an additional agent that calculates the information and stores the re-visit frequency to the corresponding weblog. Weblog specific target selection agents can use this information in following iterations to weight the importance to download the RSS feed for the weblog.

The RSS syntax is well documented and the majority of weblogs uses the format without modifications. This makes it possible to use the crawler architecture not only for one Social Media system but for a complete Social Media type. It has to be noted however that user comments in a weblog are not extracted. Additionally, some weblogs provide only excerpts in an RSS feed which makes the approach inadequate for complete discussion extractions. A more individual approach may be necessary which is discussed in the next section for Internet fora.

Internet Forum Agents

Only a very small number of forum systems provide RSS feeds. These feeds either list the last comments or the last discussions. A common structure among different systems is not available. An even smaller number provides a sitemap protocol. Therefore the individual pages have to be downloaded to extract UGC. There are two important pages for a Social Media crawler: overview pages and discussion pages. Each Social Media system realizes an individual data structure and representation which enforces individual extraction agents for each forum.

In the following it is assumed that there is a list of Internet fora that should
3.3. A Blackboard Crawler Architecture

Figure 3.4: The UML activity diagram presents the workflow to extract UGC from Internet fora. Again, only Social Media specific processing agents are necessary to be implemented (orange). All other workflow elements can be realized using general agents.

be crawled to extract UGC. Similar to the approach for weblogs there is no other data available than the URLs. The implicitly provided workflow can be seen in figure 3.4. For each Social Media system, there are again two specific processing agents necessary to extract the discussion URLs and the user comments. All other workflow elements are realized using already available agents.

Based on a universal target selection agent, URL objects of type Forum are put to the blackboard. After applying both legal audit agents and harvesting the page, a Page object of type Forum is created for further processing.

Depending on the forum system, the root page is differently realized listing links to recent discussions, links to category or tag pages or both. Therefore two different processing agents have to be applied: an overview page URL extractor (not visualized in figure 3.4 to support readability) and a discussion page URL extractor. In both cases the URLs can be extracted using regular expressions as the creation is rule based. The rule depends however on the concrete forum system which makes it necessary to realize individual agents for each Social Media system. The extracted URLs have
to be stored with type \texttt{ForumOverviewPage} or \texttt{ForumDiscussionPage} to the blackboard after checking whether the page was already downloaded and – if this is the case – whether the page should be reloaded to look for new comments. The behavior depends on the page type:

\textbf{Overview Pages} Overview pages list discussions in reverse chronological order. Thus it is possible to limit the re-visit to the first overview page to detect new comments. As most Internet fora structure overview pages in a topic hierarchy, there may be different first overview pages for each hierarchy and only these pages have to be reloaded. An overview URL extractor agent can check whether the overview page is the first overview page using the link structure.

\textbf{Discussion Pages} Analyzing two million discussions from 2007 to 2010 and 20 fora, the average duration of a discussion is 15 days. There are however some discussions with surprisingly long durations which makes a date based approach useless. To limit the download to probably changed discussion pages, it is however possible to use a simple heuristic. Overview pages list discussions in reverse chronological order with additional meta data for each discussion\footnote{Commonly, the last author and the last comment date are published for each discussion.}. This order and meta data can be used to estimate whether there was an update or not. It is thus possible to specify whether a detail page should be analyzed again.

After downloading detail pages, the discussions have to be extracted using special forum specific wrapper algorithms. The proposed crawler architecture uses simple XPath\footnote{XPath (XML Path Language) is a W3C specified language for selecting nodes from an XML document \url{http://www.w3.org/TR/xpath20/}} wrapper algorithms. In the first approach, these wrappers are manually created. Their application is explained using the pseudo code in listing \ref{listing:extract}. It uses two forum specific XPath expressions to extract the discussion title and the discussion topics that represent the
3.3. A Blackboard Crawler Architecture

function extractComments(Page p, Forum f) {
    String title = applyXPath(discussionXPath,f,p);
    Vector<String> topics = applyXPath(topicXPath,f,p);
    Discussion d = new Discussion(title,topics);

    Vector<String> commentsAr = applyXPath(commentsArrayXPath,f,p);
    foreach(commentString in commentsArr) {
        String comment = applyXPath(commentXPath,f,commentString);
        String date = applyXPath(dateXPath,f,commentString);
        String user = applyXPath(userXPath,f,commentString);
        d.addComment(new Comment(comment,date,user));
    }
    return d;
}

Listing 3.2: XPATH Wrapper algorithm to extract user comments in a Social Media specific manner.

forum categories. An additional XPath expression is used to extract all comment strings. These strings are further analyzed using forum specific XPath expressions for the comment itself, the author name and the date.

The wrapper algorithm is not forum specific but can be parameterized for different Internet fora. The extracted comments are finally put to the blackboard and stored by the general discussion storage agent.

Optimized Internet Forum Agents

The presented crawler workflow for Internet fora is able to extract the user comments out of most currently available Internet forum systems. The code is reusable and the blackboard architecture is able to scale each agent according to the needed calculation time. However it needs to download overview pages to detect discussion pages. In case of many Internet fora it is possible to improve the Politeness Policy by predicting discussion URLs. Most currently available OpenSource systems identify a discussion using an ID. These IDs are directly available in the URL in addition to the forum category and the discussion title. Both strings are only provided for Search Engine Optimization (SEO) reasons and are irrelevant to identify
the discussion. Internally, the URLs are rewritten\(^\text{23}\) and the discussion is identified using the unique ID that is continuously increased for each new discussion (cf. figure 3.5). This fact can be used to predict the URL of a discussion by generating URL patterns. Even weblogs could be crawled in this way; the ID is however typically hidden.

\[
\begin{align*}
\text{http://www.motor-talk.de/forum/citroen-ax-motor-ruckelt-t3050617.html} & \rightarrow \text{http://www.motor-talk.de/forum/string-tID.html} \\
\text{http://www.motor-talk.de/forum/sicherung-brennt-ab-und-zu-durch-t2948853.htm} \\
\text{http://www.benzworld.org/forums/u-s-midwest/1565750-get-your-oil-changed.html} & \rightarrow \text{http://www.benzworld.org/forums/string/ID-string.html} \\
\text{http://www.mathias-bank.de/veroeffentlichungen} \\
\text{http://www.mathias-bank.de/?p=21} & \rightarrow \text{http://www.mathias-bank.de/?p=ID}
\end{align*}
\]

Figure 3.5: Static URLs have many benefits especially for Search Engine Optimization. Internally, the URLs are rewritten and the discussion is identified using an unique ID. Only this ID is of importance, the rest is not considered. The URLs can be predicted in most Internet fora.

The improved method only loads data that probably contains UGC. It causes however new problems. Most obvious is the fact that the ID is finite. Increasing the ID and downloading discussion pages will result in error pages – so called 404 pages. In this situation the HTML header should response a 404 error heading which can be used to detect that the generated URL is not valid. Unfortunately not all available systems provide this header which is why the content of the page has to be analyzed for special strings that may be unique for each forum. Thus an additional agent is necessary that checks if a generated URL was valid. If there are many consecutive invalid URLs,\(^\text{23}\) e.g. using the Apache module mod_rewrite.
it can be concluded that the most current discussion was already loaded and that there is no further discussion available. The URL generator agent has to be informed to stop the URL generation and to wait some time until new URLs will be generated. The proposed architecture uses a failure rate of 5 consecutive requests. This rate was empirically estimated crawling 20 fora with more than 13 million user comments. The re-crawling interval is estimated using the last successful crawls.

In addition to the generation of invalid URLs the re-visit policy can only be based on the extracted date information of the discussions. This produces a large list of potential updated discussions. The classical approach can use the URLs in overview pages because all new comments are listed on these pages. This makes the classical approach superior to the approach presented in this discussion at least for current discussions.

Applying a combined approach is the optimal strategy to crawl a web forum: Adding a new forum to the analysis source, the automatic URL generation can be used to get older discussions. This method is continued until the newest discussion was loaded. Overview pages are used in parallel to take account of new provided data. The strategy can be simply realized adding one additional target selection agent that generates URL objects and one processing agent that checks the validity of downloaded pages. All other agents can be reused from the classical workflow.

A big advantage using this approach is that the crawler system does not behave like classical crawler systems. Applying a classical crawler approach to 20 Internet fora, the download was blocked for 20%, especially at large online communities with many thousand discussions (e.g. MotorTalk$^{24}$). The presented multi agent based crawler system has been no block based on the crawler behavior, except for too many requests in too short time. The possibility to not only extend the processing workflow but also to implicitly control the download flow has shown a big advantage compared to currently available crawler systems. Predicting discussion URLs also relaxes the content-seen problem$^{76}$ because the IDs are unique and duplicate con-

[^24]: [http://www.motor-talk.de](http://www.motor-talk.de)
tent will not be downloaded. Much less computation time is needed because it is not necessary to check whether a page was already downloaded.

### 3.4 Semi-supervised Wrapper Generation

In the previous section, a multi-agent based crawler architecture was presented in which different Social Media systems can be flexibly crawled. It is used in the context of this work to extract UGC and to store only the provided comments and their meta data. Navigation sections, advertisements and other design elements are neglected as they may negatively influence further analysis results with automotive related terms which can typically be found in these page areas.

As different Social Media systems provide no common data interface, it is necessary to realize individual wrapper algorithms. The creation of these algorithms is time consuming and not applicable in daily work for the large amount of available Social Media systems. An appropriate approach to support this task is not available as the detection of different items is only analyzed for product pages. Products provide a common data structure and are presented in a very similar way. User comments in the contrary highly differ in content, length, style and containing elements.

In this work, a new approach is presented that is able to detect UGC from detail pages and to segment the data automatically into individual entries. It first detects the main content section in which the user comments can be found by applying a modified version of the SST algorithm presented in related work (cf. section 3.1.2). Then, the main content section is analyzed for repeating structures to segment the content into individual comments. The approach was separately published in [10].

---

25 In the presented approach, at least different XPath expressions have to be provided.
26 A content may contain lists, tables, images, URLs, different number of paragraphs, videos, cites and code elements.
3.4.1 Unsupervised Discussion Detection

A modern web page not only consists of the main content but provides additional areas for navigation, advertisement and other design elements. This also applies to Social Media systems (cf. section 2.6). Their main content is based on UGC. No further listing structure needs to be analyzed to investigate this data (in contrast to product listings, cf. [107]).

There are many approaches available to detect the main content (cf. section 3.1.2). A Social Media crawler can benefit from the fact, that all available systems are template based. That means that each user comment is visualized according to style definitions. Global analysis techniques can extract these style definitions by comparing multiple pages and detecting common elements. In this work, a modified version of the SST algorithm [171] is used to detect the main content area from which user comments have to be extracted. The SST algorithm is applied without modification until importance values are generated for all HTML nodes.

- The header, navigation and advertisement sections exist on every page with similar content. They get a low importance.

- The content section instead changes more frequently – there is no second page with the same content and the number of comments varies for each discussion. This section gets the highest importance.

Instead of creating a new cleaned version of the page using this information (like the original SST algorithm does), the importance information is used as label for an XPath wrapper induction process. Therefore, the complete Style Tree is traversed for every discussion page and the nodes with the highest importance are collected. Each one is used to create an XPath expression which is saved in a list. There are three types of possibly wrong expressions:
Chapter 3. Data Collection

XPath expression for discussions with one comment:
/html/body/div[@id="content"]/div[@id="postlist"]/div[@id="post1285"]

XPath expression for discussions with multiple comments:
/html/body/div[@id="content"]/div[@id="postlist"]

Figure 3.6: By using the SST algorithm an, the most important node differs in cases of discussions with only one comment. The generated XPath expression points to the comment instead of the discussion. This XPath expression has to be generalized to be usable.

1. **Trivial expressions**: If the Social Media system provides a complete different web page\(^\text{27}\), the SST algorithm will assign the highest importance to the body node. It can be assumed however that there are more elements than the main content in Social Media systems. The XPath denoting to the body node can thus be simply disregarded as this is no valid HTML tag for the main content section.

2. **Too specific expressions**: Discussion pages consist of different numbers of comments. If there is just one comment, the SST algorithm assigns the highest importance to the comment node while on pages with multiple discussions, the node containing all discussions will get the highest value. In both cases, the generated XPath expressions are not wrong at all, they are partially too specific. By comparing the generated XPath expressions for multiple discussion pages, it is possible to detect these too specific ones. The more general discussion path is part of the comment specific XPath expression (cf. fig. 3.6). Thus the too specific paths can be simply generalized.

3. **Wrong expressions**: In a few cases, the most important node is not a content node. This happens when there is a section that does not exist on other pages or when the page has no discussion at all.

\(^{27}\) e.g. an error page or a message that the page cannot be displayed due to access limitations
The mentioned problems can be solved by comparing all generated XPath expressions for all pages. The global analysis approach allows it to generalize comment specific paths to discussion specific XPath expressions and at the same time, completely different XPath expressions can be neglected using a simple majority vote. The most extracted XPath expression is supposed to be the valid path which points directly to the discussion section. The modified version of the SST algorithm is thus used to generate Social Media specific discussion extraction algorithms in an unsupervised way.

3.4.2 Unsupervised Comment Segmentation

The discussion section has to be further analyzed to detect the complete posting structure which is based on the user comment and additional meta information. Knowing this structure, further analysis can be focused on relevant user comments. Additional meta data and even off-topic comments can be neglected which may highly influence the analysis quality. The task of comment segmentation has not been considered until now which makes a new approach necessary.

The analysis of more than 3,500 real user comments in 13 different Internet fora based on 8 fora systems lead to following observations, that extend those mentioned by [107]:

1. User comments mainly consist of textual data. On average, a user comment has 320 characters with 4.3 line breaks. 25% of all user comments have only one line. In this case, the average length is 94 characters. The complete statistical distribution can be seen in table 3.1.

2. User comments with multiple lines are presented either by using the block level element for denoting separate paragraphs (p tag) or by using simple line breaks, denoted by br tags.29

28 The fora have not been limited to the automotive domain.
Table 3.1: Statistical distribution of the length of comments, including the average, the 0.25 Quantile ($Q_1$), the Median ($Q_2$) and the 0.75 Quantile ($Q_3$).

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>$Q_1$</th>
<th>$Q_2$</th>
<th>$Q_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line breaks</td>
<td>4.3</td>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Comment length (multiple lines)</td>
<td>320</td>
<td>80.3</td>
<td>187</td>
<td>387</td>
</tr>
<tr>
<td>Comment length (single line)</td>
<td>94</td>
<td>17</td>
<td>51</td>
<td>104</td>
</tr>
</tbody>
</table>

3. The complete user comment can be found in one parent block level element (cf. section 2.6) which is possibly split by further elements: a, img, object, and sometimes ul or ol nodes. Within the user comment, quotes and code sections are commonly realized using div or table substructures. They can be nested multiple times. The semantically more correct HTML tags quote, blockquote and code are only used rarely.

4. Social Media systems are visualized using templates. Comparing user comments including additional meta data across several pages, it can be seen that the posting structure is identical for all comments. In rare cases, there are additional attributes (typically style definitions) to mark special users (e.g. administrators, moderators).

5. The complete posting structure is realized with block level elements, namely div, table, ul or ol nodes (including corresponding tags like td and li. As these tags are also used within one user comment, their semantically meaning is ambiguous.

6. Some community systems use nested structures to represent answer relations (e.g. drupal). A comment separation algorithm thus has to be able to detect nested structures.

The algorithms found in literature do not consider these observations. They are incapable to detect postings because the algorithms cannot deal with different substructures in depth and length. The presented comment segmentation algorithm uses heuristics based on the observation described above.
Figure 3.7: Post candidate search: First, the algorithm detects possible textual user comments with the help of different heuristics (left). These heuristics are based on observations described in section 3.4.2. By using the DOM structure, it is possible to generalize these candidates to detect the complete user postings including additional meta data (right). These are used in the following to generate wrapper algorithms automatically.

It is a mixture of local and global detection techniques and uses the base idea of [109]: traversing the DOM tree of the content section in a bottom up way to detect nested structures.

**Post Candidate Search** The concept of the presented approach is to first identify possible textual comment candidates by using local analysis techniques. The applied heuristics are based on the observations presented in section 3.4.2. The candidates are then generalized to complete user postings including additional meta data using the DOM structure (cf. fig 3.7). The complete algorithm is separated into four individual steps:

1. **Merge inline elements** User comments are separated by several
tags. Most of them are inline elements\textsuperscript{30} – so-called text level elements. These elements are extended with block elements that are typically used in user comments but not for posting structures: \texttt{pre}, \texttt{blockquote} and \texttt{p}. The algorithm merges their content to the corresponding parents and removes the nodes themselves. The result is a modified DOM tree with block elements only. The textual content of one user posting is not separated by additional text-level elements any more.

2. **Find text nodes** In the modified DOM tree, the algorithm looks for possible textual comment candidates. It takes advantage of the observations made and uses the following selection criteria:

- The text node consists of at least 50 letters and new line characters (\texttt{	extbackslash r,\textbackslash n}). This rule includes 75\% of all user comments (cf. table 3.1)

- If there is no line break, the text node consists of at least 150 characters. The rule ensures to include long comments while at the same time, structural text nodes (e.g. the user nickname, the date, etc.) are neglected. According to the comment length distribution, more than 75\% of all user comments without line breaks are however not detected by this rule.

The parent nodes $c_i$ of these comment candidates are collected in a list $L$. The criteria ensure that no posting structure elements and meta information are selected as candidates. The list $L$ is however not necessarily complete because the algorithm may miss short entries. In the following, this problem is called *problem A*.

Many Social Media systems realize substructures with the same HTML tags that are already used to realize posting structures (cf. section 3.4.2). The selection criteria described above do not disambiguate

\textsuperscript{30} The W3C definition of inline elements is used which includes \texttt{span}, \texttt{img}, \texttt{b}, \texttt{i}, \texttt{strong}, \texttt{em}, \texttt{a}, \texttt{acronym}, \texttt{abbr}, \texttt{code} and \texttt{br} nodes (http://www.w3.org/TR/REC-html40/struct/global.html#h-7.5.3)
Figure 3.8: The list of candidates may contain too specific comment candidates if substructures like code and quote sections are realized with the same HTML tags that are already used for posting structures. In the presented screenshot, the selection criteria will falsely select two candidates as the comment itself and the substructure within the comment are longer than 150 characters.

whether the textual data is found in substructures. Thus, the list of candidates can also contain too specific text nodes like code or quote sections (cf. fig. [3.8]). The list has to be cleaned up in the following to fix two possible errors:

- If the required textual length is achieved in the comment and in the substructure at the same time, there will be two candidates for one comment. This error is called problem B₁.
- If the comment is too short to be selected, L may just contain the substructure element and thus a too specific node. This error is called problem B₂.

3. **Clean text nodes** In addition to the length distribution, the observations state that the complete user comment can be found in one parent node. This property is used to remove duplicate candidates \( c_i \in L \). For each candidate \( c_i \), the approach recursively selects all parents \( p_1, \ldots, p_n \). If there is a parent \( p_i \) already in the candidate list \( L \), the candidate \( c_i \) can be removed because the list contains a more general posting node \( c_j \) already. Problem B₁ is solved. This step ensures that there is only one candidate for each posting (or none due to problem A).
Chapter 3. Data Collection

Input: Node list candidates

\[ \text{listChanged} = \text{true}; \]
\[ \text{while } (\text{listChanged}) : \]
\[ \text{listChanged} = \text{false}; \]
\[ \text{foreach } (c_i \text{ in candidates}) : \]
\[ p = c_i.\text{getParent}(); \]
\[ \text{candidateChildren} = p.\text{getChildrenWithinList}(\text{candidates}); \]
\[ \text{if } (\text{candidateChildren}.\text{size()}==1): \]
\[ \text{candidates}.\text{replace}(c_i, p); \]
\[ \text{listChanged} = \text{true}; \]

Listing 3.3: Each comment candidate \( c_i \) can be generalized traversing the DOM tree to the root node until some stopping criteria are true.

4. **Generalize Nodes** The parent nodes \( c_i \) have been stored in \( L \) for all comment candidates. They do not represent complete posting structures yet because additional meta information like the publication time and the author name is missing. The algorithm has to generalize each candidate to find the complete posting structure. It takes advantage of the observation that each comment can be found in its own subtree. The DOM tree has to be traversed by replacing each candidate \( c_i \) with its parent node \( p_i \), until one of the following stopping criteria is true:

a) \( p_i \) does not exist

b) \( p_i \) is also parent of another candidate \( c_j \in L, j \neq i \)

These stopping criteria ensure to get no overlaps while generalizing candidates. The problem of too specific text node candidates due to quote or code sections (problem \( B_2 \)) is solved. The algorithm is presented in listing 3.3.

The proposed algorithm generates a list of posting candidates for each page. The applied heuristics are based on observations presented in section 3.4.2. It can thus be assumed that these candidates represent user postings. By first detecting user comments via the text length and then generalizing
these candidate nodes, it is ensured that nested structures are considered. Whether a nested structure is used instead of a list structure can be detected by comparing the parent nodes. If there is more than one parent for all extracted posting structure candidates \( c_i \in L \) the comments are nested. Otherwise, the user comments or represented in a list form.

The list of potential user postings is however still incomplete. The selection heuristics have missed short entries and this problem was not solved until now (problem A). It is handled by applying a global comparison in the next section: the wrapper induction.

**Fault-tolerant Wrapper Induction** Local and global analysis techniques are time consuming, making it difficult to apply the previous algorithm steps for every page in large scale applications. In case of template based Web pages, the application of wrapper algorithms is preferable. The *post candidate search* provides a list of automatically labeled data that can now be used in an *wrapper induction* process to speed up further content extraction. In contrast to algorithms presented in related work (cf. section 3.1.2), this induction process has to tolerate potentially incomplete and wrong labeled data. Therefore, a new robust approach is presented that takes advantage of a global analysis technique: As the web pages are created on top of the same template, more than one discussion page can be analyzed to generate reliable XPath expressions.

Currently available DOM libraries provide the possibility to generate an XPath expression for each node in the DOM tree. This generation is however limited to two levels of detail:

1. General XPath expressions: These expressions just use the DOM element information without node attributes and any position information.

   Example: `/html/body/div/div/table/`

2. Specific XPath expressions: These expressions use the position information and are therefore very specific.
Both detail levels are not appropriate. The first one is too general. The usage of those XPath expressions may cause many wrong nodes. The second level however is too detailed because it points to exactly one DOM node. Using this level of detail would make it necessary to use one XPath expression for each posting position. It would be not possible to catch missing postings as the XPath expressions do not generalize the posting structure. A more generalizable XPath expression is needed which is at the same time as specific as possible. Therefore, a new XPath generation algorithm is presented here that uses the fact that in case of (X)HTML documents some node attributes are typically used for visual definitions. Most common is the class, the id, the style and in case of table structures the align attribute. The algorithm uses the availability or the absence of these attributes:

Example: /html/body/div[@id="content"]/div[@style="color:black"]/n

The id attribute has another important role. It can be used as anchor tag in all currently available web browsers. Adding the id to an URL\textsuperscript{31}, the browser will position the view port to the corresponding DOM node immediately. This possibility is used in many forum systems which provide direct posting links. Each HTML id attribute is therefore extended with the unique database ID of the posting (e.g. post12533). The XPath generation algorithm has to check for these numerical values within an attribute. It uses the special XPath function \texttt{contains} to generate generalizable expressions:

Example: /html/body/div[@id="content"]/ul/li[contains(@id,"post")]/n

With this XPath generation algorithm, it is possible to generate a generalizable XPath expression for every posting candidate $c_i \in L$. According to the observations presented in section \textsuperscript{3.4.2} these posting candidates are realized identically with small differences in additional (visual) attributes.

\textsuperscript{31} e.g. \texttt{http://www.example.com/page.html#ID}
3.4. Semi-supervised Wrapper Generation

By comparing all generated XPath expressions on multiple pages, it is thus possible to detect the common posting structure. Therefore, each XPath expression is divided into sub-patterns by splitting the XPath expression at slash characters. If all expressions have the same node type at position $i$, the sub-pattern is added to the resulting XPath expression including common node attributes (cf. fig 3.9). The XPath creation is stopped if one node type is different or if all sub-patterns have been checked.

In nested structures, the global analysis technique has to be altered by inverting the sub-pattern list first to check nodes in reverse order. Identical nodes are added at the front of the generated XPath expression and the algorithm stops by adding an additional slash character at the front to make different “root” nodes possible.

