

Advances in Computer Vision and Pattern Recognition



Frank Hopfgartner *Editor*

# Smart Information Systems

Computational Intelligence for Real-Life  
Applications

 Springer

The Springer logo, which is a stylized white chess knight piece on a pedestal, followed by the word "Springer" in a white serif font.

# **Advances in Computer Vision and Pattern Recognition**

## **Founding editor**

Sameer Singh, Rail Vision, Castle Donington, UK

## **Series editor**

Sing Bing Kang, Microsoft Research, Redmond, WA, USA

## **Advisory Board**

Horst Bischof, Graz University of Technology, Austria

Richard Bowden, University of Surrey, Guildford, UK

Sven Dickinson, University of Toronto, ON, Canada

Jiaya Jia, The Chinese University of Hong Kong, Hong Kong

Kyoung Mu Lee, Seoul National University, South Korea

Yoichi Sato, The University of Tokyo, Japan

Bernt Schiele, Max Planck Institute for Computer Science, Saarbrücken, Germany

Stan Sclaroff, Boston University, MA, USA

More information about this series at <http://www.springer.com/series/4205>

Frank Hopfgartner  
Editor

# Smart Information Systems

Computational Intelligence for Real-Life  
Applications

 Springer

*Editor*  
Frank Hopfgartner  
Technische Universität Berlin  
Berlin  
Germany

ISSN 2191-6586                      ISSN 2191-6594 (electronic)  
Advances in Computer Vision and Pattern Recognition  
ISBN 978-3-319-14177-0              ISBN 978-3-319-14178-7 (eBook)  
DOI 10.1007/978-3-319-14178-7

Library of Congress Control Number: 2014957701

Springer Cham Heidelberg New York Dordrecht London  
© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media  
([www.springer.com](http://www.springer.com))

# Preface

In the eighteenth century, the world was at a shifting point with new manufacturing processes being introduced that allowed humans to perform mechanical work much more efficiently than the preceding hand production methods. The steam engine, patented in 1781 by Scottish engineer James Watt, is often considered to be one of the key inventions of that time that triggered the industrial revolution. Due to inventions such as the computer, the Internet, and other digital devices, we are currently witnessing what can be referred to as the digital revolution, i.e., the shift from an analog, mechanical, and electronic world to a digital one.

While the driving force behind the industrial age was to optimize the industrial process using mechanical tools, today we aim to optimize work processes by analyzing data that are created by digital devices. In fact, an increasing number of companies rely on new technologies and services that produce, analyze, and access this data in multiple forms. Consider, for example, an online fashion retailer that assists us in selecting suitable clothing by analyzing our physical appearance and current fashion trends, mobile apps that rely on real-time sensor data to help us to successfully avoid traffic in big cities, or news content providers that assist in comprehending the world around us by providing hierarchically structured news overviews. Truly, the acquisition and analysis of data can be considered to be the new oil that drives our economy.

We usually distinguish between two types of data, namely *big data* and *small data*. A popular definition of big data is provided by IBM, which describes it in terms of four dimensions: volume, variety, velocity, and veracity. Pollock argues in his keynote address at the 2012 European Data Forum in Copenhagen that although there is a big hype around the analysis of big data, the real challenges and opportunities arise from the analysis of small data such as local household energy expenditure or time schedules of local buses. A definition of small data is provided by former McKinsey consultant Allen Bonde who argues at the 2013 Data Pulse Summit in Boston that “small data connects people with timely, meaningful insights (derived from big data and/or ‘local’ sources), organized and packaged—often visually—to be accessible, understandable, and actionable for everyday

tasks.” He also refers to it as “the last mile of big data,” i.e., the data users or customers can interact with.

Key techniques for a data-centric optimization of work processes are personalization, data mining, machine learning, knowledge discovery, and information management approaches. In other words, context-aware algorithms are required to understand, interpret, and react upon input data, and adapt their output based on external input parameters. The English physicist Stephen Hawking even argues that the ability to adapt to change is an indicator for intelligence. Following this argument, algorithms that adapt to change can be seen as computationally intelligent—or smart. Therefore, we refer to systems that rely on such computational intelligence as *Smart Information Systems*. As early as 2008, Marissa Mayer, the current CEO of Yahoo! and former vice president of Search Products and User Experience of Google Inc., predicted in an interview held at the LeWeb conference in Paris that “in the future personalized search will be one of the traits of leading search engines.” This statement reflects the increasing attention that smart information systems draw from both Academia and Industry. With increasing computational power, smart algorithms enable us to identify patterns, test research hypotheses, or to create data models, hence shedding light on the potential usage of this data.

However, although such techniques have matured over the past few years, there seems to be an increasing gap between current research trends in the analysis of data and the application of data analysis techniques in industry. NASA scientist Kiri Wagstaff even argued during a plenary session of the largest machine learning conference (ICML) in 2012 that “research has lost its connection to the problems of importance for the larger world of science and society.” She criticizes that much research is performed on evaluating novel algorithms using limited and artificial datasets, hence breaking adrift from answering the question of what computational analytics techniques can actually be used for.

In this book, we present smart information systems for the private and public sectors. Further, an overview of research questions that can be studied by applying computational intelligence is given, followed by a description of the algorithms, tools, measures, and evaluations used to answer these research questions. Each chapter can be seen as a guideline for transforming raw data into effective smart information services.

## **Book Outline**

This book illustrates potentials and challenges that arise from analyzing data for the provision of smart information services. In each chapter, we discuss individual use cases. All use cases cover real-world research challenges faced by parties as diverse as leading SMEs, multinational manufacturers, service companies, and the public sector, and are currently being funded (or have very recently been funded) from national and international sources. The book is composed of three sections: In the first

section, we present novel information aggregation services that illustrate how textual data can be employed to generate smart information services. We focus on three different domains, namely information aggregation services for individuals as well as services for the public and private sectors. These use cases showcase how information can be aggregated to provide easier access. In the second section, we outline personalization and recommendation systems that tailor information based on users' individual preferences. Again, we showcase application scenarios under different categories, such as the academic challenges of creating such services and provision of services based on the analysis of data. In the final part, we focus on sensor-based knowledge acquisition services, i.e., we concentrate on the analysis of sensor data that can then be used to provide a clear picture of our world. We present four different scenarios that showcase how computational intelligence allows communities, companies, and individuals to better understand their own environment and products.

## Meet the Marks Family

Although the individual use cases can be treated separately from each other, they all have in common that they focus on the analysis of data to provide smart information services that ease our everyday life. In order to illustrate this connection,



Clara, Suzanne, Steven, and Carl Marks. This and other graphical illustrations depicting scenes from the use cases that are presented in this book at courtesy of Sebastian Preuße, Berlin.



each chapter starts with a short episode from the lives of members of the Marks Family.

The Marks—Steven and Suzanne and the two kids Clara and Carl—represent an average family that lives in a suburb of a larger European city. Both parents go to work, their son Carl still goes to high school, their daughter just graduated and is now doing an internship before deciding what to do next with her life. Any resemblance to real persons, living or dead, is purely coincidental. Each episode outlines the need for a smart information system that would help them in typical situations that we all might be facing on a daily basis.

## **Authors of the Book Chapters**

The authors of the book chapters are members, project partners, or associates of the Competence Center Information Retrieval and Machine Learning<sup>1</sup> (CC IRML) of DAI-Labor of Technische Universität Berlin. The lab focuses on the development of intelligent systems and solutions, referred to as “smart services and smart systems,” that support us in our everyday life. Key to the success of such systems is the preceding analysis of data which is constantly created by these systems. The authors collaborate closely with industrial partners on a daily basis. Consequently, they have extensive experience in grasping the significant differences between Academia and Industry and are able to communicate this in their chapters, thus bridging the gap between both communities. Their expertise in computational intelligence puts the authors in an ideal situation to provide a state-of-the-art introduction to this field.

Berlin, September 2014

Frank Hopfgartner

---

<sup>1</sup> <http://www.dai-labor.de/en/irml/>

# Contents

## Part I Smart Information Aggregation Services

<b>1</b>	<b>Intelligent News Aggregator for German with Sentiment Analysis</b> . . . . .	<b>5</b>
	Danuta Ploch	
<b>2</b>	<b>Twitter Sentiment Tracking for Predicting Marketing Trends</b> . . . .	<b>47</b>
	Cagdas Esiyok and Sahin Albayrak	
<b>3</b>	<b>Health Assistance for Immigrants</b> . . . . .	<b>75</b>
	Till Plumbaum, Funda Klein-Ellinghaus, Anna Reeske, Kristin Pelz and Frank Hopfgartner	
<b>4</b>	<b>Information Aggregation in an Enterprise</b> . . . . .	<b>99</b>
	Erwin Gunadi and Sahin Albayrak	

## Part II Personalization and Recommendation Services

<b>5</b>	<b>Semantic Movie Recommendations</b> . . . . .	<b>125</b>
	Andreas Lommatzsch	
<b>6</b>	<b>News Recommendation in Real-Time</b> . . . . .	<b>149</b>
	Benjamin Kille, Andreas Lommatzsch and Torben Brodt	
<b>7</b>	<b>Personalized Information Access Using Semantic Knowledge</b> . . . .	<b>181</b>
	Till Plumbaum and Andreas Lommatzsch	
<b>8</b>	<b>Personalized Fashion Advice</b> . . . . .	<b>213</b>
	Till Plumbaum and Benjamin Kille	

**9 Gamification of Workplace Activities . . . . . 239**  
Michael Meder, Brijnesh Johannes Jain, Till Plumbaum  
and Frank Hopfgartner

**Part III Sensor-Based Knowledge Acquisition and Signal  
Processing Services**

**10 Optimization of In-House Energy Demand . . . . . 271**  
Stephan Spiegel

**11 Detecting Violent Content in Hollywood Movies  
and User-Generated Videos . . . . . 291**  
Esra Acar, Melanie Irrgang, Dominique Maniry  
and Frank Hopfgartner

**12 Discovery of Driving Behavior Patterns . . . . . 315**  
Stephan Spiegel

**13 Intermodal Mobility Assistance for Megacities . . . . . 345**  
Esra Acar, Marco Lützenberger and Marius Schulz

**Index . . . . . 369**

# Contributors

- Esra Acar** Technische Universität Berlin, Berlin, Germany
- Sahin Albayrak** Technische Universität Berlin, Berlin, Germany
- Torben Brodt** Plista GmbH, Berlin, Germany
- Cagdas Esiyok** Technische Universität Berlin, Berlin, Germany
- Erwin Gunadi** Technische Universität Berlin, Berlin, Germany
- Frank Hopfgartner** Technische Universität Berlin, Berlin, Germany
- Melanie Irrgang** Technische Universität Berlin, Berlin, Germany
- Brijnesh Johannes Jain** Technische Universität Berlin, Berlin, Germany
- Benjamin Kille** Technische Universität Berlin, Berlin, Germany
- Funda Klein-Ellinghaus** Leibniz Institute for Prevention Research and Epidemiology, Bremen, Germany
- Andreas Lommatzsch** Technische Universität Berlin, Berlin, Germany
- Marco Lützenberger** Technische Universität Berlin, Berlin, Germany
- Dominique Maniry** Technische Universität Berlin, Berlin, Germany
- Michael Meder** Technische Universität Berlin, Berlin, Germany
- Kristin Pelz** AOK Bundesverband, Berlin, Germany
- Danuta Ploch** Technische Universität Berlin, Berlin, Germany
- Till Plumbaum** Technische Universität Berlin, Berlin, Germany

**Anna Reeske** Leibniz Institute for Prevention Research and Epidemiology,  
Bremen, Germany

**Marius Schulz** Technische Universität Berlin, Berlin, Germany

**Stephan Spiegel** Technische Universität Berlin, Berlin, Germany

# Part I

## Smart Information Aggregation Services

### Overview

Documents such as The Book of Kells, a manuscript written around 800 AD are impressive examples that illustrate how much time and effort went into creating early textual documents. Books were expensive to produce and, consequently, owning large book collections was considered to be both a source of knowledge and also a status symbol for great power and wealth. This changed significantly in the fifteenth century when German blacksmith Johannes Gutenberg triggered the printing revolution with his invention of mechanical movable type printing. Suddenly, distributing and owning text documents was no longer just a privilege of the richer parts of society as his invention allowed for the mass production and spread of printed documents. Undeniably, his invention is one of the most important events in modern history since it laid the foundations for our knowledge-based society. Two more recent inventions significantly changed the way we create, distribute, and interact with textual documents even further. The introduction of the computer allowed us to create documents in digital format, hence enabling us to create multiple copies of the same textual document without any quality loss. The second important invention was the Internet which allowed us to easily distribute these digital documents. Given the widespread access to Internet that allows almost everyone to create and share text, it is a logical consequence that we are facing an ever-increasing amount of information in textual form. In fact, as of September 2014, more than one billion websites with even more webpages are available online. This constant information input is often referred to as information overload since the sheer amount of information that is created is impossible to be processed by the average user. Therefore, approaches and methods need to be developed that support us in finding the right information in this “data ocean.”

In the first part of this book, we present four use cases that center around helping users to overcome the information overload that they are facing. We focus on three different approaches: (1) analyzing textual documents to provide a summarized view of the documents’ content, (2) providing semantically enriched access to information, and (3) easing access to information by aggregating documents.

In Chap. 1, Ploch approaches the information overload challenge in the context of online news. Nowadays, it is doubtful that there is any major newspaper that does not maintain an online portal. Apart from saving costs that occur for printing and distributing traditional newspapers, the main advantage of distributing news online is that readers can be reached almost immediately. On one hand, the wide range of news brings many benefits to readers; they can find comprehensive information and capture news from different perspectives. On the other hand, the increasing amount of news material complicates their handling, which requires tools for facilitating consumption of news articles. In addition to reporting facts, news articles also contain opinions which may be very important for helping readers making decisions and for public figures to control their perception in the media. Analyzing the large number of news articles manually is next to impossible. Ploch presents how online readers of newspapers can be offered a structured overview of news. Focusing on news published in the German language, she illustrates how news articles can be categorized by topic and time of publication. In addition, she illustrates means to track the development of news events over time and to track opinions and resonance in the media about popular topics, persons, or organizations.

Besides professionally edited content on news portals, various alternative information sources exist on the web. Social networks and services like Twitter offer a wealth of information as thousands of users publicly exchange information. These so-called microblogs give voice to billions of people who often use this technology to express their opinions about brands, products, and persons. Analyzing these opinions can be of high value for companies since knowing where a brand is popular can be an important lead for the marketing strategy. Esiyok and Albayrak discuss in Chap. 2 how tweets can be analyzed to identify users' opinions about brands and present an application that displays the popularity of brands in specific locations on a map. This helps to identify trends and trendsetters and can offer aid for marketing decisions.

Chapter 3 focuses on a specific type of information portal. Addressing the trend that users more often use the internet for informing themselves about any types of topics, healthcare providers and governments started setting up education campaigns on the WWW. Although healthcare providers have specific interest in providing health information services to all their clients, immigrants have been identified as a vulnerable population cohort that benefits less from existing healthcare systems since language and cultural barriers prevent them from using existing prevention services. Plumbaum et al. present an online health assistant that consists of three parts: (1) a multilingual health information assistant, (2) a cooking assistant, and (3) a virtual trainer. These assistants present a comprehensive approach to support people for healthier living by giving them information about health topics, supporting healthier eating and getting enough exercise.

In Chap. 4, Gunadi and Albayrak address the information overload challenge in the workspace environment. They argue that the bigger a company, the more complex is their IT infrastructure and, consequently, more resources exist where employees can store information. Examples include companies' web server,

internal file servers, but also the employees' personal desktop computers. In their chapter, they present an information aggregation system that eases employees' information gathering task when accessing distributed information. They outline challenges that distributed information cause and present different methods to aggregate retrieval results coming from these different sources.



# Chapter 1

## Intelligent News Aggregator for German with Sentiment Analysis

Danuta Ploch

**Abstract** The comprehensive supply of information from different points of view, e.g., from the thousands of news articles published online every day, is a tremendous advantage of the digital era. However, the immense amount of news material poses a significant challenge to interested readers: It is hardly possible to fully digest this wealth of information, so that the need for systems supporting intelligent news consumption arises. This chapter describes an approach to automatically mining opinions from topically related news article clusters. We focus our work on the extraction of quotations from German news articles and on analyzing the quotations according to the sentiments they express. Our approach is realized as a news aggregation system capable of handling real-world news streams. We describe the architecture and interface of our news aggregator, and present a rule-based method for quotation extraction as well as our supervised approach to sentiment analysis. We evaluate the implemented models on two human-annotated datasets, which can be made available upon request.

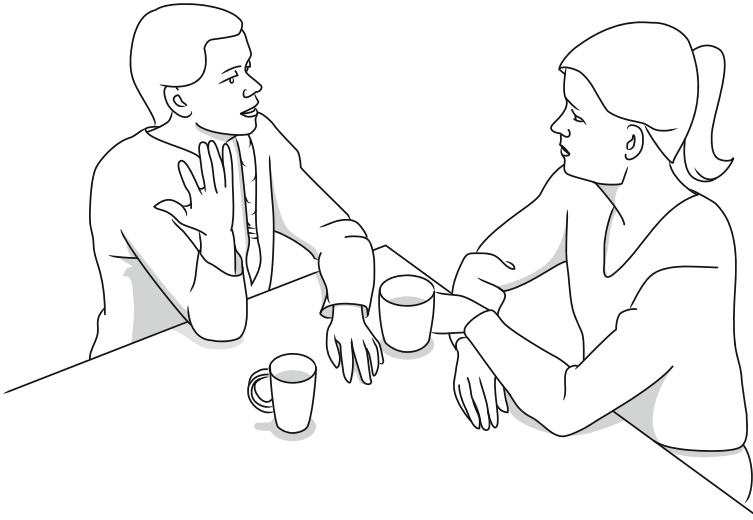
### “As Many Heads, So Many Opinions” (Horace)

Since her 15th birthday Clara dreamed of becoming a journalist. She worked for the school newspapers and was a member of the debating society. Now, after passing the exams, Clara finally started living her dream. She remembered the day when she received the good news that they had accepted her application for an internship at the local newspaper. “That’s a great chance for you, don’t ruin it”, her father told her. What a typical statement from her dad, she thought. Why would she ruin it? And what is there to ruin in an internship anyway? In fact, her tasks so far seemed quite forward. Even Dad would be able to do that, she thought with a grim on her face. At the beginning her tasks were restricted to copying articles and typing e-mails, but her boss quickly recognized her talent and assigned her an important research task for

---

D. Ploch (✉)  
Technische Universität Berlin, Berlin, Germany  
e-mail: danuta.ploch@dai-labor.de

a special issue on “espionage amongst friends”, a topic that gained attention after the revelations of the US-whistleblower Edward Snowden in 2013. In particular, Clara and her internship friend Martha were supposed to analyze newspaper reports about the behavior of US-president Barack Obama and the German Chancellor Angela Merkel before and after the revelations. The interns should identify the politicians’ attitudes toward national security-related topics and if they possibly changed after the publication of the secret documents by Snowden. How the politicians have commented on the revelations and spying in general? Did they disagree on all points or was there also agreement? Which statements about spying among allies were the most controversial? The interns were also advised to work out whether Snowden’s actions influenced Merkel’s positions to crucial election topics during her campaign for the German federal election in 2013.



Clara and Martha split up their task into two parts. While Martha took over the investigation concerning Barack Obama, Clara started to search for news articles about Angela Merkel. She entered ‘Merkel Snowden 2013’ in the search field of her search engine and immediately thousands of news articles were presented to her as a never ending result list. “It will take days to go through all that news articles”, Clara whined. But since Clara has always been a fighter, she sorted the news articles by date and began with her research—news by news. Clara realized soon that sorting news articles by date was not really helpful for detecting topics and positions. Although she limited the date range and disabled displaying duplicates, the mass of news articles was overwhelming and she had to face problems like off-topic news articles concerning nonpolitical issues. During the coffee-break she met her friend Martha and complained: “I’m feeling like Cinderella: ‘The good ones go into the pot, the bad ones go into your crop’”. “I thought journalism would be fun”, she added and couldn’t hide her sense of frustration. “But it is!”, answered Martha, Haven’t you tried out our in-house news archiving system? It does all the stressful preprocessing-work for you. Depending on your search query the system clusters news articles to topics and

identifies thematically connected topics. It even extracts quotations and their polarity toward the contained opinion target”. Clara had already worked for 2 months for the newspaper but had never heard of such a system. Fortunately, Martha told her now because 10 min ago she was on the verge of discarding her evening plans and working overtime instead of going to the cinema. She grabbed her coffee cup and returned to the office with a smile on her face—her evening was saved.

## 1.1 Introduction

In times of Twitter, Facebook, and other social media services news is broadcast around the world in no time at all. If something more or less newsworthy happens, users immediately take their smartphones and post it. Today’s social media services bring a lot of benefits. For example, it is not possible to imagine reporting without Twitter from crisis regions. Still, since potentially everyone may distribute information, the question of reliability arises. In 2013, the ambiguous Twitter hashtag #nowthatchersdead confused a wide range of Twitter followers.<sup>1</sup> Many users interpreted the hashtag as “Now that Cher’s dead” and retweeted that the pop queen Cher has died. In reality, the death of the former Prime Minister Margaret Thatcher (“Now Thatcher’s dead”) was announced. This example shows how fast rumors may come up and be spread around if no professional journalists are involved. Thus, editorially written news articles still remain one of the most important information sources.

In comparison to user-generated reports on the Web, journalistic work of reputable news agencies is considered reliable and credible. News articles are thoroughly researched and well formulated and they often report not only the piece of news itself but also provide additional context and background information. Besides, local newspapers or special issues on specific topics may cover topics not mentioned in social media which still are important for a large number of readers. As with user-generated content, the amount of editorially prepared news material is remarkable. In general, print media publish their data also on the Web and in addition there are numerous online news papers available.

The range of available news material allows users to stay informed about what happened and to look at news from different points of view. At the same time the considerable number of news articles complicates their handling and requires tools for helping the users to search and browse them. News aggregation systems support users in exploring news articles by analyzing and organizing news articles. To avoid information overload, they first detect (nearly) duplicate news articles and hide them from the reader. This is necessary because news articles published by well-known press agencies like *dpa*<sup>2</sup> (Deutsche Presse-Agentur GmbH) are redistributed by various news providers. Then, their main task is to identify news articles dealing with

---

<sup>1</sup> <http://news.msn.com/pop-culture/confused-by-thatcher-tweets-cher-fans-upset-by-numbernowthatchersdead/>.

<sup>2</sup> <http://www.dpa.de/>.

the same news story or topic from a large stream of text messages. As news stories evolve over time, they continuously group and rank news articles in order to constantly reveal relevant and hot topics and to enable their monitoring.

News texts do not only report facts about what has happened but also reflect opinions of involved entities such as persons or organizations. They serve therefore as a valuable source of opinions and help users making their decisions based on them. Depending on the news type the opinions are directed toward a wide variety of topics or other entities. For example, before political elections news articles echo the politicians' attitudes toward current election issues and influence the vote behavior. The perception of products or services is a precious piece of information for companies and often a key factor in a company's decision-making process.

As with events and topics, news aggregation services facilitate the search for and exploitation of opinionated text. Manually finding and evaluating opinion-relevant parts may be infeasible for users. Therefore, the detection of subjective text parts and the classification of text into different types of opinions are crucial tasks in news processing systems.

In this chapter we focus on news aggregation services to organize and analyze news articles. Section 1.2 describes approaches for grouping news articles depending on events and topics. We start by describing methods to detect and track short-term events in news streams. Then, we discuss the clustering of events into more abstract meta-topics. News articles often contain citations that underline reported issues. Therefore, Sect. 1.3 concentrates on the extraction and evaluation of citations. Section 1.4 covers services to analyze news material with regard to expressed opinions. We present a news aggregation system that incorporates all introduced steps in Sect. 1.5 and conclude the chapter in Sect. 1.6.

## 1.2 News Aggregation Model

News aggregation systems like Google,<sup>3</sup> Bing<sup>4</sup> and Yahoo!<sup>5</sup> organize and present news articles from a large number of sources in order to offer users a comprehensive supply of information. The enormous amount of news material published every day requires a continuous and suitable preparation. In addition to the standard categorization of news article into the main columns such as "Politics", "Economy", "Sports", etc., and sorting the news items by date and/or language news aggregators apply Topic Detection and Tracking (TDT) techniques to group news articles related to the same events. Enhanced news aggregators offer additional services based on a deep analysis of the news sources and material. For example, Google assesses the sources and categorizes the content as *opinionated*, *detailed* or *preferred by the user*. We introduce a system that not only focuses on a high-level classification of news

---

<sup>3</sup> <https://news.google.com/>.

<sup>4</sup> <http://www.bing.com/news/>.

<sup>5</sup> <http://news.yahoo.com/>.

articles but also examines the content of the articles in order to grasp their meaning. Besides TDT at topic-level, the proposed system recognizes more abstract topics and performs quotation extraction and sentiment analysis based on the identified quotations. The system is capable of offering deep insight into single events and topics by highlighting named entities along with direct and indirect quotations. The users may inform themselves about involved entities, compare their comments, and learn about the perception of the entities and topics in the media landscape.

The proposed system was developed in close collaboration with Neofonie GmbH.<sup>6</sup> It is modeled with a processing pipeline as the central component. The system's structure is schematically outlined in Fig. 1.1. When going through the processing pipeline, the documents are enriched with more and more information. For each crawled news article a linguistic preprocessing is performed. The news articles are split into tokens and sentences and annotated with part-of-speech tags, named entities and lemmas (Sect. 1.2.1). Subsequently, the news articles are mined. On the

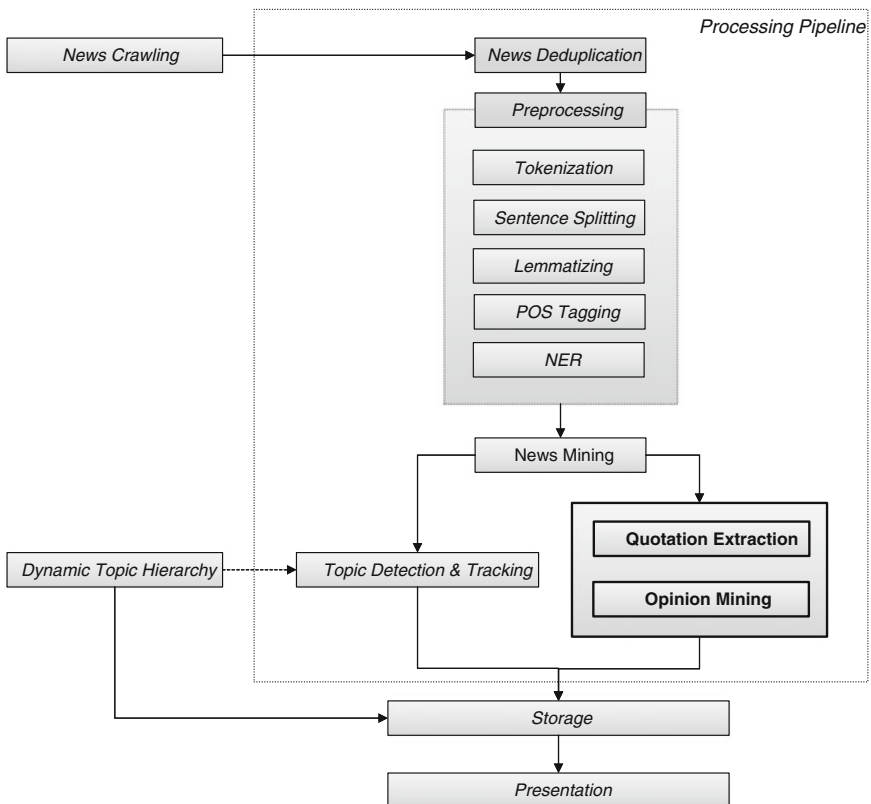


Fig. 1.1 System overview

<sup>6</sup> <http://www.neofonie.de/>.

one hand the news texts are regarded as objective information sources reporting facts. They are clustered according to events by the Neofonie GmbH (Sect. 1.2.2). On the other hand the system aims at identifying subjective parts in the form of quotations (Sect. 1.2.4) and at determining the sentiment polarity of the expressed statements (Sect. 1.2.5). In parallel to the pipeline, the Neofonie GmbH assigns the news articles to automatically identified abstract meta-topics, which connect thematically related topics (Sect. 1.2.3). The graphical user interface is described in Sect. 1.5 where also screenshots of the system are presented.

### ***1.2.1 Preprocessing***

The initial step for the analysis of news articles is linguistic preprocessing. After having crawled and deduplicated the news articles, we first split the text of each article into tokens and sentences. This is an important prerequisite for various linguistic tasks such as part-of-speech (POS) tagging, chunking, and also our quotation extraction approach. The next step is lemmatizing all words of the text. Mapping words to their canonical form allows looking them up in dictionaries in steps that follow. We find verbs starting quotations in this way. The task of POS taggers is to assign each word of a text its part of speech. We use the output of the POS tagger at several points in our system. For instance, we make use of POS information to compile feature vectors for our supervised sentiment analysis approach. The system exploits a lexicon-based named entity recognition approach. We use the German version of Wikipedia<sup>7</sup> for identifying and linking named entities. The named entities contribute to the concept vectors required for our topic detection and tracking approach (Sect. 1.2.2).

### ***1.2.2 Topic Detection and Tracking***

The Topic Detection and Tracking program defines an event as “something that happens at some specific time and place along with all necessary preconditions and unavoidable consequences”. Topics (or stories) comprise a triggering event and all directly related events and activities [17]. As news stories evolve over time, the task of TDT approaches is to either identify news articles starting a topic or to assign news articles to existing topics. In our system we employ an incremental agglomerative clustering approach for TDT. We represent each news document as a vector of concepts including named entities. Each cluster represents a topic and is specified by a centroid vector with averaged concept weights of the covered news articles. Incoming news documents are compared to the centroid vectors of all existing topics. If

---

<sup>7</sup> <http://de.wikipedia.org/>.

the similarity to all existing centroids is lower than a predefined threshold then the incoming news article starts a new topic. The similarity measure is a combination of the cosine similarity between two vectors and a time-dependent penalty. Including a time-dependent penalty favors the assignment of news articles to new topics and at the same time avoids an infinite growth of existing topics [20].

### ***1.2.3 Dynamic Topic Hierarchy***

Classical news aggregators organize news articles, and the topic they belong to, in top-level categories like “Politics”, “Economy”, or “Sports”. In order to better navigate and track the development of related news stories, news stories may be organized in a hierarchy. The arrangement of topics in a hierarchy links not directly related topics and supports readers to recognize relationships between them. This is especially helpful in the context of searching news archives and recommending related news articles for further reading. Since news stories evolve over time and new events happen continuously we do not sort the news articles in predefined hierarchies such as the Metadata Taxonomies for News by the International Press Telecommunications Council<sup>8</sup> (IPTC) but propose the creation of a dynamic topic hierarchy arising from the current news situation. Based on previously detected topics (Sect. 1.2.2) we build thematically connected meta-topics and assign labels to them. We select the most probable headline from the set of news articles belonging to the meta-topic.

### ***1.2.4 Quotation Extraction***

Quotations are a common stylistic device to clarify and strengthen a statement. Basically, a distinction is made between direct and reported speech. Considering quotations in news articles, they underline reported facts and may express positions or views of the cited persons or organizations. By employing quotations at specific points in an article the author highlights statements that are especially significant and worth to be cited. In addition, quotations may be a suitable source for identifying subjective passages of a news article. In our system we apply a rule-based approach to quotation extraction. We address the extraction of direct and reported speech and assign a speaker to each identified quotation. Our solution normalizes quotation marks, makes use of linguistic annotations to detect reporting verbs or phrases that introduce quotations, the boundaries of direct and reported quotation parts, and finally the speaker, which we also call quotation holder in the following. We describe our approach to quotation extraction in detail in Sect. 1.3.

---

<sup>8</sup> <http://www.iptc.org/site/NewsCodes/>.

### 1.2.5 *Sentiment Analysis*

Sentiment analysis aims to detect and assess opinionated text. Often the synonym “opinion mining” is used to refer to the same task. Journalistic content such as news articles is considered to be a reliable information source and may serve as a basis for various natural processing (NLP) tasks. News articles are objective reports and ideally, the authors do not express their attitudes. But this does not mean that news articles cannot contain any opinionated text. Usually, news texts reflect opinions and sentiments of newsworthy people, which are directed toward topics or toward other entities. The identification of passages that echo subjective statements, attitudes, or views and distinguishing them from the objective parts reporting facts is the first step in many sentiment analysis approaches before analyzing the identified passages. Exploiting quotations is one way to tackle the problem of mining opinions from news articles. Quotations are a trustworthy form of mirroring what people or organizations have said in an original and genuine manner. In addition, given a quotation, its speaker is usually the opinion holder, the entity that expresses the opinion. We propose a supervised two-stage approach where we first identify subjective quotations and, second, classify the subjective quotations in either POSITIVE or NEGATIVE. All remaining quotations are regarded as NEUTRAL (Sect. 1.4.3). We explore a range of features established in sentiment analysis of other text genres (e.g., product reviews, tweets) and examine to what extent they are suitable to separate subjective from neutral and positive from negative quotations (Sect. 1.4.6).

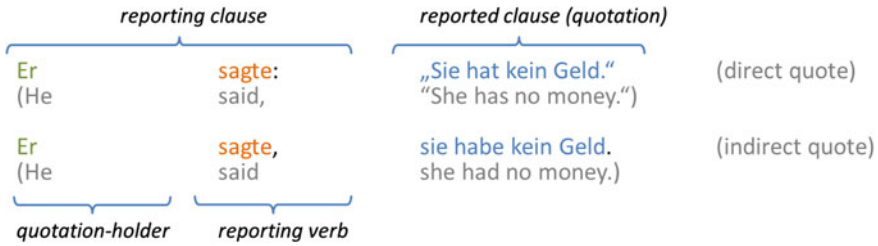
## 1.3 Quotation Extraction

Quotations report what persons or organization have said. In news articles they are often used to confirm claims made by the author and indicate the importance of the transported information. Thus, they may be an important piece of news article for various text processing tasks such as sentiment analysis or news summarization. In this section we present our approach to quotation extraction, which covers both the extraction of direct and indirect quotations as well as the assignment of a quotation speaker.

### 1.3.1 *Introduction*

Quotations repeat a speech, text, or statement expressed by a speaker and can be distinguished into direct and reported speech. We refer to reported speech also as indirect speech and to a speaker also as quotation holder in the following. Quotations are composed of a reporting and a reported clause. Following Krestel et al. [24] Fig. 1.2 shows an example of the structure of both, a direct and an indirect quotation. The reporting clause introduces the quotation. Besides the quotation holder, it may





**Fig. 1.2** The structure of quotations. A quotation is composed of a reporting and a reported clause. The reporting clause introduces the quotation. It includes the quotation speaker and an optional reporting verb. The reported clause encompasses the quoted content/text

also contain a reporting verb such as “sagte” (said) or “berichtete” (reported) and other circumstantial information such as the addressee or diverse other descriptive text. The reported clause encompasses the actual content that has been said.

*Direct speech* repeats things that have been said by a speaker as they are without any modifications. The repeated text is enclosed by quotation marks. In contrast to direct speech, *indirect speech* reports statements by modifying them grammatically or even rephrasing them. The grammatical change indicates that the expression was not uttered by the author, but by the original speaker. Indirect speech is composed of a main (reporting) and a subordinate (reported) clause. In German, the reported clause of an indirect speech often is introduced by the conjunction “dass” and uses the subjunctive mood for verbs. Quotations may consist of several reported clauses and we define quotations as *mixed* if both, quoted and unquoted reported clauses build the quotation.

**Our Contribution.** In our work we process German news articles and extract direct and indirect quotations along with a quotation speaker. We propose a rule-based approach that exploits linguistic information. Modeled as a processing pipeline our quotation extraction component first enriches news articles with linguistic annotations, which then are used to mine the complete quotation. We detect units of direct quotes by applying a pattern that takes into consideration different types of quotation marks. We exploit the presence of reporting verbs and other common phrases indicating quotations to underpin direct quotation candidates found by our pattern and to locate potential indirect quotations. In order to assign each quotation a speaker we make use of the output generated by a named entity recognizer and a part-of-speech tagger. We compile a list of candidate speakers and then apply rules that consider the type of the candidates and the proximity to the reporting verb to determine the quotation speaker. For evaluating our approach we manually created a quotation corpus from a set of German news articles. The corpus provides for each quotation its boundaries, the quotation speaker, a reporting verb or phrase, and the type of the quotation (direct, indirect or mixed). The corpus is available upon request and signing of an agreement.

### 1.3.2 Related Work

In the past, numerous solutions for the task of quotation extraction from newspaper material were proposed. The approaches differ in which technique they use, which language they support, in whether they extract direct, mixed, or indirect quotations (or all), and in how detailed they determine specific quotation units such as the quotation holder and other circumstantial information.

The majority of previously published work detects reporting verbs in news articles from a predefined or precompiled list and then extracts quotations based on rules that are derived by experts. An exact analysis of the news material in advance and the knowledge about the structure of quotations allow the identification of more or less fine-grained patterns that may vary from language to language. Usually, the patterns differ in the presence and position of lexical terms or syntactic information. There is consensus that for each quote a speaker needs to be extracted, because the information without the assignment of a speaker is of little use in most cases. Thus, many researchers represent quotations as a triple consisting of the quoted text, the quotation holder, and an optional reporting verb or a quotation introducing phrase.

The rule-based system presented by Pouliquen et al. [41] extracts around 2,600 direct quotations per day from a multilingual news stream. In order to keep the system extensible to other languages, the approach does not rely on linguistic information but on lexical patterns. The system recognizes quotation marks, reporting verbs, and person names (along with further information such as temporal or spacial modifiers, titles, and determiners) and applies three general and a couple of language-specific rules to find quotations. A simple named entity disambiguation solution serves as the accurate assignment of quotation holders. Still, the system misses quotations with speakers referenced by pronouns, since it does not perform anaphora resolution.

Krestel et al. [24] assemble a set of six basic patterns to extract quotations from news articles in English. They detect the most frequent reporting verbs using a finite state transducer and implement the identified patterns as a regular grammar. Existing GATE<sup>9</sup> components provide additional circumstantial information required during the quotation extraction process. In contrast to [41], that limit their approach to direct quotations, the authors treat indirect quotations as well.

The great part of the effort on quotation extraction and attribution has been made for English texts [24, 26, 35, 38]. Still, several publications focus their work on other languages than English. In particular, quotation extraction for Portuguese [10, 39] and French has been studied [11, 52].

Sarmiento and Nunes [10] present a system that handles Portuguese news articles. It finds direct and indirect quotations by applying 19 patterns and by exploiting a list of 35 reporting verbs. The system does not implement anaphora resolution for pronouns or noun phrases and therefore detects only speakers referenced by their proper name. The authors evaluated their approach manually on 570 quotations extracted by system.

---

<sup>9</sup> <https://gate.ac.uk/>.

De La Clergerie et al. [11] present an approach to quotation extraction from French news articles. Their rule-based approach includes a comprehensive linguistic processing chain with a deep parser. A postprocessing component constructs direct and mixed quotations based on parsing results, 230 quotation verbs, and direct speech parts signaled by quotation marks that were retrieved in previous processing steps. As with [11], the authors in [52] focus their quotation extraction approach on French news articles. Again, a rule-based approach is driven that exploits an automatically created lexicon of reporting verbs. The authors recognize 16 patterns matching indirect quotations and implement them as an unlexicalized grammar using an finite state machine.

Besides the rule-based systems [1, 10, 11, 24, 26, 41, 45, 52] a range of supervised approaches has been presented for the task of quotation extraction [35, 38, 39]. Fernandes et al. [39] propose a supervised solution using an Entropy Guided Transformation Learning (ETL) algorithm. They automatically generate rules instead of manually designing them. The work regards quotation extraction as a two-task problem. First, their system identifies quotations and, second, the quotations are associated with a speaker. Recognized named entities and the output of a co-reference component serve as a basis for the speaker assignment. To solve the subtasks different sets of features (named entities, terms, co-references, part-of-speech tags, etc.) are applied to the ETL algorithm. The developed system is capable of extracting direct and mixed quotations from Portuguese news articles. In order to train their system, the authors create the GLOBOQUOTES corpus.

The approach to quotation extraction from English texts proposed by O’Keefe et al. [35] makes use of supervised techniques as well. The authors solve the quotation extraction part by using a regular expression looking for text between quotation marks. Regarding quote attribution, which means finding the speaker of a quote, they cast the problem to a sequence labeling task. Inspired by Elson and McKeown [13], the authors encode news articles by replacing specific terms with symbols and by removing unnecessary information. Then, a set of features is calculated which includes distance, paragraph, nearby, quote, and sequence features, again following Elson and McKeown [13]. In order to efficiently predict the target speaker from a list of candidate speakers, the authors compare different types of class models and sequence decoding. They examine the effects of creating feature sets with and without gold standard labels. They conduct their experiments on three different datasets and find that when leaving out gold labels for feature calculation the performance drops significantly for classic literature but remains comparable regarding news articles.

Pareti et al. [38] focus their work on the extraction of indirect and mixed quotations from English-language news articles. The authors explore two supervised algorithms, namely a Conditional Random Fields (CRF) and a Maximum Entropy (ME) classifier. The token-based CRF classifier predicts IOB labels (I-inside, O-outside or B-beginning), marking the beginning and the end of a quotation, whereas the ME classifier decides whether a phrase-structure parse node is or is not a quotation. The classifiers largely rely on the same features but also incorporate classifier-dependent features. Instead of using a predefined list of reporting verbs, the authors train a

k-nearest neighbor (k-NN) classifier working with 20 feature types, that predicts whether an identified verb group introduces a quotation or not. The authors conclude that the token-based approach using a CRF classifier outperforms the rule-based baseline as well as the constituent-based approach using the EM classifier for all quotation types. Regarding quotation attribution, the authors transfer four methods described in O’Keefe et al. [35] for direct quotation and find that they all are suitable for indirect and mixed quotations.

There is few scientific work that aims the extraction of quotations exclusively from German news articles. To the best of our knowledge only Pouliquen et al. and Akbik and Schenck [1] deal with German language news articles. While Pouliquen et al. include German into their multilingual system as one of many languages, Akbik and Schenck present a system that automatically collects news from the main German news sites and then extracts direct quotations from these news articles. Their approach detects text between quotation marks as quote candidates and uses a named entity recognizer to identify potential speakers. Then a set of heuristics is used to determine the resulting quote-speaker tuples.

### 1.3.3 Approach

The proposed approach for extracting direct and reported speech from German news articles is rule-based. For each quotation the system identifies a speaker, a reporting verb, or a preparative phrase (like “..., so Angela Merkel”), and the quotation text with all its parts. We divide the task into five subtasks and model our quotation extraction approach as a processing pipeline where the news articles are annotated in each step of the pipeline with further information. Figure 1.3 demonstrates the included components and the working flow. Starting with a document preprocessing component we perform linguistic analysis like part-of-speech tagging and lemmatizing that serve as a basis for further processing steps. The normalization of quotation marks is important at this point as well. Detecting a reporting verb helps to identify the reporting clauses and is also a strong indicator of indirect speech. We therefore search for them in the next step of our pipeline. Subsequently, we identify the reporting clauses of direct and indirect quotations and determine the quotation parts and exact boundaries of the entire quotation. Note that the boundaries of indirect or mixed quotations may be ambiguous and in many cases difficult to recognize even by humans. In the last step of our pipeline we attribute one or more quotation holders to the previously identified quotations.

#### 1.3.3.1 Document Preprocessing

**Quotation marks normalization.** News articles may contain malformed markup. Especially in systems with automatic news harvesting from heterogeneous sources the collected texts may be erroneous, e.g., in terms of incomplete articles, misplaced

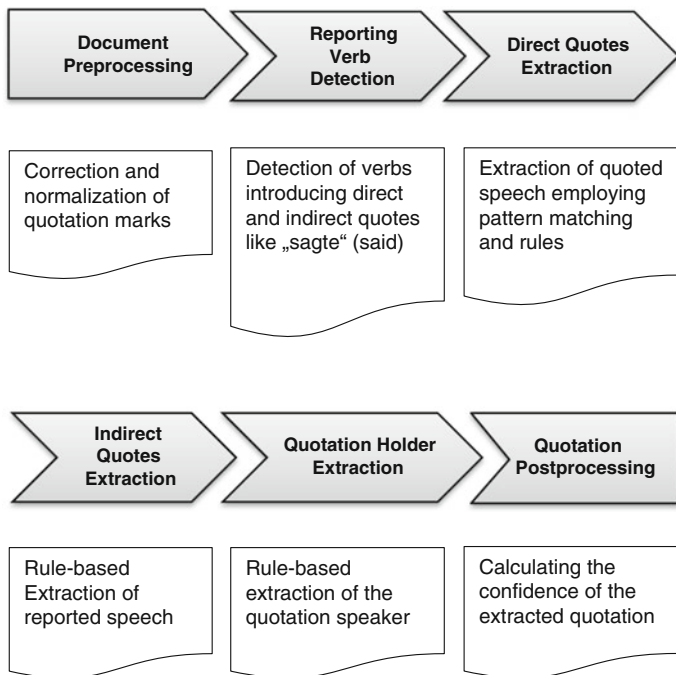


Fig. 1.3 The components of the quotation extraction pipeline

meta information or missing quotation marks. Regarding quotation extraction, texts released by different publishers may also contain varying styles of quotation marks like «»», “” or ‘ ’. Since quotation marks are crucial indicators for both direct speech and quotation boundaries, the correction and normalization of quotation marks is an important subtask in quotation extraction systems. In our system a document preprocessing component corrects errors arising from inconsistent quotation marks. It first replaces all quotation marks with uniform quotation marks and then counts the number of quotation marks. The component does not patch texts with an odd number of quotation marks, but adjusts quotations with different start and ending quotation marks like quotations starting e.g., with “and ending with ‘.

**Sentence Detection.** Quotations may consist of several sentences or sentence parts. For example, the quotation “*Wir sind noch immer hier. Wir kämpfen noch immer*”, sagte Santorum. (“*We are still here. We are still fighting*”, Santorum said.) is composed of two sentences. In such case the quotation extraction must recognize both parts and determine correct quotation boundaries. A sentence detection is therefore an essential preprocessing step. Furthermore, other linguistic algorithms used in news text analysis require sentences as a basis for their calculations. Our quotation extraction pipeline uses the Apache OpenNLP<sup>10</sup> Maximum Entropy sentence detec-

<sup>10</sup> <https://opennlp.apache.org/>.

tor. The sentence detector uses a predefined model trained on Tiger corpus data [6] for the German language together with a self-developed heuristic, i.e., we created a set of likely (‘.’, ‘?’ and ‘!’) and unlikely (Fr., Hr., Prof. ...) ends of a sentence and check the output of the OpenNLP sentence detector. If a sentence chunk has an ending that should never be an ending we merge again sentence parts that were incorrectly split.

**Lemmatization.** Determining the lemma of a word is a necessary step for our lexicon-based reporting verb finder described in (Sect. 1.3.3.2). In German the lemma of a noun normally is the “nominative singular” and of a verb it is the “infinitive present active” form. In our quotation extraction pipeline we make use of a morphological lemmatizer that looks up the words of the news article in a lexicon. The lexicon<sup>11</sup> was generated with “Morphy”,<sup>12</sup> a software tool for the morphological analysis of German text [25].

**Part-of-Speech Tagging.** Part-of-speech tagging is the task of predicting the grammatical category (noun, verb, adjective, ...) of a word based on the word’s definition and its surrounding context. In computer linguistics each word of a sentence is assigned a label from a predefined set of part-of-speech labels. For German often the “Stuttgart-Tübingen-Tagset” (STTS)<sup>13</sup> is used for labeling [44]. The proposed quotation extraction approach works with the Apache OpenNLP maximum entropy part-of-speech tagger. Together with the predefined model trained on the Tiger corpus the tagger predicts STTS labels for words of a given text.

**Noun and Verb Chunking.** The chunking component analyzes sentences and determines verb and noun groups. The groups then serve as input for the recognition of potential reporting verbs or quotation holders. For example, a speaker may be referenced as “die deutsche Bundeskanzlerin” (the German chancellor) or a reporting verb may be a compound of two words like “teilte mit” (informed). The phrases output by the chunker help to determine the correct boundaries. Our processing pipeline uses the Apache Open NLP maximum-entropy-based chunker. To recognize noun chunks we use an out-of-the box model distributed by Gunnar Aastrand Grimnes.<sup>14</sup> For the recognition of verb chunks we trained a model on the Tiger Corpus [6]. The Tiger Corpus contains 50,000 sentences in German taken from the “Frankfurter Rundschau” which are POS-tagged and annotated with syntactic structure.

**Named Entity Recognition.** When citing persons or organizations a pronoun, a noun phrase or the proper name of an entity can be used to reference the quotation speaker. State-of-the-art named entity recognizers mainly detect top-level entities like persons, organizations, and locations which may be a starting point for the detection of quotation holders. We integrated the Stanford named entity recognizer into our quotation extraction pipeline. The Stanford recognizer is implemented as a Conditional Random Field classifier [16]. We use a pre-trained model for German provided by [15] that labels tokens as person, organization, location, and miscel-

<sup>11</sup> <http://www.danielnaber.de/morphologie/>.

<sup>12</sup> <http://www.wolfganglezius.de/doku.php?id=cl:morphy>.

<sup>13</sup> <http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html>.

<sup>14</sup> <http://gromgull.net/blog/category/machine-learning/nlp/>.

**Table 1.1** The list of reporting verbs used for the quotation extraction approach

German reporting verbs				
Sagen	Behaupten	Aussprechen	Abraten	Teilen
Meinen	Warnen	Erwähnen	Raten	Klären
Fragen	Betonen	Bejahen	Ausfragen	Aufklären
Denken	Loben	Ermahnen	Ausplaudern	Mitteilen
äußern	Zugeben	Beichten	Erklären	Begründen

laneous. Since the Stanford classifier sometimes misses some named entities we decided to augment the list of named entities returned by the Stanford classifier by the named entities identified by the part-of-speech tagger described above. The type of the entities recognized in this way is tagged as UNKNOWN since the tagger marks the entities without providing a type.

### 1.3.3.2 Reporting Verb Detection

The detection of reporting verbs, that is verbs introducing quotations, is especially important for the recognition of reported speech and a quotation holder. Our reporting verbs detection approach is lexicon-based. We manually assembled a list of 25 common reporting verbs. We started with a set of six seed reporting verbs and extended the set by adding synonyms from Wortschatz Leipzig.<sup>15</sup> The Wortschatz Leipzig also outputs a frequency class that reports the relation of a word's frequency to the most frequent word in the corpus. We pruned the list by removing rare words (high frequency class) and very ambiguous words. Table 1.1 gives an overview of the common German reporting verbs that the reporting verb detector uses in our quotation extraction approach. Analyzing a text the reporting verb detector checks for each word's lemma if it occurs in the list. The corresponding words are then treated as reporting verb candidates for the quotations to extract.

### 1.3.3.3 Direct Quotation Extraction

All quotations within quotation marks are regarded as direct quotes (quoted speech). The direct quote collector detects quotations employing pattern recognition and handcrafted rules. It first compiles a set of quotation candidates (text parts enclosed by quotation marks) and then applies the set of handcrafted rules to them to construct the final direct quote.

The applied pattern is composed of different combinations of left and right quotation marks which must enclose at least one character. In order to avoid the detection of single words or phrases that are emphasized with quotation marks, quotation candidates are discarded if they consist of less than four words. Furthermore, we check

<sup>15</sup> <http://wortschatz.uni-leipzig.de/>.

whether the quotation candidates contain a verb. Our investigations have shown that quoted phrases with less than four words and without a verb are in most cases simply highlighted text parts such as proper names. A direct quote may be composed of several quotation candidates. That is why the component examines each quotation candidate and decides whether it is the beginning of a new quotation or the part of a compound quotation. It searches the environment of the quotation candidate for incomplete sentences (sentences not ending with a *period*, *exclamation* or *question mark*) and reporting verbs. Incomplete preceding sentences are concatenated to the quotation candidate. If the preceding sentence has been completed the component checks whether it contains a reporting verb. Sentences with a reporting verb are concatenated to the direct quotation candidate, because our experiments have shown that these sentences often are reporting clauses that provide a quotation speaker. Sentences following a quotation candidate are processed in a similar way. If a sentence contains a reporting verb or is incomplete, it is concatenated to the quotation candidate. Subsequent sentence parts containing the word “so” are also attached. We cover in this way cases like “‘...’, so Angela Merkel”. Quotation candidates are connected to each other if a quotation candidate directly succeeds a reporting clause or a quotation candidate.

#### 1.3.3.4 Indirect Quotation Extraction

Reported speech is not put in quotation marks. It is composed of a main (reporting) and a subordinate (reported) clause. In German, the reported clause often is introduced by the conjunction “dass” and uses the subjunctive mood for verbs. In order to extract indirect quotations from a news article we apply a rule-based approach. The indirect quote extraction depends on the output of the direct quote collector. Therefore, the indirect quote collection must succeed the direct quote collection in our processing pipeline. Our approach is to first identify a reporting and a reported clause and then construct the final indirect quotation. The indirect quote collector exploits the occurrence of reporting verbs. To avoid duplicate quotation extraction (identifying quotations as direct and indirect) the collector exclusively regards reporting verbs that have not been already assigned to a direct quotation. If a detected reporting verb is not already part of a direct quotation, we assume that the verb indicates the reporting clause of an indirect quotation. We build up an indirect quotation by analyzing the surrounding sentences or sentence parts. A strong indicator for the reported clause is the presence of the conjunction “dass” (that) together with the finite verbs “sei, seien, habe, werde, würde, würden” that are usually used in reported clauses to repeat what someone has said. The occurrence of “dass” and one of the verbs implies a reported clause and we infer a quotation. The quotation encompasses the reporting and the reported clause. Sentences containing a reporting verb in the reporting clause but missing “dass” in the reported clause are treated in the same way, if they contain the finite verbs mentioned above. We also detect indirect quote indicated by ‘, so’ (as) and ‘, hieß es’ (it was said).



### 1.3.3.5 Quotation Holder Extraction

The aim of the quotation holder extraction is to attribute speakers to the identified direct and indirect quote. Our approach is based on the observation that quotation holders in most cases are named entities or references to named entities that are mentioned nearest the reporting verb. For example, we choose the pronoun ‘er’ (he) regarding the fragment ‘, sagte er dem Spiegel’ (, he said to ‘Der Spiegel’). To determine a quotation’s holder we first create a set of candidates. As candidates we consider named entities, pronouns (only “er” (he) and “sie” (she)) and noun chunks from the reporting clause. We exclude candidates originating from the reported clause. Then, we sort the list by proximity to the reporting verb but prioritize named entities and pronouns over noun chunks. Pronouns are still left in order with named entities, so that passages like “, sagte *er* zu *Angela Merkel*” ( *he* said to *Angela Merkel*) do not get assigned to the wrong holder. If no reporting verb has been assigned to the quotation we search for the word “so” in the reporting clause and sort the candidates according to how near they are placed to the word “so”. Concerning direct quotations there also may be quotations without a reporting verb and the word “so”, since they are detected with the aid of quotation marks. In this case we simply select the candidate nearest to the reported clause. Our approach to quotation holder extraction also includes a simple form of co-reference resolution. If we determine a person as quotation holder we attempt to resolve its name to the longest form of it in the text. If the assigned speaker is a pronoun then we choose the first named entity before the quotation.

### 1.3.4 Corpus

We manually annotated a corpus of 714 news articles containing direct and reported speech. The news articles are all in German and were published over a time period of three months from February 23, 2012 to May 21, 2012. The corpus allows the evaluation of determining quotation text boundaries and of recognizing reporting verbs and quotation holders.

For the annotation process we had to assure a sufficient coverage of direct and reported speech. That is why we preprocessed the news stream provided by Neofonie GmbH and preselected some news documents before we started with the annotation procedure. We automatically detected different types of quotation marks and a set of predefined reporting verbs within the news articles. Thereafter we randomly sampled 1,000 news articles. We chose:

- **250 news articles** containing at least one direct quotation (text passages identified by the occurrence of quotation marks and that are longer than 24 characters)
- **250 news articles** containing at least one of the following reporting verbs: “sagte”, “berichtete”, “berichteten”, “gestand”, “erklärte”, “erklärten”
- **500 news articles** without any restrictions.

### Zitat-Annotierung

Bitte lesen sie die [Anleitung](#) vor dem Annotieren!

Annotieren sie alle Zitatstellen im geeigneten Text mit den unteren Buttons. Beantworten sie bitte vor dem Speichern die untere Frage. Annotieren sie jedes neue Zitat einzeln. Wechseln sie zwischen Zitaten mit den Buttons [-] und [+].



Fig. 1.4 The annotation tool used for the creation of the quotation extraction corpus

We asked the annotators to identify all quotations in a news article and advised them to mark for each quotation the quoted text, the quotation holder, and a reporting verb if available. A screenshot of the annotation tool is shown in Fig. 1.4. For quotation holders not referenced by their proper name but by, e.g., a personal pronoun or only by the last name, the annotators should assign the full proper name if possible. If a quotation or a reporting verb was composed of several parts, the annotators were asked to mark all parts (**teilte** der Sprecher **mit**, *the spokesman said*). They were also advised to mark if a news article does not contain any quotes at all.

We succeeded in annotating 714 news articles. 339 of the news articles were annotated twice, 27 three times, and 2 even four times. The remaining 347 news articles were annotated by only one annotator. The annotators exactly agreed upon the quotations in 287 news articles. At that point we speak of exact agreement if the boundaries of the quotation holder, the reporting verb, and the quote text match accurately comparing the annotated tokens. Finally, the resulting corpus of 287 news articles contains 383 quotations, whereof 256 quotations are direct, 98 indirect, and 29 mixed (including at least a direct and indirect part) quotations. A news article contains 1.3 quotations in average. 87% of the quotations are attributed with a reporting verb. We succeeded in annotating a quotation holder for each quotation. For 202 quotation holders we could resolve the reference and assign proper names.

### 1.3.5 Evaluation

We evaluate our quotation extraction approach using a human-annotated corpus of 287 news articles where at least two annotators exactly agreed upon the contained

**Table 1.2** Results for the quotation extraction

	Reporting clause						Reported clause		
	Holder			Verb			P	R	F1
	P	R	F1	P	R	F1			
All quotations	0.801	0.649	0.717	0.932	0.728	0.817	0.862	0.821	0.841
Direct quotations	0.791	0.672	<b>0.727</b>	0.914	0.679	0.779	0.89	0.895	<b>0.892</b>
Indirect quotations	0.852	0.596	0.701	0.989	0.815	<b>0.893</b>	0.747	0.782	0.764
Mixed quotations	0.727	0.653	0.688	0.852	0.767	0.807	0.913	0.505	0.65

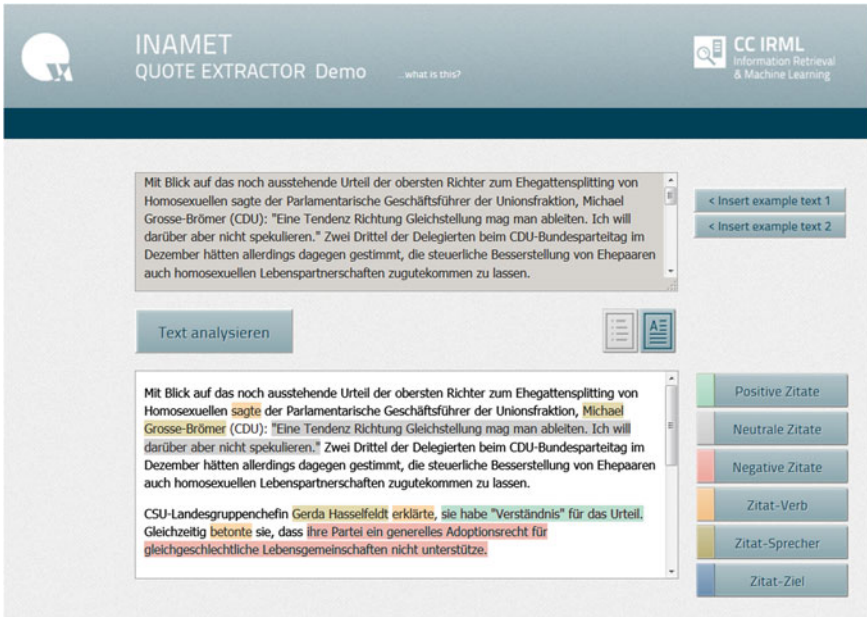
The direct quotation extraction performs best. Our component achieves an F1-score of 0.89 for the extraction of the reported clause and an F1-score of 0.73 for assigning a speaker. The extraction of indirect quotations is more difficult. Still, achieving an F1-score of 0.76 for the extraction of the reported clause, it produces reasonable results

quotations. In order to measure the performance of our approach we make use of standard information retrieval measures and compute a token-based recall, precision and F1-score. We regard all overlapping tokens as true positives. All missing tokens are regarded as false negatives and all unnecessarily annotated tokens as false positives. We then calculate the overall performance by summing up all intermediate results and by calculating final micro-averaged results for the subtasks of holder, verb, and reported clause extraction.

Table 1.2 summarizes the obtained results. The results meet our expectations. We achieve the best micro-averaged F1-score of 0.89 for the extraction of direct quotation parts. With an F1-score of 0.76 our approach for extracting reported speech performs less effective but still reasonable. Considering quotation holders, the proposed algorithm behaves comparably for all quotation types. It achieves an F1-score of 0.72. The detection of reporting verbs or clauses introducing a quotation performs quite well with an F1-score of 0.82. It is striking that the extraction of reported clauses of mixed quotations is most challenging. Here, our algorithm does not exceed an F1-score of 0.65.

In order to facilitate a manual evaluation of our extraction approach and also for its further refinement and improvement, we developed a web demonstrator that visualizes the results calculated by our component (Fig. 1.5). The upper field allows to insert an arbitrary text<sup>16</sup> which our quotation component analyzes subsequently. The demonstrator shows the identified quotations in the preview window below. The detected text spans are highlighted in specific colors. At a glance the user sees whether the extraction was successful or whether the algorithm provides erroneous annotations.

<sup>16</sup> Note that our approach is calibrated on news articles and could produce insufficient markup for text types other than news articles.



**Fig. 1.5** The web demonstrator for visualizing automatically extracted quotations. It allows to insert arbitrary text into the *upper* field which is then analyzed by our quotation extraction component. The results are highlighted in the area *below*. The users can choose which information the demonstrator should display and which should be hidden

### 1.3.6 Conclusion

We presented an approach to quotation extraction that includes the extraction of direct and indirect quotations and the assignment of a speaker to each quotation. Our approach is rule-based and relies on a handcrafted list of reporting verbs. The implemented rules are manually created as well. As valid speakers we allow text spans covering pronouns (she and he), noun phrases, and named entities. We resolve pronouns to appropriate named entities mentioned earlier in the text. The results achieved with our unsupervised approach compare favorably with other approaches, and do not rely on the availability of training data. Especially the extraction of direct quotations and attributing them to a speaker already works very satisfactorily. Regarding indirect quotations, finding the boundaries of the reported clause and the correct quotation holder is more challenging, regarding mixed quotations even more. Since our approach to indirect quotation extraction relies on a list of reporting verbs and clues, potential results are limited to those parts near such predefined reporting verbs or clues. Thus, our future work includes among others, the extension of our approach by an automatic reporting verb recognizer [38]. We plan to detect previously unseen reporting verbs as well as the disambiguation of verbs. For example, the ambiguous verb “to add” may lead to a mistake by regarding it as a reporting

speech indicator in a wrong context. By automatically determining reporting verbs and phrases we aim to improve the recall of indirect quotations. We also intend to complement our co-reference resolution approach with a state-of-the-art component. We want to determine co-reference chains and then choose the correct one as quotation speaker [39]. In order to consolidate quotations uttered by one speaker across different news documents, we plan to link speakers to Wikipedia entries by applying a named entity disambiguation approach. Our future work will also include the incorporation of supervised approaches. On the one hand we plan to treat quotation extraction as a sequence labeling task and on the other hand we plan to train a binary classifier that predicts quotations at sentence level. Our goal is to identify appropriate features for German-language texts and to let an ensemble combine the output of the rule-based approach with the output of the new classifiers.

## 1.4 Sentiment Analysis

Publicly available texts such as product reviews, social media contributions, or news articles discuss almost every thinkable entity and topic. Besides transporting facts, the texts often cover opinions as well, and even more than facts, the expressed opinions may influence readers. Regardless of whether someone wants to buy a new camera or wants to find out which party to vote in the next election, people in general read first what other people think and what experiences they have had before making their own decisions. That is the reason why companies are interested in a positive perception of their products and services in the media. Here, well-analyzed opinionated texts may serve as a basis for a multitude of sentiment-related applications like reputation monitoring or opinion summarization systems. Sentiment analysis may also be performed to improve other natural language processing tasks that rely on factual data. Separating opinionated text from objective text turned out to be beneficial for information extraction [43]. In this section we show how opinions can be extracted from news articles. We focus our work on news articles because they are a reliable information source and mirror the opinions of newsworthy people, which often serve as role models. We first introduce the term “sentiment analysis” and “opinion mining” and then present our comprehension of “opinions”. The main part describes our supervised approach to sentiment analysis. We limit our approach to quotations because we assume that quotations are the most subjective parts of news articles.

### 1.4.1 Introduction

Sentiment analysis aims at identifying subjective language in texts and determining the orientation and strength of expressed opinions or sentiments toward the corresponding targets. Often the term “opinion mining” denotes the same task and is

used synonymously [36]. The task of sentiment analysis may be decomposed in subjectivity detection and polarity (or orientation) classification. The goal of subjectivity detection is to distinguish objective from opinion-oriented text. While objective parts solely report facts without any personal assessment or evaluation, opinionated text parts may contain any type of subjective expressions that reflect the private state of a holder. This includes personal attitudes, views, statements, feelings, etc., expressed by the text’s author or by other people mentioned or cited in the text. Having detected a subjective text part, the type and orientation of the expressed opinion must be determined. The most common orientations are *positive* and *negative*. For example, a literature critic may conclude that a reviewed book is excellent, which can be regarded as a positive opinion toward the book. Other classification schemes distinguish between supporting and opposing expressions. This variant of sentiment classification allows to contrast different points of view toward topics or political issues.

Our definition of opinion is driven by Liu and Zhang [28]. Following the authors, we define opinion as a quintuple consisting of a

- **target entity** (e.g., a product, an individual, a topic)
- **target aspect of the entity** (e.g., product features, subtopic)
- **orientation** (e.g., positive/negative/neutral)
- **holder** (the entity holding the opinion)
- **time** (when the opinion was expressed)

In order to gain a valuable opinion information, not all parts of the quintuple must necessarily be extracted. The perception of a movie in the media, e.g., can be inferred without knowing when the review was submitted or by whom. However, the extraction of a target entity and the valuation of the entity is essential. The opinion target may be a mentioned (named) entity or concept with a concrete text-anchor, but also an abstract topic that makes the identification of the opinion target especially difficult. In contrast to concrete entities, abstract topics are nonlocal information that have to be mined from the context. In our scenario, which is sentiment analysis based on newspaper quotations, the structure of quotations already provides an opinion holder. Knowing who uttered a quotation, we consider the quotation holder being also the opinion holder.

**Our Contribution.** We present work that aims at determining opinion orientation in German news articles. In contrast to many other sentiment analysis systems we focus our approach on direct and reported speech identified previously in the news articles. We assume that quotations are the most subjective parts of news articles and that they transport the opinions of the cited speakers as they are. We cast the task of sentiment classification to a three-class problem and label each quotation as either negative, positive, or neutral. Objective quotations reflecting facts are marked as neutral as well. We propose a supervised approach where we first search for subjective quotations and, second, decide the polarity of the subjective quotations. Both steps are separately solved by a Support Vector Machine classifier. As part of our work we examine the effectiveness of diverse sentiment classification features

to find the most suitable feature set for both the subjectivity detection and polarity classification task. We train and evaluate our approach on a human-annotated corpus of German quotations. The corpus consists of 742 neutral, 71 positive, and 38 negative quotations. It can be made available for research purposes after signing an agreement.

### ***1.4.2 Related Work***

Related research on sentiment analysis varies from simple lexicon-based approaches looking up words in opinion lexicons to supervised approaches exploiting linguistic features and enhanced machine learning algorithms. The main part concentrates on the classification of text as either POSITIVE, NEGATIVE or NEUTRAL toward a specific entity or topic explicitly mentioned in the text.

Sentiment analysis treats texts at different levels. There is work examining entire document texts like entire reviews or news articles, attempting to predict the overall sentiment of a document text [37, 49]. But there is also work that performs sentiment analysis at statement level [5, 18, 46], sentence level [23, 33, 43, 54] or even phrase level [49]. Often, sentiment analysis work on reviews also aims at extracting product properties and the opinions toward these properties, which is called aspect-oriented sentiment analysis [21].

Sentiment analysis has also been applied to different text types. A great part of the work examines customer reviews, like product [21, 49] or movie [37] reviews. Since reviews are meant to share experiences and report opinions, they contain many subjective text parts and are therefore predestined for sentiment analysis. Yet, reviews can also contain objective parts summarizing the properties of the reviewed entities. Regarding movie reviews, one challenge is to separate plot information, which itself may be characterized as positive or negative, from opinions toward the movie. All work treating customer reviews must handle challenges arising from user-generated content such as potential spelling mistakes and grammatical errors.

Early work in classifying product reviews used lexicon-based techniques together with natural language processing algorithms in order to create opinion summarization. Hu and Liu [21] propose a three-stage approach to aspect-based opinion summarization. They first search for product features in customer reviews by applying association mining with some pruning. Then, the authors determine the polarity of sentences mentioning the features. Whether a sentence has to be classified as positive or negative results from the orientation of the individual opinions words (adjectives) in the sentence that is summed up to an overall orientation. The orientation of opinion words is pre-calculated based on a list of seed adjectives and the application of WordNet's information on synonyms and antonyms. Similar to Hu and Liu, Turney [49] categorizes product reviews in either 'recommended' or not 'recommended' by calculating the average sentiment orientation of the review's phrases. Turney calculates the orientation of phrases containing adjectives and adverbs by determining the mutual information between a phrase and the words "excellent" and "poor" and subtracting both values to obtain a final sentiment orientation score.

The so far discussed customer reviews are predominantly medium or long texts. With the mass distribution and utilization of social media services like Twitter or Facebook in the recent years, a part of the sentiment analysis work shifts toward the analysis of short texts generated by users. Because of the language used in such texts, new challenges arise for the task. Often, users write their texts colloquially and they do not care about spelling and punctuation. In addition, the texts mostly are very short and comprise phrases rather than complete sentences. Considering Twitter, a short message must not exceed 140 characters.

As one of the first, Go et al. [18] classify English-language tweets according to a query as either positive or negative. They adopt a supervised approach using diverse classifiers including a Naive Bayes, a Maximum Entropy, and a Support Vector Machine (SVM) classifier. In order to train the classifiers, the authors propose using tweets containing positive or negative emoticons (mapped to ‘:(’ and ‘:’)’) as noisy labeled training data. The authors explore a range of standard text classification features like word uni- and bigrams and part-of-speech tags for representing the tweets. After having evaluated their approach on manually tagged tweets from different categories (177 negative and 182 positive tweets independent of emoticons), Go et al. conclude that the automatically created training dataset is suitable for training the examined algorithms, which solve the task reasonably. Using a combination of word uni- and bigrams the Maximum Entropy classifier achieves an accuracy of 83 %. Yet, there are no large differences between the classifiers and feature sets.

In comparison to customer reviews, news articles may express opinions less explicitly. Since journalists (ought to) write objectively and avoid emotional language, the identification of the implied opinions is especially challenging. In addition, the opinion holder must be extracted. Different from customer reviews, it is not the author’s opinion expressed in the news article but the opinion of other people and organizations the article deals with. In 2006 Kim and Hovy [23] approached the task of opinion mining in English news articles by proposing a four-stage system. The authors extract opinions, determine the opinion topic, and assign an opinion holder by applying semantic role labeling. The authors separate subjective from objective sentences, perform semantic role labeling utilizing opinion-related frames and frame elements from FrameNet,<sup>17</sup> and choose the opinion target and holder out of the semantic roles. Finally, the extracted opinion triples consisting of the holder, topic, and opinion are stored in a database.

The work proposed by Nakagawa et. al [33] addresses sentiment classification at sentence level. The authors use conditional random fields with hidden variables, representing polarity of dependency sub-trees, to infer the polarity of the entire subjective sentences. The approach was evaluated on English and Japanese opinion texts and is promising. Among others, it was evaluated on Japanese news articles with an accuracy of 83 %, which shows its effectiveness on this text type. However, the work bases on subjective sentences and skips the task of subjectivity detection.

Strongly related to our work is the work of Balahur et al. [2–4]. The authors apply sentiment analysis to news articles. Although the team mainly explores approaches

---

<sup>17</sup> <https://framenet.icsi.berkeley.edu/fndrupal/>.



to classify English news quotations, it is an important requirement to develop approaches that are applicable with as little effort to other languages, since the authors incorporate their results into the Europe Media Monitor (EMM) news engine.<sup>18</sup> EMM collects and processes news articles from multilingual news sources. In [2] Balahur and Steinberger present reflections on how sentiment analysis applied to news articles differs from sentiment analysis on highly subjective texts like online reviews. The authors find that the task of sentiment analysis on news articles can be decomposed into three subtasks: determining the sentiment target, distinguishing between “good and bad news content” and “good and bad sentiment expressed on the target”, and classifying explicitly expressed sentiments on the sentiment target that does not require any world knowledge or the interpretation by the reader. The inter-annotator agreement increases if the task is clearly defined in advance. Finally, the authors work out three possible perspectives on news articles: the author’s (news bias research), the reader’s (interpretation by readers influenced by their backgrounds), and the text’s view. Each view requires a different approach to sentiment analysis and the authors limit their work to identifying sentiments concretely expressed in the text. In [3] Balahur et al. provide a comparison of different sentiment resources and classification strategies to categorize news quotations as positive, negative, and neutral. In their studies, the authors also examine how a preceding subjectivity detection step affects the classification results. They conclude that using large sentiment lexicon and a previous subjectivity filtering improves the results considering vocabulary-based methods. Straightforward bag-of-words approaches are limited and not effective enough for sentiment analysis on news quotations. The authors also conclude that exploiting sentiment annotations based on single topics are not suitable for the open-domain sentiment analysis on news. Thus, they propose a topic-dependent sentiment analysis with specialized models.

The work of Balahur et. al in [4] analyzes two aspects of sentiment analysis for English-language quotations in news articles. First, the authors examine how different word windows around an opinion target influence sentiment classification accuracy. Second, they exclude sentiment-bearing words that are category-specific words at the same time, in order to separate good or bad news content from positive and negative sentiments toward the opinion targets. The sentiment score is calculated by summing up the sentiment scores of all quotation words. As a result, the authors argue that taking into account only a word window around the target entity instead of including the entire quotation text yields better results. Considering the lexicons, the authors find that there are large differences between their performances and that combining them helps. However, the accuracy of the approach does not only depend on a large lexicon.

Sentiment analysis on German-language texts has been applied in [27, 31, 46]. Momtazi presents a rule-based approach to classify sentiments toward celebrities mentioned in short German social media texts. In order to label the short texts as positive or negative and assign the strength of the sentiment, the author creates and applies a sentiment dictionary and a list of booster and negation words. Mom-

---

<sup>18</sup> <http://emm.newsbrief.eu/overview.html>.

tazi evaluates her approach on a hand-annotated dataset of 500 short texts about celebrities. Her unsupervised polarity classification method outperforms standard supervised classifiers on the given dataset. The main contribution of Li et al. [27] is an annotation scheme for labeling sentiments in German political news articles and a dataset manually annotated according to the presented scheme. Each annotation frame consists of the text anchor (labeled as idiom, phrase, word or compound noun), the target, the source, and auxiliary words that may be intensifiers, diminishers, or negations. In addition, the opinion frames may be marked with the attitude's polarity, type (context-dependent or -independent), and intensity. With the aid of the relation extraction tool DARE<sup>19</sup> and the annotated relation examples, rules are automatically learned to extract the opinion source, target, and polarity. In their first experiments the authors achieved promising results. Another corpus for German sentiment analysis on news articles is contributed by Scholz et al. [46]. The corpus consists of around 1,500 statements labeled with the viewpoint (corresponding to our "opinion target"), either CDU<sup>20</sup> or SPD,<sup>21</sup> and the tonality of the statement (positive, neutral, negative). The authors use parts of the dataset to generate sentiment dictionaries containing entries scored with different measures. For the sake of media response analysis, the authors evaluate news material and propose a supervised machine learning approach which is similar to ours. Scholz et al. analyze the news in two stages. First, they detect subjective statements and, second, they classify the subjective statements as either positive or negative. In contrast to our findings, the subjectivity detection seems to perform better on the dataset of Scholz et. al. than the polarity classification part.

Detailed surveys on opinion mining and sentiment analysis can be found in [28, 36, 48].