Example: 

```xml
//div[contains(@id,"post")]/
```

By comparing all XPath expressions on multiple pages, the wrapper induction algorithm provides one XPath expression for all user postings. If the number of investigated pages is large enough, it can be assumed that every position in a discussion was considered. The induced XPath expression thus also returns user comments that have been missed in previous steps. Problem A is solved by applying this global analysis technique.

### 3.4.3 Evaluation

The unsupervised wrapper induction algorithm creates two different wrapper algorithms in form of XPath expressions to extract UGC from Social Media systems. The first expression extracts the discussion structure; the second one separates the discussion section to individual user postings.
Both aspects are evaluated in the following.

The algorithm was tested on 51 real Internet fora, based on 14 different fora systems. Each forum uses a different template and was only used for evaluation. Therefore, the underlying assumptions can also be verified. The evaluation was performed by loading 20 discussion pages per forum and applying the proposed algorithm. The resulting XPath expressions were manually checked for suitability (table 3.2).

The modified SST algorithm correctly detects the main content section in all tested Internet fora. Only in a small number of Social Media systems, there are additional elements within the result (e.g. links to other discussions). These elements are not removable by the presented approach as they are in the same HTML subtree as the discussion. The XPath wrapper approach is not able to handle this situation. It can be partially tolerated by the posting segmentation process as these additional elements are commonly realized with a complete different HTML structure than the user postings. Thus, the posting segmentation returns perfect results for 44% of the non-perfect detected discussion sections.

The posting segmentation results are more problematical: In rare cases, the generated XPath expression is not correct at all. One problem is that the first comment may be realized with a complete different HTML structure. Thus the base assumption of equal posting templates was not valid (e.g. http://stackoverflow.com). The first entry has to be analyzed separately. In other cases, the discussion is split into different parent nodes. This was detected by the discussion detection algorithm, but the gener-

<table>
<thead>
<tr>
<th></th>
<th>discussion detection</th>
<th>Posting Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>perfect</td>
<td>82.4%</td>
<td>64.0%</td>
</tr>
<tr>
<td>correct</td>
<td>100%</td>
<td>90.0%</td>
</tr>
</tbody>
</table>

Table 3.2: Wrapper Quality for Discussion and Posting Detection: An XPath expression for the discussion and posting detection is perfect, if it points directly to the discussion or posting. It is correct, if additional elements are within the result (e.g. links to other discussions).
3.4. Semi-supervised Wrapper Generation

<table>
<thead>
<tr>
<th>system</th>
<th>quantity</th>
<th>correct</th>
<th>perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burning Board</td>
<td>3</td>
<td>100%</td>
<td>66.7%</td>
</tr>
<tr>
<td>drupal</td>
<td>6</td>
<td>83.3%</td>
<td>0%</td>
</tr>
<tr>
<td>IPB</td>
<td>5</td>
<td>75%</td>
<td>50%</td>
</tr>
<tr>
<td>myBB</td>
<td>4</td>
<td>66.7%</td>
<td>66.7%</td>
</tr>
<tr>
<td>phpBB</td>
<td>7</td>
<td>100%</td>
<td>57.1%</td>
</tr>
<tr>
<td>SMF</td>
<td>3</td>
<td>100%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Unclassified NewsBoard</td>
<td>2</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Vanilla</td>
<td>3</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>vBulletin</td>
<td>13</td>
<td>92.3%</td>
<td>84.6%</td>
</tr>
<tr>
<td>miscellaneous</td>
<td>5</td>
<td>60%</td>
<td>60%</td>
</tr>
</tbody>
</table>

Table 3.3: By comparing the segmentation quality per forum engine it can be seen that the approach benefits of better semantical posting realizations as done in the Vanilla engine or modern vBulletin templates.

alization step in the posting segmentation algorithm fails. The XPath expression would have to add an additional slash to the XPath expression (e.g. //td) to solve this problem. This would cause worse results in other cases however. The non-perfect results are mainly based on limitations of the XPath approach, as the HTML structure does not provide usable HTML attributes. For drupal-based Social Media systems for example (e.g. http://drupal.org/forum), the algorithm returns the valid XPath //div/div as the corresponding HTML tags provide no usable style information. It would be however necessary to include sibling information to be more precise.

By comparing the posting segmentation results depending on used forum engines (Table 3.3) it can be seen that modern Social Media systems like Vanilla\(^{32}\) or the Unclassified NewsBoard\(^{33}\) provide very good results. This is due to a proper semantical representation of discussions and substructure elements. In older systems (e.g. phpBB\(^{34}\)), many templates are realized using nested tables and it is quite difficult to disambiguate the meaning of each table. As this problem is also available in major search engines, a

\(^{32}\) http://vanillaforums.org/

\(^{33}\) http://newsboard.unclassified.de/

\(^{34}\) http://www.phpbb.com/
proper semantical HTML realization becomes more and more important in terms of search engine optimization. Modern templates in current versions of these forum engines (e.g. *phpBB 3.0* or *vBulletin 4.0*\(^{35}\)) are a big improvement for a clean content detection. It can thus be assumed that the overall quality of the presented approach increases with improved templates.

### 3.4.4 Semi-Supervised Processing Agent Generation

As the evaluation has shown, the proposed algorithm is not able to detect all user postings in every Social Media system. To ensure high quality extraction results, it is thus not recommendable to apply the algorithm in an unsupervised manner. Instead the analysis sources should be selected and added manually. The proposed algorithm can support this by suggesting XPath expressions to generate new processing agents and therefore to handle the large amount of possibly relevant Social Media systems. In this application even non-perfect XPath expressions are helpful because it is much easier to improve a too general XPath expression than to generate an XPath expression from scratch. The unsupervised discussion detection wrapper induction algorithm can thus be applied in a semi-supervised approach in the complete crawler architecture.

### 3.5 Structure Modification Detection

Social Media systems are continuously changing. This is not only caused by users that create UGC but also by the system owners themselves who apply some changes in layout and functionality. Using a wrapper approach makes it necessary to analyze the system continuously for possible changes to ensure that the extracted UGC is still valid. As mentioned in related work (section 3.1.4), a common method for wrapper verification is the application of regression tests. In these tests already downloaded pages are reloaded and the stored data is compared to the data that is newly extracted by using the wrapper algorithm. In contrast to related work, it is however not

\(^{35}\) [https://www.vbulletin.com](https://www.vbulletin.com)
3.6. Conclusion

necessary to tolerate changes within the comments because old entries are rarely changed. In many Social Media systems, this is even not possible at all for end users. More common are changes because a user wanted to be deleted or due to legal restrictions.

An appropriate Social Media wrapper verification algorithm has to compare the wrapper results on posting level. Thus, new user postings and already deleted user postings can be simply tolerated. By verifying the wrappers in a global analysis approach again, the probability of coincidentally available changes can be minimized. If a regression test fails, the wrapper is invalid and the Social Media system should not be crawled until the system administrator has fixed the wrapper which can be done in an interactive way again.

Using the blackboard architecture it is quite easy to implement this verification approach. Two different agents are necessary. The first agent generates an URL object that will be downloaded and processed, the second agent has to compare the processed data to stored data. It has to mark a wrapper algorithm as invalid, if the comparison fails.

3.6 Conclusion

This chapter proposes a new crawler architecture that is capable to crawl UGC from different Social Media sources within one common system. In the multi-agent based crawler, each individual task is handled by a highly specialized agent that can be reused for different data sources. Hereby, reimplementation affords can be reduced while code quality increases. The implicitly provided workflow supports the realization of individual selection and re-visit policies. Thus, the overall resource handling, especially with regard to bandwidth limitations, can be optimized. In addition, the host penetrations are minimized by only investigating web pages that probably contain UGC. This was shown for Weblogs and Internet fora.

A Social Media specific crawler system has not only to provide a flexible architecture but also to extract the data of interest. Navigation sections,
advertisements and other design elements have to be neglected to reduce noisy information. By limiting the data extraction to the content section, the analyses can be focused on relevant information. Furthermore, each user comment has to be extracted separately. This is the only way to properly detect references like quotes while irrelevant off-topic comments within a discussion can be neglected. Therefore, the architecture supports Social Media specific processing agents. They apply specific extraction algorithms to store the UGC instead of complete web pages. Two general processing agents have been presented in the context of weblogs and Internet fora. For weblogs, the extraction is based on a simple RSS parser. Agents for Internet fora use manually defined XPath expressions which first denote the discussion section and within this discussion section, the individual user postings.

The manual definition of XPath expressions is very time consuming. Until now, there was however no focus on the automatic extraction of UGC from Social Media systems. Therefore, a new algorithm was developed. It first detects the discussion section by comparing multiple pages. Using a modified version of the already known Site Style Tree (SST) algorithm, it labels subsequently important HTML nodes. These labels are used to induce a discussion specific XPath expression. The method was shown to be able to detect all discussions in all tested Internet fora. The extraction of user postings within the discussion section is realized similarly. Due to the high variation in the text length and structure, global analysis techniques are however not applicable. Instead, each discussion has to be analyzed separately applying different heuristics. These heuristics are based on observations which have been gained by investigating 3,500 real user comments (section 3.4.2). They first detect comment candidates which are generalized to posting candidates. They are used as unsupervised labeled data in a wrapper induction algorithm. This algorithm has to be fault-tolerant because of possibly missed short comments. It uses a special method to generate general and at the same time specific XPath expressions. The induced Social Media specific XPath expressions have been shown to be applicable
to 90% of all tested Internet fora. They provide the same structure than
the manually defined XPath expressions. Hereby, the generated wrappers
are directly applicable in the multi-agent based crawler architecture.

To ensure a valid content extraction, the available wrapper algorithms are
continuously monitored. Therefore, simple regression tests are applied on
posting level. If the newly extracted data is identical to already stored data,
the wrapper is still valid.
There is an immense number of textual data extracted from Social Media systems. Thus, different analysis algorithms are necessary to investigate the content. One important task is to extract the discussed topics. However, it is very difficult to define what a topic may look like. Pederson distinguishes headed and unheaded contexts. In headed contexts, the topic is explicitly mentioned in the data, while in unheaded data it has to be extracted using the complete textual environment. Topics may be further described on different abstraction levels. So are comments mentioning the navigation map for example also discussing the navigation system. The different abstraction levels describe the same headed or unheaded topic simultaneously. The deduction of the hierarchy is context dependent. In case of an integrated system, the navigation system, for instance, also describes the radio, the entertainment system and finally the interior equipment. Moreover, comments concerning the entertainment system are not only part of the interior equipment, but also of the electronic equipment. The map discussions also describe service components that needs regular
Chapter 4. Hierarchical Fuzzy Clustering

All premium manufacturers are promoting less fuel consumption with modern cars. With my new BMW, I’m absolutely satisfied. But I’m quite happy that I have no start-stop functionality as a friend of mine has told me that in his smart, no air conditioning is working then.

Figure 4.1: In UGC, there may be multiple topics discussed simultaneously.

*software updates,* etc.. The topic structure is defined in consequence by a directed, acyclic graph structure, which may be realized as an ontology.

In case of quality analyses, topics within headed and unheaded contexts have to be considered. They can deal with a specific product, a feature, a product behavior, a customer behavior, a symptom or simply an expectation. Within one user comment, there may be multiple topics discussed simultaneously. For example, a discussion describing problems concerning a special product feature for a specific product deals with three different topics: the problem itself, the product feature and the specific product. Comments can also include comparisons in which a product or a feature is compared to other products and features, even other manufacturers (cf. fig 4.1). In consequence, the discussed topic is the combination of such individual topics which have to be identified by a topic analysis algorithm in potentially overlapping textual regions.

An appropriate topic analysis system has to provide the possibility to analyze these multiple topics, regardless whether the topics are explicitly mentioned or have to be concluded using the textual context. The different abstraction levels provide analysis protagonists the opportunity to apply analyses according to their needs. Furthermore, it makes the huge amount of data manageable. Abnormalities can be detected on abstract levels in which a large number of documents is summarized. Detailed information can be gathered on deeper levels. The topic hierarchy thus supports deep-dive analyses. In addition to this hierarchy, following requirements have to
be fulfilled:

- The analysis approach has to be aware of already known topics within a-priori defined topic hierarchies. At the same time, new topics have to be extracted to detect possible blind spots.

- The topic detection has to be deterministic and thus reproducible. Each time, the analysis is applied on the same data set, the analysis results have to be identical. Otherwise the analysis results are less reliable and it is difficult to compare the extracted information over time.

- As UGC is a very fast growing data source, the data analysis has to be stable towards data changes. New data should not change to overall analysis results.

All these requirements cannot be handled within one single algorithm. A-priori known topics have to be detected according to the provided hierarchical structure while at the same time a-priori unknown classes can provide completely new topic hierarchies. Both hierarchies have not necessarily any topic in common. Thus, different algorithms have to be applied in a Social Media analysis system.

Different approaches have already been published to handle multiple a-priori known topics for headed and unheaded contexts (cf. [159]) which can be hierarchically structured (e.g. [52, 22, 134, 149]). These stable and deterministic techniques are based on classification approaches which use a set of features to assign the corresponding topic. Two examples are discussed in detail in the next chapter (section 5.4).

A-priori unknown topics are detected by investigating the data itself to create data groups with common properties. This is typically done using clustering algorithms (cf. [156, sec. 8.1.3]). The large number of use cases causes many different clustering approaches. The analysis of related work
(section 4.1) does not identify an algorithm that fulfills all previously described requirements. Either the approaches are indeterministic, the number of clusters is a-priori limited or the number of multiple assignments is restricted. Therefore, a completely new hierarchical clustering algorithm is developed in section 4.2. It extends well-known agglomerative hierarchical clustering algorithms to fuzzy logic. The deterministic approach can flexibly handle multiple topics on different abstraction levels. However, the clustering technique generates binary clusters including cluster hierarchies not necessarily representing a topic. Thus, a new topic generation algorithm is presented in section 4.3. Section 4.4 shows that the hierarchical data representation benefits of this new clustering algorithm. Hereby, a new quality based evaluation technique is introduced. Moreover, it is shown that the deterministic approach generates stable clusters. The overall approach has been presented in [13].

4.1 Related Work

In Information Retrieval, it is typically assumed that documents relevant to a given request can be separated from irrelevant documents. In consequence, a set of different requests should be separable, too. This assumption known as cluster hypothesis [164, chap. 3] allows to combine similar elements to homogeneous groups, so-called clusters. Whether two elements are similar highly depends on the analysis scenario. To apply any kind of clustering, two decisions have to be made:

1. Feature Definition: First of all, it has to be defined which features are used to calculate the similarity between two elements, an element and a cluster and between two clusters. It has to be specified, how the features are obtained and in which way the similarity is calculated.

2. Algorithm Definition: Different problem situations need different clustering concepts. To select an appropriate clustering method, the anal-
ysis requirements have to be carefully specified. On top of these re-
quirements, the clustering algorithm has to be selected.

This section provides a short overview of related clustering concepts and
theories. Section 4.1.1 starts with a basic introduction to a classical way
of feature generation using the Vector Space Model (VSM). It is a very
common mathematical representation form for textual data that makes it
very easy to process textual information. It is the base technique for many
different clustering algorithms, including the proposed approach. Section
4.1.2 presents common clustering algorithms that use the VSM to generate
different types of clusters. Of special interest are hierarchical clustering
algorithms with the possibility of multiple data assignments.

4.1.1 Feature Generation

Document clustering aims to group similar documents so that documents
in the same cluster are more similar than documents in different clusters.
Calculating the similarity between two documents requires a mathematical
representation of each document. Most common is the Vector Space Model
(VSM) which was introduced in 1975 by Salton et al. [141]. They proposed
a vector representation for each document for which distances to other
documents can be easily calculated using classical metrics.

In the following, a text collection (corpus) \( X \) is given by \( N \) documents \( x_i \):

\[
X = \{x_1, x_2, \ldots, x_N\}; \quad N \in \mathbb{N}
\] (4.1)

The VSM is based on a lexicon \( L \) that consists of all used terms (types) \( t_i \)
in the text collection (corpus) \( X \):

\[
L = \{t_1, t_2, \ldots, t_n\}
\] (4.2)

\(^1\) If not specified otherwise, this section is based on [113] and [77].
Each document $x_i$ is represented by a multidimensional vector $\vec{x}_i$. Each dimension denotes a weight $w_i$ for the corresponding type in $L$:

$$\vec{x}_i = \{w_1t_1, w_2t_2, \ldots, w_nt_n\}$$

(4.3)

The VSM assumes each type to be independent from all others and therefore documents are data points in a multidimensional Cartesian coordinate system. The concept is also known as bag-of-words approach, because the word ordering gets lost. The generation of each weight $w_i$ depends on the concrete realization. It is typically based on three components:

**Local component** The local component $w_{local}(\vec{x}_i, t)$ specifies the importance of type $t$ in document $D$.

**Global component** The global component $w_{global}(t)$ takes the global usage of type $t$ in the complete corpus into account. Especially often used terms can be re-weighted by this component.

**Normalization component** The normalization component $w_{norm}(\vec{x}_i)$ is used to neglect the influence of unwanted document properties, e.g. the document lengths that may have direct influence to $w_{local}(\vec{x}_i, t)$.

Table 4.1 presents some classical variants to calculate the three components that are combined in one term weight:

$$w(\vec{x}_i, t) = \frac{w_{local}(\vec{x}_i, t) \cdot w_{global}(t)}{w_{norm}(\vec{x}_i)}$$

(4.4)

Using the VSM, the proximity of two documents can be calculated using a metric which is defined by four properties (cf. [21]):

1. $\forall \mu \in F, d(\mu, \mu) = 0$ Reflexivity
2. $\forall \mu, \nu \in F^2, d(\mu, \nu) = 0 \Rightarrow \mu = \nu$ Unambiguity
3. $\forall \mu, \nu \in F^2, d(\mu, \nu) = d(\nu, \mu)$ Symmetry
4. $\forall \mu, \nu, \xi \in F^3, d(\mu, \nu) \leq d(\mu, \xi) + d(\xi, \nu)$ Triangle inequality
4.1. Related Work

**Local variants for type $t$ in document $x_i$:** $w_{\text{local}}(x_i, t)$ (for $tf_{x_i}(t) > 0$)
- Binary weighting for the availability of $t$
- Term frequency ($tf$)
  
**Global variants for $t$:** $w_{\text{global}}(t)$
- No global weighting
- Inverse document frequency ($idf$)
  
**Normalization variants for $x_i$:** $w_{\text{norm}}(x_i)$
- No normalization
- Cosine normalization

$$
\sqrt{\sum_{i=1}^{m} (w_{(x_i, t)} \cdot w(i)_{\text{global}})^2}
$$

Table 4.1: Overview for the most common term weighting components

The lower $d$ is, the more similar two documents $\mu$ and $\nu$ are. This metric definition is also known as distance function. The best known distance function is the Euclidean distance:

$$
d(\mu, \nu) = \sqrt{\sum_{i=1}^{n} (\mu_i - \nu_i)^2}
$$

(4.6)

It is also possible to define an inverse function, using following properties (cf. [21]):

1. $\forall \mu \in \mathbb{R}, s(\mu, \mu) = 1$ \hspace{1cm} Reflexivity
2. $\forall \mu, \nu \in \mathbb{R}, s(\mu, \nu) = s(\nu, \mu)$ \hspace{1cm} Symmetry
3. $\forall \mu, \nu, \xi \in \mathbb{R}, t[s(\mu, \xi), s(\xi, \nu)] \leq s(\mu, \nu)$ \hspace{1cm} T-Transitivity using $t$ as $t$ norm

(4.7)

This definition is known as similarity measure. It can be derived from the distance function using the equation $s = \frac{1}{d+1}$ (cf. [77], chap. 2.5). The most common similarity measures are listed in table 4.2.

The VSM proposed by Salton et al. [141] is very common and there are many applications using this approach. Nevertheless, there have been many
Chapter 4. Hierarchical Fuzzy Clustering

<table>
<thead>
<tr>
<th>similarity function $s(x, y)$</th>
<th>binary calculation</th>
<th>weighted calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching coefficient</td>
<td>$</td>
<td>x \cap y</td>
</tr>
<tr>
<td>Dice coefficient</td>
<td>$\frac{2</td>
<td>x \cap y</td>
</tr>
<tr>
<td>Jaccard coefficient</td>
<td>$\frac{</td>
<td>x \cap y</td>
</tr>
<tr>
<td>Cosine similarity</td>
<td>$\frac{</td>
<td>x \cap y</td>
</tr>
</tbody>
</table>

Table 4.2: Similarity functions for binary and weighted vector calculation.

Researchers proposing additional variants and improvements. Next to the loss of word ordering, the VSM model lacks the usage of synonyms and other inter-word relations which makes the dimensions linearly dependent. Pedersen [130] provides a sophisticated overview about different possibilities to improve the feature generation for similarity calculation. A very simple method is to enrich the original document with known related concepts based on thesauri or ontologies. More mathematically motivated concepts analyze a larger set of documents to learn related concepts typically used in the same topic. Well known is the LSI algorithm [47], the LDA approach [20] and the ESA algorithm [60, 59]. Stein and Anderka [155] provide a very well structured unifying view to these different similarity calculation approaches.

Although the variants are very promising, their application is out of focus for this work. The advantages of the new clustering approach will be shown by comparing results with results obtained by classical clustering approaches using the same data set and same data representation. More sophisticated data representations (e.g. by using the LDA approach) would improve the new proposed and the classical clustering algorithms in the same way. Thus, Salton’s classical $tf \cdot idf$ technique (cf. [142]) is used:
4.1. Related Work

\[ w(\vec{x}_i, t) = \frac{tf_{x_i}(\vec{x}_i, t) \cdot \log \frac{N}{df(t)}}{\sqrt{\sum_{i=1}^{m} (tf_{x_i}(\vec{x}_i, i) \cdot \log \frac{N}{df(i)})^2}} \] (4.8)

The local variant counts the importance of a given term \( t \) in document \( D \) while the global inverse document frequency takes the relative importance of \( t \) into account. As normalization method the cosine method is used. As each document vector \( x_i \) has a length of 1 due to the normalization method, the cosine similarity can easily be calculated using the dot product:

\[ s(\vec{x}_i, \vec{x}_j) = \sum_{a=1}^{t} x_{ia} \cdot x_{ja} \] (4.9)

The representations ensures that different document lengths are unimportant to calculate the feature weight. This is especially important in analyzing UGC with varying content lengths.

4.1.2 Clustering Algorithms

“Clustering aims to find useful groups, where usefulness is defined by the goals of the data analysis.” [156, sec. 8.1.3]. The idea is to use “only […] information found in the data that describes the objects and their relationships” [156, sec. 8.1.1]. As the analysis requirements are varying, there is an impressive amount of different clustering approaches available. The algorithms can be categorized according to individual algorithm strategies. The most important categorizations will be discussed in the following.

Hierarchical vs. Partitional Clustering The most discussed categorization is surely based on the created cluster type, in particular “whether the set of clusters are nested or unnested, or in more traditional terminology, hierarchical or partitional” [156, sec. 8.1.2]:

1. **Partitioning clustering algorithms** separate a data space \( X \) into \( k \) clusters \( C_i \). The created partition \( \mathcal{P} \) satisfies two properties:
Figure 4.2: Hierarchical clusters group data elements into nested clusters. The figure shows these groups in a so-called dendrogram. The hierarchical cluster can be created in a bottom-up way, starting with individual data elements, or in a top-down approach by recursively splitting each data group.

2. Hierarchical clustering algorithms on the other side create a set of nested clusters (cf. fig. 4.2). In case of single document assignments, this set represents a tree. A graph is created in case of multiple assignments. The available algorithms can be separated into two subgroups:

Agglomerative algorithms create a cluster hierarchy in a bottom up way. Typically, all data elements are regarded as one-element clusters – so called singletons – which are recursively merged until all elements are in one root cluster.

Divisive clustering approaches use a top down mechanism. Starting with a root cluster containing all data elements, the cluster is recursively split using some heuristics until only singleton clusters are available.

Exclusive vs. Overlapping Clustering Another common distinction is the analysis whether the clustering algorithm creates exclusive or over-
4.1. Related Work

lapping clusters:

1. **Exclusive:** The data element $x_i$ belongs exclusively to $C_j$. The clusters are also known as hard or crisp clusters.

2. **Overlapping:** The data element $x_i$ can belong to several clusters. In practice this is a common situation, as it is not always possible to exclusively assign an element to one group (e.g. tagging systems). The clusters are also known as soft clusters.

A generalization of the overlapping method is a fuzzy assignment, in which every data element $C_i$ “belongs to every cluster $[C_j]$ with a membership weight $[\mu_{ij}]$ between 0 (absolutely doesn’t belong) and 1 (absolutely belongs)” [156, p. 492]. The paradigm follows the concepts of fuzzy logic (cf. [172]). It extends classical boolean logic, in which a statement is not only true or false but can be partially true [140, chap. 11]. Instead of using hard thresholds an object has to reach, a membership function for different object properties is used (cf. [75]). This so-called fuzzy membership has to be distinguished from probability statements. A cite, based on Prof. Sol Golomb, makes the difference more clear:\footnote{The cite was mentioned in an email from Elwyn Berlekamp to Lotfi A. Zadeh in August, 2000. It can be found at \url{http://www.dbai.tuwien.ac.at/marchives/fuzzy-mail2000/0644.html} (accessed on 06/17/2012).}

Question: If there a salami sandwich in the refrigerator?
Answer: 0.5
If “probability”, then there either is or isn’t, with probability one half.
If “measure”, then there is half a salami sandwich there.
If “fuzzy”, then there is something there, but it isn’t really a salami sandwich. Perhaps it is some other kind of sandwich, or salami without the bread...

In clustering algorithms, it is typically assumed that all membership weights of one data elements are in sum 1. This defines each cluster as a fuzzy set,
Chapter 4. Hierarchical Fuzzy Clustering

specified by all data elements and the corresponding fuzzy memberships:

\[ \sum_{i=1,...,N}^{j=1,...,k} \mu_{ij} = 1 \]  \hspace{1cm} (4.10)

**Similarity Distinction** The created cluster types can be further distinguished into the way, the inter-cluster similarity is calculated.

- Prototype-based approaches generate clusters by specifying a data prototype. Related data elements have to be closer to this prototype than to any prototype of other clusters.

- Graph-based algorithms represent the data as a graph with objects as nodes and connections between the objects as links. The clustering approaches group “the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters” [144]. The technique is of advantage in irregularly structured or intertwined data (cf. [156, chap. 8]).

- Clustering algorithms can however also take the data density into account. These density-based methods are able to create non-convex clusters by taking the inter-cluster similarity into account.

**Clustering Process Distinction** Finally, the clustering process itself can be used for categorization: Is it a *iterative* process or a *single pass* algorithm? This directly effects the complexity of the algorithm which is a very important decision factor for many applications. To get fast clustering results, suboptimal solutions are often accepted. This often correlates to *indeterministic* approaches that create different clustering results each time, the algorithm is applied.

There are of course also other categorizations available. The previously mentioned groups are however most common. Based on the requirements for a clustering technique in this work, the discussion of related work will
be done in the following by first describing relevant partitioning algorithms with possible hard- and soft-clustering methods. An analysis of agglomerative hierarchical clustering concepts will follow including related work to create hierarchical soft-clusters.

\section{4.1. Related Work}

The most known and applied (cf. \cite{16}) clustering method is the \textit{k-means algorithm} \cite{74}. The partitioning clustering algorithm is the base algorithm for many other variants. Given an a-priori defined number of clusters $k$, each data element $\vec{x}_i$ is assigned to exactly one cluster $C_j$. Each cluster $C_j$ is defined by the center of mass of its members, defined as centroid $\vec{c}_j$, which causes a prototype-based clustering strategy. After a random initialization, each data point $\vec{x}_i$ is assigned to the nearest centroid $\vec{c}_j$ in several iterations to minimize an a-priori defined target function $SSE$ (sum of squared errors), which typically defines the proximity of all data points to the corresponding centroids:

$$SSE(\mathcal{P}) = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(\vec{c}_j, \vec{x}_i)^2$$ \hspace{1cm} (4.11)

Using the Euclidean distance, the local optimum of $c_j$ can be calculated by taking the first derivative of the SSE and setting it equal to 0:

$$\vec{c}_j = \frac{1}{l} \sum_{i=1}^{l} \vec{x}_i; \quad \vec{x}_i \in C_j, l = |C_j|$$ \hspace{1cm} (4.12)

The \textit{k-means clustering algorithm} “can be treated as an optimization problem” \cite{156} p.513. It uses a gradient descent method (cf. listing 4.1).

Due to the random initialization, different runs typically create different partitions. The \textit{k-means algorithms} is therefore an indeterministic clustering algorithm that may return suboptimal clustering results due to local minima. It is generally assumed that the overall quality improves by executing the algorithm several times and selecting the clustering result with
Input: number of clusters $k$, distance function $d(a,b)$, data elements $\vec{x}_i$, minimal error $\epsilon$, maximum iteration MAX ITERATION

Select $k$ elements randomly as initial centroids $\vec{c}_i$, $i = 1, \ldots, k$

$v = 0$

While $(\text{SSE}(\mathcal{P}) > \epsilon \mid \mid v \leq \text{MAX ITERATION})$

\forall \vec{x}_i \in X:

assign $\vec{x}_i$ to $\vec{c}_u$, $d(\vec{x}_i, \vec{c}_u) \leq d(\vec{x}_i, \vec{c}_l)$; $l = 1, \ldots, k$; $l \neq u$

\forall \vec{c}_j, j = 1, \ldots, k:

recalculate all centroids $\vec{c}_i$, $i = 1, \ldots, k$

$v = v + 1$

Listing 4.1: $k$-means clustering algorithm.

minimum error $\text{SSE}(\mathcal{P})$. It is however not guaranteed that the result improves (cf. [156, p. 502]). To overcome this potential problem, a popular variant initializes only the first centroid randomly and selects the other ones by selecting the data points with furthest distance to already selected centroids. This approach is however critical for outliers. The method is therefore only applied for randomly selected subsets (cf. [156, p. 503]).

Another problem of the $k$-means algorithm is the number of a-priori defined clusters $k$ which cannot be altered during the clustering process. Thus, the algorithm has to be executed several times with different $k$ to find the optimal number of clusters. By using some cluster based metrics, which determine too general or too specific clusters, it is tried to reduce this time-expensive task. Most known are the cohesion and separation metrics [156, pp. 536]:

$$\text{cohesion}(C_i) = \sum_{\vec{x} \in C_i} \sum_{\vec{y} \in C_i} d(\vec{x}, \vec{y})$$ \hspace{1cm} (4.13)

$$\text{separation}(C_i, C_j) = \sum_{\vec{x} \in C_i} \sum_{\vec{y} \in C_j} d(\vec{x}, \vec{y})$$ \hspace{1cm} (4.14)

Different approaches propose to split clusters in case of a too large $\text{SSE}(\mathcal{P})$ or a too large $\text{cohesion}(C_i)$. Too specific clusters $C_i$ and $C_j$ should be
merged. These clusters may be identified by a small separation \( C_i, C_j \) (cf. [156, pp. 499]).

There are many other improvements available, but the \( k \)-means clustering algorithm remains an indeterministic partitioning clustering algorithm which needs at least an estimation of the number of clusters \( k \). It is further a clustering algorithm that assigns a data element to exactly one cluster \( C_i \). It is therefore a crisp clustering algorithm.

**Fuzzy-C-Means**

In 1981, Bezdek [19] extended the classical \( k \)-means algorithm to fuzzy logic. To do this, he included the fuzzy membership \( \mu_{ij} \) of element \( \vec{x}_i \) to cluster \( C_j \) into the target function \( SSE(\mathcal{P}) \):

\[
SSE(\mathcal{P}) = \sum_{i=1}^{n} \sum_{j=1}^{k} \mu_{ij}^b ||\vec{x}_i - \vec{c}_j||^2
\]  

(4.15)

In equation 4.15, \( b > 1 \) denotes a variable called Fuzzifier. It defines the fuzziness degree. For \( b \to 1 \), the algorithm creates crisp clusters.

The Fuzzy-C-Means algorithm tries to minify the SSE similarly to the \( k \)-means algorithm. This is however more complicated than in the \( k \)-means algorithm because some conditions have to be ensured:

1. \( \mu_{ij} \in [0, 1], \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, k\} \)
2. \( \sum_{j=1}^{k} \mu_{ij} = 1, \forall i \in \{1, \ldots, n\} \)
3. \( 0 < \sum_{i=1}^{n} \mu_{ij} < n, \forall j \in \{1, \ldots, k\} \)

Due to these conditions, it is not longer possible to take the first derivative. Bezdek [19] uses the Lagrange method instead to provide formulas for the centroid \( \vec{c}_j \) and the fuzzy membership \( \mu_{ij} \). Using the Euclidean distance as distance function, the centroid \( \vec{c}_j \) is calculated by:
The fuzzy membership $\forall i = \{1,\ldots,n\}, \forall j = \{1,\ldots,k\}$ is calculated by:

$$\mu_{ij} = \frac{1}{\sum_{l=1}^{k} \left( \frac{||\vec{x}_i - \vec{c}_j||^2}{||\vec{x}_i - \vec{c}_l||^2} \right)^{\frac{1}{b-1}}},$$ \hspace{1cm} (4.17)

The overall Fuzzy-C-Means algorithm is similar to the classical $k$-means algorithm and can be seen in listing 4.2.

Mendes and Sacks \cite{115} provided the formula for using the cosine similarity instead of the Euclidean distance, which is more appropriate for text mining applications. Their proposed algorithm Hyper-spherical Fuzzy c-Means (H-FCM) defines the distance function $d(\vec{x}_i, \vec{x}_j)$ using the VSM as:

$$d(\vec{x}_i, \vec{x}_j) = 1 - s(\vec{x}_i, \vec{x}_j) \quad \quad (4.18)$$

$$= 1 - \sum_{l=1}^{t} w_i w_{jl} \quad \quad (4.19)$$

Using this definition in equation 4.15, the sum of squared error is defined as:

\begin{lstlisting}[language=plaintext,frame=lines] Input: number of clusters $c$, Fuzzifier $b > 1$, data elements $\vec{x}_i$, minimal error $\epsilon$, maximum iteration $\text{MAX ITERATION}$

Select $c$ elements randomly as initial centroids $\vec{c}_i, i = 1,\ldots,k$

Initialize membership matrix $\mu$ according to preconditions $v = 0$

While $(\text{SSE}(\mathcal{P}) > \epsilon \mid v \leq \text{MAX ITERATION})$

$\forall \vec{c}_j, j = 1,\ldots,k$: calculate $\vec{c}_j$ according to equation 4.16

re-calculate $\mu$ according to equation 4.17

$v = v + 1$

Listing 4.2: Fuzzy-C-Means clustering algorithm.
\end{lstlisting}
4.1. Related Work

\[ SSE(\mathcal{P}) = \sum_{i=1}^{n} \sum_{j=1}^{k} \mu_{ij}^b (1 - \sum_{l=1}^{t} w_{il}w_{jl}) \]  
(4.20)

To minimize this formula, an additional condition must be true:

4. \[ s(x_i, x_i) = \sum_{l=1}^{t} x_{il}x_{il} = \sum_{l=1}^{t} x_{il}^2 = 1, \quad \forall i \in \{1, \ldots, n\} \]

Using the Lagrange method again, Mendes and Rodriques [115] provide an updated version of the fuzzy membership and the centroids for the cosine based similarity measure. The fuzzy membership is nearly identical to the Euclidean distance version:

\[ \mu_{ij} = \frac{1}{\sum_{m=1}^{k} \left( \frac{1 - \sum_{l=1}^{t} w_{il}w_{ml}}{1 - \sum_{l=1}^{t} w_{il}w_{ml}} \right)^{\frac{1}{b-1}}} \]  
(4.21)

For centroids however, the formula has completely changed:

\[ \bar{c}_j = \sum_{i=1}^{n} \mu_{ij}^b x_i \sqrt{\frac{1}{\sum_{l=1}^{t} (\sum_{i=1}^{n} \mu_{ij}^b x_{il})^2}} \]  
(4.22)

The updated formula makes it possible to apply the Fuzzy-C-Means clustering algorithm in the text mining domain using the well-known VSM and the cosine similarity measure. Independent to the used distance function the shortcomings of the \( k \)-means clustering algorithm are however still present.

**Agglomerative Hierarchical Clustering**

Previously mentioned clustering algorithms divide the data space in (overlapping) clusters. Instead of defining the a-priori unknown number of clusters \( k \), hierarchical algorithms create a sequence of different partitions including an ordering information whether one cluster belongs to another cluster. This implicitly provides an acyclic directed graph. In case of crisp
Input: Singleton clusters $C = C_1, \ldots, C_N$, proximity function $d$

Create proximity matrix $D$ for all $C_i$

While (|C| > 1)

Select $C_i$ and $C_j$ with $d_{ij} = \min(d_{kl})$, $\forall k, l$, $k \neq l$

Delete $C_i$ and $C_j$ in proximity matrix $D$

Insert merged cluster $C_{i+j} = C_i \cup C_j$ into $D$

Listing 4.3: Agglomerative Hierarchical Clustering – Base algorithm

Agglomerative clustering algorithms are typically known as greedy because they combine the clusters $C_i$ and $C_j$ with smallest distance (cf. [113]). At each step, the described algorithm in listing 4.3 decides locally, which clusters should be merged. This distinguishes agglomerative hierarchical clusterings from previous described partitioning clustering algorithms that try to optimize an objective function. The algorithms do not need to solve an optimization problem and in consequence, problems of local minima can be avoided (cf. [156, chap. 8.3.4]).

The hierarchical clustering techniques vary in the way, in which the similarity of a merged cluster $C_{i+j} = C_i \cup C_j$ is inserted in the proximity matrix $D$. While the calculation for singleton clusters (and thus data elements) is already described in section 4.1.1, it is necessary to define in which way the proximity of one data element to a cluster (element-cluster proximity) and of a cluster to another cluster (cluster-cluster proximity) can be calculated. This is done using a linkage metric which differs between available clustering strategies. The most important metrics are presented in the following (cf. [156], [113], [77]):

**Nearest Neighbor (Single Linkage)** The single linkage metric for agglomerative hierarchical clustering algorithms defines the inter-cluster
proximity as the minimum distance between two data elements:

$$d_{i,j} = \min_{\bar{x}_i \in C_i, \bar{x}_j \in C_j} d(\bar{x}_i, \bar{x}_j) \tag{4.23}$$

The concept is based on a graph-based view by merging clusters with the shortest edge between two subsets. It is often called the MIN method, although the name is misleading in case of similarity measures:

$$s_{i,j} = \max_{\bar{x}_i \in C_i, \bar{x}_j \in C_j} s(\bar{x}_i, \bar{x}_j) \tag{4.24}$$

Single Linkage may be problematic, because it tends to create very long clusters. This so-called Chaining Effect is based on the fact that only the nearest data points are taken into regard instead of the complete cluster. It is therefore sensitive to noise and outliers.

**Farthest Neighbour (Complete Linkage)** The complete linkage metric is completely inverse to Single Linkage. The inter-cluster proximity is based on the graph-based concept of merging clusters with the smallest maximum distance between two subsets:

$$d_{i,j} = \max_{\bar{x}_i \in C_i, \bar{x}_j \in C_j} d(\bar{x}_i, \bar{x}_j) \tag{4.25}$$

This method is also called the MAX method, which is misleading in case of similarity measures:

$$s_{i,j} = \min_{\bar{x}_i \in C_i, \bar{x}_j \in C_j} s(\bar{x}_i, \bar{x}_j) \tag{4.26}$$

The Complete Linkage is more robust with regard to noise and outliers. It tends however to break large clusters and favors round clusters.
Average Neighbor (Group Average) The Group Average Linkage metric combines the concepts of Single Linkage and Complete Linkage by taking all data elements into account.

\[
d_{i,j} = \min_{\bar{x}_i \in C_i, \bar{x}_j \in C_j} \frac{1}{|C_i| + |C_j|} \sum_{\bar{x}_i \in C_i} \sum_{\bar{x}_j \in C_j} d(\bar{x}_i, \bar{x}_j) \tag{4.27}
\]

\[
s_{i,j} = \max_{\bar{x}_i \in C_i, \bar{x}_j \in C_j} \frac{1}{|C_i| + |C_j|} \sum_{\bar{x}_i \in C_i} \sum_{\bar{x}_j \in C_j} s(\bar{x}_i, \bar{x}_j) \tag{4.28}
\]

Each data element in cluster \(C_i\) and \(C_j\) is typically weighted equally. A big advantage in agglomerative hierarchical clusters is however, that it would be also possible to weight the data elements according to a-priori defined conditions, e.g. the cluster size. Thus, it is possible to influence the cluster creation.

Centroid Based Linkage Previous linkage metrics are known as graph-based metrics as all data elements are used to calculate the inter-cluster proximity (cf. [156]). It is however also possible to use further geometrical information. Best known is the Centroid based Linkage metric which follows a prototype-based clustering approach. For each cluster \(C_l\) containing \(m_l\) data elements, a centroid \(\hat{C}_l\) is calculated:

\[
\hat{C}_l = \frac{1}{m_l} \sum_{\bar{x}_j \in C_l} \bar{x}_j \tag{4.29}
\]

The inter-cluster distance (similarity) is calculated using the prototypes of the corresponding clusters.

\[
d_{i,j} = \min d(\hat{C}_i, \hat{C}_j) \tag{4.30}
\]

\[
s_{i,j} = \max s(\hat{C}_i, \hat{C}_j) \tag{4.31}
\]
4.1. Related Work

<table>
<thead>
<tr>
<th>Clustering Method</th>
<th>$\alpha_i$</th>
<th>$\alpha_j$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Linkage</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>$-\frac{1}{2}$</td>
</tr>
<tr>
<td>Complete Linkage</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>Group Average</td>
<td>$\frac{m_i + m_j}{m_i + m_j}$</td>
<td>$\frac{m_i}{m_i}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Centroid</td>
<td>$\frac{m_i + m_j}{m_i + m_j}$</td>
<td>$\frac{m_i}{m_i}$</td>
<td>$-m_i m_j$</td>
<td>0</td>
</tr>
<tr>
<td>Ward’s</td>
<td>$\frac{m_i + m_j}{m_i + m_j + m_k}$</td>
<td>$\frac{m_i + m_j}{m_i + m_j + m_k}$</td>
<td>$-m_k$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.3: Table of Lance-Williams coefficients. $m_X$ denotes the number of elements in cluster $C_X$.

**Ward’s Linkage** The Ward’s method [167] defines the inter-cluster proximity “as the increase in the squared error that results when two clusters are merged” [156, p.523]. It is thus the hierarchical pendant to the $k$-means algorithm and implements a gradient optimization strategy.

**Recurrence formula** Hierarchical clustering algorithms are known to be more time and space complex than partitioning clustering algorithms, because it is necessary to keep the data elements in memory to calculate the linkage metric for the merged cluster $C_{i+j}$ (cf. [156, chap. 8.3.1]). Lance and Williams [100, 101] describe a method that makes it possible to calculate the new proximity on base of the proximity of $C_i$ and $C_j$. Their recurrence formula can be applied for all previously mentioned linkage metrics. To be more comparable to the literature, only the distance definitions $d(C_a, C_b)$ are used in the following.

\[d(C_i \cup C_j, C_k) = \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) + \gamma |d(C_i, C_k) - d(C_j, C_k)|\]  

(4.32)

The parameters $\alpha, \beta$ and $\gamma$ depend on the Linkage metric. The values for all metrics are listed in table 4.3.

**Ultrametric properties** The discussed agglomerative hierarchical clustering algorithms insert a new Cluster $C_{ij} = C_i \cup C_j$ into the proximity
matrix $D$ at each step. This process induces a dissimilarity function which describes the Hierarchical Clustering Scheme (HCS). It can be used as bijective function to reproduce the cluster process. In 1967, Johnson [88] analyzed this dissimilarity function and showed that in case of Single Linkage and Complete Linkage this induced dissimilarity function is an ultrametric, a special metric (equation 4.5) that extends the triangle inequality:

\[ \forall \mu, \nu, \xi \in \mathcal{F}, d(\mu, \nu) \leq \max\{d(\mu, \xi), d(\xi, \nu)\} \quad (4.33) \]

The induced ultrametric ensures that the new cluster $C_{i+j}$ is less similar (more distant) to other clusters than $C_i$ and $C_j$ were. The dissimilarity function decreases (increases) monotonically. The behavior supports the original concept of agglomerative hierarchical clustering algorithms that at each step the clusters with the largest similarity (smallest distance) should be merged. “Nonmonotonic increases would imply that more distant points are merged into clusters before having merged points which are actually closer.” [119]. Thus, many researchers postulate that “hierarchical clustering only makes sense if the similarity function is monotonic” [113, p. 502]. Milligan [119] showed however, that this is not true for all common Linkage metrics. By using the recurrence formula provided by Lance and William [100, 101], he extended Johnson’s analysis and proved that except for centroid based algorithms, the linkage variants described by the recurrence formula induces ultrametrics. The often used centroid based approach does not induce a monotonic dissimilarity function. Batagelj [14] further shows that hierarchical clustering strategies based on the recurrence formula are monotonic, if the following conditions hold:

\[
\begin{align*}
(1) & \quad \gamma \geq -\min\{\alpha_i, \alpha_j\} \\
(2) & \quad \alpha_i + \alpha_j \geq 0 \\
(3) & \quad \alpha_i + \alpha_j + \beta \geq 1
\end{align*}
\quad (4.34)
\]

Pyramidal Clustering Assigning a document to only one cluster is an unacceptable situation for real world applications. It might be necessary
to assign a document to more than one cluster because different topics are covered. Diday [48] proposed an extension to the agglomerative hierarchical clustering algorithm to overcome this limitation. Defining a total order $\phi$ on the available data elements, the proposed *pyramidal ascending classification* algorithm (PAC) creates pyramids that extend hierarchies in that way that one cluster may be assigned to two adjacent parent clusters (cf. fig. 4.3).

Figure 4.3: The image shows a crisp (a) and a pyramidal (b) dendrogram. While hierarchical crisp clustering algorithms assign each cluster to exactly one other cluster, multiple assignments are possible in pyramidal clusters, if a special data ordering is given.

Lasch [102] compared the coefficients for crisp clustering algorithms using Lance-Williams’s recurrence formula [100, 101] with the correspondent pyramid coefficients and proved that in case of pyramidal clustering, there is no ultrametric given for Single Linkage and Complete Linkage. Group Average creates an ultrametric. This behavior is based on the fact that the pyramidal clustering algorithm enforces the existence of the total order $\phi$. Nevertheless, Diday [48] shows that there is a bijection between the pyramidal clustering and the induced dissimilarity function.

**Fuzzy-C-Means based Hierarchical Clustering Algorithms** Due to the order limitations, Diday’s pyramidal clustering approach is not very common in practice. This is why different methods for hierarchical fuzzy clusterings have been proposed on top of the well-known Fuzzy-C-means algorithm. Torra [158] proposed a straight-forward bottom-up algorithm
that iteratively applies the Fuzzy-C-Means algorithm to create a hierarchical fuzzy cluster. In each iteration, the algorithm uses a fuzzy partition $\mathcal{P}$ as input that is represented using a set of membership functions $\mu_i$ for $c$ clusters. This number of clusters is selected large (the original paper is not more specific) and is reduced with each iteration until a root cluster is available (cf. listing 4.4).

**Listing 4.4:** Torra’s agglomerative hierarchical fuzzy clustering algorithm on base of the Fuzzy-C-Means algorithm.

Applying the Fuzzy-C-Means algorithm for the data elements is well studied. The appliance of the fuzzy clustering algorithm in upper levels of the hierarchy requires however a definition of how the distance between two fuzzy sets can be calculated. Bloch [21] provides an overview about possible techniques. Torra uses $\alpha$-cuts which transform the fuzzy sets to crisp sets:

$$d(\mu, \nu) = \int_0^1 D(\mu_\alpha, \nu_\alpha) d\alpha$$  \hspace{1cm} (4.35)

The variables $\mu_\alpha$ and $\nu_\alpha$ are $\alpha$-cuts and describe all elements with $\mu_i > \alpha$ or $\nu_i > \alpha$. Using this distance method, Torra applies the Fuzzy-C-Means algorithm on upper hierarchical levels. He missed however to adopt the formulas for the centroids $c_i$ and fuzzy membership $\mu$. According to the results of Mendes and Sacks (cf. [115]) and their extension of the Fuzzy-C-Means algorithm to use the Cosine similarity (section 4.1.2), it is very unlikely that the formulas are identical. The paper results are therefore
not interpretable. The iterative application of the Fuzzy-C-means algorithm additionally leads to hardly reproducible clustering results. Torra also mentions that the overall algorithm is very time consuming due to the multiple computation of fuzzy partitions. This becomes even more important as several runs have to be applied to ensure good results due to random initializations.