### 1.4.3 Sentiment Classification

We solve the problem of quotation sentiment classification by employing a supervised two-stage approach. We first apply a subjectivity detection step where we mark quotations as either neutral or subjective. We then classify all subjective quotations according to their polarity in either positive or negative quotations. As a result of our sentiment classification approach each processed quotation is labeled as either neutral, positive, or negative. For both tasks, subjectivity and polarity classification, we train separate Support Vector Machine (SVM) classifiers [8, 50] with a different feature set and with different hyperparameters. We choose a radial basis kernel for both SVMs and select the hyperparameters  $\gamma$  and  $C$  by performing tenfold cross-validation on the dataset described in Sect. 1.4.5. We represent the quotations as vectors of diverse features (Sect. 1.4.4). Among others we include the part-of-speech tags and sentiment words as features that turned out to be essential for sentiment

---

<sup>19</sup> <http://dare.dfki.de/>.

<sup>20</sup> Christlich Demokratische Union Deutschlands (Christian Democratic Union (Germany)).

<sup>21</sup> Sozialdemokratische Partei Deutschlands (Social Democratic Party of Germany).

analysis. We weight the features by different weighting schemes ranging from simple counts to enhanced weighting schemes like tf-idf and tf-delta-idf, a sentiment-based tf-idf value, as proposed in [30].

**Opinion Target Extraction.** We also provide a supervised approach for the extraction of opinion targets, which may be reasonably applicable as long as the targets are explicitly mentioned within the quotations and can be localized by text anchors. In our target extraction approach we first select a set of candidates and then classify each candidate with a binary classifier to predict whether it is the wanted target or not. The opinion target candidates are represented as feature vectors of POS tags surrounding each candidate in a window of two words before and two words after the candidate. We perform a multistage decision process to prefer specific candidates over other candidates if more than one candidate was classified as opinion target. We check the environment nearby the candidate and accept only candidates conforming specific predefined POS patterns.

#### 1.4.4 Sentiment Features

Finding an appropriate representation of the data at hand is a crucial task since the performance not only depends on the chosen machine learning algorithms but also to a large extent on the selected features [12]. In this section we present the features explored for our sentiment analysis approach. Besides primitive features we also exploit derived lexical and linguistic features. Following [37] we include position information for each feature. We encode whether the features were calculated based on the beginning, the end, or the middle part of the text or whether the entire text was considered.

**Bag-of-Words.** A standard representation of documents for natural language processing tasks is the bag-of-words model [29]. It represents a document as a vector of weighted terms from a dictionary. We built up a dictionary with uni- and bigrams and calculated the idf and delta-idf based on German news articles from a time period of three months of 2012, the same time period as used for creating the evaluation corpus described in Sect. 1.4.5. The lexicons with both uni- and bigrams were limited to 10,000 entries each. We included bigrams because they encode word order, which adds meaningful sentence structure information to the feature vector representation. Previous work shows that bigrams help in the task of sentiment analysis [51]. In order to compile a feature vector for a document we remove stop words and lower-case and stem each term using Apache's German Analyzer.<sup>22</sup> Then we weight each term (uni- and bigram) using one of four different schemes: occurrence flag (0/1), tf (term frequency), tf-idf (term frequency x inverse document frequency), tf-delta-idf (term frequency x delta inverse document frequency).

---

<sup>22</sup> [https://lucene.apache.org/core/3\\_6\\_2/api/all/org/apache/lucene/analysis/de/GermanAnalyzer.html](https://lucene.apache.org/core/3_6_2/api/all/org/apache/lucene/analysis/de/GermanAnalyzer.html).

**Parts-of-Speech.** Part-of-speech tags (POS tags) serve as the basis for various natural language processing tasks. Especially, in sentiment analysis the part of speech information is widely exploited. The presence of certain parts of speech like adjectives [19] and word phrases corresponding to certain part-of-speech patterns [49] correlates with opinion-oriented language and is therefore a strong predictor of subjectivity [36]. We assembled a dictionary of POS tag uni- and bigrams and calculated the idf and delta-idf analogously to the dictionary of term uni-and bigrams. Based on this dictionary we form the feature vector as a bag-of-pos-tags. Again, we assign the following values to each items: occurrence flag, tf, tf-idf, tf-delta-idf.

**Sentiment Words.** Sentiment lexicons like SentiWordNet [14], WordNet-Affect [47] or MicroWNOp [7] often are a central resource for many sentiment analysis approaches [3]. Sentiment bearing words have been applied for lexicon-based solutions [4] and as features for supervised classification approaches. For German the SentimentWortschatz (SentiWS) has been well established [42]. SentiWS is a lexicon covering affective words along with weights for their polarity that range between  $-1$  (very negative) and  $+1$  (very positive). In addition to the sentiment value, the part of speech and a set of inflections is available for each entry. SentiWS contains 1,650 positive and 1,818 negative adjectives, adverbs, nouns, and verbs. We use SentiWS for compiling term and aggregated features. First, we represent a quotation as a vector of SentiWS terms with either an occurrence flag, the term frequency, or the term frequency multiplied with the polarity weight of the term in SentiWS. Second, we aggregate features by grouping positive, negative, or all SentiWS words and count and weight their occurrences. Given quotations like “*Das ist **nicht gut**”*, *sagte Angela Merkel.* (“That’s **not good**”, Angela Merkel said.) we must consider negation to avoid erroneous feature values. Simply regarding SentiWS word’s polarity value as it is, we would assign a positive value to the term ‘gut’ and label the quotation probably as POSITIVE although a negative sentiment is expressed. For a more precise feature calculation we again make use of POS information and identify five POS tag patterns starting with PTKNEG (negation particle ‘nicht’ (not)) or PIAT (attributive indefinite pronoun without determiner ‘kein’ (no)), both inverting the polarity value of the following word. Table 1.3 shows the set of five patterns capturing phrases commonly used in German for negating adjectives, verbs, and nouns. We apply the

**Table 1.3** The list of POS patterns used for shifting polarity weights of quotation terms

POS tag pattern	Examples
PTKNEG ADJD	Nicht richtig (not correct), nichtüberzeugend (not convincing)
PIAT NN	Kein Gegner (no opponent), kein problem (no problem)
PTKNEG VVPP	Nicht gelungen (not succeeded), nicht vorbestraft (not previously convicted)
PTKNEG VVINF	Nicht tolerieren (not tolerating), nicht bessern (not becoming better)
PIAT ADJA	Kein gutes (no good), kein unzumutbares (no unacceptable)

The examples are taken from our evaluation corpus described in Sect. 1.4.5

patterns to the quotation’s text and augment our feature vector by adding a negated form of the SentiWS term in form of NOT\_SENTIWS\_TERM to the vector. We also invert the SentiWS value for the negated term, if the weighting scheme requires this. In the example above we add NOT\_GUT (not good) to our term vector and regard the term as negative for calculating our aggregated features.

**Valence Shifters.** Valence shifters are words or phrases like “nicht” (not), “höchst” (extremely), or “weniger” (less), that change the intensity or polarity of other lexical items. We distinguish between three types of valence shifters: negations, diminishers and intensifiers. Previous work examined the effect of valence shifters in sentiment classification of movie reviews and concluded that incorporating valence shifters slightly increases classification accuracy [22]. In order to create our features, we exploit a list of around 100 valence shifters derived from the MLSA Corpus, a multi-layered reference corpus for German-language sentiment analysis [9]. The corpus consists of three layers with sentiment annotations at different granularity levels. Layer 2 provides polarity related annotations for words and phrases. At phrase-level the text spans are labeled as positive, negative, bipolar, and neutral. Words are labeled in addition as diminishers, intensifiers, and shifters (negations). With the aid of these annotations we compile feature vectors in the form of bag-of-valence-shifters and derived features that accumulate the three types of valence shifters.

**Discourse Markers.** Discourse markers are words or phrases that connect sentences or sentence parts and thereby express the semantic relations between them. Examples are “weil, aber, abgesehen davon dass, sogar, dennoch...” (because, but, apart from this, even, however...). The usage of discourse markers may influence the orientation or intensity of sentiments like in the quotation “*Wir sind zufrieden mit dem Stand der Dinge, **aber** wir wollen mehr*”, sagte Vettel. (“We are happy with the situation, but we want more”, Vettel said.) In our approach we search quotations for discourse markers from a predefined list. The list is derived from the online lexicon for German grammar<sup>23</sup> of the “Institut für Deutsche Sprache”<sup>24</sup> (IDS, Institute for German Language). It contains around 350 discourse markers of different types. The resulting feature vector encompasses all discourse markers making no distinction between the types. We assign each marker a value (occurrence flag and term frequency) and encode whether the quotation contains discourse markers and how many.

**All Features.** Table 1.4 provides an overview of all feature groups that we use in our sentiment analysis approach. A feature vector containing all feature combinations consists of 160K entries. If we include text position information into the feature vector the number of entries rises to almost 650K. The relative big dictionaries, in comparison to the short quotations, result in very sparse feature vectors that we have to deal with.

---

<sup>23</sup> <http://hypermedia.ids-mannheim.de/index.html>.

<sup>24</sup> <http://www1.ids-mannheim.de/start/>.

**Table 1.4** Overview of the sentiment analysis features and values

Feature Name	Feature Value
Bow-of-Words (uni- and bigrams)	Occurrence flag, tf, tf-idf, tf-delta-idf
Parts-of-Speech (uni- and bigrams)	Occurrence flag, tf, tf-idf, tf-delta-idf
SentiWS words	Occurrence flag, tf, tf x polarity-value
SentiWS pos/neg/all	Occurrence, count, count x polarity-value
Valence shifters	Occurrence flag, tf
Valence diminishers/intensifiers/shifters	Occurrence, count
Discourse markers	Occurrence flag, tf

### 1.4.5 Corpus

The corpus for evaluating our sentiment analysis approach consists of 851 quotations extracted from a dataset of German news articles dated from February 23, 2012 to May 21, 2012. The manually annotated quotations of the quotation extraction corpus described in Sect. 1.3.4 served as the basis for the sentiment corpus. We asked four annotators to tag each quotation as either NEUTRAL, POSITIVE, NEGATIVE, or MIXED and if possible to mark the opinion target.<sup>25</sup> In order to obtain a consistent corpus we defined a set of annotation rules, which we describe in detail below. In general, personal attitudes of the annotators, their moral perceptions, or political views must not influence the tagging. The task was to determine the opinion of the quotation speakers. If a quotation appeared to be incomplete, the annotators also had the possibility to tag a quotation as DON'T KNOW.

#### 1.4.5.1 Neutral Quotations

A quotation should be regarded as neutral if the statement serves solely to transport facts. Information, announcements, or intentions of the speaker without any personal assessment by the speaker are considered to be neutral. It is not relevant whether the fact itself is positive or negative from a moral, political, or any other point of view. In the remainder of this section, we provide examples of different types of quotations. The content of the quotation in Example 1.1 solely reports fact:

*Example 1.1* Alisade berichtete, dass bald 500 weitere KFC-Filialen landesweit eröffnen würden. (Alisade reported, that soon 500 more KFC-stores would open countrywide.)

<sup>25</sup> We do not use the annotated opinion targets yet, but describe them here for the sake of completeness.

The speaker announces an event without any personal assessment, so that the quotations in Example 1.2 has to be marked as neutral as well:

*Example 1.2* **Es werde Sicherheitskontrollen an den Einlässen geben**, sagte ein Sprecher. (There will be security checks at the entries, a speaker said.)

### 1.4.5.2 Subjective Quotations

Subjective quotations could be marked as POSITIVE, NEGATIVE, or MIXED. To make their decision the annotators should answer the question whether the speaker supports or dislikes the topic or the expressed intention or announcement. In the following quotation the speaker explicitly expresses a negative opinion by using the term “Unverschämtheit” (impertinence). Annotators should rate it as NEGATIVE:

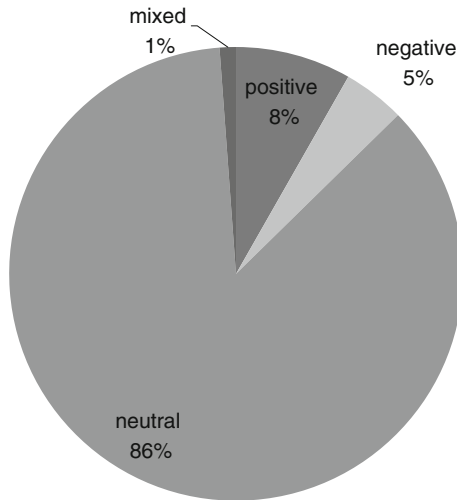
*Example 1.3* **“Der Fakt alleine ist eine absolute Unverschämtheit gegenüber dem Klub und dem Team”**, sagte Horstmann. (“The fact alone is an absolute impertinence toward the club and the team”, Horstmann said.)

The phrase “verdient gewonnen” (corresponds to “deserved to win”) indicates a positive opinion of the speaker in the following quotation so that this quotation should be classified as POSITIVE:

*Example 1.4* Claus Finger war zufrieden: **“Das Team hat schnell ins Spiel gefunden und verdient gewonnen.”** (Claus Finger was happy: “The team quickly got into the game and deserved to win”.)

### 1.4.5.3 Corpus Overview

The final corpus exclusively contains quotations annotated by at least two annotators. The gold standard answers were determined by majority voting. We discarded quotations where the annotators predominantly disagreed or where the majority of the annotators marked a quotation as DON'T KNOW. The inter-annotator agreement amounts to 79 %. Figure 1.6 shows the distribution of NEUTRAL, POSITIVE, NEGATIVE, and MIXED quotations. The majority of the quotations, namely 86 %, are NEUTRAL, whereas only 8 % of the quotation are POSITIVE and only 5 % NEGATIVE. Thus, we



**Fig. 1.6** The corpus for evaluating the sentiment analysis approach is highly unbalanced. It consists of 742 neutral, 71 positive, 38 negative, and 10 mixed quotations

have to deal with a highly unbalanced corpus. We discard the 1% MIXED quotations, because we do not aim at the classification of such quotations.

### 1.4.6 Evaluation

We conduct our experiments on a human-annotated corpus of 851 quotations tagged as POSITIVE, NEGATIVE, or NEUTRAL. We first evaluate each classifier of our two-stage approach separately and then assess the performance of the overall sentiment classification. In our experiments we first examine our sentiment features individually and then if combining them helps to solve the task of subjectivity and polarity classification.<sup>26</sup> We measure the effectiveness of our approach according to the precision, recall, and harmonic mean between precision and recall, the F1-score. We consider all classes equally important, determine the evaluation scores for each class separately, and then macro-average the scores across the classes. Our evaluation is performed as tenfold cross-validation where 90% of the data is used to train the classifier and the remaining 10% to test it in each evaluation run. Within the 10 folds the distribution of quotations is pertained. We normalize the feature values to fit into the interval of [0, 1]. In each run we perform a nested tenfold cross-validation to find

<sup>26</sup> We skip the evaluation of our target extraction solution because of the lack of text anchors for targets in our corpus. The corpus contains only a few target annotations because in most cases the targets are abstract topics or expression of sentiments rather than entities or nouns and therefore the annotators could not mark them within the quotation.



**Table 1.5** Results for the subjectivity classification part

	Neutral			Subjective			Macro-averaged			Accuracy
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	
NB baseline (all)	0.914	0.856	0.884	0.314	0.450	0.370	0.614	0.653	0.627	0.804
All	0.921	0.925	0.923	0.472	0.459	0.465	0.696	0.692	0.694	0.865
All-bow	0.918	0.923	0.921	0.457	0.440	0.449	0.688	0.682	0.685	0.861
All-postags	0.924	0.903	0.913	0.429	0.495	0.460	0.676	0.699	0.687	0.851
all-bow-postags	0.925	0.942	0.933	0.547	0.477	0.510	0.736	0.710	<b>0.722</b>	0.883
Bow	0.888	0.973	0.929	0.474	0.165	0.245	0.681	0.569	0.587	0.870
Discourse markers	0.881	0.725	0.795	0.150	0.330	0.206	0.515	0.528	0.501	0.675
Postags	0.906	0.867	0.886	0.298	0.385	0.336	0.602	0.626	0.611	0.805
Sentiws	0.918	0.939	0.929	0.511	0.431	0.468	0.715	0.685	<b>0.698</b>	0.874
Valence shifters	0.874	0.795	0.833	0.136	0.220	0.168	0.505	0.508	0.501	0.722

Isolated, the SentiWS features are most suitable for subjectivity classification with a SVM. Our approach outperforms the Naive Bayes baseline. The best performing feature set achieves an F1-score of 0.72. It includes SentiWS terms, discourse marker and valence shifters

the best hyperparameter  $\gamma$  and C for our SVM by employing a grid search. As our corpus is highly unbalanced we set the penalty for class SUBJECTIVE 7 times larger than for class NEUTRAL and the penalty for class POSITIVE 2 times larger than for class NEGATIVE.

**Subjectivity Classification.** For the assessment of our subjectivity classifier we make use of all 851 annotated quotations. We consider all quotations tagged as positive or negative being SUBJECTIVE. By doing so we prepare a corpus of 109 subjective and 742 neutral quotations. The first experiments evaluate our sentiment features individually to examine their impact on the subjectivity classification task. Table 1.5 shows the results. Isolated, the SentiWS features achieve the best F1-score of 0.698. We determine the best-performing feature set, containing the SentiWS term, valence shifter, and discourse-marker-based features, by conducting a feature ablation study. We leave out one feature type in each experiment that does not improve or even worsens the classification result. Using the best-performing features we achieve a *macro-averaged F1-score of 0.72* and succeed in improving the F1-score by 0.095 over the Naive Bayes baseline and by 0.024 over the F1-score exploiting solely the SentiWS features. Regarding the classes NEUTRAL and SUBJECTIVE retrieving neutral quotations works notably better than the retrieval of subjective quotations.

**Polarity Classification.** We evaluate our polarity classification approach on the 109 subjective quotations of the entire sentiment corpus. As with subjectivity classification the most relevant features for polarity classification are the SentiWS features (Table 1.6). Including only SentiWS features our approach already achieves an F1-score of 0.82. We are able to improve our results by adding POS tags and discourse markers as features. Together, the three feature types achieve a *macro-averaged F1-score of 0.86*. The score is 0.062 higher than the F1-score that results by including

all features into the set and 0.169 higher than the F1-score achieved by the Naive Bayes baseline with all features. We find that polarity classification is more difficult for the NEGATIVE class.

**Sentiment Classification.** Our sentiment classification approach aims to solve a three-class classification problem. We evaluate our overall approach by putting together the results of our two SVM classifiers. Starting with the results of the subjectivity classifier we pass all quotations classified as subjective to our polarity classifier for further separation into POSITIVE and NEGATIVE. The overall sentiment classification results in a macro-average *F1-score* of 0.51. Remember, the dataset is unbalanced and contains 86% neutral citations. We also conduct experiments providing Gold Standard answers for one of the tasks in order to evaluate the impact of the second classifier on the performance of the overall system. First, we simulate the output of the subjectivity classifier by taking the Gold Standard answers as input of the polarity classifier (Table 1.7, Gold Subjectivity Answers + Polarity Classification) and, second, assign Gold Standard answers to quotations marked as subjective by the subjectivity classifier (Table 1.7, Subjectivity Classification + Gold Polarity Answers). We then measure the effect of each classifier when integrated in an optimal system. With Gold Subjectivity answers our overall system achieves a *F1-score* of 0.86. Looking at it the other way round, using the subjectivity classifier’s output and the Gold polarity answers, our system achieves a *F1-score* of 0.61, which is around 0.1 higher than the overall system result but around 0.25 lower than the system grounded on the Gold subjectivity answers. These results correlate with the results obtained when testing the classifiers separately. The main error source is the subjectivity classifier.

**Table 1.6** Results for the polarity classification part

	Positive			Negative			Macro-averaged			Accuracy
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	
NB baseline (all)	0.763	0.859	0.808	0.655	0.500	0.567	0.709	0.680	0.688	0.734
All	0.825	0.930	0.874	0.828	0.632	0.716	0.826	0.781	0.795	0.826
All-bow	0.889	0.901	0.895	0.811	0.790	0.800	0.850	0.845	0.848	0.862
All-bow-valence	0.890	0.916	0.903	0.833	0.790	0.811	0.862	0.853	<b>0.857</b>	0.872
All-bow-valence-postags	0.863	0.887	0.875	0.778	0.737	0.757	0.820	0.812	0.816	0.835
Bow	0.693	0.986	0.814	0.875	0.184	0.304	0.784	0.585	0.559	0.706
Discourse markers	0.663	0.775	0.714	0.385	0.263	0.313	0.524	0.519	0.513	0.596
Postags	0.743	0.775	0.759	0.543	0.500	0.521	0.643	0.637	0.640	0.679
Sentiws	0.863	0.887	0.875	0.778	0.737	0.757	0.820	0.812	<b>0.816</b>	0.835
Valence shifters	0.711	0.831	0.766	0.539	0.368	0.438	0.625	0.600	0.602	0.670

Our polarity SVM classifier outperforms the Naive Bayes baseline with an F1-score of 0.86 achieved on all 109 subjective quotations in our corpus. It uses a feature set consisting of SentiWS terms, POS tags and discourse markers. As with subjectivity classification the most appropriate features are the SentiWS features

**Table 1.7** Results for the overall sentiment classification

	Gold subj. answers + Pol. classification			Subj. classification + Gold pol. answers			Subj. classification + Pol. classification		
	P	R	F1	P	R	F1	P	R	F1
	Positive	0.849	0.873	0.861	1.0	0.521	0.685	0.642	0.479
Negative	0.75	0.711	0.730	1.0	0.105	0.191	0.077	0.053	0.063
Neutral	1.0	1.0	1.0	0.916	1.0	0.956	0.912	0.949	0.93
Macro-avg	0.866	0.861	<b>0.864</b>	0.972	0.542	<b>0.611</b>	0.543	0.493	<b>0.514</b>
Accuracy	0.977			0.920			0.870		

Our approach achieves a macro-averaged F1-score of 0.51. While the polarity classifier performs reasonable, the subjectivity classifier introduces a large error. Many negative quotations are marked as neutral and therefore are not further examined by the polarity classifier. Given correct subjectivity labels the overall performance rises to an F1-score of 0.86

### 1.4.7 Conclusion

We solve the problem of sentiment classification of quotations in news articles by employing a two-stage approach where we first separate subjective from neutral quotations and, second, categorize the subjective quotations as either positive or negative. Our approach performs the best for both tasks with only a subset of the presented sentiment features. In either case SentiWS features strongly contribute to an efficient sentiment classification. Leaving them out decreases the F1-score considerably. In contrast to the SentiWS features, leaving out simple bag-of-word features (uni- and bigrams) increases the classification quality so that we exclude them from the final feature sets. The relatively low overall F1-score of 0.51 mainly results from the output of the subjectivity classifier. The subjectivity classifier introduces a large error in the first step. It misses many subjective quotations which the polarity classifier would tag correctly. Particularly, the majority of negative quotations is filtered out by the subjectivity classifier. Generally speaking, separating objective from subjective quotations is especially challenging in our scenario. It is easier to classify quotations as subjective if they are positive. If quotations are negative the algorithm classifies them more often as neutral. The polarity classification quality for negative and positive quotations is comparable. As Pang et al. [37] we find that incorporating position information into the feature vectors hardly influences sentiment classification effectiveness and therefore can be excluded from the feature vectors.

Inspired by Polanyi and Zaenen [40] we intend in our future work to imply more contextual shifters and patterns for German to calculate contextual feature weights instead of only encoding the presence and frequency of valence shifters. At the same time we plan to consider discourse markers for feature weight calculation following Mukherjee and Bhattacharyya [32]. In addition, appraisal groups may serve as supplementary information for the feature vectors [53]. Considering sentiment

targets our future work will include on the one hand the extension of our corpus by more text-anchored sentiment targets and on the other hand we will shift our work toward topic-oriented sentiment target detection that aims at determining supporting or opposing statements [34].

To extract even more opinionated passages from news articles in the future work we not only want to consider quotations but also include other news article parts. Furthermore, a context-dependent sentiment analysis requiring world knowledge or interpretation could retrieve additional subjective text [27].

### 1.5 Application

We present the results of our news aggregator to users via a web interface. The home page of our *news'd* demonstrator provides an overview of the currently most important events discussed in the news. In order to tackle the enormous amount of news material, our interface implements various ranking scores. Besides sorting news clusters by actuality or size, users may navigate news clusters ordered by their “hotness”. The hotness measure combines different cluster characteristics. It weights appropriately a cluster’s total growth since, its creation time, and its recent growth in a sliding time window to calculate one score that indicates how “hot” the news cluster is. Figure 1.7 shows the main page of our *news'd* demonstrator. As in other commercial news portals our interface also allows browsing news events by categories like “Politics”, “Economy”, etc. Each department page is organized

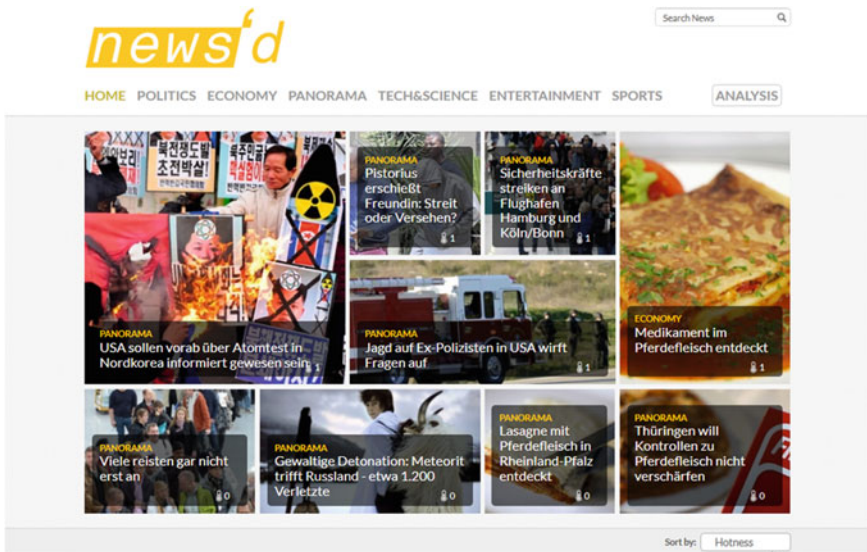


Fig. 1.7 The main page of the web application presenting the current top events

analogously to the main page except that it filters the news clusters corresponding to the category.

For a deeper, more structured insight into single events the interface organizes the view for each news cluster by dividing the page into specific sections presenting different types of information instead of simply listing all contained news articles. A news cluster view starts with its leading article, followed by the most recent news articles grouped by their age. On the right-hand side the users are provided with meta-information like a visualization of the cluster’s development or the key concepts and named entities the news cluster is dealing with.

An extra analysis page offers the search for quotations and sentiment tags. Users have the possibility to specify whether to search for quotations or sentiment tags and to select how to order the results. At this point our interface allows sorting by relevance and actuality. The resulting quotations or sentiment tags, respectively, are presented at the top of the page and the corresponding news articles thereunder. Figures 1.8 and 1.9 show a preliminary presentation of the analysis search results.

The future work includes an extension of the analysis page by offering additional statistical information on quotations and their opinions and the implementation of a view that directly compares opinionated quotations according to a topic or a target entity. The comparison view allows users to easily grasp, e.g., the most opposing or the most frequent/important comments. We aim to create a service with full archive



Fig. 1.8 The analysis page of the web application presenting quotations by and about “Isabelle Werth”

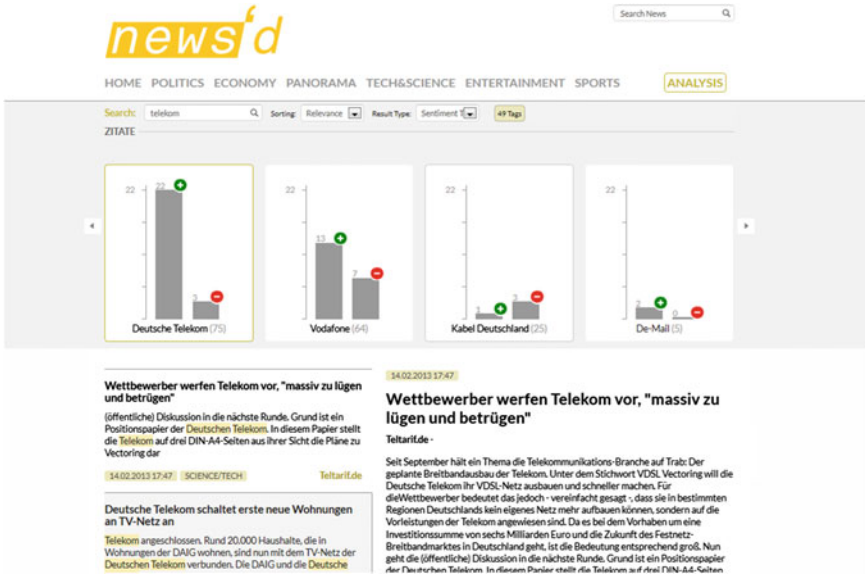


Fig. 1.9 Sentiment tags evaluating the perception of “Telekom” in comparison to related items

support, both to research topics and newsworthy entities as well as their perception in the press.

### 1.6 Conclusion

We presented a system for aggregating and analyzing German news articles collected from a wide range of news sources. In addition to topic detection and tracking of news stories, and building a dynamic topic hierarchy based on the current news situation, the central component implements methods for quotation extraction and sentiment analysis. As a stylistic means, quotations are used to underline significant information and can be regarded as a trustworthy piece of information, because they reflect statements of cited people and organizations in an original and genuine manner. Our approach to quotation extraction is rule-based and exploits text parts surrounded by quotation marks for retrieving direct quotations and the presence of reporting verbs and phrases for detecting indirect quotations. Each quotation is assigned a quotation speaker by applying a set of predefined rules and including a crude form of co-reference resolution. Evaluated on a manually created dataset with German-language quotations, the method yields convincing results, in particular for direct quotations. We assume quotations being the most subjective parts of news articles and base our sentiment analysis approach on quotations previously extracted by our quotation extraction component. We decompose the task into subjectivity

detection and polarity classification and train a separate SVM classifier for each subtask. In order to find the most appropriate feature set for both tasks, we assess a range of text classification features according to their applicability to sentiment analysis. The results suggest that sentiment words are most suitable for both tasks and that the classifiers perform best using a slightly different feature set. While part-of-speech information positively effects polarity classification, incorporating valence shifters and discourse markers improves subjectivity detection. Overall, the subjectivity detection in news articles appears more challenging than the polarity classification. To evaluate our work, we have created two corpora. The first corpora aims to support developing and assessing methods for quotation extraction. It contains direct and indirect quotations attributed with a quotation speaker and a reporting verb or clue if available. The second corpus bases on our quotation corpus and provides a sentiment label (positive, negative or neutral) for each quotation and an opinion target, if it is explicitly mentioned in the quotation. Both corpora are freely available for research purposes upon request.

In the future work, we plan to improve the recall of indirect quotations by automatically detecting reporting verbs instead of using a predefined list. Concerning the extraction of quotations speakers, we intend to incorporate a sophisticated approach to co-reference resolution. The future work on our sentiment analysis approach will include incorporating additional information during feature vector calculation to represent the text more precisely. We plan to shift our work toward topic-related and context-dependent opinion retrieval and allow also other text parts than quotations for sentiment analysis. In order to benefit from our results we plan to implement an extended view on newspaper quotations. The users will be presented a direct comparison of quotations expressed by different speakers according to a topic or entity, and a timeline of opinions to facilitate monitoring developments and estimating trends.

**Acknowledgments** We would like to thank Neofonie GmbH for providing news articles and the infrastructure for our demonstrator and developing important components of the news aggregator. In particular, the topic detection and tracking component and the component mining a dynamic topic hierarchy were designed and implemented by Neofonie GmbH. We would also like to thank Neofonie GmbH for regular communication, many helpful discussions, and valuable suggestions. Our thanks also go to Sascha Narr, Kerstin Schütt, Michael Hülfenhaus, Jonas Katins, Xenofon Chatziliadis and Leonhard Hennig. This work was funded by the Federal Ministry of Economic Affairs and Energy (BMWi) under funding reference number KF2392309KM1.

## References

1. A. Akbik, M. Schenck, *QuoteMine: A Repository of Newsworthy Quotes* (Darmstadt, Germany, 2013)
2. A. Balahur, R. Steinberger, Rethinking sentiment analysis in the news: from theory to practice and back, in *Proceeding of WOMSA'09* (2009)
3. A. Balahur, R. Steinberger, E. van der Goot, B. Poulliquen, M. Kabadjov, Opinion mining on newspaper quotations, in *Proceedings of the 2009 IEEE/WIC/ACM International Joint*

- Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT'09*, vol. 3 (IEEE Computer Society, Washington, 2009), pp. 523–526
4. A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. van der Goot, M. Halkia, B. Poulighen, J. Belyaeva, Sentiment analysis in the news, in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010, ed. by N. Calzolari (Conference Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odiijk, S. Piperidis, M. Rosner, D. Tapias (European Language Resources Association (ELRA), 2010)
  5. L. Barbosa, J. Feng, Robust sentiment detection on Twitter from biased and noisy data, in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING'10*. (Association for Computational Linguistics, Stroudsburg, (2010) pp. 36–44
  6. S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, H. Uszkoreit, Tiger: linguistic interpretation of a German corpus. *Res. Lang. Comput.* 2(4), 597–620 (2004)
  7. S. Cerini, V. Compagnoni, A. Demontis, M. Formentelli, G. Gandini, Micro-WNOP: a gold standard for the evaluation of automatically compiled lexical resources for opinion mining, in *Language Resources and Linguistic Theory: Typology, Second Language Acquisition, English Linguistics*, ed. by A. Sanso (Franco Angeli Editore, Milano, 2007)
  8. C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2(3), 27:1–27:27 (2011)
  9. S. Clematide, S. Gindl, M. Klenner, S. Petrakis, R. Remus, J. Ruppenhofer, U. Waltinger, M. Wiegand, MLSA—a multi-layered reference corpus for German sentiment analysis, in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)* (2012)
  10. L.A.D.F. de Moraes Sarmiento, S.S. Nunes, Automatic extraction of quotes and topics from news feeds, in *4th Doctoral Symposium on Informatics Engineering* (2009)
  11. É. de La Clergerie, B. Sagot, R. Stern, P. Denis, G. Recourcé, V. Mignot, Extracting and visualizing quotations from news wires, in *Human Language Technology. Challenges for Computer Science and Linguistics*, Lecture Notes in Computer Science, vol. 6562, ed. by Z. Vetulani (Springer, Berlin, 2011), pp. 522–532
  12. P. Domingos, A few useful things to know about machine learning. *Commun. ACM* 55(10), 78–87 (2012)
  13. D.K. Elson, K. McKeown, Automatic attribution of quoted speech in literary narrative, in *AAAI*, ed. by M. Fox, D. Poole (AAAI Press, 2010)
  14. A. Esuli, F. Sebastiani, SentiWordNet: a publicly available lexical resource for opinion mining, in *Proceedings of Language Resources and Evaluation (LREC)* (2006)
  15. M. Faruqui, S. Padó, Training and evaluating a German named entity recognizer with semantic generalization, in *Proceedings of KONVENS 2010* (Saarbrücken, Germany, 2010)
  16. J.R. Finkel, T. Grenager, C.D. Manning, Incorporating non-local information into information extraction systems by Gibbs sampling, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)* (The Association for Computer Linguistics, 2005), pp. 363–370
  17. J.G. Fiscus, G.R. Doddington, Topic detection and tracking evaluation overview, in *Topic Detection and Tracking*, ed. by J. Allan (Kluwer Academic Publishers, Norwell, 2002), pp. 17–31
  18. A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009) pp. 1–6
  19. V. Hatzivassiloglou, J.M. Wiebe, Effects of adjective orientation and gradability on sentence subjectivity, in *Proceedings of the 18th Conference on Computational Linguistics, COLING'00* (Association for Computational Linguistics, Stroudsburg, 2000), pp. 299–305
  20. L. Hennig, D. Ploch, D. Prawdzyk, B. Armbruster, H. Düwiger, E.W. De Luca, S. Albayrak, SPIGA—a multilingual news aggregator, in *Proceedings of GSCL'11* (2011)
  21. M. Hu, B. Liu, Mining and summarizing customer reviews, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'04* (ACM, New York, 2004), pp. 168–177



22. A. Kennedy, D. Inkpen, Sentiment classification of movie reviews using contextual valence shifters. *Comput. Intell.* **22**, 2006 (2006)
23. S.-M. Kim, E. Hovy, Extracting opinions, opinion holders, and topics expressed in online news media text, in *Proceedings of the Workshop on Sentiment and Subjectivity in Text, SST'06* (Association for Computational Linguistics, Stroudsburg, 2006), pp. 1–8
24. R. Krestel, S. Bergler, R. Witte, Minding the source: automatic tagging of reported speech in newspaper articles, in *Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008)* (European Language Resources Association (ELRA), 2008), pp. 2823–2828
25. W. Lezius, Morphy—German morphology, part-of-speech tagging and applications, in *Proceedings of the 9th EURALEX International Congress* (2000), pp. 619–623
26. J. Liang, N. Dhillon, K. Koperski, A large-scale system for annotating and querying quotations in news feeds, in *Proceedings of the 3rd International Semantic Search Workshop, SEMSEARCH'10* (ACM, New York, 2010), pp. 7:1–7:5
27. H. Li, X. Cheng, K. Adson, T. Kirshboim, F. Xu, Annotating opinions in German political news, in *8th ELRA Conference on Language Resources and Evaluation. International Conference on Language Resources and Evaluation (LREC-2012), 8th, 23–25 May, Istanbul, Turkey* (European Language Resources Association (ELRA), 2012). Accepted for publication
28. B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in *Mining Text Data*, ed. by C.C. Aggarwal, C.X. Zhai (Springer, US, 2012), pp. 415–463
29. D.C. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, Cambridge, 2008)
30. J. Martineau, T. Finin, Delta TFIDF: an improved feature space for sentiment analysis. *Artif. Intell.* **29**, 258–261 (2009)
31. S. Momtazi, Fine-grained German sentiment analysis on social media, in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (European Language Resources Association (ELRA), Istanbul, Turkey, May 2012)
32. S. Mukherjee, P. Bhattacharyya, Sentiment analysis in Twitter with light weight discourse analysis, in *Proceedings of COLING 2012*, Mumbai, December (2012). The COLING 2012 Organizing Committee, pp. 1847–1864
33. T. Nakagawa, K. Inui, S. Kurohashi, Dependency tree-based sentiment classification using CRFs with hidden variables, in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Association for Computational Linguistics, Los Angeles, 2010), pp. 786–794
34. T. O'Keefe, J.R. Curran, P. Ashwell, I. Koprinska, An annotated corpus of quoted opinions in news articles, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Association for Computational Linguistics, Sofia 2013), pp. 516–520
35. T. O'Keefe, S. Pareti, J.R. Curran, I. Koprinska, M. Honnibal, A sequence labelling approach to quote attribution, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Association for Computational Linguistics, Jeju Island, (2012), pp. 790–799
36. B. Pang, L. Lee, Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2**(1–2), 1–135 (2008)
37. B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, EMNLP'02*, vol. 10 (Association for Computational Linguistics, 2002), pp. 79–86
38. S. Pareti, T. O'Keefe, I. Konstas, J.R. Curran, I. Koprinska, Automatically detecting and attributing indirect quotations, in *EMNLP (ACL, 2013)*, pp. 989–999
39. W. Paulo, D. Fernandes, E. Motta, R.L. Milidiú, in *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*, Chapter Quotation extraction for Portuguese (2011)
40. L. Polanyi, A. Zaenen, Contextual valence shifters, in *Working Notes—Exploring Attitude and Affect in Text: Theories and Applications (AAAI Spring Symposium Series)* (2004)

41. B. Pouliquen, R. Steinberger, C. Best, Automatic detection of quotations in multilingual news, in *Proceedings of the International Conference Recent Advances in Natural Language Processing (Borovets, 2007)*, pp. 487–492
42. R. Remus, U. Quasthoff, G. Heyer, Sentiws—a publicly available German-language resource for sentiment analysis, in *Proceedings of the 7th International Language Resources and Evaluation (LREC'10) (2010)*, pp. 1168–1171
43. E. Riloff, J. Wiebe, W. Phillips, Exploiting subjectivity classification to improve information extraction, in *Proceedings of the 20th National Conference on Artificial Intelligence, AAAI'05*, vol. 3 (AAAI Press, Pittsburgh, 2005), pp. 1106–1111
44. A. Schiller, S. Teufel, C. Thielen, *Guidelines für das Tagging deutscher Textcorpora mit STTS*. Technical Report, IMS-CL, University Stuttgart (1995)
45. N. Schneider, R. Hwa, P. Gianfortoni, D. Das, M. Heilman, A.W. Black, F.L. Crabbe, N.A. Smith, Visualizing topical quotations over time to understand news discourse. Technical report, Technical Report TR CMU-LTI-10-013, Carnegie Mellon University, Pittsburgh (2010)
46. T. Scholz, S. Conrad, L. Hillekamps, Opinion mining on a German corpus of a media response analysis, in *Text, Speech and Dialogue*, vol. 7499, ed. by D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C.P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, P. Sojka, A. Horák, I. Kopeček, K. Pala (Springer, Berlin, 2012), pp. 39–46
47. C. Strapparava, A. Valitutti, WordNet-affect: an affective extension of WordNet, in *Proceedings of the 4th International Conference on Language Resources and Evaluation (ELRA, 2004)*, pp. 1083–1086
48. M. Tsytsarau, T. Palpanas, Survey on mining subjective data on the web, in *Data Mining and Knowledge Discovery* (Springer, New York, 2011)
49. P.D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL'02* (Association for Computational Linguistics, Stroudsburg, 2002) pp. 417–424
50. V.N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, New York, 1995)
51. S. Wang, C. Manning, Baselines and bigrams: simple, good sentiment and topic classification, in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, vol. 2 (Association for Computational Linguistics, Jeju Island, 2012), pp. 90–94
52. St. Weiser, P. Watrin, Extraction of unmarked quotations in newspapers, in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, ed. by N. Calzolari (Conference Chair), K. Choukri, T. Declerck, M.U. Doğan, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis (European Language Resources Association (ELRA), Istanbul (2012)
53. C. Whitelaw, N. Garg, S. Argamon, Using appraisal groups for sentiment analysis, in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM'05* (ACM, New York, 2005), pp. 625–631
54. H. Yu, V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP'03* (Association for Computational Linguistics, Stroudsburg (2003), pp. 129–136

## Chapter 2

# Twitter Sentiment Tracking for Predicting Marketing Trends

Cagdas Esiyok and Sahin Albayrak

**Abstract** We present a web-based Twitter sentiment tracking tool for brands. The tweets about four companies, namely, Facebook, Twitter, Apple, and Microsoft are collected by this system. The collection is implemented in an hourly basis in 17 Anglophone cities from which these tweets are sent. After collecting the tweets, the system classifies them as positive or negative by using the Naïve Bayes and Maximum Entropy classification methods. Later on, the system determines the winner brand of each city according to the percentage of positive tweets sent by users located in the aforementioned cities. Lastly, the winner brands of the day can be monitored on a web page using Google Maps. To increase the performance of classification methods, the tweet texts are preprocessed, such as through converting all the letters to lower case, both for training hand-classified dataset and for the collected tweets. Furthermore, statistical tracking charts can be viewed via web page of the system. A dataset is produced by collecting 362,529 tweets in 9 days via Twitter API for the research, which is automatically classified by the system. Performance of the Naïve Bayes and Maximum Entropy classification methods is also evaluated with the hand-classified dataset.

### Carl Marks Is an Intern

Finally, holidays had started—no school for a couple of weeks. “Once we have holidays, I will do nothing but chill in the sun,” he promised himself only weeks before the last day of school. His parents weren’t too happy about this attitude though: “Carl, just one more year until you finish high school. I think you should spend this summer holiday working as an intern somewhere”, his mom told him. “Look at your sister. She didn’t do anything last year and now, she has to do an internship to find out what she is interested in”, his dad added. “She is losing a whole

---

C. Esiyok (✉) · S. Albayrak  
Technische Universität Berlin, Berlin, Germany  
e-mail: cagdas.esiyok@dai-labor.de

S. Albayrak  
e-mail: sahin.albayrak@dai-labor.de

year because of that. You go and find yourself an interesting internship now. It will definitely make it easier for you to decide next year what to do next.” Eventually, Carl backed down. He actually already knew that he would like to become a computer scientist, but he also understood that his parents wouldn’t let him enjoy his summer holidays this year. It took him two days only until he had his first offer. “Being a computer scientist is awesome”, he thought. “The companies are just waiting out there to hire IT experts.”

Indeed, his internship turned out to be pretty interesting. He was in particular pleased by the work environment that he found: Free soft drinks and the mandatory Foosball table in the corner guaranteed a deluxe start-up experience. While his mind was occupied with these thoughts, his project manager Sandra entered the room.



“Hello Carl, did you get a chance to have a look at my e-mail?” Sandra asked.

“Hi Sandra,” Carl said, smiling. “Which one do you m—”

“The last one,” Sandra said, cutting Carl off. “I have sent it just now,” she added, smiling and blinking her eyes.

“Come on Sandra, how come I could get a chance to check it,” he said, laughing. “I am not a superhero.”

“Yeah, that’s true,” she acknowledged, smiling. “Please let me summarize it then . . .”

“Of course,” Carl agreed, “please!”

“To sum up, the main task is tracking positive and negative comments about our company and the opponent companies in Twitter,” she said, taking a deep breath.

“Hmmm,” Carl pondered, “it seems that we need to develop a web-based tracking system for Twitter, don’t we?” he asked.

“Absolutely right,” she acknowledged. “Actually, we could separate the main task into sub tasks . . .” she added. “Firstly, the system is supposed to collect tweets about companies from several cities.”

“It sounds we are going to employ the Twitter API,” Carl mumbled.

“Yes, Carl, it seems that you are very familiar with Twitter due to daily online activities.”

“So familiar!” he sighed, rolling her eyes.

“I remember Carl, you had told me that you wanted to work on different projects,” she admitted. “But to be honest, I think you are one of the most competent interns who could achieve this task on time owing to your experiences,” she asserted.

“Here I am,” Carl bragged, smiling.

“Then, please stop to whine and keep on listening,” she said, smiling. “The second task is to classify the tweets that you collected in the first task as positive or negative.”

“Hmmm, the second task is bipolar sentiment analysis,” he said.

“Yes, it is,” she told. “After sentiment analysis, for each city, your system should detect the company that has the highest percentage of positive tweets as the third task.”

“It sounds as if it is a kind of competition,” he said. “We are going to determine the winner company of each city according to ratio of positive tweets received.”

“Kind of a competition,” she agreed. “We need to pre—”

“How will we . . .” Carl interrupted. “How will we present the results?” he asked, “By means of a map or illustration . . .” he added.

“If you didn’t interrupt me, I was about to say,” Sandra said, smiling.

“Oops, sorry . . .” he said, looking up.

“As a last step, the winner brands of the day can be monitored on a web page using the Google Maps”, she told.

“Let me conclude,” he said. “The first step is collecting the tweets, the second one is applying bipolar sentiment analysis and the last step is developing an interface so as to present the results,” he muttered.

“Good brief!” she told. “That’s what we are going to do.”

“Then, what is the main ambition of this project for our company?” he wondered.

“To detect any bad trend,” she replied.

“What do you mean by detecting any bad trend?” he asked.

“I mean, by means of this system, our company can intervene in any bad trend,” she told, fingering her pendant.

“Would you please give me an example?” he asked.

“For example,” she answered, “reactions of users to a new product could be tracked and analyzed automatically in order to learn whether users liked or disliked it.”

“That would be really nice for companies,” he said, smiling.

“Definitely,” she agreed. “Time is of the essence, I wish you luck Carl.”

## 2.1 Introduction

As described in Carl’s story, sentiments and opinions of customers might be very important for companies in our times. Social and micro-blogging platforms are mostly utilized to get this kind of information. Especially, Twitter, whose popularity

is incrementally increasing day by day, is one of the latest trends in the recent era. This 140-character-allowing micro-blogging social platform has a wide range of users varying from people to organizations, such as politicians, celebrities, and companies. According to Kwak et al. [16], the number of Twitter users was 40 million in the world in July 2009, but in August 2014, it is revealed on Twitter's official web page<sup>1</sup> that Twitter has 271 million monthly active users, although it is a young company established in 2006. This drastic change in the number of users sheds clear light on the growth of the company.

The recognition that such a growing company has a vital impact on everyday life has become an integral element of encouragement for researchers to conduct studies in regards to the reflections of tweets on the real world to make some predictions. For example, Asur and Huberman [3] were able to forecast box-office revenues for some movies by using the tweets. Another study showed that Twitter has a vital role in elections if used effectively. Tumasjan et al. [31] discovered that messages in favor of a candidate party can alter the election result. A similar study by Diakopoulos and Shamma [10] also demonstrated that Twitter is one of the best ways to predict the election results. In that study, the tweets, which were sent by the users during the 2008 USA presidential debate, were tracked. It was found that the number of negative tweets posted by the users was less than the number of negative tweets posted when McCain spoke. Afterward, Obama won the election against McCain. Jansen et al. [13] analyzed more than 150,000 micro-blog posts which contain brand comments, sentiments, and opinions. They showed that micro-blogging is a kind of electronic word-of-mouth of customers which are related to brands and products.

As it can be understood from the studies above, Twitter has become one of the best ways of getting customers' opinions and making predictions about the results of elections and events. Considering all the predictions made in this way, several precautions can be taken in case of an unfavorable outcome. For example, a company can decrease the prices by tracking the sentiment of tweets about a particular product. A negative trend on twitter can lead to a decrease in prices. In accordance with this kind of purposes, in this chapter, a web-based system was created in order to extract information from Twitter by tracking sentiment.

In this chapter, the primary objective is to present a web-based Twitter sentiment tracking tool. This tool collects the tweets about four brands namely, Facebook, Twitter, Apple, and Microsoft, in an hourly basis in 17 Anglophone cities from where these tweets were sent. The list of the cities used in this analysis can be observed in Fig. 2.1. After collecting tweets, the system analyzes sentiments of tweets and classifies them as positive or negative by using two classifier methods namely Naïve Bayes and Maximum Entropy. Later on, the system determines the winner brand of each city according to the percentage of positive tweets by using the information coming from the users located in selected cities. At the end, the winner brands can be seen using Google maps. For example, if the winner brand is Microsoft in New York on a selected day, the system used in this chapter shows the Microsoft logo

---

<sup>1</sup> <https://about.twitter.com/company/>.

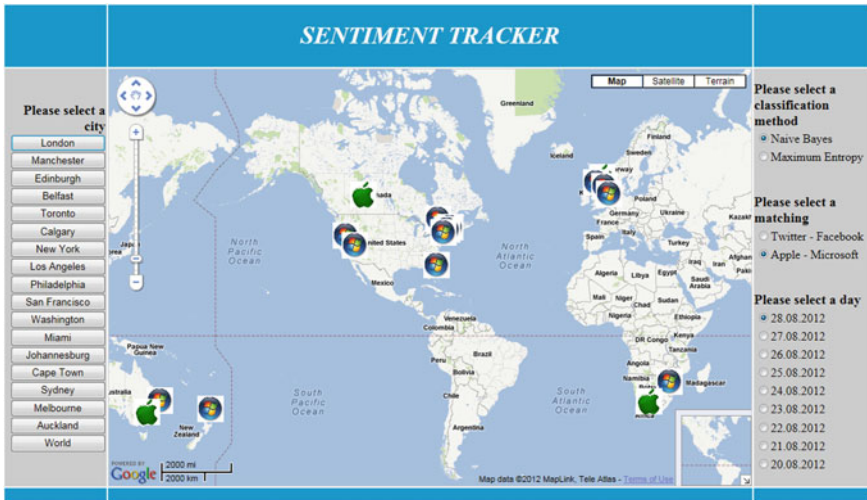


Fig. 2.1 Screenshot of web interface

on New York in Google maps as it can be seen from the Fig. 2.1. Furthermore, the statistical indicators can be viewed by the system as well.

It is demonstrated that the Naïve Bayes classification and Maximum Entropy classification methods can be effectively used to perform sentiment analysis on tweets. To the best of our knowledge, this is the first geographic location-based sentiment tracking system for Twitter, which allows one to monitor the brands in line with the views of people in different cities.

Section 2.2 starts with the concepts of blogging and microblogging. Afterward, Twitter, Sentiment Analysis, Natural Language Processing, Maximum Entropy, and Naive Bayes classification methods are briefly described. Section 2.3 provides an informative introduction to the technologies used while developing the web-based project. Python and Natural Language Tool Kit (NLTK) are introduced, followed by Twitter API, Google Maps API, and hand-classified dataset. Section 2.3 also describes how the tracking sentiment for Twitter project was implemented. Section 2.4 covers the evaluation of the system. Section 2.5, finally, summarizes the chapter and gives a brief outlook for future studies.

## 2.2 Background

### 2.2.1 Blogging and Micro-Blogging

Blogging can be described as a platform where people can share their hobbies and personal experiences on the World Wide Web. It has become one of the social

phenomena with Web 2.0. Also known as Weblogs, blogs are updated in a regular pattern in an attempt to incorporate most recent archived posts.

One of the most vital features of blogs is that a single author promotes and maintains each of them with the newest shares appear at the top. In general, blog posts include texts only; nevertheless, they may incorporate photos or some other multimedia content. Most of the blogs provide hypertext links that allow users to go to other websites just through a click, and many blogs make it possible for the users to leave comments. Recently, advances in technology facilitated the development of blogging and enhanced its accessibility. Blogs have been used intensively in the popular media; this has been evident with the intensive use of the blogging in political campaigns, new organizations and businesses. Blogs that are specifically allocated for politics, news, and the share of technological developments are the main blogs and websites in general that receive a great number of visitors a day.

In such a diverse environment consisting of various types of blogs, Herring et al. [12] categorized them under three types. The first one is the individually authored personal journals. The second is entitled as “filters” as they select and share commentary on information received from other websites. The last one is knowledge blogs. A vast majority of their sample consists of the personal journal type, which is responsible for 70.4% of their sample. In this type, authors post their experiences in their lives and inner thoughts, opinions, and feelings.

One can define micro-blogging as the type of blogging that allows people to share their opinions and actions at the time of writing as short messages. In other words, it fills the gap between instant messaging and blogging. This relatively new type of blogging makes it possible for individuals to post laconic text updates, using a variety of communication channels ranging from text messages for mobile phones and instant messaging to e-mail and the Web.

The main difference between the regular blogging and micro-blogging is the text size restrictions appearing in the micro-blog posts. Micro-bloggers are permitted and confined to present their post in a limited size of text message. This feature enables micro-blogs to be amendable by sending text messages from mobile clients such as mobile phones. Appearing as an easily accessible system via mobile clients, micro-blogging has become very popular with the contributions of a wide range of users composed of average persons, celebrities, and commercial organizations. For distinct purposes, individual users such as politicians, actors, musicians, academics, and students use this blogging type regularly. Businessmen, institutions, and activists use this system intensively as well.

Micro-blogs may indicate what the micro-blogger is doing and thinking. Micro-blogs may also provide information about the news, entertainment sector, and good deals. The ones providing specific data, in general, provide reference to an external resource owing to their limited size, which makes it hard to convey the news by themselves. As broadcasting is briefly defined as spreading information over a large range of audience, micro-blogs can be used as a source of broadcasting information about anything the users want to learn about. There are various micro-blogging



services ranging from *Tumblr*<sup>2</sup> to *Plurk*,<sup>3</sup> and the most popular of all, as indicated above, is *Twitter*.<sup>4</sup>

### 2.2.2 *Twitter*

Twitter is a popular micro-blogging tool which has taken big steps since the date of establishment in October 2006. As the popularity of micro-blogs grows among Internet users, the interests of academics on micro-blogs grow accordingly. The high number of micro-blog posts enables researchers to specialize and focus on different research areas. Because micro-blogging is a new concept, the studies and research on the matter is new as well.

In Twitter, posts, or in Twitter jargon “tweets,” are confined to 140 characters. The posts can consist of plain texts, links, and keywords that possess a special meaning in Twitter such as hashtags, mentions, and retweets. Hashtags are single word tokens, which follow the hash symbol, ‘#’. They can appear anywhere in a tweet, and they are used to tag a tweet, and a tweet can only be hash tagged by its author. Mentions, on the other hand, are the user names used in Twitter, which go after an “at” symbol, ‘@’. A user in Twitter can use the pattern ‘@ <username>’ in order to address another user. Retweets are used when a user wants to spread a tweet published by another user. To underline that a tweet is a repeat (re-tweet) of another tweet, users write RT in their tweets.

### 2.2.3 *Natural Language Processing*

Natural Language Processing (NLP) can be described as a subtitle of computer science, which deals with languages and uses Machine Learning techniques to process human language.

NLP incorporates many subfields and tasks some of which are automatic summarization, discourse analysis, machine translation, relationship extraction, and answering questions. The improvement of these fields has positive repercussion on the developments of many other areas in different fields.

The studies on NLP commenced as early as the 1940s. The very first application of NLP could be observed as a Machine Translation application developed during the World War II in order to break codes. In 1950, a criterion of intelligence was suggested by Alan Turing, which in present time is referred as the “Turing test” [32]. With this criterion, computers were rendered able to imitate a person in a conversation with a human judge.

---

<sup>2</sup> <http://www.tumblr.com/>.

<sup>3</sup> <http://www.plurk.com/>.

<sup>4</sup> <http://www.twitter.com/>.

After the 1960s, NLP studies were enriched with Artificial Intelligence (AI). With the impact of AI, NLP studies focused on world knowledge and tried to get better in the construction and manipulation of meaning representations. The first vital work shaped by AI was Green et al. [11] BASEBALL question-answering system. Starting in 1961, the system was working on the problems of addressing and constructing data and knowledge.

In 1966, the report published by the Automatic Language Processing Advisor Committee (ALPAC) asserted that the 10-year-long research could not satisfy expectations. In the light of the report, the research on NLP diminished considerably in international sphere.

Starting from the late 1980s, the inclusion of ML approaches to NLP paved the way for the resurrection of the studies. In that regard, the work of Rosenschein and Shieber [25] is of great importance. Their research handled a scheme for syntax-directed translation, reflecting upon compositional model-theoretic semantics.

The advances in computer science considerably paid off in making the 1990s the expansion period for NLP. Distinct approaches have become a source of examination with the contributions of improved computerized methods. A valuable study in the 1990s is the study of Berger et al. [5]. With an efficient implementation of the approach, their study provided a maximum-likelihood approach for automatic construction of maximum entropy models.

Joachims [14] studied on the text classifiers learning, using the Support Vector Machine (SVM). The work of Joachims is very crucial since examining certain features of learning with text data proved the suitability of SVM. Not only did it provide theoretical but it also created empirical evidence during the process of examination.

Another success story in this field comes from the study of Soderland [30]. In his research, Soderland presented a system, which was designed to cope with different text styles. The system strives to handle different sets of rules requirement problem of Information Extraction (IE) systems, by grasping the rules of text extraction automatically. It also targets to deal with various text styles in a wide range including high structured ones and free texts.

Starting from the early twenty-first century, NLP has turned out to be a rooted area incorporating distinct branches related to many areas. Numerous studies are carried out today owing to the contributions provided by NLP techniques.

#### ***2.2.4 Sentiment Analysis of Text***

Boiy et al. [6] define sentiments as “emotions, or as judgments, opinions or ideas prompted or colored by emotions.”

Determining the attitudes, feelings, and opinions of a writer or speaker, which is in a text or video related to a topic, would be the definition of sentiment analysis. Pang and Lee [22] express this process as the computational examination of an opinion,

sentiment, and attitude. The combination of the work done for this sake is described in the literature as opinion mining, sentiment analysis, and/or subjectivity analysis.

Determining the attitude of a speaker or a writer is of great importance for the sentiment analysis as it is the main purpose of it. However, this can be very difficult at times. In Li and Wu's [17] own words: "The attitude can be any forms of judgment or evaluation, the emotional state of the author when writing, or the intended emotional communication."

In sentiment analysis, two primary approaches are used, namely, linguistic and machine learning. In linguistic approaches, studies are conducted by creating a set of rules, and then by comparing them with the analyzed text. An example of the linguistic approach could be the study of Benamara et al. [4] who proposed a sentiment analysis technique based on adverb–adjective combinations (AAC). The technique utilizes a linguistic analysis of adverbs of degree.

Devitt and Ahmad [9] put forward that sentiment analysis in computational linguistics has closely observed how textual features, such as lexical, syntactic, and punctuation, alter the emotional content of the text. Furthermore, the sentiment analysis considerably contributes to the automatic detection of these features so as to gather a sentiment metric for a word, sentence, or the whole text.

On the other hand, in machine learning approaches, methods rely on statistical evaluations and analyzes such as frequency of positive and negative entities in any text.

The reason behind the growing interest in this field stems from the benefits it can provide. The main advantages of this research area are observed in stock market. Predicting stock market behavior based on the sentiment results of Twitter posts, according to Bollen et al. [7], can result in favorable outcomes. Moreover, O'Conner et al. [21] underscore measuring public opinion poll in regards to presidential elections from blog data. Pang and Lee [22] mention the advantages of using the sentiment analysis in dealing with business intelligence tasks with respect to customer feedback.

### ***2.2.5 Text Classification***

Text classification can be defined as assigning predefined category labels to documents such as e-mails to detect whether they are spam or nonspam, or web pages to detect whether they are in English, German, or Turkish.

In this chapter, a supervised learning method was used, which is to say, first a set of training documents were labeled, and then a machine learning algorithm was applied to the document for classification.

Chen et al. [8] clearly state the increasing importance text classification. They argue that the enhanced availability of digital texts and incremental increase in the need to access them rendered text classification as a vital task. For a long time until recently, various methods based on machine learning and statistical theory have been implemented in text classification.

The methods implemented in this chapter are Naïve Bayes and Maximum Entropy. These methods have been efficiently applied to text classification studies in the literature. There are many successful examples in the literature about Bayesian Probabilistic classifiers [1, 15, 28, 33] and Maximum Entropy classifiers [2, 18, 23].

### 2.2.5.1 Naïve Bayes Method

The Binary Independence Model was developed by Yu and Salton [34] and Robertson and Jones [24] in the 1970s. The model held the status of being one of the first models utilized in probabilistic information retrieval. The Naïve Bayes Method can be briefly reviewed as follows:

Let  $\vec{x}$  be a vector to be classified, and  $c_k$  be a possible class. The information to be known is the probability that the vector  $\vec{x}$  belongs to the class  $c_k$ . First, the probability  $P(c_k|\vec{x})$  is transformed using Bayes' rule.

$$P(c_k|\vec{x}) = P(c_k) \times \frac{P(\vec{x}|c_k)}{P(\vec{x})} \quad (2.1)$$

$P(c_k)$ , i.e., the class probability can be estimated from training data. Due to the sparsity of training data, in most cases direct estimation of  $P(c_k|\vec{x})$  is impossible.  $P(\vec{x}|c_k)$  is decomposed below,

$$P(\vec{x}|c_k) = \prod_{j=1}^d P(x_j|c_k) \quad (2.2)$$

where  $x_j$  is the  $j$ th element of vector  $\vec{x}$ . So  $P(c_k|\vec{x})$  becomes as follows:

$$P(c_k|\vec{x}) = P(c_k) \times \frac{\prod_{j=1}^d P(x_j|c_k)}{P(\vec{x})} \quad (2.3)$$

By using this equation,  $P(c_k|\vec{x})$  can be calculated and  $\vec{x}$  can be classified with the highest  $P(c_k|\vec{x})$ .

### 2.2.5.2 Maximum Entropy Method

Nigam et al. [20] defined maximum entropy as a technique for estimating probability distributions using data. The most important rule in maximum entropy is that when nothing is known, the distribution should be kept uniform; in other words, distribution should have maximal entropy. In order to gather a set of constraints for the model, which describe class-specific expectations for the distribution, labeled training data

is utilized. The constraints are signified as expected values of “features,” any real-valued function of an example.

It is noteworthy of highlighting that there is no conditional independence assumption between features, as the Naïve Bayes classifier does. Importantly, it makes no conditional independence assumption between features, as the Naïve Bayes classifier does.

## 2.3 Implementation

### 2.3.1 Technologies Used

#### 2.3.1.1 Python

Python can be described as a programming language, which allows one to work fast and integrate the user systems efficiently. The gains in productivity and decrease in maintenance costs can be observed in the short-run once starting to use Python.

Sanner [27] defines python as “an interpreted, interactive, object-oriented programming language . . . [which] provides high-level data structures such as list and associative arrays (called dictionaries), dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc.” He adds that despite having a quite simple yet elegant syntax, it is a powerful programming purpose for general purpose. The language was developed in 1990 by Guido van Rossum. It is free as in the case of other scripting languages, including for commercial purposes. Another vital feature of the language is that it can be used in any modern computer.

Sanner also declared that an important resource for Python, apart from the available books, is the Python website.<sup>5</sup> The website generates access to code, documentation, articles, mailing lists, and packages.

The system described in this chapter mainly uses Python for collecting the tweets via Twitter API, making preprocesses on collected tweets, and writing, reading database.

#### 2.3.1.2 Natural Language Toolkit

Natural Language Toolkit (NLTK) is one of the best ways for studying natural language processing using Python. It is an open source toolkit and can be run on all platforms, which are supported by Python such as Linux, Windows, and Unix.

What NLTK means is clarified in detail on the website of the NLTK.<sup>6</sup> It is stated that NLTK is a platform in which Python programs are developed to work with human

---

<sup>5</sup> <http://www.python.org/>.

<sup>6</sup> <http://www.nltk.org/>.

language data. The system offers interfaces that are not difficult to use to more than 50 corpora and lexical resources including WordNet. Moreover, the system also provides a set of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Regarding further benefits of the system, Loper and Bird [19] stated that NLTK leads to a simple, extensible, and even framework for projects and assignments. They declared that NLTK is well documented, easy to understand how it works, and simple to use.

In this chapter, NLTK is used to classify the tweets by using the Naïve Bayes and Maximum Entropy classifier methods.

### 2.3.1.3 Twitter API

Twitter API is described, by Sharifi et al. [29], as an API based completely on HTTP, and it is provided by Twitter. With Twitter API, users can accomplish nearly any task that can be achieved through Twitter's web interface. As for the nonwhite listed users, Twitter Rest API allocates 150 requests per hour to a user.

Fortunately, Twitter Search API, which is used in this chapter, does not have this kind of a restriction for developers. But frequency and complexity of requests is important to avoid being in blacklisted users.

Certain points are found crucial to be grasped before using the Search API. For example, the Search API is an index composed of the most recent tweets, not an index demonstrating all tweets. Currently, the index incorporates tweets of 6–9 days. Furthermore, the Search API cannot be used to search for tweets that are older than a week. Queries are subject to restrictions owing to complexity. In this case, the Search API will report an error as a response. All queries are made without identification to be provided; in other words, search does not require authentication. The search pays attention to relevance, not to completeness. This may result in some tweets and users' being missed from the search results. The Search API cannot use the near operator, so the geo-code parameter should be used. Queries are restricted to 1,000 characters, including any operators. During the process of geo-based searches with a radius, 1,000 different sub-regions will be taken into consideration when evaluating and processing the query.

In this chapter, Twitter Search API is used to collect tweets.

### 2.3.1.4 Google Maps API

The Google Maps API, which is a free service provided by Google, allows developers to embed high-resolution maps into their web pages by using the JavaScript technology.

Furthermore, the API provides various functions that enable manipulation of the maps as well as making it possible to make additions to the content of the map via lots of services. Using this API, the users are enabled to design and create strong maps applications on their websites.

As stated in the study of Rousseaux and Lhoste [26], satellite views in high resolution are provided for certain zones as well. Apart from this, the street view that covers 360° panoramic street level views of plenty of cities is offered as well. In this chapter, Google Maps API is used to show the tracking results on the web page.

### 2.3.2 Hand-Classified Dataset

We use a hand-annotated dataset for training and testing the Twitter sentiment analysis algorithms purposes, it is composed of 1,035 hand-classified tweets as positive and negative.

In order to preprocess the raw tweets, Python script was written. This script mainly reads all tweets from our dataset and preprocesses all of them as can be seen in Fig. 2.2. Then, it writes these preprocessed tweets into a new dataset.

### 2.3.3 Background Processes

#### 2.3.3.1 Tweet Collecting

A Python script written for this chapter collects all tweets about the four brands based on 17 cities where the tweets are sent. Figure 2.3 shows the flowchart of this script.

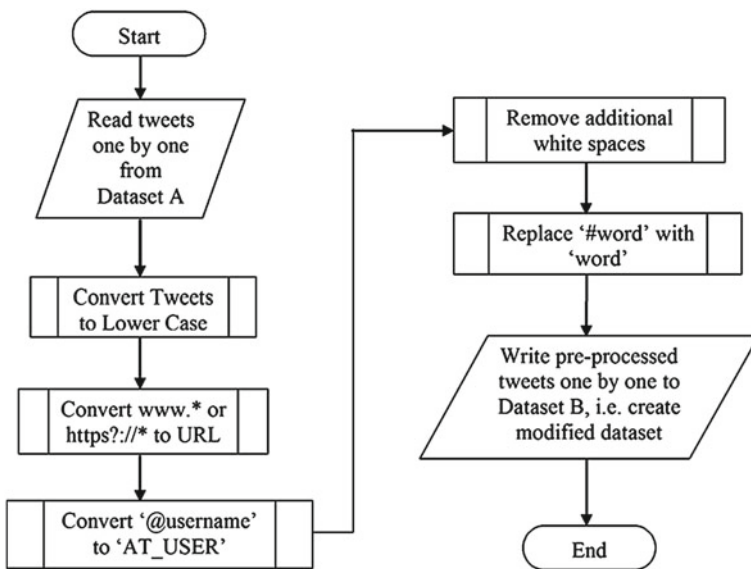


Fig. 2.2 Flowchart of preprocess steps for dataset

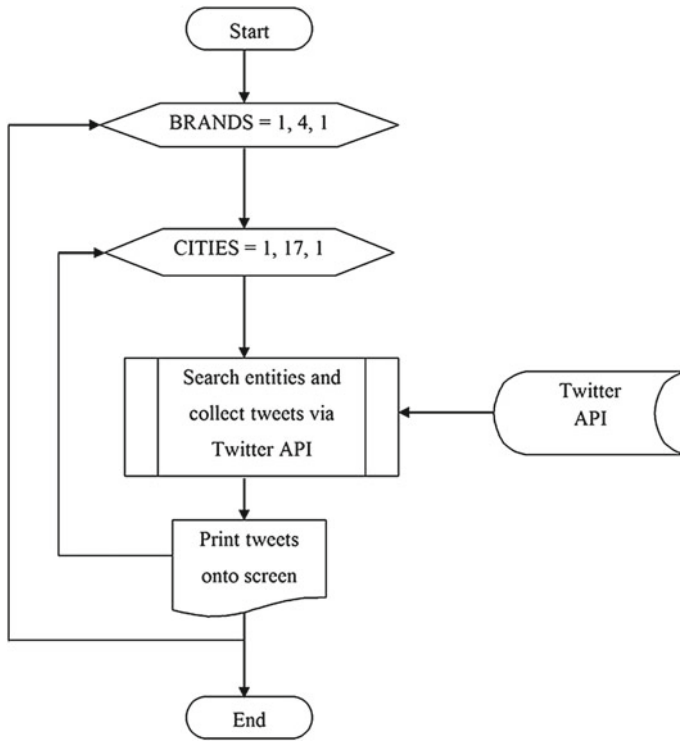


Fig. 2.3 Flowchart of tweet collecting steps

Relevant tweets that match a query are returned in the JavaScript Object Notation (JSON) format by the Twitter API. As described on JSON web page,<sup>7</sup> JSON's properties, which make JSON an ideal data-interchange language, are listed as the followings: a very lightweight data-interchange format, easy for humans to read and write as well as easy for machines to parse and generate based on a subset of the JavaScript Programming Language, JSON is a text format that is not dependent upon language at all, but it uses conventions that are well-known to programmers of the C-family of languages.

### 2.3.3.2 Sentiment Analysis

In this chapter, the Naïve Bayes classification method and Maximum Entropy classification methods are used to make the sentiment analysis. A Python script was written in order to classify the tweets collected. Basically, this script first trains the Maximum Entropy classifier and the Naïve Bayes classifier with training-modified

<sup>7</sup> <http://www.json.org/>.



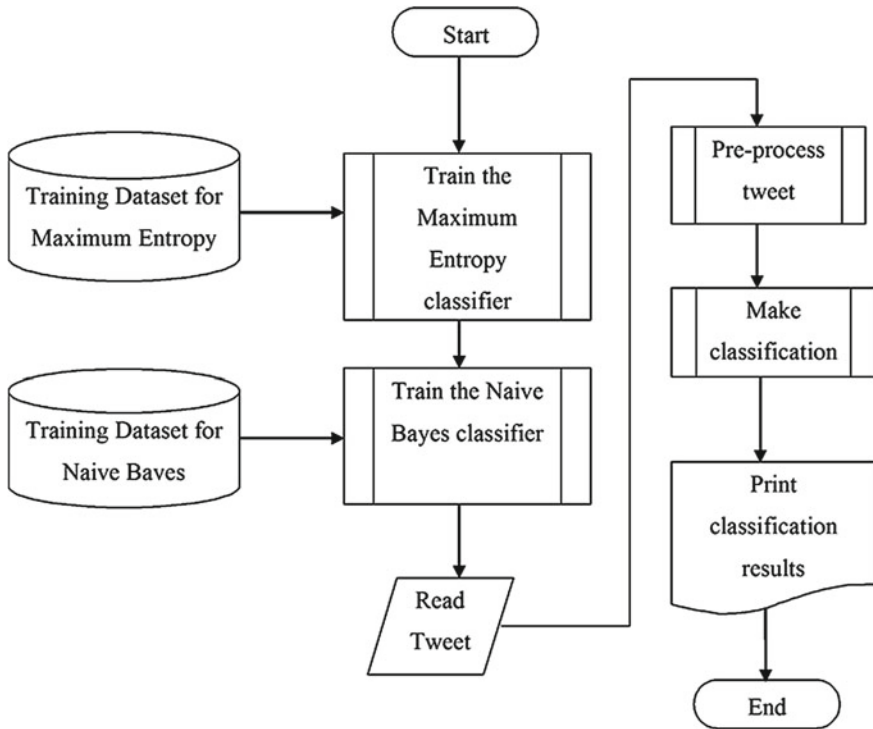


Fig. 2.4 Flowchart of classifier Python script

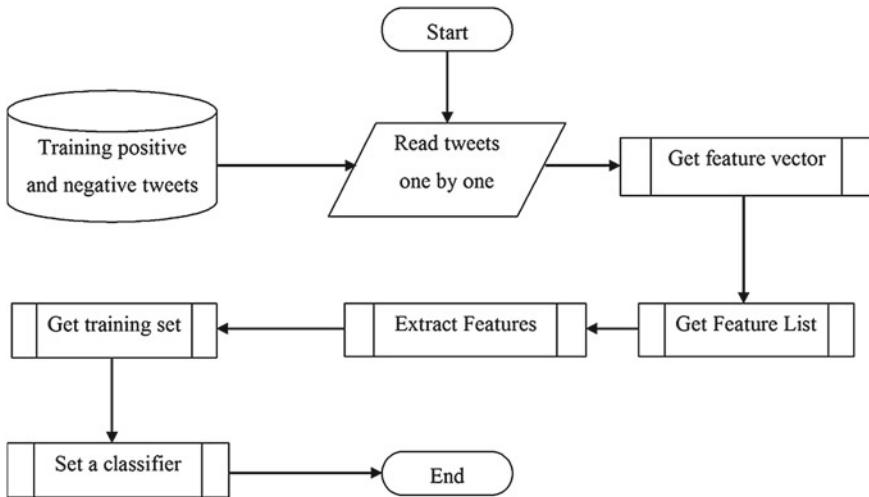
datasets which are presented in the subsection of the hand-classified data set and then read the given tweets. Second, script preprocesses these tweets, and then, the tweets are classified as positive or negative (Fig. 2.4).

**Step by Step Training Process**

*Step 1* To automatically classify a tweet, first the classifier needs to be trained. To do that, a list of hand-classified tweets is required. 512 hand-classified tweets are used to train the Maximum Entropy classifier. 1,035 hand-classified tweets are used to train the Naïve Bayes Classifier. The reason of using the 512 hand-classified tweets rather than 1,035 for Maximum Entropy classifier is to avoid the slow training process. Even when 512 tweets are used, the training process with 40 iterations takes unfeasible duration for an online system.

*Step 2* A feature vector needs to be created. The feature vector is the most crucial item in employing a classifier. A good feature vector can foresee how successful the results of the classifier will be.

*Step 3* After creating the feature vector, a sequenced feature list is produced. The most frequently used word is the first member of the feature list array. The feature list is used to train classifiers.



**Fig. 2.5** Flowchart of training process

*Step 4* After producing the feature list, the next step is extracting features. In a sample tweet such as “He has changed in his bag,” the feature words to be extracted are “changed”, “bag,” “has,” and “he.” Then, these feature words are examined whether they are included in the feature list words in order to extract features.

*Step 5* Features are applied to the classifier. To sum up, a flowchart of the training process is set as it can be observed from Fig. 2.5.