Bordogna et al. \cite{158} propose an algorithm similar to the algorithm of Torra \cite{158}. Instead of calculating the Fuzzy-C-Means algorithm on upper hierarchy levels on base of $\alpha$-cuts, the identified centroids are used. Thus, the missing adoption of the centroid and fuzzy membership formula can be avoided. Bordogna et al. used the Cosine similarity however without adapting the formula for the centroids and fuzzy memberships in consequence. The paper results are thus also not interpretable. The algorithm drawbacks are identical.

Mendes and Rodrigues \cite{117} proposed a more scalable hierarchical fuzzy clustering algorithm on top of the Fuzzy-C-Means algorithm. The Hierarchical Hyper-spherical Fuzzy-C-Means algorithm ($H^2$-FCM) uses the Cosine version of the Fuzzy-C-Means algorithm (section 4.1.2). Instead of guessing the a-priori unknown number of clusters $c$, the proposed algorithm strongly over-specifies the number of clusters. This causes a data element $\vec{x}_i$ to be surely assigned to more than one cluster $C_j$. The larger the cluster $C_j$ is, the smaller is the membership value $\mu_{i,j}$. This multiple assignment can be used to create a hierarchy using an asymmetric similarity measure $S$, proposed by Tversky \cite{162}. He describes a hierarchy in terms of inheritance: Each child has at least the properties of the parents. This definition is the same which is used in hierarchical clustering algorithms in which the children clusters represent more specific data groups. A child has thus to be more different to its parent than vice versa. A cluster with centroid $\vec{c}_i$ is a child of cluster with centroid $\vec{c}_j$ if and only if $S(\vec{c}_i, \vec{c}_j) < S(\vec{c}_j, \vec{c}_i)$. Mendes and Rodrigues use the asymmetric similarity defined in equation 4.36 to link cluster centroids hierarchically:
Chapter 4. Hierarchical Fuzzy Clustering

\[
S(\vec{c}_i, \vec{c}_j) = \frac{\sum_{t=1}^{t} \min\{c_{it}, c_{jt}\}}{\sum_{t=1}^{t} c_{it}}
\]  

(4.36)

In a top-down approach, this asymmetric similarity is used to create a cluster hierarchy. \( S \) is calculated between all pairs of cluster centroids. The hierarchy is generated as described in the following:

1. Define \( V_F \) as data set of hierarchically unassigned clusters. \( V_H \) is the set of already assigned clusters, which is initial \( V_H = \emptyset \). In each step, a cluster \( \alpha \in V_F \) is selected with

\[
S(\vec{c}_\alpha, c_i) = \max_{j=1,...,c} S(\vec{c}_\alpha, \vec{c}_j)
\]

2. If \( V_H = \emptyset \) or \( S(\vec{c}_\alpha, \vec{c}_i) < t_{PCS} \), cluster \( \alpha \) is a root cluster. Else find the cluster \( \beta \in V_H \) with the highest value \( S(\vec{c}_\alpha, \vec{c}_\beta) \) and assign \( \alpha \) to cluster \( \beta \).

The higher the threshold variable \( t_{PCS} \) is chosen, the more cluster elements are root nodes. Thus the hierarchical level size can be controlled. The \( H^2-FCM \) algorithm creates in consequence a hierarchical fuzzy cluster with low computational costs by applying the Fuzzy-C-Means algorithm one time. Nevertheless the cluster results and the hierarchy highly depend on the random initialization. This enforces several runs to ensure good clustering results. Due to the indeterministic approach, the clustering results are hardly reproducible.

On top of the Fuzzy-C-Means algorithm, there are also divisive hierarchical clustering approaches available (cf. [158], [26]) which suffer from the same problems and are thus out of focus for this work.
4.2 Generalization of Agglomerative Crisp Clustering Algorithms

By analyzing related clustering approaches (section 4.1), it has to be stated that there is currently no deterministic clustering algorithm available which creates multiple abstraction levels while considering several topics within one data element. Therefore, a new clustering algorithm is developed. It generalizes classical agglomerative hierarchical crisp clustering algorithms to fuzzy logic to support multiple assignments without requiring any data ordering. It is discussed in detail in the following sections. Listing 4.5 presents an abstract overview of the complete agglomerative clustering algorithm approach. As the algorithm is based on a symmetric similarity matrix, the discussion and the algorithm are limited to the upper triangular matrix.

4.2.1 Basic Concept

The proposed clustering algorithm is based on well-known agglomerative hierarchical crisp clustering algorithms. It requires a symmetrical similarity matrix $S$ which is initially filled with all similarities between data points (singleton). At each iteration, the algorithm looks for the highest similarity value $s_{i,j}, i \neq j$ to select two clusters $C_i$ and $C_j$, which are merged in the next step. A classical agglomerative hierarchical crisp clustering algorithm would delete all similarity entries $s_{i,l}$ and $s_{j,l}$ of these clusters. In contrast

Listing 4.5: The abstract hierarchical fuzzy clustering algorithm.
the proposed algorithm does not delete any entry but updates only the similarity value $s_{i,j}$ to 0 (cf. eq. 4.37).

$$S = \begin{bmatrix}
C_1 & \ldots & C_i & \ldots & C_j & \ldots & C_n \\
C_1 & 1 & \ldots & s_{i,1} & \ldots & s_{j,1} & \ldots & s_{n,1} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_i & \vdots & 1 & \ldots & 0 & \ldots & s_{n,i} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_j & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
C_n & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 1
\end{bmatrix} \quad (4.37)$$

With the proposed method it is possible to reuse the remaining similarity values of $C_i$ and $C_j$ for multiple cluster assignments. This important strategy change in agglomerative hierarchical clustering supports the analysis of UGC in two ways:

1. First of all, multiple topics can be analyzed as all similarity values to other clusters are still available. The classical strategy only used the maximal similarity.

2. The clustering process is more robust with regard to newly added documents. The combination of two documents to one cluster is independent to the newly inserted documents as the original similarity values remain in the similarity matrix $S$. It can be expected that topic shifts over time can be handled more easily.

Identically to the classical hierarchical clustering approach, the newly created cluster $C_{i+j} = C_i \cup C_j$ has to be added to the similarity matrix $S$. As the similarity values of $C_i$ and $C_j$ are still available, the matrix dimensions grow by 1 each iteration. This is an important difference to agglomerative crisp clustering algorithms, in which the matrix dimensions are reduced by 1 each iteration. It is in consequence no longer possible to use well-known
4.2. Generalization of Agglomerative Crisp Clustering Algorithms

linkage measures to calculate the similarity between the new cluster \( C_{i+j} \) and an existing cluster \( C_k \). It is necessary to deal with common subgroups. Otherwise the clustering algorithm is not finite. To ensure that \( C_i \) and \( C_j \) will not be re-merged with cluster \( C_{i+j} \), the following conditions must be satisfied:

\[
\begin{align*}
    s_{i+j,i} &= 0 \\ 
    s_{i+j,j} &= 0
\end{align*}
\] (4.38a)

Generally, if a cluster \( C_k \subset C_{i+j} \), the similarity has to be set to 0. In the following, this condition is called the **Special Subgraph Property**. Extending well-known linkage measures in this way, a finite hierarchical clustering algorithm is ensured although the similarity matrix continuously increases. The concept can be seen in equation 4.39 for \( s_{i+j,i} \) and \( s_{i+j,j} \).

\[
S = \begin{bmatrix}
    c_1 & \ldots & c_i & \ldots & c_j & \ldots & c_n & c_{i+j} \\
    c_1 & 1 & \ldots & s_{i,1} & \ldots & s_{j,1} & \ldots & s_{n,1} & s_{i+j,1} \\
    \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    c_i & 1 & \ldots & 0 & \ldots & s_{n,i} & 0 \\
    \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    c_j & 1 & \ldots & s_{n,j} & 0 \\
    \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    c_n & 1 & \ldots & s_{i+j,n} \\
    c_{i+j} & & & & & & & & 1
\end{bmatrix}
\] (4.39)

By allowing multiple cluster assignments, an additional potential overlap has to be considered: \( C_k \not\subset C_{i+j}, C_k \cap C_{i+j} \neq \emptyset \). This special situation is only possible if a cluster was multiply assigned. It depends on the used linkage measure whether it is useful to distinguish between common and uncommon data points in both clusters:
• If Group Average, Minimum Variance or Centroid based linkage metrics are used, it will make sense to use all data points for similarity calculation because outliers are tolerated.

• If Single Linkage is used, it will be helpful not to use common data points. Otherwise, the algorithm would prefer to merge clusters with common nodes.

If the algorithm is wanted to disregard common data points for similarity calculation, virtual clusters $C'_k$ and $C'_{i+j}$ are needed that only consist of uncommon data points. The linkage measures have to be modified to use these virtual clusters for similarity calculation which means that the algorithm follows a non-combinatorial strategy. It already fulfills the Special Subgraph Property because the similarity will be 0 if there are no uncommon data points. In the following, this method is called the **General Subgraph Property**.

The presented clustering process does not necessarily create a complete graph with edges to all nodes. This is due to potentially dissimilar clustering elements that will never be merged in the similarity matrix. To ensure a complete clustering graph, the finally induced root node connects all available cluster nodes with $s = 0$.

**Fuzzifier**

In most application domains and especially in the text mining domain, there are many non-zero similarity entries. It is not appropriate to use all of them to generate a hierarchical cluster graph. On the contrary it should be possible to flexibly adjust the degree of multiple assignments. Fuzzy C-Means introduces therefore a fuzzifier $b$ that scales the fuzziness.

To limit the degree of multiple assignments in the presented approach, there are two extreme possibilities:
4.2. Generalization of Agglomerative Crisp Clustering Algorithms

1. Define a minimum similarity.

2. Define the maximum number of multiple assignments.

Both variants are inappropriate because they ignore the similarity distribution in $S$. Whether a similarity value of one cluster $C_i$ to another cluster $C_k$ is relevant should depend on all available similarity values of cluster $C_i$ instead. Thus, a fuzzifier $f \in [0, 1]$ is introduced. It is multiplied with all similarity values of the selected clusters $C_i$ and $C_j$ except to the self similarity and the maximum similarity of $C_i$ and $C_j$ which is already 0. This causes a similarity value over time:

$$s_{i,l}(t) = s_{i,l}(t-1) \cdot f \quad \forall l \neq i, \quad (4.40a)$$
$$s_{j,l}(t) = s_{j,l}(t-1) \cdot f \quad \forall l \neq j. \quad (4.40b)$$

With the help of the fuzzifier it is possible to specify the degree of multiple assignments. For $f = 1$, all similarity entries in $S$ keep their original value which causes a highly branched cluster graph. A fuzzifier $f = 0$ creates crisp clusters. Thus the presented approach is a generalization of agglomerative hierarchical crisp clustering algorithms. If this similarity modification is applied in each iteration, a more general formula can be presented on top of the original similarity value:

$$s_{i,l}(t) = s_{i,l}(0) \cdot f^\gamma \quad \forall l \neq i, \quad (4.41a)$$
$$s_{j,l}(t) = s_{j,l}(0) \cdot f^\gamma \quad \forall l \neq j. \quad (4.41b)$$

In these equations $\gamma$ denotes how many times the cluster $C_i$ and $C_j$ have been selected for combination. If a a minimum similarity $\epsilon$ is additionally defined, the method can even be improved:

$$\forall l: s_{i,l}(0) \cdot f^\gamma < \epsilon \Rightarrow s_{i,l} = 0, \quad (4.42a)$$
$$\forall l: s_{j,l}(0) \cdot f^\gamma < \epsilon \Rightarrow s_{j,l} = 0. \quad (4.42b)$$
Figure 4.4: During the clustering process, the similarity value $s_{i,j}$ can be stored to the cluster edges to support the calculation of the fuzzy membership of documents $D_i$.

Allowing different fuzzifiers for each cluster level, it is even possible to flexibly adjust the degree of multiple assignment for different use cases.

**Fuzzy Membership**

Until now, the proposed clustering algorithm creates a directed acyclic graph providing a binary membership information whether a node is the child of another node. To fulfill all requirements of a fuzzy clustering, a fuzzy membership value $\mu_{ij}$ has to be provided which represents the more granular information to which degree a cluster $C_i$ belongs to a cluster $C_{i+j}$. This information can be gained by the clustering process itself: It merges two clusters $C_i$ and $C_j$ with similarity $s_{i,j}$ to cluster $C_{i+j}$. This similarity value can be stored in the clustering graph at the corresponding cluster edges as weights (cf. fig. 4.4). After the complete cluster is generated and all weights are assigned in the cluster graph, the weight can be normalized to satisfy the fuzzy cluster property of equation 4.10. It now represents the fuzzy membership $\mu_{C_i,C_{i+j}}$ of cluster $C_i$ to cluster $C_{i+j}$.

In the following, $C_i$ is a cluster node, $C_{i+j}$ is the parent cluster generated of $C_i$ and $C_j$. $C_p, p = 1, \ldots, m$ are all clusters that have been merged with $C_i$ to other parents $C_{i+p}$. The membership of $C_i$ to its parent $C_{i+j}$ can be calculated in this way:
4.2. Generalization of Agglomerative Crisp Clustering Algorithms

\[ \mu_{C_i, C_{i+j}} = \frac{S_{i,j}}{\sum_p S_{i,p}} \]  

(4.43)

In fig. 4.4, this means:

\[ \mu_{C_D, C_{D+E}} = \frac{s_{D,E}}{s_{D,E}} \]
\[ \mu_{C_E, C_{D+E}} = \frac{s_{D,E}}{s_{D,E} + s_{E,F}} \]
\[ \mu_{C_E, C_{E+F}} = \frac{s_{E,F}}{s_{D,E} + s_{E,F}} \]
\[ \mu_{C_F, C_{E+F}} = \frac{s_{E,F}}{s_{E,F}} \]

Using this fuzzy membership calculation in combination with the fuzzifier \( f \), a flexible deterministic agglomerative hierarchical clustering algorithm is presented that extends well-known clustering methods to fuzzy logic. Figure 4.5 presents two small clustering examples with \( f = 0.0 \) and \( f = 0.5 \). Both are generated by analyzing 300 randomly selected documents out of the RCV2 text corpus\(^3\). This corpus provides news articles with multiple topics. In the example, the fuzziness is only used on document level so that only documents can be assigned to multiple clusters. On higher hierarchy levels, crisp clustering is applied to keep the clustering graph more interpretable. To show the benefit of multiple assignments some documents have been marked as red in the crisp cluster tree. The same documents are also highlighted in the fuzzy cluster graph. Related cluster nodes are highlighted in blue. It can be seen that there are four subgraphs in both graphs. In the fuzzy cluster (b), these graphs are related with each other, there is an inter-graph similarity. In the crisp cluster (a), this similarity is missing. Furthermore, one graph is only a singleton in the crisp graph because the similarity to other documents cannot be considered. The fuzzy mode supports the consideration of all relevant similarities. Thus, there is a complete subgraph in the fuzzy cluster graph (b).

\(^3\text{http://trec.nist.gov/data/reuters/reuters.html}\)
Figure 4.5: Hierarchical clustering in crisp (a) and fuzzy mode (b) for 300 documents ($f = 0.5$ on document level, $f = 0.0$ else). By highlighting identical documents (red) in both graphs, it can be seen that the fuzzy cluster graph (b) can consider multiple similarities by inserting additional cluster nodes (blue). Considering only one similarity per document, it could happen that documents are represented as singleton clusters although there is some similarity (e.g. upper-right corner in the crisp graph).
4.2.2 Monotonic Behavior

Johnson and Milligan showed that hierarchical crisp clustering algorithms are monotonic except for centroid and median based linkage measures (cf. [88, 119]). In case of hierarchical crisp clustering, this means that all these linkage measures induce an ultrametric (cf. [88]). Pyramidal clustering – introduced by Diday – no longer induces an ultrametric but a Robinson metric while creating still monotonic clusters (cf. [48]). Inducing a monotonic increasing dissimilarity measure is an important theoretical feature of agglomerative hierarchical clustering algorithms because the monotonic behavior supports the theoretical idea of merging these clusters with minimal distance (maximal similarity). “Nonmonotonic increases would imply that more distant points are merged into clusters before having merged points which are actually closer.” [119].

As the proposed clustering algorithm is a generalization of already available clustering algorithms, it is necessary to analyze the monotonic behavior for the presented approach. It is obvious that it does not induce any metric because the triangular inequality is no longer valid in case of multiple assignments. Johnson [88] proved the equality of ultrametric and monotonic behavior however only for crisp clustering approaches. A monotonic behavior could still be possible in fuzzy clusters. The following analyzes whether the created hierarchical fuzzy clusters are monotonic for different linkage measures. Centroid and median based linkage methods are not analyzed because these measures are already not monotonic in non-overlapping clusters (cf. [119]). Similar to crisp clustering algorithms, the similarity $s_{i+j,i}$ for $C_i \subset C_{i+j}$ has to be 0. Thus the distinction has only to deal with overlapping clusters because otherwise the recurrence relation introduced by Lance and Williams [100, 101] could be used. To prove monotonic clusters, in the following a dissimilarity measure $d$ is used to make comparison to literature easier. The General Subgraph Property and the Special Subgraph Property are handled separately. It is proven that in case of the Special Subgraph Property, the created clusters still induce monotonic clusters while for the General Subgraph Property this is only true for Single and Complete Link-
**Clustering Method** | $\alpha_i$ | $\alpha_j$ | $\beta$ | $\gamma$
---|---|---|---|---
Single Linkage | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $-\frac{1}{2}$
Complete Linkage | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$
Group Average | $\frac{n_i + n_j - n_{ij}}{n_i + n_j + n_k} | \frac{n_i + n_j - n_{ij}}{n_i + n_j + n_k} | 0 | 0
Minimum Variance | $\frac{n_i + n_j + n_k + m}{n_i + n_j + n_k + m} | \frac{n_i + n_j + n_k + m}{n_i + n_j + n_k + m} | 0 | 0

Table 4.4: Updated Table of Lance-Williams coefficients in multiple assignment mode. $n_X$ denotes the number of elements in cluster $C_X$. $n_{XY}$ denotes the number of elements common in cluster $C_X$ and $C_Y$. $m = -n_{ij} - n_{ik} - n_{jk} + n_{ijk}$

**Special Subgraph Property**

When applying only the Special Subgraph Property all elements are taken into account to calculate the dissimilarity to other clusters. As the Single Linkage and the Complete Linkage are independent of the number of cluster elements, the recurrence relations \[100, 101\] can be reused for these metrics which therefore create monotonic clusters \[119\].

The recurrence relations for Group Average and Minimum Variance (also known as Ward method) have to be adapted because the original formulas do not consider common elements which only appear once in the new cluster. The number of common elements has to be removed in $\alpha_i$, $\alpha_j$ and $\beta$ ($\gamma = 0$). The updated recurrence formula can be seen in table 4.4.

Assuming non-empty clusters, it is sure that $\alpha_i > 0$, $\alpha_j > 0$ and $\gamma \geq -\min(\alpha_i, \alpha_j)$ for both measures. Furthermore, it can be stated that $\alpha_i + \alpha_j + \beta \geq 1$ in the Group Average linkage:

$$\frac{\alpha_i + \alpha_j + \beta}{n_i + n_j} \geq 1$$

$$\frac{n_i + n_j}{n_i + n_j - n_{ij}} \geq 1$$

$$0 \geq -n_{ij}$$
4.2. Generalization of Agglomerative Crisp Clustering Algorithms

By assuming that $\alpha_i + \alpha_j + \beta \geq 1$, the inequality for the Minimum Variance can be written as:

$$\frac{\alpha_i + \alpha_j + \beta}{n_i + n_j + n_k - n_{ik} - n_{jk}} \geq 1 \geq \frac{n_i + n_j + n_k - n_{ij} - n_{ik} - n_{jk} + n_{ijk}}{n_{ij}} \geq n_{ijk}$$

The intersection $n_{ijk}$ of all clusters $i,j$ and $k$ is surely equal or smaller than the intersection $n_{ij}$ of cluster $i$ and $j$. Thus, this inequality is true. According to Batagelj [14], both measures are monotonic in consequence.

□

General Subgraph Property:

When disregarding common elements, it is not possible to use the recurrence relation [100, 101] for overlapping clusters anymore because $d_{i,k}$, $d_{j,k}$ and $d_{i,j}$ may change. In this situation, the algorithm follows a non-combinatorial strategy so that the original data must be retained.

To analyze monotonic behavior, it has to be checked whether $d_{i,j} \leq d_{i+j,k}$ for each created cluster. If this is true, the algorithm will create monotonic clusters. The analysis has only to care about dissimilarities $d_{i+j,k}$ with $C_{i+j} \cap C_k \neq \emptyset$.

• Single Linkage: The clustering algorithm ensures that $d_{i,j} \leq d_{i,k}$ and $d_{i,j} \leq d_{j,k}$. Without loss of generality assume that $e \in C_k$ is the nearest element to cluster $C_i$ and that $e$ is also element of cluster $C_j$. This means that $d_{i+j,k} > d_{i,k}$ if common elements are not used. Otherwise if $e \notin C_j$, the dissimilarity $d_{i+j,k} = \min(d_{i,k}, d_{j,k}) \geq d_{i,j}$. Therefore it is clear that $d_{i,j} \leq d_{i+j,k}$ and the created clustering is monotonic.
• **Complete Linkage**: The clustering algorithm ensures that $d_{i,j} \leq d_{i,k}$ and $d_{i,j} \leq d_{j,k}$. Assuming that $e \in C_k$ is the furthest element to cluster $C_i$ and that $e$ is also element of cluster $C_j$. Then $d_{i,k} \leq d_{i,j}$ which is only possible if both dissimilarities are equal. Otherwise if $e \not\in C_j$, the dissimilarity $d_{i+j,k} = \max(d_{i,k}, d_{j,k}) \geq d_{i,j}$. Thus $d_{i+j,k} \geq d_{i,j}$ and the created clustering is monotonic.

• **Group Average, Weighted Average and Minimum Variance**: Every measure uses all elements available in the cluster. It cannot be ensured to get monotonic clusters when using only a part of each cluster for calculating dissimilarity values.

Except for Single Linkage and Complete Linkage, the created fuzzy clusters are not necessarily monotonic anymore.

\[\square\]

### 4.3 Topic Groups Generation

At each step, agglomerative hierarchical clustering algorithms merge two clusters and create thus a binary cluster graph. The induced cluster hierarchy is not necessarily representing topical groups. As it can be seen in figure 4.5, there are several cluster nodes (blue) for which the user has to decide himself whether the cluster represents a valid abstraction level or if it just exists due to the clustering process. The potential irrelevant abstraction levels increase with fuzzyfier $f$. To overcome this shortcoming, an additional step is necessary to generate topic groups with more than two elements. This is done by a special pruning process. It reduces the graph hierarchy to a predefined hierarchy level $h$ by recursively merging clusters. This is equivalent to removing cluster nodes in the cluster graph.

Each cluster node $C_{i+j}$ is extended with two similarity values: the **bottom similarity** $s_{i+j}^b$ and the **top similarity** $s_{i+j}^t$. Each similarity value is initialized
with the corresponding similarity value \( s_{i,j} = s(C_i, C_j) \) that caused the creation of cluster node \( C_{i+j} \) (cf. fig. 4.6(a)):

\[
s_{i+j}^b = s(C_i, C_j) \tag{4.44a}
\]
\[
s_{i+j}^t = s(C_i, C_j) \tag{4.44b}
\]

Removing any cluster node \( C_k \) influences the overall cluster structure. To minify this structural modification, a cost function \( c_k \) is introduced. It is defined as:

\[
c_k = s_k^t - \frac{\sum_{\text{parents}} s_p^b}{|\text{parents}|} \tag{4.45}
\]

In crisp clusters, this cost function for cluster \( C_k \) and its parent \( C_p \) can be simply written as:

\[
c_k = s_k^t - s_p^b \tag{4.46}
\]

The cluster node \( C_k \) is removed after replacing the bottom similarity of all parent clusters \( C_p \) with the bottom similarity of cluster \( C_k \). This corresponds to a merging process in the dendrogram (cf. fig. 4.6). For all child nodes \( C_c \) of \( C_k \) the Fuzzy memberships \( \mu_{C_k,c} \) have to be multiplied with the membership \( \mu_{C_k,C_p} \) for all parents \( C_p \).