### 2.3.3.3 Automated Tweet Collecting and Classification

The tweet collector script presented in the subsection of tweet collecting and the classifier script presented in the subsection of sentiment analysis above are combined as a new Python script to produce an automated tweet collecting and classification system. First, the script trains the Maximum Entropy classifier and Naïve Bayes classifier with the training-modified datasets. Second, the system collects tweets about the four companies from the users located in several cities, and lastly, the script classifies tweets as positive or negative, and then it stores them into database. This script is converted into an executable file format to run it hourly as a background process on web server. Another reason why we converted it into the executable file format is to be able to run it without requiring a Python compiler installation. Below, Fig. 2.6 shows the flowchart of this executable file.

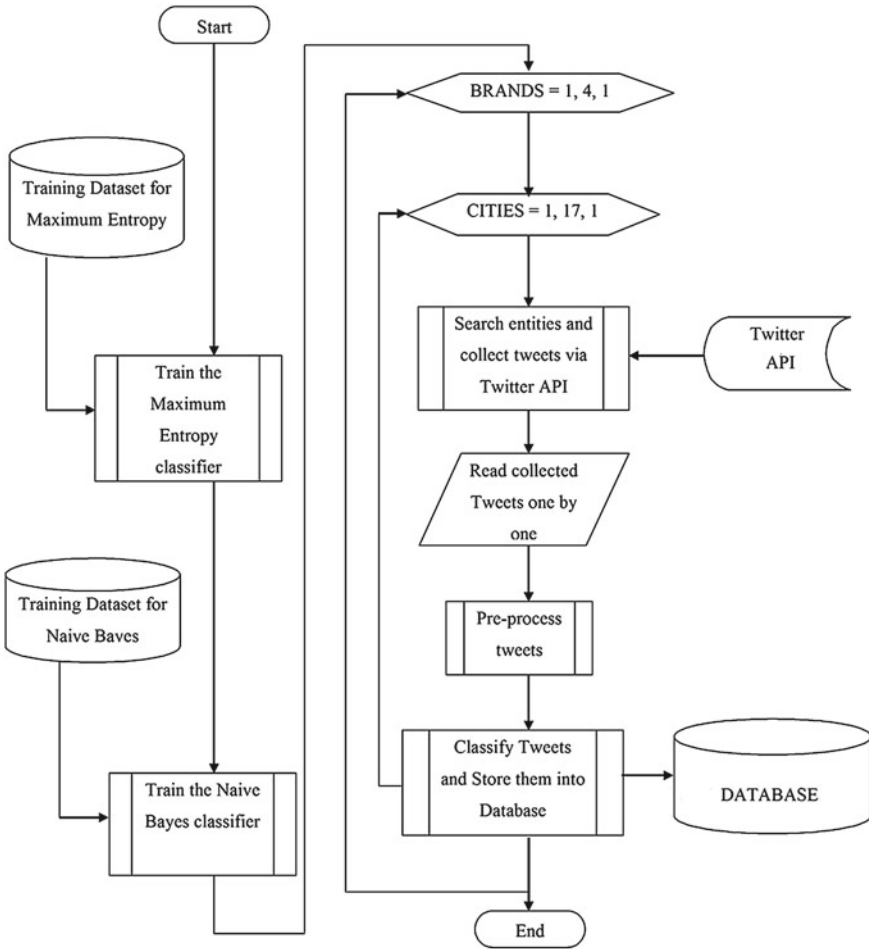


Fig. 2.6 Flowchart of tweet collecting and classification executable file

### 2.3.4 Web Interface

One of the main objectives of this chapter is to create a web-based sentiment tracking system for Twitter. This section briefly presents web interface of our tracking system.

#### 2.3.4.1 Main Page

The main page of web interface shows the winner brand of the day on Google Maps. In order to monitor the result, Google API and JavaScript are employed. Screen shots of the main page can be seen from Figs. 2.1 and 2.7. On the left side, users

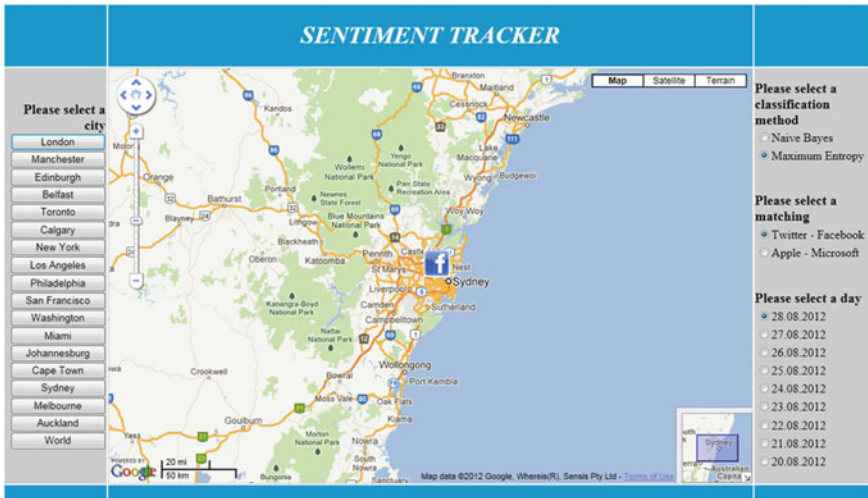


Fig. 2.7 Screen shot of the main page after selecting the city of Sydney

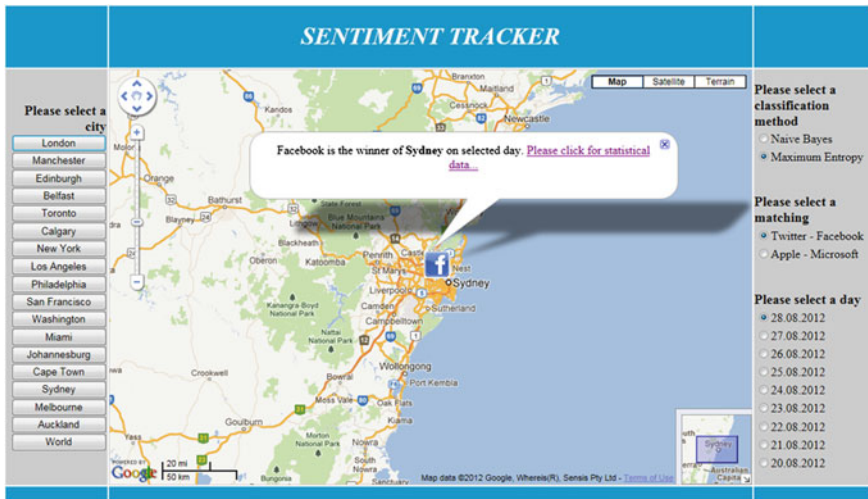


Fig. 2.8 Screen shot of the main page after clicking on a brand logo

can select the cities in order to monitor the winner brand. Users can also choose the classification method, brand matching, and the day on the right side of the web page.

If user clicks on the brand logo, a bubble appears as it can be seen from the Fig. 2.8. By following the link, the user can monitor the statistical charts related to cities and brands.

### 2.3.4.2 Chart Page

The aim of this page is to show the line charts of statistical data of brands according to days. If a user clicks on the “Please click for statistical data” link which can be seen from Fig. 2.8, the chart page is opened and shows the data as can be seen from the Fig. 2.9. First, it connects to database to collect the last 9 days results and then retrieves the following data to draw the charts.

- Percentage of positive tweets about Twitter classified by the Naïve Bayes method.
- Percentage of positive tweets about Facebook classified by the Naïve Bayes method.
- Percentage of positive tweets about Twitter classified by the Maximum Entropy method.
- Percentage of positive tweets about Facebook classified by the Maximum Entropy method.

### 2.3.4.3 Trigger Page

The aim of this page is to run scheduled tasks using only ASP.NET without setting a Windows web service. By means of a Trigger page, the web interface of our system could be run on every web server which supports ASP.NET hosting. Trigger page is called hourly to do scheduled tasks which can be listed as follows. First, it calls the executable file which is responsible for tweet collecting and classification—described in detail in the sub section of automated tweet collecting and classification—so as to collect and classify the last tweets sent. Then, it connects to the database in order to draw the last count of positive and negative tweets, such as last count of tweets about Twitter by users located in London on a given date. Third, it amends the database after getting the last count of positive and negative tweets. Lastly, it deletes duplicate records because it is possible that Twitter API might collect the same tweets as collected in the previous call.

To sum up the whole system, Fig. 2.10 shows the flowchart of the Web Interface.

## 2.4 Evaluation

### 2.4.1 *Performance of Classification Methods by Number of Data*

In this section, the performance of the Naïve Bayes and Maximum Entropy classification methods of the NLTK are evaluated according to the number of training data. In order to do this task, training datasets are produced based on hand-classified dataset which is described in Sect. 2.3. For all of the evaluations, the same testing

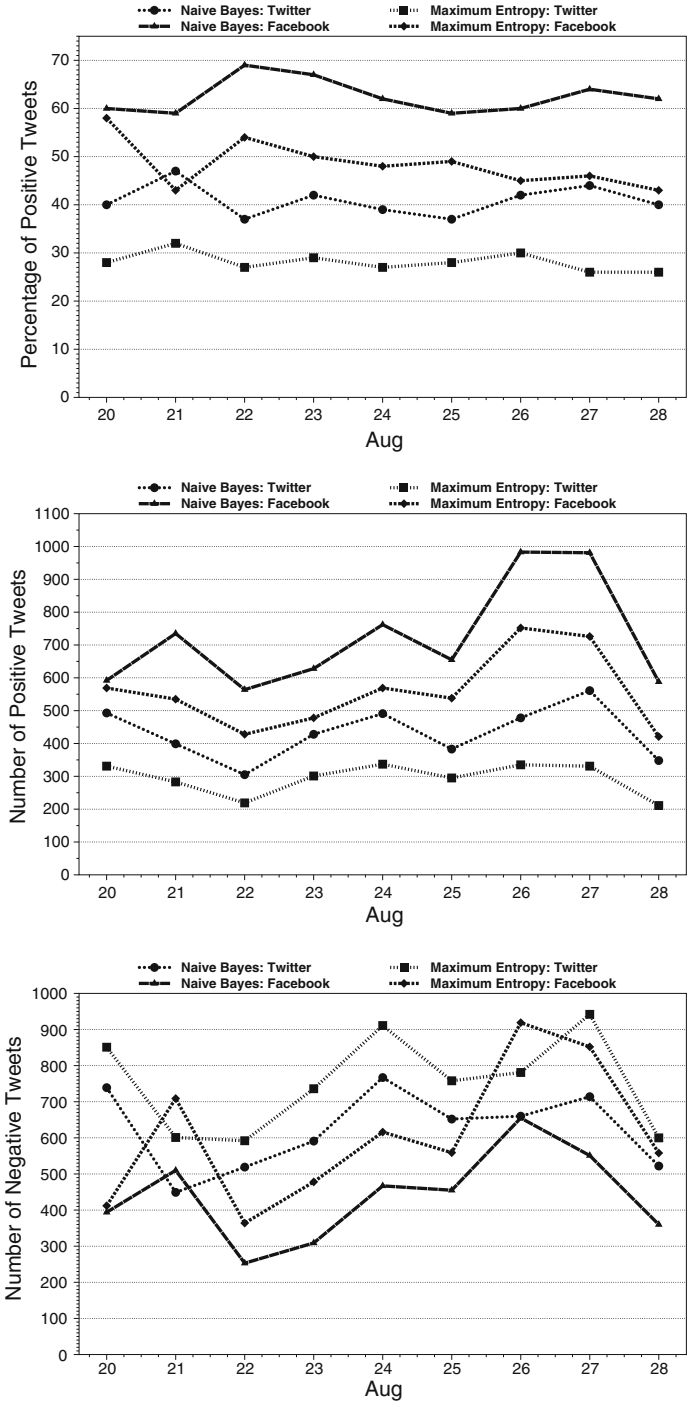


Fig. 2.9 Charts produced by system

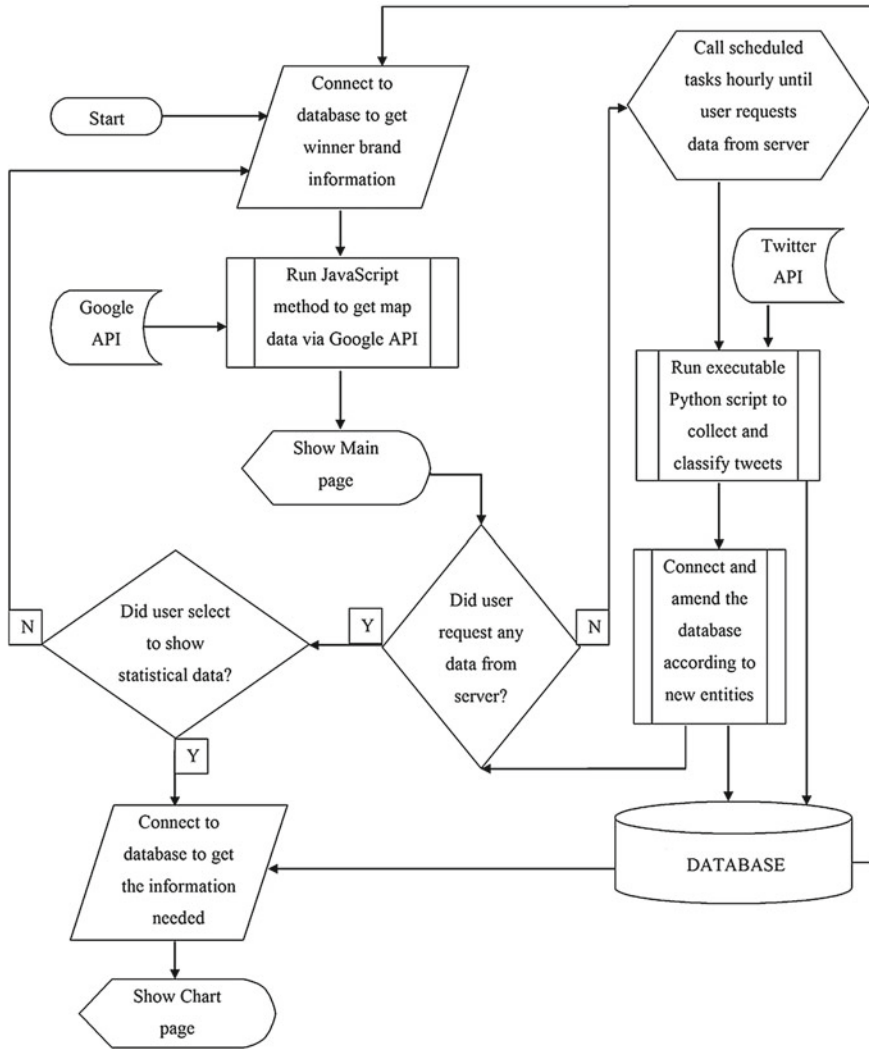
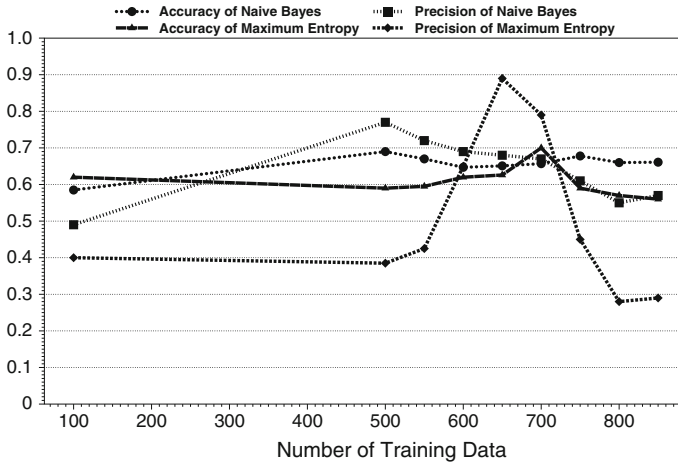


Fig. 2.10 Flowchart of the web interface

dataset is used, and the number of data in the training set is increased continuously. Figure 2.11 shows the accuracy and the precision of classification methods.

As it can be observed from the Fig. 2.11, the precision of the Maximum Entropy reaches its optimum level at the point where the number of training data indicates 650. It reveals almost no change from 550 training data to 100 training data.

On the other hand, the accuracy level of the Maximum Entropy does not demonstrate a considerable change. It reaches its peak level at 700 training data point.



**Fig. 2.11** Accuracy and precision of classification methods through number of training data

The accuracy of the Naïve Bayes pursues nearly a straight line from 850 to 600 training data. It reaches its maximum level at 520 training data, and then follows a slow decrease.

The precision level of the Naïve Bayes increases from 850 to 520 training data, and reaches its optimum level at 520. Afterward, it demonstrates a decreasing trend.

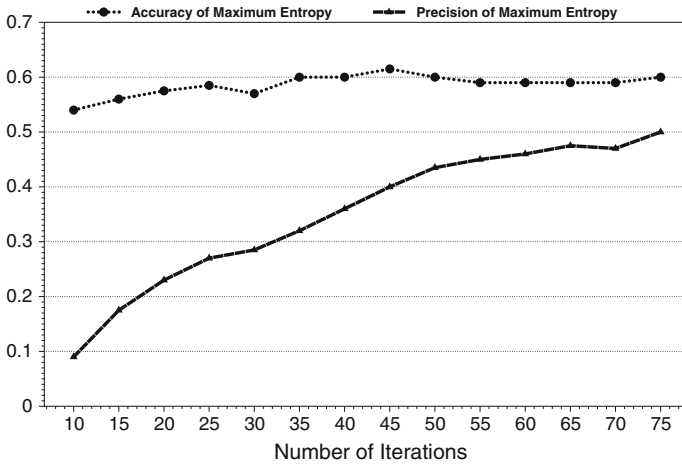
### 2.4.2 Performance of Maximum Entropy by Iteration Number

In this section, how the Maximum Entropy classification method of NLTK is affected by the change in the number of iterations is demonstrated. The number of iterations is set as 10 at first, and then increased by 5 until 75. The training dataset and the testing dataset are not changed, the only altered element is the iteration number of the Maximum Entropy method. The Fig. 2.12 shows the accuracy and precision values of the Maximum Entropy method.

As one can spot from the figure, the accuracy value of the Maximum Entropy demonstrates an increasing pattern until the point where the iteration number reaches 25, and then it goes through a slight decrease; however, it increases until the point where the iteration number indicates 45 where the accuracy value reaches its peak level. After the peak, the value pursues a slowly decreasing path. After the iteration number shows 50, the accuracy value of the Maximum Entropy does not change almost at all regardless of the increase in the number of iterations.

The precision value of the Maximum Entropy demonstrates a steady increase until the point where the number of iteration reaches 65 although its pace decreases for some time. It shows discontinuation for a short period of time between the iteration





**Fig. 2.12** Accuracy and precision of Maximum Entropy

number levels 65 and 70, and then it follows an increasing trend. Considering this, it can be concluded that an increase in the number of iteration affects the precision value positively.

### 2.4.3 Sentiment Mining on Tracking Results

A database is produced and classified by our Twitter sentiment tracking system. It has 362,529 automatically classified tweets collected by tracking in 9 days via the Twitter API.

The tracking process that commenced on August 20 was finalized on August 28. Throughout the 9 days period, recent news that might have an impact on the views of Twitter users on the four tracked brands (Facebook, Twitter, Apple, and Microsoft) are screened.

During this time frame, Microsoft announced on August 23, 2012 that it renewed its 25-year-old logo. A lot of tweets with regard to the new logo were collected. With the spread of the news, it was observed that in 11 cities among 17 cities, the percentage of positive tweets of Microsoft increased on August 24.

Particularly, the increase in four cities was remarkable. The increase in one of the cities, Miami, can be observed from Fig. 2.13.

### 2.4.4 Constancy of Tracking Results

When the tracking result charts are monitored, it is seen that, in general, there are no big jumps or declines in the percentage of positive tweets. This stability case

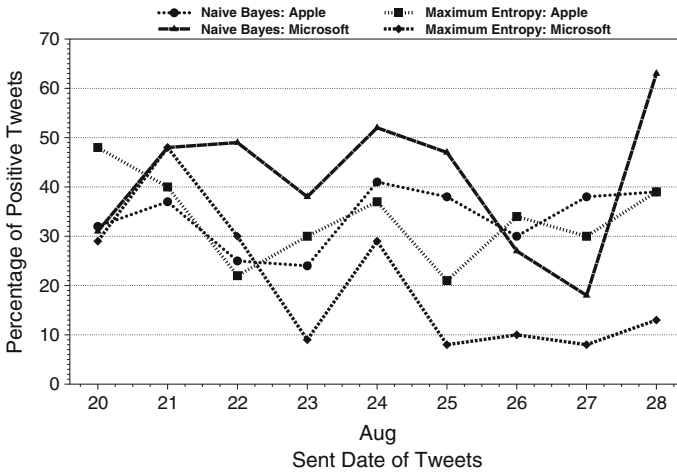


Fig. 2.13 Increase of percentage of positive tweets of Miami on 23 August

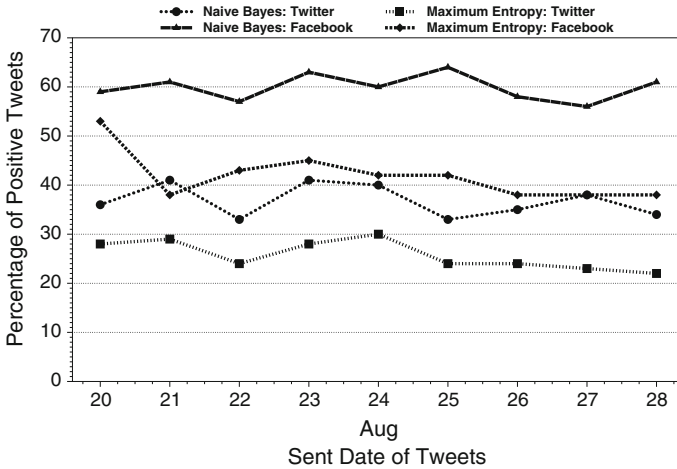


Fig. 2.14 Percentage of positive tweets of London

can be better observed particularly in the cities where a higher number of tweets are collected such as London and New York. As it can be seen from Fig. 2.14, even though the classification method changes, the percentage of positive tweets does not change much.

As a result, the increase of data; in other words, the increase in the number of collected tweets, renders the results produced by the system more reliable. In this way, the system can generate more stable results.

## 2.5 Conclusion and Future Works

### 2.5.1 Conclusion

Twitter is one of the most popular micro-blogging and social networking services. It allows visitors to read and post short messages limited to 140 characters. With its increasing popularity as a micro-blogging system, Twitter has become one of the best ways of monitoring the views of the users regarding certain products or things, in general. What is more, the enhanced use of the system and sharing of the users' views about specific matters before the actual date of their appearance as a concrete happening make it possible to make predictions by analyzing the current tweets. Twitter helps to identify the negative and positive opinions about a brand or a product. In order to manage the analysis of the users' posts about certain issues or things, a web-based tracking sentiment system for Twitter is developed which is able to satisfy the requirements of Carl described in the use case section.

In Sect. 2.2, first, blogging and micro-blogging are described. Afterward, Twitter and statistical data about Twitter are presented. Then, a short story of the Natural Language Processing is provided. The concept of sentiment analysis is also deeply analyzed and the Naïve Bayes and Maximum Entropy classification methods are briefly explained.

The technologies used in this chapter are laconically introduced in Sect. 2.3. First, Python and Natural Language Tool Kit are introduced. Afterward, the properties of Twitter and the Twitter API are elaborated, followed by Google API. The implementation section also covers the subsections: hand-classified dataset, background processes, and web interface. The hand-classified dataset includes 1,035 hand-classified tweets as positive and negative. To enhance the performance of our classification methods, tweet texts in the training dataset and the tweets collected via Twitter API are preprocessed. How our Python script collects and classifies tweets is represented in the subsection of background processes. In this section, basic properties of our Web Interface are described. How our system performs scheduled tasks, such as hourly tweet collection, is also expressed in this section.

Some evaluations are made in Sect. 2.4 to test and increase the performance of the Naïve Bayes and Maximum Entropy classifiers. For example, the performance of the Naïve Bayes and Maximum Entropy classification methods are evaluated by altering the number of training data. How the Maximum Entropy classification method is affected by the change in the number of iteration is shown as well.

It is also demonstrated that the Naïve Bayes classification and the Maximum Entropy classification methods can be used to conduct the sentiment analysis, but Maximum Entropy method is quite slow during the training process in comparison to the Naïve Bayes. Furthermore, 362,529 tweets are collected and automatically classified, which is described in Sect. 2.3. While collecting the tweets about the four companies in 9 days, we noticed that current news about brands have an impact on the views of Twitter users. During tracking, on August 23 Microsoft announced that the brand logo was changed. A lot of tweets were collected about that news and it was

observed that in 11 out of 17 cities, the percentage of positive tweets on Microsoft increased on August 24.

### 2.5.2 Future Works

As future works, the number of training dataset could be increased. This will pave the way for performance-enhancing results, particularly for the Naïve Bayes classification. The preprocessing step can be further developed. To illustrate, emoticons can be used for sentiment analysis. For example, an emoticon for smiling could be placed with SMILE or “hahaha” can be altered to LAUGH. On the other hand, the Support Vector Machine method might be another method in addition to the Naïve Bayes and Maximum Entropy classifier methods. Furthermore, the tracking process may not be confined to in days. With a longer tracking period, a better and more effective data mining can be implemented. This system can be also modified in order to monitor the views of the electors on the candidates in the elections in each city. Or, how a person, institution, or opinion is perceived in different parts of the world can be tracked with a modified version of the system.

**Acknowledgments** The first author has been funded by the Ministry of National Education, Republic of Turkey.

### References

1. K.M. Al-Aidaroos, A.A. Bakar, Z. Othman, Medical data classification with Naive Bayes approach. *Inf. Technol. J.* **11**(9), 1166–1174 (2012)
2. D. Allard, D. D’Or, R. Froidevaux, An efficient maximum entropy approach for categorical variable prediction. *Eur. J. Soil Sci.* **62**(3), 381–393 (2011)
3. S. Asur, B.A. Huberman, Predicting the future with social media, in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 1 pp.492–499 (2010)
4. F. Benamara, C. Cesarano, A. Picariello, D.R. Recupero, V.S. Subrahmanian, Sentiment analysis: adjectives and adverbs are better than adjectives alone, in *ICWSM* (2007)
5. A.L. Berger, V.J. Della Pietra, S.A. Della Pietra, A maximum entropy approach to natural language processing. *Comput. Linguist.* **22**(1), 39–71 (1996)
6. E. Boiy, P. Hens, K. Deschacht, M.-F. Moens, Automatic sentiment analysis in on-line text, in *ELPUB*, pp. 349–360 (2007)
7. J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market. *J. Comput. Sci.* **2**(1), 1–8 (2011)
8. J. Chen, H. Huang, S. Tian, Feature selection for text classification with Naive Bayes. *Expert Syst. Appl.* **36**(3), 5432–5435 (2009)
9. A. Devitt, K. Ahmad, Sentiment polarity identification in financial news: a cohesion-based approach, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (2007)

10. N.A. Diakopoulos, D.A. Shamma, Characterizing debate performance via aggregated Twitter sentiment, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'10* (ACM, New York, 2010), pp. 1195–1198
11. B.F. Green, A.K. Wolf Jr, C. Chomsky, K. Laughery, Baseball: An automatic question-answerer, in *Papers Presented at the 9–11 May 1961, Western Joint IRE-AIEE-ACM Computer Conference, IRE-AIEE-ACM'61 (Western)* (ACM, New York, 1961), pp. 219–224
12. S.C. Herring, L.A. Scheidt, S. Bonus, E. Wright, Bridging the gap: a genre analysis of weblogs, in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, January 2004*, pages 11 pp.- (2004)
13. B.J. Jansen, M. Zhang, K. Sobel, A. Chowdury, Twitter power: tweets as electronic word of mouth. *J. Am. Soc. Inf. Sci. Technol.* **60**(11), 2169–2188 (2009)
14. T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features* (Springer, New York, 1998)
15. K. Sang-Bum, H. Kyoung-Soo, R. Hae-Chang, H. Myaeng, Some effective techniques for Naive Bayes text classification. *IEEE Trans. Knowl. Data Eng.* **18**(11), 1457–1466 (2006)
16. H. Kwak, C. Lee, H. Park, S. Moon, What is Twitter, a social network or a news media? in *Proceedings of the 19th International Conference on World Wide Web, WWW'10* (ACM, New York, 2010), pp. 591–600
17. N. Li, D.D. Wu, Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decis. Support Syst.* **48**(2), 354–368 (2010)
18. T. Liu, W. Che, S. Li, Y. Hu, H. Liu, Semantic role labeling system using maximum entropy classifier, in *Proceedings of the Ninth Conference on Computational Natural Language Learning, CONLL'05* (Association for Computational Linguistics, Stroudsburg, 2005), pp. 189–192
19. E. Loper, S. Bird, NLTK: the natural language toolkit, in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics—Volume 1, ETMTNLP'02* (Association for Computational Linguistics, Stroudsburg, 2002), pp. 63–70
20. K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, in *IJCAI-99 Workshop on Machine Learning for Information Filtering*, vol. 1, pp. 61–67 (1999)
21. B. O'Connor, R. Balasubramanyan, B.R. Routledge, N.A. Smith, From tweets to polls: linking text sentiment to public opinion time series, in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (2010)
22. B. Pang, L. Lee, Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2**(1–2), 1–135 (2008)
23. D. Quercia, J. Ellis, L. Capra, J. Crowcroft, Tracking “gross community happiness” from tweets, in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, CSCW'12* (ACM, New York, 2012), pp. 965–968
24. E.S. Robertson, K.S. Jones, Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.* **27**(3), 129–146 (1976)
25. S.J. Rosenschein, S.M. Shieber, Translating English into logical form, in *Proceedings of the 20th Annual Meeting on Association for Computational Linguistics, ACL'82* (Association for Computational Linguistics, Stroudsburg, 1982), pp. 1–8
26. F. Rousseaux, K. Lhoste, Rapid software prototyping using Ajax and Google map Api, in *IEEE Second International Conferences on Advances in Computer-Human Interactions, ACHI'09*, (IEEE, 2009), pp. 317-323
27. M.F. Sanner, Python: a programming language for software integration and development. *J. Mol. Graph. Model.* **17**(1), 57–61 (1999)
28. K.-M. Schneider, A comparison of event models for Naive Bayes anti-spam e-mail filtering, in *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics—Volume 1, EAACL'03* (Association for Computational Linguistics, Stroudsburg, 2003), pp. 307–314
29. B. Sharifi, M.-A. Hutton, J. Kalita, Summarizing microblogs automatically, in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT'10* (Association for Computational Linguistics, Stroudsburg, 2010), pp. 685–688

30. S. Soderland, Learning information extraction rules for semi-structured and free text. *Mach. Learn.* **34**(1–3), 233–272 (1999)
31. A. Tumasjan, T.O. Sprenger, P.G. Sandner, I.M. Welp, Predicting elections with Twitter: what 140 characters reveal about political sentiment. *ICWSM* **10**, 178–185 (2010)
32. A.M. Turing, Computing machinery and intelligence. *Mind* **59**, 433–460 (1950)
33. K. Tzeras, S. Hartmann, Automatic indexing based on Bayesian inference networks, in *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'93* (ACM, New York, 1993), pp. 22–35
34. C.T. Yu, G. Salton, Precision weighting; an effective automatic indexing method. *J. ACM* **23**(1), 76–88 (1976)

## Chapter 3

# Health Assistance for Immigrants

**Till Plumbaum, Funda Klein-Ellinghaus, Anna Reeske,  
Kristin Pelz and Frank Hopfgartner**

**Abstract** Our personal health should be one of our main concerns, but unfortunately, due to modern lifestyle, far too many people ignore their own well-being. Consequently, methodologies need to be developed that assist us in living a healthier life. In this chapter, we present a health assistance system for immigrants. The system consists of two parts: A health information system that allows users to search for health information using natural language queries composed of multiple languages and a prevention service that assists users in their cooking routine and motivates them to perform frequent physical exercises. The information system uses NLP techniques to understand the user query, matches it to a health ontology we developed, and offers the user a comprehensive answer. The prevention service is embedded in a smart home environment. We present the technical details of both systems and show a user study to demonstrate that the system works well in providing highly relevant health information.

### Toward a Healthy Life

There were two things that Steven liked a lot about his work: Free coffee and the view from the kitchen window. The office kitchen was in the thirteenth floor, the coffee machine right next to the window. Indeed, it happened quite a few times that

---

T. Plumbaum (✉) · F. Hopfgartner  
Technische Universität Berlin, Berlin, Germany  
e-mail: till.plumbaum@dai-labor.de

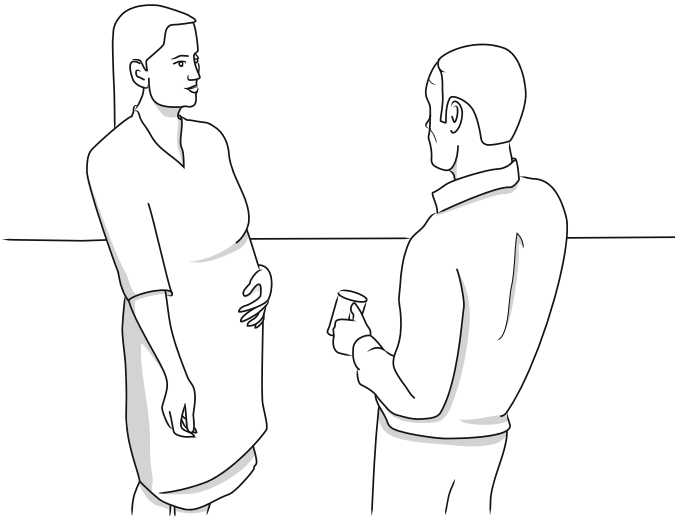
F. Hopfgartner  
e-mail: frank.hopfgartner@tu-berlin.de

F. Klein-Ellinghaus · A. Reeske  
Leibniz Institute for Prevention Research and Epidemiology, Bremen, Germany  
e-mail: kleinelf@bips.uni-bremen.de

A. Reeske  
e-mail: reeske@bips.uni-bremen.de

K. Pelz  
AOK Bundesverband, Berlin, Germany  
e-mail: kristin.pelz@bv.aok.de

he caught himself just gazing outside, observing the tiny looking people passing by in their tiny little cars. “We really live in a world of constant motion...”, he thought while he waited for this Latte Macchiato to come out of the coffee machine. “Good afternoon, Mr. Marks.” Steven turned away his eyes from the street outside to check who just welcomed him. It was Selma, the young new employee from Turkey. Steven turned around and straightened up. “İyi günler, Selma. Nasılsın?”—Good morning. How are you? These were pretty much the only few words he could say in Turkish. “Teşekkürler, iyiyim. Sen?”, Selma returned his greeting. “I am sorry, Selma, but I am way behind with my Turkish lessons. I am fine, if that’s what you have asked me?”, he responded, switching to English. Selma smirked. “Yes, I was asking how you are these days.” Steven “Okay, that’s an important sentence to know. I will try to remember it.” But you know, I am not the youngest any longer, picking up new languages is very hard.”



Of course Selma knew how hard it is to learn new languages. After all, it was only months ago that she and her husband decided to leave their beloved home in Ankara to start a new life in Germany. Although she enjoyed living in Ankara, the job offer that she received from here was just too good to reject. Working as a research engineer for a car manufacturer. Developing the cars of tomorrow. When they saw the job advertisement, both her husband and Selma were hooked. Cars had been their passion for a very long time. In fact, they first met each other at a motor show. And when they eventually got the job offer—both of them—there was no holding back anymore. In record time, they said good-bye to sunny Turkey and moved to Germany.

The first problems started shortly after though. Only a few weeks after settling in her new home, Selma started to feel sick in the morning and hence decided to see a medical doctor. His diagnosis was a little shock for them. Selma was pregnant! Her husband and she had tried for quite a while now to become pregnant but the idea of conceiving a child in a foreign country scared her. So many questions popped



up in her mind: Would she have to change her lifestyle to ensure that the baby can develop properly? Would she have to give up her job now or would she be able to get financial aid from the state? After all, she didn't know anything about Germany and its healthcare system.

In fact, it was Steven Marks who managed to reduce her panic level significantly. Being the deputy manager of the research department, he was one of the first people at work that she informed about her pregnancy. "Don't worry Selma", he said back then. "You can get paid maternity leave and have a guaranteed right to return to work after that. There should also be courses and information services that your health insurance provider provides." Indeed, only a few days after she reported her pregnancy to her health insurance provider, they send her a brochure about their online health care information system. According to their brochure, the system was designed to provide information in different languages and should help her in answering most pregnancy-related questions that would arise in the next few months. She tried out the system on the same day. Being able to query in Turkish, German, or English, the system provided her a lot of information about pregnancy in general and information about services she could participate in. Given that all information is provided in her mother tongue Turkish, Selma grew in confidence that they would manage their new life as young parents in this unfamiliar country.

Selma smiled. "Yes, I know very well, Mr. Marks", she responded and reached for a bottle of water from the fridge...

### 3.1 Introduction

Nowadays, more and more people suffer from so-called "Western diseases," i.e., health conditions that are caused by lack of exercise, poor eating habits, and unhealthy lifestyles [32]. While these diseases of affluence are a serious threat to our personal well-being, they are even more critical for pregnant women who are not only responsible for their own health, but also for the health of their unborn children. Addressing the need for healthier living, large education campaigns and seminars have been funded to promote healthier lifestyles, advertising low-fat diets, weight management, or physical exercises. With the increasing availability of the Internet, healthcare providers and governments extended the education campaign to the WWW, offering health information portals for multiple topics online. Two conclusions can be drawn from this provision of information. First of all, it is of high importance to provide accurate information, i.e., the content should be created by professionals and evaluated for accuracy [10, 20]. Secondly, such information sources should not be used for self-diagnosis; their only purpose should be to allow patients to better understand their physician's diagnosis. Although healthcare providers intend to provide health information services to all their clients, immigrants have been identified as vulnerable population [7] that benefit less from existing healthcare systems since language and cultural barriers prevent them from using existing prevention services. This is a problem especially for countries with significant numbers of immigrants

such as Germany, where roughly 20 % of the population has an immigrant background [5, 38] and consequently, a potentially large group of citizens may not be able to understand the main language of the host nation.

Apart from providing users easy access to health-related content, an equally important approach to support people in living a healthier life is to actively educate them. In the context of food consumption, this is done via nutrition facts labels that are legally required on most packaged food. Although different regulations exist in individual countries on what has to be written on these labels, they serve as guidelines on different targets for nutrients such as energy, protein, or fat. A study on the effects of these nutrition fact claims on consumer product evaluation is presented by Keller et al. [19]. Although providing overviews of ingredients helps raising awareness about individual eating habits, the share of people who cook their own food is declining. At the same time, we can observe a significant growth of the ready-meals market. Van der Horst et al. [42] study the association between overweight, cooking skills, and ready meal consumptions. Their study illustrates the importance to actively promote healthy food preparation.

The third important aspect of healthy living is frequent physical exercises. Various studies (e.g., [44]) report a direct connection between physical activities and personal well-being. Regular physical activity is a resource for body and soul [39]. On the one hand, a physically active lifestyle can contribute to reduce the risk of cardiovascular diseases, obesity, and complaints of the muscular and skeletal system [1]. On the other hand, regular physical exercise can reinforce the mental well-being. The World Health Organization describes lack of exercise as the fourth important risk factor for mortality [46]. They recommend adults a medium intense activity of 2, 5 h/week. As mentioned before though, lack of physical activity is one of the main consequences of a sedentary lifestyle.

In order to address the issues mentioned above, we introduce a health assistant for immigrants [30] that consists of two subsystems, namely a multilingual health information service and a prevention service. Figure 3.1 illustrates the connection between these individual components. Combined, the services are a comprehensive approach to support people for healthier living by giving them information about health topics, supporting healthier eating, and getting enough exercises.

The multilingual health information service [29] guarantees personalized access to professionally created healthcare content. The system, in the remainder of this chapter referred to as GID, enables immigrants to inform themselves about medical conditions and preventive health services. By providing them with available information in the language of their choice, GID helps those people who have language-related difficulties in understanding their physicians. Information is adapted based on users' personal context such as pre-existing medical conditions and their location.

The prevention service, referred to as PS, consists of a cooking assistant and a virtual trainer. The cooking assistant provides the user a wide range of different recipes and nutrition information about different meals. There is also a possibility to personalize the recipe search with the aid of different criteria. The trainer supports the physical activity of the user with three different activity sessions.

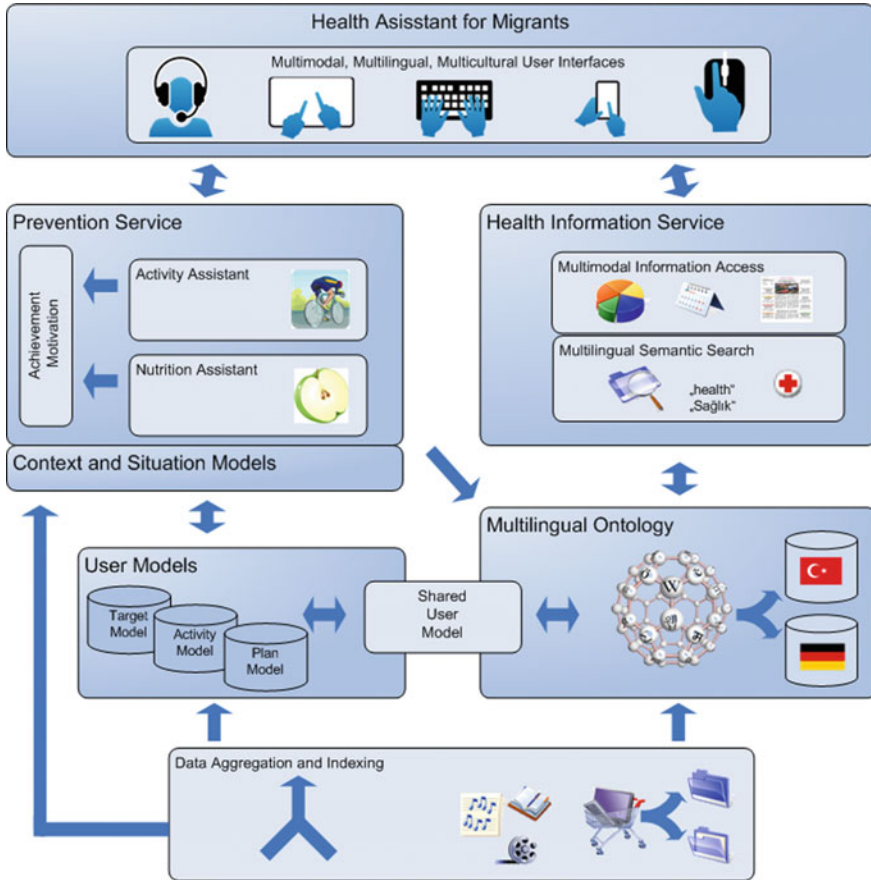


Fig. 3.1 System overview of the health assistant and its parts

This chapter is structured as follows. In Sect. 3.2, we discuss the health services and survey existing work that is related to our system. In Sect. 3.3, we introduce the system infrastructure of the health information system and outline the scientific challenges that it addresses. We then present in Sect. 3.4 an initial evaluation of the system. Following that, we present in Sect. 3.5 the prevention service. A discussion and conclusion is given in Sect. 3.6.

### 3.2 Related Work

This work touches upon various research areas, including the use of online health information services, knowledge bases, the identification of user context, semantic information management, healthy nutrition, and physical activity. In the remainder of this section, we discuss related work on these research aspects.

### 3.2.1 *Online Health Information Services*

With the growing importance of the Internet, one could also witness the growth of online health-oriented information platforms such as PubMed<sup>1</sup> and Scopus,<sup>2</sup> which enable researchers and professionals to check up on latest research results on bio-medical topics [8]. While these databases provide detailed access to state-of-the-art research results, they are less suitable for the general public who would like to check up on symptoms that they are experiencing. According to Morahan-Martin, up to 4.5 % of all Internet searches are about health-related topics [25], indicating the significance of this topic in our life. An overview of different resources in the Web that can be used for this information gathering task is provided by Johnson et al. [18]. Generally, three types of services can be identified: (a) Professionally maintained health advice and information services where users can check their symptoms in a constantly updated database (e.g., the NHS Direct<sup>3</sup> service maintained by the English National Health Service) or check up on public service announcements (e.g., by the World Health Organization), (b) unsupervised sources such as Wikipedia and (c) discussions of similar cases in online forums, blog posts, or biased advertisements for specific products that can be retrieved by standard Web search engines. From a medical point of view, relying on such sources is not recommended and the consultation of a professional is highly advised. Therefore, the American Medical Library Association recommends to “trust your physician, not a chat room” [9]. Morahan-Martin suggests to approach this unconsidered information handling by asking physicians to point their patients to reliable health portals and to work on improving such sites, e.g., by improving search and retrieval techniques. Our works build on her suggestion.

### 3.2.2 *Computer-Supported Knowledge Bases*

In the healthcare domain, an exact and unambiguous definition of diseases, symptoms, etc., is indispensable. Only when it is clear what the problem (disease) is, physicians can say how to treat it. Given the complex nature of this topic, physicians have to rely on an extensive knowledge base. Traditionally, this knowledge base has been maintained in books and journals but since the introduction of computer-based knowledge systems, methods have been developed that support physicians in the anamnesis process. A promising method that allows for a structured processing of data is the use of ontologies. Ontologies define “the concepts, relationships, and other distinctions that are relevant for modeling a domain” [11]. Ontologies, in the context of computer and information sciences, are also machine-readable and sharable. Thus, ontologies

---

<sup>1</sup> <http://www.ncbi.nlm.nih.gov/pubmed/>.

<sup>2</sup> <http://www.scopus.com/>.

<sup>3</sup> <http://www.nhsdirect.nhs.uk/>.

represent an ideal basis for an intelligent healthcare information system. In the following, we present some approaches to create a unified model in the health domain.

One of the biggest computer-supported biomedical knowledge bases is the Unified Medical Language System (UMLS),<sup>4</sup> a platform that provides a unified vocabulary for biomedical and health terms. UMLS is maintained by the US National Library of Medicine and updated on a frequent basis. The knowledge base of UMLS consists of three components: (a) Biomedical terms from various controlled vocabularies (such as SNOMED-CT<sup>5</sup>) are defined in a Metathesaurus as semantic concepts. (b) The semantic relationship between these concepts are defined in a Semantic Network and the concepts are categorized into broad categories (semantic types). (c) Syntactical, orthographic, and morphological information about the biometric terms are defined in a lexicon. UMLS has a strong focus on the US health market, and hence, multilingual aspects are not fully supported, making it difficult to apply it to the scenario that is outlined in this chapter.

Other health-related ontologies include MEDCIN and SNOMED-CT, two classification systems used to store patient health records. The International Health Terminology Standards Development Organisation (IHTSDO) promotes SMOMED-CT as a standard for health records. The aforementioned approaches define common vocabularies and their relation to each other for the medical domain, but they define the terms in a proprietary format that is not machine-readable and shareable. These constraints make reusing and sharing of data complicated though. OpenGALEN [33, 37] is a medical ontology developed in the European GALEN project. OpenGalen offers a comprehensive knowledge base of medical terms and relations. It provides three types of ontologies: A high-level ontology, defining general structures, a common reference model defining the reusable parts intended to be shared between ontologies, and extensions for subdomains and specific use cases. The GALEN ontology is available under an open source license.

The presented approaches all contribute to the goal of a common, shareable, and reusable notion of the medical domain for building health-related applications. Nevertheless, none of these approaches fulfill the requirements to serve as the basis for a multilingual health information system. In particular, they lack modeling of multilingual descriptions and a connection to related health information is not available.

### ***3.2.3 Identification of User Context***

An important feature of online information systems is to adapt information based on users' individual requirements [17]. Requirements can depend on different factors such as the users' demands and context. In the health domain, contexts such as health condition of the user and their location plays an important role. For instance, recommendations for a healthy diet for a person with Type 2 diabetes and a pregnant

---

<sup>4</sup> <http://umlsinfo.nlm.nih.gov/>.

<sup>5</sup> <http://www.ihtsdo.org/snomed-ct/>.

person differ significantly and thus should be reflected in the information that is tailored to the users' needs.

Tailoring information based on these user contexts requires storage of user-centric information. In order to easily connect this information with ontology-based knowledge bases, i.e., for providing a personalized healthcare system, a promising approach is to model information using user-centric (semantic) ontologies. Semantic-based user ontologies are for instance:

- FOAF: The Friend-of-a-Friend model is an RDF-based user model mainly designed for the web. It defines demographic data such as name, age, and friendship relations. Common applications of FOAF are social networks services [3].
- SWUM: The Semantic Web User Model defines a comprehensive user model designed for the needs of the modern social semantic web. It models information about the users' demographics, friendship, etc. [31].
- GUMO: The General User Modelling Ontology tries to provide a user model covering all aspects of life. GUMO models health related information such as blood pressure or temperature [15].

In this work, we introduce an approach to map users' demands (as expressed by the search query) with the computer-based knowledge basis under consideration of the context that is stored within a semantic-based user ontology, thus presenting a healthcare information system that can provide answers based on context.

### ***3.2.4 Multilingual Semantic Information Management***

As argued above, one main challenge in the described scenario is to deal with different languages that might be relevant to provide health-oriented information, e.g., the difference between the patients' mother tongue (*Language A*) and the language used by the physician (*Language B*). Multilingual information management has been studied extensively in the literature. A straightforward approach to cross-lingual information management is either a direct translation approach [27] or using an inter-lingual mapping like EuroWordNet [43], which have been shown to work well at the CLIR task [6]. Imprecise translations are acceptable for the retrieval performance, as the document search itself is more important than disambiguation of individual translated terms [16]. Other approaches map individual documents into a higher dimension semantic feature space that is uniform for different languages. Thus, it is possible to map similar documents to nearby points in the feature space, even if they do not share a language. Sorg and Cimiano [34] have exploited the explicit links between related Wikipedia articles of different languages to map documents to a Wikipedia feature space in which documents are considered similar when they are semantically similar to the same set of articles. A similar approach is to map documents to multilingual ontology concepts, which can be represented as points in a feature vector space [12]. In this work, we employ such ontology mapping method to link concepts expressed in different languages.

### ***3.2.5 Graph-Based Search***

The advantage of ontology-based knowledge bases is that they can be exploited to identify related concepts. A common approach to identify these related concepts is to apply graph-based search algorithms. These algorithms infer links between ontology nodes that are not explicitly stored in the ontology (e.g., [36]). Most graph search algorithms (e.g., [40, 41]) are based on breadth-first search or depth-bounded depth-first search. These algorithms find related nodes by calculating a relatedness score to the input nodes. A high relatedness means that the nodes can be reached by several parallel short paths [22]. The scoring function should consider the specific properties of the ontology domain, taking into account edge weights and semantics while computing the path weights. The nodes with the highest relatedness score are considered to be most similar to the input nodes. Within this work, we apply graph-based search to retrieve relevant health information based on the users' search query and their personal context.

### ***3.2.6 Healthy Nutrition***

Unhealthy nutrition is one of the main reasons for diseases of affluence and, consequently, there is a grand need to convince people to choose healthy food rather than convenience food. Unfortunately, people are constantly facing advertisement campaigns whose main purpose is not to sell healthy products [13]. Active measures against this situation are information campaigns. This includes approaches such as the traffic light rating system, i.e., the regulation that food producers and providers have to clearly state on their product how healthy their product is. Other approaches include marketing campaigns which are often financed by the government or health insurance companies. While these are considered to be rather passive information campaign, another approach is to actively assist the people to prepare their food. This can be done in the form of cooking classes where exclusively healthy food options are taught [4], but also in the form of software systems (e.g., [14]) that take over the task of the chef instructor. In this work, we introduce a software-based nutrition assistant that assists users in preparing healthy food.

### ***3.2.7 Physical Activities***

Various studies suggest that personal achievement is one of the main driving forces behind sports activities. Nicholls [26] argues, for example, that one of the main reasons for the success of competitive sports such as running, tennis, or swimming is the possibility to directly compare one's physical abilities with others. Another motivation is to experience (and to extend) physical limitations. This can in particular be observed in extreme sports such as bungee jumping, base diving, or other dangerous

sports. He refers in this context to task-oriented and ego-oriented sports. In both cases, individual achievement, either by outperforming others or by reaching new limits, is the main reason to perform sports. Hence, in order to motivate people to get more physically active, personal aims need to be identified and targeted. Members of the Quantified Self movement rely on the power of numbers to measure personal achievements. By recording their physical activities using step counters, accelerometers, or other wearable sensors, people can directly measure how far they are from reaching their personal aims (e.g., [35]). A promising method to help people in determining what to achieve is to rely on badges, points, or leaderboards, i.e., principles that have shown to be very successful in games (e.g., [24, 45]). The use of these principles in a non-gaming environment is commonly referred to as gamification.<sup>6</sup> Within this work, we aim to motivate people to get physically active by providing a virtual trainer. This trainer teaches the user how to perform various exercises. Whenever a user successfully repeats an activity, they receive points as an indication of personal achievement. Moreover, a personal activity list is created, thus allowing users to compare their achievements with other users of the system.

### 3.3 Semantic Health Information System

The following sections describe the main components of the multilingual health information system. The system builds on semantic technologies to perform the task of multilingual information supply. We introduce the underlying ontology that allows us to generate a semantic knowledge base of health information. Furthermore, we introduce the data acquisition and management tasks, the user modeling technique, supported querying modes, and introduce the user interface.

#### 3.3.1 *The Health Ontology*

In order to provide a knowledge base that can easily be maintained by a computer system, we define a simple health ontology (HO) that defines basic health concepts and their relations. Figure 3.2 provides an overview of the ontology, detailed descriptions of the concepts and relations are listed in Tables 3.1 and 3.2. The concepts in the Health Ontology are enriched with multilingual labels, e.g., the concept *pregnancy* is labeled with the Turkish term “gebelik” and the German expression “Schwangerschaft.” Besides, relevant documents are attached to the concept nodes. In a healthcare scenario, information quality is of crucial importance. False information not only leads to a loss of trust but also may lead to serious harms. Therefore, information in the HO should be maintained by a group of experts. In the GID project, physicians maintained information in the ontology.

---

<sup>6</sup> A detailed overview of gamification is provided in Chap. 9.



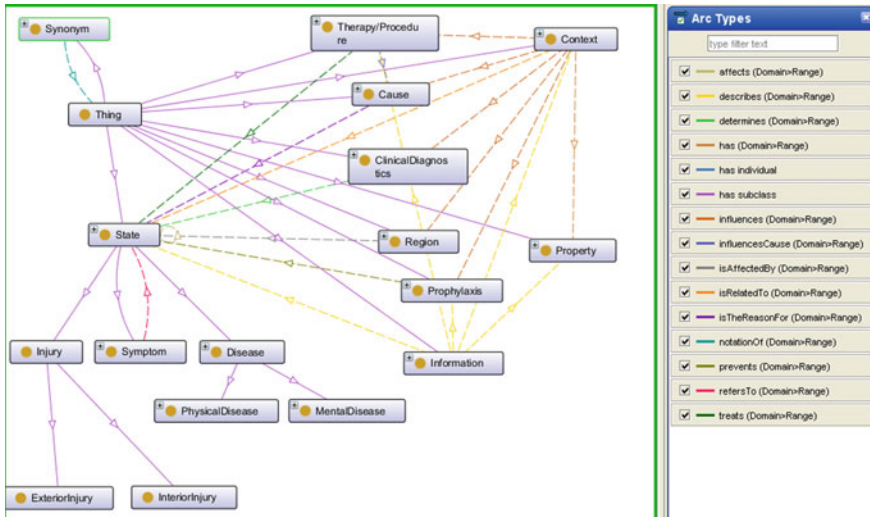


Fig. 3.2 The GID health ontology

### 3.3.2 User Model

As explained above, personalization services rely on the creation of user models to store personal information in a user profile. User profiles are then exploited to adapt information accordingly. In the health scenario, the most important factors are the users’ demographic details (i.e., age, sex) and their personal context (e.g., their hometown, language knowledge and pre-existing medical conditions). In order to receive personalized health information, users of the GID system are required to provide the above information.

### 3.3.3 Multilingual Semantic Search

As explained above, concepts in the health ontology are enriched with multilingual labels. When a user enters a search query, we process the entered text by lowercasing it and pruning unusual characters. The search terms are then used as a query to retrieve matching concept labels contained in the Health Ontology. We do not perform language-specific stemming or remove stop-words of the search input since this enables us to easily extend the GID system with additional languages without requiring any changes to program code. Rather, we use a fuzzy search based on the open-source information retrieval system Lucene<sup>7</sup> to cover slight alterations in term surface forms. Taking the multilingual search query “*Hamilelikte hangi besinleri yememeliyim?*” (Translation: “*Which food must I not eat when I am pregnant?*”)

<sup>7</sup> <http://lucene.apache.org/>.

**Table 3.1** Concepts of the health ontology

Concept	Description
Context	This concept describes a composite information node composed of property nodes and state nodes. A context concept for instance can describe that certain therapies are only applicable for pregnant women
Property	Is connected to a context node and specifies the context, for instance a certain pregnancy week
Therapy/Procedure	This node describes possible therapies to cure a disease
Cause	Cause nodes describe the reason for a medical condition, e.g., obesity is a cause for diabetes
ClinicalDiagnostics	Describes an approach to make a diagnosis. For instance to take an x-ray of somebody
Region	Part of the human body plus psych. Indicates where a disease or injury is situated
Prophylaxis	This node describes prevention procedures, e.g., regular teeth-brushing helps preventing cavities
Information	The information node is the node that is displayed to the user with information about the node it is connected too. This can be information about diabetes, sport, pregnancy, or others. Information nodes can be a text or website, a video or picture
State	State is the general node to describe a condition of the body. Sub-nodes are injury or disease
Injury	Injury is a sub-node from State. It is splitted in inner and exterior injuries
InteriorInjury	Internal Bleeding is an example for an InteriorInjury node
ExteriorInjury	An abrasion is an example for exterior injuries
Symptom	Symptom nodes define possible clinical signs that indicate a State
Disease	Disease is a sub-node from State and is split into mental and physical disease
MentalDisease	This node comprises all diseases connected to a humans psych. For example, depression is a MentalDisease
PhysicalDisease	The PhysicalDisease node describes all body related diseases
Synonym	The synonym node is important for all multilingual aspects of the described system. It is explained later in Section Multilingual Semantic Search

as an example, the language-independent concepts *Pregnancy* and *Malformation* are identified, since they have been labelled using the keyword “pregnant.” Furthermore, the concepts *Nutrition* and *Alcohol* are identified by the label “besinleri.”

Using the retrieved query-relevant ontology concept nodes, we employ a graph-search to find conceptually related information nodes. Different concepts from ontology classes like diseases, diagnostics, and treatments are semantically linked with weighted edges in our HO. Our algorithm performs a full graph search along these edges, bounded in-depth. Information nodes found during this traversal are ranked based on the proportion to the edge-weights of the path of the originating concept and anti-proportional to that path’s length. Information nodes found via multiple paths receive the sum of the relevance values of all of those paths. Using the example query

**Table 3.2** Relations of the health ontology

Relation	Description
isTheReasonFor	Connects Cause and State nodes. For example, obesity is caused by bad eating habits
determines	Connects ClinicalDiagnostics and State. Application: Obesity is determined by measuring the BMI
Affects	The affects edge defines effects that one State has on another. For example, Obesity affects Diabetes
Prevents	The prevents edge connects the Prophylaxis and State node. A healthy diet prevents obesity
refersTo	refersTo connects Symptom nodes and State nodes, e.g., breathing problems indicate obesity
treats	The influences edge connects the context node with the Region, Prophylaxis, Therapy/Procedure, Cause and ClinicalDiagnostics node
influences	This node describes prevention procedures, e.g., regular teeth-brushing helps preventing cavities
isAffectedBy	Connects Region node with State node, such as Psyche is affected by depression
Describes	The describes edge connects information with the Property, Context, Prophylaxis, Therapy, State node. Thereby, we can add information (pdfs, websites, etc.) to a node
influencesCause	Connects Therapy/Procedure with Cause nodes. E.g., the treatment of respiratory distress is dependent on whether the cause is asthma or bronchitis
Has	has connects Context and Property nodes. This allows defining a special context. For example, the state pregnant can be constrained with the Property 9th month, to indicate that a treatment is only allowed in the 9th month
isRelatedTo	Connects context with state. See example above
Symptom	Symptom nodes define possible clinical signs that indicate a State
notationOf	Connects Synonyms with all type of nodes to add multilingual information

above, we find among others an information node concerning *Alcohol Consumption during Pregnancy*. The information node is ranked highest because it is linked very closely to all of the three health concepts found for our keywords. As mentioned above, information nodes in the HO have documents attached to them, which are used as search results. These documents can be of any language, and have a language tag attached. Direct translations of documents are marked as copies. In order to gather search results for visualization, we collect the documents attached to the most relevant found information nodes during the graph search step and rank the list of documents depending on their relevance values combined with the user's language preferences. Depending on these values, it is possible for users to find documents in a different language near the top of the list, if the document is not available in the preferred language but is highly relevant. Thus, users will always see a list of results balanced by relevance and their language preferences. The results on this list are independent of the language of the search query, since the results were found via a mapping to language-independent ontology concepts.

In above example, multiple documents of Turkish and German languages are attached to the information node *Alcohol Consumption during Pregnancy*. Assuming the user specified a Turkish language preference, the first search result presented to them would be the document “*Gebelikte alkol kullanımı*” (Translation: “*Alcohol consumption during pregnancy*”).

### 3.3.4 Graphical User Interface

In addition to the search result list, the GID Web User Interface provides several supportive UI elements that help serve the users’ information need and let them adjust the search. A screenshot is shown in Fig. 3.3. In the remainder of this section, we introduce different features of this interface.

Entering the website, the user can log in to receive context-based search results. On top of the interface, the user can type in a search query. Since GID matches search terms with concepts in the health ontology, the search query can be formulated in different languages. In the screenshot, the user Selma has logged in and typed in the search query used in the example above: “*Hamilelikte hangi besinleri yememeliyim?*” (Translation: “*Which food must I not eat when I am pregnant?*”) Under the search box, the interface lists the concepts that GID extracted from the search query. In the

The screenshot displays the GID web interface. At the top, there is a navigation bar with the GeM logo and three main sections: 'Başlangıç Ana Sayfa', 'SBH Sağlık Bilgi Hizmeti', and 'KH Koruyucu Hizmet'. Below this, the user 'Selma' is logged in, and the language is set to 'türkçe'. The search bar contains the query 'Hamilelikte hangi besinleri yememeliyim?'. Below the search bar, there are options to sort results by language ('deutsch', 'türkçe') and a search icon. A dropdown menu for 'Arama Sonuçları' is visible. The main content area shows a search result for 'Hamilelikte hangi besinleri yememeliyim?'. The result includes a summary: 'GID verdiğiniz soruya şöyle bir anlam vardı. Bilgi edinmek istediğiniz konu **Bişimsizlik, Gebelik, Alkolsüz yağlı karaciğer hastalığı, Gut** hakkında, **Beslenme, Gebelik** bağlamı çerçevesindedir. Bulunanlar listesi ikamet ettiğiniz yere göre ayarlandı: **Berlin**. Hasta dosyanızın bağlamı çerçevesinde arama yapın:  Diyabet Tip II'. Below this, there is a section titled 'Seçeneğinize bağlı şu cevapları verebiliriz:' with a sub-section 'Hamilelikte beslenme'. This section includes an image of a bowl of fruit and text: 'Hamile olduğunuz dönemde yediklerinizin, bebeğinizin büyümesini ve gelişmesini, vücudunuzun da oluşan değişikliklerle başa çıkabilmesi için yeterli enerji ve protein sağlayacak şekilde olmasına dikkat etmeniz gerekir. Bu makalede, hamile olduğunuz dönemde sağlıklı yiyecekler seçmenize yardımcı olacak pratik tavsiyeler bulunmaktadır.' Below this, there is a section 'Neler yemelisiniz?' with a list of recommendations: 'Apeşide sayılan türden çeşitli yiyecekler yemeye çalışmanız önemlidir.' followed by a bulleted list: '• Bol miktarda meyve ve sebzeler (baze, domuz, tenekle kutuda konserve, kurutulmuş veya bir bardak meyve suyu) – günde en az beş porsiyon hedefleyin', '• Bol miktarda ekmekek, pirinç ve patates gibi nişastalı yiyecekler', '• Öğdeğin yağsuz et ve tavuk, balık (biri yağlı balık olmak üzere haftada en az iki kez balık yemeyi hedefleyin), yumurta ve bakliyat (örneğin fasulye ve mercimek) gibi protein. Bunlar aynı zamanda iyi birer demir kaynağıdır.', '• Kahverengi ekmekek, makarna, pirinç, bakliyat ve meyve ve sebzelerde bulunan bol miktarda lif – (bu kasıtlı olarak denemeye yardımcı olur)', '• Süt, peynir ve yoğurt gibi, kalsiyum içeren süt ürünleri...'. Below this, there is a section 'Hamilelik hakkında bilmek istediğiniz herşey' with a bulleted list: '• GID'e sor', '• Konuya özel AOK'. At the bottom, there is a section 'Bilgi bankamızdan Web siteleri' with a link to 'GEBELİKTE ALKOL KULLANIMI' and a 'Sonuç Grafığı'.

Fig. 3.3 Screenshot of the GID search interface

example screenshot, the following concepts have been detected from the search query: malformation (Bıçimsizlik), pregnancy (Gebelik), non-alcoholic fatty liver disease (non-alkolik yağlı karaciğer hastalığı), and gout (Gut) in the context of nutrition (beslenme) and pregnancy (gebelik). Furthermore, the following context has been extracted from Selma’s profile that will be considered when ranking search results: home town (Berlin) and pre-existing medical conditions (Diyabet Tip II, Type II Diabetes). Under this visualization, the interface provides a topic box containing professionally edited information for the identified concepts. Under this topic box, the search results are displayed in descending order of relevance. The user can adjust their language setting by dragging a slider seamlessly between two languages. This sets a gradual preference of one language over the other and can exclude a language completely if dragged to the very end of the other language. Changing this setting immediately affects the search results and re-sorts the result list to reflect different language preferences. The user can seamlessly observe how search results are rearranged based on their language settings.

Each search result offers a button that when clicked shows a view of the paths of the Health Ontology that were traversed to find this search result. A screenshot is shown in Fig. 3.4. The semantic graph allows the user to understand *why* this search result is relevant to their search query. It also makes transparent the underlying workings of the ontology. Additionally, the interface enables users to visually

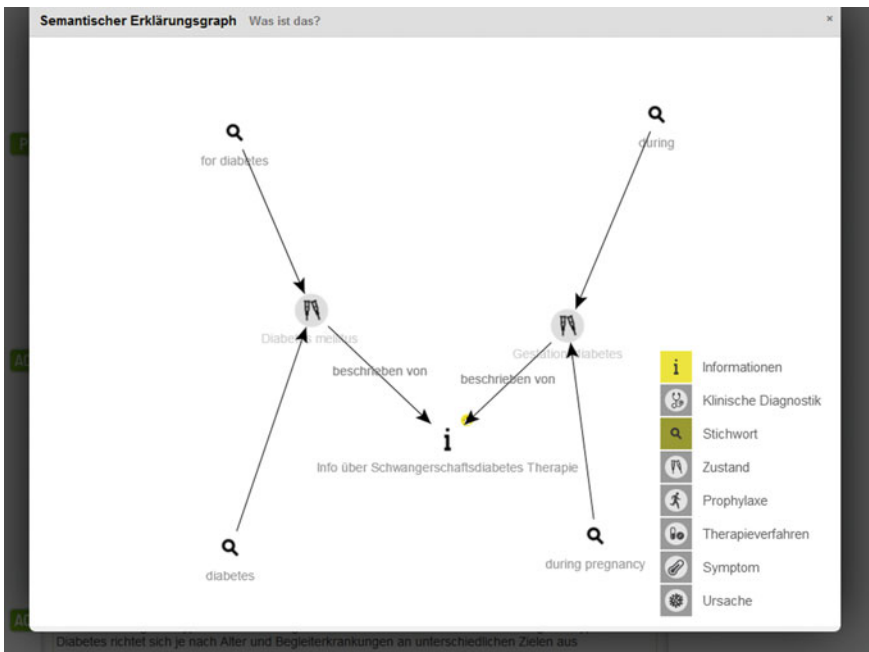


Fig. 3.4 Graph search explanation

browse through the ontology concepts by expanding concept nodes and traversing semantic edges between concepts, and also finding documents attached to a concept's information nodes.

### **3.4 Evaluation**

In order to assess immigrants' expectations and needs for a personalized health information assistant, we conducted a demand analysis among Turkish migrants in a mid-sized town in Germany. In addition, we evaluated the development and implementation of the health assistant. Applying a qualitative research approach, we recruited a small study cohort from the local Turkish community as well as a German control group. In the second phase of the evaluation, they were asked to interact with the system in a supervised scenario and were further asked to assess the assistant with respect to its technical implementation, relevance of the presented content, usability in daily life, and its potential to present health information in a structured manner. In particular, we focused on the following research questions: (a) How do the participants assess the adaptation of search results based on personal context? (b) How do they evaluate the system's handling of cultural differences? (c) How do they evaluate the usability of the graphical user interface (especially the ability to change language settings)? In the remainder of this section, we outline the setup of the user evaluation.

#### ***3.4.1 Participants and Recruitment***

Building on experiences from former migrant studies [47], we applied a cultural-sensitive recruitment concept using key persons from existing social networks (family and youth centers) in order to establish the access to our study population. We recruited nine families with a Turkish migrant background and four German families as a control group. As suggested by Lamnek [21] and Pelz et al. [28], we invited a homogeneous focus group of six participants with similar socioeconomic backgrounds from two different districts to participate in the user evaluation.

#### ***3.4.2 System Interaction***

During the user-centric evaluation of the system, the study group was separated into a German-speaking group and a Turkish-speaking group due to limited knowledge of the German language of some participants. Both groups interactively tested the health assistant prototype in a supervised user-scenario. During the test, they were asked several questions concerning the design, usability, and cultural sensitivity.

The moderated focus group discussion was held in Turkish language. First, the participants had the opportunity to summarize their impressions on the health assistant. Second, the discussion was moderated according to our interview guideline and focused mainly on the usability and the potential for the implementation and use in daily life. We compiled the main messages in a mind map during the focus group discussion for discussion structuring and subsequent data analysis.

### ***3.4.3 Evaluation Outcome***

For data analysis, the bilingual interviews and the focus group discussion records were transcribed and translated. The data were analyzed using the method of content analysis by Mayring [23]. A summary of the group discussion is summarized in the remainder of this section.

#### **How Do the Participants Assess the Adaptation of Search Results Based on Personal Context?**

Participants found the search results to be appropriate and clearly presented. It was suggested by some members of the group that the detailed answer box should contain specific advice on actions to be taken. Most participants found the adaptation of search results based on the context of a personal user profile useful and found that this feature helped in making an individualized search experience. However, some users were reluctant in providing certain pieces of profile information and wished to know specifically how they would be used.

#### **How Do They Evaluate the System's Handling of Cultural Differences?**

Users wished for specific information about the Turkish governmental health system, especially compared to the German system.

#### **How Do They Evaluate the Usability of the Graphical User Interface (Especially the Ability to Change Language Settings)?**

Performing semantic search was found to be intuitive by a large majority of users in the study. Many users entered full health-related questions, as intended, without needing any instructions. The feature to enter mixed-language queries was found to be very useful by all participants, and was considered to be of high importance for everyday use by the group. However, most users needed to be made aware of functionality, since it was not readily apparent from the interface itself.

Language preference settings were found to be especially helpful by people with little proficiency in the German language. At first however, the difference in functionality between the language preference slider and the language selection dropdown-box was not intuitive to some participants and needed to be explained.

### **Additional Comments**

Asking the participants for advice on how to improve the system further, they suggested that GID should also enrich their results with pictures and videos. They remarked that reading a lot of text makes interaction with GID unappealing. Additionally, we noted that the ontology graph helped younger users to understand how their search results were found, but did not help older users that much.

## **3.5 Prevention Service**

The second service of the health assistant is the prevention service, referred to as PS. This service consists of two subsystems focusing on nutrition and activity support for people: a cooking assistant and an activity assistant. In this section, we first introduce the cooking assistant that uses structured knowledge about food, healthy eating habits, and user information to build a personalized cooking and eating plan for a single user or a group of users. Additional services such as a food-shopping assistant complete a service that allows creating healthy eating behaviors. Then, we present the virtual trainer which uses 3D-camera techniques to track users doing exercises at home. To motivate users doing their sports, the trainer combines serious games to make it more fun with motivational parts such as the combination with the cooking assistant. If a user makes more sports, the cooking assistant gives positive feedback by allowing the user to choose “unhealthier” menus.

### ***3.5.1 Nutrition Assistant***

The nutrition assistant, shown in Fig. 3.5, helps users eat in a healthy way. Users can get a list of cooking recipes that are tailored to their physical activity, medical conditions, and cultural background. For example, if the user has diabetes, recipes with high sugar content are avoided. Besides, the interface provides an overview on how healthy their current lifestyle is with respect to physical activities and nutrition habits. This helps users to learn about the food they eat.

The assistant provides the user with different cooking sessions. The user can select their meal from a range of different recipes. For the selection they can choose different health criteria, their country, the level of difficulty, and the category of the dishes. In the category they can decide if they want a dessert, a main meal, or





Fig. 3.5 Screenshot of the nutrition assistant

a vegetarian meal. Furthermore, they can select if they eat alone or with another family member. Therefore, they have the possibility to preset the likes and dislikes or diseases of the persons. When connected to a smart home, the system can also be used to control the domestic appliances such as the stove or the microwave. After the step-by-step cooking sessions the dinner will be ready. In some cases the cooking sessions are video-tailored. Furthermore, the cooking assistant gives an overview of all ingredients needed for the meal. So the user has the possibility to create their own shopping list and can send it to one of the family members. The user also gets nutrition information about their meal so that they have an overview of the calories, protein, carbohydrates, sugar, fat, and fibers.

### 3.5.2 Activity Assistant

The activity assistant intends to defeat one’s weaker self. By using game mechanics and rewards, people are motivated to do physical exercises at home [2]. As shown in Fig. 3.6, users see a digital trainer who demonstrates an exercise (e.g., Jumping Jacks) that should be imitated by them. The user has the possibility to select his trainer of three different figures—women, man, or ogre. The trainer offers the user three different activity sessions. Users’ motions are tracked using a XBOX Kinect 3D camera and compared to the instructed movements. During the training sessions the virtual trainer is giving a feedback, if the exercises are performed correct or not. For each activity, the users can earn activity points. The more exercises the users



**Fig. 3.6** Activity assistant: The user, shown in the *upper right corner*, is captured by the XBOX Kinect 3D camera and should follow the exercises of the trainer (*left*)

perform, the more activity points they earn. These points are directly fed into the nutrition assistant, i.e., the food suggestion depends on the users' individual energy expenditure.

### 3.6 Conclusion and Discussion

In this chapter, we presented a health assistant that addresses the specific needs of immigrants. The system consists of two parts: A health information system and a prevention service. The health information system provides a search facility on a semantic database to assist people in finding health-oriented information such as details about services provided by the health insurance provider answers for specific health-oriented questions. The system is mainly designed to assist immigrants with limited knowledge of the national language used in their host country in finding relevant information. Therefore, the system allows users to formulate search queries in their mother tongue, the host's language, and in a mix of both languages. Addressing the specific immigration situation in Germany, we focused on the implementation of German and Turkish languages. Due to the structured data processing method that is introduced in this chapter, other languages can easily be incorporated. Search

results can be adapted based on the users' preferred language and other personal contexts such as the users' location or pre-existing medical conditions. The underlying technology of this system is a health ontology that has been introduced in this chapter. Relevant information is retrieved by exploiting semantic relations between different concepts in this ontology. In order to evaluate the usability of this system, we followed a qualitative analysis scheme. Discussions in a focus group following this evaluation indicate that the system can be employed to assist immigrants to find information in their own and in the host nations' language.

The second part of the system, the prevention service, consists of two parts. The first part is a cooking assistant that, based on users' profiles, assists the users in selecting healthy food as well as in preparing the meal. Apart from providing detailed information about the ingredients of various dishes, the system recommends meals on the users' individual requirements. The system provides a step-by-step guide on how to prepare the meal and allows the user to control the home appliances needed for cooking. Moreover, the prevention service offers an activity assistant that motivates users to perform exercises in a fun setting. Choosing from different virtual trainers, users have to repeat physical exercises. Users can earn points by accurately performing these exercises.

**Acknowledgments** This work was funded by the Federal Ministry of Education and Research (BMBF) under funding reference number 01IS10055A-C.