The pruning process aims to merge binary clusters to topical groups. A minimal modification of the clustering structure can be achieved by reducing the deletion costs. Let \( h \) be the maximum distance between two nodes in the created directed acyclic graph. While \( h \) is greater than a predefined level, the nodes with minimal deletion costs are iteratively removed. In consequence clusters are merged according to the cost function. The practical usage has shown an appropriate maximal cluster level size of \( h_{\text{max}} \in [5; 10] \) which is closer to real world hierarchy structures (cf. [64 sec. 5.2]). It is easier to interpret by users.
Figure 4.6: The cost function $c_k$ specifies the costs with which the cluster structure can be modified with minimal structural effects (a). Merging the cluster with minimal distance (c) corresponds to the deletion of the corresponding cluster node in the node representation (b & d).

It is obvious that the described algorithm not necessarily reduces the hierarchy size $h$ in each iteration. On the contrary, the merging step considers only the removing costs $c_k$. This is quite important as an additional consideration of the hierarchy size $h$ would cause an unexpected merging behavior: Long graph paths would be merged in advance. This is not wanted at all.

Figure 4.7 presents a real world example based on a very small document set of 300 documents. It is based on the cluster graphs shown in figure 4.5 which are now limited to the maximum hierarchy level $h_{\text{max}} = 5$. Figure 4.7 (a) shows the crisp cluster tree. In figure 4.7 (b) the fuzzy clustering result is visible ($f = 0.5$ for the document level, $f = 0.0$ else). The results show that due to the merging process, clusters with more than two elements
4.3. Topic Groups Generation

Figure 4.7: By pruning the cluster graphs of figure 4.5, many cluster nodes (blue) can be removed. Especially upper clusters contain more than two cluster elements.
are available. This is especially true for upper hierarchy levels in which the modification costs are lower than on document level.

It is important to mention that the created topical clusters are quite small in this example. The original crisp cluster tree has a hierarchy size of $h = 8$, the fuzzy cluster graph has a cluster depth of $h = 12$. Thus, not many clusters have to be merged. It is obvious that a cluster graph with more than 1,000 documents provides a larger cluster depth. It would be necessary to merge more clusters and thus bigger topic groups are generated. It is however not possible to display such cluster graphs in a classical graph view any more which is why this small data set was used instead. Figure 4.8 shows a larger crisp cluster with more than 10,000 documents and a maximum hierarchy level $h_{\text{max}} = 5$. It is visualized with a squarified cushion treemap algorithm (cf. [152, 165, 30]). The topic groups can be detected very well.

### 4.4 Evaluation

The evaluation of clustering results is a heavily discussed topic in the scientific community. This is due to the typical use case of applied clustering algorithms that is an exploratory data analysis in which an additional evaluation does not bring any benefit (cf. [156, sect. 8.5]). In addition, the success of a clustering algorithm depends not only on the created clustering results but also on the behavior of the clustering approach that has to fit to certain a-priori given circumstances and requirements. The requirements of the clustering approach in Social Media environments have already been discussed in the introduction of this chapter. The applied algorithm has to:

1. provide different abstraction levels,
2. create deterministic clustering results,
3. group documents according to the discussed topics,
4. take care for potentially multiple topics within one document,
5. provide stable clustering results even if new documents are added.
4.4. Evaluation

Figure 4.8: In a squarified cushion treemap, each document is visualized as a small box. The hierarchical cluster structures are visualized using shading effects. They can be interactively highlighted by further colored borders. The presented hierarchical crisp cluster with more than 10,000 documents was limited to $h_{\text{max}} = 5$. The pruning process generates clusters with much more than two elements per abstraction level.
The algorithm design already ensures that the requirements 1 and 2 are fulfilled: Agglomerative hierarchical clustering algorithms are known to create different abstraction levels. The described Linkage Metrics are deterministic and thus create reproducible results. The requirements 3, 4, and 5 have to be analyzed using different evaluation techniques. Similar to the numerous clustering algorithms, there are many different evaluation approaches. To provide a clear distinction of related work and own provided approaches and results, the following discussion will be separated in two parts: In section 4.4.1 related evaluation techniques used in the literature are presented. Then the applied evaluation technique is described in detail, followed by the corresponding evaluation results.

4.4.1 Related Work

There are three different categories of evaluation methods (cf. [156, chap. 8.5]):

**Unsupervised** Using the available internal data of each cluster, it is possible to evaluate the cluster in an unsupervised manner. Very common is the usage of the $SSE$, the cluster cohesion (eq. 4.13) to measure the cluster compactness and the cluster separation (eq. 4.14) to measure the cluster isolation. Special variations have been presented for fuzzy clustering algorithms (e.g. [128, 173, 61, 170, 18, 27]).

**Supervised** In contrast to unsupervised evaluation methods, supervised approaches rely on externally given data that was not used by the clustering algorithm. An example is the entropy, which measures the agreement of cluster labels with externally supplied classes (cf. [156, p. 535]).

**Relative** A relative cluster evaluation uses supervised or unsupervised evaluation measures to compare different clustering results. The technique is typically used for indeterministic clustering approaches in which multiple clustering results need to be investigated to overcome local minima (e.g. the $k$-means algorithm, cf. [156, p. 535]).
In case of fuzzy clustering algorithms unsupervised measures like separation and cohesion are difficult to measure. Although special fuzzy evaluation techniques have been published, it was shown that “all of the existing indices work well on data sets containing only well-separated [...] clusters, but many of them fail if the data set contains overlapping clusters.” [27]. Furthermore, it is difficult to mathematically proof the superiority of an evaluation measure. The performance of the quality measures can only be estimated using externally given data (cf. [27]). Relative cluster evaluation does not provide any information how well a cluster represents real data structures. That is why related work is evaluated using supervised evaluation techniques (e.g. [116, 117, 25]). According to [156, chap. 8.5.7], the available methods can be separated into two different approaches:

### Classification Evaluation

is a very common evaluation technique that originates from the analysis of classification results. It measures the extend that a cluster $C_T$ contains data elements $x_i$ of a given class $l$, often described as reference cluster $C_R$. There is a large number of well-analyzed measures, e.g. precision, recall or the F-measure to analyze each cluster.

Precision is defined as:

$$P_{TR} = \frac{|C_T \cap C_R|}{|C_T|}$$

(4.47)

where $|C_T \cap C_R|$ denotes the number of elements in cluster $C_T$ and $C_R$, $|C_T|$ specifies the total number of elements in cluster $C_T$.

Recall is defined as:

$$R_{TR} = \frac{|C_T \cap C_R|}{|C_R|}$$

(4.48)

where $|C_R|$ specifies the total number of elements in cluster $C_R$.

The F-measure combines the precision and the recall and is defined as:

$$F_{TR}^{\xi} = \frac{(\xi^2 + 1) \cdot P_{TR} \cdot R_{TR}}{\xi^2 \cdot P_{TR} + R_{TR}}$$

(4.49)
where $\xi$ is a weighting factor to control the relative weight of precision and recall.

The analysis results for a complete partition $P$ are commonly generated by calculating the weighted average for all clusters (cf. [116]):

$$
P = \frac{\sum_{i=1}^{c} N_i P_{Ti}}{\sum_{i=1}^{c} N_i}$$

(4.50)

$$
R = \frac{\sum_{i=1}^{c} N_i R_{Ti}}{\sum_{i=1}^{c} N_i}
$$

(4.51)

**Similarity based Evaluation** measures how many objects within one cluster have the same label. A well-known representative of this evaluation technique is the $\Gamma$ statistic [65, 83, 177]. It defines a matrix $A$ as similarity matrix with $s_{i,j} = 1$ for all elements $i$ and $j \in [1; n]$ within the same cluster and 0 otherwise. Set matrix $B$ as similarity matrix with $s_{i,j} = 1$ for all elements $i$ and $j$ with the same label, 0 otherwise. The $\Gamma$ statistic is defined as correlation between $A$ and $B$ and thus analyzes the agreement of partition $P$ with externally provided data.

$$
\Gamma = 2 \frac{n(n-1)}{n^2-1} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} A_{ij} B_{ij}
$$

(4.52)

A similar concept is applied in the **Rand Index** [84]. It defines four quantities for all pairs of distinct objects (cf. [156]):

$f_{00}$: number of data elements with different label and cluster.

$f_{01}$: number of data elements with different label, but identical cluster.

$f_{10}$: number of data elements with identical label, but different cluster.

$f_{11}$: number of data elements with identical label and cluster.
The Rand Index is defined as:

\[ R_I = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \]  
(4.53)

This formula is identical to the Simple Matching Coefficient [113, chap. 8.5] that is used to calculate the similarity of binary vectors. Milligan and Cooper [120] showed however that the Rand Index correlates with the number of clusters \( c \).

Ben-Hur et al. [15] extended the evaluation concept to detect the most appropriate partition in a hierarchical cluster tree. For each partition \( \mathcal{P}_i \) in this cluster tree, they calculate a similarity matrix \( A_i \) according to the definition above. Then they measure the similarity of two matrices using the simple matching coefficient to estimate the partition with the biggest similarity to neighbor partitions. This partition is the most stable partition in the cluster tree. The method is often applied to estimate the stability of two partitions \( \mathcal{P}_i \) and \( \mathcal{P}_j \). Although the concept is based on supervised evaluation techniques it is a relative method as no external data is used.

Most clustering evaluation approaches are historically based on crisp clustering algorithms. Fuzzy clustering algorithms are commonly defuzzified to be able to reuse established methods. Mendes and Sacks [116] propose to use various membership thresholds to calculate the precision and recall for different \( \alpha \)-cuts. This method is however not able to take the benefits of multiple assignments into account. Bäck and Hussain [7] extended therefore similarity based evaluation methods to fuzzy logic. In a first step, they use an alternative definition of the Rand Index, which can simply be calculated using a matching matrix \( M = (m_{ij}) = U V' \). \( U \) and \( V \) denotes the membership matrix of cluster \( C_i \) and \( C_j \) with \( u_{ik} \in \{0,1\} \). Then the Rand index can be defined as:

\[ R = [T - \frac{1}{2} P - \frac{1}{2} Q + \frac{n(n-1)}{2}] \frac{2}{n(n-1)} \]  
(4.54)
where \( T = (\sum_{i=1}^{c} \sum_{j=1}^{c} m_{ij}^2) - n \), \( P = \sum_{i=1}^{c}(\sum_{j=1}^{c} m_{ij})^2 - n \) and \( Q = \sum_{j=1}^{c}(\sum_{i=1}^{c} m_{ij})^2 - n \).

Instead of using crisp membership matrices \( U \) and \( V \), Bäck and Hussain propose to use fuzzy membership matrices. The Rand Index can be calculated analogously. In their publication, they note however that for real fuzzy partitions, \( R(U,V) \neq 1 \) so that the fuzzified Rand Index can only be used to compare a fuzzy partition with a crisp partition. In recent work, this was very useful because most evaluations were done using crisp partitions as reference.

Bäck and Hussain also investigate the evaluation criterion of Fowlkes and Mallows [58]:

\[
B = \frac{T}{\sqrt{PQ}} \tag{4.55}
\]

In contrast to the Rand Index, \( B \) decreases with increasing number of clusters \( c \). Using fuzzy partitions, it can be shown that the upper limit of 1 is still available, but there is no lower level any more. In addition, \( B(U,V) \neq 1 \) in case of two fuzzy partitions. This is why a new evaluation measure was introduced: the MC Index: \( U = (u_{ik}) \) and \( V = (v_{ik}) \) are both fuzzy partitions. The MC Index is defined as:

\[
MC = 1 - \frac{1}{2n} \sum_{i=1}^{c} \sum_{j=1}^{c} (v_{ik} - u_{jk})^2 \tag{4.56}
\]

Similar to the Rand Index and the evaluation criterion of Fowlkes and Mallows, the MC index is in \([0, 1]\). There are however some limitations using the approach. First of all it requires both partitions to have the identical number of clusters \( c \), which is very unlikely in real world applications in which the number of clusters is a-priori unknown. In addition, most available standard corpora do not provide any reference cluster. The Reuters RCV Corpus\(^4\) for example only provides multiply tagged documents. Mul-

---
\[^4\]http://about.reuters.com/researchandstandards/corpus/
4.4. Evaluation

Multiple labels however cannot be considered with this evaluation index which makes it inappropriate for fuzzy clustering algorithms.

Another approach was presented by Nuovo and Catania [126]: the DNC Index. The evaluation measure uses the (un)certainty information, to which degree a document is assigned to one pattern $x_j$. They define the degree of confidence of the $j$-th pattern as:

$$\Delta_j = (u_{aj} - w_{bj})$$ (4.57)

Comparing the membership function with labeled information and a threshold $\alpha$, the confidence is classified into good, bad and uncertain, which is combined to one overall index. The index is however not appropriate for real fuzzy clusters because it punishes equal patterns for clusters. It is not unlikely that one topic is as important as another topic in one document. In addition, the algorithm also requires the definition of a reference clustering.

All proposed evaluation measures are limited to evaluate partitions. Recent hierarchical fuzzy clustering algorithms have been analyzed by reducing the cluster tree to a partition and by reusing partition based classification (e.g. [158]) or similarity measures (e.g. [26]). There is no evaluation technique known to the author that analyzes hierarchical fuzzy clusters.

4.4.2 The Cosine Quality Index

By investigating related work, it can be seen that there is no appropriate evaluation technique to analyze hierarchical fuzzy clustering results. Partitioned fuzzy clusters have been evaluated using crisp evaluation measures or by applying modified indices that punish multiple document assignments.

To measure the quality of the presented hierarchical clustering algorithm, a new evaluation method is therefore proposed in the following. It supports the analysis of hierarchical crisp and fuzzy cluster graphs so that the advantages of the algorithm become visible. Instead of relying on externally provided reference clusters, it uses labels for each document. This is a much
Figure 4.9: After assigning the externally provided labels to the document clusters (singletons), each cluster $C_i$ in the graph can be labeled by following the weighted edges and propagating the child labels according to the membership value. This example shows a crisp cluster example with multiply labeled documents and clusters.

more common situation in textual data environments, especially in Social Media, in which users are able to tag any type of data$^5$.

The evaluation measure is applied to a directed, weighted, acyclic cluster graph as generated by the proposed clustering algorithm. For each document $d_j$, each label $l_i$ is initially assigned to the corresponding singleton cluster $C_{d_j}^{l_i}$ in this graph. Each label $l_i$ is weighted with factor $w_i = 1$.

\[
C_{d_j}^{l_i} = \{l_i, l_j, \ldots\} \tag{4.58}
\]

Following the graph edges, it is possible to propagate the labels to related clusters according to the membership values by multiplying the fuzzy membership $\mu_{i,j}$ with the label weight $w_i$. Thus, each cluster $C_i$ gets a weighted set of labels (cf. the crisp example in fig. 4.9) that are accumulated among the children:

\[
C_i = \{w_i l_i, w_j l_j, \ldots\} \tag{4.59}
\]

$^5$ e.g. links [http://delicious.com], videos [http://youtube.com] or discussions [http://stackoverflow.com]
4.4. Evaluation

The method assigns a weighted label set to each cluster $C_i$ in the cluster graph. It can be used to estimate the individual cluster quality by using the concepts of the Vector Space Model (cf. [131]): Assuming that each label is independent of each other\footnote{Similar to words in the VSM, this may not be true for labels.}, each label represents a dimension in the Cartesian coordinate system. The weight specifies the dimension’s value. By calculating the cosine similarity of the document $\tilde{C}_d^j = \{l_i, l_j, \ldots\}$ and the cluster $\tilde{C}_i = \{w_i, w_j, l_j, \ldots\}$ it can be determined how well the document $d_j$ fits into the cluster $C_i$:

$$Q_{\cos}(d_j, C_i) = \tilde{C}_d^j \ast \tilde{C}_i \quad (4.60)$$

The closer the value of $Q_{\cos}(d_j, C_i)$ is to 1 for a given document $d_j$, the better it is represented by the cluster $C_i$. As the labels of cluster $C_i$ are based on the labels of all contained documents, this corresponds to a homogeneous cluster. The overall clustering quality can be estimated by calculating the average of $Q_{\cos}$ for all documents and clusters in the cluster graph. In contrast to related work, this thesis prefers however to provide a deeper statistical insight into the analysis results. Therefore, there is no single quality index provided to estimate the overall cluster quality. Instead, different $\alpha$-quantiles are used, namely the 0.25- and 0.75-quantiles and the median (0.5-quantile). These measures are known to be robust with regard to outliers. Additionally, the minimum and the maximum value is provided to give a more detailed view on the clustering behavior. Due to the calculation method, the quality measure is called the **Cosine quality index** in the following.

The proposed quality measure is a precision based measure. It only measures the representation quality of each document in all clusters in the cluster graph. It does not care about recall. Therefore, it is necessary to check if all relevant elements are assigned to a cluster. Unfortunately it is quite difficult in a hierarchical cluster graph to define whether an element belongs to a cluster. Figure 4.9 shows an example in which it is not clear if
the documents $C_1$ and $C_3$ should be already merged on level 2 or on level 3. It depends on the externally given conditions. Instead of defining a recall, the presented evaluation technique uses the fuzziness behavior by counting the number of multiple assignments per document. The more clusters are connected to a document without reducing $Q_{cos}$, the better the overall clustering result is. The same holds for the cluster size: The more elements are within a cluster without reducing the cluster quality, the better the result is. Similar to the statistical analysis of the Cosine quality index, the $\alpha$-quantiles will be investigated to provide a better insight in the clustering behavior.

### 4.4.3 Topical Quality Analysis

The first analysis investigates the overall clustering quality. It has to be analyzed whether the degree of multiple assignment influences the representation quality of a document. Due to the used linkage metric, it is assumed that there are no quality changes. Otherwise the similarity measure would be misleading. An increased fuzzifier should provide however the possibility to assign a document to more than one cluster. It has to be expected in consequence that the degree of multiple assignment increases, leading either to larger clusters or to more clusters. The pruning process has to be further investigated to analyze whether the generated clusters represent topical groups.

To provide reproducible evaluation results, the publicly available Reuters RCV1 and RCV2 news corpora are used. These corpora are an established data set that is often used in many different Information Retrieval publications. Each corpus contains multiple sub corpora for different languages. Each document in these sub corpora is multiply labeled.

For the quality and topic evaluation, seven different document collections have been randomly generated using an English and a German sub corpus. Each collection consists of between 9,350 and 26,501 documents. The fuzzy

\footnote{http://trec.nist.gov/data/reuters/reuters.html}
behavior is analyzed for $f \in \{0.0, 0.1, \ldots, 0.8\}$ on document level. Upper hierarchy levels are clustered with $f = 0$ (in this scenario, it is only necessary to multiply assign documents, not cluster structures). To generate monotonic clusters, the algorithm uses the Group Average linkage with the Special Subgraph Property. All clusters are limited to level size $h = 5$ to take the topic generation into account.

Table 4.5 shows the Cosine quality for all clusters of level 1 (started from the document level) and different fuzzifiers $f$. As it can be seen, the Cosine quality is very high for all fuzzifiers $f$. This result is expected as the clusters have been created by combining similar documents. A worse Cosine quality would imply that the linkage metric is inappropriate to investigate textual data. With varying fuzzifier, the overall quality for all documents does not change much. Nearly 75% of the data is not affected by any quality change. For 25% of the data the Cosine quality decreases slightly with increased fuzzifier $f$.

<table>
<thead>
<tr>
<th>Overall Quality of hierarchy level $h = 1$ for $f$</th>
<th>min</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.01</td>
<td>0.91</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>0.91</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.2</td>
<td>0.00</td>
<td>0.90</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.3</td>
<td>0.00</td>
<td>0.90</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.4</td>
<td>0.00</td>
<td>0.90</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.5</td>
<td>0.00</td>
<td>0.89</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.6</td>
<td>0.00</td>
<td>0.89</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.7</td>
<td>0.00</td>
<td>0.88</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.8</td>
<td>0.00</td>
<td>0.88</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 4.5: The Cosine quality measure for all clusters of level 1 is analyzed using different statistical indicators, namely the minimum, the 0.25-quartile (Q1), the median (Q2), the 0.75-quartile (Q3) and the maximum. It can be seen that the quality is nearly constant for all fuzzifiers $f$.

By investigating the fuzziness behavior (table 4.6), it can be seen that 25% of the documents are multiply assigned. This applies already for very small
fuzzifiers and increases monotonically with $f$.

<table>
<thead>
<tr>
<th>Fuzziness</th>
<th>$f$</th>
<th>min</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0.6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>0.7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 4.6: By investigating the fuzziness behavior, it can be seen that nearly 25% of the documents are multiply assigned with increased fuzzifier $f$. The degree of multiple assignments directly correlates with $f$.

The fuzziness behavior could lead to the assumption that the quality decrease is caused by multiple document assignments. Analyzing only the elements which have been multiply assigned, this quality reduction cannot be explained however (table 4.7). The results vitiate the assumption that the quality decrease may be caused by multiple assignments. Therefore, the pruning process must cause these quality changes: Higher fuzzifier values create larger clusters before the pruning process is applied. More clusters have to be merged to limit the maximum cluster size $h_{\text{max}}$. The quality differences are however very small. The multiple assignment does not negatively influence the overall clustering results.

The constant cluster quality is not necessarily based on equal topical groups. It could be caused by smaller cluster sizes for larger fuzzifiers. Analyzing the cluster elements of hierarchy level 1 and counting each assignment for a cluster, this assumption cannot been approved (table 4.8). The cluster sizes slightly grow with increasing $f$. Thus, it is ensured that the quality of topical groups are equal for all fuzzifiers $f$.

By investigating the number of clusters per hierarchy level (table 4.9), it
4.4. Evaluation

<table>
<thead>
<tr>
<th>Cluster Size Level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 4.8: The cluster sizes for level 1 grow with increasing fuzzifier $f$. Thus it can be ensured that constant cluster quality is not due to decreased cluster sizes.

Table 4.7: The Cosine quality for multiply assigned documents does not decrease with increasing fuzzifier $f$. The fuzziness behavior does not influence the cluster quality in consequence.

<table>
<thead>
<tr>
<th>Multi assigned Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>0.8</td>
</tr>
</tbody>
</table>

can be seen that the majority of multiple assigned documents lead to new clusters. Thus, the base assumption of the fuzzified agglomerative hierarchical clustering algorithm is proven: It is necessary to investigate all similarity values to take notice of all relevant topics. The effect can also be seen in figure 4.5 in which the crisp clustering creates a singleton cluster for a document (highlighted in read). In the fuzzy graph, this document is
part of a complex graph (also highlighted in red).

<table>
<thead>
<tr>
<th></th>
<th>Number of Clusters for $f$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td>$h = 1$</td>
<td>2980</td>
</tr>
<tr>
<td>$h = 2$</td>
<td>539</td>
</tr>
<tr>
<td>$h = 3$</td>
<td>76</td>
</tr>
<tr>
<td>$h = 4$</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4.9: The average number of clusters per hierarchy level shows that the number of clusters strongly increase with larger fuzzifier $f$. The growth does not increase monotonically as the pruning process deletes different cluster nodes in the cluster graph.

The pruning process does not negatively influence the cluster quality for hierarchy level 1. Comparing the median cluster sizes for all fuzzifiers $f \in \{0.1, 0.2, \ldots, 0.8\}$ and all hierarchy levels $h \in [1; 4]$, it can be seen that the pruning process mainly applies for upper hierarchy levels (table 4.10). This was already theoretically excepted (cf. section 4.3). The Cosine quality remains high for all fuzzifiers $f$ and hierarchy levels $h$ (table 4.11). Thus, it is shown that the pruning process supports the creation of topical clusters.

### 4.4.4 Stability Results

The proposed deterministic hierarchical clustering algorithm has been shown to provide high quality topical clusters. Multiple topics can be flexibly con-
### 4.4. Evaluation

<table>
<thead>
<tr>
<th>h = 1</th>
<th>0.92</th>
<th>0.91</th>
<th>0.91</th>
<th>0.91</th>
<th>0.90</th>
<th>0.90</th>
<th>0.89</th>
<th>0.89</th>
</tr>
</thead>
<tbody>
<tr>
<td>h = 2</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>h = 3</td>
<td>0.98</td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>h = 4</td>
<td>0.98</td>
<td>0.99</td>
<td>0.98</td>
<td>0.96</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 4.11: The median Cosine Quality for different fuzzifiers and hierarchy levels shows that even in case of generated topical groups the overall cluster quality is high. Thus, the clusters represent valid topical groups.