## References

1. S.N. Blair, H.W. Kohl, R.S. Paffenbarger Jr, D.G. Clark, K.H. Cooper, L.W. Gibbons, Physical fitness and all-cause mortality. A prospective study of health men and women. *JAMA* **262**, 2395–2401 (1989)
2. M.N.K. Boulos, Xbox 360 kinect exergames for health. *Games Health J.* **1**, 326–330 (2012)
3. D. Brickley, L. Miller, FOAF vocabulary specification 0.91. namespace document, FOAF project (2007)
4. B.J. Brown and J.R. Hermann, Cooking classes increase fruit and vegetable intake and food safety behaviors in youths and adults. *J. Nutr. Educ. Behav.* **37**, 2005.
5. S. Bundesamt. Bevölkerung und Erwerbstätigkeit. Technical report, 09 2012
6. P. Clough, M. Stevenson, Cross-language information retrieval using Eurowordnet and word sense disambiguation. *Adv. Inf. Retr.* 327–337 (2004)
7. K.P. Derose, J.J. Escarce, N. Lurie, Immigrants and health care: sources of vulnerability. *Health Aff.* **09**, 1258–1268 (2005)
8. M.E. Falagas, E.I. Pitsouni, G.A. Malietzis, G. Pappas, Comparison of pubmed, scopus, web of science, and Google scholar: strengths and weaknesses. *FASEB J.* **22**(2), 338–342 (2008)
9. S. Fox, L. Raine, Vital decisions: how internet users decide what information to trust when they or their loved ones are sick. Plus a guide from the medical library association about smart health-search strategies and good websites (2002)
10. C. Glenton, E.J. Paulsen, A.D. Oxmanm, Portals to wonderland: health portals lead to confusing information about the effects of health care. *BMC Med. Inform. Decis. Mak.* (2005)
11. R.T. Gruber, Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.* **43**(5–6), 907–928 (1995)
12. J. Guyot, S. Radhouani, G. Falquet, Ontology-based multilingual information retrieval *CLEF Workshop Working Notes Multilingual Track*, pp. 21–23 (2005)

13. J.C.G. Halford, J. Gillespie, V. Brown, E.E. Pontin, T.M. Dovey, Effect of television advertisements for foods on food consumption in children. *Appetite* **42**(2), 221–225 (2004)
14. R. Hamada, J. Okabe, I. Ide, S. Satoh, S. Sakai, H. Tanaka, Cooking Navi: assistant for daily cooking in kitchen, in *Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA'05* (ACM, New York, 2005) pp. 371–374
15. D. Heckmann, E. Schwarzkopf, J. Mori, D. Dengler, A. Kröner, The user model and context ontology Gumo revisited for future web 2.0 extensions, in *Proceedings of the International Workshop on Contexts and Ontologies: Representation and Reasoning (CO:RR)*, vol. 298 of *CEUR Workshop Proceedings*. *CEUR-WS.org*, ed. by P. Bouquet, J. Euzenat, C. Ghidini, D.L. McGuinness, L. Serafini, P. Shvaiko, H. Wache (2007)
16. D. Hiemstra, F. De Jong, Disambiguation strategies for cross-language information retrieval, *Res. Adv. Technol. DL*. 852–852 (1999)
17. F. Hopfgartner, J.M. Jose, Semantic user profiling techniques for personalised multimedia recommendation. *Multimed. Syst.* **16**(4–5), 255–274 (2010)
18. P.T. Johnson, J.K. Chen, J. Eng, M.A. Makary, E.K. Fishman, A comparison of world wide web resources for identifying medical information. *Acad. Radiol.* **15**(9), 1165–1172 (2008)
19. S. Keller, M. Landry, J. Olson, A. Velliquette, S. Burton, J.Craig Andrews, The effects of nutrition package claims, nutrition facts panels, and motivation to process nutrition information on consumer product evaluation. *J. Public Policy Mark.* **16**, 256–269 (1997)
20. O. Kuss, G. Eysenbach, J. Powell, E.R. Sa, Empirical studies assessing the quality of health information for customers on the world wide web: a systematic review. *J. Am. Med. Assoc.* **05**, 2691–2700 (2002)
21. S. Lamnek, *Qualitative Sozialforschung* (Beltz Verlag, Weinheim, 2005)
22. A. Lommatzsch, T. Plumbaum, S. Albayrak, A linked dataverse knows better: boosting recommendation quality using semantic knowledge, in *Proceedings Advances in Semantic Processing*, Wilmington, pp. 97–103 (2011)
23. P. Mayring, *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (Beltz Verlag, Weinheim, 2007)
24. J. McGonigal, *Reality is Broken: Why Games Make Us Better and How They Can Change the World* (Penguin Press, New York, 2011)
25. Janet M. Morahan-Martin, How internet users find, evaluate, and use online health information: a cross-cultural review. *Cyberpsychol. Behav.* **7**(5), 497–510 (2004)
26. J.G. Nicholls, *The Competitive Ethos and Democratic Education* (Harvard University Press, Cambridge, 1989)
27. D. Oard, A comparative study of query and document translation for cross-language information retrieval *Mach. Transl. Inf. Soup* 472–483 (1998)
28. C. Pelz, A. Schmitt, M. Meis, Knowledge Mapping als Methode zur Auswertung und Ergebnispräsentation von Fokusgruppen in der Markt- und Evaluationsforschung (2004)
29. T. Plumbaum, S. Narr, E. Eryilmaz, F. Hopfgartner, F. Klein-Ellinghaus, A. Reese, S. Albayrak, Providing multilingual access to health-oriented content, in *Proceedings of the 25th European Medical Informatics Conference, MIE* (IOS Press, 2014), pp. 393–397
30. T. Plumbaum, S. Narr, V. Schwartze, F. Hopfgartner, S. Albayrak, An intelligent health assistant for migrants, in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth*, (IEEE, 2013), pp. 307–308
31. T. Plumbaum, S. Wu, E. William De Luca, S. Albayrak, User modeling for the social semantic webs, in *Proceedings Workshop on Semantic Personalized Information Management: Retrieval and Recommendation*, pp. 78–89 (2011)
32. M.T. Pollard, *Western Diseases: An Evolutionary Perspective* (CUP, Cambridge, 2009)
33. J. Rogers, A. Rector, Galen's model of parts and wholes: experience and comparisons, in *Proceedings of the AMIA Symposium*, pp. 714–718 (2000)
34. P. Sorg, P. Cimiano, Cross-lingual information retrieval with explicit semantic analysis, in *Working Notes of the Annual CLEF Meeting* (2008)
35. M. Swan, Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. *Int. J. Environ. Res. Public Health* **6**(2), 492–525 (2009)

36. B. Taskar, M.F. Wong, P. Abbeel, D. Koller, Link prediction in relational data, in *NIPS* (2003)
37. H. ten Napel, J. Rogers, Assessment of the Galen methodology on holistic classifications for professions allied to medicine. *Medinfo* **10**(Pt 2), 1369–1373 (2001)
38. United Nations, Department of Economic and Social Affairs. World population policies 2005. Technical report, 03 2006
39. U.S. Department of Health and Human Services. Physical activity and health: A report of the surgeon general. Technical report, CDC, 2006
40. D. Vallet, F. Hopfgartner, J.M. Jose, Use of implicit graph for recommending relevant videos: a simulated evaluation, in *ECIR*, pp. 199–210 (2008)
41. D. Vallet, F. Hopfgartner, J.M. Jose, P. Castells, Effects of usage-based feedback on video retrieval: a simulation-based study. *ACM Trans. Inf. Syst.* **29**(2) (2011)
42. K. van der Horst, T.A. Brunner, M. Siegrist, Ready-meal consumption: associations with weight status and cooking skills. *Public Health Nutr.* **14**, 239–245 (2010)
43. P. Vossen, *EuroWordNet: A Multilingual Database with Lexical Semantic Networks* (Kluwer Academic, The Netherlands, 1998)
44. D.E.R. Warburton, C.W. Nicol, S.S.D. Bredin, Health benefits of physical activity: the evidence. *CMAJ* **174**, 801–809 (2006)
45. J.R. Whitson. Gaming the quantified self. *Surveill. Soc.* **11** (2013)
46. World Health Organization, Global recommendations on physical activity for health. Technical report, WHO, 2010
47. Y. Yilmaz, S. Glodny, O. Razum, Soziale netzwerkarbeit als alternatives konzept für die rekrutierung türkischer migranten zu wissenschaftlichen studien am beispiel des projektes saba. “pflegebedürftig” in der gesundheitsgesellschaft, in *Hallesche Beiträge zu den Gesundheits- und Pflegewissenschaften* (2009)

# Chapter 4

## Information Aggregation in an Enterprise

Erwin Gunadi and Sahin Albayrak

**Abstract** In this chapter we discuss the application of a distributed information retrieval system in an enterprise environment. Focusing on the characteristics of information in enterprises such as the heterogeneity of available information, security policies, and the distributed nature of available data repositories we investigate how state-of-the-art distributed information retrieval approaches can be applied to build a distributed information retrieval system. Introducing a case study, we present an application of these techniques in an office environment that allows employees to find the most relevant documents across different data repositories without neglecting access rights. The application illustrates the advantages of using a multi-agent software infrastructure where individual components of such retrieval engine are realized by specific software agents.

### Berry Picking...

To whom it may concern.

I am the owner and inhabitant of the old windmill at Newgarden Avenue. As you are aware, the building has recently been placed under a preservation order and is now considered to be a historic monument. Due to the historic structural condition of the building, various home remodeling tasks are now inevitable. For example, the windows need to be replaced since they do not protect from the wind any longer. Also, thermal insulation needs to be applied to reduce heat loss. I understand that specific regulations must be considered when renovating historic landmarks. Could you please let me know which regulations are relevant for me? Also, please let me know which forms I need to fill in to receive a permit to start with the conversation of the building. Furthermore, please point me to the appropriate funds that I can apply for...

---

E. Gunadi (✉) · S. Albayrak  
Technische Universität Berlin, Berlin, Germany  
e-mail: erwin.gunadi@dai-labor.de

S. Albayrak  
e-mail: sahin.albayrak@dai-labor.de

Suzanne sighed and stopped reading. Her current job as a responsible official in her city's administration office was her dream job: It was challenging enough to compensate for the often quite tough times that she went through while studying part time for her Master's degree. At the same time, the standard 9-to-5 work hours allowed her to plan her day appropriately, leaving her sufficient time for her family. And most of all, having a tenure as civil servant came with its own undeniable advantages. Nevertheless, it were requests like this one that made her moan and wallow in self-pity.



Responding to such letters usually meant digging into the administration's intranet, browsing through all kinds of legal applications from citizens that were scanned and saved on different file servers, reading information that was outlined in the administration's badly maintained Wiki, finding old emails that referred to similar cases, and so on. Her colleague Barbara had a term for it: "Berry picking", as they had to go from source to source to pick up the information they required. Just like they used to do as kids when they picked berries from different bushes on their way to school. "Berry picking season once again!" she muttered and started to concentrate.

The citizen was right. If he wanted to renovate his property, he had to consider specific regulations to preserve the building for future generations. Suzanne remembered that she had read a memo a few months ago where they were informed about new regulations that would come into effect soon. She went back to her email client program and started to search in her inbox...There it was! An email from..., *oh wow!* ...from over a year ago where they mentioned these new regulations. Suzanne mugged. "Call me old, but at least my memory is still intact." She read through the email. They mentioned that they intended to build a task force with experts from the environmental, building, and monument protection authorities who should work out the details to implement the regulations. Unfortunately, they did not mention any contact persons, which made Suzanne frown. Over the years, she experienced that

finding out the corresponding contact persons on her own was a very challenging task. The faster method to find the information she was looking for was to search for the information herself in the intranet.

“Berry picking indeed,” she thought. Her city’s intranet was rather large and finding the right information was quite time-consuming. Being a rather conservative work environment, they did not possess a fancy content management system that allowed for a structured organization of divisions or projects. Instead, their IT infrastructure had grown over the years, resulting in multiple file servers where information was stored rather uncoordinated. Each division in the administration had implemented their own internal guidelines on how to store and share their data in the intranet. Suzanne knew her own department’s guidelines well enough to find the information she needed in a relatively short time. But searching on other departments’ network drives sometimes appeared to be “Mission Impossible.” To make things even worse, task forces consisting of employees from different authorities followed their own agenda when storing their files. If only there was a central system that would assist her in finding her way in this file server jungle...

## 4.1 Introduction

Enterprise environments such as the one described in the preceding scenario are very common. In almost every company, data are created, processed, and accessed on a daily basis. With the introduction of the computer in the workspace, the amount of digital documents increases even further. These digital documents come in many different formats. It can be textual documents (e.g., word or pdf documents), emails, images, graphics, or many other proprietary formats.

In most companies, the data can be stored in many different locations or repositories such as file servers, web pages, databases, email servers, and many others. Given this distributed file storage and given the heterogeneous nature of the information, finding the right information, also referred to as the information gathering task, can be challenging. In the preceding scenario, Suzanne has to use many different native search facilities such as the search function of her email client or the search mask of the department’s internal Wiki to find the information she is looking for. This approach of manually querying each repository is only possible though when the user knows about the actual existence of the resource. The more resources and repositories are available, the more likely it is that users might not be able to find the right documents.

According to Hawking et al. [15], different approaches can be used to realize an enterprise search system. The approaches differ in how data gathering for the indexing process is achieved. The choice of these approaches depends on different factors, such as network bandwidth, geographical locations, and/or repository sizes. These approaches considerably influence how search queries are processed in an enterprise search system. One approach is to introduce a search engine that includes all available repositories that can be found within an enterprise. Using this approach,



enterprise search engines can be designed to provide access to all resources via one shared interface. The main requirement for setting up such system is to crawl and index each of the repositories in separated indices.

Another important aspect is data protection. The presence of multiple document repositories, or collections, in an enterprise that are not necessarily regulated by a single point of administration pose a unique structure of the data collections. Companies are well advised to restrict access to certain information such as blueprints of their prototypes, customer data, email correspondence, or other sensitive data, hence protecting their assets from potential data theft [12, 14]. While some data are available for the whole enterprise (e.g., a company directory service or the company's web pages), various restrictions are applied. For example, data access could be restricted based on hierarchical boundaries, such as a department boundary. This means that data belonging to one department may not be shareable with employees of other departments. Hence, data protection is an important aspect in enterprise environments.

File distribution, heterogeneity, and access restriction play a key role in the application of enterprise search systems that aim to assist employees in their daily information gathering tasks. In this chapter, we introduce an enterprise search system with distributed indices that addresses the data accumulation task of enterprise search systems. The framework incorporates the idea of data mining agents, a technique, which has been successfully employed to create data warehouses [19]. We use autonomous agents for every task in the data accumulation and indexing activity, i.e., each agent provides core services that cover a specific part in the back-end. Complex tasks such as crawling and indexing a file server is achieved by combining the corresponding agents, i.e., the autonomous agents form a community to provide a joint service in creating search engine capabilities. When multiple data repositories (collections) need to be indexed we use these agent communities to build a distributed search engine. Search requests are handled by broker agents that verify users' identity and their access rights using the enterprise's directory access constraints that are defined using Lightweight Directory Access Protocol (LDAP).

The chapter is structured as follows. Section 4.2 introduces related work in the fields of desktop and enterprise search. Section 4.3 introduces technical challenges that need to be considered when building a distributed search engine in an enterprise environment. A comparison of different search result aggregation approaches is presented in Sect. 4.4. An exemplary implementation of such system is described in Sect. 4.5. Section 4.6 concludes this book chapter.

## 4.2 Related Work

This work builds upon prior work from different research domains, including enterprise search, distributed information retrieval, and multi-agent systems. In the remainder of this section, we present these domains and highlight state-of-the-art research approaches.

### 4.2.1 Enterprise Search

In recent years, various studies have been performed that focus on recognizing the characteristics and challenges of enterprise search (e.g., [12, 14, 25]). A key development in the study of enterprise search was the organization of the Enterprise Track as part of TREC 2005–2008 (see [7] for an overview of the first instance of this track). The provision of common data corpora within TREC enabled the study of important research challenges such as the development of better ranking methods, a better understanding of the users, and research on the creation of relevance assessments. A side effect of this focus on existing datasets was the limited attention to other important issues of desktop and enterprise systems, e.g., the crawling and indexing of data from distributed sources. Mukherjee and Mao [25] refer to this constant data accumulation process as a key task of an enterprise search system. They define an enterprise as an environment with the following characteristics:

- Heterogeneous document types: Data can be held in many types of documents, such as web pages, wiki, pdfs, emails, word documents, etc.
- Multiple document repositories: Documents are normally not held in a single file server or system. Depending on the importance and how critical these documents are, documents can be saved on dedicated servers, such as file servers or web servers. Users may have to mount multiple file servers in order to get various documents.
- Access restriction: Enterprise environments consist of hierarchy and roles. Therefore, each document has its own access list. An enterprise search system must be able to retain these rules and apply it into their search result.
- Data generation process: The pace and frequency of document creation and update also pose challenges on how to manage index updates since new documents should be searchable within a reasonable amount of time.

They argue that apart from the need to handle diverse data types (e.g., html pages, emails, database entries, and other documents), detailed information is required about the location of these datasets in the intranet of the enterprise and the access rights to these repositories.

According to Hawking [15] existing enterprise search systems can be classified into two categories: (1) systems that create one centralized index and (2) systems that depend on distributed independent indices. A centralized index can be used when it is possible to crawl all of the relevant data sources into a single index structure. However, since in most cases information is stored at different locations and due to physical constraints such as geographical location, low bandwidth connections and administration restrictions, gathering data in one search index is not always feasible [15]. Because of the advantages distributed indices can offer we decided to choose this option during the implementation of our enterprise search system.

Another important requirement of an enterprise search system is that it has to be able to handle security and rights management issues [14, 25]. Addressing this requirement Bailey et al. [4] introduce different architectures for the application

of document level security in enterprise search systems. In contrast to collection level security, document level security, or DLS, enforces document-level restriction according to the user's permission. In their paper [4], the authors outline two possible architectures on how user credentials can be used to filter the search results: (1) security is handled by the repositories itself or (2) the search engine takes care of which documents the user can find. In the first architecture the repositories act as interfaces that control which documents are readable for the respective user and thus filter the results from the search engine. As the second architecture type the search engine is responsible for checking user credential and which access rights the user has. The search engine also holds the access list for each document either on crawling/indexing time or during query preparation in order to filter the appropriate search results. With this information the search engine should then deliver the search results containing appropriate documents.

Currently there are some commercial enterprise search products available, such as Microsoft's SharePoint,<sup>1</sup> Oracle's SES<sup>2</sup> (sales of standalone version discontinued in 2014) and Google's GSA (Google Search Appliance).<sup>3</sup> All of the commercial solutions provide a flexible extension mechanism to add new file types for indexing. Regarding desktop search these products offer a various degree of integration. The Oracle's SES (Secure Enterprise Search), for example, has no native desktop search but can access GDFE (Google Desktop for Enterprise) for search results from local desktop.

#### ***4.2.2 Distributed Information Retrieval***

Distributed information retrieval has been researched and improved for more than two decades [5, 10, 30, 34, 37, 39]. Distributed information retrieval investigates search algorithms on distributed indices without the need to completely build a large centralized index of multiple collections. Shokouhi et al. [8, 30] describe which advancements have been achieved in this research area. Federated search is our primary base in implementing our system such that multiple distributed indices are built to cover information needs for different divisions in an enterprise environment. In a federated search system there are two main types of settings: (1) cooperative environments and (2) uncooperative environments [31]. A cooperative environment is a setting where the distributed collections provide the broker with extra information about themselves, which can be useful in collection selection and result merging. On the other hand, in uncooperative environments, a broker can only receive search results as a response to search queries. Regardless of the environment types there are always three steps that a federated search system should handle [6, 30]:

---

<sup>1</sup> <http://www.microsoft.com/enterprisearch/>.

<sup>2</sup> <http://www.oracle.com/technetwork/search/oses/overview/index.html>.

<sup>3</sup> <http://www.google.com/enterprise/search/>.

- **Collection Representation:** In this step, collections are evaluated. The evaluation results are used to determine the relevance of the collection with respect to the search query. In uncooperative environments, collection representation is built by sending random queries to get samples of each collection. This process is called Query by Sampling (QBS) [29, 30]. This sampled data is then used for estimating relevance of the represented collection for an incoming query.
- **Collection Selection:** In this step, relevant collections are selected and queried. Prior works for collection selection can be categorized into three categories [9, 30]: (1) large document-based collection selection [6, 36], (2) small document-based collection selection [28, 34] and (3) classification-based collection selection [2].
- **Result Merging:** In this step, documents retrieved from different collections are merged to a single result list. This includes score normalization, which makes these documents comparable and thus rankable as a single result list [6, 22, 32, 33].

In the context of the application of distributed information retrieval in an enterprise environments the various aspects, such as heterogeneous document types, access restriction, etc., characterized in the previous section should be considered. We describe in the following sections how the characteristics of the enterprise environment are handled in our distributed search system.

### 4.2.3 Multi-Agent System

In a multi-agent system, agent interact with each other to provide different functionalities to the users [35]. Concerning the concept of agent-oriented software development, Jennings et al. [18] provide an extensive description of how agent's paradigm can be compared with other software engineering paradigms. One of the advantages of using software agents is that we can model each agent to handle different unique tasks, such as crawling, searching, and management tasks. An example of agent concept usage in information retrieval field is shown in [1]. In this paper, agent technology is used to personalize search results based on the users' profile, i.e., contents is filtered based on the user's information need. In our prototype system we use JIAC Release V [21] as the framework for implementing our multi-agent-system backend. It provides robust and established functionalities for implementing distributed agent-system communication.

Multiple works are available on the application of multi-agent systems in the context of information retrieval [19, 27, 38, 40]. A similar approach of using a multi-agent-system for implementing enterprise search was introduced by Zhou et al. [40], who, however, relies on ontologies to model user access. Our system is more flexible since its user access is managed automatically by exploiting the existing access rights saved in LDAP.

### 4.3 Technical Challenges

As mentioned in Sect. 4.2.1, there are two approaches in realizing search systems for enterprise users. One possibility is to create a single index that contains documents from various repositories. However, restrictions such as physical locations, different administration policies, and bandwidth limitation make the data crawling process difficult to perform efficiently [15]. Therefore, the creation of various distributed indices is more feasible as it eliminates the need to transfer large amount of data for creating a centralized index. In this section, we outline various conditions and requirements for creation of a distributed search engine in an enterprise environment. The main focus of the section is on presenting technical issues that occur when such search system is set up. Section 4.3.1 first describes the types of data collections that often occur in enterprise environments. Section 4.3.2 then outlines the required steps for building multiple indices. The querying process of a distributed search engine is illustrated in Sect. 4.3.3.

#### 4.3.1 Typical Data Repositories

Enterprise is an organizational entity with a defined structure and boundaries and involving many parties with common interest. Through the defined structure and boundaries, information available within an enterprise environment can typically be categorized based on their content and their respective access rights.

The first type of information is publicly available and hence can be accessed by both employees as well as other parties who show interest in the company. A typical example is the company's webpage that can be accessed from anywhere in the world. These types of repositories can be freely searched regardless of user's permission.

The second type of repository contains information that can only be accessed internally within the company's physical network. We can further divide this type into two categories: (1) repositories that do not need authentication and (2) repositories that require authentication. Intranet webpages, wiki pages, and similar data repositories that can be found in the company's intranet fall under the first category. As long as users are using the company's ip-ranges they can freely open and access the information. The second category represents repositories which contain protected data, i.e., some sort of authentication is required before they can be accessed. This means accessing through company's physical address alone is not enough, users should validate their credential by logging in. Typical examples of such repositories are file servers. Each file in these servers inherits explicit read and write rights for individuals, as well as defined groups. By logging in, users will be authenticated and through this authentication users' rights including information about group memberships can be obtained. This credential information predefines and limits which data or files a user can access. Obviously, a search engine that accesses these repositories has to consider these permissions to avoid security leakage.

The last type of repositories are the employees' own workstations. Only the employee working on this workstation has direct access to these local files on their machines. These files can hold valuable and important information and are not replaceable with data from other repositories as they can contain work-in-progress files and work-related notes.

Without an enterprise search engine, users have to rely on system-native search interfaces to access the individual repositories. This means, however, that users have to repeat their search query multiple times, i.e., for each of the repositories, until they find the information they were looking for. This also means that without the availability of single sign-on, user needs to re-verify itself to each repositories. The more repositories are available within an enterprise, the longer this information gathering task can take. By applying distributed search techniques on these repositories, a single user interface in which all of these repositories will be queried, can be provided. This means that the user only needs to process a single result list that contains relevant entries that have been aggregated from relevant repositories.

### ***4.3.2 Crawling and Indexing***

In order to access documents using an information retrieval system, they need to be crawled and indexed first. As outlined above, the specific nature of the repositories in an enterprise calls for a distributed search infrastructure with multiple disjoint indices that need to be created separately. In this section, we discuss important aspects that need to be considered to prepare these indices.

It is important that the crawling task is properly adapted to the system's resources. Not all systems have the same amount of memory and processor power. The differences are particularly evident between dedicated file servers and desktop computers. Crawling processes on desktop computers must run unobtrusively in the background and only consume primary memory when no or little activities are currently active. On the other hand, file servers are capable of running many background tasks and have little constraints on the size of the memory.

According to [12, 15], one of the most important properties of data in an enterprise environment is the varying degree of the structure of the documents. Documents on file servers are mainly unstructured data with mostly no explicit references to other documents. This poses a challenge on how to create a good index structure during the crawling process. Creating a reliable structure from these unstructured data can be of benefit to users if they want to categorize their search results. Regarding the document level security we decided to use the second architecture type proposed by Bailey et al. [4]. In this type the search engine controls which documents for which users can be included in the search results. During indexing we also gathered all access list for the crawled files and include as part of the files' metadata. We need to emphasize that in order to keep the most actual access list for every indexed files a proper re-crawling interval should be configured for the crawling process.

As explained in Sects. 4.2.1 and 4.2.2, we create multiple indices for each of the repositories we want to search. An interesting challenge is how to access these indices. [5, 37] propose to use software brokers as a centralized component that can communicate with these indexes. In order to implement this communication model we realize our distributed search system as a multi-agent system. We model the multiple indices to be handled by search agents. These search agents are brokered by a broker agent as a centralized component. By using a broker agent we are able to process a search query without direct communication with every search agent. The broker agent is also responsible for verifying users' credentials. After successful user verification the broker agent collects user reading rights and group membership from the LDAP server through a LDAP agent and forwards this information to the search agents. The search agents then match the users' rights with the access list acquired from the crawling process. This means that users can only receive documents as search results for which they have access rights.

### 4.3.3 Retrieval

In contrast to an information retrieval system with one single index, distributed information retrieval systems rely on multiple indices that are usually created independently from each other [10, 30]. When users trigger a retrieval by formulating a search query, related documents are retrieved within each index and then returned as a result list. Although not every index will necessarily contain relevant documents, it is more than likely that documents will be found in more than one index, i.e., multiple ranked lists are created. An interesting research challenge is to merge these lists, hence presenting all search results in one larger result list. We compare in Sect. 4.4 the performance of different state-of-the-art unsupervised result merging algorithms using the FedWeb 2012 dataset [26].

When the broker agent receives answers from all search agents the broker agent normalizes these results and re-ranks them as a single search result list. As long as all of these repositories are reachable by one broker, only this broker is required to access all indices. However, there are cases when repositories cannot be reached by a broker, e.g., due to physical network boundaries. A typical example is the local desktop of a user. Local desktop computers can usually access file servers, but not vice versa. In this case, multiple brokers need to be considered. Each of these brokers is responsible for a specific group of indices in the network.

## 4.4 Evaluation of Result Merging Algorithms

One of the tasks in distributed information retrieval (see Sect. 4.2.2) is result merging. Its purpose is to merge multiple result lists to a single re-ranked list. For this task, the broker needs to normalize every document's score from each result list from

different collections and re-rank them as a new single list. In this section we compare different state-of-the-art unsupervised merging algorithms on experiments using the FedWeb 2012 dataset [26]. We first introduce in Sect. 4.4.1 the four different merging algorithms used in our experiments. Then, we present the results of these experiments by calculating information retrieval metrics (precision, recall, normalized discounted cumulative gain) resulted from these approaches with different retrieval setting in Sect. 4.4.2.

### 4.4.1 Algorithms

In previous years, various algorithms were introduced that merged result lists from different indices. In the remainder of this section, we introduce the most common techniques for unsupervised merging algorithms, namely CORI, Weighted MinMax, and round robin. We also introduce and compare our result merging algorithm, naive merger, with these algorithms.

#### 4.4.1.1 CORI

CORI was introduced by Callan et al. [6], who suggested to calculate the relevance of a collection as weight and to use this as a parameter value to recalculate each document score. It is important to note that CORI can also be used to rank collections that we do not do in our study. Let  $R$  be the notation for collection (from which a document is retrieved),  $d$  be a retrieved document, and  $q$  be defined as notation for an incoming query, then CORI re-rank calculation is defined as:

$$s_{\text{norm}}(d|q) = \frac{1 + 0.4 \cdot s_{\text{MinMax}}(R|q)}{1.4} \cdot s_{\text{MinMax}}(d|q) \quad (4.1)$$

The value 0.4 is proposed by the authors as the default value to define how important the collection weight should be. CORI is considered to be a state-of-the-art algorithm since experiments indicate that it is a robust unsupervised linear score normalization method.

#### 4.4.1.2 Weighted MinMax

Markov et al. [22] proposed a modification of the CORI algorithm, referred to as weighted MinMax. In this paper the authors replace the constant 0.4, which represents the importance of a collection, with variable Lambda. The authors investigated how result merging performance for CORI is influenced by varying the Lambda-parameter. The authors concluded that by setting the Lambda-parameter to infinity ( $\lambda \rightarrow \infty$ ) they can outperform other unsupervised linear score normalization



algorithms including the original CORI method itself. By eliminating this parameter the document score normalization is resulted from direct weight with the collection score:

$$s_{\text{norm}}(d|q) = s_{\text{MinMax}}(R|q) \cdot s_{\text{MinMax}}(d|q) \quad (4.2)$$

#### 4.4.1.3 Round Robin

The simplest approach to merge result lists that use different scored is the round-robin method. This algorithm does not have any score normalizing process. By applying round robin, the documents' positions of different collections are alternated. The algorithm takes from each incoming collection for each round every  $n$ th document and re-positions it in as a new list. Algorithm 1 lists the interleaving technique using pseudo code.

---

#### Algorithm 1 Round robin algorithm for result merging

---

INPUT Search results from n-collections

OUTPUT Merged search results

```

1: List<Document> mergedList
2: Set<Collection> processedCollection
3:  $n \leftarrow 0$ 
4: while  $processedCollection.length < collection.length$  do
5:   for each  $c$  in collection do
6:     if  $n < c.document.length$  then
7:       mergedList.add( $c.document[n]$ )
8:     else
9:       processedCollection.add( $c$ )
10:    end if
11:   end for
12:    $n \leftarrow n + 1$ 
13: end while

```

---

#### 4.4.1.4 Naive Merger

Naive merger is our novel result merging algorithm. This score normalization method consists of two steps. In the first step we calculate the weight of each collection using Formula 4.3. A collection's weight value is defined by the proportion of its document number with the total document number from all collections ( $\frac{|R_i|}{\sum_{i=1}^n |R_i|}$ ), multiplied by the maximum number of relevant documents that can be returned by the collections for the given search query ( $R_q$ ).

$$W(d|q) = \frac{|R_i|}{\sum_{i=1}^n |R_i|} \cdot R_q \quad (4.3)$$

The second processing step, shown in Formula 4.4 is then used to normalize the score for each of the document by multiplying the document’s original score with its collection weighted value. By using this strategy the more important a collection for the given search query the more boost the returned documents from this collection will get.

$$D' = W(d|q) \cdot D \tag{4.4}$$

### 4.4.2 Merging Effects

In order to illustrate the effect that different merging methods can have on the performance of the search results merging, we performed an experiment using the FedWeb 2012 dataset [26]. This dataset consists of documents from 108 different sources that are divided into 12 categories. Table 4.1 lists the complete subjects and examples of the used search engines. This dataset includes 50 TREC queries with human relevance judgments for documents retrieved from each search engine.

In order to evaluate the result merging performance, we executed experiments using 50 queries delivered with this dataset and measured these metrics: precision, recall, and Normalized Discounted Cumulative Gain [17]. We further differentiate these metrics in different length cut @5 and @10. To gain insight about the performance changes we executed an experiment round multiple times with different selected collection numbers as environment setting. In our result presentation we present the three metrics mentioned above using collection numbers from three, six, and twelve categories, respectively.

Figures 4.1 and 4.2 show the results for precision@5 and precision@10, respectively. Recall measurements, r@5 and r@10, are shown in Figs. 4.3 and 4.4.

**Table 4.1** Overview of the categorization in FedWeb 2012 dataset

Category	Count	Examples
General Web Search	10	Google, Yahoo, AOL, Bing, Baidu
Multimedia	21	Hulu, Youtube, Photobucket
Q & A	2	Yahoo Answers, Answers.com
Jobs	7	LinkedIn Jobs, Simply Hired
Academic	16	Nature, CiteSeerX, SpringerLink
News	8	Google News, ESPN
Shopping	6	Amazon, eBay, Discovery Channel Store
Encyclopedia/Dict	6	Wikipedia, Encyclopedia Britannica
Books and libraries	3	Google Books, Columbus Library
Social and social sharing	7	Facebook, MySpace, Tumblr, Twitter
Blogs	5	Google Blogs, WordPress
Other	17	OER Commons, MSDN, Starbucks

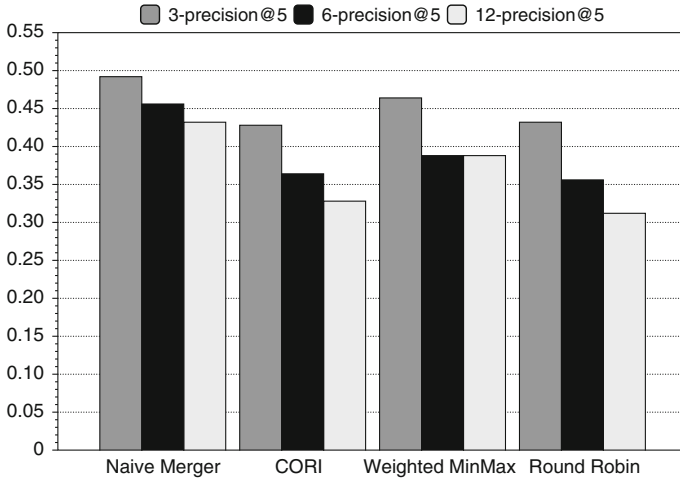


Fig. 4.1 Results precision@5

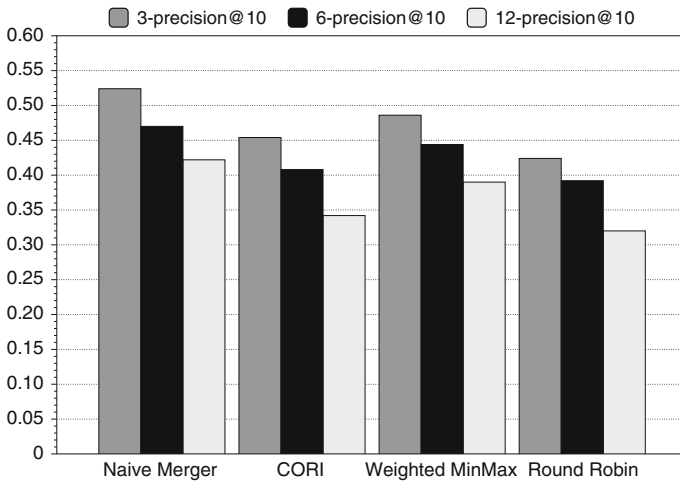


Fig. 4.2 Results precision@10

The measurement results for Normalized Discounted Cumulative Gain, or ndcg, are illustrated in Fig. 4.5 for ndcg@5 and Fig. 4.6 for ndcg@10.

The results suggest that the naive merger algorithm performs equally or better than the Weighted MinMax, the modified CORI algorithm proposed by Markov et al. [22]. In most cases however, naive merger maintains a higher score when more collections are selected for result merging. We also see that by selecting more collections in the result merging process the overall performance of all algorithms decreases. We conclude that this method can be used to merge search results from different sources.

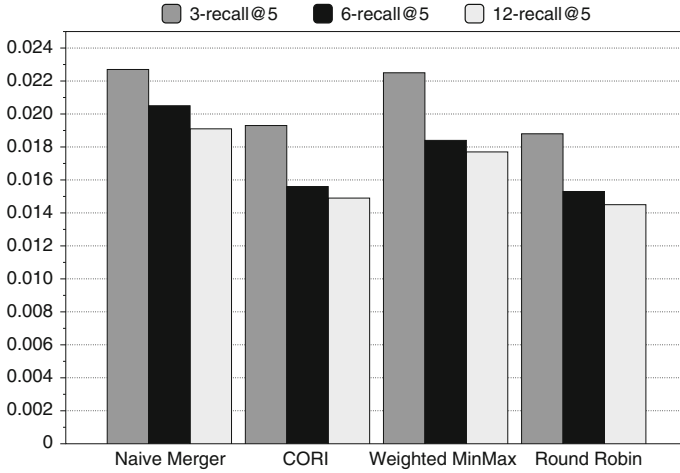


Fig. 4.3 Results recall@5

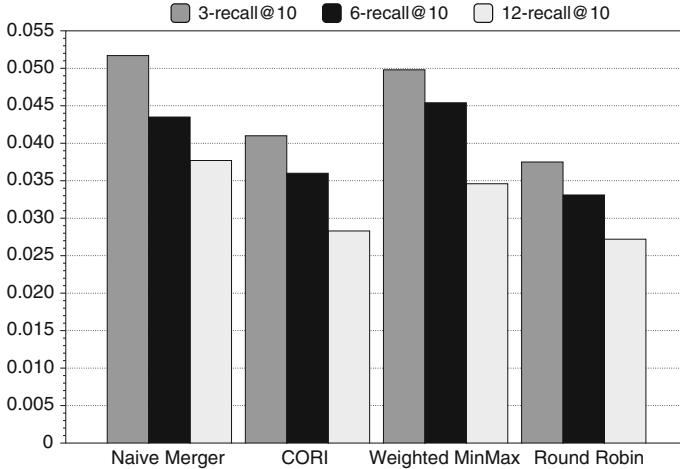


Fig. 4.4 Results recall@10

### 4.5 System Overview

In the previous sections we introduced the challenges of enterprise search and how distributed search techniques can be used to solve the enterprise search problem. In this section we discuss the structure of an exemplary distributed enterprise search system, PIA Enterprise [13], which is currently trialed at the administration offices of the city of Berlin, Germany. The enterprise search system is built as a multi-agent system (MAS). Our MAS-framework choice is JIAC release V [21], an open source MAS-framework. By implementing our system as a MAS-system

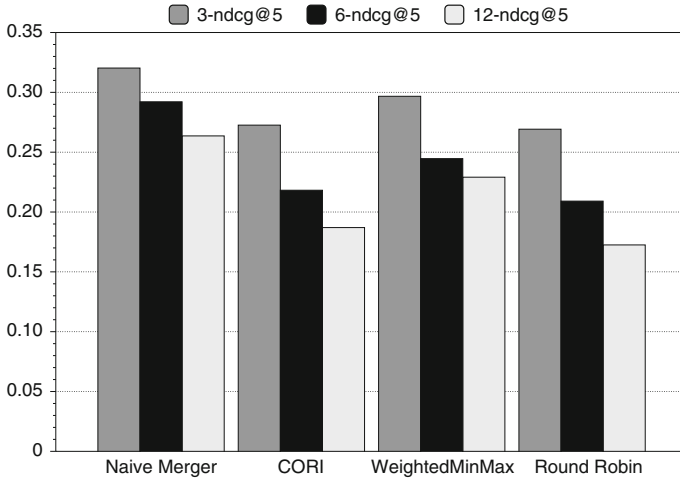


Fig. 4.5 Results ndcg@5

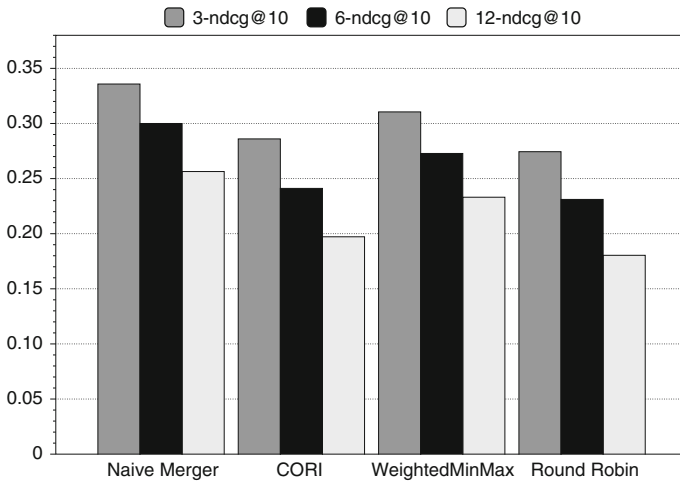
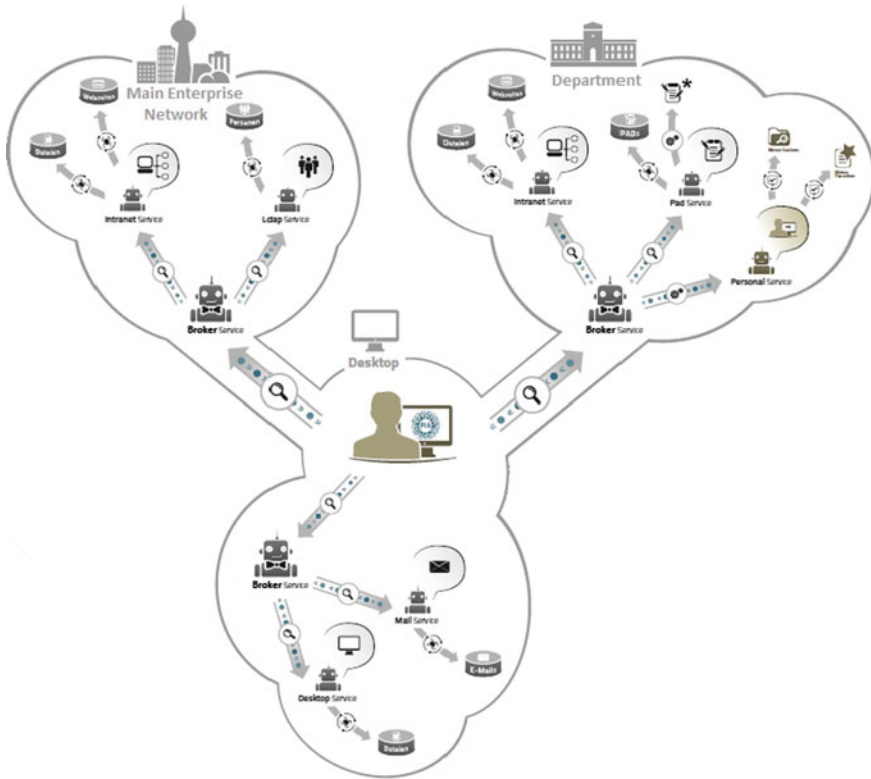


Fig. 4.6 Results ndcg@10

with JIAC V (see Sect.4.2.3) we can concentrate on the core functions without reimplementing the communication structures between distributed components. In this section, we present the multi-agent system implementation of the back-end system as well as the implementation of the web application on the clients. The communication between backend agents and client-interface is realized using Web-API and the HTTP protocol. This allows easy integration with other third-party applications.



**Fig. 4.7** The setting of the distributed enterprise search platform applied in the pilot project with three different broker agents contacted by the client

Figure 4.7 visualizes an employee in their working environment. When triggering a search, the search client on the employee’s desktop communicates with three different so-called broker agents. Each broker agent represents different repositories that the employee has access to. See Sect. 4.3.1 for details about these repositories. Each broker agent is responsible for a group of multiple repositories containing disjoint indices. Before accessing the indices of the individual repositories, the employee’s credentials are submitted to guarantee access in accordance to the company’s requirements. In the remainder of this section, we introduce the separated network areas in detail. Section 4.5.1 describes how the city administration’s meta directory and internal websites are indexed and made available for all employees of the city. Section 4.5.2 describes how the system is used to access files and information that are stored in the internal repositories of individual government branches. Indexing and retrieval of local files, i.e., files that are stored on the employee’s desktop computer is presented in Sect. 4.5.3. The graphical user interface of the system is described in Sect. 4.5.4.

### ***4.5.1 Main Enterprise Network***

The main enterprise network contains all information that is easily accessible. In our pilot project most of this includes the city's web pages where information about the city itself is presented to the general public. In addition, there are also intranet pages being maintained to help employees in their daily work. For example, a list of contacts and their work division is published on the website. These lists are very important for the employees, e.g., when they want to arrange a meeting with a colleague for a particular purpose. Summarizing, web pages both in internet and intranet represent a common platform to represent the city of Berlin either externally or internally. Consequently, these are valuable resources that should always be accessible for the employees. Accessing this information usually requires no additional authentication since both extranet as well as intranet are usually accessible for anyone whose computer is located in the main enterprise network.

In order to provide access to the web and intranet pages for anyone within the main enterprise network, we deploy multiple crawling ensembles, consisting of crawling and search agents. Therefore, we utilize open source tools Nutch<sup>4</sup> and Solr<sup>5</sup> and integrate them with our implemented software agents. This enables us to customize and automate the crawling and indexing process for vast amounts of data. In the Berlin administration office, employees' general information (i.e., their names, departments, email addresses, and telephone numbers), i.e., the administration's meta directory, is maintained using an LDAP server. For crawling such lightweight data we rely on Lucene.<sup>6</sup>

All of these crawling processes run independently, creating different disjoint indices in the process. A search agent is responsible for one of these indices. Search requests are forwarded from the broker agent to these search agents. The broker agent is also responsible to merge the search results into a single result list. This result list is then presented to the user using the graphical user interface.

### ***4.5.2 Department Network***

The bigger the size of an enterprise, the more complex its hierarchical structure will be. It is not unusual that companies are split into different divisions or branches. From an IT infrastructure point of view, these subdivisions often have their own subdomains within the main domain of the company or administrative division. Since these subdivisions may, to some extent, work independently of each other, it is not unusual that each has its own data administration and access policy. Besides, each subdivision usually maintains its own file servers that should exclusively be accessed by the employees of this particular division only.

---

<sup>4</sup> <http://nutch.apache.org/>.

<sup>5</sup> <http://lucene.apache.org/solr/>.

<sup>6</sup> <http://lucene.apache.org/>.

In our pilot project we need to include some city districts and departments in our enterprise search system. Each city district is treated as a single exclusive department network that is administered by their internal system administrators. The administrators are responsible for their own data management and manage their own subdomains within the main domain of the city of Berlin. They also maintain the employed users' credential and access rights. One of the unique aspects in our use case is that when an employee is in a department network, they can access the main enterprise network but not the opposite. Another additional requirement is that a user must first be logged in with a department account. After successful login a user is verified as an authorized user and is able to access files on the file servers. We deploy a broker agent for each district or department participating in the pilot project. These broker agents are required to verify users' credentials using the respective department's LDAP server information.

Similar to the crawling processes in the main enterprise network (Sect. 4.5.1), we deploy agent ensembles, consisting of crawler agents and search agents, to create multiple indices from files and data saved in the department network. In this case, the crawler agent reads and indexes these data including their access list information. This access list will be used during the processing of search queries. The broker agent first checks the user's reading rights and groups that they belong to and forwards this information to the search agents. The search agents use this authentication to filter out the relevant documents. Finally, the broker agent of the department network is responsible for merging the results coming from the individual search agents to a single search result list.

### ***4.5.3 Local Desktop Search***

The third important repository that employees access on a daily basis is their own personal desktop. The desktop computer is the main place where an employee creates their files, reads their emails, and saves exchanged data from other employees. During the completion of their daily tasks, an employee may need to retrieve these data.

In order to incorporate this repository, we implemented a specialized broker agent that runs as low-priority background process on the employee's desktop computer. To maintain data security, the local broker agent only processes search requests coming from the desktop computer itself. Moreover, we provide a desktop agent and mail agent that index local files and local mail archives. These three agents are packed as a single local service package that should be installed on a single user computer. Upon receiving search requests the local broker agent forwards the queries to the desktop and mail agents and the search results coming from these agents are then re-ranked and merged to a single result list by the local broker agent.



#### 4.5.4 *Web Client*

In order to give a complete user experience we have implemented a web client. The web client is a client-side javascript single-page application implemented using dojo toolkit.<sup>7</sup> The different result lists, coming from different broker agents, are aggregated in the client by sorting the documents according to the normalized score calculated by the respective broker-agent. Users can also explicitly filter the search results according to the origin of the network area (main enterprise network, department network, or local desktop).

Figure 4.8 depicts a screenshot of the graphical user interface of the demo system that has been deployed in the network of the Berlin administration offices. The screenshot shows the result list for the search query *pia technical documentation* that has been provided by the broker agents of the main enterprise network. On the left-hand side of the interface, the user can filter the results based on the different sources. Without logging in, the user will have no access to the department network and the filter icon for the department network search gives a hint that the user should first log in.

After successful login, the user is then able to search the department network. By logging in, the user credential is forwarded from the department network's broker agent to the search agents in the network area. This credential is used by the search agents to identify the search results that are relevant for the search query. Together with the search results from other broker agents the client will then sort these search results as a single result list. The sorting is based on the normalized score calculated by each broker agent based on the multiple repositories (Sect. 4.4).

#### 4.5.5 *Summary*

In this chapter we described how our secured distributed enterprise search system is built and deployed in our pilot project. The network environment in our pilot project consists not only a main network but also many different department networks. Overall we have to consider the following type of network areas with different security requirements: (1) The main enterprise network is the area where information is available to all employees as long as these employees are located in the physical network of the enterprise; (2) the department network is an area where employees must be authenticated in order to access documents. This authentication is also used to filter out the relevant documents so that employees only get search results in accordance to their own access rights; (3) the local desktop is a special area that is only accessible by the desktop owner. Each of these areas has their own dedicated broker agent. By using the web client the employees input their search queries. To receive search results from department network area the employees must be logged in. Upon receiving a search request, the contacted broker-agent verifies the user's credential (if required)

---

<sup>7</sup> <http://dojotoolkit.org/>.

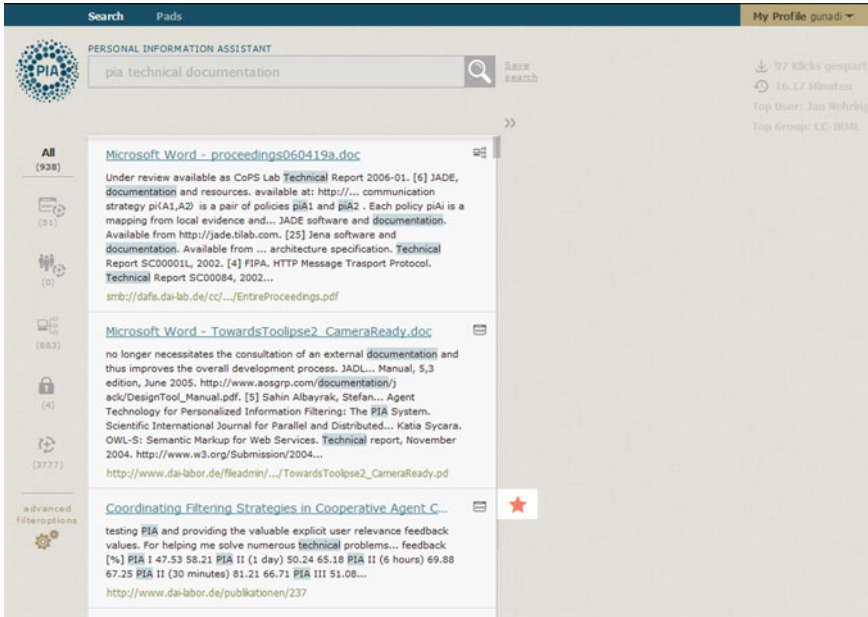


Fig. 4.8 Our web-client showing search results from the main department network, local desktop, and department network

and forwards this information to the search agents. The search agents process in turn the search query along with the user’s credential in order to filter out the relevant documents. This process allows the search system to return search results containing only documents accessible to the user.

## 4.6 Conclusion and Outlook

In this chapter we discussed the application of secure distributed information retrieval in enterprise environments. Focusing on the example of Suzanne and her workplace environment, we illustrated the challenges employees face in an enterprise environment. Data repositories like email accounts, file server, or web pages provide the employees with important information needed to complete their daily tasks. An employee should decide which repositories are important and must search these repositories to find the relevant files for information gathering task.

Based on this example we first described the distinctive characteristics of enterprise environments. These characteristics, such as the heterogeneity of the available data, security policy, and the distributed nature of the available data repositories, should be addressed to create an enterprise search system. Enterprise search system can be implemented in two forms: (1) by relying on a single centralized index

and (2) by using distributed information retrieval on multiple disjoint indices. The choice of which form is ideal depends on many factors such as geophysical limitation and/or network capacity. Distributed information retrieval is a research area with open research questions in well-defined processing steps. These steps are collection representation, collection selection, and result merging. Distributed information retrieval addresses the challenge of managing multiple indices by using a broker. Another research topic that we discussed is multi-agent systems. In a multi-agent system, multiple agents interact with each other in a distributed manner. This feature provides us an approach for the implementation of distributed information retrieval engine. In our case, we encapsulate functionalities such as crawling, retrieving, brokering, user profile management in specialized agents. Search agents are contacted by a broker agent, thus facilitating the result merging between different collections.

We introduce our distributed enterprise search system that is deployed in a pilot project with the administration offices of the city of Berlin. In this pilot project we emphasized how different network areas, each with multiple repositories, are handled with our implementation using JIAC V as multi-agent framework. The separation between each of the city districts as an independent department requires deployment of multiple broker agents. Each network area has its own characteristics regarding user authentication. In our implementation we decided that the enforcement of document level security should be managed by the search engine. This is solved by adding available access lists to every file we crawled and by saving this information during the indexing process. When processing a search request, every contacted broker agent verifies the user's credential and forwards this information to the search agents. The search agents process the search query along with the user's credential to filter the relevant documents. This process allows the search engine to return search results containing only documents accessible to the user.

For future works we want to improve the merging of search results. In order to have a better merged result we need to learn to prioritize relevant data collections. Collection selection is a step in distributed information retrieval that has not yet been properly explored in our enterprise setting. In normal environments, collection selection can rely only on relevance for the search queries. However, in enterprise environments, we also have to consider the security aspect before retrieving results. For example, even though a repository is relevant, it is possible that most of the documents in this repository are not accessible for the current user. This means that it could limit the results a user can retrieve and may reduce the recall value when another relevant repository is not selected. Addressing this issue, we currently investigate how to gather the right evidences [3, 11, 20] in selecting the right collections considering the security aspect of enterprise environment. In addition, we aim to incorporate multimedia content into our system, which requires further processing [16]. Finally, we intend to improve user interaction, e.g., by introducing gamification elements into the system that incentivize users to interact with the system. Preliminary studies [23, 24] in this direction are promising.

**Acknowledgments** We would like to thank ITDZ Berlin for their support and cooperation in realizing the pilot project.

## References

1. S. Albayrak, S. Wollny, N. Varone, A. Lommatzsch, D. Milosevic, Agent technology for personalized information filtering: the PLA-system, in *Proceedings of the 2005 ACM Symposium on Applied Computing, SAC'05* (ACM, New York, 2005), pp. 54–59
2. J. Arguello, J. Callan, F. Diaz, Classification-based resource selection, in *Proceeding of the 18th ACM Conference on Information and Knowledge Management—CIKM'09* (ACM Press, New York, 2009), p. 1277
3. J. Arguello, F. Diaz, J. Callan, J.F. Crespo, Sources of evidence for vertical selection, in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 315–322 (2009)
4. P. Bailey, D. Hawking, B. Matson, Secure search in enterprise webs: tradeoffs in efficient implementation for document level security, in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management CIKM '06* (2006)
5. J. Callan, Distributed information retrieval, in *Advances in Information Retrieval* (Kluwer Academic Publishers, 2000), pp. 127–150
6. J.P. Callan, Z. Lu, W.B. Croft, Searching distributed collections with inference networks, in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '95* (1995)
7. N. Craswell, A.P. de Vries, I. Soboroff, Overview of the TREC 2005 enterprise track, in *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland 15–18 November* (2005)
8. F. Crestani, I. Markov, Distributed information retrieval and applications, in *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 7814, ed. by P. Serdyukov, P. Braslavski, S.O. Kuznetsov, J. Kamps, S. Rger, E. Agichtein, I. Segalovich, E. Yilmaz (Springer, Berlin, 2013), pp. 865–868
9. F. Crestani, I. Markov, Distributed information retrieval and applications, in *Proceedings of ECIR*, pp. 865–868 (2013)
10. P.B. Danzig, J. Ahn, J. Noll, K. Obraczka, Distributed indexing: a scalable mechanism for distributed information retrieval, in *Proceedings of the 14th Annual SIGIR Conference* (ACM Press, 1991) pp. 220–229
11. F. Diaz, M. Lalmas, M. Shokouhi, From federated to aggregated search, in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR'10*, pp. 910–910 (2010)
12. R. Fagin, R. Kumar, K.S. McCurley, Searching the workplace web, in *WWW 2003 Proceedings of the 12th International Conference on World Wide Web* (2003)
13. E. Gunadi, M. Meder, T. Plumbaum, C. Scheel, F. Hopfgartner, S. Albayrak, Distributed enterprise search using software agents, in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems AAMAS '14*, pp. 1623–1624, Paris, France (2014)
14. D. Hawking, Challenges in enterprise search, in *Proceedings of the 15th Australasian Database Conference*, vol. 27 (2004)
15. D. Hawking, Enterprise search, in *Modern Information Retrieval*, ed. by R. Baeza-Yates, B. Ribeiro-Neto, 2nd edn. (Addison-Wesley, 2010), pp. 645–687
16. F. Hopfgartner, *Understanding Video Retrieval* (VDM, Saarbruecken, 2007)
17. K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. (TOIS)* **20**(4), 422–446 (2002)
18. N.R. Jennings, M. Wooldridge, Agent-oriented software engineering. *Artif. Intell.* **117**, 277–296 (2000)
19. M. Klusch, S. Lodi, G. Moro, Agent-based distributed data mining: the KDEC scheme, in *AgentLink*, pp. 104–122 (2003)
20. N. Limsopatham, C. Macdonald, I. Ounis, Aggregating evidence from hospital departments to improve medical records search, in *Proceedings of the 35th European Conference on Advances in Information Retrieval ECIR'13*, pp. 279–291 (2013)

21. M. Lützenberger, T. Küster, T. Konnerth, A. Thiele, N. Masuch, A. Heßler, M. Burkhardt, J. Tonn, S. Kaiser, J. Keiser, Engineering industrial multi-agent systems—the JIAC V approach, in *Proceedings of the 1st International Workshop on Engineering Multi-Agent Systems (EMAS 2013)*, ed. by M. Cossentino, A.E.F. Seghrouchni, M. Winikoff, pp. 160–175 (2013)
22. I. Markov, A. Arampatzis, F. Crestani, On CORI results merging, in *Proceedings of the 35th European Conference on Advances in Information Retrieval ECIR'13*, vol. 4, pp. 752–755 (2013)
23. M. Meder, T. Plumbaum, F. Hopfgartner, Perceived and actual role of Gamification principles, in *Proceedings of the IEEE/ACM 6th International Conference on Utility and Cloud Computing UCC'13*, (IEEE, 2013), pp. 488–493
24. M. Meder, T. Plumbaum, F. Hopfgartner, Daiknow: a Gamified enterprise bookmarking system, in *Proceedings of the 36th European Conference on Information Retrieval ECIR'14* (Springer, 2014) pp. 759–762
25. R. Mukherjee, J. Mao, Enterprise search: tough stuff. *Queue* **2**(2), 36–46 (2004)
26. D. Nguyen, T. Demeester, D. Trieschnigg, D. Hiemstra, Federated search in the wild: the combined power of over a hundred search engines, in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 1874–1878 (2012)
27. J.B. Odubiyi, D.J. Kocur, S.M. Weinstein, N. Wakim, S. Srivastava, C. Gokey, J. Graham, Saire—a scalable agent-based information retrieval engine, in *Proceedings of the First International Conference on Autonomous Agents, AGENTS'97* (ACM, New York, 1997) pp. 292–299
28. M. Shokouhi, Central-rank-based collection selection in uncooperative distributed information retrieval, in *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 4425, ed. by G. Amati, C. Carpineto, G. Romano (Springer, Berlin, 2007), pp. 160–172
29. M. Shokouhi, M. Baillie, L. Azzopardi, Updating collection representations for federated search, in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR'07* (ACM Press, New York, 2007) p. 511
30. M. Shokouhi, L. Si, Federated search. *Found. Trends® Inf. Retr.* **5**(1), 1–102 (2011)
31. M. Shokouhi, J. Zobel, Federated text retrieval from uncooperative overlapped collections, in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR'07*, pp. 495–502 (2007)
32. M. Shokouhi, J. Zobel, Robust result merging using sample-based score estimates. *ACM Trans. Inf. Syst.* **27**(3), 1–29 (2009)
33. L. Si, J. Callan, Using sampled data and regression to merge search engine results, in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'02* (ACM, New York, 2002) pp. 19–26
34. P. Thomas, M. Shokouhi, SUSHI: scoring scaled samples for server selection, in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston* (2009)
35. M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd edn. (Wiley, Chichester, 2009)
36. J. Xu, W.B. Croft, Cluster-based language models for distributed retrieval, in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval—SIGIR '99* (ACM Press, New York, 1999) pp. 254–261
37. B. Yuwono, D.L. Lee, Server ranking for distributed text retrieval systems on the internet. in *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications*, pp. 41–49 (1997)
38. H. Zhang, V. Lesser, Multi-agent based peer-to-peer information retrieval systems with concurrent search sessions, in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'06* (ACM, New York, 2006) pp. 305–312
39. K. Zhou, R. Cummins, M. Lalmas, J.M. Jose, Which vertical search engines are relevant? in *WWW 2013 22nd International World Wide Web Conference* (2013)
40. L. Zhou, Multi-agent based distributed secure information retrieval, in *CMC'10* vol. 1, pp. 76–79 (2010)

# Part II

## Personalization and Recommendation Services

### Overview

Differing from the use cases that are presented in the first part of the book, where textual documents are enriched and aggregated to ease users' access, another approach to address the information overload challenge is to rely on recommendation services. The main principle of recommender systems is to proactively provide information to users that they might be interested in. For instance, online retailers such as Amazon.com recommend other products that their customers might be interested in. Recommender systems hence can inform users about things they might not be aware of and have not been actively searching for. Two paradigms dominate the recommender systems' domain: content-based recommender systems and collaborative filtering systems. Content-based recommender systems assume that systems can successfully discover users' preferences from their liked items' contents. They provide suggestions by determining content similarities. Collaborative filtering systems aim to exploit the opinion of people with similar tastes. Thus, items are recommended when similar users of the recommender system showed interest in them. Thereby, collaborative filtering systems are able to recommend any kind of item disregarding their contents. In addition, recommender systems may combine both paradigms, obtaining hybrid approaches. In this part of the book, we present five use cases where different recommendation techniques are employed.

The first recommendation scenario focuses on video recommendation. The movie industry is a multi-billion dollar business with thousands of new movies released every year, e.g., by large Hollywood and Bollywood studios, but also by independent film makers. Given this large number of movies, finding new content that matches individual preferences is a challenging task. Lommatzsch presents in Chap. 5 a semantic movie recommender system which takes into account semantic similarity of movies. He argues that movies are semantically similar when they share specific aspects such as the same directors, actors, or belong to the same genre. He first discusses the challenges in creating such recommender system, then argues for the exploitation of a graph-based knowledge to provide recommendations and finally analyzes the advantages of semantic recommender systems.

Although a multitude of new movies are released every year, the frequency of these releases is far lower than it is in other domains. In Chap. 6, Kille et al. present the use case of online news recommendation that differs significantly from the movie scenario. In the news domain publishers constantly provide new news articles, resulting in vast amounts of items and a constantly changing dataset. Besides, freshness is an important aspect in news recommendations. While users may appreciate movie recommender systems suggesting movies from decades ago, they will most likely not be interested in receiving yesterday's news as a recommendation. Moreover, news publishers often have very limited knowledge about their readers, i.e., recommendation algorithms have to deal with incomprehensible as well as inconsistent user profiles.

Focusing on a hybrid recommendation technique, Plumbaum and Lommatzsch showcase in Chap. 7 how knowledge about individual users can be exploited to provide recommendations. Focusing on the news domain, they outline a system that provides news from the entertainment field that match users' preferences. In order to capture these preferences, they introduce an ontology-based user behavior model and present an evaluation that showcases the benefits of using such approach. Recommendation services have successfully entered online retailing businesses. Online retailers rely on their ability to direct users to products they will enjoy. In Chap. 8, Plumbaum and Kille investigate specificities of recommending fashion to men. Use cases mentioned before target consumable products. Users watch movies and read news articles. In contrast, systems recommending fashion articles are subject to further restrictions. Recommended products not only have to appeal to users but also match to pre-existing items in their wardrobe. In their chapter, the authors introduce strategies to deal with these additional requirements. In the last chapter of this part, Chap. 9, Meder et al. illustrate how gamification can be applied to motivate users to provide manual recommendations. Gamification refers to the use of principles borrowed from computer gaming to increase user engagement. They embed their scenario in an office workspace environment where various IT systems exist that are designed to share knowledge between employees. They argue that such systems are often not used by employees and suggest to gamify these systems, hence providing the means to increase users' activities. Furthermore, they propose a methodology to identify different types of employees, referred to as player types, which allows them to adapt the use of gamification elements based on the preferences of these player types.

# Chapter 5

## Semantic Movie Recommendations

Andreas Lommatzsch

**Abstract** The overwhelming amount of video and audio content makes it difficult for users to find new high-quality content matching the individual preferences. Recommender systems are built to suggest potentially interesting items by computing the similarity between users and items. The big challenges while creating recommender systems are the sparsity of data (the knowledge about users and items is often limited) and the popularity bias (most recommender algorithms tend to recommend popular items already known to the user). Semantic techniques supporting the graph-based representation of knowledge and the integration of heterogeneous datasets allow us to overcome these problems. The aggregation of knowledge from several different sources enables us to take into account many different aspects while computing recommendations. In addition, semantic recommender systems can provide explanations for suggested items helping the user to understand why an unknown item matches the individual user preferences. In this chapter we discuss the challenges in creating recommender systems and explain semantic approaches for the recommendation domain. We discuss the steps for building a semantic recommender system and present a semantic movie recommender system in detail. The advantages of semantic recommender systems compared to traditional recommender approaches are analyzed.

### Having a Wonderful Video Evening

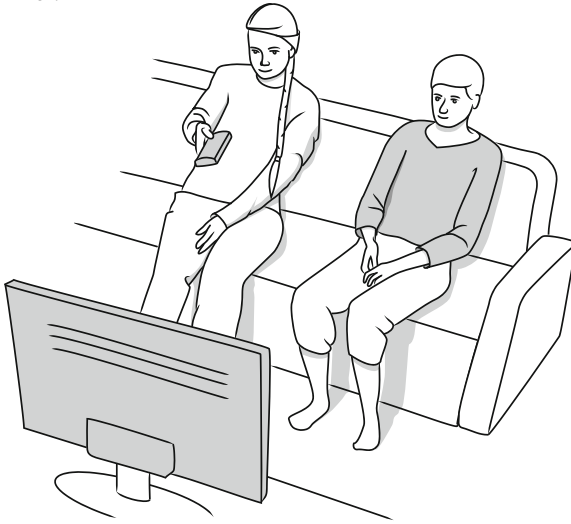
It is a cold and dark December late afternoon. Carl, Clara, and Steven have enjoyed the wonderful winter weather having several snowball fights with some school friends in the park. But now a strong wind comes up and Carl, Clara, and Steven return home hungry. In the comfortable well-heated living room they sit around the old round oak table. Suzanne serves hot chocolate for the kids and ginger tea with lemon and honey for Steven. On the table, there are fresh cookies made based on the secret recipe she learned from her great-grandmother. Since outside the weather become very windy,

---

A. Lommatzsch (✉)  
Technische Universität Berlin, Berlin, Germany  
e-mail: andreas.lommatzsch@dai-labor.de



they close the roller blinds and enjoy the cookies. The gramophone plays film music from Hans Zimmer.



Exhausted from the outdoor activities and thinking about the nice day the children make plans for spending the evening. Carl suggests playing bridge. Clara who never has luck in card games is not happy about this idea. She prefers watching TV. She takes the newspaper and reads the suggestions. None of the current programs seems interesting to her. Listening to the music, she gets an idea: They could test the new *Semantic-Movie* recommender and watch the suggested movie using a streaming service. So, she says happily: “Let’s test the semantic movie recommender and watch a movie with music composed by Hans Zimmer.” Clara likes the idea of watching a movie, but it must be an action movie. She says: “Yes. Let’s watch a funny action movie!” Suzanne is afraid that her children (Carl and Clara) will choose a movie not well-suited for kids. But she wants to give the new semantic movie recommender, because she has heard that the *Semantic-Movie* recommender application considers different types of preferences and has a special function for filtering out movies based on the movie’s age classification. The father loves fantasy movies. Since everyone has specific preferences but they all would like to spend the evening together as family they hope that there is a movie matching all the preferences. The mother has bad experiences with recommender system, providing in a suggestion based on mysterious algorithm nobody understands. She would like to understand why the suggested movies are really matching their aggregated preferences.

## 5.1 Introduction

The situation that the family is facing is commonly referred to as information overload. The overwhelming availability of products, movies, and books makes it difficult

to find items matching the individual preferences of users. Recommender systems address this problem by analyzing a huge amount of data taking into account several different criteria. Items (potentially unknown yet to the user) are ranked based on the user's (implicitly and explicitly) defined taste. In order to compute high quality recommendations comprehensive knowledge is required that covers all the aspects important for computing the relevance of items.

Recommender systems are used in a wide variety of domains, such as online shops (e.g., fashion), news articles, restaurants, travelling, and entertainment (music, movies, books). Based on limited (sparse) knowledge about the user recommender systems suggest items potentially unknown, but helpful to the user. Due to the complexity of the recommendation task, recommender systems apply sophisticated machine learning approaches enabling the systems aggregating different types of data and to extract knowledge useful for computing highly relevant suggestions. In this chapter we explain semantic recommendation algorithms and show at a concrete example how the approach can be used for building a personalized semantic movie recommender system.

The chapter is structured as follows. First, we discuss traditional recommender approaches and explain the challenges (Sect. 5.2). Subsequently, we analyze semantic approaches for managing and processing knowledge. Semantic techniques help us to overcome the problem of sparse data and allow us the aggregation of comprehensive knowledge collections while computing recommendations. Semantic resources, datasets as well as mapping and scaling models are needed for representing knowledge in an efficient way. These strategies are discussed in Sect. 5.3. Then, we explain how semantic approaches can be applied for building a semantic movie recommender system (Sect. 5.4). In Sect. 5.5, we present our implemented Semantic Movie Recommender system and discuss the evaluation results. Finally a conclusion and an outlook are given.

## 5.2 Challenges in Recommender Approaches

Traditional recommender approaches are usually classified as collaborative or content-based [1]. Collaborative recommenders analyze the user's rating behavior [15] whereas content-based recommender approaches focus on analyzing the content-based features of items [22]. Although these recommender algorithms are widely used, traditional recommenders show several weaknesses and shortcomings.

**The new user problem:** In order to provide recommendations meeting the individual user preferences the recommender system needs detailed information about the user. When a new user registers at a recommender system the user must create a profile describing liked and disliked items. Since most users start with a small initial profile that does not give complete information about the user's preferences, the recommendation quality for new users is limited.

**The new item problem:** Recommender system using Collaborative filtering algorithm compute the relevance of items based on the ratings of users. Items not

rated yet by an adequate number of users cannot be recommended by Collaborative Filtering algorithms. That is the reason why new items (might highly relevant to the user) are not recommended. To tackle this problem, the recommender system should apply content-based recommender algorithms, since content-based knowledge does not depend on user ratings.

**The system's cold-start problem:** When setting up a new recommender system, the number of users and items is limited. Since Collaborative Filtering is based on the idea of computing the similarity between users and items, the quality of the recommendations depends on the number of similar-minded users to the current user. If the number of users in a system is low, the probability is high that no similar users can be found. This results in reduced recommendation accuracy.

**The popularity bias:** Recommender algorithms should assist users in finding potentially relevant items. Items already known to the user are not good recommendations because they do not mean useful information. Since Collaborative Filtering algorithm tend to recommend items positively rated by many users, collaborative filtering algorithms have a strong bias toward popular items. Recommender systems must be aware of this fact and ensure that at most of the recommendations are new and useful to the user.

**Missing support for multi-lingual data:** Natural language descriptions of content are a big challenge in the recommendation domain. On the one hand, content-based description (e.g., movie reviews) are typically available in several different languages. If a recommender system supports only content in one language, a big amount of relevant data cannot be processed resulting in sparse data and a low recommendation quality. On the other hand, natural language texts are often ambiguous requiring detailed linguistic knowledge for resolving ambiguous terms. In order to overcome the problem of multi-lingual natural language texts, a recommender system should be able to represent knowledge in a language independent way. In addition, the system should support the aggregation of knowledge from different languages ensuring a rich, dense knowledge base. Knowledge extracted from texts in different languages as well as from content-based and collaborative knowledge sources should be represented in a unified data format allowing the efficient management of heterogeneous knowledge.

One approach for representing knowledge in a universal, natural-language independent way is the use of semantic techniques and graph-based approaches. That is the motivation for us to discuss these approaches in the next section in detail.

### 5.3 Semantic Approaches and Knowledge Resources

The challenges of missing data, the use of different languages, the need of integrating different types of knowledge, and the lack of explanations can be solved using semantic techniques for managing knowledge and for computing recommendations. The semantic representation of knowledge aims to overcome the problem of traditionally proprietary data formats tailored to one specific scenario. Semantic data

formats abstract from concrete domains and represent different types of knowledge in a unified graph aggregating heterogeneous data sources and knowledge types.

### 5.3.1 *Semantic Data Formats*

Semantic techniques for representing knowledge are based on ontologies and graphs. Ontologies are designed to store various types of knowledge in a unified machine readable way. This enables machines to understand the meaning of data and simplifies the sharing and reuse of data in different scenarios [14].

The semantic knowledge representation is based on ontologies defining the relevant aspects of the modeled domain. Ontologies define the concepts and the relevant relationships between the concepts. In addition, ontologies may contain rules enabling deriving implicit knowledge as well as checking the consistency of knowledge. Ontologies describe the structure of the domain and define the basic concepts.

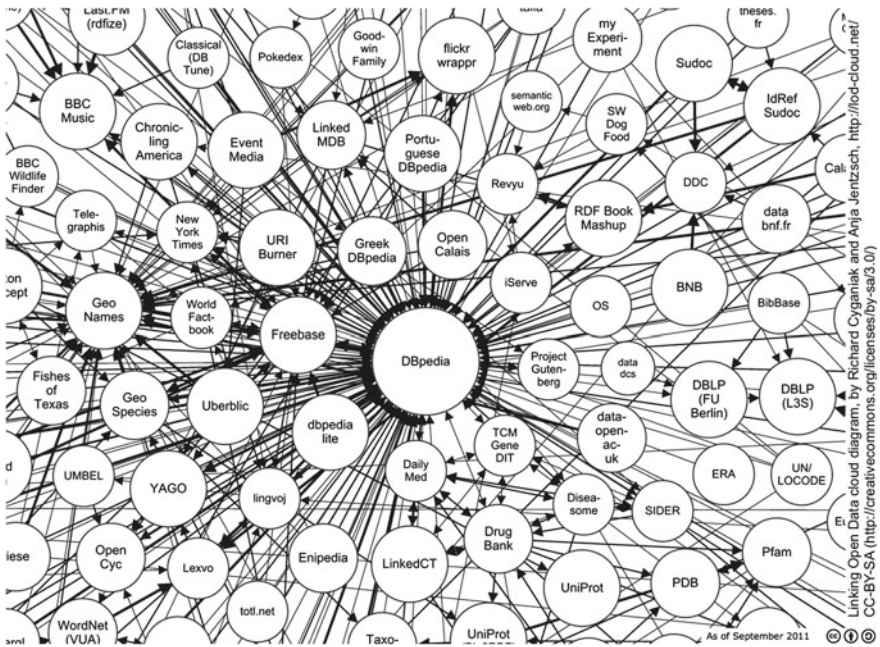
Knowledge about the entities and the relationship between entities is represented based on graphs consisting of nodes and edges. The nodes describe concepts and instances, the edges the relations between the nodes. In general, the edges are labeled allowing a fine-grained description of the relations. In order to store graphs in a flat way, graphs can be decomposed into a set of triplets consisting of subject, predicate, and object.

The most popular data formats used for storing semantic knowledge are the Web Ontology Language (OWL) [11] and the Resource Description Framework (RDF) [3, 4]. Both data formats are endorsed by the W3C and enable the efficient representation of semantic knowledge. RDF focusses on the representation of knowledge as triples, OWL supports additionally semantic relations (e.g., `sameAs`) and logical predicate logic (reasoning). Using RDF and OWL, comprehensive knowledge resources can be built providing a valuable knowledge base for a wide variety of scenarios.

### 5.3.2 *Semantic Resources*

Knowledge resources providing semantically represented knowledge exists for many domains [6]. The most popular knowledge resources are visualized in the Linked-Open-Data Cloud (see Fig. 5.1).

The central node of the linked open data cloud is the DBPEDIA [2]. DBPEDIA contains data extracted from Wikipedia in a semantic data format (RDF). DBPEDIA provides knowledge for a wide variety of domains and acts as the most important hub for connecting the different sources in the Linked Open Data cloud.



**Fig. 5.1** The figure visualizes the Linked Open Data cloud. The semantic sources are grouped by domain. The central node of the LOD cloud is DBPEDIA being best connected with the other data sources

The sources in the Linked Open Data cloud are grouped by the application domains. Popular sources providing data for the entertainment domain are IMDB,<sup>1</sup> FREEBASE<sup>2</sup> and DBTROPES.<sup>3</sup>

**FREEBASE:** FREEBASE is a large semantic, collaboratively created knowledge source [8]. FREEBASE aggregates knowledge harvested from many sources, such as various wikis and portals and databases. FREEBASE's mission is to create a global knowledge base allowing people and machines to access common information effectively. FREEBASE data are freely available for commercial and non-commercial use under a Creative Commons Attribution License [10], and an open API allowing the efficient and easy access to the knowledge base. In contrast to other knowledge sources FREEBASE provides images and links to external resources in addition to the semantic facts.

**IMDB:** The Internet Movie Database is a popular online database of information related to films and TV programs. It provides detailed information about actors, production crews, fictional characters, biographies, plot summaries, trivia, and many

<sup>1</sup> <http://www.imdb.com/>.

<sup>2</sup> <http://www.freebase.com/>.

<sup>3</sup> <http://dbtropes.org/>.

more aspects. In addition to content-based information, IMDB also contains ratings and reviews. Most of the IMDB's data are available as RDF triples ensuring the efficient automatic processing of the information.

**Discussion:** Semantic knowledge resources are valuable sources for recommender systems because these resources provide different types of knowledge ranging from fine-grained descriptions of entities (e.g., movies and actors) to ratings and to user created reviews. Although most resources are not perfectly tailored to a recommender scenario, the machine readable representation of knowledge makes it easy to use these data in a recommender system. The comprehensive collection of data allows semantic recommender systems to overcome the cold-start problem and to consider a greater number of aspects than traditional recommender systems. In addition, the graph-based representation of knowledge and the separation of structure (nodes and edges) from natural language labels ("node names" might available in different languages) simplifies the creation of explanations for computed recommendations.

The use of semantic approaches has many advantages; but there are also several challenges arising from the complexity and the heterogeneity of semantic datasets. The use of semantically represented data in recommender systems leads to several new research questions. Most traditional recommender systems using semantic datasets focus on datasets having only two node types avoiding problems with the heterogeneity of edge semantics.

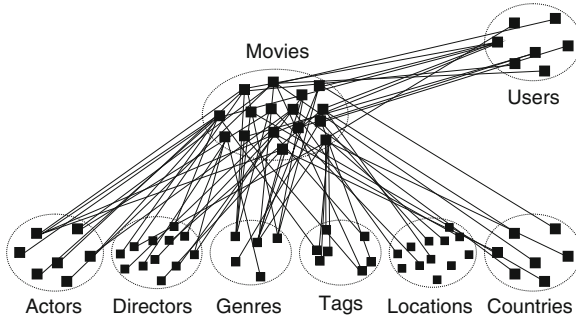
The aggregation of heterogeneous data into one big dataset requires a deep analysis of the different sub-datasets focusing on the meaning of the respective entities and relationships. In addition, optimized scaling as well as weighting models and domain-specific edge algebras should be defined reflecting the semantics of the specific datasets [19]. Last but not least, the computational complexity of processing huge graphs must be taken into account when defining a recommender approach for semantic datasets.

In the next sections we discuss the challenges in detail and present an approach for learning a universal semantic recommender.

### ***5.3.3 A Dataset for a Semantic Movie Recommender***

In order to implement a semantic movie recommender system, we have to find semantic data sources providing detailed knowledge about movies and all aspects potentially relevant for computing recommendations. In addition, training data for optimizing the recommender models are needed. In this section we describe our semantic movie dataset and discuss the characteristics of the dataset. Subsequently, we explain step-by-step how to build a powerful semantic recommender based on the dataset.

The *Internet Movie Database* (IMDB) is an online database containing information related to movies, actors, directors, and production data. IMDB is one of the most popular online entertainment websites with over 160 million visits each month [12, 28]. Most of the movie data are freely available as Linked Open Data simplifying



**Fig. 5.2** Our semantic movie dataset consists of six bipartite relationship set providing knowledge about movies. The *Movie–User* relationship set describes the user preferences

the integration and the processing of these data. Unfortunately, the freely available IMDB data lack personalized rating and usage information.

We obtain personalized movie preferences from the *MovieLens* dataset [13]. MOVIELENS is a recommender system and virtual community website that allows users to create profiles and subsequently obtain movie recommendations. The MOVIELENS dataset provides rating data including timestamps. Since the MOVIELENS and the IMDB dataset have a large overlap in the set of considered movies, the two datasets can be combined aggregating encyclopedic and rating-based knowledge. The mapping is performed by computing concordant properties (e.g., title, elapsed time, genres). A frequently used dataset combining data from MOVIELENS and IMDB is the HETREC dataset. The dataset has been created for the International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HETREC 2011)<sup>4</sup> and can be retrieved from GROUPLENS.<sup>5</sup>

We use the aggregation of the IMDB and the MOVIELENS dataset for creating a semantic movie recommender system. The structure of the dataset is shown in Fig. 5.2. The central entity type is *Movie*. The entity *movie* is directly connected with the entity types *Actors*, *Directors*, *Genres*, *Tags*, *Locations*, and *Countries*. In general, the dataset can be seen as a multi-graph, supporting several different edges between two nodes of the graph.

In addition to the content-based movie descriptions, the relationship *Movies–Users* provides user ratings for movies. The user ratings are used for optimizing and benchmarking the learned recommender strategies. For the evaluation of our approach, we split the user profiles (obtained from MOVIELENS) based on a global timestamp into a training set and a test set. We filter out user profiles having less than ten entries in the training set or the test set. We handle the dataset as a collection of bipartite relationship sets each consisting of undirected, equally weighted edges. The size of the entity sets and edge sets used in the evaluation is shown in Table 5.1.

<sup>4</sup> <http://ir.ii.uam.es/hetrec2011/>.

<sup>5</sup> <http://www.grouplens.org/node/462/>.

**Table 5.1** The table shows the size of the entity and relationship sets of our movie dataset

Entity type	Number of entities	Relationship movies ↔	Number of edges	Graph density
Actors	95,321	Actors	231,742	1.80E-05
Directors	4,060	Directors	10,155	4.24E-06
Genres	20	Genres	20,809	9.80E-06
Tags	5,297	Tags	51,795	2.09E-05
Locations	187	Locations	49,167	2.30E-05
Countries	72	Countries	10,197	4.80E-06
Users	760	Users (train)	525,318	2.42E-04
Movies	65,133	Users (test)	330,280	1.52E-04

**Discussion:** In this section we explained our semantic datasets for the movie recommendation domain. We presented the dataset we use for learning a semantic movie recommender and discussed the properties of the dataset. The dataset comprises encyclopedic knowledge as well as rating knowledge describing the user’s preferences. Data collected from different sources are combined in one large graph and stored in a uniform semantic data format. All nodes (*entities*) in the created graph are identified by unique uniform resource identifiers (URIs). The dataset statistic (Table 5.1) shows, that the properties of the entity sets and the relationships sets highly differ (according to number of elements and according to the density of the relationships sets). Thus, the dataset allows us to analyze how our approach can handle the heterogeneity in large semantic datasets.

### 5.3.4 Challenges and Requirements for Learning Recommenders

Having created one large semantic knowledge graph that aggregates data from heterogeneous sources, we define an approach for learning a recommendation strategy. The challenges of creating a powerful semantic recommender are: (1) The heterogeneity of the aggregated sub-graphs according to the number of nodes and edges, (2) the sparsity of sub-graphs, and (3) the diversity of noise in the aggregated sub-datasets. In addition, the sub-graphs may use different edge types and schemes for assigning edge labels requiring a domain-specific model reflecting the heterogeneity of the aggregated graphs. In the following paragraphs we discuss the challenges in detail and explain approaches for the processing of heterogeneous semantic data.



### 5.3.4.1 The Graph-Based Knowledge Representation

We represent knowledge in a large graph consisting of nodes and edges. For ensuring that all knowledge of the graph can be taken into account and for simplifying the processing, we make sure that the graph is connected. Disconnected components might be connected with the main component by adding edges with a very low weight (e.g., in a music genre graph each node may be connected with a general genre node). Entities with the same semantic meaning retrieved from different sources should have the same URI. If the considered sources use different URIs for semantically identical entities, we unify the URIs using “mapping” edges, such as `owl:sameAs`.

### 5.3.4.2 Mapping Edge Labels to Similarity Scores

Recommender systems usually compute recommendations based on the estimated relatedness between users and items [9, 16]. For ensuring that knowledge from different sources can be combined, we model the relatedness between entities with numerical similarity values. Thus, we map the labels of edges that indicate a similarity (such as *liked* or *user has bought an item*) to numerical values (e.g., on a scale [0, 1]). The resulting distributions of similarity scores must be analyzed when computing the recommendations. That is why we discuss scaling and weighting models adapting the similarity scores for the needs of the applied recommender algorithms in the next section.

### 5.3.4.3 Scaling Models

We define the relatedness of two entities in a graph based on the edges connecting these entities. Initially, we assign for each semantic edge connecting the nodes  $n_i$  and  $n_j$  (having an influence on the relatedness of two nodes) a similarity score  $w_{ij} = 1$ ; if two nodes  $n_i$  and  $n_j$  are not connected by a semantic edge, we assign the weight  $w_{ij} = 0$ . Thus, we get an adjacency matrix containing only the values 0 and 1. Due to the fact that most graphs are sparse, we suggest using sparse matrixes for storing the graphs, keeping only the nonzero weighted edges.

Since different semantic edge types may have a different impact on the relatedness of two nodes, we define the scaling factor for each semantic edge type. We compute the adapted similarity score by multiplying the initial score with a scaling factor. The scaling factor is defined based on expert knowledge. For example, the semantic edge “user  $u$  has bought item  $i$ ” usually implies a higher relatedness than “user  $u$  has read the description of item  $i$ ”.

The node degree (the number of nodes directly connected with the respective node) is another import aspect that should be considered when computing the relatedness score of two nodes. Entities, highly connected, often represent popular entities (*liked* by almost everyone). Edges connecting *popular* item nodes with user nodes often do not contain much information about individual user preferences. Thus, dependent

from the respective scenario, the node degree should be taken into account while assigning the relatedness scores to semantic edges.

For figuring out adequate scaling models, the analysis of the node degree distribution usually is a good starting point for defining a dataset specific scaling strategy. Typical scaling models for social network data are based on power law probability distributions. The scaling is done using logarithmic or polynomial scaling functions. Adequate parameters should be learned based on training data. In the experiments conducted in our evaluations we found that the applied scaling models have a strong impact on the recommendation quality [21]. Inadequate scaling models or *missing* scaling models cannot be repaired in the later steps.

### 5.3.4.4 Path-Based Relatedness Models

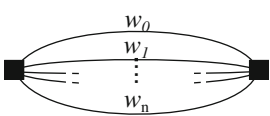
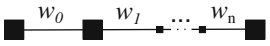
Semantic datasets represent knowledge as a large graph consisting of nodes and edges. In order to compute recommendations, a measure is needed for calculating the relatedness between all node pairs taking into account the edges between the nodes. In the previous sections, we explained how to assign relatedness scores to directly connected nodes. In this section we focus on *edge algebras* allowing us to assign relatedness scores also for node pairs connected by complex paths (characterized by parallel edges and long edge sequences).

First, we define criteria according to which the score for complex path should be computed [21]:

- If two nodes are directly connected by exactly one edge, the relatedness of the nodes is defined based on the edge weight.
- Two nodes are the more semantically related the more parallel paths between the nodes exist.
- Two nodes are the less semantically related the longer the path between the nodes.

We analyze three different approaches for combining the edge weights. Our approaches are based on the *distance* between nodes. We define the distance between two nodes as the reciprocal of the relatedness score. Thus, great distance value result

**Table 5.2** The table shows the formulas for calculating the path weights for (a) parallel edges and (b) for a sequence of edges

		Weighted path	Resistance distance	Shortest path
(a)		$w = \sum_{i=0}^n w_i$	$w = \frac{1}{\sum_{i=0}^n w_i}$	$w = \min_{i=0}^n w_i$
(b)		$w = \gamma^n \prod_{i=0}^n w_i$	$w = \sum_{i=0}^n w_i$	$w = \sum_{i=0}^n w_i$

The discount factor  $\gamma$  ensures that short paths get a higher weighting than long paths

in small relatedness scores. If two nodes are not directly connected by an edge, we assign a relatedness score of 0, resulting in a distance of infinity. The basic rules for calculating the distance between two nodes in a semantic graph are visualized in Table 5.2. The characteristics of the different edge algebras are discussed in the next paragraphs.

**Shortest Path:** We define the *distance* of two nodes in the graph based on the shortest path between two nodes. The shortest path edge algebra assigns the minimal distance of all paths between two nodes if several parallel paths exist. The distance values are summed up for a sequence of edges.

The shortest path distance has the advantage that it can be efficiently implemented (e.g., based on depth-bounded search). Additionally, the shortest path principle is well-understood by many users and widely accepted. Unfortunately, the shortest path approach does not take into account the number of parallel paths between two nodes. Thus, this algebra should not be used if the number of parallel paths is the dominant criteria for computing the relatedness of two nodes.

**The Resistance Distance:** In contrast to the shortest path algebra, the resistance distance [29] considers all parallel paths between two nodes. The resistance distance can be computed based on the Moore-Penrose pseudo-inverse of the LAPLACIAN matrix of the graph. The Resistance Distance is more complex than the shortest path algebra, but it fits well with the proposed criteria. In many application scenarios the resistance distance is computable with an acceptable effort. Unfortunately, the resistance distance is difficult to understand for most users, making it hard to generate good explanations.

**The Weighted Path Algebra:** The weighted path algebra defines an very efficient approach for computing the similarity of two nodes. The algebra is induced by a standard dot product of the adjacency matrix of a graph. It assumes that the edge scores are between 0 and 1 ensuring that the score for a long path is lower than the score of each edge in the path. The advantage of this algebra is, that the underlying assumptions are well-understood by most users. Additionally, it can be efficiently computed based on matrixes. A disadvantage is that the weight of a complex path can be above the upper bound of 1.

### 5.3.4.5 Complexity and Noise

Real-world datasets are often huge, sparse and noisy. Since Linked-Open-Data is often generated by volunteers in their spare time and not by professional experts, Linked-Open datasets might contain inconsistencies and errors [7]. In user-generated entertainment datasets, the amount of information might differ from domain to domain. User-generated descriptions might contain spelling mistakes or invalid characters. Thus a recommender component should be aware of these challenges and provide robust algorithms able to cope with noisy, heterogeneous data.

The dataset complexity as well as the differences in sparsity and in the noise between the aggregated datasets must be taken into account by the recommender framework. The recommender system should provide strategies for reducing the

dataset complexity (resulting in a reduced sparsity) and for extracting the most relevant information (resulting in a reduced level of noise). Clustering approaches as well as dimensionality reduction methods can be applied to detect irrelevant data allowing us to reduce the complexity and the sparsity of the dataset [5].

**Discussion:** In this section we discussed how to create and optimize a unified graph based on data retrieved from several different sources. We presented approaches for the domain-specific scaling of edge weights and models for aggregating the edge weights of a complex path. Based on the optimized semantic graphs, recommender models can be computed. In the next section we analyze algorithms for computing recommendations considering the computational complexity, the recommendation accuracy, and the ability for providing explanations.

## 5.4 Semantic Recommender Approaches

Having defined a unified graph with numerical edge weights, we analyze methods for computing recommendations. In the following sections we discuss the different recommender approaches and analyze the respective strengths and weaknesses. In our analysis we focus on (1) Memory-based recommenders, (2) Model-based recommenders, and (3) ensemble-based approaches.

### 5.4.1 *Memory-Based Recommender*

Approaches for graph-based recommenders can be classified according to the data structures internally used. *Memory-based* recommenders compute suggestions directly on the graph. This simplifies the adding and removing of data due to the fact that there is no internal model that must be adapted to new data. Memory-based recommenders for semantic graphs compute suggestions directly on the graph using graph-search algorithms, such as `BRANCH AND BOUND` [24]. Since the run-time complexity of these algorithms grows exponentially with the considered path length, memory-based approaches usually consider only entities reachable by relatively short paths. It is assumed that the relevant entities can be found in the near environment around the input entities.

Memory-based approaches have the advantage that updates in the semantic graph immediately affect the computed recommendations. Consequently, no additional resources for model updates are needed. Moreover, memory-based recommenders can provide human readable explanations, visualizing the nodes and edges considered while computing the recommendations. In most scenarios the path length is limited so that the explanations are not too complex ensuring that the explanations are understandable for the users. A visualization of an explanation generated by a memory-based music recommender is shown in Fig. 5.3. The example explanation visualizes how starting from an input node (e.g., from the user profile), a recommendation is computed. Starting from the movie node `King Kong (2005)`, the recommender

considered eight nodes having the type Actor. All these nodes have a direct edge to the Actor node *The Hobbit (2012)*. In addition, paths via the relationship sets *Misc*, *Producers*, *Writers*, and *Directors* are taken into account. Edge weights and edge labels are not shown in the explanation graph in order to keep the explanation simple.

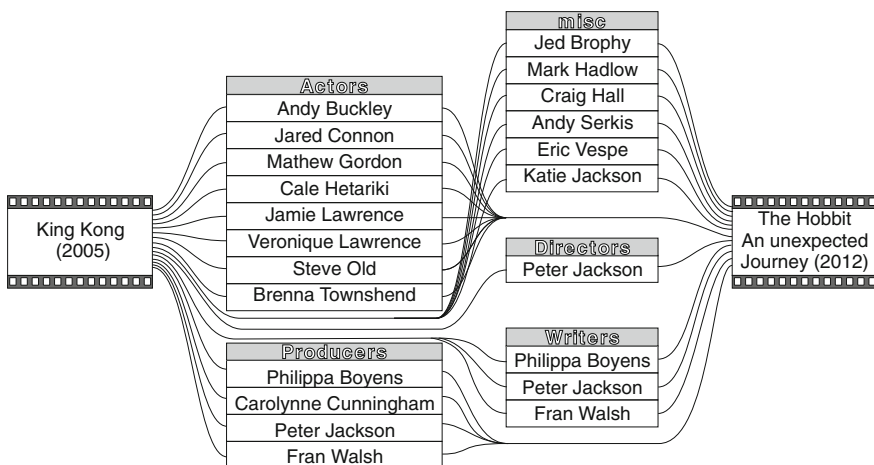
The disadvantages of memory-based recommenders are that these recommenders do not provide strategies for handling inconsistencies and noise. Due to the runtime complexity of memory-based recommenders only short paths can be taken into account.

In a nutshell, memory-based recommender approaches should be used, if the data change frequently, the data do not require a model-based cleanup, and if human-readable explanations should be provided.

### 5.4.2 Model-Based Recommender

Real-world datasets are often huge, noisy, and sparse [26, 27]. Thus, adequate models are needed to extract the relevant information and to remove irrelevant data (e.g., noise).

The computation of the *low-rank approximation* of the graph's adjacency matrix is a popular technique for extracting the dominant information from a graph [17]. The approximation can be implemented by calculating the singular value decomposition (SVD) of the adjacency matrix  $A$  of the graph  $G$ . In order to focus on the most dominant information, the first  $k$  latent dimensions are taken into account.



**Fig. 5.3** The figure visualizes the explanation for a path-based recommendation (used in our movie recommendation web application). The graph shows the nodes and the edges taken into account when suggesting *The Hobbit (2012)* for the user input *King Kong (2005)*

$$A = USV^T \cong U_k S_k V_k^T$$

The adjacency matrix  $A$  is decomposed into a diagonal matrix  $S$ , containing the singular values of  $A$  in descending order. The matrices  $U$  and  $V$  consist of the left-singular and right-singular vectors for  $S$ . The low rank approximation of  $A$  considers only the largest  $k$  singular values of  $A$  and the respective singular vectors ( $U_k, V_k^T$ ).

The SVD-based approach allows us an efficient reduction of the adjacency matrix  $A$ . It has been shown that the low rank approximation is a good model for large sparse matrices [18]. The low-rank approximation of the adjacency matrix allows us to consider long paths in the graph (by computing the powers of the matrix  $A$ ).

The disadvantages of the SVD-based low-rank approximation is that the approach is resource-demanding and highly depends on the applied scaling approach for the matrix  $A$ . In general, dataset updates require a re-calculation of the model. Moreover, the SVD-based approaches use nonreversible projections making it difficult to provide human-understandable explanations.

**Cluster-based Models:** Clustering is an alternative approach for reducing the dataset complexity. It is based on the assumption that similar entities should be aggregated in order to reduce the number of distinct entities. The clusters focus on the characteristic properties the objects (aggregated in a cluster) have in common and abstract from noise. Clustering is a very flexible approach since the similarity measures can be chosen in a wide variety of distance functions. Dependent from the respective dataset different clustering algorithms (e.g., K-Means-Clustering, Hierarchical clustering [30]) can be applied. The concept of clustering is well understood by many users. This enables the generation of human readable explanations based on clusters.

In summary, clustering is a flexible, well-accepted approach for reducing the complexity of a dataset. Depended on the clustering algorithms and the similarity measures the degree of aggregating the entities can be controlled. In general, the definition of adequate clustering strategies and similarity measures requires expert knowledge in order to match the specific characteristics of the recommendation scenario.

**Models for Text-based Recommenders:** Semantic recommender approaches focus on entities explicitly connected by labeled edges. In many real-world scenarios comprehensive textual meta-data for entities exist. For example, in the movie domain plot descriptions and reviews are available. The textual descriptions can be used as an additional knowledge source when analyzing the semantic relation between entities. The similarity between two texts can be computed based on the number of common words or by counting the number of common entities (using Named Entity Recognition and Named Entity Disambiguation algorithms [20]).

Since textual descriptions do not only contain keywords, but also grammatical structures (such as articles and conjunctions) having only a very small impact on the content, texts should be preprocessed before computing the relatedness between two texts. Techniques used for preprocessing natural language texts are stop word removal and stemming that efficiently reduce the vector space spanned by the words of a set of given texts. In addition, these techniques improve the quality of the similarity computation due to the fact, that the words having no semantic meaning are ignored.

### 5.4.3 Ensemble-Based Recommender Approaches

Instead of creating one *universal* recommender for one large graph, we consider learning several different recommenders for sub-graphs and combining them in an ensemble. Ensembles have the advantage that the recommenders in the ensemble have a lower complexity and recommenders can be incrementally updated. In addition, new algorithms can be integrated in order to cover new aspects. For example, results from recommenders optimized for encyclopedic knowledge and recommenders trained on personalized ratings can be combined in an ensemble. The strength of ensemble approaches is that different algorithms can be combined considering the heterogeneity within the graph.

The disadvantage of ensembles consists in the overhead for managing different algorithms in the ensemble and in the additional effort for combining the suggestions from different algorithms.

In summary, ensemble approaches allow us the flexible combination of optimized recommender algorithms. Ensemble approaches often enable improving the recommendation quality as well as the trust in the system [23].

## 5.5 A Semantic Movie Recommender

We evaluate the developed approach in a web-based movie-recommender application. The system has been created based on our semantic movie dataset aggregating data from MOVIELENS, FREEBASE, and IMDB. The recommender system suggests users interesting movies based on user-defined lists of favorite movies. The recommendations are computed using agent ensembles combining the suggestions from different semantic graphs.

In this section we explain the system architecture, present the graphical user interface, and discuss the advantages for the users.

### 5.5.1 The System Architecture

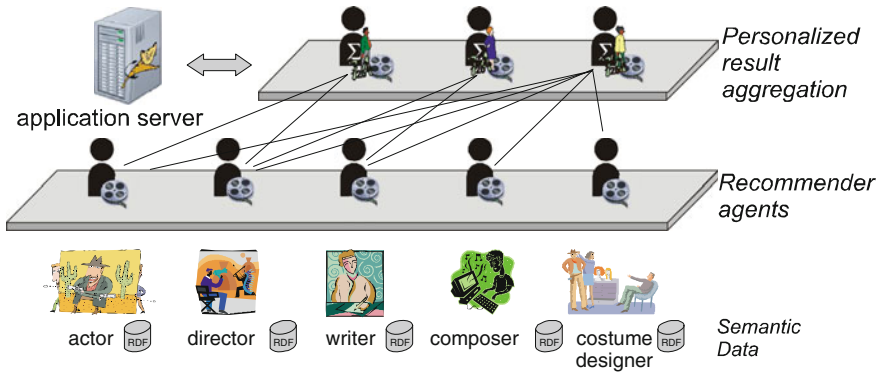
We implement the movie recommendation system as an open, extensible web application. The user interface is implemented using GRAILS<sup>6</sup> [25] running on an APACHE TOMCAT<sup>7</sup> web server.

The system architecture is visualized in Fig. 5.4. Each semantic relationship set is wrapped by one agent allowing us updating and adding semantic relationship. The recommender agents are optimized according to the specific properties of the wrapped semantic relationship sets. In order to provide personalized

---

<sup>6</sup> <https://grails.org/>.

<sup>7</sup> <http://tomcat.apache.org/>.



**Fig. 5.4** The movie recommender system is implemented as an agent-based system. Each semantic relationship set is wrapped by one recommender agent tailored to the specific properties of the respective semantic dataset. The suggestions from the recommender agents are aggregated taking into account the individual user preferences

recommendations, there is one agent for each user managing the individual user profile and the selection of the recommender agents for each request.

Incoming user requests are handled by the web server. When a user requests recommendations, the web server delegates the task to the personal agent. This agent sends the request to all recommender agents relevant for the current user. The results from the recommender agents are collected and aggregated into a final result list. Finally, the aggregated list (might be filtered by user-defined criteria, such as *motion picture rating* or *popularity*) annotated with additional information is presented to the user. For each suggested movie the system provides an explanation describing the semantic relations between the suggested movie and the user's favorite movies (explicitly defined by the user).

### 5.5.2 The Graphical User Interface

Due to the fact that the developed recommender system is based on a semantic graph, users must define the preferences by selecting preferred entities (graph nodes). Our system handles the problem by suggesting users the entities matching best the user input. This approach efficiently supports the users in finding the preferred entities and avoids problems with ambiguous entities. Figure 5.5 shows an example for the *auto-completion*, suggesting movies matching the user input based on the textual similarity.

Based on the defined query the system calculates the recommendations considering several different semantic relationship sets. Thus, it computes the entities most strongly related according to the semantic relationship sets. Our movie recommender system aggregates nine semantic relationship sets: Actors, Directors, Misc (e.g., stuntman, location scouts, and caterer), Composers, Producers,





**Fig. 5.5** The figure shows the *auto-completion* function suggesting the user entities matching the user string input. This avoids problems with ambiguous queries and allows the system to compute recommendations directly on the semantic graph

Authors, Genres, Keywords, and Plot Descriptions. The recommendations computed based on the different semantic relationship sets are combined into one *mixed* list. The combination takes into account individual user preferences by giving a higher weight to recommender agents that provided recommendations the user liked in the past.

**Browsing the results:** Our recommender system computes relevant movies based several semantic relationship sets and combines the results in one list. Thus, on the first view the user finds a list of recommended entities aggregated according to the individual preferences. Due to the fact, that most of the suggested entities might unknown to the user, the system provides for each recommended movie a trailer (a YOUTUBE video), a movie description (retrieved from FREEBASE), and a detailed list of actors, directors, producers (retrieved from IMDB). In addition, the system provides an explanation, visualizing how the recommended movie is related to the user query (or user profile entries). Figure 5.6 shows a screenshot of our web application illustrating the generated explanations.

Advanced users interested in the details might browse the recommendations from each recommender agent (*semantic relationship*) in detail. This feature allows the user to get different points of view on the recommendations. Users can adapt the weights for the different points of view on the recommendations. Users can adapt the weights for the different recommender agents (controlling the influence of each recommender agent on the mixture) and explore how new weights change the suggestions. In addition, users can define detailed filter options, such as movie popularity, average movie rating and age rating. In our evaluation, the popularity filter has been

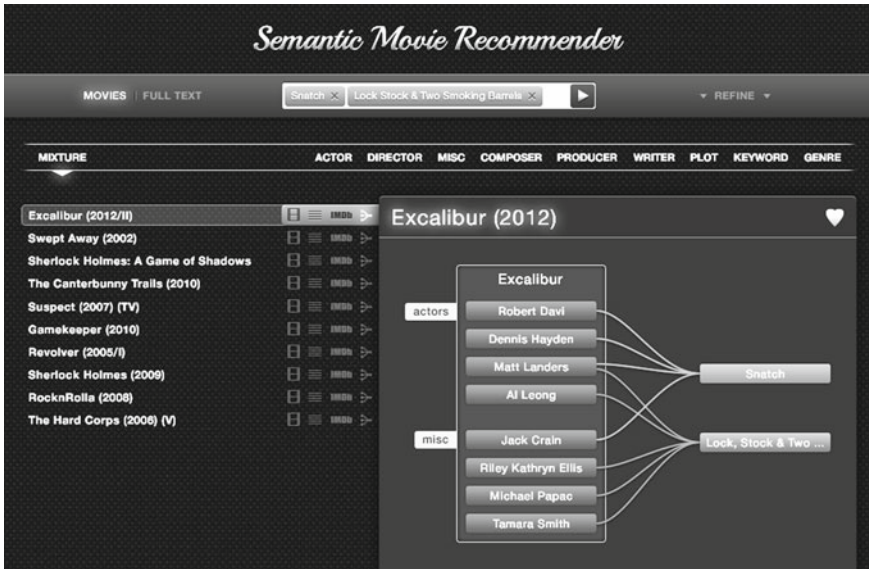


Fig. 5.6 A screenshot of our movie recommender. The system provides detailed explanations based on the semantic dataset in order to improve the confidence in the suggestions

shown as very useful since this filter allows users sorting out movies almost everyone knows and exploring less popular movies (probably unknown to the user). Figure 5.7 shows the options for tuning the requests.



Fig. 5.7 The screenshot shows the options for tuning the request. Users might assign specific weights to each recommender agents or define preferences according to rating and popularity

### 5.5.3 User Study

In order to evaluate our recommender system we conducted a user study (with students from our university). We analyzed the user behavior logged by the web-server and observed the user's interactions with the system. In discussions with participants, we recognized that a difficult problem in recommender systems is how to cope with movies unknown to the user. On the one hand, users expect unknown, "serendipitous" items (following the idea that a recommender should provide new suggestions); on the other hand, users cannot rate the recommendation quality of the system, if all recommended items are unknown. To handle this problem, the meta-recommender (aggregating the results from the "simple" recommender agents) should provide a diverse set of recommendations ensuring that the result set contains popular as well as less popular entities: Relevant popular entities (probably known to the user) improve the trust in the system. Less popular entities (probably unknown to the user) cover the requirement of providing serendipitous movies. Moreover, the system should provide human understandable explanations describing why a suggested movie matches the individual user preferences. Good explanations encourage the user to accept unknown movies as useful recommendations. Explanations generated based on semantic data are helpful, since they describe the aspects in which a recommendation is relevant to the user (even though the recommendation is not obvious). Additional information, such as movie trailers or detailed movie descriptions or movie posters, is often useful to the user giving a first impression on the suggested items. The user preferences differ from one another. The personalized combination of different recommender agents has been seen as an adequate approach to consider individual preference schemes based on encyclopedic semantic recommenders.

In general, most users liked the developed approach of *faceted recommendation* giving the user many new ideas about potentially interesting movies. The visualization of the recommendations encourages users exploring new facets they have not been aware of before. Users can explore new movies relevant according to the individual preferences. The explanations help users to understand why an unknown movie is a relevant recommendation according to the personal profile.

## 5.6 Conclusion

We presented a semantic movie recommender system that overcomes the problem of traditional recommender systems. The developed semantic recommender system is able to aggregate different types of knowledge (rating/collaborative-based and content-based knowledge) from heterogeneous sources. The wide variety of integrated knowledge prevents the cold-start problem and improves the quality of the provided suggestions. In addition, the system is extensible allowing the system provider integrating additional knowledge resources. Since the knowledge of the semantic recommender system is represented as one big graph (consisting of nodes

and edges), there are no problems arising from the processing of natural language text (such as handling of spelling mistakes and ambiguous queries). In addition, the developed semantic system provides human readable explanations for suggestions based on the sub-graph considered while computing the relevance of items.

The creation of a semantic recommender system is a complex process that reveals several challenges: Starting from selecting and integrating appropriate knowledge sources to aggregating heterogeneous data in a unified graph to learning scenario-optimized recommender models able to cope with the complexity of data and ensuring a fast response time. In this section we discussed different approaches and showed that a semantic movie recommender system can be successfully learned using the developed approach. Since semantic systems represent knowledge based on graphs, the knowledge processing is independent from natural language descriptions. This allows system designers separating natural language methods from the processing of facts. Furthermore, the support for additional languages can be added by integrating labels for new languages to the existing nodes.

Coming back to the initial scenario, the presented semantic movie recommender system provides a powerful solution for the problems that Marc and Clara see in traditional recommender systems. The semantic movie recommender system integrates ratings and content-based knowledge provided by huge knowledge stores. This allows the recommender system to consider fine-grained preferences about favorite composers, actors, and producers. In addition, the recommender system can suggest high-quality movies, still not known to everyone. By considering the age classification of movies (retrieved from knowledge basis for the movie domain), the parent's concerns are encountered that the recommended movies are suitable for the children.

Summing up, the presented semantic recommender system allows us to overcome the shortcomings of traditional recommender systems. The graph-based representation of knowledge enables the aggregation of different types of knowledge and the integration of knowledge from many heterogeneous sources. Based on comprehensive knowledge graphs better recommendations can be computed considering several different facets. This ensures highly useful, serendipitous recommendations. Explanations computed based on the knowledge graph improve the transparency of the recommendation process and the user's acceptance of the recommender system.

**Acknowledgments** This research was supported by the Deutsche Forschungsgemeinschaft, DFG, project number AL 561/11-1.

## References

1. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6) (2005)
2. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: a nucleus for a web of open data, in *The Semantic Web*, Lecture Notes in Computer Science, vol. 4825, ed.

- by K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Springer, Berlin, 2007), pp. 722–735
3. D. Beckett, B. McBride, RDF/XML syntax specification (revised). Technical report. *World Wide Web Consortium (W3C)* (2004)
  4. D. Beckett, T. Berners-Lee, E. Prud'hommeaux, Terse RDF triple language. Technical report. *World Wide Web Consortium (W3C)* (2011)
  5. C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, New York, 2006)
  6. C. Bizer, T. Heath, K. Idehen, T. Berners-Lee, Linked data on the web (LDOW2008), in *Proceedings of the 17th International Conference on WWW* (ACM, New York, 2008) pp. 1265–1266
  7. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, DBpedia—a crystallization point for the web of data. *Web Seman. Sci. Serv. Agents World Wide Web* 7(3), 154–165 (2009). The Web of Data
  8. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08* (ACM, New York, 2008), pp. 1247–1250
  9. S. Bourke, K. McCarthy, B. Smyth, Power to the people: exploring neighbourhood formations in social recommender system, in *Proceedings of the 5th ACM Conference on Recommender Systems, RecSys'11* (ACM, New York, 2011) pp. 337–340
  10. Creative Commons Matt Lee. Creative commons Wiki. Web resource, 15th July 2014. <http://wiki.creativecommons.org/>
  11. M. Dean, G. Schreiber, OWL, web ontology language. W3C recommendation. *World Wide Web Consortium (W3C)* (2004)
  12. M. Fatemi, L. Tokarchuk, An empirical study on IMDb and its communities based on the network of co-reviewers, in *Proceedings of the First Workshop on Measurement, Privacy, and Mobility, MPM'12* (ACM, New York, 2012) pp. 7:1–7:6
  13. GroupLens Research. MovieLens data sets. Online resource, available at <http://www.grouplens.org/node/73>, October 2006
  14. T. Heath, C. Bizer, Semantic annotation and retrieval: web of data, in *Handbook of Semantic Web Technologies*, ed. by J. Domingue, D. Fensel, J.A. Hendler (Springer, Berlin, 2011), pp. 191–229
  15. J. Herlocker, J. Konstan, A. Bochers, J. Riedl, An algorithmic framework for performing collaborative filtering, in *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)* (1999)
  16. J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. (TOIS)* 22(1), 5–53 (2004)
  17. Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems. *Computer* 42(8), 30–37 (2009)
  18. J. Kunegis, A. Lommatzsch, Learning spectral graph transformations for link prediction, in *Proceedings of the 26th Annual International Conference on Machine Learning ICML'09* (ACM, New York, 2009), pp. 1–8
  19. A. Lommatzsch, B. Kille, S. Albayrak, An agent-based movie recommender system combining the results computed based on heterogeneous semantic datasets, in *Proceedings of the 13th GI International Conference on Innovative Internet Community Systems and the Workshop on Autonomous Systems, I2CS'13* (VDI, Düsseldorf, 2013)
  20. A. Lommatzsch, D. Ploch, E.W.D. Luca, S. Albayrak, Named entity disambiguation for German news articles, in *Proceedings of LWA2010 - Workshop-Woche: Lernen*, ed. by M. Atzmüller, D. Benz, A. Hotho, G. Stumme (Wissen and Adaptivität, Kassel 2010)
  21. A. Lommatzsch, T. Plumbaum, S. Albayrak, A linked dataverse knows better: boosting recommendation quality using semantic knowledge, in *Proceedings of the 5th International Conference on Advances in Semantic Processing* (IARIA, Wilmington, 2011), pp. 97–103

22. P. Lops, M. Gemmis, G. Semeraro, *Content-Based Recommender Systems: State of the Art and Trends*, Chapter 3 (Springer, New York, 2011), pp. 73–105
23. R. Polikar, Ensemble based systems in decision making. *Circuits Syst. Mag. IEEE* **6**(3), 21–45 (2006)
24. S.J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edn. (Pearson Education, Upper Saddle River, 2003)
25. G. Smith, P. Ledbrook, *Grails in Action*, 1st edn. (Manning Publications, 2009). ISBN: 978-193398893
26. D.E. Sullivan, B. Smyth, D. Wilson, Preserving recommender accuracy and diversity in sparse datasets. *Int. J. Artif. Intel. Tools* **13**(01), 219–235 (2004)
27. N. Sundaresan, Recommender systems at the long tail, in *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys'11* (ACM, New York, 2011), pp. 1–6
28. The Internet Movie Database Team. IMDb press room, about imdb. web resource, August 2014. Available online at <http://www.imdb.com/pressroom/about/> retrieved on 15th July 2014
29. A. Tizghadam, A. Leon-Garcia, Betweenness centrality and resistance distance in communication networks. *IEEE Netw.* **24**(6), 10–16 (2010)
30. Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets, in *Proceedings of the eleventh International Conference on Information and Knowledge Management, CIKM'02* (ACM, New York, 2002), pp. 515–524

# Chapter 6

## News Recommendation in Real-Time

Benjamin Kille, Andreas Lommatzsch and Torben Brodt

**Abstract** Recommender systems support users facing information overload situations. Typically, such situations arise as users have to choose between an immense number of alternatives. Examples include deciding what songs to listen to, what movies to watch, and what news article to read. In this chapter, we outline the case of suggesting news articles. This task entails a number of challenges. First, news collections do not remain relevant unlike movies or songs. Users continue to request novel contents. Second, users avoid creating consistent profiles thus reject login procedures. Third, requests arrive in enormous streams. Having short consumption times, users quickly request the next article to read. Handling these challenges requires adaptations to existing recommendation strategies as well as developing novel ones.

### Coffee Time

Suzanne shivered while looking out of the window. It was one of these cold December afternoons where you just want to stay at home, enjoy a cup of hot coffee, and relax next to the fireplace in the living room. “I hope Laura and Linda will make it on time today” she thought, a little worried about the safety of her friends. It was not the first time for them that they’d miss their little get-together—or “gossip club,” as her husband Steve used to call it. She always complained when he said that, but actually, she secretly had to admit that he wasn’t too far from the truth in the analysis of her circle of friends. They really were gossip! Especially Laura seemed to know everything about everyone in the neighborhood and was more than motivated to share

---

B. Kille (✉) · A. Lommatzsch  
Technische Universität Berlin, Berlin, Germany  
e-mail: benjamin.kille@dai-labor.de

A. Lommatzsch  
e-mail: andreas.lommatzsch@dai-labor.de

T. Brodt  
Plista GmbH, Berlin, Germany  
e-mail: tb@plista.com

her knowledge with anyone who couldn't run away on time. Finally, she saw Laura's car coming around the corner, Linda sitting right next to her. Quickly, Suzanne rushed to the door to welcome her two friends.



“Suzanne, I wanted to ask you...,” started Laura while they enjoyed their first piece of cheesecake. “I heard that you no longer get your newspaper delivered?” Suzanne had to smile. That’s Laura at her best! Always well informed about almost anything. “Yeah, that’s true,” she replied. “We realized that we read most of the news that we are interested in online anyway. So by the time we can read it in the newspaper, we could have read it online already.” “What about local news?”, Linda asked getting curious. “Can you get even those online?” “Sure, there are plenty of websites dedicated to all varieties of news. I have even found a website providing news about gardening. You know how much I like to rearrange my garden”, Suzanne replied and pointed out of the window. “How do you find articles that are relevant within the masses of contents published online though?”, Laura asked. Suzanne kept silent for a while. “That is actually a hard task. Usually, I just browse the home page of a selection of news portals until something catches my attention.” Laura raised her hand, indicating that she wanted to say something after swallowing the piece of the cheesecake which she had just lifted from her plate. “I prefer the old-fashioned newspaper,” she said after her throat was empty again. “You get a piece of all categories of importance. You do not miss any substantial story.” Linda nodded her head. “That may be true”, she added, “but I could certainly relinquish reading all those sport and business related articles.” “Think of all the trees that had been logged for nothing!” Suzanne argued, causing her friends to roll their eyes in amusement. “I do prefer to have a newspaper in my hands which I can flip myself” Laura mentioned. “With the current generation of tablet computers you can almost get the same feeling. And you do consume less paper.” Suzanne argued. “And you can actually search for terms and thus avoid to parse the text manually.”

At this moment, the deafening noise of the teapot whistle interrupted their little chat. Suzanne went to the kitchen and returned with three cups of tea. Laura was



the first to continue talking. “How are Carl and Carla doing?” she asked Suzanne. “Well, Carla just started her internship at that newspaper office. She is doing great. On the other hand, we are a bit worried about Carl. He seems to have a hard time in school.” “I am sorry to hear that,” Linda intervened. “I have read that the government is planning to revise the lessons plan. Perhaps, the lectures will become less difficult in the next term.” she added. “Where did you read that?” Laura was getting curious. “Actually, I read that online. While I was browsing my favorite news portal, I stumbled upon a section named *suggested readings*. There I found the article” Linda replied. Suzanne’s curiosity began to stir. “How exactly does this work? I mean, how does the news portal select articles that it deems relevant for you?” she asked Linda. “I have no idea.” Linda answered. “There were also four or five other suggestions which I found quite uninteresting” she added. “I think, I have seen this type of suggestions on different news portals.” Laura told the other two. “I could not find any article of interest though” she noted. The three started to think about occasions where they had seen similar services. Suzanne was the first to notice that her favorite online shop did offer a list of recommended items. The women agreed that this service had been in place for a couple of years at least. Conversely, suggested reading on news portals appeared to them as comparably new features. “It has to be much harder to suggest news compared to products in an online shop” Laura claimed. “Why is that?” Linda asked. “You have to consider that the online shop knows who you are after you bought something. As you log in you identify yourself. Conversely, the news portal does not know much about you, does it?” Laura explained. “You are right” Linda agreed. The three women started to discuss how they would suggest articles to one another. “You cannot go wrong suggesting Linda articles about animals” Suzanne claimed. All three started to laugh cheerfully. “I would suggest all articles about the latest gardening trends to you, Suzanne” Linda returned her joke. The women realized that they knew each other well after all these years.

## 6.1 Introduction

News reading behavior is considerably shifting toward online consumption. More and more users appreciate the advantages of reading news online. Users enjoy instantaneous access to breaking news. Conversely, old-fashioned newspapers delay access to breaking news due to the printing and distributing process. In addition, newspapers dictate the selection, quantity, and source of news which they comprise. Editors decide about which events to include, their articles’ position in the layout, and what space they may cover. However, anxiously editors prepare their newspapers, users’ preferences vary too extensively to consider the result a perfect fit for them. Users may request more information about certain events that exceed the available space. Further, users may enjoy reading articles enlightening events from different perspectives. Newspapers rarely publish several articles about an individual event. Additionally, users may prefer the writing styles, content focus, or presentation of different journalists and newspapers. For instance, users may prefer reading local news from

a residential news source. Mainstream news sources may not cover their local events at all. Simultaneously, users may read sports-related news rather from mainstream news sources as they can afford journalists to travel to these events. Hence, users require services which online news portals provide in contrast to their analogous counterparts.

As a consequence thereof, users increasingly face the information overload problem. Recommender systems have established as the suited means to overcome information overload. They filter available items thus reduce the decision problem significantly. Users avoid to browse large sets of items. Instead, recommender systems provide a small fraction of items which they deem most relevant to the user at hand. Research has focused on recommender systems in terms of preference elicitation methods [55]. In the context of news, users have rather preferences for latent concepts than actual items. Recommendations of products such as movies, songs, or books differ from news article suggestions in this aspect. In the following, we present a recommendation method that allows dealing with requirements inherent to news recommendations. These requirements include dynamic item collections, incomplete user profiles, and differences between individual news portals.

Dynamic item collections refer to the rates at which items either enter or exit the systems. Editors add novel news items as they emerge to provide readers with information about recent events. On the other hand, news articles decrease in relevance over time as more and more users become aware of them. News collections exhibit much higher addition/deletion rates compared to collections of movies or songs. Users may want to reconsume their favorite movies or songs. Contrarily, readers will seldom read old news articles again.

Recommender systems' quality depends on how well their models reflect user preferences. Typically, system operators require users to create explicit profiles by design. Thus, they are able to feed preference directly linked to a specific user. Contrarily, news portals do rarely require explicit profiles to be created. Supposedly, readers are unwilling to spend time creating profiles. Privacy concerns represent another reason keeping users from providing their personal information. News portal operators tend to identify their users with session identifiers. However carefully they monitor session identifiers, user profiles may contain errors. We mention three kinds of such errors. First, readers may use several devices to consume news items. For instance, they may read news on their tablets as well as their desktop computers. News portal operators will struggle as they seek to merge these profiles based on session keys. Second, readers may share their computers with other. For instance, a couple which lives together might use the same computer for browsing news. Thus, a profile emerges which captures not one but two preferences. Third, users may block the session monitoring due to privacy concerns. Thus, the system operators monitor various users which they cannot differentiate.

Having spent time and resources to build a user profile, users expect to benefit of adequate recommendations. Conversely, users may consider continuing using the system and not abort. On the other hand, news readers behave differently. Users may choose to frequent several news portals. Consequently, users' profiles scatter over various domains. Incomplete profiles impede creating suggestions. The less

information is available about the user at hand, the harder it becomes to select relevant readings.

A set of challenges arises to news recommender systems based on the specific characteristics of news. *What news item reflects a certain latent interest best?* We discuss strategies to deal with the dynamics of news. *How to link interactions to users profiles split over a variety of news portals?* We present ways to construct user profiles representing preferences that allow to provide relevant suggested readings. *How to handle the velocity, veracity, variety, and volume of large streams of interactions of popular news portals?* We elaborate on techniques to cope with big data requirements in the context of news recommendation.

This chapter is structured as follows. Section 6.2 introduces previous research on news article recommendations. Subsequently, we present specifics of our use case in Sect. 6.3. These characteristics include technicalities and requirements as well as system particularities. In Sect. 6.4, we show results of observing how users consume news online. We cover essential aspects including sparsity, popularity bias, as well as contextual factors. Section 6.5 illustrates recommendation algorithms which have been applied to a variety of recommendation problems. We discuss how individual methods suit news recommendation. Likewise, we highlight aspects impeding the application of certain methods. Section 6.6 details design choices faced as we seek to evaluate the performance of recommendation algorithms. Finally, we conclude and give an outlook to future research directions in Sect. 6.7.

## 6.2 Related Work

News portals have evidentially changed the way we consume news. This section presents related research dedicated to support users consuming news. Billsus and Pazzani [8] refer to four types of systems which have developed to support us consuming news. First, they introduce systems which enable personalized access to news. The personalization manifests as news portals present varying news items depending on individual preferences. News recommender systems rank among this kind of systems. Second, Billsus and Pazzani list adaptive news navigation systems. These systems control how news stories link together. Ideally, they reduce users' efforts to turn back to home pages before continuing reading. Third, Billsus and Pazzani mention contextualized news systems. These systems present their contents depending on users' current contexts. Context includes aspects such as location, time, and current interests. Finally, they introduce news aggregation systems. These systems take collections of news articles and automatically extract the very essential information. We focus particularly on systems recommending news articles. These systems became invaluable supportive to online news readers as more and more news became available. This growth induced an information overload problem. Recommender systems represent a specific kind of information filter. Information retrieval systems filter information contained in document collections having received a query [39]. In contrast, recommender systems attempt to learn preferences from previous interactions

to avoid explicit querying. This feature becomes particularly helpful in situations where users lack a defined information need. Instead, users require systems to provide information that will likely be of interest to them.

Researchers have proposed a variety of ideas to carry out the selection process. The ideas range from rather simplistic approaches to highly sophisticated methods carrying a plethora of parameters with them. Trivial methods include randomly recommending items as well as suggesting items based on their popularity. Two paradigms cover a large fraction of the more advanced methods: collaborative filtering and content-based filtering. The former builds on the idea of leveraging other users' preferences to provide recommendations. The latter strives to discover items whose contents share similarities with items users have liked in the past. A comprehensive discussion of both exceeds our scope. Still, we present a selection of ideas tailored for the news domain. We refer readers interested in recommender systems in general to [1, 43, 52].

Proposed news recommendation approaches either utilize other users' interactions with news portals, (possibly enriched) news contents, or both. Thus, we recover both paradigms of regular recommender systems.

Liu et al. [40] introduce a Bayesian framework to allow hybrid recommendations of news articles to users in a personalized fashion. They showed that considering content features increased news consumption compared to a collaborative filtering baseline. Li et al. [38] model news recommendation as a contextual-bandit problem. They show that replaying recorded interactions enables researchers to consistently evaluate their recommendation methods. They provide the theoretical foundations for the unbiasedness of such a methodology. De Francisci et al. [21] make use of three kind of inputs to their news recommendation system. First, they consider interactions in terms of clicks. Second, they extract contents from micro-blogs. Finally, they consider the social relation between the micro-blogging service's users. They represent the problem as learning to rank task. The proposed method considers all three factors to adjust the ranking of news articles for target users. Son et al. [57] propose to consider users' current locations to improve the news item selection process. Additionally, the authors utilize semantic data to enrich the representations of users' interests and locations' relevant concepts. Capelle et al. [14] investigate whether semantic similarities between named entities in news articles can be leveraged to improve recommendation quality. The method requires name entity recognition as a preprocessing step. Bogers and van den Bosch [9] propose a probabilistic framework to provide better news suggestions. Their work looks at the problem from an information retrieval perspective. They analyze the impact of the selected relevance model on the recommendation quality. Li et al. [37] propose a personalized news recommendation framework. Their work emphasizes the issues arising due to the dynamics inherent in item collections. Consequently, they propose to represent the recommendation task as a contextual bandit problem. Li and Li [35] propose to leverage co-occurring interactions to improve news recommendations. Their method models relations between concepts in news texts as hypergraphs. The approach considers both user behaviors and contents. Garcin et al. [25] investigate whether context trees enable news recommender systems to provide relevant news

items to anonymous users. Their method builds context trees based on observed user behaviors. The authors pay particular attention toward recommending novel and diverse items. Cantador et al. [12, 13] leverage two kinds of data to select more relevant news items. On the one hand, they derive semantic concepts from an existing ontology. This represents a content-based approach. On the other hand, they use contextual features to better account for recent trends. Das et al. [17] present insights from a large-scale news recommendation system operated by Google. Their work emphasizes the requirements which operating recommender systems face. They discuss how algorithms including MinHash and probabilistic latent semantic indexing enable news recommender systems to apply the collaborative filtering paradigm in large-scale settings. Montes-Garcia et al. [46] propose a news recommender system tailored specifically towards the needs of journalists. Their approach pays particular attention toward personal preferences as well as contextual factors. Gao et al. [24] analyze how well micro-blogs support news recommendation by indicating trends in an early stage. They investigate the trade-off between popular news and personal tastes. Phelan et al. [47] present a socially-driven news recommendation service which extracts data from micro-blogging services as well as RSS feeds. The authors compare whether RSS contents, micro-blog contents, or a combination of both lets news recommendation services select the most relevant news items. Kompan and Bielikova [32] present a news recommender system based on content similarities. The authors discuss the importance of low computational complexity induced by short response times. Lv et al. [44] propose a method utilizing a variety of factors to estimate articles' relatedness. These factors include relevance, novelty, connectivity, and transition smoothness. For a detailed survey on personalized news recommendation algorithms, we refer the reader to Li et al. [36].

Evaluating recommendation algorithms depends on a variety of factors. First, we have to define the recommendation algorithm's objective. This entails specifying the notion of a good recommendation. At first, this may appear trivial. Researchers have come up with several different specifications. Recommender systems attained increased attention with the "Netflix Prize" challenge [7]. This competitions seeked to reduce the error rate when predicting users' preferences for movies. The organizers decided to use the root mean squared error to compensate for larger deviations. Subsequently, researchers continued to optimize rating prediction scenarios [18, 29, 33, 50, 53, 58]. In addition, researchers started to define recommender systems as ranking mechanisms. They argued that recommender systems ought to rank items according the user preferences. Accurately estimated preferences yield such rankings. Still, they do not constitute an essential input as long as algorithms keep the pairwise order of preferences. Optimizing metrics including normalized discounted cumulative gain (nDCG) and mean reciprocal rank (MRR) provide such rankings [41, 51, 56, 60]. Some researchers argue that users refute to consider all available items. Instead, users limit their attention toward few most relevant items. We find evaluation criteria accounting for these desires in the field of information retrieval. Hereby, systems cut rankings at a pre-defined position. We measure recommendation quality in terms of precision, recall, or a combination thereof [4, 16, 19, 30, 49, 61]. In addition, evaluations may consider further factors determining systems' qualities.

These factors include diversity [34], novelty [60], stability [3], and scalability [5, 54, 58]. Having decided which criteria to optimize, we face another design choice: Do we rely on recorded data or do we aim to interactively conduct experiments with users [27, 55]? Both alternatives have advantages. Offline experiments entail little costs. Additionally, other researchers can reproduce results as the data used for evaluation is fixed. Conversely, conducting experiments with actual users may better reflect the actual use-case. User studies as well as deploying novel algorithms into existing recommender systems represent two alternatives for online experimentation.

Related work covers a wide spectrum of news recommendation's aspects. Most recent works focus on two of these aspects. First, researchers seek to improve recommendation quality by using additional data sources. These sources provide textual descriptions, interaction with users, and social relations. We still cannot satisfyingly tell how to determine additional data's value in advance. Second, research investigates potentials to algorithmically improve recommendations. Due to inherent requirements, we struggle to transport established, sophisticated methods to the news domain. Besides these two major aspects, researchers seek to discover better evaluation protocols along with means to deal with the real-time character of news recommendation.

### 6.3 The Plista Case

We introduced recommending news articles as a challenge for science and industry in Sect. 6.1. Subsequently, we outlined methods enabling news portals to suggest news articles in Sect. 6.2. Both occurred on a rather abstract level. In this section, we present an actual news recommendation scenario. The scenario focuses on the plista GmbH. Plista runs a content and advertisement recommendation service on thousands of premium websites. These websites include portals dedicated to news and entertainment among other topics. Having a large customer base, plista processes millions of user visits on a daily basis. Each visit has to be handled in real-time as web portals attempt to instantly deliver their contents. Portals include recommendations by means of a widget.

The quality of their recommendations represents a major asset to plista. Users accepting recommendations do not only provide revenues. Evidence for increased visitor satisfaction facilitates acquiring new portals to serve with recommendations. Consequently, plista continuously seeks to improve their recommendation algorithms. Similarly, *Netflix* sought to improve their movie recommendations thus releasing a large rating data set in 2006. The *Netflix Prize* competition has shown that combinations of recommendation algorithms provide better recommendations [6]. Combinations of algorithms have shown to better reflect contextual factors [2]. Hence, plista seeks to acquire new algorithms thus improving their system's recommendation quality.

Acquiring novel algorithms represents an endeavor to *plista*. In contrast to Netflix, *plista*'s item collections are subject to continuous changes (see Sect. 6.4). Thus, an algorithm which performs well on news of two months ago could provide inadequate suggestions today. We cannot guarantee that an algorithm will achieve similar performance on novel items. As a result, *plista* created a platform providing researchers and practitioners with access to actual interactions. The platform was first released in 2010 as the "Open Recommendation Platform" (ORP) for internal usage. ORP allowed *plista*'s recommendation engineers to conveniently add novel recommendation algorithms to their eco-system. Three years later, *plista* opened the platform for interested researchers and other third parties to evaluate their recommendation algorithms. Moreover, the platform supported *plista* to stay connected with the research community and actively exchange ideas. ORP ought to provide a representative selection of news portals. Otherwise, evaluations may include biases toward certain aspects. Thus, *plista* directly included two large-scale general news portals along with a selection of minor, rather topic-specific clients. ORP enables participants to interact with real users in a real-time setting. Interaction takes place in a two-stage process. First, news portals visitors load a news page initiating a request for recommendations. Second, the participants' server receives the requests and returns a list of suggested news items. The news portal embeds the list in the news page shown to the visitor. This setup reflects a genuine use-case. Methodologically, we refer to such settings as "living labs". This is due to the unpredictability of future interactions. Note that ORP represents a subset of all news portals served by *plista*. Having the idea of ORP in mind, *plista* contacted publishers with whom they had long-term relationships. Insightful discussion covered both advantages and disadvantages of data sharing with and contributions by researchers. *Plista* managed to include a representative group of publishers into ORP. The group of publishers comprises minor, medium, and large scale news portals. Furthermore, the news topics cover general selections as well as news portals providing news for specific subjects. The selection contains some news portals which operate on a similar regional level allowing evaluating recommendation methods which exchange information between domains. The included publishers use different types of widgets. Thus, ORP allows us to eliminate biases due to graphical user interfaces to a certain degree. These biases include position relative to the news article and the number of recommendations among others. We describe major components as well as vital aspects of ORP in the following subsections.

### ***6.3.1 Involved Parties***

News recommender systems concern different interest groups. These groups include news portal operators, content providers, advertising companies, recommendation providers, and visitors amongst others. We outline the individual perspective of each group.

### 6.3.1.1 Visitors

Visitors represent the target group of news recommendations. They require recommender systems to filter relevant items from large collection which they cannot review themselves. Hence, recommender systems provide them value in terms of the returned items' *relevancy*. Systems may determine how relevant visitors perceive suggested news articles with different means. We may conduct surveys asking visitors about the relevancy of their recommendations. This entails high costs. Therefore, we may restrict surveys to rather small proportions of visitors. Alternatively, we may evaluate visitors' dwelling times, return frequencies, or click rates.

### 6.3.1.2 Content Providers

We refer to content providers as editors in the context of news. Editors create and/or select the contents to be distributed through news portals. They require recommender systems to reasonably link news articles. Recommender systems ought not to confuse readers with misleading suggestions but provide relevant resources. This reflects the newspapers paradigm of structuring contents by grouping them in categories. Visitors may expect to receive suggestion conforming to their previous interactions. We may gauge recommendation algorithms quality in terms of *representativeness* from editors' perspective. How well does a recommendation represent the previous interactions? Alternatively, we may consider assessing how quickly visitors find desired contents. For instance, we may count how often visitors immediately abandon contents.

### 6.3.1.3 Advertisers

Advertisers strive to attract visitors. They want them to pay attention to their advertisements and ideally buy their products or services. Typically, advertisers pay per click. Although the click-through-rate fails to reflect their interests. Conversely, advertisers prefer few clicks coinciding with a high conversion rate. Conversion refers to visitors turning to customers. In our use case, we restrict our focus on click rates. ORP does not provide access to data about visitors converting to customers.

### 6.3.1.4 Operators

News publishers pursue two main objectives. On the one hand, they try to distribute informative and/or entertaining news to readers. On the other hand, they seek to maximize their rentability. This causes them to align the targets of users and advertisers. Users have learned to ignore adverts on webpages [11]. Prompting users to continue reading news increases the chances that they will notice adverts. Enlarged dwelling times ought to lead to higher conversion rates. Consequently, news portals' earnings will increase and improve their cost-effectiveness.



### 6.3.1.5 Recommendation Providers

Recommendation providers capitalize on their algorithms. Typically, portal operators pay them by click. Hence, recommendation providers seek to maximize the probability of visitors clicking on their recommendations. Hereby, they face a dilemma which we refer to as “exploration exploitation trade-off”. Recommendation providers prefer to use the methods most likely to maximize click rates. However, even if individual methods have performed successfully in some scenarios, it stays unclear, which method suits the current context best. Consequently, they have to evaluate different methods which in turn may perform worse.

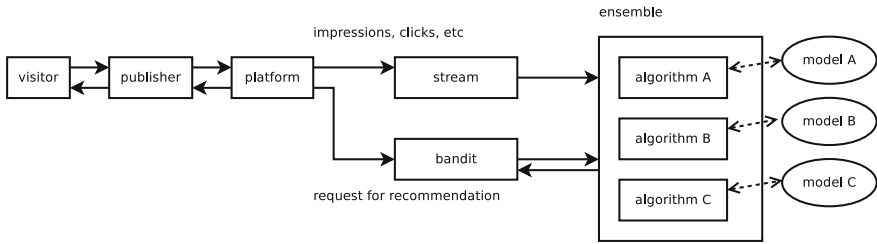
## 6.3.2 Technical Requirements

Plista have an eco-system at their disposal tailored precisely to news recommendation. In contrast, researchers using ORP may have rather limited resources. A selection of technical challenges impedes applying highly sophisticated recommendation methods. Real-time response times represent such a challenge. ORP sets the maximum response time to 100 ms. This affects both computational complexity and model updates. News portal operators require ORP to provide recommendations within a predefined time slot. Exceeding this time slot, they cannot include the recommendation into the displayed web page. Simultaneously, real-time responses require recommendation models to be available at all times. On the other hand, recommendation models ought to include recent news since visitors are likely interested in what currently happens. Thus, operators have to find ways to update their models while concurrently continue to provide recommendations. Thereby, update frequency constitutes a significant parameter. Plista’s observations indicate that decreasing update frequencies negatively affects the click-through rate. Evaluating recommendation algorithms on recorded data (cf. the “Netflix Prize” challenge [7]) cannot cover this time-related aspects. Plista simultaneously runs a variety of recommendation algorithms to account for different factors determining recommendation quality. The system continues updating algorithms as news items appear, new interactions occur, and articles get updated. The frequency with which the system updates algorithms depends on the method. We report findings which Plista observed for certain types of algorithms. Recommenders based on content perform well even when updated in low-frequency. In contrast, collaborative filtering methods require high update frequencies as users’ interests shift. Additionally, collaborative filtering struggles to recommend items which have not obtained interactions. Further, recommendation algorithms suggesting popular news articles performed best when updated with high frequencies. ORP’s users will also have to deal with the technical requirements listed above.

### 6.3.3 System Communication

ORP operates on an event-driven interaction model. Events include visitors requesting recommendations, visitors responding to recommendations by clicking, and news articles added to the collection or updated thereafter. Events occur in predefined contexts. ORP represents context as feature vectors. These vectors comprise information such as publisher, article, categories, and more. Events trigger messages containing the contextual information. For instance, as a user visits a news article, all of ORP's participants will receive a message. Participants may use this information to build their recommendation models. Although, ORP will randomly select an individual recommendation provider to serve this very request. ORP provides participants with an application programming interface (API). The API allows participants to connect their recommendation servers with plista's eco-system. The API uses JSON for data encoding. ORP uses HTTP POST messages to exchange requests including item updates, event notification, and recommendation requests. The contextual data in the ORP is represented through vectors. The system represents such vectors as values mapped to IDs. IDs are represented as integers. They refer to certain types of context. Vectors comprise individual IDs or lists of them. Thus, vectors allow describing an object by layering attributes. ORP distinguishes two types of vectors. One type classifies input vectors while the other refers to output vectors. Input vectors describe the context of events and messages and may be used by the participants for contextual optimization. Input vectors are static and cannot be modified. Output vectors are used to convey information about calculations. During transmission, vectors are grouped together by their type and packaged in a map where the key is the vector's ID and the value related to an instance (depending on its type). The vectors group maps are again grouped together depending on their class. Internally, ORP adapts a multi-armed bandit component. Multi-armed bandit models enable systems to balance the exploration-exploitation trade-off [45]. This trade-off implies that the system fails to accurately estimate recommendation algorithms' performance beforehand. Therefore, the system has to occasionally select seemingly suboptimal strategies to verify that it continues to apply the best strategy. ORP randomly selects recommendation algorithms among active participants. The system disables participating algorithms in case they continuously fail to provide recommendations. Having fixed technical issues, participants can re-establish the communication with ORP and again receive requests. This approach guarantees simple exploration, minimal pre-testing, and low risks of recommenders crashing. Additionally, the system contains a fallback recommender which it activates as participating servers continue to fail.

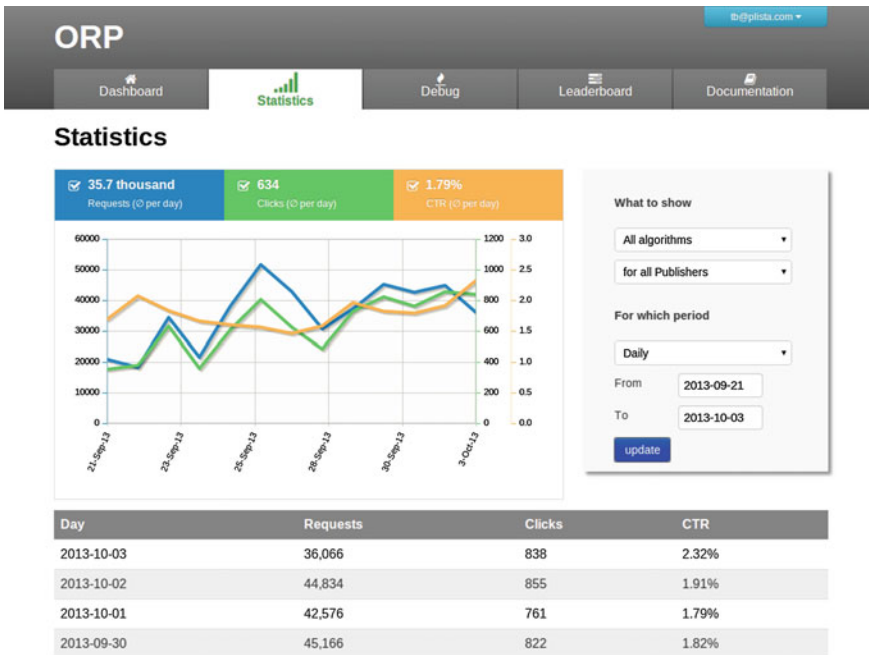
Figure 6.1 depicts the system's structure and its components. Publishers integrate recommendations as static javascript. The javascript loads recommendations by asynchronously querying ORP. ORP returns a widget box captioned "You might also be interested in...", "Recommended articles:", or similar texts. Frequently, ORP includes small pictures next to recommendations. Recommendations consist of a headline and the initial phrases up to 256 words of the recommended articles.



**Fig. 6.1** Integration overview. Visiting a news portal initiates a circular, event-driven sequence of messages. Publishers pass the request for recommendations to the platform. Subsequently, the platform issues the request to recommendation servers. These reply with lists of recommended items. A bandit component blends these lists and forwards the resulting selection via platform and publisher to the visitor

### 6.3.4 Graphical User Interface

ORP supports participants with a graphical user interface displaying their algorithms’ performances. We identify three performance affecting factors: impressions, clicks, and click-through rate (CTR). ORP shows all of them on a daily basis. Impressions



**Fig. 6.2** Illustration of the graphical user interface (GUI) of ORP. The header section offers a sequence of tabs to access different sections. The figure shows the statistic section. ORP displays trajectories of the number of requests, clicks, and their relation. Additionally, ORP provides a table with absolute values for each variable

refer to requests participants received. Clicks represent recommendations which users followed by clicking. CTR describes the ratio of clicks to impressions. ORP's goal is to discover recommendation algorithms which maximize the CTR. Figure 6.2 shows an exemplary dashboard illustrating the clicks, impressions, and CTR graphically and as table. Additionally, ORP provides a leaderboard where participants can compare their performance to others.

### 6.3.5 Participation

There are plenty of reasons for both researchers and practitioners to contribute to ORP. There are hardly any opportunities to get access to actual interactions between users and items. Thus, ORP provides a unique way to evaluate recommendation algorithms. Existing implementations of application programming interfaces facilitate getting started. Plista as well as members of different research institutions have contributed implementations in Java,<sup>1</sup> PHP,<sup>2</sup> python,<sup>3</sup> and Node.js.<sup>4</sup> In addition, we have organized a variety of workshops and competition where researchers along with practitioners published results obtained through ORP. These events include the “International News Recommendation Workshop and Challenge”<sup>5</sup> [59], the “Workshop on Benchmarking Adaptive Retrieval and Recommender Systems”<sup>6</sup> [15], and *CLEF NEWSREEL*, the “News Recommendation Evaluation Lab”<sup>7</sup> [10, 28, 31].

The Open Recommendation Platform provides a unique chance for researchers to evaluate recommendation algorithms with actual user feedback. We have seen which technical requirements it entails. Systems have to reply to request within 100 ms. This prevents plista's performance from dropping below a level where customers suffer substantial losses. ORP commits to open standards with respect to data interchange and interfaces. Researchers and practitioners have already contributed implementations in a variety of programming languages. We encourage researchers to start or continue contributing recommendation algorithms to discover new ways to support users struggling to find relevant news.

---

<sup>1</sup> <https://github.com/plista/kornakapi/>, <https://github.com/plista/orp-sdk-java/>.

<sup>2</sup> <https://github.com/plista/orp-sdk-php>.

<sup>3</sup> <https://github.com/plista/contest-py/>.

<sup>4</sup> <https://github.com/plista/contest-js/>.

<sup>5</sup> <http://recsys.acm.org/recsys13/nrs/>.

<sup>6</sup> <http://www.bars-workshop.org/>.

<sup>7</sup> <http://www.clef-newsreel.org/>.

## 6.4 News Consumption Phenomena

This section introduces a variety of phenomena which we observed as users interact with online news portals. These phenomena distinguish the case of recommending news from other subjects such as movies, songs, or books. We dedicate a subsection to the aspects *sparsity* (6.4.1), *popularity* (6.4.2), *dynamics* (6.4.3), and *context* (6.4.4).

Recommender systems have established in a variety of use-cases. They support users' decision making. Typical use cases include deciding which movie to watch, which song to listen to, and which product to buy. Recommender systems have proved to be valuable in those scenarios. In contrast, suggesting news entails a variety of challenges. We discuss sparsity, popularity biases, dynamic item collections, and contextual factors. These aspects represent the major challenges for operators of news portals running recommender systems.

### 6.4.1 Sparsity

We observe users interacting with items. Interactions cover a range of actions depending on the items. For instance, users may buy products, listen to music, watch movies, or read news articles. We can quantify interactions by the cardinalities of the involved sets of users and items. Let  $u \in \mathcal{U}$  and  $i \in \mathcal{I}$  denote users and items. Further, let  $\text{card}(\cdot) = |\cdot|$  denote the function returning the number of elements contained in a set. Equation 6.1 defines sparsity. Sparsity reflects the fraction of interactions we actually observed by the number of possible interactions. Note that  $\mathbb{I}(u, i)$  represents the indicator function returning 1 if  $u$  interacted with  $i$  and 0 otherwise (see Eq. 6.2).

$$\text{sparsity} = 1 - \frac{\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \mathbb{I}(u, i)}{|\mathcal{U}| |\mathcal{I}|} \quad (6.1)$$

$$\mathbb{I}(u, i) = \begin{cases} 1 & : \text{ if we observe an interaction between } u \text{ and } i \\ 0 & : \text{ otherwise} \end{cases} \quad (6.2)$$

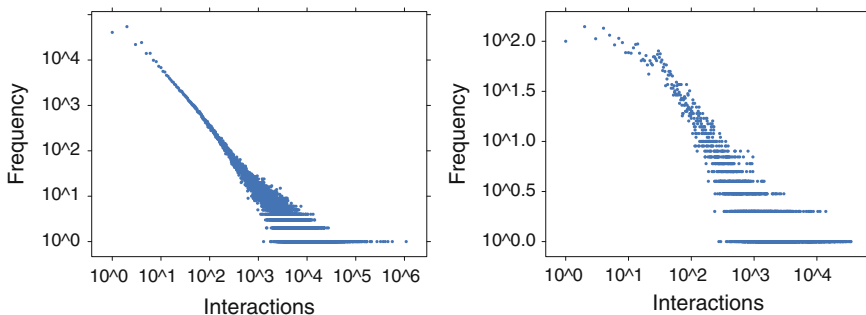
Recommender systems operate on domains with high sparsity. Recommending items with almost complete profiles represents a rather trivial problem. The lack of such comprehensive information induces the need for intelligent suggestion mechanisms. Table 6.1 displays sparsity levels of a selection of datasets. We observe that most datasets include less than 3% of potential interactions. We determine potential interactions by multiplying the numbers of users and items. Additionally, Table 6.1 shows the relation of observed interactions to potential interactions. For instance, the *Netflix* data set exhibits 1 in 86.4 potential interactions. In contrast, we recorded data from two news portals where we observe 1 in 66622.8 potential interactions. This illustrates the difficulty to select appropriate news articles as recommendations.

**Table 6.1** Levels of sparsity for a selection of well-known data sets

Data set	Sparsity	Proportion of interactions	References
Netflix prize challenge	0.98842593	86.4	[7]
Book-crossings	0.99998546	68796.6	[62]
Movielens 100k	0.95840128	15.9	[26]
Movielens 1M	0.98691797	23.9	[26]
Movielens 10M	0.98827612	76.4	[26]
EachMovie	0.97631161	42.2	[58]
Jester	0.43662440	1.8	[58]
Y!Music	0.99915117	1178.8	[20]
<b>News Portal 1</b>	0.99998499	66622.8	
<b>News Portal 2</b>	0.99996663	2996.8	

## 6.4.2 Popularity

We encounter popularity as some items comprise a considerably larger fraction of interactions compared to others. Previous work has documented the occurrence of a popularity bias in a variety of domains. These domains include movies, songs, and books. We have grown accustomed to call popular items with specialized names. “Blockbuster”, “hit”, and “bestseller” refer to such popular movies, songs, and books. Recommender systems consider these type of items as adequate suggestions. We expect visitors to accept suggestions of popular items. The acceptance holds as users’ tastes do not deviate from the majority of users. On the other hand, users may already be aware of the items. In such cases, the suggestion lacks serendipity. We discover popularity biases as we analyze the distribution of interactions over items. Popularity

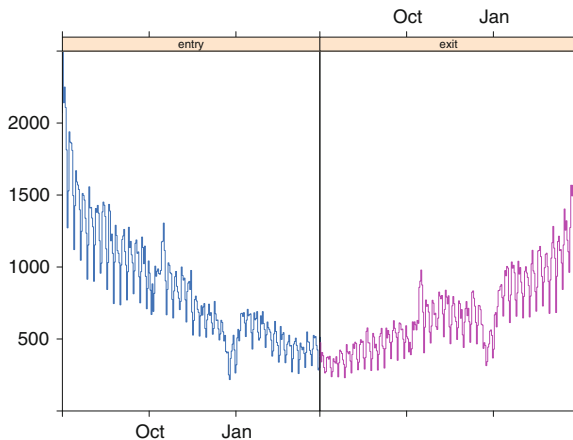
**Fig. 6.3** Popularity distribution for a news portal (*left*) and the Movielens (*right*) data set

frequently induces a power-law distribution of interactions. A power-law distribution manifests as few items comprise a relatively large fraction of interactions. Conversely, a large fraction of items comprises only relatively few interactions. Popularity has been found to affect establishing users' trust into the recommender system [48]. Recommender systems which suggested popular items had a better chance to engage users to interact with them. Figure 6.3 shows the popularity distribution of a news portal's articles along with the Movielens movie rating data set. We observe that both exhibit similar shapes. Few individual items comprise a majority of interaction. Conversely, the majority of items comprises only few interactions.