As UGC is a data source with high creation frequency, it has to be further evaluated whether the algorithm provides stable analysis results. This is done by using 9,343 randomly selected user comments from different automotive Internet fora. This data set was iteratively reduced by randomly removing 5% of the documents until 10 different data sets $S_1, \ldots, S_{10}$ are available. The evaluation is performed in the reverse direction to simulate the creation of UGC over time (cf. 4.10).

All 10 data sets have been clustered in crisp mode and additionally in fuzzy mode with fuzzifier $f = 0.5$ for $h = 1$, $f = 0.0$ else. As linkage measure, the Group Average linkage with the Special Subgraph Property was chosen. The pruning process was applied with the maximal hierarchy levels $h_{\text{max}} = 5$ and $h_{\text{max}} = 15$.

![Figure 4.10](image)

Figure 4.10: To measure the cluster stability, a randomly created data set ($S_1$) is iteratively reduced by 5%. The evaluation is performed in the reverse direction to simulate the creation of UGC.
Based on the analysis concept of Ben-Hur et al. [15], the cluster stability is measured by analyzing two related data sets $S_i$ and $S_{i-1}$. Due to the pruning process, the cluster graphs are comparable by analyzing each partition separately. For each partition $P_i \in \{P_{S_i}^h, \ldots, P_{S_i}^{h_{\text{max}}-1}\}$ a similarity matrix $A_i$ is calculated. Its elements are defined as cooccurrence of two documents $d_i$ and $d_j$ within the same cluster in $P_i$:

$$a_{ij} = \begin{cases} 1 & \text{if } d_i, d_j \text{ within the same cluster in } P_i \\ 0 & \text{else} \end{cases}$$  \hspace{1cm} (4.61)

The similarity of two matrices $A_i^{S_i}$ and $A_i^{S_{i-1}}$ is retrieved by calculating the Jaccard Coefficient [113, p. 299]:

$$s(A_i^{S_i}, A_i^{S_{i-1}}) = \frac{N_{11}}{N_{01} + N_{10} + N_{11}}$$  \hspace{1cm} (4.62)

$N_{11}$ denotes the number of document pairs with $a_{ij} = 1$ in both matrices, $N_{01}$ denotes the number of document pairs with $a_{ij} = 0$ in matrix $A_i^{S_i}$ and $a_{ij} = 1$ in matrix $A_i^{S_{i-1}}$ and so on. The similarity measure estimates the stability of two partitions of a given hierarchy level.

Figure 4.11 shows the average similarity values for all hierarchy levels and all 10 clusters. For $h_{\text{max}} = 15$, it can be seen that the cluster graphs are much more stable for lower hierarchy levels $h \in [1; 5]$. In this case, the possibility of multiple assignment clearly outperforms crisp clustering results. Upper hierarchy levels are almost equally stable. For $h_{\text{max}} = 5$, the crisp clustering algorithm slightly outperforms the fuzzy clustering however. This can be explained by analyzing the pruning process: With an increased fuzzifier $f$ more clusters have to be merged to get the same hierarchy level $h_{\text{max}}$. Thus documents that have been previously clustered in partition $P_i$ are more likely clustered in upper partitions $P_{i+1}$ if additional documents are available and with increased fuzzifier. This is why the the stability decreases for higher fuzzifiers and low maximal hierarchy levels $h_{\text{max}}$. In
return, the pruning process ensures that the overall clustering results are very stable independently to $f$.

![Graph](image1.png)

![Graph](image2.png)

Figure 4.11: For $h_{\text{max}} = 15$, the proposed clustering algorithm clearly outperforms crisp clustering results (a) – especially for lower levels in which multiple document assignments are possible. Very small maximal hierarchies ($h_{\text{max}} = 5$) force the pruning process however to merge many clusters so that crisp results are slightly more stable than fuzzy results (b). No matter of $f$, the overall stability is very high.
4.5 Conclusion

In this chapter, an agglomerative hierarchical clustering algorithms was proposed that is able to assign a document to multiple clusters. The design of the clustering algorithm ensures a deterministic and thus reproducible clustering result. Due to its hierarchical creation process, it generates a directed acyclic weighted graph that provides different abstraction levels to the end user. The number of abstraction levels can be flexibly adjusted.

It was shown that the proposed clustering algorithm generalizes well-known agglomerative hierarchical clustering algorithms to fuzzy logic. Well-studied linkage measures can be reused by considering the Special Subgraph Property. It ensures a finite clustering algorithm. The recurrence formulas by Lance and William [100] have been adapted to fuzzy logic and it was proven that the created clusters still induce a monotonic cluster hierarchy. In case of the General Subgraph Property, this only applies to Single Linkage and Complete Linkage.

As agglomerative hierarchical clustering algorithms create a binary cluster graph, it has to be assumed that not every abstraction level represents a topical group. Therefore, a special pruning algorithm was proposed to limit the cluster graph to a predefined hierarchy size. This pruning takes the cluster structure into account and merges binary clusters to topical clusters.

The clustering behavior is investigated by a newly created evaluation technique. This is the first evaluation method that takes multiply labeled documents into account and thus supports the quality analysis of hierarchical fuzzy clusters. The evaluation has shown that a document can be assigned to multiple clusters without influencing the overall cluster quality. Hereby, the degree of multiple assignments can be flexibly controlled with the fuzzifier $f$. The cluster sizes slightly correlate to $f$. With an increasing fuzzifier, the number of clusters strongly grows. Thus, new topic groups become available that have been missed in crisp clustering approaches. The quality of these groups is very high for all hierarchy levels. It can be stated in
4.5. Conclusion

consequence that the overall clustering algorithm supports the analysis of UGC with multiple topics per user comment on different abstraction levels.

The overall clustering results are robust with regard to newly added documents, independent of the applied fuzzifier $f$. A stability evaluation has shown that fuzzy clustering could clearly outperform crisp clustering results due to the possibility of multiple assignments. However, it has to be stated that the pruning process could negatively influence the cluster stability due to more cluster nodes which have to be merged in a fuzzy cluster graph. On the other hand, it provides the possibility to compare hierarchical clusters as it ensures to have an equal number of partitions.

The proposed clustering algorithm offers unique characteristics. It is the first algorithm that integrates fuzzy logic to deterministic agglomerative hierarchical clustering algorithms. Moreover, it creates topical clusters using the overall cluster graph structure. An analysis protagonist can use the approach to investigate UGC on different a-priori undefined abstraction levels. The only required parameters are – next to the linkage measures – the desired fuzziness degree and the number of abstraction levels. These parameters depend on the user requirements. The analysis results are reproducible and stable with regard to newly added user comments.
A Social Media Monitoring System for Quality Analyses

The multi-agent based crawler architecture (chapter 3) and the fuzzified hierarchical clustering algorithm (chapter 4) are focusing two special aspects concerning UGC. They have to be combined with further analysis algorithms to provide an usable Social Media monitoring system. This chapter is dealing with the complete realization of such a system. It therefore proposes the interactive analysis system AIM (Automotive Internet Mining) to extract quality related information from UGC. This system has to fulfill different requirements:

**Modularity:** Social Media analysis is currently a strongly discussed research area in which new approaches and concepts are frequently published. Especially methods for topic analyses and the extraction of correlated sentiments are evolving very fast. An appropriate analysis architecture should be able to benefit from these improvements without the need to reimplement the entire architecture for each analysis improvement or extension.

**Scalability:** UGC is a very fast growing data source. An analysis architecture has to scale with the monitored data, as the number of analyzed
documents should be at least as large as the number of newly created user comments. However, this is not sufficient because the improvement of different analysis processes requires a complete reprocessing to provide consistent analysis results. Thus, an appropriate scalability is a fundamental need for a Social Media monitoring architecture.

Scalability not only concerns the analysis algorithms. It is also related to the analysis protagonists. They have to track many different data sources, e.g. survey results, hardware log statistics, etc.. A Social Media monitoring system will only be used if the quality insights can be gained with minimal time efforts. It is impossible for an analysis protagonist to read all user comments manually. Therefore, the Social Media analysis system has to provide aggregation mechanisms which summarize the extracted information on higher abstraction levels.

**Flexibility** There are different use cases in which AIM is used to gather deeper insight into UGC. They can be distinguished in three abstract analysis scenarios:

1. First of all, AIM should provide quality related information for well-specified *a-priori defined topics*. This includes the possibility to specify disambiguation information so that the meaning of different terms can be considered\(^1\). These topics are regularly analyzed. The provided information has to have a high reliability so this measure can be used as a KPI\(^2\).

2. The second use case concerns *ad-hoc analyses*. This is an important aspect in AIM because ad-hoc analyses are currently only possible by expensive customer surveys. An analysis protagonist should only need to specify the expected terms of a desired topic without the need to train the system for a long time. Fast analysis responses may support the decision-making whether further

---

\(^1\) This is for example necessary to disambiguate the term *A8* which could represent a car model or a German autobahn.

\(^2\) Key Performance Indicator; A KPI is typically used to measure the success of a particular activity.
investigations are necessary and possible. In this scenario, the main focus is on interactive usage and different deep-dive analysis possibilities to get a deeper impression of discussed issues.

3. Finally, **blind spots** are a critical analysis issue. Detecting a-priori undefined topics supports the analysis protagonists in learning the customers’ definition of quality and thus could influence further ad-hoc and regular analyses.

An appropriate analysis architecture has to support all three requirements simultaneously. For example it is necessary to analyze predefined topics and to apply a blind-spot analysis at the same time to take care for previously unknown problems. An analysis protagonist needs the possibility to freely select the topics of interest in an interactive way because of different responsibilities. Hereby, the topics have to be provided on different abstraction levels as a quality protagonist is typically interested in a more abstract analysis than a quality engineer who additionally needs the original user comments to learn the problem circumstances.

By investigating related work (section 5.1), it is shown that there are already analysis systems available that try to extract quality related information from UGC. The analysis possibilities are however limited and the analysis results are less reliable. Therefore, the new Social Media monitoring system AIM (Automotive Internet Mining) is created. It first applies a Social Media specific pre-processing pipeline to structure the a-priori unstructured data (section 5.2). Hereby, the pipeline applies several state-of-the art algorithms. For its realization, different analysis frameworks have been analyzed to find the most appropriate processing architecture. The resulting survey has been published in [12]. It states UIMA as the most evolved analysis architecture at least in terms of infrastructural capabilities. The structured information is used in the second step to provide a fast and reliable quality  

---

3 An analysis protagonist is typically specialized to a specific product (e.g. “W212”) or / and product feature (e.g. “entertainment system”).
insight into an arbitrarily large amount of data. Hereby, three new quality indices are proposed (section 5.3):

- The Market Satisfaction Index provides the possibility to compare several product features among different products or manufacturers. It is usable for classical SWOT analyses to identify strengths, weaknesses, opportunities and threads.

- The Product Satisfaction Index extracts the advantages and disadvantages of a product. Similar to the approach of Kano et al. [89], it provides the information which feature is worth further improvements.

- The newly introduced Relevancy Index provides a robust marketing and market penetration independent importance information. It can be used to weight findings according to their relevance. Furthermore, it identifies anomalies with unexpected high discussion rates.

The complete analysis system is presented in section 5.4. Its application and the evaluation of the analysis results is described in section 5.5. Hereby, the AIM indices are compared with state-of-the art quality indices in the automotive domain, namely the J.D. Power IQS index. Additionally, some real world use cases are presented which prove the advantage of the AIM system with regard to classical quality sensors. The concepts have been presented in [8] and [9].

5.1 Related Work

The idea of gathering quality related information from UGC is not new. As it was discussed in section 2.3, many different e-commerce systems are already providing customer reviews that are supposed to provide reliable quality insights. Although this type of recommendation system is available

---

4 J.D. Power and Associates is the leading market research company for customer satisfaction and product quality in the automotive domain. [http://www.jdpower.com/](http://www.jdpower.com/)
in all major shopping portals and although the recommendation system is broadly accepted by customers, it is known today that this type of UGC is not reliable at all due to abuse. Neither the user reviews themselves nor their corresponding helpfulness votes are objective. Nevertheless the data is used in many publications as training data. Kim et al. [91] proposed for example a supervised classification system based on available helpfulness votes to automatically rank user reviews. Zhang and Varadarajan [176] suggested to use a regression based algorithm. The published results are in consequence not reliable. This also applies to other research areas using this data source, e.g. to generate sentiment lexica (cf. [129]). An interesting contrast was presented by Liu et al. [110]. They propose an algorithm to automatically classify helpful reviews to summarize reviews for potential customers. Instead of using the biased helpfulness-votes, they create their own training corpus by defining specific rules whether a review is helpful or not. In their work, they notice a relevant detail for this work: On feature level there are more positive expressions. The reason is not analyzed.

Especially newer publications handle the huge amount of data by structuring product reviews according to mentioned product features. Scaffidi et al. [143] presented the system “Red Opal” that extracts product features and assign the overall rating to each feature. They assume that all features in a review share this rating. This assumption was proven to be invalid by Kano et al. [89] many years before. They have shown that it is possible to satisfy customers although some nice-to-have features are not functional. In addition, the above discussed rating problems are not considered. Guo et al. [69] presented a more sophisticated system that can be applied on semi-structured data in which positive and negative features are mentioned separately. This semi-structured data is generally not available in UGC from weblogs and Internet fora. The most interesting paper for this work was presented by Liu et al. [108]: Their system “Opinion Observer” extracts product features including available sentiment information. They limit their analysis on unstructured product reviews as this kind of reviews generally provides linguistically correct sentences. Each feature is
extracted using matching algorithms based on either explicit features that are mentioned in the text or implicit features which have to be extracted using the context. Thus headed and unheaded contexts can be analyzed. For each feature, different rules based on word 3-grams are applied to identify relevant phrases. Using a sentiment analysis algorithm, the polarity of each phrase is accumulated to present the number of positive and negative connotated phrases. The user is able to analyze the abstract overview in detail by looking at these phrases instead of the complete review and thus to handle the huge amount of data. It is not possible to read the complete comment which might be a problem for a quality analyst. Liu et al. provide the possibility to normalize the extracted information according to the “maximal number of opinions on any feature of any product” [108]. This causes however that more frequently discussed topics receive higher polarity scales. A similar approach was published by Popescu and Etzioni [133]. Instead of extracting feature-sentiment relations using a \( k \)-word window, they use a parser algorithm to take the linguistic sentence structure into account, assuming that product reviews can be parsed with classical parser algorithms.

Previously mentioned methods do not take time into account, so that it is not possible to identify quality problems over time: The review based recommendation systems assume that there are no quality changes. This is surely not correct for products which are sold for a long time like cars or hotels.

5.2 Pre-Processing Workflow

Before any knowledge can be extracted from the large amount of UGC, it is necessary to structure the a-priori unstructured data. This is done using a multi-stage pre-processing workflow that applies different natural language processing (NLP) algorithms (cf. fig. 5.1). An important requirement for this workflow is to be prepared for future research and development achievements, especially in the fields of topic and sentiment analysis. It should be
5.2. Pre-Processing Workflow

possible to replace any algorithm with more sophisticated versions as each individual method is just a means to an end. In consequence, this section is not meant to provide an overview about all possible algorithms. This is already done by different well structured surveys (e.g. [113, 129]). Instead, the following section is limited to algorithms applied in the presented pre-processing workflow. It presents the minimal requirements to structure the a-priori unstructured data for a sophisticated and reliable quality analysis.

Figure 5.1: In a pre-processing pipeline, different state-of-the-art algorithms are applied to extract relevant information from textual data. This information is stored as so-called stand-off annotations (red) including the extracted information, an annotation type and the start and end position of the annotation. These annotations are later used for fast interactive analyses.

Analysis Architecture

There are different analysis architectures available that try to reduce re-implementation efforts while supporting code exchangeability and modularity. Most known free architectures are Ellogon [131], Gate [11], Heart of Gold [146] and UIMA [55] with UIMA being the first OASIS Standard for Unstructured Information Management Architectures [99]. All of them provide a more or less standardized workflow management architecture that enriches a subject of analysis with additional information. The architectures have been analyzed in [12] according to infrastructural capabilities and requirements of the Language Engineering task. It is stated that UIMA is “the most evolved and comprehensive architecture available” [12]. The OASIS standard provides an impressive capability to modularize the overall analy-
sis process with very flexible workflow configurations, including parallelization and distribution. In consequence UIMA was selected as the underlying base architecture. It was extended to support cascaded workflows\(^5\) and an improved analysis aware resource management system. The implementation details are not relevant for the scientific discussion presented in this work and can be found in \[169\].

**Language Detection**

A fundamental step in many text analysis systems is the language identification, as most algorithms highly depend on the underlying language. An overview about available algorithms is presented in \[147, section 6.1\]. In the proposed workflow, a character \(n\)-gram based algorithm is applied as proposed by \[37\]. It uses a training set, for which the algorithm creates a character \(n\)-gram profile \(P_l\) for each available language \(l\). Each profile includes relative probabilities for different \(n\)-gram sizes \(n = n_1, \ldots, n_l\). In the same way, a new text \(t\) is processed to generate a profile \(P_t\). To assign the proper language, the similarity between the training set based profiles and the test profile has to be calculated. According to Cavnar and Trenkle, this is done using an ad-hoc rank ordering statistic to be insensitive to omissions.

\[
S_{\text{Rank}}(P_l, P_t) = \sum_{n_i \in P_t} |\text{Rank}_{P_l}(n_i) - \text{Rank}_{P_t}(n_i)| \tag{5.1}
\]

As implementation, the open source library NGramJ\(^6\) is used.

There are also more sophisticated algorithms that utilize word positions. A well-known algorithm was presented by Dunning \[53\] who uses Markov Models for every language in the training data. It is however necessary to mention that these approaches are difficult to apply in UGC as the sentence

\(^5\) A cascaded workflow applies document and corpus centered analyses in a combined way. Among other things, this is used to generate global lexica and applies different corpus based analyses.

structure and the word positions do not follow common grammatical rules. This was already shown by Schierle [147]. It is also known that especially for very short texts Naive Bayes classifiers outperform other methods [68]. As the analyzed text in this work is limited to user comments out of weblogs and Internet fora, it is ensured however that the textual data is large enough. Thus, the approach by Cavnar and Trenkle is well-suited for the task of language identification.

Tokenizer

Tokenization is the process to separate a text into its smallest meaningful units, so-called tokens. According to Manning and Schütze [113, p. 124], these tokens typically represent words, numbers and punctuations. Although this step seems to be very easy to realize, there is a wide list of available algorithms with high influence on following analysis algorithms. An overview about state of the art methods is described in [147, section 6.2]. In this work, a regular expression based approach is used. It extends the classical token types to URLs, mail addresses, nicknames (denoted by @Nickname), smilies, money expressions, numbers and dates.

Sentence Detection

The sentence detection is realized using a baseline algorithm. Based on the analysis results of the tokenizer, punctuation annotations including comma, colon, semicolon and line breaks are used to specify the sentence segmentation. This baseline is possible as punctuation in numbers are already handled by the tokenizer. The algorithm is extended using manually specified abbreviation lists to increase accuracy. There is no sophisticated algorithm (e.g. [93]) used due to special linguistic conditions in UGC for which no training data is available. Especially line breaks without any punctuation have to be considered, e.g. due to manually defined lists, tables structured, etc.. The applied algorithm takes care of these cases.
Topic Detection

According to the pre-defined requirements, the topic detection has to be applicable in three different user scenarios, namely regularly applied high-quality KPI analyses, fast ad-hoc analyses and blind-spot analyses. Each scenario enforces a different analysis algorithm. Ad-hoc analyses for example cannot be applied in the pre-processing phase as they require an interactive analysis system in which each topic of interest has to be specified without long training phases to get fast analysis results. KPI analyses on the other side are based on an a-priori specified topic hierarchy. Each topic has to be detected with high quality. Blind-spot analyses should provide the topic hierarchy on their own. The hierarchy does not need to have any topic in common with a-priori specified hierarchies. In the pre-processing phase, only the KPI analysis and blind-spot analysis scenario can be covered.

**Blind-Sport analyses**  In a blind-spot analysis, the analysis protagonist needs to investigate a large set of documents to ensure that there is no missed topic. The large amount of data makes it impossible to read all documents manually. Therefore, an analysis algorithm is necessary. The already presented fuzzified hierarchical clustering approach (chapter 4) supports this analysis scenario. It requires no further data than the similarity matrix. In especially, the clustering technique does not require any topical definition. It is thus suitable for blind-spot analyses. The algorithm generates deterministic and stable clustering results. Regularly applied analyses are thus comparable over time. The protagonist can investigate the generated topical groups on different abstraction levels. The possibility to assign one document to multiple clusters ensures that no relevant topic may be missed.

The application of the blind-spot analysis is not limited to the pre-processing phase. It is an interactive analysis scenario. Typically, the user specifies a data set of interest on which a blind-spot analysis should be applied. This data set can be manually selected or it can be specified using some kind of data filtering, e.g. all documents dealing with a specific product or from
5.2. Pre-Processing Workflow

a specific data source. While the first data set cannot be pre-processed, the second data set can be regularly analyzed. This can be done in the pre-processing phase. The filter requires some kind of data labeling. This is realized by supervised topic detection algorithm which is also used for KPI analyses.

KPI analyses In the automotive domain many different topics can be predefined. The user comments should be classified for example according to the discussed manufacturers, products, product features and components. These topics are hierarchically structured and thus describe different abstraction levels. Unfortunately, there is no publicly available labeled data source. This is not only due to the automotive data environment. On the contrary, topic detection in UGC out of Internet fora and weblogs has not gained much attention in the scientific world, especially for German corpora. Thus a training corpus is not available although the product and feature taxonomy is well defined. The large list of possible products and product features makes the creation of a training corpus very time consuming. Classical supervised classification algorithms are not applicable in consequence.

Schierle and Trabold [149] proposed an alternative method that can be used in this situation: They presented a taxonomy-based multilingual context aware classification algorithm. Each taxonomy element defines a concept, which is represented by a set of synonyms for each language. Each synonym can be extended by Part Of Speech (POS) tags or other contextual information in which a given topic – in this case a product or a product feature – is valid (cf. fig. 5.2). The approach is very practical because it is possible to define a taxonomy quite fast to generate first analysis results in very short time while at the same time it is possible to increase precision enriching the taxonomy with additional contextual information for disambiguation. In addition, the classification method does not require any labeled data but a taxonomy structure which in practice is already available for products and product features, although not used for NLP processes but for marketing issues and competitive intelligence processes.
Figure 5.2: A taxonomy structure allows the interlingual definition of concepts with additional linguistic and environmental definitions like POS tags and contextual information. It can thus be used for high quality supervised topic detection in headed contexts with no limitation to the analyzed language (based on [149]).

The applied approach of Schierle and Trabold is a multi assignment algorithm which means that it is possible to detect more than one topic in one user comment. During the analysis process, the algorithm expands the taxonomy according to defined synonyms and contextual information [147, section 10.6]. Using a trie representation for each language and all synonyms in all synsets with one token per node, a very efficient matching algorithm can be applied using simple string matching methods. It is important to mention however, that this process requires an identical tokenizer behavior for the token separation in the analysis object and in the taxonomy itself. If a sequence of 1 to \( n \) tokens \( t_1, \ldots, t_n \) has been recognized, the associated part of speech tags have to be compared for disambiguation. If further contextual information is available, the left and right side of the matched region is analyzed.

**Sentiment Analysis**

A very important step towards quality related information extraction is known as sentiment analysis. Today, there are many different concepts in this extremely active research area. A very interesting survey was published by Pang and Lee [129]. In their publication, they state that a “fundamental technology in many current opinion-mining and sentiment-analysis applications is classification” because “many problems of interest can be formulated
5.2. Pre-Processing Workflow

I love the integrated navigation system, but the installed maps are absolutely outdated.