### 6.4.3 Item Collection Dynamics

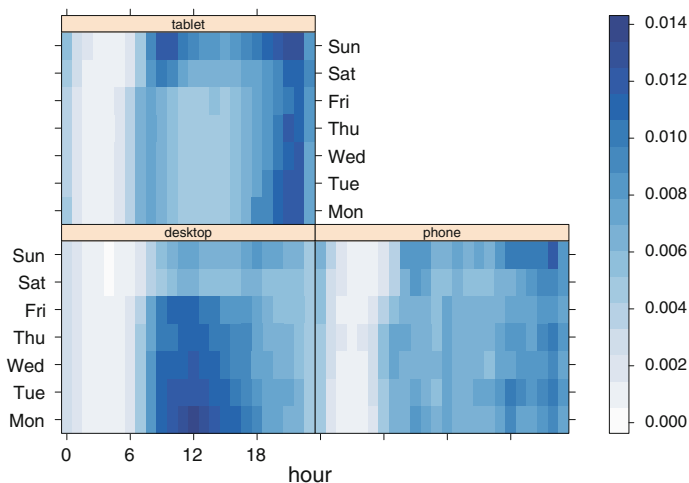
Continuously adding new items to existing collections represents a major reason for the information overload. Additions incur as film studios create new movies, music labels release new albums, or editors publish new books. Some of the novel items may become popular ones attracting plenty interactions. Others may remain barely known. The frequency with which items enter collections depends on the type of item. According to [22], European publishers released about 535,000 books in 2013. In contrast, news articles represent a much more high-frequency type of item. Individual news portals account for hundreds of thousands articles published per year.

News consumption differs from other domains. On the one hand, movies, songs, and books attract users throughout longer periods. For instance, we consider the rating data from the Movielens data set. Each interaction conveys a timestamp. Thus, we compute the duration in between the last and first interaction for each movie.



**Fig. 6.4** Number of items entering and exiting the news portals over time. On the *left-hand side*, the figure shows how many items we observed whom users interact with for the first time. On the *right-hand side*, the figure illustrates how many items we observe no future interactions with

We observe the durations' median at 2,254 days. On average, news articles obtain more than half of their interaction within 24 h after their publication. The proportion of interactions concentrated on the first 24 h even increases for more popular news articles. This illustrates that users as a group consume news more rapidly than movies. On the other hand, users occasionally re-consume books and more frequently movies and songs. Users may willingly trigger the re-consumption as they choose to listen to their favorite songs or watch their favorite movie again. Additionally, broadcasters and television stations tend to re-air popular songs and movies. We have found no evidence that users frequently re-consume news articles. Figure 6.4 displays the evolution of news article collections with respect to user interactions. The data span a time of roughly eight months for a large-size news portal. On the left-hand side, we observe the number of items whom users start to interact with. Note that in the very beginning, there may be previous interactions which our data disregard. On the right-hand side, we observe the number of items for which we do not observe any future interactions. Notice that in the rightmost part, there may occur additional interaction which our data disregard. We observe a down-peak for both entries that exist during Christmas time.



**Fig. 6.5** Relative frequencies of interactions by daytime, weekday, and device. We observe that visitors tend to use their desktop computers in the typical working times (Monday to Friday, between 8 a.m. and 5 p.m.). Conversely, tablets account for relatively more interaction at the evenings as well as on the weekends. Smartphones lack such a clear tendency. All device types have comparably few interaction in the nights



### 6.4.4 Contextual Factors

News consumption is subject to a variety of contextual factors. Our news consumption differs with respect to the time of day, day of week, location, device, mood, and more. Determining the current context represents a difficult problem. In particular, confounding contextual factors impede recognizing situations correctly. Contexts manifest as combinations of contextual factors. For instance, users reading news on a weekday at noon on their desktop in good mood represent a specific context. Altering an individual factor may provide a context requiring a different kind of suggestions. For instance, users reading news on a weekday at noon in a good mood but on their tablet devices may dislike reading comprehensive articles due to their limited screen sizes. Figure 6.5 shows the relative frequencies of interactions grouped by daytime, weekday, and device. The majority of interactions recorded for desktop computers concentrates on the working times. Contrarily, phones as well as tablets account for larger proportions of interactions during evenings as well as weekends. Generally, we observe neglectable proportions of interactions during the night times for all device types. Suppose we ought to select a recommendation algorithm for a particular request. Context represents an important aspect we need to consider. Requests are more likely to arise from mobile devices on the weekend. Mobile devices provide less space to display recommendations on. Thus, we should consult the recommendation method which performs best under these circumstances.

We have seen that sparsity, popularity, dynamics, and context represent major impeding factors for news recommendation. Sparsity hampers establishing valuable user and item profiles. Sparsity represents a particular challenge for newly added users and items. This is due to the system having almost no knowledge about preference relation with the entity. The system struggles to determine what items a new user will like. Conversely, it cannot reliably select potential consumers. Popularity skews consumption distributions as few items concentrate large amounts of interactions. Contrarily, unpopular items see hardly any interactions. Dynamics refer to the system characteristic of fluctuating item collections. In established domains songs, movies, and books remain recommendable items. Conversely, news' relevance fades with time. Finally, systems have to consider users' current context to select enjoyable articles. Users may dislike reading comprehensive articles on mobile devices. In Sect. 6.5, we discuss a selection of algorithms and their abilities to deal with these specificities.

## 6.5 Recommendation Algorithms for News

Recommendation algorithms are subject to a vigorous research community. Researchers continuously propose and evaluate novel methods or extend existing ones. Methods differ with respect to complexity, applicability, and the underlying ideas. In the following, we introduce and discuss four kinds of such underlying ideas and their implementations:

**Table 6.2** Notation used in the algorithmic descriptions

Symbol	Meaning
$\mathcal{I}$	Set of items
$\mathcal{U}$	Set of users
$\mathbb{I}(u, i)$	Interaction indicator function
$R$	Interaction matrix
$\text{top}(k, c, X)$	Function returning the $k$ largest values with respect to criteria $c$ of a collection $X$

- simple methods with low complexity
- collaborative filtering
- content-based filtering
- ensembles of the former 3 notions.

In addition, we highlight the applicability of various implementations for news recommendations. News recommendation entails specific requirements due to their characteristics (see Sect. 6.4). Table 6.2 introduces basic notation which we use in the algorithmic descriptions.

### 6.5.1 Simple Methods

Researchers introduce simple methods as baselines to elucidate the improvements which their novel method provides. Nevertheless, simple methods carry some advantages with them. Typically, simple methods can be easily implemented and exhibit low complexity. Frequently, simple methods target specific factors. In other words, simple methods follow a single idea. For instance, always recommend the most popular item the requesting user has not yet interacted with. We call this simple method the “most popular” recommender. We will elaborate on this method and introduce two additional ones.

#### 6.5.1.1 Most Popular

The *most popular* recommender suggests items according to their popularity. This follows the notion that items comprising interactions with a majority of users will be relevant for other users as well. This resembles lead articles in newspapers—the analogous counterparts of digital news portals. Lead articles obtain more attention than articles situated in latter parts of newspapers. Algorithm 1 describes the procedure to build a model based on the most popular recommender. The algorithm requires a matrix of interactions, the set of items, and the number of items to recommend as input. Subsequently, the method iteratively evaluates the popularity of each item. Items enter the list of recommendations as they are amongst the  $k$  most popular

items. The algorithm can be altered to consider different timeframes by restricting the interactions which it receives (see Sect. 6.4.2).

---

### Algorithm 1 Most Popular Recommender

---

INPUT matrix of interactions  $R$ , set of items  $\mathcal{I}$ , number of items to recommend  $k$

OUTPUT list of  $k$  items sorted by popularity

```

1: for all  $i \in \mathcal{I}$  do
2:    $\text{popularity}(i) \leftarrow \sum_{u \in \mathcal{U}} \mathbb{I}(i, u)$ 
3: end for
4:  $\text{recommendations} \leftarrow \text{top}(k, \text{popularity}, \mathcal{I})$ 

```

---

#### 6.5.1.2 Most Recent

The *most recent* recommender builds upon the notion of recency. As Algorithm 2 illustrates, most recent recommendation ranks items according to their appearance in the collections. The algorithm takes the set of items, their creation time, the current time, and a specification of how many items to recommend as input. Subsequently, we obtain an items age subtracting the date of creation from the current time. The method determines which items to recommend by cutting the list of items ordered by their ages at position  $k$ . As new items enter the collection, they move on top of the list replacing the former top-ranked ones. Thus, the method keeps the items to recommend up to date (see Sect. 6.4.3).

---

### Algorithm 2 Most Recent Recommender

---

INPUT set of items  $\mathcal{I}$ , timestamps of item creation  $\tau(i)$ , current time  $T$ , number of items to recommend  $k$

OUTPUT list of  $k$  items sorted by date of creation

```

1: for all  $i \in \mathcal{I}$  do
2:    $t(i) \leftarrow T - \tau(i)$ 
3: end for
4:  $\text{recs} \leftarrow \text{top}(k, -t, \mathcal{I})$ 

```

---

#### 6.5.1.3 Random

Recommending random items represents another simple method. Randomly picking items yields the risk of suggesting irrelevant items. On the other hand, it could provide access to items which are neither popular nor recent and thus would not have been found by users. Algorithm 3 depicts the random recommendation procedure. It randomly adds items to the list of recommendations until the list has the desired capacity. Items may not be redundant.

---

**Algorithm 3** Random Recommender
 

---

 INPUT set of items  $\mathcal{I}$ , number of items to recommend  $k$ 

 OUTPUT list of  $k$  items to recommend

```

1: while |recommendations| <  $k$  do
2:    $i \leftarrow \text{rand}(\mathcal{I})$ 
3:   if  $i \notin \text{recommendations}$  then
4:     recommendations  $\leftarrow$  recommendations  $\cup i$ 
5:   end if
6: end while

```

---

### 6.5.2 Collaborative Filtering

Collaborative filtering (CF) adapts the notion of taste similarity continuity. In other words, if two users exhibit similar tastes in the past, collaborative filtering assumes that they will continue to prefer similar items. Previous research provides an abundance of algorithms for collaborative filtering. Adomavicius and Tuzhilin [1] distinguish memory-based and model-based collaborative filtering algorithms. Memory-based CF uses all available data for recommendation. In contrast, model-based CF generalizes patterns apparent in interactions and provides recommendations based on these models. Matrix factorization techniques have established among the most successful model-based CF methods.

Algorithm 4 illustrates memory-based recommendation from the user perspective. The method requires the sets of users and items, a similarity function, the number of neighbors to consider, along with the length of the recommendation lists to produce. The algorithm iterates first the set of users to determine whose taste resembles the target user's taste. Subsequently, the method predicts the preferences for each item the target user is unaware of. The algorithm returns the  $k$  items with the highest scores.

Algorithm 5 shows memory-based recommendation from the item perspective. In contrast to Algorithm 4, the method compute similarities between items in terms of their interactions. This is advantageous in cases where  $|\mathcal{I}| \ll |\mathcal{U}|$  since we skip the computational more expensive loops over the larger user dimension.

Matrix factorization has established as one of the most successful type of collaborative filtering. These algorithms reduce the dimensionality of a  $M$  by  $N$  interaction matrix  $R$  to a lower rank approximation. Projecting user and item profiles in this lower space enables recommender systems to compute similarities between them. We present two methods to learn these low rank approximations. Algorithm 6 learns low rank approximations with an alternating least squares procedure. Hereby, we randomly initialize two factor matrices. These matrices' dimension follows the number of users, items, and the desired latent factors. Subsequently, the algorithm iteratively optimizes a target function. This target function measures how close the predicted interactions match the observed interactions. Root mean squared error (RMSE) represents a popular choice for such a function. The algorithm keeps one feature matrix

**Algorithm 4** User-based K-nearest Neighbor CF

INPUT set of users  $\mathcal{U}$ , set of items  $\mathcal{I}$ , similarity function  $\sigma(\cdot, \cdot)$ , number of neighbors  $l$ , number of item to recommend  $k$

OUTPUT list of  $k$  recommended items

```

1:  $u$  ▷ target user
2:  $N \leftarrow \emptyset$  ▷ set of neighbors
3: recommendations  $\leftarrow \emptyset$  ▷ list of recommendations
4: for all  $v \in \mathcal{U} \setminus u$  do
5:    $s \leftarrow \sigma(u, v)$ 
6:   if  $s \geq \sigma(u, N_l)$  then
7:      $N \leftarrow N \cup (v, s)$ 
8:   end if
9: end for
10: for all  $i \in \mathcal{I} \setminus \mathcal{I}_u$  do ▷ ( $\mathcal{I}_u$  refers to items which  $u$  already knows)
11:   for all  $n \in N$  do
12:     if  $\mathbb{I}(n, i) = 1$  then
13:        $\hat{r}_n \leftarrow s_n r(n, i)$ 
14:     end if
15:   end for
16:    $\hat{r} \leftarrow \sum_{\mathbb{I}(n, i)=1} \hat{r}_n$ 
17:   if  $\hat{r} > \text{sort}(\text{recommendations}_k)$  then
18:     add( $i$ )
19:     if |recommendations|  $> k$  then
20:       remove(recommendations $_{k+1}$ )
21:     end if
22:   end if
23: end for

```

**Algorithm 5** Item-based K-nearest Neighbor CF

INPUT set of users  $\mathcal{U}$ , set of items  $\mathcal{I}$ , similarity function  $\sigma(\cdot, \cdot)$ , number of items to recommend  $k$

OUTPUT list of  $k$  recommended items

```

1:  $u$  ▷ target user
2:  $S$  ▷  $|\mathcal{I}| \times |\mathcal{I}|$  similarity matrix for all combinations of items
3:  $N \leftarrow \emptyset$  ▷ set of neighbors
4: recommendations  $\leftarrow \emptyset$  ▷ list of recommendations
5: for all  $i \in \mathcal{I}$  do
6:   for all  $j \in \mathcal{I} \setminus i$  do
7:      $S_{i, j} \leftarrow \sigma(i, j)$ 
8:   end for
9: end for
10: for all  $i \in \mathcal{I}_u^c$  do ▷  $\mathcal{I}_u^c$  refers to all items the target user  $u$  did not interact with
11:    $\hat{r}_i \leftarrow u \otimes S_i$  ▷  $u$  refers to items a user has interacted with
12:   recommendations  $\leftarrow \text{top}(k, \hat{r}, \mathcal{I}_u^c)$ 
13: end for

```

fixed while determining the gradient with respect to the remaining matrix. The algorithm switches matrices in the next iterative step. As soon as a stopping criterion is met, the procedure terminates providing the low rank approximation. Stopping criteria include thresholds as well as maximum iterations. Thresholds define a limit for the improvement between iterations. As we observe less improvement than defined, we terminate the procedure. Conversely, a maximum number of iterations aborts disregarding improvements. Both approaches have advantages. Thresholds guarantee convergence to the desired quality. Unfortunately, this may lead to long running times. In contrast, maximum iterations assure limited running. Still, the algorithm may provide only sub-optimal solutions. We obtain recommendations as we map user and item profiles onto the low ranked subspace.

---

### Algorithm 6 Alternating Least Squares Matrix Factorization CF

---

INPUT interaction matrix  $R_{u,i}$ , number of latent factors to consider  $k$ , termination condition  $\epsilon$ , optimization function  $q(\cdot, \cdot)$   
 OUTPUT predicted interactions

- 1:  $P \leftarrow \text{rand}(|\mathcal{U}|, k)$  ▷ randomly initialize latent user factors
- 2:  $Q \leftarrow \text{rand}(k, |\mathcal{I}|)$  ▷ randomly initialize latent item factors
- 3: **while**  $\epsilon = \text{false}$  **do**
- 4:    $P \leftarrow \arg \max_P q(R, PQ^\top)$  ▷ Optimize  $P$  keeping  $Q$  fixed
- 5:    $Q \leftarrow \arg \max_Q q(R, PQ^\top)$  ▷ Optimize  $Q$  keeping  $P$  fixed
- 6: **end while**
- 7: recommendations  $\leftarrow \text{top}(k, \langle P_u, Q_i \rangle, R)$

---

Algorithm 7 illustrates an alternative way to obtain low rank approximations. Instead of iteratively optimizing user or item factors, the algorithm randomly picks interactions. Subsequently, we compute the gradients for both users and item factors and adjust the factor matrices accordingly. Identical stopping criteria apply to this setting.

---

### Algorithm 7 Stochastic Gradient Descent Matrix Factorization CF

---

INPUT interaction matrix  $R_{u,i}$ , number of latent factors to consider  $k$ , termination condition  $\epsilon$ , optimization function  $q(\cdot, \cdot)$ , learning rate  $\nu$   
 OUTPUT predicted interactions

- 1:  $P \leftarrow \text{rand}(|\mathcal{U}|, k)$  ▷ randomly initialize latent user factors
- 2:  $Q \leftarrow \text{rand}(k, |\mathcal{I}|)$  ▷ randomly initialize latent item factors
- 3: **while**  $\epsilon = \text{false}$  **do**
- 4:    $(u, i) \leftarrow \text{rand}(R)$  ▷ pick random interactions
- 5:    $e \leftarrow q(R(u, i), P_u Q_i^\top)$  ▷ determine prediction quality
- 6:    $P \leftarrow P \cdot \nu \nabla_e P$  ▷ update user factors
- 7:    $Q \leftarrow Q \cdot \nu \nabla_e Q$  ▷ update item factors
- 8: **end while**
- 9: recommendations  $\leftarrow \text{top}(k, \langle P_u, Q_i \rangle, R)$

---

### 6.5.3 Content-Based Filtering

Content-based Filtering (CBF) supposes that users will continue to interact with items that share similar contents. For instance, users interact with songs. The system observes that a user frequents a certain artist. As a consequence, the system suggests other items related to the artist. Algorithm 8 shows the content-based recommendation algorithm. The system requires the set of items, its features, a user profile, along with a similarity function. The algorithm computes the similarities between any combinations of items. Finally, we project the user profile onto the similarity matrix. As a result, we obtain a score for each item. The system recommends the top  $k$  items excluding items the users is already familiar with. This approach directs the major efforts towards the choice of similarity metrics as well as the decision on which features to use.

---

#### Algorithm 8 Content-based Filtering

---

INPUT set of items  $\mathcal{I}$ , item feature matrix  $F$ , user profile  $U$ , similarity function  $\text{similarity}(X, Y)$ , number of recommendations  $k$

OUTPUT similar items

```

1:  $S \leftarrow \emptyset$  ▷ Initialize similarity matrix  $S$ 
2: for all  $do i \in \mathcal{I}$ 
3:   for all  $do j \in \mathcal{I} \setminus i$ 
4:      $S_{i,j} \leftarrow \text{similarity}(F_i, F_j)$ 
5:   end for
6: end for
7: recommendations  $\leftarrow \text{top}(k, \langle U, S, \rangle, \mathcal{I} \setminus U)$ 

```

---

### 6.5.4 Ensembles

So far, we have introduced a variety of recommendation algorithms. These algorithms entail different ideas and require varying data. Machine learning research has shown that combining various algorithms yields potential improvements [23].

In the context of news recommendation, we may combine individual algorithms using different methods. Multi-armed bandits represent such a method [37]. Multi-armed bandits target the problem of uncertainty with respect to the choice of algorithm, parameter, or data. Uncertainty arises as the system cannot determine which algorithm, parameter, or data will perform best. We refer to this problem as “exploration–exploitation” dilemma. The problem manifests as systems try to avoid selecting sub-optimal algorithms, parameter, or data. Conversely, system cannot judge the performance differences between different algorithms, parameter, or data unless they continuously evaluate them against each other. We may define multi-armed bandits in different forms. First, we use them to switch different methods. For

**Table 6.3** Computational complexity of recommendation algorithms for news

Algorithm	Complexity
Most popular	$\mathcal{O}(MN)$
Most recent	$\mathcal{O}(N)$
Random	$\mathcal{O}(N)$
User-based CF	$\mathcal{O}(M(M - 1)N)$
Item-based CF	$\mathcal{O}(MN^2)$
ALS CF	$\mathcal{O}(MNk^2)$
SGD CF	$\mathcal{O}(S)$
Content-based filtering	$\mathcal{O}(MN^2)$

$M$  refers to the number of users while  $N$  refers to the number of items.  $S$  represents an unknown variable which depends on the configuration with which (user, item) pairs are selected

instance, the system switches between implementations of collaborative filtering, content-based filtering, and other methods. Second, we may keep the algorithm fixed. The multi-armed bandit switches parameters in this scenario. For instance, we select item-based collaborative filtering. This algorithm expects inputs including similarity function. Pearson's correlation coefficient and cosine similarity represent examples of such similarity functions. The multi-armed bandit may then switch these. Finally, we may limit the data we use to learn a model representing interaction patterns. For instance, we may argue that with time passing the relevancy of news diminishes. Thus, we may consider various time frames. For instance, we may learn a model based on interactions which occurred up to 3 h, up to 6 h, and up to a day ago. The multi-armed bandit may switch which data to use. Lommatzsch [42] describes a sophisticated way to allay negative effects induced by exploration. The proposed method evaluates all configurations in a slightly delayed time. In other words, instead of averaging performances over time, the method re-issues every request to all configurations. Thus, the system assesses performances more reliably. Consequently, the system learns to select the most promising configuration more quickly. Results show that algorithms performances strongly depend on contextual factors. As a result, individual algorithms cannot dominate other algorithms consistently.

### 6.5.5 Scalability

As discussed in Sect. 6.4, recommending news articles entails technical requirements. In particular, systems must deal with a large volume of requests arriving in high rates. Consequently, recommendation algorithms have to scale at such conditions.

Table 6.3 refers each algorithms to an estimated complexity. Note that intelligent ways of situating data and similar tools may decrease the actual complexity. The table ought to illustrate differences between individual methods. For instance, random and most recent methods operate independent from the user dimension. The complexity



of the more sophisticated methods including ALS and SGD collaborative filtering depends on the stopping criterion. These methods either stop as the optimization target surpasses a threshold or after a predefined number of iterations.

Besides algorithmic optimization, a selection of frameworks enables systems to parallelize their computation thus achieving considerable speed-ups. These frameworks include *hadoop*,<sup>8</sup> *spark*,<sup>9</sup> and *storm*<sup>10</sup> amongst others. Additionally, news recommender system operators may consider to pre-compute recommendations as soon as possible. For instance, they may estimate the probability that a novel article will become popular. If the probability estimate is sufficiently high, the system could start recommending it more often.

## 6.6 Evaluation Criteria

This section treats aspects related to news recommender systems' evaluation protocols. Section 6.2 discussed aspects which we need to consider when evaluating news recommender systems. First, we have to define quality criteria. These criteria relate to the use-case introduced in Sect. 6.3. We aim to assess how visitors, advertisers, as well as operators benefit of having the recommender system in place. ORP does not reveal information about earnings or users converted to customers. Hence, we rely on the interactions which we observe. These interaction represent implicit preference indicators. In contrast, users may explicitly rate items on a pre-defined scale. Lacking such graded feedback, we dismiss error-based metrics—e.g., RMSE, MAE—as we disregard ranking-based criteria including normalized discounted cumulative gain (nDCG) and mean reciprocal rank (MRR). Measures used in information retrieval dispense with numerical preferences. Recall and precision require knowing whether or not a certain item is relevant to a user. Our observations fail to provide such information for all (user, item)-pairs. We may infer relevancy as users select news articles. Still, articles remain ambiguous until we observe interactions with users. Have users missed to see the article? Have users seen the article and decided not to read them? We can evaluate search engine as we predefine each document's relevance given a query. Unfortunately, we have no analogous concept for recommender systems. This is due to individual users' varying preferences. We cannot tell whether a specific news article interests a user unless the user reads it. Thus, we adhere to the notion of click-through-rates (CTR). CTR relates the number of clicks to the number of requests which the recommender system received.

ORP supports evaluating recommendation algorithms by means of live interactions with users. Additionally, we may record such interactions. Subsequently, we can use these records to replay the stream of interactions. We can apply various recommendations methods and assess their qualities having future click events recorded.

---

<sup>8</sup> <http://hadoop.apache.org/>.

<sup>9</sup> <https://spark.apache.org/>.

<sup>10</sup> <https://storm.incubator.apache.org/>.

Li et al. [38] showed that this methodology yields unbiased results as long as we disregard recommendations which have not been shown to users. We cause the offline evaluation to fine-tune our methods and obtain better strategies for exploration.

## 6.7 Discussion

In this section, we summarize our findings and provide an outlook to future research directions. Suggesting relevant news articles to visitors represents a major challenge to online news portals. In particular, as systems typically have to deal with insufficient information about users preferences. Most users refrain from interacting with plenty news articles but focus their attention on smaller subsets. In addition, a stream of new articles continuously enters the portals' collections. This blurs relations between visitors and articles. Established recommendation algorithms generally assume rather static preferences. Thus, news portals had to come up with novel methods to support visitors as they seek for relevant news. Portals use to implement various recommendation algorithms in order to cover plenty of aspects reflecting different facets of relevancy. Combinations of these algorithms serve visitors with recommended readings. They consider factors including context, popularity, recency, and more. Barriers between academia and research impede further improving the algorithmic performance. Companies avoid publishing data. On the one hand, they may fear privacy issues. On the other hand, they consider their data as asset to their company which they seek to preserve. Conversely, academia generates ideas on how to provide better suggestions. Although, they struggle to evaluate their approaches due to lacking data. Recently, the company plista constructed the "Open Recommendation Platform" (ORP). The platform provides researches access to an actual news recommendation system. Plista expects to improve their recommendation quality. Researchers get the chance to evaluate their ideas with the feedback of actual users. Simultaneously, research faces the technical requirements of a large-scale content provider. A large volume of requests has to be handled at high rates. The system grants as much as 100 ms to send the list of recommended items. Researchers who manage to overcome these restrictions have the unique opportunity to evaluate on a large scale. Millions of users request news article recommendation through ORP. Evaluation concentrates on the click-through-rate (CTR). Other evaluation criteria require graded feedback. For instance, root mean squared error (RMSE; evaluation criteria of the *Netflix Prize*) requires numerically expressed preferences. Users reading news online tend to express their preferences by selection at most.

We identify various directions for future research. We admit that the CTR might not fully capture user preferences. Users may accidentally click on recommendations. Other may immediately abandon the recommended item. Conversely, users may not click on recommendations as they did not perceive them. For instance, recommendations placed on the bottom of the web page require users to scroll down to be seen. Future research may enrich evaluation with additional factors such as dwelling times. Detecting hidden patterns in interactions represents another future

research topic. User profiles are typically sparse as they interact with few items. We may consider recommending not for individual users but for groups of similar users. This idea reflects the notion of certain users sharing similar preferences. For instance, some users may focus on sports-related news. Hence, news recommender systems could recommend articles to the group of these users rather than to each individual. Discovering similarities in highly sparse data represents a major scientific challenge. Finally, we consider early trend detection as a means to further improve recommendation quality. Imagine that a novel item enters the collection of news articles. Systems ought to estimate how likely it will attract a lot of interest. If the system manages to accurately estimate the probability, it will be able to boost interesting items early. Thus, the system will collect a larger amount of clicks than continuing to recommend items which users disregard.

**Acknowledgments** This work was funded by the Federal Ministry of Economic Affairs and Energy (BMWi) under funding reference number KF2392313KM2.

## References

1. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
2. G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in *Proceedings of the 2008 ACM Conference on Recommender Systems—RecSys’08*, p. 335 (2008)
3. G. Adomavicius, J. Zhang, Stability of recommendation algorithms. *ACM Trans. Inf. Syst.* **30**(4), 1–31 (2012)
4. F. Aiolli, Efficient top-n recommendation for very large scale binary rated datasets, in *ACM RecSys*, pp. 273–280 (2013)
5. X. Amatriain, Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explor. Newsl.* **14**(2), 37 (2013)
6. R. Bell, Y. Koren, Lessons from the Netflix prize challenge. *ACM SIGKDD Explor.* **9**(2), 75–79 (2007)
7. J. Bennett, S. Lanning, The Netflix prize, in *KDD Cup*, pp. 3–6 (2007)
8. D. Billsus, M.J. Pazzani, Adaptive news access, in *The Adaptive Web*, Chapter 18, ed. by P. Brusilovsky, A. Kobsa, W. Nejdl (Springer, New York, 2007), pp. 550–570
9. T. Bogers, A. van den Bosch, Comparing and evaluating information retrieval algorithms for news recommendation, in *Proceedings of the 2007 ACM Conference on Recommender Systems—RecSys’07* (ACM Press, New York, 2007) p. 141
10. T. Brodt, F. Hopfgartner, Shedding light on a living lab: the CLEF NEWSREEL open recommendation platform, in *IliX’14: Proceedings of Information Interaction in Context Conference* (ACM, 2014), pp. 223–226
11. M. Burke, A. Hornof, E. Nilsen, N. Gorman, High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten. *ACM Trans. Comput.-Hum. Interact. (TOCHI)* **12**(4), 423–445 (2005)
12. I. Cantador, A. Bellogín, P. Castells, News @ hand: a semantic web approach to recommending news. *Adapt. Hypermed. Adapt. Web-based Syst.* **5149**, 279–283 (2008)
13. I. Cantador, A. Bellogín, P. Castells, Ontology-based personalised and context-aware recommendations of news items, in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (IEEE, 2008), pp. 562–565

14. M. Capelle, F. Hogenboom, A. Hogenboom, F. Frasinca, Semantic news recommendation using WordNet and Bing similarities categories and subject descriptors, in *Symposium on Applied, Computing*, pp. 296–302 (2013)
15. P. Castells, F. Hopfgartner, A. Said, M. Lalmas, Workshop on benchmarking adaptive retrieval and recommender systems: Bars 2013, in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'13* (ACM, New York, 2013) p. 1133
16. P. Cremonesi, Performance of recommender algorithms on top-n recommendation tasks categories and subject descriptors, in *Proceedings of the 2010 ACM Conference on Recommender Systems*, pp. 39–46 (2010)
17. A. Das, M. Datar, A. Garg, S. Rajaram, Google news personalization: scalable online, in *WWW*, pp. 271–280 (2007)
18. M.S. Desarkar, Aggregating preference graphs for collaborative rating prediction, in *Proceedings of the 2010 ACM Conference on Recommender Systems*, pp. 21–28 (2010)
19. M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* **22**(1), 143–177 (2004)
20. G. Dror, N. Koenigstein, Y. Koren, M. Weimer, The Yahoo! music dataset and KDD-Cup'11, in *KDD Cup*, pp. 8–18 (2012)
21. G. De Francisci, A. Gionis, C. Lucchese, From chatter to headlines: harnessing the real-time web for personalized news recommendation, in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pp. 153–162 (2012)
22. Federation of European Publishers. European book publishing statistics (2013)
23. Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in *Computational Learning Theory*, pp. 23–37 (1995)
24. Q. Gao, F. Abel, G.-J. Houben, K. Tao, Interweaving trend and user modeling for personalized news recommendation, in *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* (IEEE, 2011), pp. 100–103
25. F. Garcin, C. Dimitrakakis, B. Faltings, Personalized news recommendation with context trees, in *ACM RecSys*, pp. 105–112 (2013)
26. J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, 1999), pp. 230–237
27. J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. (TOIS)* **22**(1), 5–53 (2004)
28. F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, T. Heintz, Benchmarking news recommendations in a living lab, in *Proceedings of the Fifth International Conference of the CLEF Initiative, CLEF'14* (Springer, New York, 2014), pp. 250–267
29. M. Jahrer, A. Töschler, R. Legenstein, Combining predictions for accurate recommender systems, in *KDD*, pp. 693–701 (2010)
30. A. Karatzoglou, M. Larson, GAPfm: optimal top-n recommendations for graded relevance domains, in *Proceedings of the 22nd ACM Conference on Information and Knowledge Management* (ACM, 2013) pp. 2261–2266
31. B. Kille, T. Brodt, T. Heintz, F. Hopfgartner, A. Lommatzsch, J. Seiler, Overview of CLEF NEWSREEL 2014: news recommendation evaluation labs, in *Proceedings of the Fifth International Conference of the CLEF Initiative, CLEF'14* (Springer, New York, 2014)
32. M. Kompan, M. Bielikova, Content-based news recommendation, in *EC-Web*, pp. 1–12 (2010)
33. N. Lathia, S. Hailes, L. Capra, Temporal collaborative filtering with adaptive neighbourhoods, in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR'09* (ACM Press, New York, 2009), p. 796
34. N. Lathia, S. Hailes, L. Capra, X. Amatriain, Temporal diversity in recommender systems, in *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR'10*, (ACM Press, New York, 2010), p. 210
35. L. Li, T. Li, News recommendation via hypergraph learning: encapsulation of user behavior and news content, in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 305–314 (2013)

36. L. Li, D.-D. Wang, S.-Z. Zhu, T. Li, Personalized news recommendation: a review and an experimental investigation. *J. Comput. Sci. Technol.* **26**(5), 754–766 (2011)
37. L. Li, W. Chu, J. Langford, R.E. Schapire, A contextual-bandit approach to personalized news article recommendation, in *Proceedings of the 19th International Conference on World Wide Web—WWW'10* (ACM Press, New York, 2010), p. 661
38. L. Li, W. Chu, J. Langford, X. Wang, Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms, in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining—WSDM'11* (ACM Press, New York, 2011), p. 297
39. B. Liu, *Web Data Mining* (Springer, New York, 2008)
40. J. Liu, P. Dolan, E. Rønby Pedersen. Personalized news recommendation based on click behavior, in *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 31–40 (2010)
41. N.N. Liu, Q. Yang, Eigenrank: a ranking-oriented approach to collaborative filtering, in *SIGIR*, pp. 83–90 (2008)
42. A. Lommatzsch, Real-time news recommendation using context-aware ensembles, in *Advances in Information Retrieval* (Springer, New York, 2014), pp. 51–62
43. L. Lü, M. Medo, C.-H. Yeung, Y.-C. Zhang, Y.-K. Zhang, T. Zhou, Recommender systems. *Phys. Rep.* **519**, 1–49 (2012)
44. Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, Y. Chang, Learning to model relatedness for news recommendation, in *Proceedings of the 20th International Conference on World Wide Web—WWW'11*, p. 57 (2011)
45. F. Maes, L. Wehenkel, D. Ernst, Learning to play k-armed bandit problems, in *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART 2012)* (2012)
46. A. Montes-García, J.M.Á. Rodríguez, J.E. Labra-Gayo, M. Martínez-Merino, Towards a journalist-based news recommendation system: the Wesomender approach. *Expert Syst. Appl.* **40**(17), 6735–6741 (2013)
47. O. Phelan, K. Mccarthy, M. Bennett, B. Smyth, Terms of a feather: content-based news discovery and recommendation using twitter, in *ECIR*, pp. 448–459 (2011)
48. P. Pu, L. Chen, Trust-inspiring explanation interfaces for recommender systems. *Knowl.-Based Syst.* **20**(6), 542–556 (2007)
49. Y. Ren, G. Li, W. Zhou, Learning user preference patterns for top-n recommendations, in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 137–144 (2012)
50. S. Rendle. Learning recommender systems with adaptive regularization, in *WSDM*, pp. 133–142 (2012)
51. S. Rendle, C. Freudenthaler, Improving pairwise learning for item recommendation from implicit feedback, in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining—WSDM'14*, pp. 273–282 (2014)
52. F. Ricci, L. Rokach, B. Shapira, P.B. Kantor, *Recommender Systems Handbook* (Springer, New York, 2011)
53. R. Salakhutdinov, A. Mnih, G. Hinton, Restricted boltzmann machines for collaborative filtering, in *Proceedings of the 24th International Conference on Machine Learning—ICML'07* (ACM Press, New York, 2007), pp. 791–798
54. M. Seeger, Scalable collaborative bayesian preference learning, in *AISTATS*, vol. 33 (2014)
55. G. Shani, A. Gunawardana, Evaluating recommendation systems, in *Recommender Systems Handbook* (Springer, New York, 2011)
56. Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, A. Hanjalic, CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering, in *RecSys*, pp. 139–146 (2012)
57. J.-W. Son, A.-Y. Kim, S.-B. Park, A location-based news article recommendation with explicit localized semantic analysis, in *SIGIR*, pp. 293–302 (2013)
58. G. Takacs, I. Pilaszy, B. Nemeth, D. Tikk, Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.* **10**, 623–656 (2009)

59. M. Tavakolifard, J.A. Gulla, K.C. Almeroth, F. Hopfgartner, B. Kille, T. Plumbaum, A. Lommatzsch, T. Brodt, A. Bucko, T. Heintz, Workshop and challenge on news recommender systems, in *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 481–482 (2013)
60. S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in *Proceedings of the Fifth ACM Conference on Recommender Systems—RecSys'11*, p. 109 (2011)
61. X. Yang, H. Steck, Y. Guo, Y. Liu, On top-k recommendation using social networks, in *Proceedings of the Sixth ACM Conference on Recommender systems—RecSys'12* (ACM Press, New York, 2012), p. 67
62. C.-N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in *Proceedings of the 14th International Conference on World Wide Web* (ACM, 2005), pp. 22–32

# Chapter 7

## Personalized Information Access Using Semantic Knowledge

Till Plumbaum and Andreas Lommatzsch

**Abstract** Handling the amount of information on the Web, known as the information overload problem, requires tremendous effort. One approach that relieves the user from this burden is offering personalized information access. Systems that adopt to users' preferences are called adaptive systems. Based on a user profile containing details about the users' preferences, the system adapts its content or the user interface to the user. In this chapter, we present a personalized news information system, providing users with entertainment news tailored to their needs. Using semantic technologies, the time to learn user preferences is reduced to a few interactions. We present the system in detail, and present an evaluation showing the benefits coming with the semantic approach.

### Show Me What You Like

Hanging out on the schoolyard during a break, Carl and his friends are reading and discussing some news about their favorite music artists. One of Carl's friends starts a discussion about the new Pop band coming next week to play a concert. While everybody agreed that the music is pretty cool, opinions are deeply divided about the origins of the different band members.

To settle the problem, all friends including Carl pull out their cell phones and start searching the Internet and Wikipedia for more information. Carl uses a new app he just downloaded because of a recommendation from his sister Clara. SERUM, which is the name of the app, is an information system for news using semantic knowledge to provide more information than only showing news and event dates. Carl quickly asks SERUM for information about the band, and after a few clicks through the information graph he can easily provide information about the different band members. His friends are impressed and immediately start asking more questions,

---

T. Plumbaum (✉) · A. Lommatzsch  
Technische Universität Berlin, Berlin, Germany  
e-mail: till.plumbaum@dai-labor.de

A. Lommatzsch  
e-mail: andreas.lommatzsch@dai-labor.de

which Carl can easily answer. One fact all of them notice is the information quality. Carl had just started using SERUM, but the app already provides him with relevant information regarding his information need. Other apps need far more interaction for that.



A few days later, all friends have downloaded the app, SERUM causes the next round of discussions on the schoolyard. By using SERUM, almost everyone now gets news about their favorite band. But SERUM shows more than pure news. It also gives information about related content, such as bands similar to the one currently in the news. As a result, everybody finds some new bands, which are in their personal opinion better than the others. This leads to endless discussions before, during, and after school among Carl's friends. Carl stays out of all these discussions; he is still a fan of the band starting all that fuzz and he cannot wait to go to their concert tomorrow.

## 7.1 Introduction

The flood of available information and products offered by Web applications like online retailers and news portals overwhelms today's users. To handle this information overload, applications typically offer some kind of personalization techniques, in most cases, in the form of personalized filtering or personalized recommendations [3, 36]. However, personalized recommendations that adapt to the users' individual taste are a major challenge [1]. On the one hand, personalized recommendations improve user satisfaction and can motivate users to return. Bad recommendations, on the other hand, may cause users to turn their back on those applications. A common recommendation approach is Collaborative Filtering (CF). CF utilizes historical user information, like ratings or interactions, to compute recommendations [37].



Personalized recommendations also help users to discover interesting information and products based on their preferences and tastes. One challenge is to gather data about the users' preferences. One way is simply asking the user. This is an obtrusive way and may lead to losing users who are not willing to put time and effort in training such an algorithm. Another way is to learn preferences by tracking user interaction.

Finding relevant news on the Internet is becoming increasingly difficult as the number of news published everyday is exploding. A search on Google News<sup>1</sup> for the term "Ukraine" returned 61,500 results retrieved in one day. To master this information overload, several personalized filtering approaches have been proposed. One of the first projects was the 1992 started GroupLens project [16] that recommended Usenet news based on collected ratings from other readers. With our web-based application SERUM (Semantic Recommendations based on large unstructured datasets), we support users in finding interesting and up-to-date news about their favorite topics, currently focusing on entertainment news. Therefore, we utilize a broad range of semantic technologies to further enhance the personalization and recommendation quality. While other work focus on only one aspect of semantic personalization support (e.g., [12, 39]), we build a holistic semantic approach, including frontend and backend solutions, to better learn a user's interest and thus to better recommend news matching these interests. We incorporate semantic information on the client-side, using RDFa<sup>2</sup> in the user interface and a user-tracking component that is able to track this RDFa information [31]. In the backend, we have a semantic knowledge base that includes information from semantic encyclopedic datasets and semantic technologies that model the users' interest using ontologies to link and enrich them with semantic information.

In this chapter, we answer the following question: How can semantic tracking and data management technologies be leveraged for personalization and recommendation services? In order to address this question, we present SERUM (Semantic Recommendations based on large unstructured datasets), a news recommendation system that utilizes semantic technologies to collect implicit user behavior and to build semantic user models. These models, combined with large-scale semantic datasets, are then used to compute personalized news recommendations using graph-based algorithms. We introduce the building blocks of SERUM for semantic data management, personalization, and recommendation, with the main focus on the implicit user behavior collection. SERUM uses RDFa annotations and a RDFa tracker [28] to collect meaningful user behavior and the User Behavior Ontology (UBO) [29], described in Sect. 7.3, to build semantic user behavior models. In the following sections, we first introduce the idea and goal of the SERUM project, followed by an introduction of the SERUM system. Then, we present the use cases that the semantic web usage mining approach covers and showcase an example based on the SERUM system. Finally, we present an evaluation computing on recommendations with a focus on new users that have not interacted much with the system.

---

<sup>1</sup> <http://news.google.com/>, search conducted on September 19th, 2014.

<sup>2</sup> RDFa (or Resource Description Framework—in—attributes) is a W3C Recommendation that allows to embed rich metadata within Web documents.

## 7.2 The SERUM Architecture

The SERUM architecture consists of four building blocks:

- the news crawler,
- the named-entity recognition and disambiguation component (NER/NED),
- the user modeling component, and
- the semantic recommender.

*The news crawler* component, provided by Neofonie GmbH,<sup>3</sup> collects around 60,000 news articles from German and English news sites everyday. *The NER/NED component* [25] identifies and extracts named entities from these news texts and links them to a dataset collected from Freebase.<sup>4</sup> Freebase is a semantic encyclopedic data collection, comparable to DBpedia.<sup>5</sup> The dataset consists of  $\approx 400,000$  artists,  $\approx 1,700,000$  tracks and albums, and  $\approx 2,000$  genres, connected by  $\approx 1,9$  million edges. These data are interlinked with the news corpus through the entities detected in news articles using the NER/NED component. The NER/NED associates a Freebase entry to every entity found in an article by linking a Freebase URL to the entity. The news corpus currently contains over 7,200,000 news articles, growing daily by the newly crawled articles, and builds together with the Freebase data the knowledge base for the recommender. The recommendation algorithm itself is explained in detail in Sect. 7.4 and in [22].

*The user modeling component* implicitly collects the users' reading behavior to build a user model containing the users' interest in topics or entities. Figure 7.1 shows the user interface of SERUM with the personalized news stream. Under each news article, all entities are displayed, which are detected in the article. Each user interaction with an article or an entity is tracked and incorporated in the user model. In the current system, we focus on four user interactions that can be tracked (Fig. 7.2):

- User clicks on an article: The news and all related entities are marked as interesting.
- User clicks on an article in a list: The clicked article and all related entities are marked as interesting for the user, while all other surrounding articles are marked as less interesting.
- User clicks on recognized entities in an article and
- Triggered mouse-over events: Entities clicked by the user or marked by the mouse pointer are given a higher interest rating.

This user feedback is collected using the semantic user behavior tracker described in [26], which is part of the web application. The data are stored on the server-side in an RDF store using the User Behavior Ontology (UBO), described in Fig. 7.7. We build on the idea presented in [35] to use a distinct behavior model but use a more comprehensive model to not only track events but also to track semantic relations between entities on a webpage as presented in [31]. The UBO describes all events

---

<sup>3</sup> <http://www.neofonie.de/>.

<sup>4</sup> <http://freebase.com/>.

<sup>5</sup> <http://dbpedia.org/>.

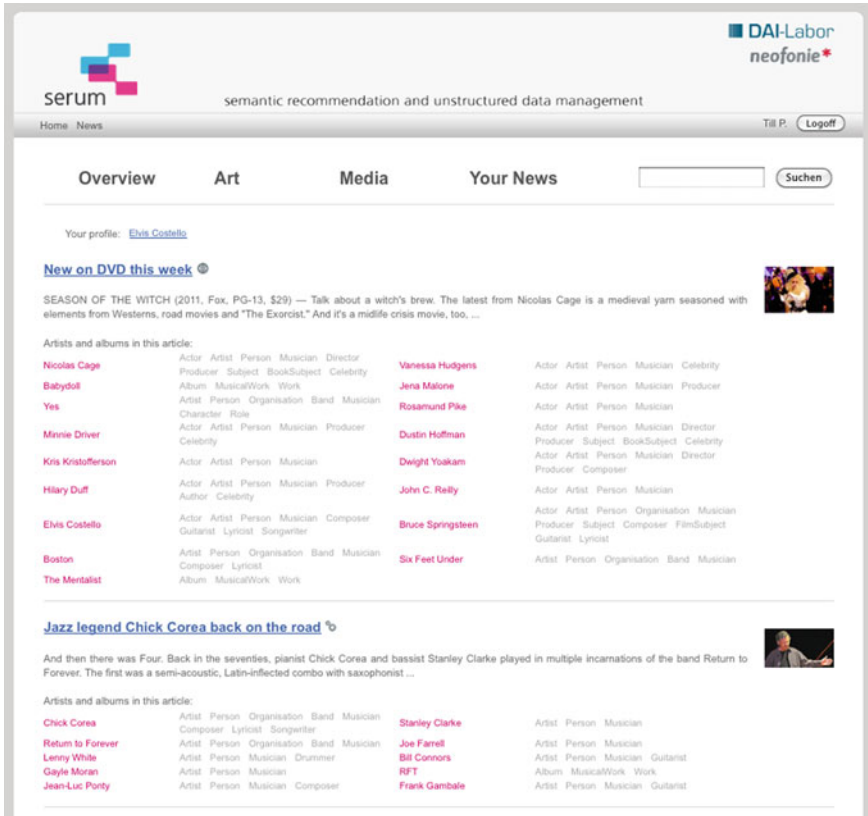


Fig. 7.1 SERUM interface showing recommended news articles and recognized entities

relevant for modeling the user behavior such as user clicks or mouse-over events. Events, triggered by the user (e.g., clicks), are linked to news articles and named entities (e.g., artists in the news article) the user interacted with.

Based on a statistical interaction analysis the user behavior events are aggregated to identify named entities (e.g., topics, musicians, and genres) the user is interested in. The analysis includes the last *n* sessions of the user (in our current system *n* is set to five) where the interaction of a user is analyzed and the entities are ranked according to the interaction frequency. The analysis also includes a time aspect where an interaction has a higher weight if the session is a current one. Furthermore, we deploy semantic data (from Freebase) to extend the knowledge about identified named entities to produce a richer user model. Thus, musicians recognized to be interesting to the user are expanded with data about produced albums and collaborating artists. For example, if the user only stated interest in “Madonna,” we can add genre information (e.g., pop) and information about collaborations with other artists. These enriched user profiles are used as the input for our graph-based recommender. The more

information in the user profile, the more likely it is to find related news for a user. The news recommendation strategy is based on the recentness of the news as well as the correlation of computed interests and their occurrence in the news. Based on the defined architecture and introduced trackable user interactions, we demonstrate the SERUM system in the next section.

### 7.2.1 SERUM Use Case—User Behavior Collection

In order to explain the interaction of the semantic tracking component with the news recommendation system, we walk through the first and fourth trackable user interaction outlined in the previous section, and detail the resulting user model and the recommendations. As mentioned in Sect. 7.2, SERUM is a personalized news recommender where the user profile is created by tracking and analyzing user behavior. Initially, after the first login, the user profile and the personalized news stream is empty as depicted in Fig. 7.3. The picture shows the empty user profile on the left and the empty personalized news stream. To create the user profile, the user has to interact with SERUM, to read news or to search for artists.

When the user starts reading, their first interaction is with a list of news articles where they can choose what to read. The SERUM news list shows the article, an

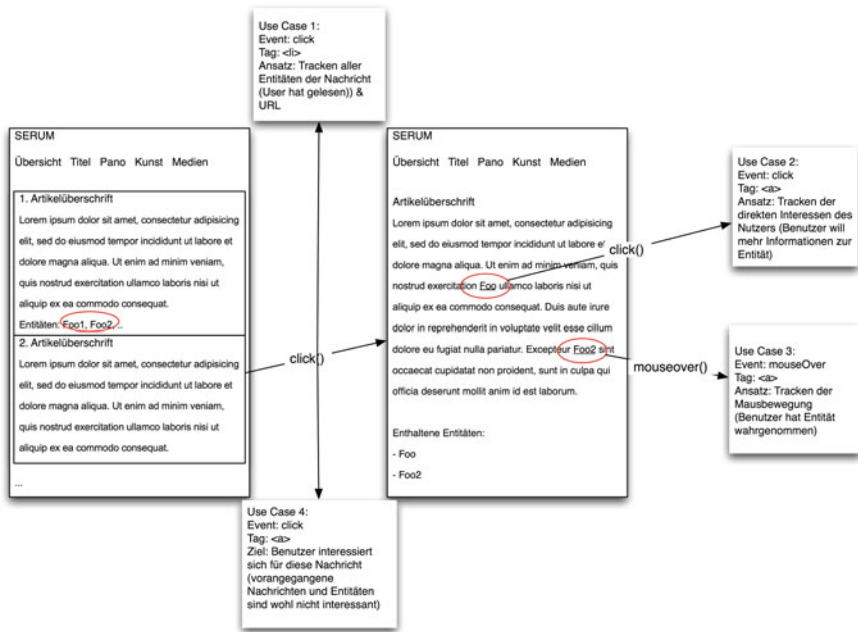


Fig. 7.2 Visualization of the SERUM user tracking use cases

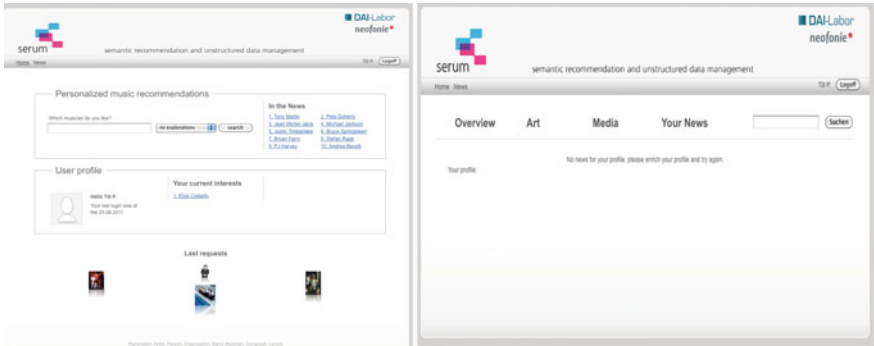


Fig. 7.3 SERUM personalization news: After the first login, no news is recommended (right side) because no profile exists (left side)

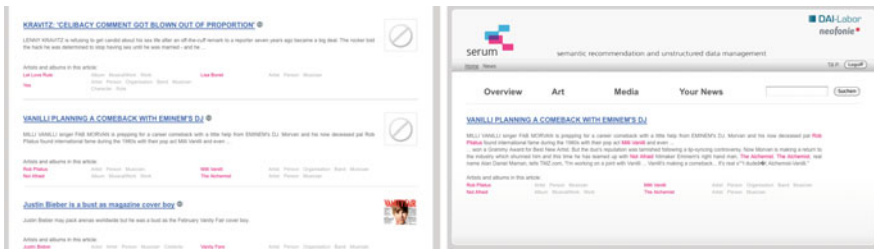


Fig. 7.4 SERUM news list and article with the artists that are part of the article

abstract, and a list of entities (artists) that are found within the text. Figure 7.4 shows the news list overview on the left-hand side and a detailed view of the selected article on the right-hand side.

When the user clicks on an article, that article, the position of the article, and the surrounding articles are tracked and sent back to the server. This tracked information allows to start building a user profile as the read article, and the connected artists, are getting a positive weight. The articles, and connected entities, surrounding the read article getting a negative weight, as they were in the user's viewport but were not as interesting as the read article.

Apart from the tracked article information, information about the user and the used device is also tracked and sent back to the server to assign the data to one user. While the users are reading the article, SERUM also tracks the mouse movement and if users hover over an artist. This is also sent back to the system as it may indicate that this artist is of special interest [10]. A direct click on an artist, which leads to an extra info site about the artist, is also tracked and treated with much higher weight for the user profile creation.

This information, tracked by our tracking system builds the base for the creation of a user interest profile. The used profile creation mechanism follows the presented use cases, e.g., clicks on an article mark all artists as interesting for a user while

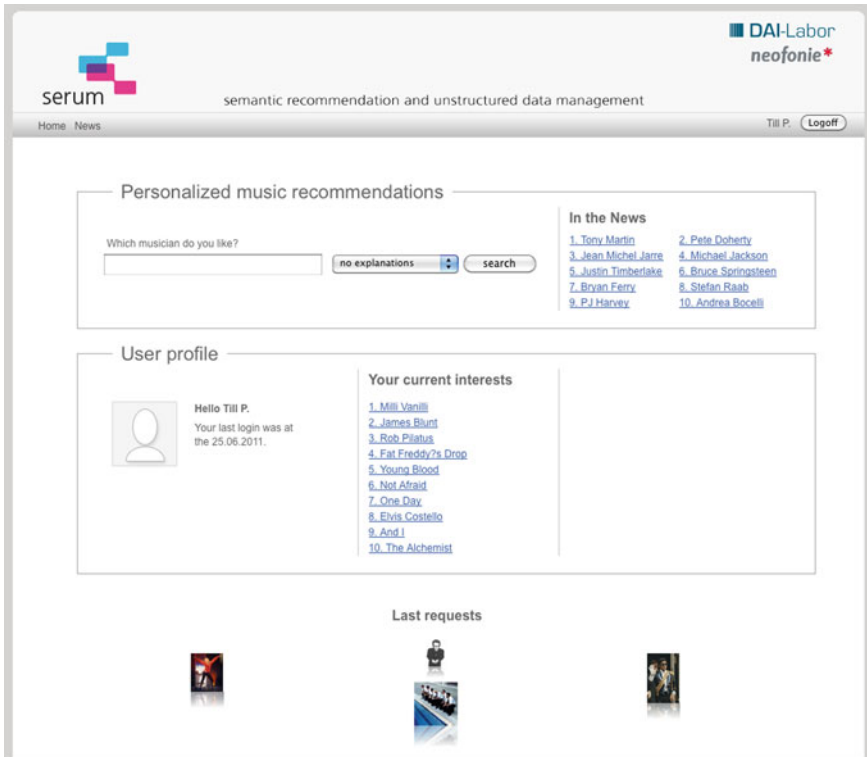


Fig. 7.5 SERUM: The user profile after reading some articles

artists from article surrounding the clicked one are marked as less interesting. The resulting user profile is shown in Fig. 7.5.

The created profile is used to create the personalized news stream, shown in Fig. 7.6. The news is based on the user profile, which is a weighted profile and the in Sect. 7.4 presented graph-based algorithm to enrich user profiles.

### 7.3 The User Behavior Model

After introducing our semantic tracking approach in the previous section and concluding that for a fully semantic tracking approach also a semantic backend is needed, in this section we introduce a new ontology-based model for the collection and management of user behavior data, the User Behavior Ontology (UBO) [27]. UBO serves two main goals:

1. Defining a common data model, an ontology, to manage user behavior information as described in the previous section: With UBO, data about user behavior can be

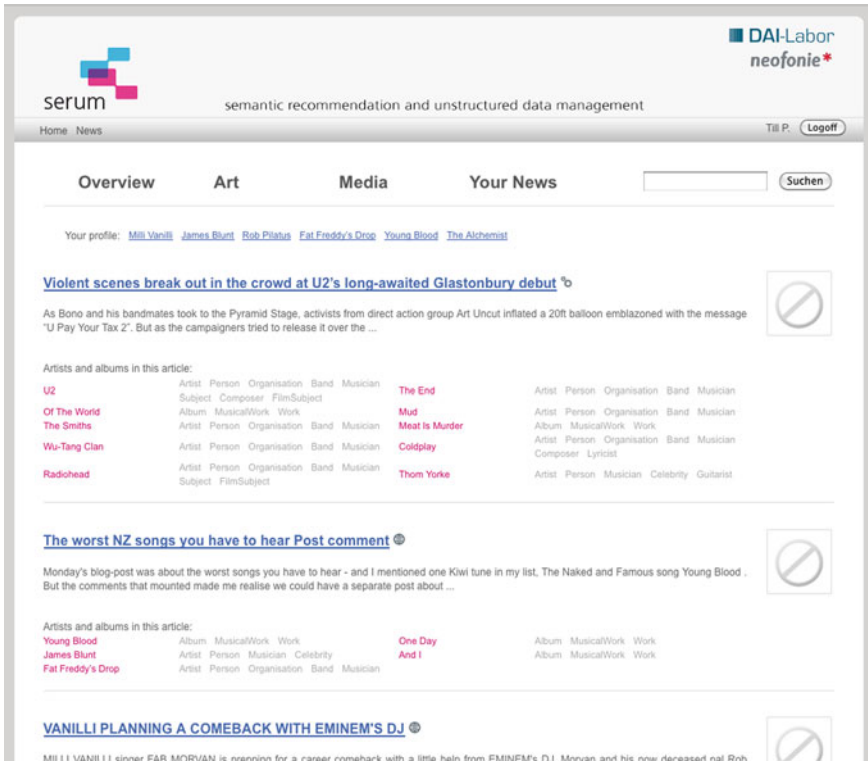


Fig. 7.6 SERUM: Personalized news

collected using a common data model and thus can be shared and reused across systems. UBO defines a common schema for the semantic collection of user behavior, where the raw interaction, as well as semantic information about the user intention, context, etc., can be stored. Previous work mostly focuses on domain-specific modeling [33]. The data management is application independent, which means that when sharing UBO data, other applications can use the data to run their own data analysis approaches and use this for personalizing recommendations or the adaptation of the User Interface (UI) [41].

2. Linking user behavior data with external knowledge following the Linked Open Data process: Due to the creation of an ontology as a common data model, UBO should also allow to link behavior with external resources. This means that collected behavior can be connected to other ontologies, adding extra knowledge, for example, about a user’s intention behind a click, or information processed by an applied machine learning approach. This, for instance, allows to model information about what an application assumes the user is interested in, which is valuable input for other applications when data are shared.

UBO has the clear goal to serve as a general model for interaction with an application, a semantic form of the server log files. It is not intended to be part of a general model for all possible types of behavior. The field theory of Lewin [20] proposes that human behavior is the function of both the person and the environment. With UBO, the focus is set on the environment, what type of application the user is interacting with, what elements are visible to the user, etc. The user itself, their current emotions, and needs are not part of UBO. This must be incorporated by other models.

As outlined in the previous section, the main goal of user behavior collection, the web usage mining process, is to get detailed information about how users interact with an application to understand what people want in order to offer better personalization or recommendation. This has to be part of the UBO, too. The challenge is to build a model that allows to manage explicit information such as a click event, and to manage the implicit information that is also tracked with our tracking system.

### 7.3.1 Conception of UBO

UBO orients itself on standard log file formats. As stated above, its purpose is to provide a semantic model for user behavior that can be extended with additional meta-information. Existing work on general user behavior ontologies is scarce. The work of Schmidt et al. [35] proposes a set of different models to capture all relevant data for website personalization. The used models cover the website structure (Web Portal Ontology), website content (Content Ontology), user profiles (User Ontology), and website usage data as well as knowledge about the adaptation process itself (Adaptation Ontology). The most important ontology is the adaptation ontology, which is used to decide if an adaptation should take place and how to do that. The adaptation decisions are based on predefined rules. The ontology most related to the UBO is the Behavior Ontology [35]. This ontology captures atomic events, such as mouse related or keyboard events, and when an event started and ended. UBO centers around the *Element* a user is interacting with. With UBO, the interaction with that of a website element is tracked, how the user interacted with the element, and also what other elements were visible and semantically connected. UBO allows collecting more information than the pure Behavior Ontology presented in [35]. The combination of the Web Portal and the Behavior Ontology allows at least connecting an event to the page structure, but still the possibility to track underlying semantic connections on a webpage is not given. It is also not explained how the Web Portal Ontology copes with partial reloads of the website. This change in the website structure is trackable with our semantic tracking solution and can be captured using UBO. Ngoc et al. [24] present an approach for generalized ontologies for user preferences, the Spatio-Temporal Ontology of User Preference (STOUP), and behavior routine, the Spatio-Temporal Ontology of User Routine (STOUR). STOUR covers part of the intended UBO functionality as it allows to model reoccurring activities in a *Routine Element* connected with time and system information. This is a higher aggregation of the UBO *Event Element* but already processed to meta-knowledge. The goal of



UBO is to be able to model and track atomic events and allow the processing of such meta-knowledge using atomic facts.

### 7.3.1.1 Model Description of UBO

UBO is a collection of different linked entities that give a complete picture of the user behavior during a session and longer periods of time. It covers the users' actual behavior as well as implicit knowledge. A complete overview of UBO is given in Fig. 7.7. UBO is divided into different parts covering all aspects of the behavior life-cycle, application-dependent aspects, user aspects, and interaction aspects. In the remainder of this section, we describe the most important entities, their functions, properties, and their intended usage.

*Application Aspect:* The application aspects cover all information about the application required that the user is interacting with. What type of application *i* is, what different views (e.g., different webpages) belong to it, and what is modeled/displayed on the page.

*Application:* The OWL class *Application* defines the name and an ID for the application that is used to identify the application. It allows links to the *ubo:Domain* to determine the scope of the application and to the different *ubo:Views* the application has. An *Application* can consist of multiple views but must define at least one. An application can cover several domains, e.g., a news website. In such case, the different *ubo:Views* should define a specific domain, e.g., sport.

*View:* The OWL class *View* defines a single view (e.g., webpage) of a *ubo:Application*. It can define a *ubo:Domain* (which can be different from the general application domains) and link to different *ubo:Elements*. A *View* can contain several

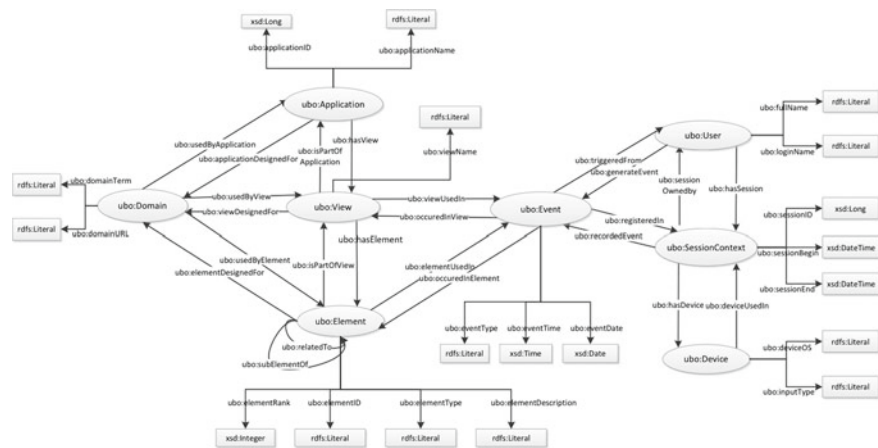


Fig. 7.7 The User Behavior Ontology with all classes and relations

*ubo:Elements*. Those *ubo:Element* objects can describe an entity, e.g., an artist, that is covered in an article or define a link to a different *View*.

*Element*: The OWL class *Element* marks parts of the website as relevant. An *Element* can be an artist on a news page or a link to another *ubo:View* or external page. *Elements* can also refer to each other in one *ubo:View* to define that *Elements* are related. With the *ubo:elementRank* property, rankings can be defined, e.g., the rank of the element in a search result list. This is helpful when computing an interest model as *Elements* above the element the user interacted with my not be interesting.

*Domain*: The *Domain* defines the topic of a *ubo:View* or *ubo:Application*. It allows us to define a name for it, which can be in the form of a textual description. More important is the property *ubo:domainURL* which defines the domain by giving it a unique URI, which is a commonly agreed description for the topic. It is recommended to use URLs from large encyclopedic resources such as Wikipedia or its semantic equivalent DBpedia. This approach, which follows the Linked Data Principles (see [6, 7]), allows other applications to understand what the application or view is about.

*User Aspects*: User aspects include all information about the user, the device that is used to access an application and session information. This information allows to identify a user and to distinguish between different devices that they may use. This can help to identify contexts, e.g., mobile or at home, and give better recommendations based on the context. The session information helps to narrow down the context, as it allows the unambiguous differentiation when the user did what. This allows us to create context-related information, such as at work, during lunch, etc.

*User*: The *User* entity in UBO allows to identify a user. As mentioned in Sect. 7.3, UBO is not focusing on the user itself, but has the goal to collect data about the user interaction and the context, the environment, of the interaction to have sophisticated data that allows for inferring interests and intention of a user. Therefore, the *User* entity only allows to set a login name, which can be a user name or ID, and to link it to a *Session*.

*Device*: The *Device* entity describes all relevant properties of a device, mobile, PC, etc., that helps to later distinguish between different devices of a user. That could be a notebook and PC which both run the same OS but with different screen resolutions, or a mobile device. This could be used to adapt UI elements or to determine a context, office, home, or on the road.

*Session Context*: The OWL class *SessionContext* describes a time frame when a user interacted with an application or multiple applications without a longer pause in between. It defines a start and end time and sets the used devices. A *SessionContext* belongs always to one *ubo:User*.

*Interaction Aspects*: The *interaction aspects* cover all entities that help to manage the actual behavior. Information about what the user did on a webpage (e.g., reading an article, clicking on a link or hovering over a picture, etc.) is important for later personalization and recommendation purposes. While the application aspects give

us insights into how the application is structured and thus allows us to draw implicit conclusions from the way the user interacted with it, the interaction gives us explicit feedback. The click on a recommended item indicates that it matches the users' interests; to what extent depends on the further interaction. If the user, for instance, buys an item on a website, this action is a strong indicator for a positive perception, while a quick return to the recommendation lists indicates that the recommended item probably did not match the users' interests.

*Event*: The OWL class *Event* describes the type of event (click, mouse over, etc.) and the *Element* or *View* the user interacted with. An event always occurs on an *Element* or *View*. With the type of the *Event*, also the time when the event happened is tracked. This allows to later identify chains of actions and create higher order events. For example, a click event on an element, followed by a mouse move, followed by a click release event on a different element could be a Drag-and-Drop event where an item is dropped into a basket.

## 7.4 The Enrichment Approach

In this section we present a semantic recommendation approach using the previously described semantic technologies. We explain the approach in detail and conduct a comprehensive evaluation to examine how the enrichment influences recommendation quality. Results show that our approach improves recommendation results, especially for users with uncommon interests.

The general idea of our enrichment approach is visualized in Fig. 7.8 with an example of a music recommendation system: The figure shows three user profiles consisting of only a few items without any overlap with the other profiles. In this case, collaborative filtering (CF) cannot be used. Our profile enrichment process adds several new items (strongly related to the already present items), so that later, the user profiles have an overlap and CF can be applied. If a user profile (middle row) initially contains user interests about 'Björk' and 'Moby,' our enrichment algorithm takes both entities as input and starts to traverse the semantic dataset which is a graph where all information is connected. The first entity that is added to the user profile is the genre entity "electronic," as both artists are directly connected to it. Then, the algorithm adds additional artists like "Morcheeba" as the band is also connected to "electronic." This enriched user profile is then used for CF.

In this section, we focus on music data to illustrate and evaluate our approach. The approach itself presented in this section is designed to work on any kind of data as long as it is presented as a graph. Figure 7.9 shows the general data structure needed for our approach. The dataset needs a user node that is connected with a like/rated relation to a set of entities, which can be connected by any kind of relation. The rate/like relation indicates a positive relation to the linked entity. Negative relations are currently not considered. The entity nodes can be music information, as in our scenario, or books, movies, etc.

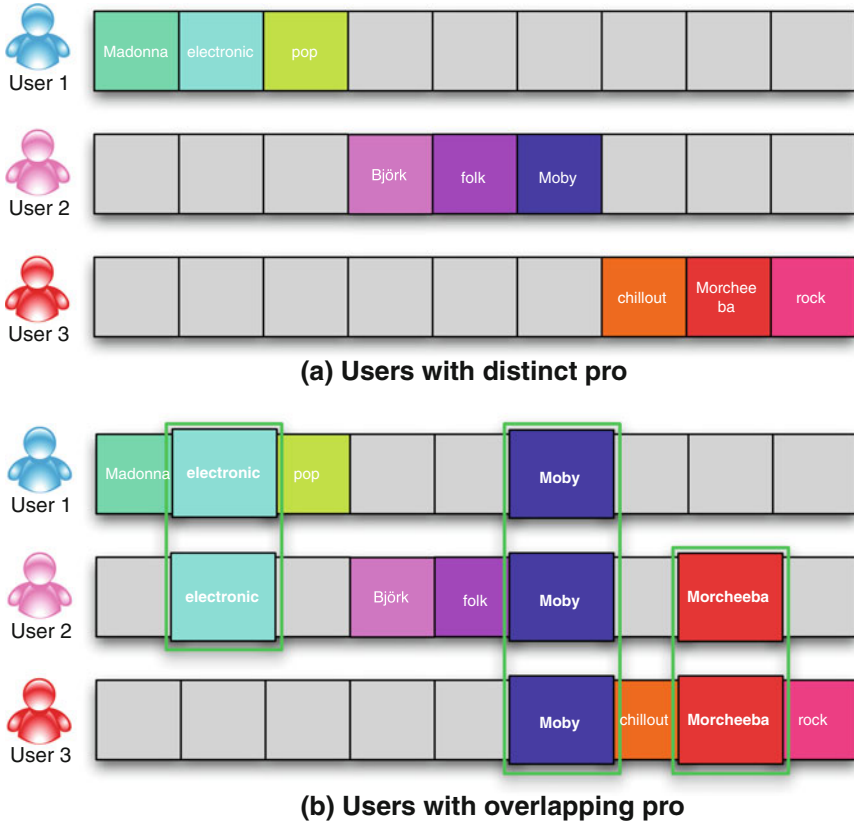


Fig. 7.8 Simplified visualization of the initial cold start problem. **a** Before the enrichment, there is no overlap between the different user profiles and collaborative filtering is not possible. **b** After enrichment, the user profiles overlap and collaborative filtering is possible

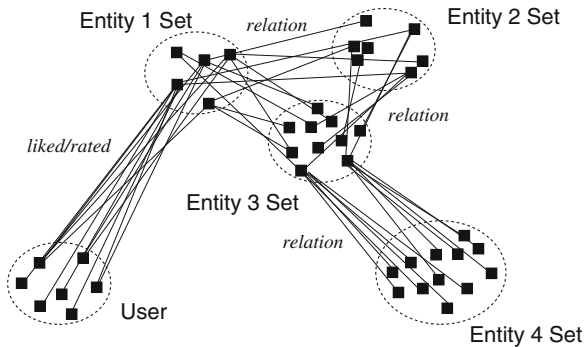


Fig. 7.9 The semantic dataset with generic information and user profiles linked to it

### 7.4.1 Enriching User Profiles Based on Semantic Data

Our motivation is to cope with the cold start problem. Therefore, we use semantic encyclopedic knowledge to extend small user profiles. Studies on Wikipedia,<sup>6</sup> as an example for online encyclopedias, showed that the quality and the accuracy of Wikipedia articles is on a high standard and hence a reliable information source [15]. Therefore, we follow the idea that semantic encyclopedic data is a good and “neutral” source for enriching user profiles with knowledge not influenced by subjective opinions or tastes. Enriching user profiles with items strongly related to the items already present in the user profile, adds “synonyms” for the existing entities. A synonym in this context means that we add interests to the user profile that are similar to already expressed user tastes, e.g., adding an additional artist that is related to an artist in the user profile. This is done to increase the overlap of the enriched user profile with other profiles. Thus, it improves the similarity calculation, but does not change the taste of the user.

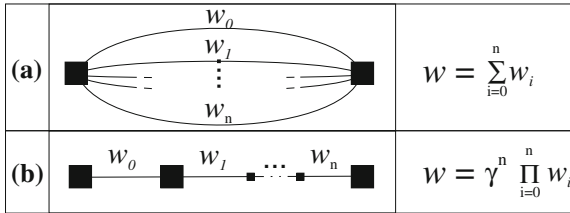
#### 7.4.1.1 Finding Related Items Based on Encyclopedic Data

Our approach for solving the complex problem of computing entities to enrich the user profile uses link prediction methods on a semantic dataset to find important related items to a given input set of items (e.g., a user profile). The link prediction task describes the problem of inferring missing links in an observed graph that are likely to exist [32, 38]. In our approach, we apply link prediction for the task of finding edges between items in the semantic dataset and a set of given entities of a user profile.

To compute related entities for a given set of input items, we determine the entities best connected to the input entities already present in a user profile. In our scenario, *best connected* from a set of input entities describes the items that can be reached by several parallel paths each consisting of a small number of edges. The computation of the related entities can be performed directly on the semantic dataset (“memory-based”) or based on a simplified network model (“model-based”). The semantic dataset is modeled as a network consisting of nodes representing the entities and edges describing the relationship between the entities (see Table 7.1). For computing entities closely related to a given user profile, we take all existing entries in the user profile as a starting point and traverse the semantic network (“path based breadth-first search”). Since an extensive search may require too many resources (CPU, RAM), we introduce a parameter to control the search depth of our approach. In this work, we use a maximum search depth of four, meaning that starting from the user profile all nodes are considered that can be reached with four steps or less. All entities that can be reached from entities in the user profile are weighted by the number of parallel paths and by the number of edges for each path. The formulas for calculating the path weights are shown in Fig. 7.10. An entity is the more relevant the more parallel

---

<sup>6</sup> <http://www.wikipedia.org/>.



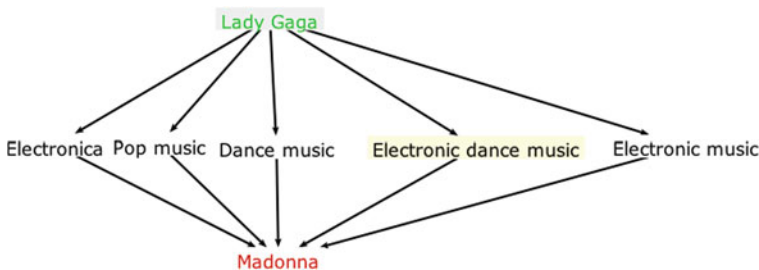
**Fig. 7.10** The figure shows the formulas for calculating the path weights for **a** parallel edges and **b** for a sequence of edges. The discount factor  $\gamma$  ensures that short paths get higher weighting than long paths

paths from the user profile exist and the shorter (based on the number of edges) the paths are. Also, the type of edge is taken into account. We evaluate for different path lengths how the profile enhancement influences the CF performance.

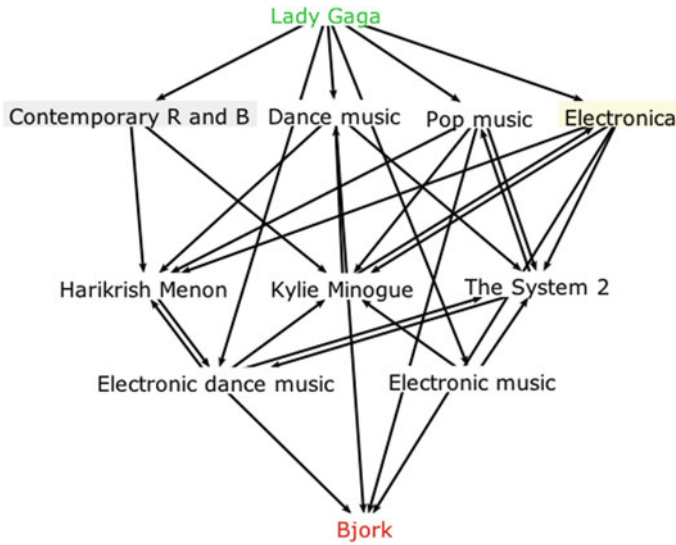
To give an impression of how the system computes related entities, Figs. 7.11 and 7.12 show example computations using only artist and genre nodes and edges connecting the nodes. Figure 7.11 shows a possible enrichment based on a user profile containing “Lady Gaga” as an interest. The path length is set to two; this means that only entities that are not more than two steps away are taken into account. In this example, “Madonna” would be used to enrich the user profile. Figure 7.12 shows the enrichment going to a depth of four. This means that entities that are not more than four steps away are taken into account. Input is the same user profile, with “Lady Gaga” as an interest.

### 7.4.1.2 Memory-Based Link Prediction

We apply a path-based approach for computing predictions. Starting from several input entities (e.g., the entities in the user profile), we traverse the semantic network. The entities reachable from the input entities are ordered according to a semantic similarity rating. This rating is calculated based on the edge weights of the respective



**Fig. 7.11** Path Length 2: Explanation of path-based enrichments over the Artist-Genre edge set. The user can see the different nodes that were used for enrichment with Madonna



**Fig. 7.12** Path Length 4: Explanation of path-based enrichments over the Artist-Genre edge set. The user can see the different nodes that were used for enrichment with Björk

path. Currently, the weight of the edge, which can be considered as the importance of the edge, has to be set manually or by using normalization strategies. One strategy is to weigh edges based on their significance to connect a node in the dataset. If the edge is the only one connecting a node, determined by the degree of a node, it is considered as more important than edges that connect a node with several other edges. For parallel edges/paths the ratings are summed up. For a sequence of edges the weights are multiplied and weighted by a discount factor (depending on the path length). In our system, we implemented the path-based approach using a breadth-first search algorithm with a limited search depth [34]. The search depth limit is set to make sure that the computed results are relevant for the input items and not only loosely connected. With the depth limit, no items are taken into account where the path length to the most relevant item is longer than the defined search limit.

Another advantage of path-based approach is that no additional effort is needed for building a model. Thus, updates in the dataset immediately affect the computed results.

### 7.4.1.3 Model-Based Predictions

Real-world datasets are often sparse and noisy. In order to cope with these problems we reduce the complexity of the dataset by aggregating similar entities into clusters. To assure that users still understand computed recommendations, we use Hierarchical Agglomerative Clustering [42] that combines entities with similar features in one

cluster. The computed clusters are treated as nodes. Thus, path-based search strategies can be used for searching relevant entities.

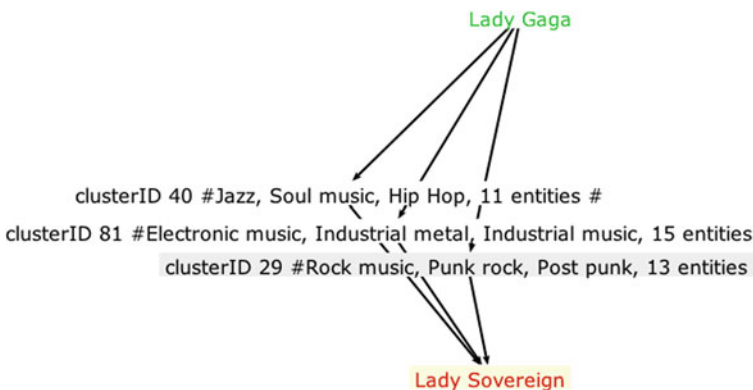
The advantages of model-based recommenders are that the complexity of the dataset can be effectively reduced to speed-up the computation of relevant entities. Furthermore, the reduction of noise in large datasets often improves the result quality. The algorithm applied for reducing the graph complexity highly depends on the domain. We decided to focus on Hierarchical Agglomerative Clustering since it enabled us to choose similarity measures and clustering parameters optimized for each relationship set. Moreover, for recommendations computed based on clustered entity sets, path-based explanation can be provided. The disadvantages of model-based recommenders are that additional effort is needed for calculating and updating the model. A prediction based on clustering is presented in Fig. 7.13. As the results of the cluster algorithm are most of the time only loosely related to the input node, the results from the clustering are not considered in the evaluation.

## 7.5 Evaluation

The goal of evaluation is to research the impact of an enriched user profile on the cold start problem for CF. We therefore consider two evaluation scenarios:

### New user and new application

The first scenario covers the cold start problem for a new music recommendation application with few users. In this scenario, we want to analyze the effect of the enriched user profiles for a new music recommendation application that has a small number of users and how recommendation quality is affected for new users.



**Fig. 7.13** Cluster-based prediction: Explanation of cluster-based enrichments using automatically generated genre cluster



## New user and large application

The second scenario is focused on a new user that joins a well-established recommendation service, such as LastFM or Facebook. We want to see how the enrichment approach works for new users in a big recommendation application which already has a lot of users.

### 7.5.1 Datasets

Evaluation is performed using two datasets from Facebook and the LastFM collected between January and September 2010. We extracted data from around 60,000 users and kept the profiles that contain data about interests in music. For evaluation we used all user profiles containing at least two music interests. Users from the Facebook dataset expressed their interests by ‘liking’ an artist. Users in the LastFM dataset showed their interests by listening to music, which is implicitly tracked information from LastFM, and by actively ‘favoring’ artists. The resulting Facebook dataset consists of 3,011 users and 14,516 liked music items. The LastFM set consists of 7,743 users and 11,333 favored music items. We only crawled user profile information, no other data from Facebook, e.g., Facebook Open Graph<sup>7</sup> information, or data from LastFM about similar artists is part of the user profile data. The user profiles only contain the user name, the artist name, or music album name, and in the LastFM set also the MusicBrainz ID.<sup>8</sup>

The semantic information that is needed for our approach is retrieved from Freebase. In our scenario, we make use of data from the music domain consisting of four music entity types, namely *Artists*, *Albums*, *Tracks*, and *Genres* relations between them. The relationship between artist and genres describes the genre in which an artist works; the relationship between album and artists describes which artist can be found on an album release, and finally the relationship between album and genre defines a genre assignment for each album. The created dataset is schematically visualized in Fig. 7.14. Table 7.1 shows the number of edges and entities contained in the dataset.

In order to analyze how semantic encyclopedic data can improve CF, we interlinked the semantic dataset retrieved from Freebase with LastFM and Facebook as explained in Sect. 7.5.2.

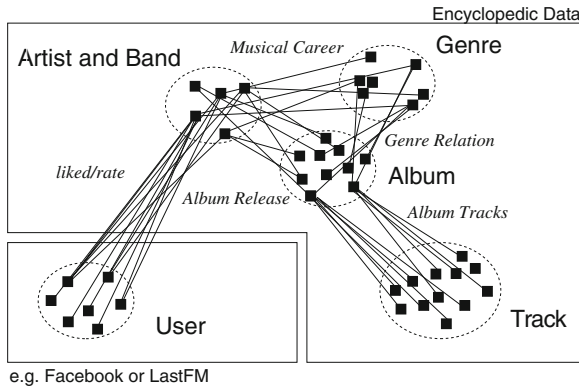
### 7.5.2 Interlinking User Profiles

The extracted Facebook and LastFM profiles are initially isolated, meaning that there is no connection to the Freebase dataset. However, our approach requires

---

<sup>7</sup> <http://developers.facebook.com/docs/opengraph/>.

<sup>8</sup> <http://musicbrainz.org/>.



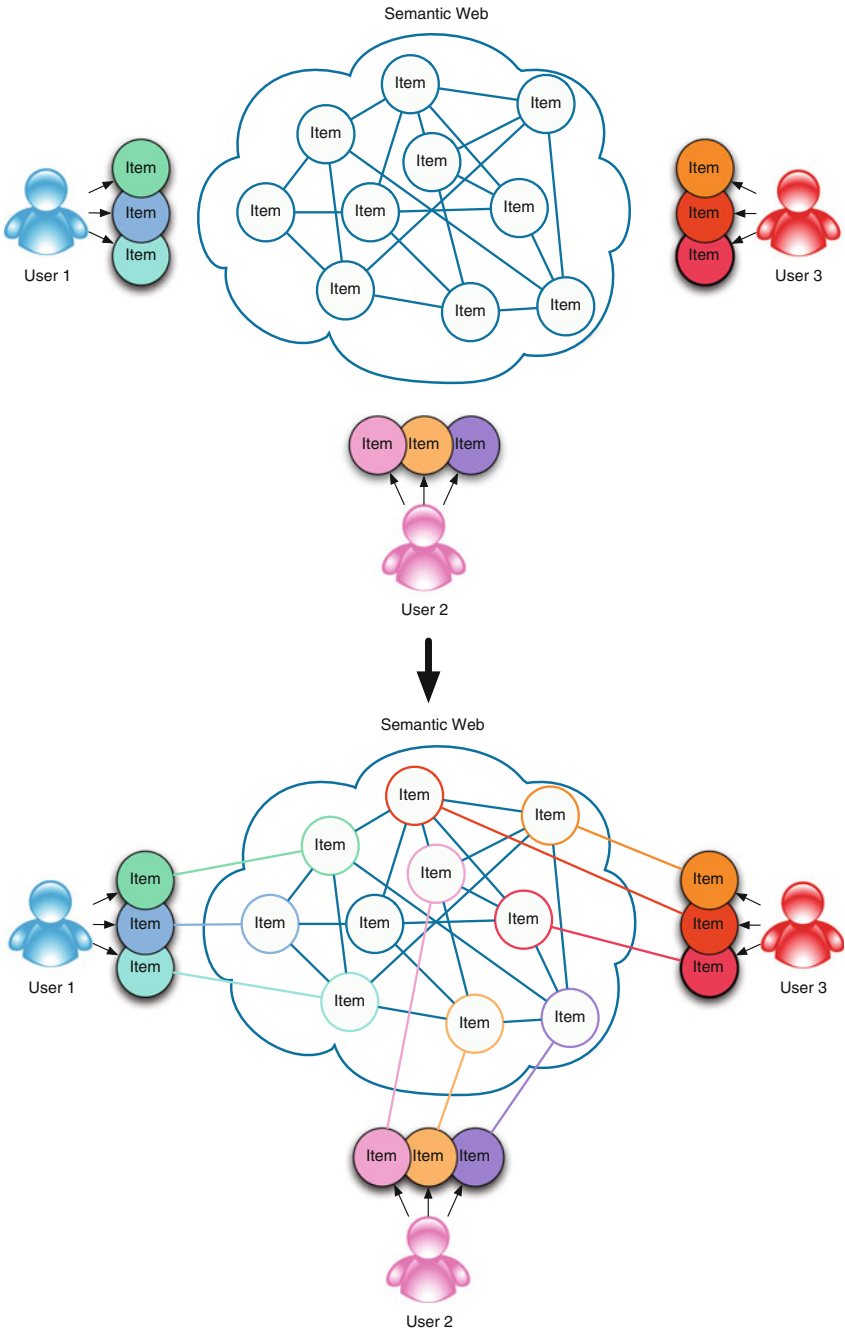
**Fig. 7.14** The semantic dataset with music information and user profiles linked to it

**Table 7.1** Music information contained in the Freebase dataset

Entities	Number of entities	Number of edges			
		Musicians	Genre	Albums	Tracks
Musicians	417,217	–	79,543	374,445	–
Genre	3,082	79,543	–	90,444	–
Albums	438,180	37,445	90,444	–	1,048,565
Tracks	1,048,576	–	–	1,048,565	–

a graph containing the user profiles and the Freebase data interlinked. The linkage is needed as our enrichment algorithm is a graph-based method. Without connected data, the profile enrichment cannot be computed. Thus, it is necessary to know that an entity such as ‘Facebook#The\_Beatles’ in a user profile is similar to the entity ‘Freebase#Beatles’ in the Freebase dataset and to create a link between them. Figure 7.15 shows the situation before and after the linkage. Linkage is done using a set of rules that connect the profiles. First, we check if we have a MusicBrainz ID (which is the case if we got the user data from LastFM). If we have the MusicBrainz ID the linkage is easy as this information is also part of the meta-information that Freebase provides about the artists. If no MusicBrainz ID is available we try to link entities based on the artist name in different spellings and languages offered by Freebase. If more than one Freebase node matches the rules and we cannot disambiguate the correct node this entity is disregarded. While we assume that this method minimizes the number of false positive linked entities, there still may be incorrectly linked entities that might lead to a reduced recommendation quality.

Having connected the user profile with the Freebase dataset, the derived semantic network can be used for enriching user profiles.

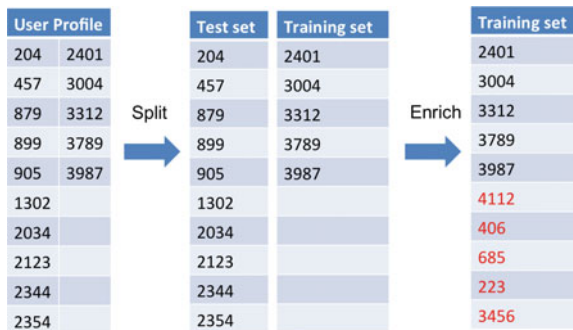


**Fig. 7.15** To compute the enriched profiles we first need to find edges between the user profiles and the semantic dataset

### 7.5.3 Selection of ‘New Users’ and Evaluation Algorithms

The selection of users who represent the new users joining a recommendation service was done by creating a subset with all users who have rated exactly 15 items. All test users must have the same number of items in the profile, to be able to compare the results of the different evaluation runs in the different scenarios. The number 15 was chosen because it gives a large enough user profile for evaluation (train and test split) and also enough users with 15 items to have statistically enough test data. From these 15 items we use a set of 10 items as our test set and for training we arbitrarily choose 1–5 of a user’s remaining items for the initial user profile (training set) to simulate the cold start problem. The process is visualized in Fig. 7.16. We conduct several test runs, starting with a user profile containing only one item, and then iteratively increase the number of items up to five. The training set is enriched with an additional five–nine items, depending on the initial size of the training set, so that it always contains ten items. Results are averaged over 200 evaluation runs for each user profile size (one–five items) using the following forms of CF algorithms:

- **CF with standard profiles:** The Baseline. A standard CF algorithm using the Tanimoto coefficient [21] to compute user similarity.
- **CF with enriched profiles:** The standard CF algorithm using the enriched user profiles instead of the standard profiles.
- **Most Popular Recommender:** A simple algorithm recommending the top  $n$  items of the dataset.
- **CF + enriched profiles:** A combined method of the first two CF methods. If the standard CF does not find a recommendation, CF with the enriched profiles is used. This approach avoids the recommendation depending mostly on the items used to enrich the profile.
- **CF + Most Popular recommendations:** An approach using most popular recommendations if the standard CF find no results.
- **Random Recommender:** Recommending randomly chosen items.

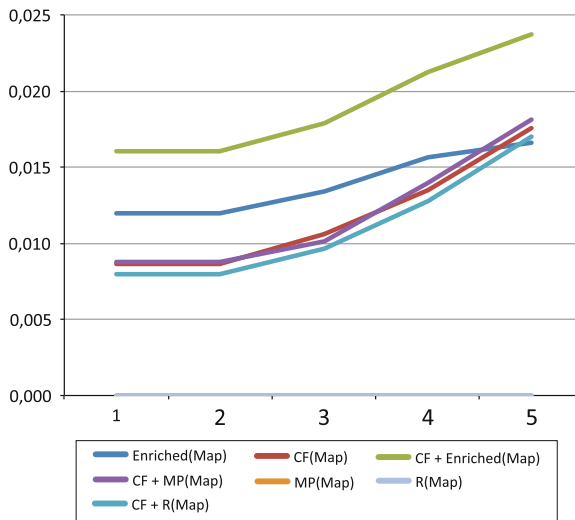


**Fig. 7.16** Example of a user profile with 15 items. First, a training test split is done with ten test items and five items in the training set. Then we enrich the user profile with five additional items. This enriched set is then used for CF

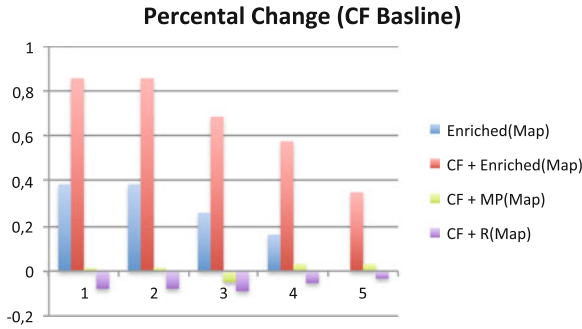
### 7.5.4 Evaluation of the ‘New User and New Application’ Scenario

The application is created by randomly selecting 5,000 users from our crawled data. These users represent users who already use the application. The test users, which are different from the 5,000 users, are also chosen from the user dataset containing only users with an interest in jazz or swing music. This was done to augment the cold start problem as most users in our dataset have a “Pop” taste. The initial user profiles are enriched using data from Freebase. Figure 7.17 presents the results for the different algorithms described in Sect. 7.5.3. The results show that the enrichment has a huge positive impact on the recommendation quality. Both approaches using the enriched profiles (*CF* using only enriched profiles and *CF* with *standard profiles* combined with *CF* with *enriched profiles*) clearly outperform the *standard CF* and *CF + Most Popular* for user profiles of size 1–4. For users with a user profile size of five, the *CF* with *enriched profiles* is slightly inferior than the *standard CF*. *Most Popular* and *Random* recommendation have no impact at all. Using only the *Most Popular* recommendations does not work as the selected test users were only interested in swing and jazz music as the common taste in the randomly selected dataset is on pop music. Thus, the list of *Most Popular* recommendations consists of pop artist and does not contain any swing or jazz artists.

Figure 7.18 shows the change in recommendation quality on a percentage basis compared to the *CF* with *standard profiles*. The usage of our enrichment approach



**Fig. 7.17** Cluster-based prediction: Explanation of cluster-based enrichments using automatically generated genre cluster



**Fig. 7.18** Cluster-based prediction: Explanation of cluster-based enrichments using automatically generated genre cluster

improves the recommendation quality by over 90 % for very small profiles (sizes 1 and 2) and over 40 % for the bigger profiles (size 4 and 5).

### 7.5.5 Evaluation of Facebook and LastFM

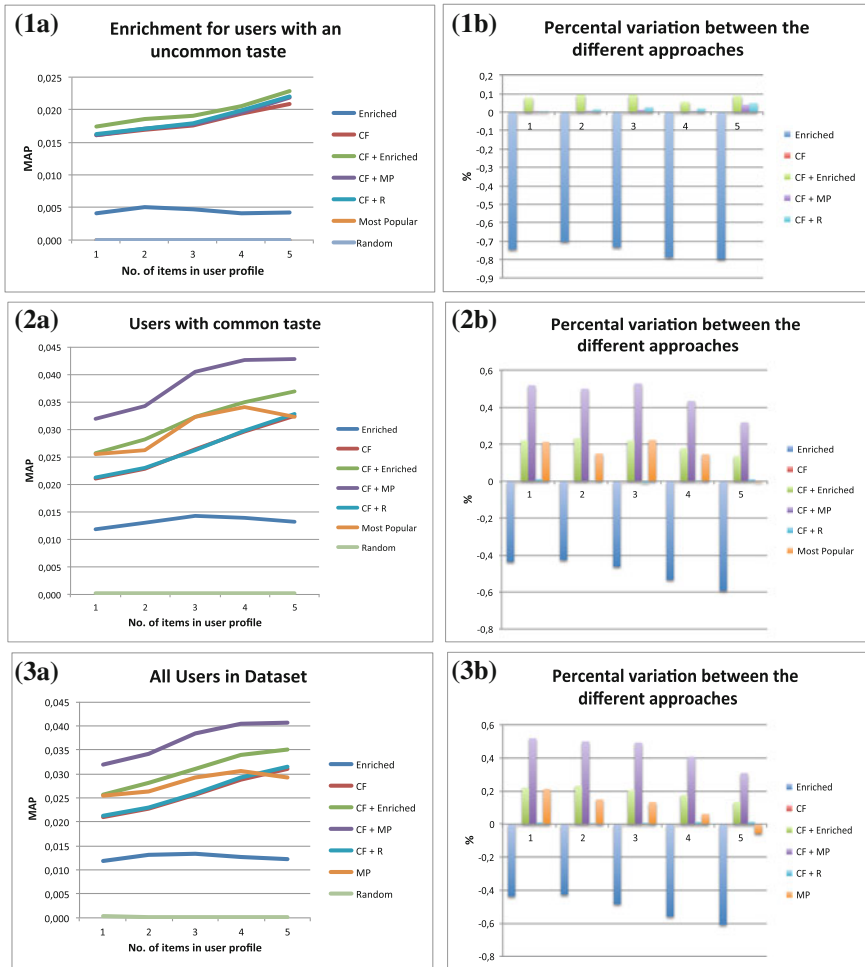
The evaluation for the second scenario, the ‘New user and large application’ scenario, is done separately for Facebook and LastFM to compare if there are differences in a Social Network and a distinguished music recommendation service like LastFM. The evaluation covers three different user subsets:

1. Recommendations based on the complete dataset.
2. Recommendations for users who have an uncommon taste. This is similar to the swing and jazz user sets used in evaluation in Sect. 7.5.4.
3. Recommendations for users who mostly like popular artists.

The split between users with an unusual taste and users with a common popular taste is done based on the average deviation of popular artists in a user’s profile. The popularity of an artist is computed based on the distribution in the Facebook and LastFM datasets. The initial user profiles (with one–five items) are enriched with five additional items from Freebase, so that the user profiles given to the collaborative filtering recommender have a size of six–ten items.

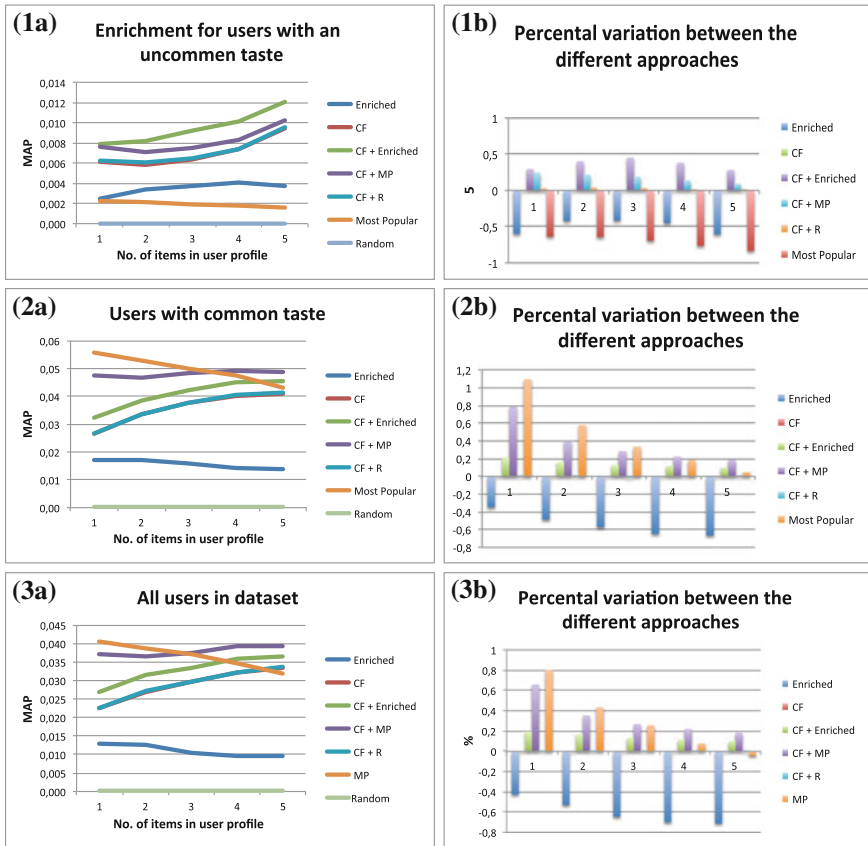
Figures 7.19 and 7.20 show the evaluation results on both datasets. Results on the Facebook dataset show that *CF with enriched user profiles* does not improve the recommendation quality compared to the *standard CF*. The enrichment even leads to a reduced precision.

This is expected as our enrichment approach adds mostly ‘popular’ entities from Freebase to the user profile, meaning that the enrichment can blur the user profile and make the user profile less personal. As explained, the enrichment algorithm takes the degree of a node in the Freebase graph into account. Thus, mostly popular



**Fig. 7.19** Results of the evaluation using the Facebook (*first column*) and LastFM (*second column*) datasets. **a** Shows the results for users with an unusual music taste, **b** shows the results for users with popular music taste, **c** shows results over the complete dataset

artist and genres are used for enrichment. This is a shortcoming of the encyclopedic Freebase dataset as there are no other indicators than distance to the user profile and degree of a node that could be used. Also, the *standard CF* benefits from the fact that it is more likely to find similar users in a common taste scenario, hence recommended items base on the original, not blurred, user profile and the CF can make use of more neighbors (similar users). On the other hand, a more detailed look on the results reveals that a combination of the standard and enriched CF can improve the quality. The reason is that in cases where *standard CF* cannot find appropriate items because no similar users can be found, enrichment helps to find



**Fig. 7.20** Results for Evaluation on the Facebook (*first column*) and LastFM (*second column*) datasets. **a** Shows the results for users with an unusual music taste, **b** shows the results for users with popular music taste, **c** shows results over complete dataset

other users based on the enriched profile and hence items to recommend. This effect becomes even more visible in scenarios where recommendations for users with an uncommon taste are computed. In these scenarios the strategy *CF + enriched profiles* outperforms all other approaches. As CF depends on a sufficient amount of neighbors to compute recommendations and finding similar users for users with an uncommon taste is more difficult, enrichment helps to overcome this problem. The results for the LastFM user profiles (Fig. 7.19) confirm the findings on the Facebook dataset. On the LastFM dataset, we used a more restrict threshold to distinguish between users with common and uncommon taste. The evaluation results show that for users with a uncommon taste *CF + Most Popular* recommendations perform bad while the *CF + enriched profiles* recommender really improves recommendation quality. For common taste users and all users, the *CF + Most Popular* recommender performs best. Both combined strategies outperform the *standard CF* recommender.



## 7.5.6 Related Work

The Social and Semantic Web has attracted a large number of researchers from different research fields to find solutions to the cold start problem. So far, different approaches have been proposed. Approaches range from manipulating the CF process or manipulating the user model before the CF calculation. In the following section we present selected works on state-of-the-art CF systems that cope with the cold-start-problem and present the recent work on user profile enrichment.

### 7.5.6.1 Collaborative Filtering

In [2] the authors present an approach that uses existing ontologies, e.g., a movie ontology, and integrate derived item information with existing user ratings. While standard CF algorithms assume that all items are distinct, the authors propose an extended CF algorithm that considers item information as well based on the item similarity, e.g., the same director. Item similarity is computed by taking into account similarity between item attributes. To compute the attribute similarities, for each attribute a similarity function must be defined and an aggregation function that combines the different attribute similarities. This way, it is possible to find similar users even if they did not rate the same, but similar movies. The approach has the disadvantage that it needs effort to build a similarity function for each attribute and it is also limited to one domain. With our approach we overcome both limitations of this work. Different weights for different relations/attributes can be learned automatically based on the number of occurrences in the graph, for example, and the domain limitation is dropped because of our semantic approach where it is easily possible to bridge different domains.

In a different approach, Middleton et al. [23] build ontological profiles for users to recommend research articles. The user profile creation is done using a topic hierarchy. To overcome the cold-start problem, the authors also attempt to use externally available information based on personnel records and user publications. The limitation is that the existence of such additional knowledge cannot be generally assumed. In some cases, like the presented research community example, public information is available, but especially on the social web, this information is locked in the different social networks. Thus, instead of requiring personal information from external sources, our approach leverages public knowledge sources like Freebase (or DBpedia).

### 7.5.6.2 User Profile Enrichment

Different strategies have been proposed to expand the knowledge about users ranging from the aggregation of user information distributed over different applications to solutions adding semantic and linguistic knowledge to user profiles [14, 17, 26].

Aggregation of personal information from several applications [30, 40] and using it for recommendations has been demonstrated in experimental setups [4]. However, this approach is not easily adoptable as most applications keep their data in ‘walled gardens’ where the application provider does not allow to get any user information out of the system, e.g., no API is offered. Thus, it is not easy to get data for one user from different applications [4, 5]. In addition, privacy and security issues may occur and users may not be willing to share passwords to allow the aggregation of data from different accounts. Other works add meta-knowledge from sources like WordNet to user profiles to describe similar items, e.g., items from the same domain [13, 19]. Of course, with the aggregation of user information from different applications a user could help to build a holistic view of the user, but as the data are hard to get, we have chosen a more applicable way by using free encyclopedic data as the source for profile enrichment.

## 7.6 Discussion and Conclusion

This study shows that with a combination of the semantic tracking system and the UBO, creation of user interests’ profiles becomes simple and effective. With no visible intervention on the website, detailed tracking of user actions is possible. This is the main requirement of our tracking system. Of course, beneath the surface the website structure has to be extended with semantic information using micro-formats or RDFa. But, relying on the semantic tracking solution, with only a few read articles, the user profile already reflects general interests of the user and allows us to offer a personalized news stream filtering the huge amount of articles. While the presented scenario in Sect. 7.2.1 only showed the tracking of mouse events, the SERUM system also tracks searches for artists and uses this information for profile creation. As a search is an explicit action, the artists the user searches for received higher weightage in the user profile. This complex tracking is unobtrusive and transparent for the user, which was another requirement of our tracking solution. The management of tracked information using the UBO allows the usage of this data for future personalization in different applications. If a user registers for a new application, his previously collected behavior data can be used to adapt the UI to personal preferences or to compute recommendations.

We also presented a new semantic recommendation approach using enriched user profiles with data retrieved from semantic encyclopedic datasets. Our evaluation shows that depending on the scenario the profile enrichment improves the recommendation quality. Especially in scenarios where the given user profile is very small and the interests of the user differ from the mean taste of the other users (see Sect. 7.5.4). However, evaluation also showed some shortcomings of the presented approach. Enrichment works very well for users with an unusual taste and in scenarios where the number of users of an application is low; in these scenarios the enriched profiles heighten the recommendation quality. By contrast, enrichment is not helpful for users with large profiles or a popular music taste. In these cases enrichment blurs the

user profile and the therein-specified user taste, because the Freebase data contains general domain information, and for users with a common taste more or less universal knowledge is added. Adding the artist “Madonna” does not make sense for user profiles already containing a lot of pop artists; it only leads to more general profiles less tailored to individual user preferences. Different strategies to overcome this problem are conceivable. On the one hand, our approach needs to weight the edge types in a more user-centric way. A user may like an artist because of a certain song but does not like the complete discography; or a user might like the artist because of the social engagement of that artist and not because of the music. Therefore, more contextual information about users is needed, enabling a context-sensitive weighting of the information used for the profile enrichment. The increasing popularity of Social Semantic Web approaches and standards like FOAF<sup>9</sup> can be one important step in this direction [8, 9]. On the other hand, semantic datasets themselves have to be enriched with more meta-information about the data. General quality and significance information like prominence nodes and weighted relations can improve semantic algorithms to better compute the importance of paths between nodes. An artist that made hundreds of bad albums may have a high number of links to, e.g., a genre node, but is not an important artist for this genre while another artist made only one or two albums but defined a genre. In this case, a significant weightage for the artists can improve the quality and performance of semantic algorithms.

Future steps are the evaluation of a focused enrichment, e.g., only using artist or genres information, based on the context of the user. Another direction is to implement an advanced weighting model (e.g., based on prominence, context, user groups or interaction time (e.g., [11])) as an overlay for the Freebase dataset, and to implement alternative network models (e.g., based on a low-rank approximation for the adjacency matrix of a relationship set [18]).

**Acknowledgments** This work was funded by the Federal Ministry of Economic Affairs and Energy (BMWi) under funding reference number KF2392305KM0.

## References

1. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Trans. Knowl. Data Eng.* **17**, 734–749 (2005)
2. S.S. Anand, P. Kearney, M. Shapcott, Generating semantically enriched user profiles for webpersonalization. *ACM Trans. Internet Technol.* **7**, 26 (2007)
3. S.S. Anand, B. Mobasher, Introduction to intelligent techniques for web personalization. *ACM Trans. Internet Technol.* **7**, 4 (2007)
4. S. Berkovsky, T. Kuflik, F. Ricci, Mediation of user models for enhanced personalization in recommender systems. *User Model. User-Adapt. Interact.* **18**(3), 245–286 (2008)
5. S. Berkovsky, T. Kuflik, F. Ricci, Cross-representation mediation of user models. *User Model. User-Adapt. Interact.* **19**(1–2), 35–63 (2009)
6. C. Bizer, T. Heath, T. Berners-Lee, Linked data—the story so far. *Int. J. Semant. Web Inf. Syst.* **5**(3), 1–22 (2009)

---

<sup>9</sup> <http://www.foaf-project.org/>.

7. C. Bizer, T. Heath, K. Idehen, T. Berners-Lee, Linked data on the web (IDOW2008), in *Proceedings of the 17th International Conference on World Wide Web, WWW'08* (ACM, New York, 2008), pp. 1265–1266
8. U. Bojars, A. Passant, J.G. Breslin, S. Decker, Data portability with Sioc and Foaf, in *XTech* (2008), <http://www.slideshare.net/CaptSolo/data-portability-with-sioc-and-foaf>
9. U. Bojars, A. Passant, J.G. Breslin, S. Decker, Social network and data portability using semantic web technologies, in *BIS 2008 Workshops Proceedings: Social Aspects of the Web (SAW 2008), Advances in Accessing Deep Web (ADW 2008), E-Learning for Business Needs, CEUR Workshop Proceedings, CEUR-WS.org*, vol. 333, pp. 5–19 (2008)
10. M.C. Chen, J.R. Anderson, M.H. Sohn, What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing, in *CHI'01 Extended Abstracts on Human Factors in Computing Systems, CHI EA'01* (ACM, New York, 2001), pp. 281–282
11. F. Hopfgartner, D. Hannah, N. Gildea, J.M. Jose, Capturing multiple interests in news video retrieval by incorporating the ostensive model, in *Second International Workshop on Personalized Access, Profile Management, and Context Awareness in Databases, PersDB'08, Auckland, New Zealand, (VLDB Endowment 08)*, pp. 48–55 (2008)
12. F. Hopfgartner, J.M. Jose, Semantic user modelling for personal news video retrieval, in *16th International Conference on Multimedia Modeling, MMM'10, Chongqing, China*, vol. 1 (Springer, New York, 2010) pp. 336–349
13. F. Hopfgartner, J.M. Jose, Semantic user profiling techniques for personalised multimedia recommendation. *ACM/Springer Multimed. Syst.* **16**(4), 255–274 (2010)
14. F. Hopfgartner, J.M. Jose, An experimental evaluation of ontology-based user profiles. *Multimed. Tools Appl.* 1–23 (2012)
15. I. Huvila, Where does the information come from? Information source use patterns in Wikipedia. *Inf. Res.* **15**(3) (2010)
16. J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, H. Volante, GroupLens: applying collaborative filtering to usenet news. *Commun. ACM* **40**, 77–87 (1997)
17. T. Kuflik, Semantically-enhanced user models mediation: research agenda, in *5th International Workshop on Ubiquitous User Modeling (UbiqUM 2008)* (2008)
18. J. Kunegis, A. Lommatzsch, Learning spectral graph transformations for link prediction, in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09* (ACM, New York, 2009) pp. 1–8
19. E. Leonardi, F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, A flexible rule-based method for interlinking, integrating, and enriching user data, in *Proceedings of the 10th International Conference on Web Engineering*. Lecture Notes in Computer Science, vol. 6189 ed. by B. Benatallah, F. Casati, G. Kappel, G. Rossi (Springer, Vienna, 2010), pp. 322–337
20. K. Lewin, *Principles of Topological Psychology* (Mcgraw-Hill Book Company Inc., New York, 1936)
21. A.H. Lipkus, A proof of the triangle inequality for the Tanimoto distance. *J. Math. Chem.* **26**(1–3), 263–265 (1999)
22. A. Lommatzsch, T. Plumbaum, S. Albayrak, An architecture for smart semantic recommender applications, in *11th International Conference on Innovative Internet Community Systems*. LNI, vol. P-186 (LNI, Berlin, 2011) pp. 105–114
23. S.E. Middleton, N.R. Shadbolt, D.C. De Roure, Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.* **22**, 54–88 (2004)
24. K. Ngoc, Y.-K. Lee, S.-Y. Lee, Owl-based user preference and behavior routine ontology for ubiquitous system, in *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*. Lecture Notes in Computer Science, vol. 3761, ed. by R. Meersman, Z. Tari (Springer, Berlin, 2005), pp. 1615–1622
25. D. Ploch, L. Hennig, A. Duka, E.W. De Luca, S. Albayrak, Gerned: a German corpus for named entity disambiguation, in *proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, ed. by N. Calzolari (Conference Chair), K. Choukri, T. Declerck, M.U. Dogan, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis (European Language Resources Association (ELRA), Istanbul, Turkey 2012)

26. T. Plumbaum, Semantically-enhanced ubiquitous user modeling, in *UMAP*. Lecture Notes in Computer Science, vol. 6075 ed. by P. De Bra, A. Kobsa, D.N. Chin (Springer, New York, 2010)
27. T. Plumbaum, User behavior ontology. (2011) <http://ubo-ontology.org/>
28. T. Plumbaum, A. Lommatzsch, E.W. De Luca, S. Albayrak, Serum: collecting semantic user behavior for improved news recommendations, in *UMAP 2011, Poster and Demo Session; Girona, Spain* (2011)
29. T. Plumbaum, A. Lommatzsch, E.W. Luca, S. Albayrak, Serum: collecting semantic user behavior for improved news recommendations, in *Advances in User Modeling*. Lecture Notes in Computer Science, vol. 7138 ed. by L. Ardissono, T. Kuflik (Springer, Berlin, 2012), pp. 402–405
30. T. Plumbaum, K. Schulz, M. Kurze, S. Albayrak, My personal user interface: a semantic user-centric approach to manage and share user information, in *HCI International* (2011)
31. T. Plumbaum, T. Stelter, A. Korth. Semantic web usage mining: using semantics to understand user intentions, in *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization, UMAP '09* (Springer, Berlin, 2009) pp. 391–396
32. A. Popescul, L.H. Ungar. Statistical relational learning for link prediction, in *Proceedings of the Workshop on Learning Statistical Models from Relational Data* (2003)
33. L. Razmerita, An ontology-based framework for modeling user behavior 2014; a case study in knowledge management. *IEEE Trans. Syst., Man Cybern. Part A: Syst. Hum.* **41**(4), 772–783 (2011)
34. S.J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edn. (Pearson Education, Upper Saddle River, 2003)
35. K.-U. Schmidt, L. Stojanovic, N. Stojanovic, S. Thomas, On enriching Ajax with semantics: the web personalization use case, in *4th European Semantic Web Conference* (2007)
36. Richard A. Spreng, Robert D. Mackoy, An empirical examination of a model of perceived service quality and satisfaction. *J. Retail.* **72**(2), 201–214 (1996)
37. X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**, 4:2–4:2 (2009)
38. B. Taskar, M.-F. Wong, P. Abbeel, D. Koller Link prediction in relational data, in *Proceedings of Neural Information Processing Systems* (2004)
39. I. Torre, Adaptive systems in the era of the semantic and social web, a survey. *User Model. User-Adapt. Interact.* **19**, 433–486 (2009)
40. K. van der Sluijs, G.-J. Houben, A generic component for exchanging user models between web-based systems. *Int. J. Contin. Educ. Lifelong Learn.* **16**(1/2), 64–76 (2006)
41. D.S. Weld, C. Anderson, P. Domingos, O. Etzioni, K. Gajos, T. Lau, S. Wolfman, Automatically personalizing user interfaces, in *IJCAI*, vol. 3, pp. 1613–1619 (2003)
42. Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets. in *Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM'02*, (ACM, New York, 2002), pp. 515–524

# Chapter 8

## Personalized Fashion Advice

Till Plumbaum and Benjamin Kille

**Abstract** Shopping online for clothes is becoming very popular recently. But finding good clothes remains a difficult task. We face a wealth of clothes on offer, and without the possibility to fit or feel the product, making decisions is not easy. In this chapter, we present a use case of an online retailer that aims to improve the shopping experience of men. Differing from conventional online shops where the customers browse through various products and eventually add items to their shopping basket, this shopping service relies on the expertise of fashion advisers who, after getting in contact with the costumers, arrange a combination of different clothes and ship them to the customers. We present a case-based recommendation approach using the available user information entered explicitly, such as price constraints and preferred colors, and also learn a user model based on purchase histories. We evaluate and compare our case-based approach with standard recommendation approaches. The evaluation shows that even with little knowledge, a suitable user model can be learned and used for computing recommendations. The evaluation bases on real data of customers of an online shop. Based on the results, using a case-based recommendation approach could help to solve cold-start problems. But for computing good recommendations for all users, more information about explicit user preferences is needed, which is currently not available.

### Steven Marks Goes Shopping

It was a cold and stormy night in November. Steven was sitting at the big kitchen table, a refurbished heirloom from his great grandmother's house, drinking a cup of black, fair trade coffee. Finally some alone-time for him. Suzanne and the kids were visiting Suzanne's family at the coast, not returning till end of next week. Steven was on his own for the next few days. "So Steven, what will you do today?", he said to

---

T. Plumbaum (✉) · B. Kille  
Technische Universität Berlin, Berlin, Germany  
e-mail: till.plumbaum@dai-labor.de

B. Kille  
e-mail: benjamin.kille@dai-labor.de

himself, wondering what to do during this rare moment of spare time. Slightly bored, he reached for the local newspaper. Are there any events in town this week? “Hey, that’s interesting,” he thought when he stumbled upon an article about the furniture store downtown. Their senior partner is retiring and they celebrate this with an Open Day. Might be interesting to visit their store and check for some bargains. He kept reading...



“Oh my god!” Steven almost jumped to his feet. Retirement? He completely forgot about his boss’ retirement party which was supposed to be at the end of this week! Being his successor, Steven was supposed to give a short laudatio at the event. He wrote the laudation quite a while ago, also giving a speech in front of an audience is no problem for him...but he forgot to get a formal suit for that night. “This is usually Suzanne’s job, she is the one with a sense for style and good taste,” he thought. His selection always tends to be a bit parrot-fashion. Steven feverishly tried to think about a way out of this situation. “What are my options now?”. Calling Suzanne was out of the question. She had buzzed him so many times that he should join her to go shopping for formal suits since he had “outgrown” his clothes a bit. And each time, he had come up with another explanation on why he cannot come this time. “No, she is out of this!” He definitely was not keen to experience yet another “I told you so” moment. Who was left? His male friends’ sense of style was not much better than his, rather worse. Asking the neighbor was probably not a good idea either after the “accident” last year, even so she definitely had an unarguably good sense of style.

Almost automatically, he opened up his laptop and started his web browser. “Let’s hope the Internet can help me,” he thought. After a bit of searching, he found the website of an online fashion store that also specialized on online style consulting. “Ah, brilliant, exactly what I need now.” Steven started playing the video on their website where they explained their service. It seemed quite straight forward. He first had to fill in a questionnaire about his personal lifestyle, what kind of goods he usually buys, and so on. Then, one of their fashion advisors would call him to inquire further details about the occasion where he wants to wear the outfit and to discuss

his general fashion style with him. Once they have all information that they need, they will prepare a package for him and send it to his address to try it on.

“So far so good,” Steven thought. “It’s worth giving them a try. After all, their fashion advisors are experts and they should know which clothes match.” He typed in his details in the online form and waited for the call to discuss his fashion needs further. He did not have to wait for long until he received a call from his personal advisor who introduced himself as Kasimir. Steven described his precarious situation, and in-between, Kasimir asked some questions to find out his attitude towards specific fashion styles. Finally, they agreed on two trousers, one beige and one gray, both with a noticeable crease. A microchecked brown shirt, and a striped blue cotton shirt. Moreover, a dark and a beige jacket which, according to Kasimir, should complement the shirts perfectly. To complete the package, Kasimir also recommended him two different ties and a pair of brown shoes.

Before hanging up, Kasimir promised that the package will be delivered within the next 48 h. “If that’s true, you have a new best friend,” Steven joked before hanging up the phone himself. Steven was satisfied that this sudden problem has already been sorted. “So Steven, what will you do with the rest of the day then?”, he wondered.

## 8.1 Introduction

What should I wear? Most of us know this question, and there is no easy answer to it. What we wear reflects who we are, what we think, and what our taste is. It also has a social impact as clothes are a big factor how we are perceived by others [3, 16, 28, 46]. When we shop, all of the aforementioned aspects have to be considered. In the good old days, when we shopped offline, the shopping experience was accompanied by a relative, a friend and/or a shopping assistant to get some social feedback and the number of available articles was limited by the store size. With the ever growing amount of online shopping possibilities, people tend to shop more and more online.<sup>1</sup> With the growth of people shopping online and an increasing amount of purchasable items, we face two challenges:

- With the increasing amount of purchasable items, and variants of those items, the customer’s ability to survey all items is just not sufficient enough. This situation is usually referred to as an *Information Overload* [25, 33, 42]. Especially shopping fashion (i.e., clothes and accessories) online is a challenge as fashion has a lot of attributes like color, cut, style, material, size, fabric, thickness, texture, brand, and many others. Not being able to directly see and touch the products, assessing the products based on these features is a major drawback when shopping for clothes online [50]. Besides, studies have shown that online shoppers often do not exactly know what they want [37], increasing the challenge of online shopping even further.

---

<sup>1</sup> The Nielsen report from 2012 shows that online shopping has become the most preferred shopping method <http://www.nielsen.com/us/en/newswire/2012/shopper-sentiment-how-consumers-feel-about-shopping-in-store-online-and-via-mobile.html>, Last visited September, 24th, 2014.



- With the opportunity to shop in front of the computer, two aspects of the shopping experience are only possible with limitations. The social feedback component has been almost lost and the chance to try fitting is not existing anymore. We rely on the description and pictures offered by the shop and comments from other users. To make sure the ordered item fits, people have to get different sizes of favored items, making it inevitable that some of the items have to be send back. This is unpleasant for the customer and expensive for the shop.

In this chapter, we present a use case of an online retailer that aims to improve the shopping experience of men. By relying on the service of this retailer, customers no longer have to go shopping in physical stores but can order suitable fashion products online. Differing from conventional online shops where the customers browse through various products and eventually add items to their shopping basket, this shopping service relies on the expertise of fashion advisers who, after getting in contact with the costumers, arrange a combination of different clothes and ship them to the customers. The customer can then try out the clothes and accessories, keep the pieces they like and return the unwanted items free of charge.

In order to recommend an outfit, the fashion advisor has to consider the customer's preferences, context, materials, and a suitable combination of individual items. From a scientific point of view, we argue that the task of finding matching outfits is far more complex than that of traditional recommendation systems that recommend movies, songs or news articles. As outlined in Chap. 5, many recommendation techniques rely on the combination of content analysis and collaborative filtering to present users a list of choices. These techniques rely on computing a group of similar users, based on the history of purchased or watched items. This approach cannot be easily applied to the above scenario though since the service aims to take away the burden of inspecting items from the user. Instead, when recommending clothes, current trends, the customer's personal style, the occasion for that the clothes are required, as well as the correct size play an important role. Customers liking the same clothes does not necessarily mean that they are a good fit or that they are suitable for the user's current fashion need. Given these constrains, we consider the fashion recommendation task to be a constraint satisfaction task rather than a collaborative filtering task. The constraint satisfaction problem (CSP) defines a task where a satisfiable solution has to be found given a set of constraints. One well-known example for a CSP is the popular game Sudoku, where the numbers from one to nine have to be placed in a 9-by-9 grid of boxes such that each row, column, and 3-by-3 sub-grid contain each number exactly once. In our use case, constraints arise as users restrict item choices according to specific factors such as price, brand, or color. Fashion assembles outfits which most adequately consider these constraints.

The chapter is structured as follows. We will first introduce the current state-of-the-art on e-commerce systems with a focus on retailers recommending and selling clothes in Sect. 8.2. The related work covers scientific approaches as well as real-world examples of existing e-commerce applications. In Sect. 8.3, we introduce knowledge-based recommender systems, which are the super class of constraint-based recommendation systems, and explain the basic knowledge required for the

successful provision of fashion recommendations. Section 8.4 covers a detailed explanation of the mentioned online shopping service for men with an overview of the starting point for our approach. This section also covers the description of the data used in the following sections. In Sect. 8.5, we present the theoretical foundation and explanation of our proposed approach. The preliminary evaluation in Sect. 8.6 presents first results which are also discussed. Section 8.7 presents an outlook and outlines directions for future research.

## 8.2 Recommender Technologies in e-Commerce

Recommendation methods used in e-commerce, from most-popular to hybrid recommendations, are a common tool to assist people in finding suitable items to buy. A well-known and often-used approach is collaborative filtering, that uses groups of similar users, based on the purchase history, to recommend items. In the following section, we give a short overview of recommendation systems used in e-commerce applications and take a closer look at the domain of clothing recommendations. We will show that there is a difference in recommending brown goods or clothes.

Studies show that the process of actual buying clothes is heavily related to the mood of the user [38]. This means that having recommendations, the decision if a piece of cloth is bought or not is out of control for the algorithm.

Other studies showed that user preferences for clothes depend on their physical features [40, 45]. Raunio identified different physical features of clothes including skin response, size and shape of the clothes, thermal comfort, and fit (looseness and over-sized) revealing levels and visual features as important for the selection of clothes.

Delong et al. [14] found that preferences are composed of two parts: cognitive and affective. Affective preferences refer to emotions and mood of the user. The cognitive preferences are again referring to physical features such as product attributes, esthetic, and social attributes. The preference for product features are either extrinsic (e.g., price or brand name) or intrinsic (e.g., style, color, fabric, care, fit and quality) but they can differ based on the category of clothes, e.g., casual wear [11, 12, 17, 32].

In the next section, we will present current approaches to recommendation in the area of e-commerce, with a focus on clothes recommendations.

### 8.2.1 Current Approaches for RS in e-Commerce

Since the dawn of e-commerce applications in the WWW, recommendation systems (RS) have become an important tool to help users cope with the *Information Overload* problem and to help shop owners sell more items [41]. One well-known example is the amazon.com RS where users get “Other people who bought this also bought...” recommendations [31]. Current RS frequently use memory-based and

model-based collaborative filtering approaches, content-based filtering methods, or hybrid strategies thereof to recommend items [1]. These approaches use historical data to compute similarities between items or users which are then used to recommend matching items. The reliance on historical data can cause problems, particularly for new users or items. The system lacks sufficient information about them and fails to compute similarities. This is also known as the cold-start problem [34, 35]. The problem often occurs in domains with high sparsity. High levels of sparsity manifest as users buy few items. Thus, systems struggle to create accurate user profiles. Simultaneously, most items are seldom bought inducing similar issues.

Recommender systems face manifold challenges as they seek to suggest outfits. These challenges arise as recommending outfits differs from recommending movies. This is due to the fact that even customers with like-minded tastes cannot necessarily wear identical clothes. Jurca et al. [27] presented a study where they examined human foot sizes in large scale experiment with 10,000 foot scans. The results showed a high dispersion of foot widths within the various length classes. As the shoe sizes are computed using the length classes, all people with the same foot length should have the same shoe size. Since people with the same foot length have different foot widths, they cannot wear identical models. Narrower models will hurt people with wider feet. Conversely, wider models fail to adequately support people with narrower feet. This fact about shoes generalizes to clothes, as people with the same height do not necessarily have the same body measures and thus cannot wear the same clothes. Cacheda et al. [10] argue that pure CF approaches do not work in domains like cloth recommendations due to the fact that they do not exploit knowledge about the item itself. Therefore, different approaches to recommendations for clothes are exploited.

There exist RS that recommend similar products based on image retrieval. Hasan et al. [23] present a method for recommending similar products based on their images. The presented method removes noise, e.g. body parts, from the pictures before computing similarities. The method showed slight improvements over other approaches [21, 24, 29] using image-based methods. Other approaches aim for recommending clothes of a user's wardrobe based on context—mostly time of day, weather and purpose. Shen et al. [44] use a scenario-oriented approach where users have a virtual wardrobe where clothes have attributes such as brand or type. The clothes are also user tagged with information such as “I am going to...” or “I want to look more...” This information is used to recommend outfits when the user asks for an outfit to wear at the beach. Selected clothes are used as user feedback. Yu-Chu et al. [51] present a study where they used a Bayesian network to recommend clothes from wardrobes for a specific situation. The small evaluation suggests that the Bayesian network can learn user preferences and recommend clothes.

What we learn from the related work is that recommending suitable clothes to a user is hard and still not satisfyingly solved. To be able to compute good recommendations for clothes for a user, different aspects have to be taken into account:

- The user preferences for clothes: For example preferred colors, brands, or material. Also price preferences should be taken into account.

- The context of the user request for clothes: For example, leisure or business clothes. Preferences for clothes can be learned for different contexts.
- The fitting of clothes: Even if a cloth matching the user’s preferences, it is still important that the clothes fit.

With this knowledge in mind, we propose our approach in Sect. 8.5 which uses a constraint-based recommender to incorporate and satisfy the above-mentioned aspects. Before we detail our approach, we first introduce the concept of constraint-based recommendations, which are a subset of knowledge-based recommendation systems, in the next section.

### 8.3 An Introduction to Knowledge-Based Recommender Systems

Knowledge-based Recommender Systems are another major category of recommender systems besides collaborative and content-based recommender systems [9]. Knowledge-based recommender systems are typically used in domains where the items to recommend are not bought or interacted with very often. Movies or books for instance are domains where a lot of people buy or rate them and thus, information as input for content or collaborative approaches is given. In domains where items are bought less frequently, such approaches are not the best suited ones. The given scenario, an online shop for clothes specialized on men, is such an scenario, where only few information is given about the user themselves, see Sect. 8.4.1.

Knowledge-based RS combine knowledge about item attributes, domains and user preferences. They seek to determine whether a particular item suits the user’s needs. For instance, a vacation location where it is warm and affordable [18, 20]. One can distinguish between two types of knowledge-based recommenders—case-based and constraint-based recommender. Constraint-based RS try to find items that exactly matches users’ requirements using a predefined knowledge base containing rules how to relate user requirements and items. Case-based RS on the other hand utilize similarity metrics to match user preferences with item descriptions.

In this work, we employ case-based recommender to compute recommendations. Case-based recommender (CBR) utilize similarity functions to find a set of items matching the users’ queries, needs and/or preferences. CBR rely on a set of known cases, the case base. CBR use the case base to adapt or transfer the knowledge from previous cases to find selections of items satisfying the current recommendation request [7, 30]. Often, user requests for an item with certain attributes cannot be successfully handled. Either because the request of the user contains conflicting requirements or there is no matching item in the item base.

Current research works on methods for intelligent relaxations of constraints or repairing inconsistencies [19, 26]. An early application using a relaxation approach is *OpAmps*, which helps customers finding amplifiers, not by exact matches between the customer preferences (often those preferences are to specialized and cannot be fulfilled), but by finding the best matching products [47, 48]. One of the first examples

of using a conversational or critiquing approach is *FIND-ME* [22]. Critiquing approaches—systems interacting with users to narrow their preference spectra—are a common approach for knowledge-based recommender systems to better understand the users' intentions and needs [13]. In *FIND-ME*, users get a list of restaurant recommendations. They may pick restaurants and critique on the selection. Subsequently, the system recommends similar but yet cheaper restaurants based on the critique in case the user criticized the prices.

In the following sections, we present our approach utilizing a case-based approach to enhance recommendation quality in a clothing shopping example. We outline the research questions, our approach, describe the data we have and show preliminary results.

## 8.4 Case-Based Recommendation to Fill the Box

In this section, we explain the contributions of this work and research questions we want to address. In Sect. 8.5 we explain our recommendation approach, which builds a personalized user model based on users' previous orders and compare those results to standard recommendation approaches.

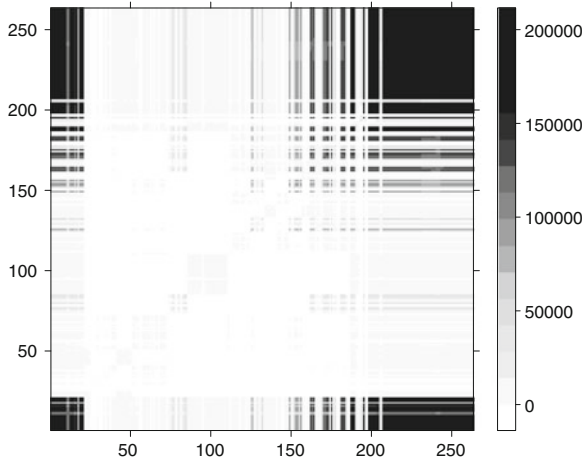
The overall goal is to improve the company's profit by reducing the return rates. What we learned from Sect. 8.3 is, that the decision of a user to keep an item is based on its quality, color, fit, and whether the user likes it or they do not. Our goal is to build a user model, that reflects those parameter and the users' personal tastes and then use this model to compute recommendations.

- **User Modeling:** How to build a user model from the given data? What information must be modeled? What can we learn from implicit feedback (returns) and explicit feedback such as the reasons customers mentioned for returning clothes?
- **Case-based Recommender:** Case-based recommender use similarities between the users' needs (in our case the user model) and the item base. We will build a CBR for the given scenario and compare results with of other recommender methods.
- **Measures for Evaluation:** Evaluating recommender systems is always challenging [43]. We will discuss what measures we use (see Sect. 8.5) to compare recommender systems and discuss implications (see Sect. 8.6).

The following section provides an description of the dataset used for the evaluation in Sect. 8.6 and also influencing the approach described in Sect. 8.5.

### 8.4.1 Data Description

Our data comprise a variety of features. These features include attributes linked to sales-related aspects, type information, and item specificities. We look at 263 features



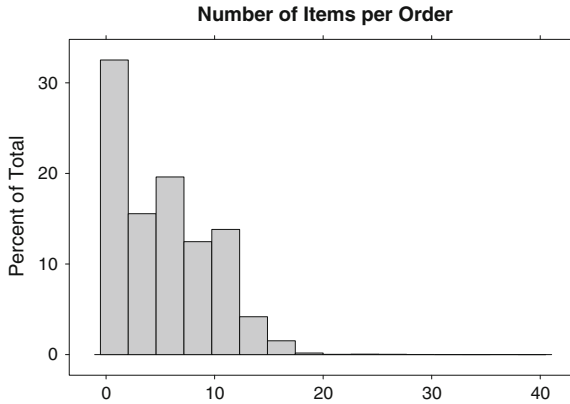
**Fig. 8.1** Co-occurrences of features in the data set. The figure represents features both as rows and columns. The order of rows and columns conforms with itself. Each point of the resulting grid shows the number of co-occurring values for each combination of features. The color intensity reflects the number according to the scheme on the *right-hand side*. We observe a large fraction of the space with relatively light coloring. This indicates high levels of sparsity

describing a transaction. Each such transaction represents an item being sent to a user. Subsequently, users decide to either keep or return the item.

Sparsity represents a vital factor for recommender systems. Figure 8.1 illustrates the data set’s sparsity levels. We compute the co-occurrences of feature values. Hereby, we refrain from considering individual values. We distinguish present values from missing values. Hence, we obtain a  $263 \times 263$  matrix whose values correspond to the number of co-occurring non-missing values. In other words, the more often two features exhibit non-missing values in transactions, the higher the count. We represent counts in terms of a color scheme detailed on the right-hand side. We observe that a relatively large fraction of space shows little to no counts. Thus, we consider our data to be highly scarce. Note that the darker regions reflect two types of phenomena. First, some features are available for all transactions. These features include identifiers for customer and article, references to the date and time, as well as the user’s decision to keep or return the item. Second, we observe that articles of the same kind commonly exhibit values for a subset of features. For instance, some features refer to shoes in particular. These attributes will lack values for all articles other than shoes. Although, shoes will likely carry values of those features even though the values may differ.

We categorize our features into five groups:

- transaction-related features
- type-related features
- descriptive features
- customer-related features



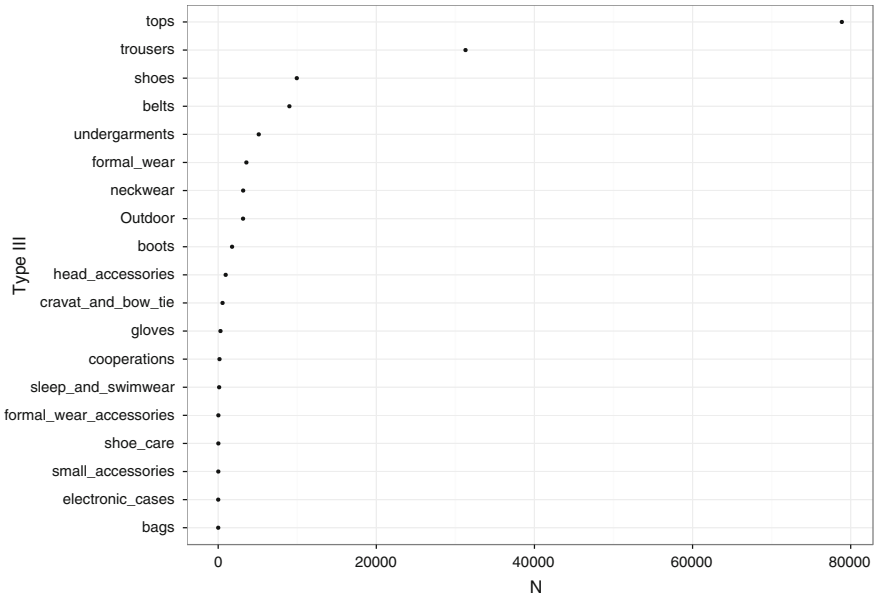
**Fig. 8.2** Distribution of items contained in orders. A majority of orders contains 1–12 items

#### 8.4.1.1 Transaction-Related Features

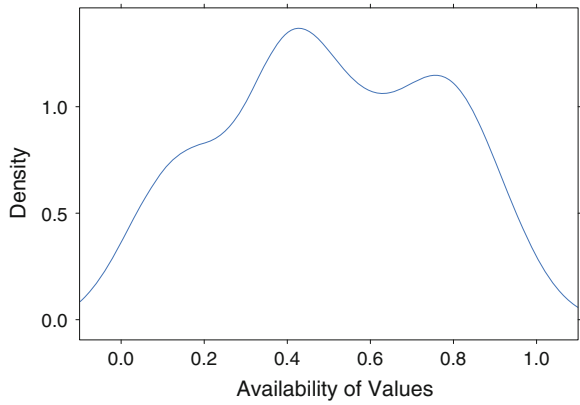
This kind of feature captures information on the transactions. Thus, these features represent the densest part of the data. Each transaction carries a referring identifier. Orders may comprise several transactions as customers receive outfits rather than individual clothes. Figure 8.2 illustrates the distribution of items per order. We observe that a majority of orders comprise up to 12 items. Additionally, transaction-related data include references to the article, the order date, and whether the article has been returned or not. The references to individual articles let us analyze their popularities. We analyze how often transactions include each article. We observe that the frequencies follow a *Power-Law* distributions. A rather large subset of transactions include a small set of popular articles. Conversely, a majority of articles rarely appears in transactions. Each transactions entails a timestamp. Thus, we can see temporal trends. The popularity bias occurs consistently over time.

#### 8.4.1.2 Type-Related Features

As Sect. 8.5.1 illustrates, clothes include a variety of items. Type-related features enable us to distinguish different types of items. The type information captures different levels of abstraction. The first level differentiates accessories ( $\approx 10\%$ ), apparel ( $\approx 83\%$ ), and footwear ( $\approx 8\%$ ). Another type refers to the dedicated gender. All items target male customers which relates to the business idea of the underlying service. Further, a feature provides type information for accessories and shoes. These categories include bags, belts, boots, shoe care, and more. Two additional features extend these categories to further include jeans, blazers, suits, and more. Individual articles may be assigned to several categories. For instance, the system assigns a belt to both accessories, belts, and targeting men. Figure 8.3 displays the distribution of instances of type III. We notice that tops and trousers account for the majority of transactions.

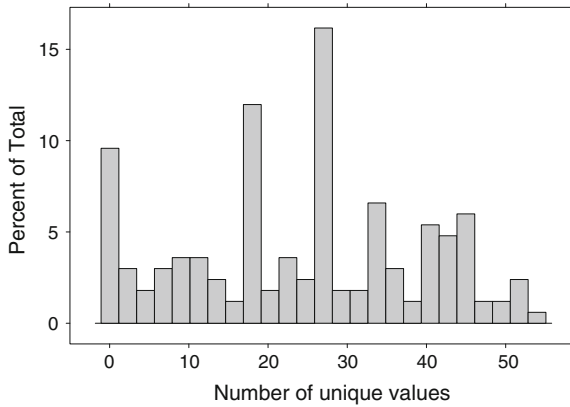


**Fig. 8.3** Exemplary distribution of type information. We observe that tops and trousers make up the largest fraction of transactions



**Fig. 8.4** Density of values for descriptive features. Note that only few attributes have values assigned to all transactions. Conversely, only few attributes do not assign values to any transactions





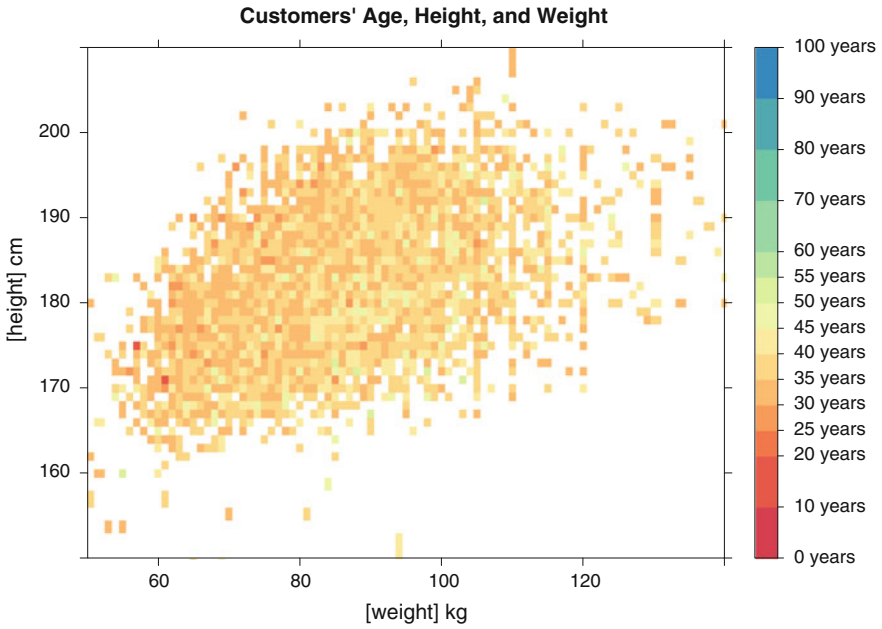
**Fig. 8.5** Distribution of unique values for descriptive features

### 8.4.1.3 Descriptive Features

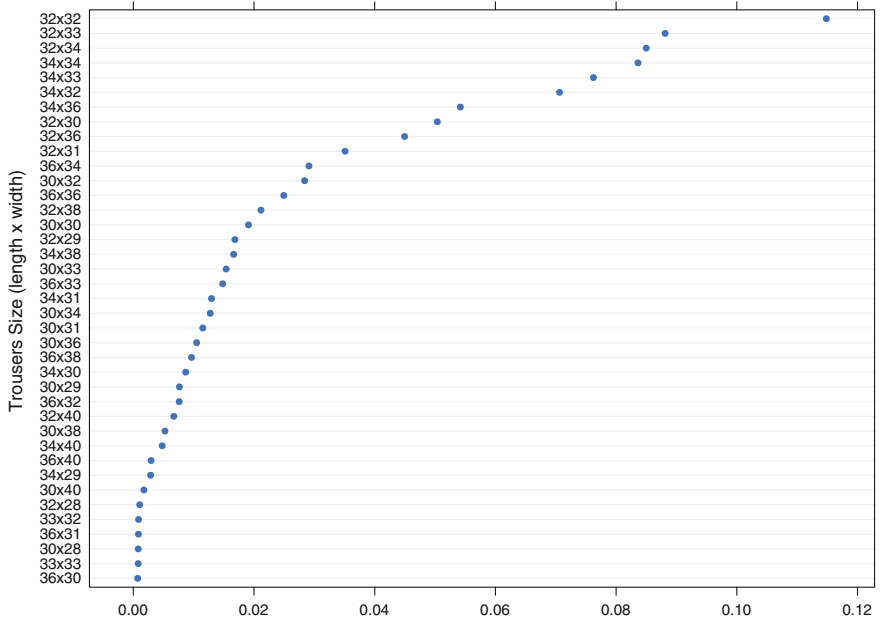
Descriptive features represent the majority of features. These features provide textual descriptions of articles. Most features apply to a specific type of article. For instance, the features “belt width” and “belt closure” apply to belts but not to hats. Consequently, most descriptive features only cover a fraction of articles. Figure 8.4 illustrates this relation. We notice that few descriptive features assign values to all or few transactions. Most features provide input for a subset of transactions. Consistency represents a major issue for textual features. This issue occurs particularly as values are manually entered. Figure 8.5 shows how the number of unique values distributes across descriptive features. We observe that most features exhibit a small number of unique values.

### 8.4.1.4 Customer-Related Features

The system receives customer-related features from two sources. On the one hand, users create profiles describing their personal characteristics including age, height, and weight. On the other hand, users fill out questionnaires about their tastes and needs. In addition, customers may comment on their experiences having received articles. We refer to this kind of features as *feedback-related* features. Figure 8.6 illustrates customers’ weight, height, and age. We see that customers’ age basically ranges from 25 to 50 years with few exceptions. Conversely, both weight and height are subject to substantial variances. Weights span from 55 to 120kg with some outliers on both sides. Heights stretch from 165 to 205 cm with outliers in both directions. Figure 8.7 shows the relative frequencies of individual trousers sizes. This information is essential for selecting appropriate sizes. Equivalently, customers provide their shoe, collar, and shirt sizes.



**Fig. 8.6** Customer characteristics. The plot shows the relations between customers' weight (x-axis), height (y-axis), and age (encoded as color scheme, see *right-hand side*)



**Fig. 8.7** Distribution of trousers sizes. The figure shows the relative frequency of individual trousers sizes

In addition, customers state the budgets they are willing to spend on new clothes. Customers may specify individual budgets for trousers, shirts, jackets, as well as shoes. Further, the system requests customers to answer a sequence of multiple-choice question. These questions ought to support systems in narrowing-down customers' tastes.

Based on the data description, we will explain our user model in Sect. 8.5.1 and the corresponding CBR in Sect. 8.5.2.

## 8.5 Case-Based Recommender Approach

In this section, we will explain our case-based recommendation approach and the underlying user model. Both, the approach and user model are fitted to the data described before.

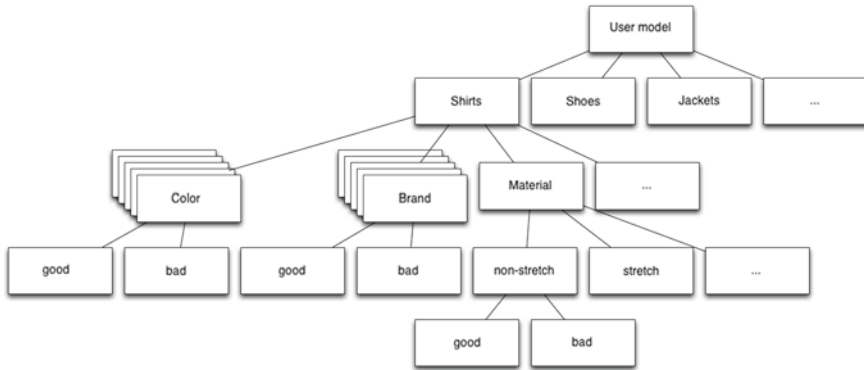
### 8.5.1 User Model

The model of the user preferences takes into account the findings of Sect. 8.2 which emphasizes items' attributes such as colors, brands, or material and fitting and user feedback, e.g., derived from the feedback field. The context, for instance leisure or business, is currently not part of the model, as the dataset does not reflect this at the moment.

The user model in this work follows the approach of overlay modeling of user interests as discussed in Brusilovsky and Millan [8]. The overlay approach models user knowledge or user interests as a subset of domain model. It is widely used in adaptive education systems where the knowledge of a student is often modeled in the form of “good/average/poor” knowledge for concept  $X$ , and  $X$  can be “math” for instance. Adapting this idea for our scenario, the domain model is given by the data we have about the clothes (see Sect. 8.4.1). The user model is an adaptation of that domain model. From the available user data, all orders of a user, a user model is build containing the user preferences (positive or negative preferences) on different abstraction layers.

The first layer is the top discrimination for clothes—the type of clothes. We have identified ten different clothes categories:

- Accessory
- Jackets
- Sakkos
- Pullover
- Shirts
- Trousers
- Shorts



**Fig. 8.8** User model for CBR: Exemplification of the user model

- Jeans
- Shoes
- Basics

For these different categories, we model the user preferences separately. Each category consists of attributes describing color preferences for the category, the brand, the material, or the price. For these attributes, we model what a user likes and what not. This allows us to look for color-matching clothes with colors the user prefers in one category but does not like for another one. We also model the preference for certain item attributes such as cut or imprints. Figure 8.8 depicts the structure of the user model.

### 8.5.2 CBR Approach

The model described in the previous section constitutes the basis for our personalized CBR approach. Given the model shown in Fig. 8.8, our CBR recommender learns from past purchases of the user what features, for what category, lead to a purchase, and which do not. This is done per user. Our CBR takes as an input all available purchases of the user, which consist of an article description with all features, the price and the information if the item was bought or returned. We have additional information about the user, containing demographic information and body size data. We also have information about the price range a user is willing to pay for an item for certain categories. One user may be willing to pay €50 for a shirt while others will abandon shirts of more than €30. These information are handled as explicit constraints by the user. Implicit constraints are derived from the article features that are bought or returned. Therefore, the CBR consist of two types of constraints—explicit ones set by the user and implicit ones derived from user purchases. Both types of constraints are handled differently. If the CBR detects a violation of an explicit

constraint, the article will be automatically rejected. In our case, currently only price constraints are set and used. In case an article is not falling into the user-defined price range, this article will not be recommended to the user, even if the article matches the implicit constraints. Our CBR approach consists of two phases. The learning phase, which builds a personalized user model, and the recommendation phase, where the user model is compared to an article to compute the similarity which is the basis for the recommendation.

The learning phase takes (procedure depicted in Algorithm 1) into account all users  $\mathbb{U} = \{u_1, u_2, \dots, u_n\}$ , purchased articles  $\mathbb{A} = \{a_1, a_2, \dots, a_n\}$  and if they bought or returned it  $\mathbb{I}(u, a) = \{\text{return}, \text{buy}\}$ . An article itself consists of a set of features  $\mathbb{F}$  defined as  $a = \{f_1, f_2, \dots, f_n\}$  where  $f \in \mathbb{F}$ . A user model consists of weights for all features of all articles the user interacted with:  $\text{um} = \{\{f_1, w\}, \{f_2, w\}, \{f_n, w\}\}$ . The code example Algorithm 1 shows the general computation of these implicit constraints. Explicit constraints, as said, are not computed but a predefined set, which is filled in the learning phase with information from the given user data.

The goal of the learning phase is to have a user model representing an ideal article for the user, which can be compared to other articles. The model consists of different sub-models for the different categories.

The recommendation phase compares a given article  $a$  using a similarity function  $\text{sim}(a, \text{thresholdPositive}, \text{thresholdNegative})$ , see Algorithm 2. The similarity function  $\text{sim}$  compares the user model with the given article. The function also takes thresholds defining when an article is marked as recommendable or not.

---

**Algorithm 1** Learning constraints for the User Model.

---

```

for all  $i \in \mathbb{I}$  do
  if  $i = \text{buy}$  then
    for all  $f \in a$  do
      Add  $f$  to user model and increase weight
    end for
  else
    for all  $f \in a$  do
      Add  $f$  to user model and decrease weight
    end for
  end if
end for

```

---

If an article matches the user model, it will be recommended. Similarity is currently measured by counting the number of features overlapping the user model and the article. Beside articles marked as recommendable, the CBR will also mark articles as definitely not recommendable (e.g. by failing the price constraint). As for the following evaluation, we focus on correct recommendations of articles.

---

**Algorithm 2** Recommending articles based on similarity between user model and item features.

---

```

for all  $f \in a$  do
  if  $f \in \text{umANDweigth} \geq \text{value}$  then
    Increase count for matching feature  $\rightarrow$  countMatchingFeatures ++
  else
    Increase count for NOT matching feature  $\rightarrow$  countNotMatchingFeatures ++
  end if
end for
if countMatchingFeatures  $\geq$  thresholdPositive then
  Recommend Article
end if

```

---

## 8.6 Evaluation of CBR

The CBR approach presented in the previous section is evaluated and compared using the described dataset. The main measure for comparison is *Precision*. More precisely, the precision for the *Not Returned* class. So, the best recommender is the one with the best precision for items a user will keep. The decision to use *Precision* was done in cooperation with the company providing the dataset. The prediction what item a user will most likely keep has the biggest impact on revenue and profit. Precision is defined as the fraction of correctly predicted items (TP) and the number of all items, correct predictions (TP), and incorrect predictions (FP), the recommender predicted as bought. In our scenario, *TP* are all clothes which are bought and which the recommender predicted as clothes that are likely to be bought. *FP* denotes the results where the recommender predicted that the clothes are likely to be bought but the clothes were not bought.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (8.1)$$

Optimizing and focusing on one measure, precision as described before, could imply that we miss out other information, about the data, that another measure would have shown. Given our scenario, we have a two-class prediction problem—Not Returned or Returned. The measure described before only takes into account the prediction for class *Not Returned*. To not loose the sight of the two-class problem, we also present results for the performance of predicting class *Returned*. The measures of choice for this is *Accuracy*. It also takes into account the correct predictions for the true negatives (TN). *TN* is here defined as items which are predicted as a probably return and which are returned. *Accuracy* is explained in Eq. 8.2. It is the fraction of all correctly predicted (*TP* and *TN*) instances versus all predictions made.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(P + N)} \quad (8.2)$$

As a comparison with our approach, we used different other approaches which we will briefly introduce.