Figure 5.3: In UGC, there are many comments with more than one topic per sentence. Thus, a granular sentiment analysis is necessary.

as applying classification / regression / ranking to a given text unit” [129, section 4]. In the context of this work, the definition of this text unit is the critical process. It has to be assumed that one document, one paragraph and even one sentence can cover more than one relevant topic for which the corresponding sentiment expressions are of interest. A real world example is presented in figure 5.3 in which two different components with different moods are mentioned. This is not only typical for product features but also for different products and manufacturers that may be compared in one user post. A granular analysis is absolutely necessary. It is however not possible to pre-define a phrase, a sentence or any other unit as the appropriate text unit. It depends on the user comment.

Pang and Lee noticed that work “in polarity classification often assumes the incoming documents to be opinionated” [129, section 4.1.2]. In consequence, many publications support the idea of using subjectivity analysis to improve the sentiment analysis (e.g. [118]). Especially for quality related text analysis, this is however not wanted. There are of course sentences that might be classified as subjective, e.g. “The car needs too much gas.”, but there are also objective sentences absolutely relevant for a quality related sentiment analysis, e.g. “There is a lot of rust at the front door.”. Limiting sentiment analysis to opinionated sentences would loose a lot of relevant data. Objective expressions also carry some relevant mood information. Thus only the intended polarity is of interest.

Most approaches – surveyed already in [129] – are not appropriate to solve the given analysis problem: They do not bother to detect multiple topics with different moods within one document or even one sentence. Addition-
ally the classification based approaches require a labeled data set that is hard to obtain. A different and more usable technique was presented by Remus [136]. He proposes a linguistically motivated phrase-level polarity analysis. Instead of relying on labeled training corpora for classification issues, he presents a rule-based mechanism using a sentiment lexicon called SentiWS\(^7\). This German-language lexicon, published in [137], consists of words, possible inflections (if known) and correlated weights. In addition, POS tags are provided for basic disambiguation as discussed in [129] section 4.2.3. The weights within the interval \([-1; 1]\) denote an a priori sentiment value that is implied by a word \(w\) without considering any contextual information. The weight is calculated using the Pointwise Mutual Information (PMI), suggested by [41]. The concept was also used by [160] and [161] to determine the semantic orientation (SO) and the strength of its associations (A). Using a manually defined seed list of positive and negative connotated terms \(P\) and \(N\), the a-priori sentiment value can be calculated by:

\[
\text{SO-A}(w) = \sum_{p \in P} A(w, p) - \sum_{n \in N} A(w, n)
\]

(5.2)

Based on this lexicon, Remus proposes a two-step phrase analysis algorithm [136]:

1. **Word-Level Polarity Analysis**: For each token, the first analysis phase looks up the connotated sentiment value in the SentiWS lexicon. If the token is not found in the lexicon, no sentiment value is assigned. The lookup process incorporates POS tags as a basic disambiguation method. In addition, the POS structure is mapped to an individual category system specifying polar and modifying categories, used in the following phrase-level analysis phase (table 5.1). This mapping is done using POS information and manually specified lists.

\(^7\) SentiWS is publicly available at [http://asv.informatik.uni-leipzig.de/download/sentiws.html](http://asv.informatik.uni-leipzig.de/download/sentiws.html)
5.2. Pre-Processing Workflow

<table>
<thead>
<tr>
<th>Abbr.</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>Adjectives, adverbs</td>
</tr>
<tr>
<td>N</td>
<td>Nouns</td>
</tr>
<tr>
<td>V</td>
<td>Verbs</td>
</tr>
<tr>
<td>NEG</td>
<td>Negations</td>
</tr>
<tr>
<td>INC</td>
<td>Strengthening Intensifiers</td>
</tr>
<tr>
<td>DEC</td>
<td>Weakening Intensifiers</td>
</tr>
</tbody>
</table>

Table 5.1: Each polarity connoted token is assigned according to a special category system that is used in the phrase-level analysis phase.

2. Phrase-Level Polarity Analysis: The phrase level analysis iterates over all annotated sentences and within these sentences over all polar words detected in the word-level polarity analysis. Applying a manually created list of rules, each phrase is annotated with a corresponding phrase sentiment. Each rule $r$ is defined as:

$$r = [(d, f, p)CAT_i \ldots CAT_j]$$

(5.3)

with $CAT_k \in \{ADJ, N, V, NEG, INC, DEC\}$ or a rule itself. The marker “…” denotes discontinuity, $d \in \{\rightarrow, \leftarrow\}$ specifies a direction, $f \in \{a_+, a_*\}$ represents an aggregation function and $p \in Q$ a parameter for $a_*$. The (nested) rules allow the complex definition of phrase patterns, in which negation and other complex dependencies can be specified. More details on the rule definition can be found in the original paper [136].

The results of Remus are based on the application of further pre-processing algorithms, namely a POS tagger and a parser. This work does not apply any of these algorithms. This is due to two special aspects. First of all, UGC is not as well structured as news articles which is why POS taggers and parsing algorithms perform worse (cf. [11]). Additionally, the performance of these algorithms is not as good as necessary to analyze the extremely...
large amount of data in appropriate time\textsuperscript{8}. That is why the original concept of Remus was slightly modified to use simple pattern rules in combination with the most often associated POS tag, which is stored in a manual list. It is clear that a sophisticated POS analysis and the application of a parser could improve the analysis results significantly. As both research areas are however not focus of this work, no research was done in how to optimize currently available approaches.

**Relation Extraction**

According to Pang and Lee an important factor in sentiment analysis is the topic consideration. A common approach is to limit the analyzed documents according to a user-specified topic [129, section 4.4.2]. This is however not possible in this work as many topics and many products are discussed in Internet fora and weblogs. The data should be analyzed instead on different abstraction levels according to component and product hierarchies for which relational information has to be extracted. A sophisticated relation extraction algorithm (e.g. [72]) is not applicable due to the missing parsing information. This is why a very basic heuristic relation extraction approach was used that considers the direct context of a topic and a sentiment phrase: A component is related to a sentiment phrase, if and only if there is a sentiment annotation in a word window of \( n \) tokens. It has been experimentally determined that \( n = 7 \) produces the best results.

The same approach was applied to determine the indented sentiment of products. These are however not often mentioned in user comments from weblogs or Internet fora as the product is already identified in additional meta information, e.g. the forum category. In the following only the relation of components and sentiment phrases is used in consequence.

\textsuperscript{8} In a test implementation different state-of-the art POS taggers and parsers needed on average one second per document. In the same time, many new user comments have been created. An application of POS taggers and parsers would enforce an expensive hardware cluster to process the large amount of data.
5.3 Quality Indices

Quality insights are gained today by several different data sources, e.g. internal and external survey results, hardware logging reports and press releases. Due to limited time, an analysis protagonist requires a Social Media monitoring system to provide fast quality insights. It is impossible to read many user comments. Therefore, textually described quality aspects are summarized to three new quality indices: the relevance index, the market satisfaction index and the product satisfaction index. Each index can be provided on different abstraction levels.

Kano et al. have shown that different product features fulfill specific needs, which might be dependent of the customers themselves. According to these findings it has to be assumed that each product and product feature is discussed in different moods and different frequencies. A pre-analysis of many thousand comments in UGC has additionally shown that many comments discuss more than one product feature. This was also shown in related work dealing with product reviews (e.g. ). Thus, the quality indices cannot be generated on top of complete user comments. Instead, they are calculated on individual topics.

Each index is based on two different dimensions: an analysis dimension and a reference dimension. Each dimension represents a topic, e.g. a manufacturer, product, product feature or component. By combining two different dimensions, the quality impact relatively to the reference dimension can be calculated. The actual implementations and the advantages of this method will be discussed in detail in the following sections. For simplicity, a topic will be defined as a product or a product feature. It is of course also possible to use any other topic as reference or analysis dimension. Thus the indices are also applicable to topics which have been identified in the clustering process.

---

9 e.g. the ADAC and TÜV quality reports.
10 The dimensions thus depend on the defined topics and can be easily extended.
5.3.1 Relevance Analysis

A very important impact for any decision is the relevancy of a discussed topic. Recent work has used absolute counts to provide this information (cf. [108]). This is however no appropriate measure. The manual analysis of user comments has shown that it is important to be aware of different frequencies of occurrence for different products: On the one side product A may be more frequently mentioned than product B because A is a mainstream product. On the other side a non-mainstream product C can be mentioned even more often because the manufacturer was able to motivate users of Social Media systems to talk about its products (e.g. Apple). The absolute counts are thus dependent to different market segmentations and the costumers’ attitude towards active participation in the Internet. A quality analysis system has to take both facts into account to provide reliable relevance information in order to support decision making. This is done by providing the relevance index.

Let $y_i$ be a given product feature, $x_j$ the corresponding product. The probability $p(y_i|x_j)$ is defined as the ratio on the frequency $f$ of the common occurrence of product $x_j$ and feature $y_i$ relative to the product itself:

$$p(y_i|x_j) = \frac{f(y_i,x_j)}{f(x_j)}$$

(5.4)

The formula measures the probability that users publish comments about a feature $y_i$ while dealing with product $x_j$. Thus, it provides an insight about how relevant the feature $y_i$ is depending on product $x_j$. It is not necessary to care about different product frequencies anymore. Therefore, the probability $p(y_i|x_j)$ is defined as relevance index. The normalization method takes different product distributions into account and ensures that products can be compared on feature level. Furthermore, it provides the possibility to compare all product features and thus to estimate the extent to which a feature is relevant for the customer. This result is related to the original approach of Kano et al. [89]: Although it is not possible to categorize each product feature into different quality levels as the corresponding
intervals are not known, the method provides the information about the feature attraction by comparing each feature to another.

The formula 5.4 defines a product \( x_j \) as reference dimension while product feature \( y_i \) represents an analysis dimension. It is of course also possible to focus on a specified product feature:

\[
p(x_j|y_i) = \frac{f(y_i, x_j)}{f(y_i)}
\]  

The formula 5.5 measures the probability of finding product \( x_j \) while focusing on a product feature \( y_i \). The normalization method neglects the frequency a feature is discussed and provides the relevancy for different products. In contrast to the relevance calculation of equation 5.4, the discussion frequency for a product or manufacturer is not considered. The measure is thus more appropriate for marketing issues in which the analysis agent wants to know whether a feature is discussed more often for a given product.

The relevancy information may be difficult to interpret if it is calculated for one product and one product feature only. Depending on the analyzed topic, the value can be quite small. A must-have product feature for example should have a smaller relevance value as it is not assumed to be worthy of discussion. The expected relevance interval is a-priori not known however, unless a Kano analysis would be applied (which is quite difficult, as discussed in section 2.1.2). By comparing different products and product features, this is not necessary at all. Analyzing a product-feature combination relative to relevant competitors makes the analysis interpretable. A deviation is worth further deep dive analyses.

### 5.3.2 Satisfaction Analysis

The sentiment analysis is thought to provide a valuable insight into customers’ satisfaction. Related work use the extracted mood directly by providing the absolute number of positive and negative comments (e.g. [108]).
This behavior is also available in many commercial Social Media analysis products. It is based on the intuitive assumption that customers’ satisfaction is the ratio of the number of positive to negative comments. The mathematical representation for the customer satisfaction $s$ can thus be defined like this: Set $x_i$ as a product and $y_j$ as a product feature. The frequency of a product and a product feature is denoted by $f(x_i, y_j)$. The number of positive comments concerning this combination is defined by its frequency $f^+(x_i, y_j)$, the number of negative ones by $f^-(x_i, y_j)$. The satisfaction of the product-feature combination $s(x_i, y_j)$ is thus based on individual probabilities $p$:

$$s(x_i, y_j) = \frac{p(x_i, y_j^+ | x_i, y_j)}{p(x_i, y_j^- | x_i, y_j)}$$

$$= \frac{f^+(x_i, y_j)}{f^-(x_i, y_j)}$$

$$= \frac{f^+(x_i, y_j)}{f^-(x_i, y_j)}$$

(5.6)

Equation 5.6 assumes to have a balanced lexicon and a balanced language. Both conditions are very unlikely. In contrast, an analysis of the extracted moods has shown that there are more positive comments than negative ones. This result was also seen by other researchers like Liu et al. [110] and Remus et al. [139]. Using the absolute number of positive and negative comments is thus not valuable as quality measure. The analysis system “Opinion Observer” [108] considers this fact by providing a visualization that compares different products or manufacturers. It is however to the analysis protagonist to ensure valid comparison values. A reliable index is not proposed.

There is however another important problem with this measure: Formula 5.6 assumes that sentiment analysis is a straight forward data analysis. Positively connotated comments are supposed to be positive and negatively connotated comments are supposed to be negative. Related work dealing
with the labeling of sentiment based data mentions however a typical Kappa agreement [42] between 0.4 and 0.6 (cf. [138]). It is known today that the sentiment rating is a highly subjective categorization that not only depends on the analyzed genre and domain (cf. [132]) but also on the individual analysis agent. Especially the distinction of negative and neutral or neutral and positive terms depends on the users themselves. The formula is not robust towards changes in the sentiment lexica.

Last but not least, the formula does not care about different product or feature penetrations and is thus not comparable among different products and product features.

**Market Analysis**

Similar to related work, the proposed satisfaction index is based on the probability that a product feature \( y_j \) is mentioned positively or negatively. The analysis dimension only considers documents in which product \( x_i \) and product feature \( y_j \) are mentioned simultaneously:

\[
p(x_i y_j^+ | x_i y_j) = \frac{f^+(x_i, y_j)}{f(x_i, y_j)}
\]
\[
p(x_i y_j^- | x_i y_j) = \frac{f^-(x_i, y_j)}{f(x_i, y_j)}
\]

A robust satisfaction index can be provided by introducing a reference dimension. It measures the probability that a product feature \( y_i \) is mentioned positively or negatively without considering the product:

\[
p(y_j^+ | y_j) = \frac{f^+(y_j)}{f(y_j)}
\]
\[
p(y_j^- | y_j) = \frac{f^-(y_j)}{f(y_j)}
\]

By combining the analysis dimension and the reference dimension, a neutral satisfaction can be defined as:
Chapter 5. A Social Media Monitoring System for Quality Analyses

\begin{equation}
    s(x_i, y_j) = \frac{p(x_i, y_j | x_i, y_j)}{\frac{p(y_j | x_i, y_j)}{p(x_i, y_j)}}
    \frac{f^+(x_i, y_j)}{f^+(y_j)} \frac{f^-(x_i, y_j)}{f^-(y_j)}
\end{equation}

In every day use, equation 5.7 was not practical as the values are in the interval \([0; \infty)\) with neutrality for \(s(x_i, y_j) = 1\). This asymmetric behavior complicates interpretations. Thus, a re-normalization was done:

\begin{align*}
    r^+(x_i, y_j) &= \frac{f^+(x_i, y_j)}{f^+(y_j)} \\
    r^-(x_i, y_j) &= \frac{f^-(x_i, y_j)}{f^-(y_j)} \\
    s(x_i, y_j) &= \begin{cases} 
        \frac{r^+(x_i, y_j)}{r^-(x_i, y_j)} - 1 & \text{if } r^+(x_i, y_j) \leq r^-(x_i, y_j) \\
        1 - \frac{r^-(x_i, y_j)}{r^+(x_i, y_j)} & \text{else} 
    \end{cases}
\end{align*}

A potential customer can compare the satisfaction value of different products \(x_i\) on feature level \(y_j\): A higher satisfaction value \(s \in [-1; 1]\) for product \(x_i\) and feature \(y_j\) implies higher satisfaction compared to other products with feature \(y_j\). Thus, the quality measure supports a market dependent analysis by defining the expected satisfaction over all products. In combination with the relevance index, the market satisfaction index supports classical SWOT analyses [96], in which the strength, weaknesses, opportunities and threats of a product are analyzed for further decisions.

By filtering the data to special manufacturers / products, it is also possible to focus on a specific market segment. This is important as special product features – e.g. driver assistance systems – are not common in all cars. Limiting the analysis to so-called premium manufacturers ensures to have similar expectations among possible customers.

The formula 5.8 is more robust with respect to different sentiment ratings as the typical sentiment assignment is considered as the expected value:
5.3. Quality Indices

The satisfaction increases (decreases) if and only if the number of positive (negative) phrases deviate upward from this value. It thus depends on the overall polarity distribution. In contrast to formula 5.6 a new positive entry in the sentiment lexicon does only influence the satisfaction index if it is more often used for a topic than it is done typically. By considering the number of documents, which are available for a given product feature, it is additionally robust towards different penetrations of discussed features.

Product Analysis

A very impressive feature in the proposed satisfaction analysis is the definition of the reference dimension, which represents the expected satisfaction. In the market scenario, the expected satisfaction is defined by the frequency of positive and negative comments in a predefined market segment (possibly all products). It is of course also possible to define the overall product satisfaction as expected satisfaction:

\[
\begin{align*}
    r^+(y_j, x_i) &= \frac{f^+(x_i, y_j)}{f^+(x_i)} \\
    r^+(y_j, x_i) &= \frac{f^+(x_i, y_j)}{f^-(x_i)} \\
    s(y_j, x_i) &= \begin{cases} 
    \frac{r^+(y_j, x_i)}{r^-(y_j, x_i)} - 1 & \text{if } r^+(y_j, x_i) \leq r^-(y_j, x_i) \\
    1 - \frac{r^-(y_j, x_i)}{r^+(y_j, x_i)} & \text{else}
    \end{cases}
\end{align*}
\]

In equation 5.9 the reference dimension and the analysis dimension are simply exchanged. The created index measures therefore whether a feature achieves higher satisfaction results than other features. This makes it possible to identify features in need of improvement from the customers’ point of view.

Identically to the market satisfaction index, the product satisfaction index only depends on the overall polarity distribution which makes it more robust towards changes in the sentiment lexica. As the number of documents for
Chapter 5. A Social Media Monitoring System for Quality Analyses

When a given product is considered, the effects of different market penetrations are neglected and the index is comparable among different products.

5.4 AIM Architecture

The Social Media monitoring system AIM is realized as a three-tier architecture (cf. fig. 5.4). This involves three separate layers for the user interface, the business logic and the data handling. Three-tier architectures are known to be very scalable, to provide an exceptional security and a fast execution [54]. Each layer is independent to each other, therefore the analysis algorithms can be exchanged with more sophisticated ones without influencing the whole analysis system. In the following, each layer will be discussed separately in detail.

5.4.1 Data Backend

The data backend is responsible to provide the required data which is used in the analysis system. Therefore, it first applies the multi-agent based crawler architecture described in chapter 3. In context of this thesis, the crawler tracks 20 manually selected automotive Internet fora and 103 weblogs. In total, the crawler agents have downloaded and extracted 23.3 million German and English user comments. These comments are grouped to 2.4 million discussions.

Each discussion is pre-processed using the processing workflow described in section 5.2. This pre-processing workflow supports the detection of topics by applying a taxonomy based classification approach [149]. The technique enforces some manual efforts to specify a topic hierarchy and to extend each topic with synonyms and contextual information. In return, it is not necessary to provide a manually labeled data set. The quality of the analysis results correlates with the manual work and can therefore scale according to the users’ needs. In the automotive domain, there are already very complex product and concept taxonomies available, which can be directly included into the topic detection approach. The applied taxonomy contains 2,081
5.4. AIM Architecture

Figure 5.4: The Social Media monitoring system AIM is realized as a three-tier software architecture. In the first layer, the presented multi-agent based crawler architecture (chapter 3) collects relevant user comments that are processed and stored for further investigations. Different analysis algorithms are applied on the data according to the business logic which is implemented in the analysis backend. The third layer realizes a Rich Internet Application (RIA) in which an analysis protagonist can interactively investigate the user comments on different abstraction levels.

As described above, UGC is processed on the discussion level. However, a discussion is not the appropriate analysis level to extract quality related automotive-related multilingual concepts (e.g. components, service terms) with 5,392 synonyms. POS tags have not been used as there is no efficient POS tagger for Internet fora available. The sentiment detection is based on the publicly available SentiWS lexicon [137]. It was extended by English translations.
information as several users can describe the same problem. Furthermore, topics may change in a discussion. Therefore, comments should be analyzed separately to provide clear analysis results. This is not possible during the pre-processing phase because relevant topical information (e.g. the problem itself) may only be mentioned in earlier comments. This dilemma is solved by annotating the user comments within the discussion. The discussion structure is used after the pre-processing to store the detected topics and related sentiment phrases per user comment. Furthermore, several sentiment phrases within a user comment are aggregated to one comment related sentiment score. As there might be positive and negative phrases simultaneously, each user comment can be annotated positively, negatively, neutrally or positive-negatively.

The processed comments are stored to a Lucene\textsuperscript{11} data index. It is used as central data repository for following analysis algorithms. The Lucene search engine provides the possibility for very fast data filtering. It enables an efficient calculation of the number of documents for given topic and related sentiment scores in the analysis backend. Each Lucene document represents one user comment. It contains the textual data, the discussion position, the extracted KPI topics, related sentiment scores and the aggregated comment sentiment score.

The Lucene search engine is not limited to data filtering. It also provides advanced full-text search capabilities. AIM uses this technique to support fast ad-hoc analyses. The analysis protagonist only has to specify the expected search terms to define a topic. It is not necessary to train any classifier or to provide complex contextual information. The applied analyses can be repeated enabling a recurring analysis. It is however important to mention that the results are less reliable because ambiguous words are not disambiguated\textsuperscript{12}.

Finally, the data backend provides the topical clusters which have been gen-

\textsuperscript{11} Lucene is an OpenSource search engine, published by the Apache Group \url{http://lucene.apache.org}
\textsuperscript{12} There are many ambiguous words available: e.g. A8 (car vs. autobahn), Smart (car vs. standard English word), etc..
erated using the proposed fuzzified hierarchical clustering algorithm (chapter 4). It supports the analysis protagonist to detect possible blind-spots. The documents are retrieved by the Lucene data index which may be filtered by the analyst.  

5.4.2 Analysis Backend

The analysis backend is an intermediate architecture layer providing an abstract interface to the data backend. In addition, it implements data-intensive analysis algorithms to allow interactive and fast analyses on the frontend.

**Topic Selection**  The *Topic Selection* is a special wrapper that unifies the results of ad-hoc analyses, KPI analyses and blind-spot analyses. This is possible because each analysis provides a set of user comments. The wrapper ensures that further algorithms do not have to handle different topic types on their own. The proposed quality indices can thus be simultaneously calculated on search results, classification results and clustering results. This is also true for the difference analysis and co-occurrence analysis.

**Quality Index Calculation**  After the analysis protagonist has specified, which topics are relevant for his analysis, the *Quality Index Calculation* process automatically identifies the required data sets using the previously discussed Topic Selection wrapper. The data sets are used to calculate the relevance, the market satisfaction and the product satisfaction. The user can freely combine different types of topics for the analysis and for the reference dimension. It is for example possible to define a cluster of the blind-spot analysis as analysis dimension while the manufacturer specifies the reference dimension.

---

13 The clustering algorithm should be applied on a filtered data set because of the large amount of data. Theoretically, it is however not limited.
Chapter 5. A Social Media Monitoring System for Quality Analyses

Difference Analysis  To provide a fast insight into a set of user comments, a difference analysis is applied. Hereby the term distribution of an analysis corpus is compared to the term distribution of a reference corpus (cf. [77, chap. 4.2]). This reference corpus is defined as the complete data set. The analysis corpus is the selected topic. It can be realized as a search result, a taxonomy-based classification result or a cluster. The difference analysis provides the terms whose occurrence probability differs most.

Co-occurrence Analysis  The co-occurrence analysis evaluates the significance of association between co-occurring terms (cf. [77, chap. 4.7]). By limiting the analysis to taxonomy-based topics – namely components, symptoms and products – it is possible to provide a fast insight into potential problem relations (cf. [48]).

Annotation Rendering  The above discussed analysis algorithms provide different abstraction techniques to investigate a large number of documents. After identifying a relevant subset, it may be necessary to read the original discussion to identify relevant circumstances and to gather background information. Therefore, each user comment is enriched with stored annotations from the Lucene data index. The skimming of the user comments is thus facilitated by highlighting already extracted information (e.g. the topic, the sentiment scores, etc.).

5.4.3 Analysis Frontend

The analysis system provides an interactive user frontend. It is a Rich Internet Application (RIA) based on GWT\textsuperscript{14}. The frontend is divided into two sections (cf. fig. 5.5): on the left side, all defined topics (taxonomy based topics, search based topics and labeled clusters) are visualized in a tree. They can be used to filter the analysis data set according to the user’s needs. On the right side, the analysis results are presented. Each

\footnote{http://code.google.com/webtoolkit/}
result depends on the current filter set and is immediately updated after the selected filter set is changed.

Figure 5.5: AIM analysis frontend: The analysis frontend is split into two sections. The left section lists all available topics (taxonomy based, search based and clustering based). It is used to filter the current analysis set. The right section displays the current analysis.

**Pivot Analysis**  The center of analysis is provided by a special pivot analysis scheme. In this scheme, the analysis protagonist can compare freely selectable analysis dimensions (e.g. product features) relatively to freely selectable reference dimensions (e.g. products). He is able to drag topics from the topic tree to the appropriate dimension in the pivot analysis. The system automatically calculates all necessary frequencies. It is possible to use taxonomy-based topics, search based topics as well as topical clusters. Each of them can be used in exchange which enables a very flexible analysis.\(^\text{15}\) The hierarchical topic information, defined in the taxonomy and in the clustering results, is further available in the pivot analysis allowing an analysis on different abstraction levels. This is a very important aspect in

\(^{15}\) Sentiment relations are only possible with taxonomy based topics which makes the analysis results more reliable.
Figure 5.6: The pivot analysis investigates an analysis dimension 3 relatively to a reference dimension 1. For each combination, the relevance index 4 and the satisfaction index 6 is calculated. In addition, the absolute frequency of documents, a sentiment distribution 2 and a satisfaction distribution 5 is provided.

AIM because an analysis protagonist is not limited to a specific abstraction level. On the contrary, it is possible to start with a very abstract view and subsequently zoom into the data.

The pivot analysis scheme visualizes all relevant indices to provide a fast data overview (cf. fig. 5.6). Next to absolute frequencies, the relevance index 4 and the satisfaction index 6 are presented. The latter one is classified into five different groups based on empirically determined borders. Each class is represented by one of five different arrows in order to give a quick satisfaction impression. The identification of controversial product features is supported by rendering the sentiment distribution 2 and satisfaction distribution 5. The product and market satisfaction are not displayed simultaneously for clarity. Moreover, the user has to select the normalization method explicitly.

Deep-Dive Analyses  In addition to the hierarchical topic definition, AIM supports further deep-dive analysis algorithms. They provide a more detailed insight into the discussed issues without requiring the analysis protagonist to read any user comment.

- The analyst can apply the proposed fuzzy clustering algorithm to
detect blind-spots in the current filter set.

- He is able to analyze the satisfaction and relevance indices over time (cf. fig. 5.7 a).
- It is possible to execute a difference analysis to get the most significant terms (cf. fig. 5.7 b)
- The analysis protagonist can investigate significantly co-occurring terms to identify relevant relations (cf. fig. 5.7 c)

Figure 5.7: AIM supports several deep-dive analyses to get a better insight into discussed issues without the need to read any user comment. All deep-dive analyses can be performed in any order which makes the Social Media monitoring system very flexible.

All deep-dive analyses can be performed in any order. Additionally, the analysis protagonist can list all original comments to get further background
information. Textual data can be easily skimmed as the manufacturers, products, symptoms and components are highlighted. The analyst can further investigate the sentiment phrases in order to verify the calculated satisfaction index. For each discussion, a link to the original discussion is provided to support potential discussions by engineers.

5.5 Evaluation

The proposed Social Media monitoring system for quality analysis is evaluated in two steps. First, the market satisfaction index is compared to the state-of-the-art quality index in the automotive domain: J.D. Power's I.Q.S. index. J.D. Power is the leading market research company in the automotive domain and their regularly applied quality surveys have a high impact on customers buying behavior. The evaluation shows the applicability of the proposed Social Media based quality index.

The unique advantages of the product satisfaction index is shown in the second step. Therefore, real world use cases proof the requirement to include the index in daily quality analyses. The relevance index provides the possibility to prioritize improvable product features.

5.5.1 Market Evaluation

The proposed market satisfaction index is evaluated using an established quality measure. As reference, the survey results of J.D. Power’s “Initial Quality Study” IQS have been selected. This study is based on a 228-question battery designed to provide manufacturers with information to facilitate identifying problems. It is an established quality index in the automotive sector. The evaluation is done by comparing 36 different cars of 10 different manufacturers in 2008. For each selected model, there are more than 400 user comments available so that reliable conclusions can be drawn.
The publicly available IQS data\textsuperscript{16} combines three different ratings: the Initial Quality, the Performance & Design and the Predicted Reliability. Each one measures the quality relative to other cars in the market.

The proposed analysis architecture analyzes different hierarchically structured features. Each feature may influence the overall quality, performance, design and reliability. It is a-priori not possible to map these features to the categories provided by J.D. Power as a manual mapping could influence the analysis results. Thus, both measures are compared on the most abstract level. For the IQS study, all three ratings are merged to one global car rating $\in [1; 5]$. A comparable concept in the taxonomic approach is the most abstract topic: \textit{component}, which consists of all other product features.

The calculated correlation coefficient of both measures is 0.46. Thus, there is no perfect match between J.D. Power’s IQS survey and the satisfaction index proposed in this work. Nevertheless there is some correlation. Due to different measure intervals, the significance of correlation can not be calculated directly. Therefore, the proposed satisfaction index $\in [-1; 1]$ has to be rescaled to J.D. Power’s measure interval $\in [1; 5]$.

$$s_2(x_i, y_j) = a \ast s(x_i, y_j) + b$$ \hspace{1cm} (5.10)

A linear regression ensures that both indices are comparable. It states $a = 3.66$ and $b = 3.31$. By scaling the satisfaction index linearly, the correlation is not changed.

As it is assumed that the satisfaction index and J.D. Power’s IQS index both measure the same quality aspects, the $H_0$ hypotheses states that the difference of both indices is 0. Using a t-test\textsuperscript{17}, $H_0 (\mu_0 = 0)$ could not be rejected. Thus, there is no evidence that there is a systematic difference. The t-test is however not able to proof a significant correlation. Therefore, the Type II error has to be minimized. This is achieved by applying the

\textsuperscript{16} For reasons of reproducibility, only publicly available data is used.

\textsuperscript{17} Both measures are normally distributed, which is ensured using the Shapiro-Wilk Normality Test.
Table 5.2: The correlation of J.D. Power’s I.Q.S. index and the satisfaction index is investigated by applying a t-test with different $H_0$ hypotheses. The only not rejectable hypothesis states that there is no systematical difference between both measures. Therefore, the correlation is significant.

<table>
<thead>
<tr>
<th>$H_0$</th>
<th>$t$</th>
<th>two-tailed $p$-value</th>
<th>interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_0 = 0$</td>
<td>-0.015</td>
<td>0.988</td>
<td>not rejectable</td>
</tr>
<tr>
<td>$\mu_0 = 0.25$</td>
<td>-1.833</td>
<td>0.073</td>
<td>rejectable for $\alpha = 10%$</td>
</tr>
<tr>
<td>$\mu_0 = 0.5$</td>
<td>-3.650</td>
<td>0.001</td>
<td>rejectable for $\alpha = 1%$</td>
</tr>
<tr>
<td>$\mu_0 = 1$</td>
<td>-7.284</td>
<td>$2.4 \times 10^{-9}$</td>
<td>rejectable</td>
</tr>
<tr>
<td>$\mu_0 = 2$</td>
<td>-14.554</td>
<td>$2 \times 10^{-19}$</td>
<td>rejectable</td>
</tr>
</tbody>
</table>

t-test with modified $H_0$ hypotheses that state that there is a difference ($\mu > 0$). All hypotheses can be rejected (table 5.2). The satisfaction index thus is significantly correlated to the I.Q.S. results. It is however not possible to model the I.Q.S. results using the satisfaction index. The root mean squared error of both measures is 0.65.

These analysis results were expected. On the one side, AIM was hoped to provide quality related information. The motivation for the entire work is based however on the assumption that there are differences between UGC and the manually executed surveys. Both assumptions have been confirmed. Figure 5.8 shows that the differences are not caused by linguistic reasons (e.g. ambiguities, irony, sarcasm, . . .). Instead, systematic reasons can be found. The most obvious difference is the rating of premium cars. Audi, BMW, Lexus and Mercedes are – except for some outliers – worse rated in UGC than in the J.D. Power study. The reason is quite simple. J.D. Power’s questionnaire is mostly limited to technical questions. Quality is thus defined in terms of technical quality. The customers in the Internet seem to define quality however differently: Especially for premium vehicles, some customer expectations are not fulfilled. The collected unrequested user feedback not only contains information about technical problems but also soft problems (e.g. service quality). These customer expectations are implicitly available in the Internet data which makes the data more valuable compared to J.D. Power’s IQS. A manufacturer gains insights beyond technical quality aspects. Thus, the target of this work is achieved: AIM
supports quality protagonists to learn the customers’ definition of quality which consists of much more than technical aspects.

Figure 5.8: Given the market satisfaction index and J.D. Powers I.Q.S. results for all evaluated 36 car models (x-axis), it can be seen that especially so-called premium models are rated worse in the Internet.

An additional result is given by the products “VW Touareg” and “Volvo V50”. While J.D. Power rates these products quite low, the customers’ satisfaction is very high according to the comments in the Internet. This is caused by a systematical problem in the survey technique, which analyzes the U.S. market. The market has however different expectations to an automobile than the European market. The Internet is in contrast not limited to a given market. Users all over the world can take part in discussions in any language. It is not possible to control this participation. Thus, the results are not very helpful for any market segmentation, but provides a more robust insight into quality expectations.
5.5.2 Product Attractiveness Evaluation

In contrast to the market satisfaction analysis, which was evaluated in the previous section, the product attractiveness normalization should denote features above or below the expected product satisfaction. It is meant to identify features worth further improvements. In the following, some real world analyses are presented that show the necessity of this quality index. Any brand information is removed to avoid any publication restrictions. It should be clear, that the data is not focused to Daimler specific UGC.

In the first analysis, a product is analyzed with regard to engine issues. As the analysis shows, the customers are generally more satisfied concerning the component engine mechanics than it is the case for other products (Fig. 5.9 left side). Calculating the product satisfaction index however, it can be seen that most engine components are disappointing and decrease the overall product satisfaction (Fig. 5.9 right side). This contradiction points to an anomaly which is worth further investigations. Different in-depth analyses support the analysis protagonist to identify the problem circumstances. A time analysis does not provide further information except of a high user participation (Fig. 5.10 a). By using a word cloud (Fig. 5.10 b), relevant internal features and suppliers (here blackened) can be identified. The cloud also lists additional product information (“CGI”) which helps to concretize the problematical product sections. By reading the corresponding user comments (Fig. 5.11), a deeper insight in problem circumstances can be gained to speed up possible problem solutions. It is obvious that this is not limited to own products but also to products from competitors.
5.5. Evaluation

Figure 5.9: Market satisfaction (left) versus product satisfaction (right): Although specific product features are realized above market average, it is possible that customers are dissatisfied. A classical market related analysis (left) could hide these problem areas. A product driven analysis (right) instead exposes these issues.

Figure 5.10: A time analysis shows active user participation during the last weeks (a). A word cloud provides further helpful clues to identify a potential problem (b).
Two weeks ago, I went with my family on holiday in Macedonia. [...] In Croatia, the engine management light suddenly light up [...] at speed of 130 ...

So far, only the injector 3 was broken. It happened on a Saturday afternoon at Lake Constance, 600km away from home ...

Figure 5.11: By reading the original user comments, the analysis protagonist gains helpful circumstance information that may help to solve the problem faster. In this case, all users were on a long journey.

In the second real-world use case, a manufacturer is investigated for anomalies. An unusually high relevance of the interior equipment leads to an in-depth analysis. Therefore, the component hierarchy is investigated in detail. The market satisfaction index shows an above-average realization of the component Information Display while the product satisfaction index indicates customers’ disappointments (Fig. 5.12). According to the time analysis the satisfaction index decreases in colder months (Fig. 5.13 a). The tag cloud mentions many display components (Fig. 5.13 b). By investigating the corresponding user comments it can be seen that the displays suffer from temperature induced electric contact problems (Fig. 5.14).

An interesting side note for both user comments in figure 5.14 is that the comments, published in 2008, describe problems for products with build year 1998. Thus, even long term quality information can be directly investigated. In addition, contributions with the aim of helping people to repair broken components on their own are very valuable as the problem solutions are directly available.

Both real world examples have proven that the product satisfaction index is an absolutely necessary quality measure to identify anomalies and to improve customers’ satisfaction. In combination with the relevance information, it is possible to weight possible product improvements and thus
5.5. Evaluation

Figure 5.12: Also in the second use case, the product satisfaction indicates an anomaly that is worth further investigations.

<table>
<thead>
<tr>
<th>Component</th>
<th>Communication</th>
<th>Information display</th>
<th>Warning and control system</th>
<th>Entertainment</th>
<th>Antenna</th>
</tr>
</thead>
<tbody>
<tr>
<td>(399782)</td>
<td>5.41 %</td>
<td>4.24 %</td>
<td>1.28 %</td>
<td>6.79 %</td>
<td>0.75 %</td>
</tr>
<tr>
<td>(4696)</td>
<td>(9388)</td>
<td>(25976)</td>
<td>(1179)</td>
<td>(8290)</td>
<td>(692)</td>
</tr>
</tbody>
</table>

(b) word cloud

Figure 5.13: A time analysis shows a decreased satisfaction for colder months. The problem seems to be correlated to the temperature (a). A word cloud shows different display components (b).
...in the instrument cluster, the display disappears gradually with increasing temperature or is illegible. If the vehicle is started cold, this problem is initially not available. ...

...a replacement should be of no use because it is most likely the contact wire which is clued with conductive adhesive on the board. The adhesive dissolves over time and the display becomes darker....

Figure 5.14: Reading the original user comments is not only helpful to identify circumstances under which a problem occurs. Furthermore, there is some advice to solve the problem in many comments. This could be used to improve the product, to learn alternative problem solutions and finally to increase the overall service quality.

to prioritize feature improvements according to the customers’ definition of quality.

5.6 Conclusion

In this chapter the interactive Social Media analysis system AIM was proposed to investigate quality aspects provided in UGC. The complete system is based on a three-tier architecture. In the first layer, the system collects, pre-processes and stores UGC. The second layer applies different kinds of analysis algorithms whose results are visualized in the frontend layer. The three-tier architecture ensures a modularized infrastructure. Every analysis and visualization algorithm can easily be exchanged with minimal change efforts. It is scalable as each algorithm can be applied in parallel. Furthermore, each architecture element can be distributed to several machines. This is also true for more complex architecture elements, namely the multi-agent based crawler system and the pre-processing architecture.

An analysis protagonist has to apply different, partially regularly executed
analyses. Therefore, AIM supports different types of topic analyses. Regularly applied *KPI analyses* are realized using a taxonomy-based classification algorithm. It takes contextual and relational information into account without the need for any labeled data. The reliability of the results correlates with the effort that was put in the taxonomy definition. Fast *ad-hoc analyses* are realized using classical full-search capabilities. Finally, *blind-spot analyses* are supported by the newly proposed fuzzified hierarchical clustering algorithm. The Social Media architecture unifies all three kinds of topics with the help of a special wrapper algorithm. Thus, all topic types can be freely combined in every following analysis algorithm. The execution order is not limited.

To handle the large amount of user comments, the analysis system supports the topic definition on different abstraction levels. As the available analysis time is however very limited, further abstraction techniques are necessary. Therefore, three special indices have been introduced to provide a fast quality insight. The Market Satisfaction Index was shown to be significantly correlated with state-of-the-art quality indices. The Product Satisfaction Index is a unique quality sensor that points to quality related anomalies. This helps to identify improvable product features even if the features have been realized above market standards. The Relevancy Index enables a feature weighting. Each index can be calculated on freely definable topics. The measures are robust towards different market penetrations and user activities. In addition, the influence of different polarity ratings is neglected because only the polarity distribution is of interest.

By using the proposed pivot analysis scheme, all three indices provide a very fast quality insight to potentially thousands of user comments without the need to read any text. The comparison of different products, manufacturers or features enables the detection of potential anomalies. These anomalies can be further investigated by additional deep-dive analyses. Word clouds, relation graphs and time analyses enables an analysis protagonist to filter the relevant data set and thus to limit the number of texts to be read.
CHAPTER 6

Conclusion and Perspectives

In this thesis an interactive Social Media monitoring system is proposed to extract quality related information from user generated content (UGC). This chapter summarizes the contributions that have been made to provide a scalable, reliable and flexible analysis architecture. Conclusions, which can be drawn from the results, will be shortly repeated. Finally, possible system improvements will be discussed.

6.1 Contributions and Conclusions

This thesis deals with the creation of the Social Media monitoring system AIM which is capable of extracting quality related information from UGC. It combines a set of state-of-the-art algorithms to structure the a-priori unstructured data. In this context, three special aspects have been identified where new methods are required to support reliable data analyses: the data collection, the topical data analysis and finally the interactive analysis of UGC.

The first aspect deals with the collection of UGC. It is discussed in chapter 3. A Social Media monitoring system cannot be based on the results of public search engines. Reliable and reproducible analysis results enforce furthermore an individual crawling process in which a Social Media specific
data extraction can be applied. Therefore, this thesis proposes a special multi-agent based crawler architecture to crawl the large amount of data within many different data sources. The flexible infrastructure supports the realization of different crawling policies, crawler behaviors and extraction algorithms for different Social Media systems with minimal reimplementation efforts. The architecture thus provides unique properties to crawl UGC in an efficient way. Two implementations have been described for weblogs and Internet fora. Both crawl only those web pages that are likely to contain UGC and thus support an optimized crawler behavior.

To store the relevant data only, including the user comments and related meta data, different content extraction algorithms have to be applied. Noisy analysis results can be avoided by neglecting navigation elements, advertisement and other superfluous elements. The multi-agent based crawler architecture therefore applies a universal wrapper algorithm that first extracts the complete discussion and then the individual comments. As the generation of these wrappers is very time consuming, an unsupervised wrapper induction algorithm is presented in chapter 3. It automatically detects discussion structures and generates XPath-based wrapper algorithms. These algorithms can be directly included in the multi-agent based crawling architecture. The approach was published in [10]. It is the first technique that automatically detects user comments to generate appropriate content extraction algorithms.

The crawled data has to be further analyzed. In doing so, the extraction of discussed topics is the most important task. The proposed Social Media monitoring system supports three different types of topic analyses. First of all, a-priori defined topics have to be detected within the user comments. On the one hand, these topics may be specified in detail. An analysis protagonist wants them to be identified with high quality. On the other hand, newly introduced topics are rarely defined and the analyst wants to get fast ad-hoc results. For both scenarios, there are many different algorithms already available. The third scenario tries to identify previously unknown topics. Hereby, a new fuzzified agglomerative hierarchical clustering
algorithm is presented in chapter [4]. This approach – published in [13] – generalizes well-known crisp clustering algorithms to fuzzy-logic. The created clusters are however no topic oriented groups but binary clusters. Therefore, a special pruning process was introduced. It modifies the created binary cluster graph to provide topical clusters. Each topical cluster can be divided into subtopics. These different abstraction levels can be used by an analysis protagonist according to his individual abstraction needs. The fuzzy clustering behavior ensures that multiple topics within user comments can be flexibly considered. The behavior is analyzed in detail and it is proven that even in case of fuzzy clustering the induced dissimilarity measure can increase monotonically. This is an important theoretical fact for agglomerative clustering algorithms as it ensures that the already clustered elements have been selected in the correct order. The deterministic clustering results are evaluated using a new evaluation measure for externally labeled hierarchical fuzzy clusters. It is the first evaluation method that allows for analyzing the quality of hierarchically created fuzzy clusters. The technique was also published in [13]. Applied to the results of the proposed fuzzified hierarchical clustering algorithm, the evaluation states that the generated topical clusters provide a high topic quality and are very stable with regard to newly added documents. The unique algorithm properties fulfill all requirements which are needed for blind-spot analyses in UGC.

The multi-agent based crawler architecture and the fuzzified hierarchical clustering algorithm are two special components of the Social Media monitoring system AIM (Automotive Internet Mining). This interactive analysis system is described in detail in chapter [5]. It is realized as a three-tier software architecture which makes the complete system scalable and its individual components exchangeable. In the data backend, the user comments are analyzed in detail and several information extraction algorithms are applied to identify discussed topics, sentiments and relations. The different kinds of topics are unified in the second architecture level, the analysis backend. Thus, the analysis protagonist can freely analyze ad-hoc defined topics, clas-
Chapter 6. Conclusion and Perspectives

The extracted information is further used in the analysis backend to provide different abstraction levels. Hereby, three new quality indices have been proposed. The Market and Product Satisfaction Index informs the analysis protagonist whether users are discussing special topics positively or negatively. The Relevancy Index states the importance of a given topic. All three indices enforce the definition of a reference dimension to provide robust and reliable reports. The special abstraction technique is used in a newly introduced pivot analysis scheme by which the analysis protagonist can investigate thousands of user comments without the need to read any user comment manually. The analysis frontend extends this analysis method by further deep-dive analysis techniques, namely time analyses, word clouds and relation graphs. They support filtering the large amount of data to a relevant but small data set. The small number of user comments within this subset can then be read manually to extract additional background information, e.g. problem circumstances and customer expectations. The unique advantages of AIM are demonstrated by two different evaluations. The first evaluation compares the Market Satisfaction Index to the state-of-the-art quality index in the automotive sector provided by J.D. Power\(^1\). The second evaluation presents use cases of real-world analyses, in which the newly introduced Product Satisfaction Index has shown unique advantages to identify shortcomings which dissatisfy customers. The proposed quality indices have been published in [8] and [9]. The applied framework for the pre-processing was investigated in [12].

6.2 Perspectives

The results of this thesis have shown that UGC provides unique analysis possibilities to discern customers’ definition of “quality”. The available abstraction techniques enable an analysis protagonist to investigate a large

\(^1\) J.D. Power and Associates is the leading market research company in the automotive sector.
number of user comments with minimal time effort. However, the gained quality insights are limited to freely provided data. This might be a disadvantage for long-running products like automobiles. In the life cycle of such a product, there are many different minor changes which are in general not visible to the customer. For example, the electronic equipment changes very fast and problem solutions typically depend on the installed software version. This important information is normally not available in UGC. Thus, Social Media based quality analysis is not necessarily able to provide all required information. In these cases, AIM just provides hints to potential problems. Additional analyses are necessary. Therefore, future work should investigate whether additional data can be integrated into the Social Media monitoring system.

One possibility is the integration of already available repair order reports. This data not only includes textual data but a large number of structured information, e.g. the exact product version. In an integrated system, it is possible to search for abnormalities in Social Media and then to switch to more structured data like repair orders to obtain additional information. However, such analyses require a common data model so that both data sets can be connected with each other. This could prove to be quite challenging, especially as both data contain different types of shortcomings.

Another very interesting addition to AIM would be the possibility to create ad-hoc surveys. Such surveys could be provided to customers to get further detailed information. As the surveys are much more specialized, they could easily gain specific required information. Most Internet fora already provide the possibility to directly execute small surveys. AIM could use these possibilities to post surveys in those Social Media systems where the abnormalities were found. Thus it would be possible to get further information by those users who have mentioned the problems. This feature could strongly minimize the costs of executing surveys.

Combining these efforts could take an analysis protagonist one step further to getting to know the customers’ concept of quality and their expectations.
Bibliography


[127] OECD. Participative web and user-created content: Web 2.0, wikis and social networking. [http://www.oecd.org/document/40/0,3746,en_21571361_38620013_39428648_1_1_1_1,00.html](http://www.oecd.org/document/40/0,3746,en_21571361_38620013_39428648_1_1_1_1,00.html) (01-19-2011), 2007.


Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den 18. Juni 2013

..........................................
(Unterschrift)
Scientific Curriculum Vitae

Personal information
Name / First name: Bank Mathias
Date of birth: 12/09/1981

Contact information
E-Mail: mathias@mathias-bank.de
Web-Page: http://www.mathias-bank.de

Education and qualification
since 04/01/2011 Head of Research, Pattern Science AG
03/01/2008 – 02/28/2011 Ph.D. student
Daimler AG, Group Research, Quality & Reliability
Supervisor: Prof. Dr. Gerhard Heyer, Department for Natural Language Processing, University of Leipzig
12/01/2007 – 02/28/2008 Research assistant
Faculty of Mathematics and Economics, University of Ulm
12/01/2004 – 11/30/2007 Diploma in Computer Science
University of Ulm
09/01/2002 – 11/30/2007 Student trainee
Daimler AG, Group Research, Quality & Reliability