- All Returns/All Kept: As most of the purchases in the dataset are returns, we used one dummy algorithm to predict that the purchase was returned. This serves as a baseline for the accuracy measure. For the precision measure, we used the opposite logic, marking all as kept purchases.
- KNN: K-Nearest Neighbor is a classification algorithm using K examples of the training set most similar to the item to classify. Classification is done by majority vote of its neighbors. In our evaluation, we have chosen  $k = 5$  neighbors [2].
- Naïve Bayes: A probabilistic classifier based on the Bayes Theorem [39]. Naïve Bayes classifier are, for instance, used in text categorization to classify items into spam or non-spam.
- Ensemble: Ensembles are a combination of different approaches with an extra algorithm combining the results from different algorithms. In this evaluation, we used Stacking [49] as the combination methodology and combined the previously mentioned approaches with our CBR algorithm. Stacking was used by the two top teams of the Netflix-Challenge [4, 6]. Stacking uses an extra learning algorithm, in our case Logistic Regression, to make a final prediction out of the results of the other approaches.

The evaluation was done on a subset of the dataset described in Sect. 8.4.1. We deleted all users with less than 20 items in their purchase history. This was done as the evaluation was conducted using a train/test split of the users purchase history. With less than 20 items, not enough data for training the algorithms and test them would have been available. We also removed some of the features, as they contained clear indicators of an item that was chosen or not. Out of the 263 features, we removed 13, so that we ended up with 250 features per item.

For the results, we conducted five test runs. In each test run, we iterated over all users (3,747 users in the evaluation data), extracting all user data from the dataset, splitted the user data into 80% training set and 20% test set randomly, and then integrated the users' training data back into the data set. As a result, we got a dataset consisting of all data except the separated test data of one user. This test set was then used to evaluate the results of the different approaches. This evaluation approach was chosen, as we compare different types of algorithms. Our CBR only takes into account the data of one user, thus it needs only the 20+ items of the user to evaluate. Other approaches, like Naïve Bayes learn their model on the complete dataset, to find discriminators to make the prediction. As we want to see how the different approaches work in a personalized setting, recommendations for one user, we chose the previously described data splitting. The results of the five test runs were then averaged. Figure 8.9 shows the result based on precision.

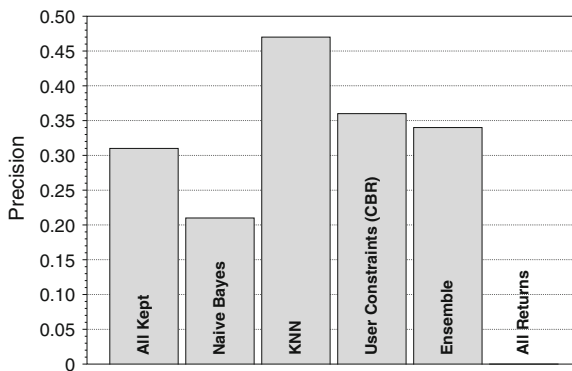
Figure 8.10 shows the result based on accuracy. We see that the performance differs compared to the precision performance.

If we look at the precision setting, we see that the Naïve Bayes approach performed poorly when compared to the baseline and the other approaches. Our approach, CBR, performs good compared to the baseline. As we only use data from one user, building

the user model without the purchase history, we can see that it is possible to deduce preferences of a user for certain types of clothing. The best approach, of the single approaches, is the KNN approach, which outperforms the others by quite a big margin. The results for KNN and Naïve Bayes can be explained by the type of data. Most of the attributes in the dataset are numeric. As Naïve Bayes cannot deduce any information out of numbers, it is reasonable that Naïve Bayes performs weak. The KNN approach uses Euclidean distance as the similarity measure. Euclidean distance, opposite to Naïve Bayes can cope with numeric variables. Thus, the performance of KNN is quite good. Our CBR approach focuses on learning preferences for the different features by taking into account previous purchases. We handle all features equally, not making differences between numbers and text. We only weigh the value positively or negatively. Thus, we also do not work with similarity of numeric values, but we are also not ignoring them as Naïve Bayes does.

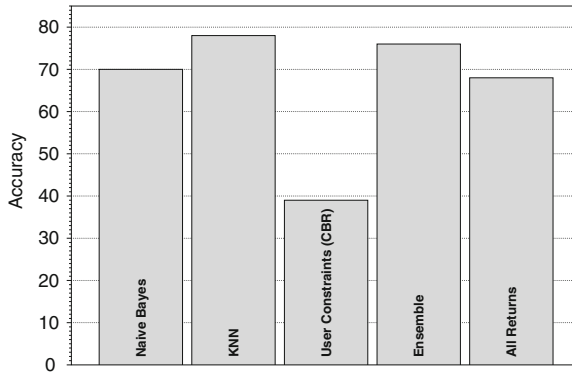
Taking a closer look at the accuracy measure, we first see, that the “all returns” algorithm performs quite well. This is of course expected and shows that the chosen measure is an important factor in an evaluation. What we also see, and why we added the accuracy measure to the evaluation, is that our CBR approach performs worst. As we optimized it, tweaking the threshold to predict kept items, this is expected as well. The performance of KNN is the best, also in the accuracy measure, leading to the conclusion that there are certain item attributes, possibly numeric attributes, that contain information allowing the prediction if an item is kept or not. The ensemble is together with the KNN the top approach with respect to accuracy.

The evaluation shows that learning preferences is a good approach to improve the rate of items kept (meaning the user bought it). But we also see that our CBR approach misses some of the information contained in the dataset. Therefore, next steps have to include an extension of the CBR that takes and handles attributes more differentiated. As the results induce, different attributes have different information



**Fig. 8.9** Results for precision measure of the different approaches. CBR is the approach discussed in this chapter





**Fig. 8.10** Results for the accuracy measure of the different approaches

value. In the next section, we present a short analysis of the different attributes and the information gain they offer.

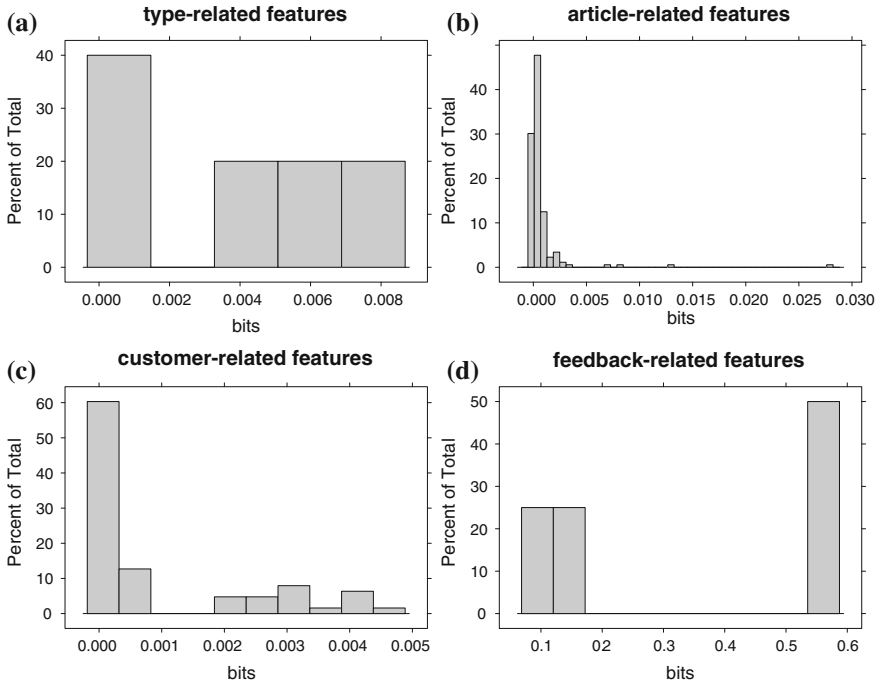
### 8.6.1 Information Value of Attributes

We noted that the recommendation quality depends on the way our algorithm treats available attributes. We have seen that treating all features equally negatively affects the performance. We suppose this to be due to individual features carrying more or less valuable information about the relation between users and items. We determined each feature’s value in terms of “mutual information” with the information about whether an article had been returned or not. Hereby, the system encodes the target quantity as a binary variable. Customers may either return or keep an article. On the other hand, features’ domains include categorical, ordinal, and numeric value ranges.

$$I_{X,Y} = \sum_{x \in X, y \in Y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \quad (8.3)$$

Equation 8.3 illustrates how we compute mutual information [36].  $X$  and  $Y$  represent two random variables. In our case,  $Y$  refers to the customer keeping or returning an article.  $X$  represents the feature for which we seek to determine the information contained. Note that we employ the logarithm with base 2 to obtain information in term of bits. Alternatively, we could use logarithms to base  $e$  which would provide information in terms of nits.

Figure 8.11 depicts how the mutual information distributes across different types of features. We notice that only feedback-related features carry a considerable amount of information. Unfortunately, feedback-related features exhibit highest levels of sparsity. In addition, customer who had not bought any article yet will lack this



**Fig. 8.11** Distributions of mutual information for the four types of features, **a** type-related, **b** descriptive, **c** customer-related, **d** feedback-related

**Table 8.1** Features with highest mutual information excluding feedback-related features

Feature	Type	Mutual Information
Care instructions	Descriptive	0.0132
Article type class 5	Type	0.0083
Brand	Descriptive	0.0067
Article type class 4	Type	0.0059
Article type class 3	Type	0.0050
Jacket preference	Customer	0.0047
Shoe preference	Customer	0.0043
Age	Customer	0.0042
Jeans preference	Customer	0.0042
Weight	Customer	0.0041

information completely. Excluding the feedback-related attributes, we detail the next best features in Table 8.1.

We observe little discriminative power even by the most informative descriptive, type-, and customer-related features. Care instructions and brands carry most information in the scope of descriptive features. The different type classes stick to

acomparable range of information levels. Preferences for jackets, shoes, and jeans carry similar information compared to customers' ages and weights. We notice that missing a buying history even the most sophisticated recommendation method will struggle. This is mostly due to lack of sufficient information contained in the available data.

## 8.7 Conclusion and Outlook

In this chapter, we presented the use case of a menswear online shop with the problem of how to improve the rate of kept items among its customers. We defined this as a constraint satisfaction problem. The user preferences are the constraints and our goal is to build a recommender that learns those preferences based on item features and recommends items which the user most likely will buy. We therefore introduced a recommender in Sect. 8.5.2 where we presented a first version of a case-based-reasoning approach using an overlay user model to make recommendations for a single user. Results are comparable with other approaches, see Sect. 8.6. We showed that without any further analysis of the impact of different features, we already reached good results. The next step here is to take a closer look at the different features, which are currently all handled equally, and see if boosting some of them improves the results. We also want to use more of the data, given in the profile, such as shirt size, etc. as explicit constraints. Another direction would be using algorithms like singular value decomposition, to deduce latent features, which could then be used to further improve our CBR approach.

The question of solving the cold-start problem could not be answered satisfactorily. To personalize, we need information about the user, more than currently available in the user model. The purchase history is still the most important source of information. As a cold-start user does not have such history, one possible solution could be learning cluster of similar users and then use those combined purchase histories as a start-up model for the cold-start user. The field of unsupervised machine learning offers a variety of techniques which allow to cluster users as well as items. This toolbox includes  $k$ -Means, hierarchical clustering, and component analysis amongst others methods [15]. All these methods take users represented as feature vectors as input. They seek to find patterns between these vectors. As a result, the system can project new users onto clusters even without purchase history. Recently, researchers have proposed representation learning [5]. Learning representations of objects provides similar capabilities. The system may select the most similar representations instead of particular clusters. Auto-Encoders represent a successful technique for representation learning. They consist of a multi-layered architecture of neural nets. Each layer provides a more abstract representation of the target object. This allows systems to learn hidden concepts. In the case of fashion recommendation, we may detect taste patterns among subsets of customers. Additionally, we plan to apply meta-learning techniques to further improve our approach's quality. *Bagging* and *Boosting* represent two possible choices of meta-learning procedures.

Systems using *Bagging* replicate instances of objects to better account for unbalanced classes. For instance, we may assume that a majority of customers prefers a certain subset of articles. In contrast, a minority of customers may dislike these. Replicating customers of the minority supports the system not to suggest the articles' subset which the minority dislikes. On the other hand, *Boosting* simultaneously learns models with varying parametrization. Thus, systems obtain more robustness by considering different aspects. Finally, we seek to support the systems as new articles enter the collections. Applying similar clustering techniques to the collection of articles allows the system to detect similarity patterns among them. Having established both article and item clusters, systems may learn models on their interactions. For instance, systems may detect that users of user cluster *A* are particularly likely to buy articles of item cluster *B*. Observing preferences of user clusters for particular item clusters enables the system to better control the selection process.

**Acknowledgments** This work was funded by the Federal Ministry of Economic Affairs and Energy (BMWi) under funding reference number KF2086104KM3.

## References

1. G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin, Context-aware recommender systems. *AI Mag.* **32**(3), 67–80 (2011)
2. D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms. *Mach. Learn.* **6**(1), 37–66 (1991)
3. J. Baumgartner, *You Are What You Wear: What Your Clothes Reveal About You* (Da Capo Lifelong Books, Boston, 2012)
4. R.M. Bell, Y. Koren, Lessons from the Netflix prize challenge. *SIGKDD Explor. Newsl.* **9**(2), 75–79 (2007)
5. Y. Bengio. Deep learning of representations for unsupervised and transfer learning, in *JMLR: Workshop and Conference Proceedings*, pp. 17–37 (2012)
6. J. Bennett, S. Lanning, N. Netflix, The Netflix prize, in *KDD Cup and Workshop in Conjunction with KDD* (2007)
7. D. Bridge, M.H. Göker, L. McGinty, B. Smyth, Case-based recommender systems. *Knowl. Eng. Rev.* **20**(9), 315–320 (2005)
8. P. Brusilovsky, E. Millan, User models for adaptive hypermedia and adaptive educational systems, in *The Adaptive Web: Methods and Strategies of Web Personalization*, Chapter 1, ed. by P. Brusilovsky, A. Kobsa, W. Nejdl (Springer, Berlin, 2007), pp. 3–53
9. R. Burke, Knowledge-based recommender systems, in *Encyclopedia of Library and Information Systems* (Marcel Dekker, New York, 2000), p. 2000
10. F. CACHEDA, V. Carneiro, D. Fernández, V. Formoso, Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web* **5**(1), 2:1–2:33 (2011)
11. M.A. Casselman-Dickson, M.L. Damhorst, Female bicyclists and interest in dress. *Cloth. Text. Res. J.* **4**, 7–17 (1993)
12. M.-H. Chae, C. Black, J. Heitmeyer, Pre-purchase and post-purchase satisfaction and fashion involvement of female tennis wear consumers. *Int. J. Consum. Stud.* **30**, 25–33 (2005)
13. L. Chen, P. Pearl, Critiquing-based recommenders: survey and emerging trends. *User Model. User-Adapt. Interact.* **22**(1–2), 125–150 (2012)
14. M.R. DeLong, B. Minshall, K. Larntz, Use of schema for evaluating consumer response to an apparel product. *Cloth. Text. Res. J.* **5**, 17–26 (1986)

15. R. Duda, P. Hart, D. Stork, *Pattern Classification* (Wiley Interscience, New York, 2001)
16. J. Entwistle, *The Fashioned Body: Fashion, Dress and Modern Social Theory* (Blackwell Publishers Inc., Malden, 2000)
17. B.L. Feather, S. Ford, D.G. Herr, Female collegiate basketball players' perceptions about their bodies, garment fit and uniform design preferences. *Cloth. Text. Res. J.* **14**, 22–29 (1996)
18. A. Felfernig, R. Burke, Constraint-based recommender systems: technologies and research issues, in *Proceedings of the 10th International Conference on Electronic Commerce, ICEC'08*, (ACM, New York, 2008), pp. 3:1–3:10
19. A. Felfernig, M. Schubert, C. Zehentner, An efficient diagnosis algorithm for inconsistent constraint sets. *Artif. Intell. Eng. Des. Anal. Manuf.* **26**(1), 53–62 (2012)
20. A. Felfernig, G. Friedrich, D. Jannach, M. Zanker, Developing constraint-based recommenders, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Springer, New York, 2011), pp. 187–215
21. C. Grana, D. Borghesani, R. Cucchiara, Class-based color bag of words for fashion retrieval, in *2012 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 444–449 (2012)
22. K.J. Hammond, R. Burke, S.L. Lytinen, A case-based approach to knowledge navigation, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95*, vol. 2 (Morgan Kaufmann Publishers Inc., San Francisco, 1995), pp. 2071–2072
23. N. Hasan, A. Hamouda, T. Deif, M.A. El-Saban, R. Shahin, Using skin segmentation to improve similar product recommendations in online clothing stores, in *VISAPP*, ed. by S. Battiato, J. Braz, SciTePress, pp. 693–700 (2013)
24. T. Iwata, S. Watanabe, H. Sawada, Fashion coordinates recommender system using photographs from fashion magazines, in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI'11*, vol. 3 (AAAI Press, 2011), pp. 2262–2267
25. J. Jacoby, Perspectives on information overload. *J. Consum. Res.* **10**(4), 432–436 (1984)
26. D. Jannach, Techniques for fast query relaxation in content-based recommender systems, in *Proceedings of the 29th Annual German Conference on Artificial Intelligence, KI'06* (Springer, Berlin, 2007), pp. 49–63
27. A. Jurca, T. Kolsek, T. Vidic, Dorothy mass foot measurement campaign, in *Proceedings of 1st International Conference on 3D Body Scanning Technologies* (Lugano, Switzerland, 2010), pp. 338–344
28. S.B. Kaiser, *Social Psychology of Clothing: Symbolic Appearances in Context*, 2nd edn. (Fairchild Publications, New York, 1997)
29. P. Kakumanu, S. Makrogiannis, N. Bourbakis, A survey of skin-color modeling and detection methods. *Pattern Recognit.* **40**(3), 1106–1122 (2007)
30. D.B. Leake, Case-based reasoning. *Knowl. Eng. Rev.* **9**(3), 61–64 (1994)
31. G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput.* **7**(1), 76–80 (2003)
32. E.D. Lowe, M.M. Dunsing, Clothing satisfaction determinants. *Home Econ. Res. J.* **9**, 363–373 (2009)
33. N.K. Malhotra, Reflections on the information overload paradigm in consumer decision making. *J. Consum. Res.* **10**(4), 436–440 (1984)
34. B. Mehta, Cross system personalization: enabling personalization across multiple systems. Ph.D. thesis, Universitaet Duisburg-Essen (2008)
35. B. Mehta, C. Niederee, A. Stewart, M. Degemmis, P. Lops, G. Semeraro, Ontologically-enriched unified user modeling for cross-system personalization, in *User Modeling 2005*, vol. 3538, Lecture Notes in Computer Science, ed. by L. Ardissono, P. Brna, A. Mitrovic (Springer, Berlin, 2005), pp. 119–123
36. M. Mezard, A. Montanari, *Information, Physics, and Computation* (Oxford University Press, Oxford, 2009)
37. W.W. Moe, Buying, searching, or browsing: differentiating between online shoppers using in-store navigational clickstream. *J. Consum. Psychol.* **13**(1—2), 29–39 (2003). Consumers in Cyberspace

38. W. Moody, P. Kinderman, P. Sinha, An exploratory study. *J. Fash. Mark. Manag.: Int. J.* **14**(1), 161–179 (2010)
39. A. Papoulis, Bayes' theorem in statistics and Bayes' theorem in statistics (reexamined), in *Probability, Random Variables, and Stochastic Processes*, 2nd edn. (McGraw-Hill, New York, 1984), pp. 38–39, 78–81 and 112–114
40. A.-M. Raunio, Favorite clothes—a look at individuals' experience of clothing. Technical Report Research Report No. 161, Department of Teacher Education, University of Helsinki (1982)
41. J.B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in *Proceedings of the 1st ACM Conference on Electronic Commerce* (ACM, 1999), pp. 158–166
42. B. Schwartz, *The Paradox of Choice* (Harper Collins, New York, 2009)
43. G. Shani, A. Gunawardana, Evaluating recommendation systems, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Springer, New York, 2011), pp. 257–297
44. E. Shen, H. Lieberman, F. Lam, What am I gonna wear?: scenario-oriented recommendation, in *Proceedings of the 12th International Conference on Intelligent User Interfaces, IUI'07* (ACM, New York, 2007), pp. 365–368
45. M.R. Solomon, N. Rabolt, Consumer Behavior, in *Fashion*, 1st edn. (Prentice Hall, Upper Saddle River, 2003)
46. G.B. Sproles, *Fashion, Consumer Behavior Toward Dress* (Burgess Publishing Company, Minneapolis, 1979)
47. I. Vollrath, W. Wilke, R. Bergmann, Case-based reasoning support for online catalog sales. *IEEE Internet Comput.* **2**(4), 47–54 (1998)
48. W. Wilke, M. Lenz, S. Wess, Intelligent sales support with CBR, in *Case-Based Reasoning Technology, From Foundations to Applications* (Springer, London, 1998), pp. 91–114
49. D.H. Wolpert, Stacked generalization. *Neural Netw.* **5**(2), 241–259 (1992)
50. T.P. y Monsuwé, B.G.C. Dellaert, K. de Ruyter, What drives consumers to shop online? A literature review. *Int. J. Serv. Ind. Manag.* **14**, 102–121 (2004)
51. L. Yu-Chu, Y. Kawakita, E. Suzuki, H. Ichikawa, Personalized clothing-recommendation system based on a modified Bayesian network, in *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT)*, pp. 414–417 (2012)

# Chapter 9

## Gamification of Workplace Activities

Michael Meder, Brijnesh Johannes Jain, Till Plumbaum  
and Frank Hopfgartner

**Abstract** Gamification—taking game design patterns and principles out of video games to apply them in non-game environments has become a popular idea in the last 4 years. It has also successfully been applied to workplace environments, but it still remains unclear how employees really feel about the introduction of a gamified system. We address this matter by comparing the employees’ subjective perception of gamification with their actual usage behavior in an enterprise application software. As a result of the experiment, we find there is a strong relationship visible. Following up on this observation, we pose the gamification design problem under the assumptions that (i) gamification consists of various types of users that experience game design elements differently; and (ii) gamification is deployed in order to achieve some goals in the broadest sense, as the problem of assigning each user a game design element that maximizes their expected contribution to achieve these goals. We show that this problem can be reduced to a statistical learning problem and suggest matrix factorization as one solution when user interaction data is given. The hypothesis is that predictive models as intelligent tools for supporting users in decision-making may have the potential to support the design process in gamification.

### Procrastination

Steven lifted his head from the office desk and instinctively looked at the clock at the wall. 5 o’clock in the afternoon. Once again, he had fallen asleep at his desk. It was the eighth time already that this had happened this month. Well, at least he had an individual office for himself so that his colleagues did not notice when

---

M. Meder (✉) · B.J. Jain · T. Plumbaum · F. Hopfgartner  
Technische Universität Berlin, Berlin, Germany  
e-mail: michael.meder@dai-labor.de

B.J. Jain  
e-mail: brijnesh-johannes.jain@dai-labor.de

T. Plumbaum  
e-mail: till.plumbaum@dai-labor.de

F. Hopfgartner  
e-mail: frank.hopfgartner@tu-berlin.de

he was sleeping at work. Feeling excessively miserable, Steven started reading short news articles on the Internet, thinking it would make him feel better (but as usual, it did not). Starting to transform his bad conscience into energy to change his work attitude, he realized that his boss had sent him an email during his short nap.

“Hi Steven, we need an overview about the requirements management system. Could you bring it upstairs to me by 6 p.m. today!” Steven frowned. “By 6 o’clock! This is less than 1 h from now,” he realized. “Writing such an overview in only 1 h is impossible...for me at least.”



Trying to calm down, Steven opened a new tab on his browser and started reading some short news. Irritated, he stopped. “I really should reconsider my online behavior,” he thought. “I am way too often procrastinating.” Yet again, he felt excessively miserable. Concerned about what he should send to his boss, he suddenly remembered the old enterprise bookmarking system that was deployed in his company. The system was designed to allow employees to collect and share links to documents. “It was actually quite nice,” he thought. By adding a document link to the system, one could also provide a short description and categorize the document using short terms, so-called tags. “Maybe someone shared a link there that can help me to write this report.” He searched for the system announcement email, because he could not remember the web address to the system. After Steven found and opened it in his browser, he was surprised by the small number of overall bookmarks stored in the system. The system was announced 3 years ago and there were only 148 bookmarks in them (in a company with around 200 employees). “Whatever,” he thought and started to search for all bookmarks with the tags *requirements* and *management*. Within the blink of an eye, the system displayed the one bookmark it had found. This was very disappointing for Steven. With a slight *whoomph* sound, his head struck on the desk and stayed there for a couple of seconds. “What shall I do now?” He already imagined sitting in front of his boss and apologizing again for missing yet another internal deadline...



This little scene happened 2 years ago. Steven had already forgotten it. It did not have any consequences for him anyway since his boss actually had already left the office and hence didn't even notice that Steven didn't deliver this report on time. Just suddenly, Steven had a little *déjà vu*. He had just woken up from yet another short power nap at work—disturbed by the loud buzzing of his email client. It was an email from the enterprise bookmarking system which was recently enriched with some game design elements. A point system, virtual badges for special activities and a leaderboard. Especially recommending bookmarks to colleagues are rewarded with the highest number of points but only if the colleague confirms the bookmark as an interesting one for him. Steven had heard that they called it gamification, meaning applying game techniques to non-gaming environments.

Steven had a closer look at the message that he had just received. Scott, one of his colleagues, who was also a good friend, recommended Steven a link on procrastination research and the top anti-procrastination smart phone apps. Scott knew about Steven's problem with getting things done and also falling asleep at work. "Interesting," Steven thought bemused. "I never thought that this bookmarking system would ever be of any use." He smiled when he remembered that one day when he hoped that this system might help him in preparing a report. What a deserted piece of software it was back then. No comparison to today at all. Ever since they gamified the bookmarking system, he and his colleagues had used it extensively. It was a fun competition in his office. Who would end up on top of the leaderboard at the end of the month? He smiled again and clicked on the link that Scott had sent him. "Okay Scott, you can have these points for recommending the link to me. But be assured, I will find something to return to you." After all, it was a tight race for the top leaderboard position. He also made a mental note to complain about bullying at the workplace the next time he would meet Scott.

## 9.1 Introduction

User engagement, participation, crowdsourcing, the "Wisdom of the crowd." In the beginning of this century, the ever-growing number of active online users raised hopes that big, information-rich, interactive online archives could be created with the help of these users. Although various popular portals (e.g., Wikipedia, Youtube, or Flickr) exist in the Web that are built around user content, Nielsen [26] and Stewart et al. [31] argue that many companies suffer from the participation inequality problem. Based on this observation, we assume that in almost every company, at least one software system exists that needs more user activity to provide a remarkable benefit for the employees and the management.

This leads us to the question of what kind of software or system is engaging and motivating to its users, especially in the long term. Long-term motivation also plays an important role in the gaming business. Well-designed board and video games provide various means to incentivize their players to continue playing. Considering the success of such games, it is worth investigating whether these methods and principles

can also be exploited to promote long-term usage of enterprise systems. In fact, since 2010, the application of these methods and principles, also referred to as gamification, is a trending topic for marketing- and business-oriented services. Deterding et al. [6] define gamification as “the use of game design elements in non-game contexts”. A rather benefit-oriented definition was provided by Huotari and Hamari [15] who define it as “a process of enhancing a service with affordances for gameful experiences in order to support user’s overall value creation”. In today’s online world, one stumbles upon gamification elements on various sites. Stackoverflow<sup>1</sup> uses a reputation leaderboard where users get points for helpful answers. Dropbox<sup>2</sup> rewards users who spread the word with more space and LinkedIn<sup>3</sup> is motivating users to complete their profiles by presenting progress bars. Not only on the Web, but also in the biochemistry [18] and education domains [21], using game elements becomes a non-neglectable part.

A common objective of gamification is to enhance motivation. In an enterprise setting, for example, the goal is to motivate employees to participate in a certain task. If, for example, a project requires constant documentation of project activities which is not a very popular task, gamifying the documentation process should address this lack of motivation. Finding the right means to increase motivation is a nontrivial task though since motivation is mainly driven by human-centric factors. For some people, being on top of a leaderboard can be motivating, but what about people who are not in the top  $N$ ? Are these people really motivated to rise up on the leaderboard? Also, what does their position on the leaderboard say about their work performance? Does it show that they are not working enough and do they have to fear negative consequences by the management? These questions indicate that especially in an enterprise scenario, it is of uttermost importance to measure challenges and risks that occur due to these differences before introducing gamification methods. On the one hand, we expect gamification to increase user participation within an enterprise. On the other hand, the visibility of user interaction (or lack thereof), e.g., the position of the employee on a leaderboard can increase the stress level of employees or even cause fear that their activities on a gamified system will be used as an indicator of their engagement with the company. Gamification could have some negative side effects (negative manipulation, denunciation, blaming) but while it could also have positive effects the implications need to be carefully investigated.

We argue that user-specific gamification design could reduce participation inequality. We aim to show that better knowledge on how to motivate each user by which game design elements can increase chances for converting a lurker into an active, regularly participating user. Furthermore, we reason that a user-specific application of game design elements can also prevent cross-cultural problems since automatic user type determination can address cross-cultural habits. More specifically, with this chapter and future studies we intend to answer the following research questions:

---

<sup>1</sup> <http://stackoverflow.com/>.

<sup>2</sup> <http://dropbox.com/>.

<sup>3</sup> <http://linkedin.com/>.

1. Is it possible to reliably predict game design elements in enterprise systems from click stream data analysis?
2. Is user specific gamification feasible? Which game design elements could be used simultaneously in one gamified system without negative impacts on others?
3. How to evaluate the improvement of user specific gamification design?

In this chapter, we examine the gamification design process in detail. The chapter is structured as follows: Sect. 9.2 provides a literature survey on gamification and its application in enterprise systems. In order to move one step closer to answering the above research questions we performed a two-part experiment in a workplace environment described in Sect. 9.3. The first part of the experiment is an online questionnaire about users' expertise and perception of gamification methods. In the second part, we introduce a gamified social bookmarking system and analyze the participants' engagement with this system over a period of one week. We then compare the actual system usage to the answers users submitted in the questionnaire. Building on this, we propose a statistical approach to solve the gamification design problem in Sect. 9.4. After discussing the implications of this chapter, we conclude the work and provide an overview of future research directions in Sect. 9.5.

## 9.2 Related Work

In this section, we provide a detailed overview of the concept of gamification. Section 9.2.1 first provides an overview of the history of gamification and outlines established definitions. In Sect. 9.2.2, basic elements of gamification, namely game design elements, are introduced. Section 9.2.3 outlines the role of the user in the gamification process. Gamified systems that are applied in a workspace environment are presented in Sect. 9.2.4.

### 9.2.1 History and Definitions

In 1886, S&H Green Stamps, a United States company, started one of the first retail loyalty programs by offering stamps to U.S. retailers. In the following years, the idea to bind customers to companies spread to other domains, e.g., by the introduction of airline frequent flyer, hotel loyalty, and car rental programs. Over the years, games started to conquer our living rooms. As early as 1990, 30% of American households owned at least one of Nintendo's NES.<sup>4</sup> The *Generation Gamer* was born. In 1996, Bartle published his four player types taxonomy [2] in which he suggests to classify gamers into different categories—socializers, killers, achievers,

---

<sup>4</sup> According to "Fusion, Transfusion or Confusion/Future Directions In Computer Entertainment." Computer Gaming World. December 1990, p. 28. <http://www.cgwmuseum.org/galleries/index.php?year=1990&pub=2&id=77>. Retrieved 12 September 2014.

and explorers. It is a starting point for many game design tasks. Thanks to the Serious Game Initiative, the term serious game became more known and as a link between video games and non-entertainment purposes like training, education, health, policy, and management issues. In 2002, Pelling coined the term gamification, “by which [he] meant applying game-like accelerated user interface design to make electronic transactions both enjoyable and fast.”<sup>5</sup> Bret Terril<sup>6</sup> and James Currier<sup>7</sup> also wrote about the term gam(e)ification in 2008 as a new marketing instrument to increase engagement by using game mechanics. Since 2010, the application of gamification [6, 15] is a trending topic for marketing and business-oriented services. In 2012, Gartner Inc. predicted that “over 70 % of global 2000 organisations will have at least one gamified application by 2014”.<sup>8</sup> Given this industry focus on gamification, the concept also received more attention from academia. In 2011, two definitions of gamification were published. Deterding et al. [6] define gamification as “the use of game design elements in non-game contexts.” Huotari and Hamari [15] define it as “a process of enhancing a service with affordances for gameful experiences in order to support user’s overall value creation.” We interpret both definitions as implying a goal as the utility of gamification. Both describe elements of the game design world which could change a user’s experience in a different context (non-game [6], service [15]). Interestingly, for Deterding et al. [6] “[...] the term ‘gameful design’—design for gameful experiences—was also introduced as a potential alternative to ‘gamification.’” Summarizing, in Deterding’s definition the goal is rather geared toward the (improved) user experience itself, in Huotari and Hamari’s definition it is the outcome driven by the user experience.

## 9.2.2 Game Design Elements

An important aspect of successful gamification is the selection of game design elements. Game design elements determine what type of gameful experiences are generated for the users. In [6], Deterding et al. provide five levels of game design elements. They distinguish between game interface design patterns, game design patterns and mechanics, game design principles and heuristics, game models, and game design methods (Table 9.1). Robinson et al. [29] propose a taxonomy built on levels of expected engagement and the required commitment of the user. This taxonomy has been conceived as a decision support for game element selection.

---

<sup>5</sup> The (short) prehistory of ‘gamification’ <http://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/>. Retrieved 12 September 2014.

<sup>6</sup> <http://www.bretterill.com/2008/06/my-coverage-of-lobby-of-social-gaming.html>. Retrieved 12 September 2014.

<sup>7</sup> <http://blog.oogalabs.com/2008/11/05/gamification-game-mechanics-is-the-new-marketing/>. Retrieved 12 September 2014.

<sup>8</sup> <http://www.gartner.com/newsroom/id/1844115>. Retrieved 12 September 2014.

**Table 9.1** Levels of game design elements by Deterding et al. [6]

Level	Description	Example
Game interface design patterns	Common, successful interaction design components and design solutions for a known problem in a context, including prototypical implementations	Badge, leaderboard, level
Game design patterns and mechanics	Commonly reoccurring parts of the design of a game that concern gameplay	Time constraint, limited resources, turns
Game design principles and heuristics	Evaluative guidelines to approach a design problem or analyze a given design solution	Enduring play, clear goals, variety of game styles
Game models	Conceptual models of the components of games or game experience	MDA; challenge, fantasy, curiosity; game design atoms; CEGE
Game design methods	Game design-specific practices and processes	Playtesting, playcentric design, value conscious game design

### 9.2.3 User Types

Designing gamification is also always a user-oriented process. This is due to the fact that users are all individuals driven by different input factors like age, gender, education, social skills, and cross-cultural influences [11, 17, 35–37]. In the game world this is considered by several player typologies developed on user observations and in-game behavior. Hamari et al. [14] list existing game player typologies. They state that player types have their legitimation because of the different behavior and motivation of players. It is a widespread assumption that also for the gamification scenario such types of players, respectively, users can be applied. Although many player typologies exist we argue that it is hard to map them to one or more specific game design elements. Beyond that, such types could change over time which seems to be a central criticism on player typologies [14]. Furthermore, we argue that applying a set of game design elements to cover all different types in a gamification scenario could have negative influence on each other.

### 9.2.4 Gamification in the Workplace

Various studies have been performed that indicate that gamification has a favorable effect on the use of enterprise systems. In [24], Dugan et al. describe the transformation of an enterprise bookmarking system into a guessing game called Dogear. In this game, bookmarks and their tags are displayed on screen and the players have to guess, *who* created this bookmark. If they guess the correct creator of the bookmark, the players can gain points. The Dogear game is inspired by von Ahn’s ESP game [34] where users gain points when they use the same tags to describe

the content of an image as their teammates. Differing from the ESP game, which exploits “human computation” for the annotation of images, Dogear focuses on providing methods to learn more about colleagues and their expertise, hence increasing familiarity within a company. They report that within the first month of the release of the system, they had 87 active players from 10 different countries. A detailed analysis is still missing though.

Farzan et al. [10] examine the impact of game mechanics, more precisely the introduction of a points system, on a social enterprise network system (Beehive, IBM). They evaluate the impact of this points system by performing A/B testing, i.e., one half of all users are made aware of the points system, while the other half (i.e., the control group) cannot see this feature. They observe that overall, the introduction of the points system increased the activity level of the users within the system. However, they also report that 72% of the users in the experimental group never visited the page which describes how to earn points. Besides, they argue that a large portion did not even notice the existence of points.

Addressing this issue further, Farzan et al. [8] also studied if there is any noticeable effect on the usage when the points system is explicitly explained to the users. Therefore, they provided further details via email and repeated the experiment. They conclude that points systems can successfully be employed to motivate users to contribute more in an enterprise social network system, especially if combined with email notifications. Further, they conclude that the type of contribution can directly be controlled by the type of gamification applied, i.e., increasing the points for certain types of contributions will indeed result in an increase of contributions of this type. In a follow-up experiment, Farzan et al. [9] increase the social interaction and diversity of content even further by introducing a badge based approach on promoting content. Although they observe an increased activity due to the introduction of gamification methods, the authors argue that they cannot make any statement about the quality of the contributions. Further studies are needed to examine this in detail.

Evaluating the effect of gamification methods from a different perspective, Thom et al. [32] study whether the *removal* of gamification features from an enterprise social media system has any measurable effect on user activity. They report a significant decline of user activities after removing gamification features, concluding that extrinsic rewards influence user behavior. Interestingly, the authors also noticed some relation between user activity and their geographical location. This supports our premise that there are many factors that can have an impact on the success of workplace gamification.

Hamari [12] evaluates the use of badges in a peer-to-peer trading service. He observes that the introduction of gamification mechanisms does not automatically result in an increased use of the system by all users, but that those users who actively inspect their own badges become more active. This supports our assumption that individual behavior plays an important role in the successful application of gamification methods in an office scenario.

Summarizing, previous research reports an increase of users’ activity in an enterprise due to diverse game mechanics. However, these studies also indicate that individual behavior has a significant influence on the success of gamification.

To the best of our knowledge, there currently exists no study on employees' perception of gamification. Therefore, we attempt to better understand employees' behavior in more detail.

### 9.3 An Enterprise Gamification Experiment

Gamification methods have been applied in various environments and for different purposes such as enterprise workplaces, education, pervasive healthcare, e-commerce, human resource management, and many more (e.g., [1, 4, 27]). Although these studies indicate that gamification can lead to increased user activity, a detailed analysis of users' perception of gamification principles has hardly been studied. We are all individuals and are driven by different input factors such as our personality, as well as social or cultural differences [11, 17, 35–37]. Especially in an enterprise scenario, it is of utmost importance to measure challenges and risks that occur due to these differences before introducing gamification methods though. On the one hand, we expect gamification to increase user participation within an enterprise. On the other hand, the visibility of user interaction (or lack thereof), e.g., the position of the employee on a leaderboard can increase the stress level of employees or even cause fear that their activities on a gamified system will be used as an indicator of their engagement with the company. Although gamification has successfully been applied in office scenarios, it remains unclear how employees really feel about the introduction of a gamified system at their workplace.

In this section, we address this issue from two directions. First, we present the outcome of an online survey where we analyze users' opinion about gamification in a workplace environment. Then, we analyze the interaction logs of a redesigned gamified enterprise bookmarking system to compare the employees' subjective perception of gamification with their actual behavior when using a gamified system. Results indicate that there is a strong relationship between employees' perception of gamification and their actual interaction with such system.

We examine the role of gamification in a workplace environment from an employee's point of view. More specifically, we aim to answer the following research questions:

- Do employees perceive gamification as a positive or negative factor?
- How is the perceived role of gamification reflected by actual usage patterns?

Aiming to address these questions, we defined an online questionnaire on users' expertise and perception of gamification methods, distributed it among employees of a technical research institute, and evaluated their responses. To evaluate whether their subjective answers are on a par with their actual interaction with a gamified system, we further introduced a gamified Social Enterprise Bookmarking System and analyzed the participants' engagement with this system over a period of 1 week.

### 9.3.1 Questionnaire

In order to address the first research question on employees' perception of gamification principles, we created an online questionnaire where participants were asked to judge various statements on a Five-Point Likert scale. In the remainder of this section, we provide further details about the participating subjects and their responses.

#### 9.3.1.1 Subject Recruitment and Details

We recruited participants by sending a brief introduction and a link to the online questionnaire to a mailing list of our research institute at an electrical engineering and computer science department of a major European technical university. Subscribers of this mailing list are over 140 members of this institute, including faculty members, administrative staff, postdoctoral researchers, Ph.D. students, and student research assistants. Given that all subjects are members of a technical research institute, we assume that all participants are highly familiar with using computer systems. To the best of our knowledge, none of the participants has professional experience with gamification in an enterprise setting. In order to participate, subjects had to authenticate using their institute account. We received a response from 53 subjects (6 female, 47 male), i.e., over one-third of all subscribers of the mailing list. As part of the questionnaire, they were asked to provide their age in a predefined range. Twenty-three subjects claim to be between 18–29 years old, 26 subjects are between 30–39 years old, 3 subjects are between 40–49 years old, and one to be 50 years or older. Since this distribution roughly matches our institute's age and sex distribution, we argue that the participants are a representative subset of the institute's workforce.

#### 9.3.1.2 Participants' Responses

In the remainder of this section, we introduce all statements that the participants had to answer on a Five-Point Likert scale.

In the first question ( $Q_1$ : "contribution"), we asked them to state how often they share or contribute content on enterprise systems such as Wikis or Enterprise CMS. Here, we could observe a rather conservative pattern, i.e., 33.9% (18 in total) of all subjects said that they sometimes contribute content, 33.9% (18) said that they seldom contribute and 11.3% (6) never share or contribute on such systems. Only 20.7% said that they often (8) or very often (3) contribute content. These answers are in line with our own (subjective) observations and the analysis of Wikipedia contributions by Ortega et al. [28], showing that content is often contributed by few individuals.

After asking the subjects to assess their current contribution and share activities on enterprise systems, the subjects were asked in question ( $Q_2$ : "familiarity"), to state how familiar they are with the term "gamification." This was also the first time we mentioned the term gamification in the questionnaire. 66% of all participants claimed that they were either to a great extent (10 subjects) or somewhat (25) familiar with



it. Fourteen subjects said that they have very little knowledge about it while four subjects were not familiar with it at all. We conclude from these responses that there is a general awareness of gamification principles (especially among those colleagues who closely work with the authors of this paper), while many lack further details to describe these principles.

In the following question  $Q_3$  (“motivation”), we were further interested in their own attitude toward gamification. Therefore, we asked them to judge whether gamification would motivate them to participate even more in enterprise systems. In order to guarantee that all subjects had the same understanding of gamification, we also provided a brief definition (as stated on Wikipedia) that should help to better understand this question. 43 % of all participants stated that they were undecided on how to judge this statement. 34 % of them agreed to this statement while 22 % disagreed (9) or strongly disagreed (3). Although only a few more than those who disagree tends to agree with this statement. With 43 % answered undecided it is evident that the subjects are not very convinced of the role of gamification in an enterprise environment. This goes in line with  $Q_2$  which indicates that the subjects are not too familiar with the concept.

In the next question ( $Q_4$ : “positive effects”), we asked the participants to state whether game mechanics like points, badges, and leaderboards have a positive effect on the enterprise and its staff. While 33.9 % were undecided, a majority of over 50 % either agree (26) or strongly agree (1) with this statement. Only eight participants disagreed (4) or strongly disagreed (4). This distribution of judgements seems to indicate that the majority of participants have a rather positive perception of gamification principles.

In the last question ( $Q_5$ : “negative effects”), we wanted to know whether the participants believe that there are negative effects on the enterprise and its staff caused by game mechanics like points, badges, and leaderboards. Here, a clear preference can not be observed. 49 % of all subjects were undecided, 22.6 % either agreed or disagreed with this statement while only 2 participants strongly agreed with it.

Summarizing, in this section, we aimed to evaluate whether gamification is seen positively or negatively within an enterprise context. The analysis of our online questionnaire revealed that although many participants of the study are (to some extent) familiar with gamification and are convinced that it can have a positive effect on an enterprise and its staff, they are not convinced that it can serve as intrinsic motivation for themselves to contribute more on enterprise systems. Addressing this question further, we present a gamified enterprise tagging system in the next section which shall shed further light on the difference between the perceived and the actual role of gamification on users’ behavior in an enterprise setting.

### ***9.3.2 Gamified Enterprise Tagging***

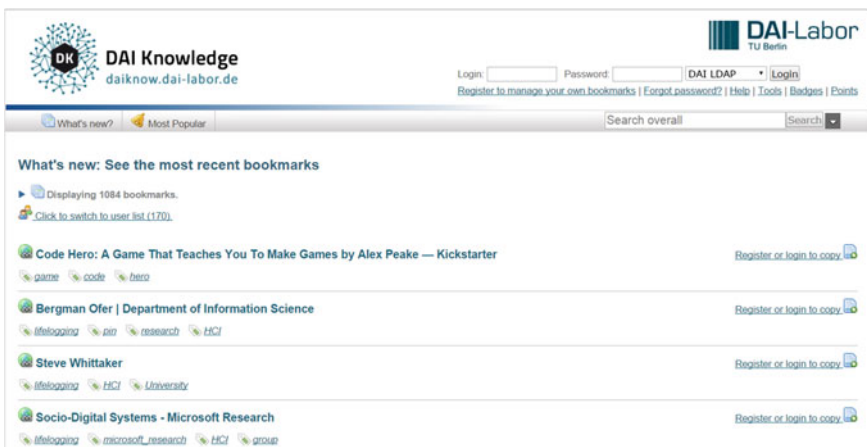
Following our first research question on employees’ perception of gamification, we were further interested in comparing this perceived role with users’ actual interaction

with a gamified system. In the remainder of this section we first introduce such a system, i.e., an enterprise bookmarking system. Then, we introduce the gamification elements that have been added to the system. An evaluation of the users' interaction with this system is provided in the next section. Later in this section, we provide usage statistics of the system *before* and *after* we enriched it with gamification functionalities.

### 9.3.2.1 Legacy Enterprise Bookmarking System

Using tags to manage items became a common tool on the Internet, among Delicious<sup>9</sup> to tag bookmarks, Flickr<sup>10</sup> uses tags to manage photos or Youtube<sup>11</sup> for video and LastFM<sup>12</sup> for music. Enterprises adopted tagging to manage and structure knowledge in an enterprise and to provide an alternative approach to helping employees find needed information. IBM for instance created the 'Enterprise Knowledge System' (ETS). The main idea of this system is that bookmarks can be annotated using tags and shared with others. The system allows tagging of documents, people, and other resources, thus linking the items to extra knowledge. People for instance can be tagged with their expertise.

In 2009, we developed a social bookmarking system (Fig. 9.1) in close cooperation with a large company. Social bookmarking systems became popular in the



**Fig. 9.1** The main view of the bookmarking system, displaying all publicly bookmarked items

<sup>9</sup> <https://delicious.com/>.

<sup>10</sup> <http://flickr.com/>.

<sup>11</sup> <http://youtube.com/>.

<sup>12</sup> <http://lastfm.com/>.

early years of this century with the rise and success of Delicious. Providing similar functionalities as the well-known Delicious system, our system additionally allows to create bookmarks for files on internal file server, taking into account existing rights management and the possibility to share bookmarks not only with other people but also with people having a certain tag.

The system has two main views, the personal site showing all bookmarks of a user, public and private ones (a user can mark bookmarks as private being only visible to them and not appearing in public searches) and a general view, called ‘What’s new’ showing all public bookmarks. Bookmarks and users can both be tagged and searched. The system also offers a set of tools to ease the bookmarking process. We developed a JavaScript-based bookmarklet, allowing people to easily bookmark web pages and a Windows tool to bookmark files. Also integrated is an automatic tags recommendation system recommending tags based on the bookmarked item (extracting the most important words from the describing text) and the most used tags by the user. We also integrated a so-called “Most Popular” section, showing the most frequent used tags of the week and of all time to help users see what topic is currently trending in the company, pictured in Fig. 9.2.

Our bookmarking system was intended to be a prototype to test the usefulness and acceptance of such a tool within the enterprise. Hence, the developed bookmarking system was not intended to be a full featured, product-like, bookmarking system. Nevertheless, the system was well accepted among the employees of the enterprise (test group was one department), and led to the management decision to install such a system enterprise wide. Consequently, the system was made available to

### Top 10 most used bookmarks this week

Click on a tag to search in bookmarks.

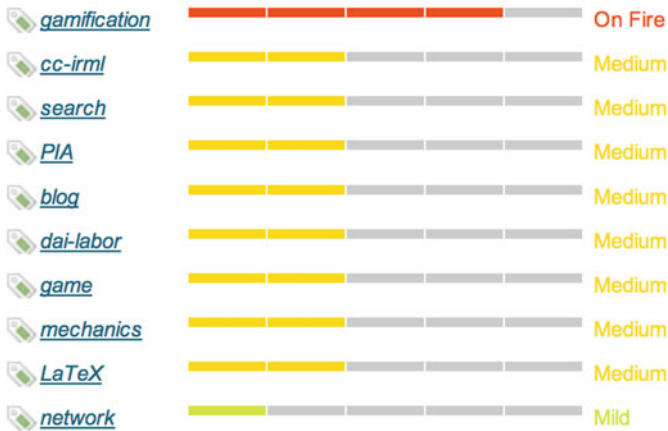


Fig. 9.2 Overview of the most used tags in the bookmarking system this week. The all time list is visualized accordingly

100 employees. However, since its introduction in 2009, the number of active users reduced significantly. The main reason is probably usability flaws since the system was never optimized with respect to user friendliness. Besides, changing staff led to a slow oblivion of the system's existence.

### 9.3.2.2 Gamification Elements

In the context of this study, the existing system was carefully extended with three well-known and researched gamification elements—*Points*, *Badges*, and a *Leaderboard* [23]. The three elements were chosen to cover different types of motivation and Gamification types. Rewards, with points and badges, for fulfilling different tasks, and a competition element with the leaderboard. While the points reward direct interaction with the system such as creating a new bookmark, badges are given for completing longer term goals such as continuously creating new bookmarks in a certain period of time. How to achieve points and badges is explained to the user by detailed descriptions of which actions score what points and what is needed to achieve a badge. The leaderboard, accessible only after the user logged in, is integrated in the main menu and within easy reach for the user. In the remainder of this section, we introduce the applied gamification elements in detail.

**Points:** Almost every user interaction with the bookmarking system gives points, including daily login (200 points), adding a public/private bookmark (100/25 points), adding a bookmark with at least one tag (25 points), accepting a bookmark recommendation (200 points) and many others. Figure 9.3 shows an overview of the actions and the respective points.

**Badges:** In the gamified bookmarking system, we introduced five different types of badges. For getting points, creating bookmarks, loyalty (regular usage), and positive recommendations. Figure 9.4 shows the badges used including the explanation of what has to be done to achieve them.

**Leaderboard:** In the leaderboard (Fig. 9.5), users are ranked by points in decreasing order. Besides, achieved badges are displayed to promote them further. The board shows two rankings: the monthly leaderboard on the left-hand side which is automatically reset every month and the all time leaderboard on the right-hand side. This is done to avoid frustrating new users and to create a new challenge every month.

**Feedback:** To ensure that the user is aware of points and badges, we integrated a message system (Fig. 9.6). From the first log in, users get a small on-site popup message, also called toast message. These messages appear every time a user gets a reward by points or badges and if the user reaches a higher rank on the leaderboard. Moreover, the user gets a message when another user copied one of her bookmarks or accepted a bookmark recommendation. After 10 s, the messages disappear automatically.

Added one public bookmark.	100
Bookmark with at least one tag.	25
Bookmark with three or more tags.	50
Added one private bookmarks.	25
Copy as a public bookmark.	50
Copy as a private bookmark.	10
Bonus: Recommendation accepted and copied.	200
Bonus: One of your bookmarks was copied.	25
Bonus: You tagged a User.	50
Daily Bonus: Login.	200
Daily Bonus: 1st bookmark.	100
Daily Bonus: 2nd bookmark.	50
Daily Bonus: 3rd bookmark.	25

Fig. 9.3 Overview of the actions one receives for performing that action




























 <b>Newbie</b> You earned at least 1 point for actions like adding a bookmark.	 <b>Beginner</b> You earned at least 500 points for actions like adding a bookmark.	 <b>Senior</b> You earned at least 1500 points for actions like adding a bookmark.	 <b>Bachelor of Bookmarking</b> You earned at least 5000 points for actions like adding a bookmark.
 <b>Bookmark-Master</b> You earned at least 20500 points for actions like adding a bookmark.	 <b>Prof. Dr. Bookmark</b> You earned at least 50000 points for actions like adding a bookmark.	 <b>First Bookmark</b> You added your first public bookmark.	 <b>First private Bookmark</b> You added your first private bookmark.
 <b>Bookmark-Copy</b> You made a public copy of a bookmark.	 <b>Private Bookmark-Copy</b> You made a private copy of a bookmark.	 <b>First Login</b> You have logged in for the first time.	 <b>Recommendation accepted</b> One user accepted your recommendation.
 <b>10 Bookmarks</b> You added 10 public bookmarks.	 <b>30 Bookmarks</b> You added 30 public bookmarks.	 <b>100 Bookmarks</b> You added 100 public bookmarks.	 <b>300 Bookmarks</b> You added 300 public bookmarks.
 <b>1000 Bookmarks</b> You added 1000 public bookmarks.	 <b>Five accepted Recommendations</b> Five of your recommendations were accepted.	 <b>Ten accepted Recommendations</b> Ten of your recommendations were accepted.	 <b>30 accepted Recommendations</b> 30 of your recommendations were accepted.
 <b>50 accepted Recommendations</b> 50 of your recommendations were accepted.	 <b>70 accepted Recommendations</b> 70 of your recommendations were accepted.	 <b>150 accepted Recommendations</b> 150 of your recommendations were accepted.	 <b>Two Logins</b> You have logged in on two different days.
 <b>Five Logins</b> You have logged in on five different days.	 <b>Ten Logins</b> You have logged in on ten different days.	 <b>30 Logins</b> You have logged in on 30 different days.	

Fig. 9.4 Set of achievable badges with a short description

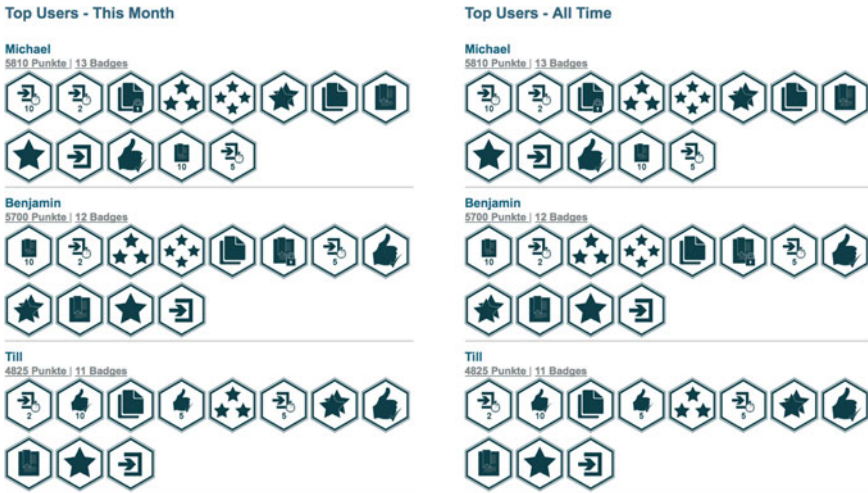


Fig. 9.5 The leaderboard with the monthly leader (left) and the all time leaderboard (right). The values are the same because it is the first gamified month

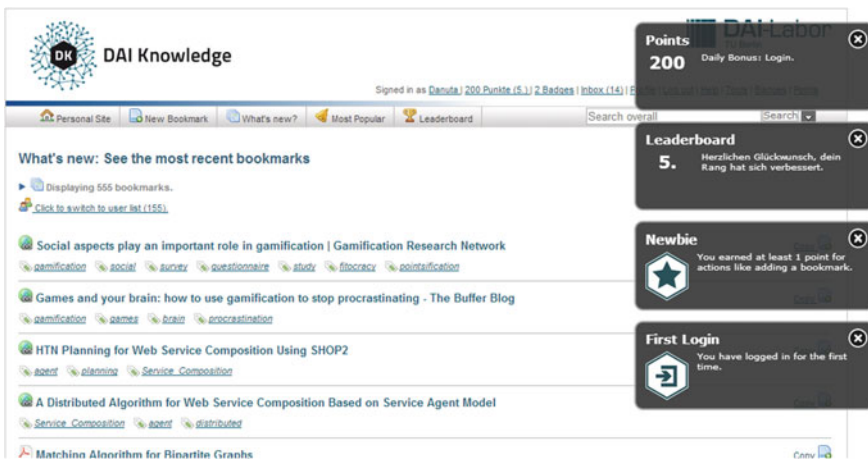


Fig. 9.6 Messages appearing on the user's first log in

Summarizing, we enriched an existing enterprise bookmarking system with the gamification functionality that, according to our literature review, has been proven useful in an enterprise scenario, namely points, badges, and leaderboards.

### 9.3.3 Experiment and Results

In order to evaluate our second research question of whether users' perception of gamification (as stated in the online questionnaire) matches with their behavior using a gamified system, we announced the relaunch of the gamified enterprise bookmarking system using the institute-wide mailing list and observed the users' interaction with the system. In this section, we provide an overview of the system's usage from 2009 till the introduction of gamification functionalities in mid-June 2013 to provide insights into the usage of the system *before* it got gamified. Following that, we present the usage statistics *after* the gamified version was introduced.

#### 9.3.3.1 System Usage Before Gamification

In the analysis of usage patterns we look at data from 2009 until the gamified version went online in June 2013.<sup>13</sup> The focus of the analysis is to look at two important aspects of the bookmarking system: The main feature, the creation of bookmarks, and the number of recommended bookmarks to colleagues, one of the main goals for enterprises to support knowledge sharing. Since a user-centric evaluation of this system was never intended for the legacy system, the (pre-gamified) system did not track any user interactions. Thus, only actions resulting in a database entry can be tracked. This was changed in the gamified version.

Table 9.2 shows the yearly distribution of created items and active users. Outliers, in this case a power user responsible for half of all system interactions, are removed from the statistics. After a solid start in 2009 with roughly one-third of employees using the system, 2010 already showed a significant decline in usage. While the usage number remained stable for the years 2010–2013, only 3% of people were using the system.

Table 9.3 shows the total number of bookmark recommendations. Similar to the created items, the number peaked in 2009. From 2009 till 2013, they remained at a stable level. This comes as no surprise, as the system's user interface (UI) supports the recommendation during a bookmark creation. Recommending an item at a later stage was still possible, but not as compelling as in the creation process. Noticeable is the comparison of the number of users. As shown in Table 9.2, we recorded a total of 30 users in 2009 with only 14 of them using the recommendation feature. This might be due to the mentioned UI flaws, but could also indicate that those users use the system only for personal managing purposes. From 2010 till 2013 usage numbers remain the same, suggesting that the users who use the system over a long period of time are well familiar with its functionalities.

The statistics show that the non-gamified version of the bookmarking system was used only rarely. In the next section, we present and discuss results of the gamified

---

<sup>13</sup> The gamified version went online on June 19. The presented analysis however only considers data till June 1, to prevent any influence of the data because of any office kitchen chatter about the bookmarking system.

**Table 9.2** Created items per year and the number of active users

	2009	2010	2011	2012	2013 <sup>a</sup>
Bookmarks	291	29	30	13	26
Users	30	4	4	3	5

<sup>a</sup> Until June 1st

**Table 9.3** Recommended bookmarks per year and the number of active users

	2009	2010	2011	2012	2013 <sup>a</sup>
Recommendations	61	34	31	26	18
Users	14	4	4	3	5

<sup>a</sup> Until June 1st

version of the bookmarking system. Then, we compare users' feedback from the questionnaires with the actual usage patterns to answer our second research question. Finally, we discuss limitations and shortcomings of the presented study and discuss future steps for gamification in enterprises.

### 9.3.3.2 System Usage After Gamification

As mentioned above, we tracked user interaction to analyze their behavior while interacting with the gamified system. For tracking their behavior, we relied on the Open Web Analytics<sup>14</sup> platform which captures users' interaction with all HTML elements of the system. Using this software, we respect the "do not track" option that can be set in the browser, resulting in incomplete or no web tracking data for some users of the system. Besides, we had to interrupt the experiment shortly after announcing the redesigned system due to a software bug. Two days later, we reannounced the system. All user interactions that took place within these two days has been omitted from this evaluation.

After announcing the system, a total of 18 users registered with the system. Seven of them, however, did not use the system at all. Fourteen of the registered users also participated in the online questionnaire, nine of these users logged in more than once. One user received a badge and 200 points for successfully recommending a document to another user. This user has left the institute over 2 years ago, i.e., he recommended this document a long time ago. Obviously, this user did not participate in the online questionnaire.

Table 9.4 provides an overview of the users' activities with the system. They are ranked based on their position in the leaderboard after 1 week. The users who are annotated with the <sup>a</sup> Symbol did *not* fill in the online questionnaire. It can be seen, for example, that after 8 days (from Wed to Wed) of use, the user with rank 1 ( $U_{\text{rank}_1}$ ) gained a total of 5,700 points, collected 12 badges, visited the leaderboard 58 times, and other pages with gamification elements 7 times. In total, the user visited pages

<sup>14</sup> <http://www.openwebanalytics.com/>.



**Table 9.4** Leaderboard 1 week after email announcement of the gamified bookmarking system

Leaderboard			# page requests has game design elements		
Rank	Points	# Badges	All	Leaderboard	Other
1	5,700	12	130	58	7
2	4,300	9	230	22	14
3	2,625	9	_ <sup>b</sup>	_ <sup>b</sup>	_ <sup>b</sup>
4 <sup>a</sup>	2,100	9	_ <sup>b</sup>	_ <sup>b</sup>	_ <sup>b</sup>
5	1,675	7	51	0	13
6	1,360	8	47	5	7
7	1,150	5	24	2	0
8	1,125	7	_ <sup>b</sup>	_ <sup>b</sup>	_ <sup>b</sup>
9	450	3	15	0	0
10	450	4	7	1	2
11 <sup>a</sup>	450	3	17	3	6
12..19	200	2	_ <sup>b</sup>	_ <sup>b</sup>	_ <sup>b</sup>

<sup>a</sup> User did not fill in the questionnaire

<sup>b</sup> Incomplete or no tracking data

containing game design elements more than twice as often as the other users. Besides,  $U_{rank_1}$  participated in the online questionnaire.

Based on these interactions, we categorize the users into four different types: The Top 3 users, a midfield (positions 4–8 in the leaderboard), users who use the system for a very short time only (positions 9–11 in the leaderboard) and Users 12–19 who logged into the system only once.

One question is how these users interacted with the system in detail. Table 9.5 shows the interaction of all users at different days of the experiment. As expected, the most interactions were recorded in the first few days of the experiment. Everyday, the leaderboard was the most visited gamified page of the system. During the weekend, i.e., Days 4 and 5 of the experiment, no direct interaction with the gamification elements was recorded. The overall number of page requests declines over the course of the experiment, suggesting that the overall interest in the system declined as well.

### 9.3.4 Comparison to Questionnaire

After providing an overview of the system usage before and after the system got gamified, we discuss in this section the user interaction with respect to their feedback in the online questionnaire, hence addressing the research question of whether the perceived role of gamification is reflected by actual usage.

Table 9.6 shows the mean average answers of all users who participated in the online questionnaire. In the first rows of the table, we segment this group in two parts: those users who logged into our system at least once (Group A, where # login > 0)

**Table 9.5** Bookmark contributions and page request over time

Day	# Bookmarks		# Requests all pages and element specific pages			
	Create/Copy	Recommend	All pages	Leaderboard	My points	My badges
1 (Wed)	13	1	371	32	19	9
2 (Thu)	25	1	135	26	1	0
3 (Fri)	11	6	126	12	3	4
4–5 (Sat/Sun)	0	0	7	0	0	0
6 (Mon)	4	3	72	16	0	0
7 (Tue)	4	0	58	10	3	1
8 (Wed)	9	0	32	4	3	0

**Table 9.6** Mean average answers in online questionnaire of users who participated only in the questionnaire (Group B) and users who, in addition, logged in at least once in the gamified bookmarking system (Group A)

	Group A (#login > 0)	Group B (#login = 0)
$Q_1$ (contribution)	<b>3.00</b>	2.50
$Q_2$ (familiarity)	<b>3.07</b>	2.56
$Q_3$ (motivation)	<b>3.29</b>	2.94
$Q_4$ (positive)	<b>3.64</b>	3.11
$Q_5$ (negative)	2.86	<b>3.20</b>

Higher value indicates higher assessment, higher frequency, and stronger agreement

and those who did *not* log in (Group B, where # login = 0). As expected, members of Group A reported a higher familiarity (3.07 versus 2.56) with gamification than Group B. Further, they also (on average) stated a higher content contribution (3.00 versus 2.50) to online systems, stated that gamification can result in higher motivation (3.29 versus 2.94) and believed more in the positive effect of gamification (3.64 versus 3.11) than their colleagues from Group B. Beyond that, the members of Group B had (on average) a stronger opinion about the negative effects than their colleagues from Group A. This seems to indicate that the employees who have a rather positive impression of gamification are also more likely to use such a system at least once.

In order to further study whether this positive attitude is also reflected in the users' constant use of the system, i.e., addressing our second research question, we split Group A further into two subgroups: the top 8 users (according to the leaderboard) and the remaining 9 users who logged in at least once. Their corresponding answers are shown in Table 9.7. Surprisingly, the Top 8 users are more aware of negative effects caused by gamification than the rest. Similar to Table 9.6, we can observe a higher frequency, higher assessment and stronger agreement by those employees who were more active on the gamified system, i.e., those users who ended up on higher positions in the monthly leaderboard. This indicates that the perceived role

**Table 9.7** Mean average answers in online questionnaire of top 8 users (Group  $A_1$ ) of the leaderboard and users on position 9–19 (Group  $A_2$ )

	Group $A_1$ (rank $\leq 8$ )	Group $A_2$ (rank $> 8$ )
$Q_1$ (contribution)	<b>3.14</b>	2.86
$Q_2$ (familiarity)	<b>3.14</b>	3.00
$Q_3$ (motivation)	<b>3.71</b>	2.86
$Q_4$ (positive)	<b>3.71</b>	3.57
$Q_5$ (negative)	<b>3.14</b>	2.57

Higher value indicates higher assessment, higher frequency and stronger agreement

of gamification is indeed reflected by actual user interaction with a gamified system, hence answering our second research question.

### 9.3.5 Conclusion

In this section, we studied the perceived and actual role of gamification in a workplace environment. We focused on two questions.

First, we were interested to know whether employees perceive gamification as positive or negative factor in an enterprise. Therefore, we distributed an online questionnaire among members of a large research institute where participants were asked to judge and respond to different statements and questionnaires on a Five-Point Likert scale. Their responses indicate that although some employees were already familiar with the idea of gamification and are convinced that it can have a positive effect on their work, nevertheless, a majority of participants stated that they are not convinced that it can serve as intrinsic motivation for themselves.

In our second research question, we were interested to evaluate whether this perceived role of gamification that is reported by the participants of the online questionnaire matches their actual behavior when using a gamified system. Therefore, we gamified an existing enterprise bookmarking system and introduced it in the same research institute. After 1 week, we analyzed the online questionnaire based on the users’ interaction with this system. We observed that the employees who showed a positive tendency toward gamification also interacted more with the gamified system. We conclude that there is a relationship between the perceived and the actual role of gamification principles in a workplace environment.

As mentioned in the previous section, when designing games, one method to approach this individuality is to regard well-known player typologies [14] to group similar player types, and to design gamification addressing all these player types. The most common technique to find out the user types is the use of questionnaires and interviews. However, this approach is associated with high efforts. Given the bias effect caused by questionnaires, we argue that it is hard to conclude on users’ actual

behavior in a gamified environment [22]. Recent studies also indicate that the effect of game design elements can change over time, which can end up with lower effects in the long term [8, 10, 12]. In the worst case, positive effects might only be caused by the novelty effect. In the next section, we outline how player types could be identified automatically, hence reducing the risk of triggering negative gamification effects.

## 9.4 Towards the Automated Identification of Play-Personas

Existing gamification definitions pursuing the increase of user experience [6] and overall value [15] indicate that the application of gamification is goal oriented. Therefore, we usually look at gamification as the necessity to *maximize an overall goal*. However, the rich variety and individuality of the users results in different behaviors, preferences, and motivating factors (e.g., [11, 17, 35–37]). In the worst case, negative effects can occur when applying gamification, as observed by Hamari et al. [13] and Mosca [25]. Hence, for successful gamification, several factors need to be considered, which makes the design process difficult and expensive. Therefore, our extended look at gamification is the necessity to *maximize an overall goal with respect to the individuality of users*.

In this section we propose a new approach for gamification based on the automatic detection of play-personas. Dixon et al. [7] consider play-personas “as a useful tool that can be used to put player type research into practice as part of the design process of gamified systems.” In order to automatically determine different personas, we need to reduce the effort to determine relevant player types for implementing gamification. Why not skip the determination of player types and directly suggest game design elements? Trying to achieve this with questionnaires and interviews can of course increase the design effort. However, what if a formula or tool that helps to select such game design elements based on experiences learned from user interaction data over time can be used instead? Under the assumptions that (i) gamification consists of various types of users that experience game design elements differently; and (ii) gamification is deployed in order to achieve some goal in the broadest sense, we pose the gamification design problem as that of assigning each user (at least) one game design element that maximizes their expected contribution in order to achieve that goal.

We suggest matrix factorization to create a generic model based on user interaction data as a suitable methodology which could help for the selection of most fitting game design elements. Parts of the treatment are based on [30, 33]. The hypothesis is that predictive models as intelligent tools for supporting users in decision-making may also have potential to support the design process in gamification. We argue that this not only reduces the design effort, but also provides a better selection of game design elements since this kind of selection would not only be based on how users perceive gamification [22] but also on their actual interaction with game design elements. We are convinced that such data-centric tool can support the design process substantially.

### 9.4.1 A General Model of Gamification

We suggest a general model of gamification consisting of the following four components:

- A task  $T$  that needs to be performed.
- A set of game design elements  $g \in \mathcal{G}$ .
- A set of users  $u \in \mathcal{U}$  processing task  $T$  enhanced by  $\mathcal{G}$ .
- A task-dependent ground truth

$$f_* : \mathcal{U} \rightarrow \mathcal{G}.$$

- A function class  $\mathcal{F}$  consisting of functions of the form

$$f : \mathcal{U} \rightarrow \mathcal{G}.$$

The gamification design problem is the problem of selecting a function  $f \in \mathcal{F}$  that best approximates the supervisor  $f_*$ .

The ground truth  $f_*$  is a function that assigns each user  $u$  a game design element  $g$  that maximizes the expected contribution of  $u$  to achieve a prespecified goal. For users that best perform without any of the game design elements contained in  $\mathcal{G}$  we include a distinguished symbol  $\varepsilon$  denoting the absence of any design element.

Typically, the ground truth is unknown for most users and therefore needs to be approximated by a function from some function class  $\mathcal{F}$  based on a small subset

$$\mathcal{Z} = \{(u_1, g_1), \dots, (u_n, g_n)\} \subseteq \mathcal{U} \times \mathcal{G}$$

of training examples. The training set  $\mathcal{Z}$  consists of  $n$  users  $u_i$  with corresponding design elements  $g_i = f_*(u_i)$  for which the ground truth is known.

Note that we do not want to memorize the training examples but rather find (learn) a function  $f \in \mathcal{F}$  that predicts the best fitting design elements for new users not considered in  $\mathcal{Z}$ .

### 9.4.2 Learning Problem

There are different ways to select (learn) a function  $f$  from  $\mathcal{F}$  in order to approximate the ground truth  $f_*$ .

One approach describes users  $u$  by a feature vector  $\mathbf{x}_u$ . The components of  $\mathbf{x}_u$  measure different properties of that user such as, for example, click behavior, mouse movements, and other features. Then a classifier such as the support vector machine [5] is trained to learn a model that predicts the best fitting game design element for new users.

Here we consider a second approach based on user interaction with different game design elements. We measure the utility of a game design element  $g$  for user  $u$  in achieving task  $T$  by means of a utility function

$$f_U : \mathcal{U} \times \mathcal{G} \rightarrow \mathbb{R}, \quad (u, g) \mapsto s_{ug}.$$

The utility-scores  $s_{ug}$  capture to which extent each user  $u$  together with design element  $g$  contributes to some overall goal. Given a utility function  $f_U$ , we select a classifier  $f \in \mathcal{F}$  according to the rule

$$f(u) = g_u^* = \arg \max_{g \in \mathcal{G}} f_U(u, g).$$

Thus,  $f$  assigns user  $u$  a game design element  $g_u^*$  with maximum utility.

Table 9.8 provides an example of a utility function  $f_U$  shown in matrix form  $S = (s_{ug})$ . In this example, we would assign game design  $g_3$  to user *Ann*. For user *Bob* the maximum score of 5 is achieved for design elements  $g_1$  and  $g_2$ . In this case, we can pick either  $g_1$  or  $g_2$  as design element for *Bob*.

In practice, however, the matrix  $S$  is sparse for various reasons. For example, users might not be willing to explore all design elements and may quit using the system. Table 9.9 provides an example for the case of a sparse matrix  $S$  of utility-scores. In this scenario, we aim at learning  $f_U$  on the basis of  $n$  observations  $(u_1, g_1, s_1), \dots, (u_n, g_n, s_n) \in \mathcal{U} \times \mathcal{G} \times \mathbb{R}$  consisting of  $n$  users  $u_i$  together with corresponding game design elements  $g_i$  and utility-scores  $s_i$ .

The problem of gamification reduces to estimating a functional relationship

$$f : \mathcal{U} \times \mathcal{G} \rightarrow \mathbb{R}, \quad (u, g) \mapsto \hat{s}_{ug}$$

that *best* predicts the utility-score  $s_{ug}$  of design element  $g$  for user  $u$  by means of  $f(u, g) = \hat{s}_{ug}$ . To clarify what we mean by *best*, we introduce the notion of loss function. A loss function  $\ell(\hat{s}, s)$  measures the cost for predicting  $\hat{s}$  when the true utility-score is  $s$ . A common choice for a loss function is the squared error loss defined as

**Table 9.8** Utility-scores for six users and seven game design elements

	g1	g2	g3	g4	g5	g6	g7
Ann	0	2	5	3	1	4	4
Bob	5	5	3	3	4	1	0
Col	1	3	4	2	3	3	5
Don	5	4	2	4	3	3	2
Elk	5	5	4	4	3	0	1
Flo	2	1	4	4	3	5	4

Scores are values from  $\{0, 1, \dots, 5\}$ . Higher scores indicate higher utility and vice versa

**Table 9.9** Sparse user-design matrix of utility-scores consisting of six users and seven game design elements

	g1	g2	g3	g4	g5	g6	g7
Ann	–	2	–	–	1	–	4
Bob	–	–	–	–	4	1	–
Col	–	3	4	2	–	–	–
Don	–	–	–	4	3	–	2
Elk	5	5	–	–	–	0	–
Flo	2	–	4	4	–	5	–

Higher scores indicate higher utility and vice versa

$$\ell(\hat{s}, s) = (\hat{s} - s)^2.$$

Our goal is to find a function that minimizes the expected loss

$$E[f] = \int \ell(f(u, g), s_{ug}) dP(u, g, s_{ug})$$

where  $P(u, g, s)$  denotes the joint probability distribution on  $\mathcal{U} \times \mathcal{G} \times \mathbb{R}$ .

Suppose that we know a function (ground truth)  $f_*$  that minimizes the expected loss  $E[f]$ . Then we are in a similar situation as in the above scenario, where each user has explored all game design elements. The complete user-design matrix  $\mathbf{S} = (s_{ug})$  has elements of the form

$$s_{ug} = f_*(u, g).$$

We can assign each user  $u$  a game design element  $g_u^*$  according to the following rule

$$g_u^* = \arg \max_g f_*(u, g).$$

In practice, we neither know  $f_*$  nor the joint probability distribution  $P(u, g, s_{ug})$ . Therefore, we cannot find a minimum  $f_*$  of  $E[f]$  directly. Instead, we try to approximate  $f_*$  by a function  $\hat{f}_*$  that minimizes the empirical loss

$$\hat{E}[f] = \frac{1}{n} \sum_{i=1}^n \ell(f(u, g), s_{ug}).$$

on the basis of a sample of observed data

$$(u_1, g_1, s_1), \dots, (u_n, g_n, s_n).$$

The sparse user-design matrix shown in Table 9.9 is an example of a sample of observed data.

According to the empirical risk minimization principle, this approach is statistically consistent [33], meaning that the approximation  $\hat{f}_*$  converges to the true minimum  $f_*$  with increasing amount of data. The learning problem consists in predicting the missing values.

This setting reduces the gamification design problem of finding the best design element for each user to the problem of regression learning for which a plethora of powerful mathematical methods are available.

### 9.4.3 Matrix Completion

The gamification design problem as proposed in this section can be regarded as a special case of a recommendation problem [16] for which matrix factorization constitutes a state-of-the-art solution [3, 19, 20].

Matrix factorization characterizes users and game design elements by  $k$  factors (properties) inferred from the utility-score patterns hidden in the user-design matrix  $S = (s_{ug})$ . Users  $u$  and game design elements  $g$  are associated with vectors  $\mathbf{x}_u \in \mathbb{R}^k$  and  $\mathbf{y}_g \in \mathbb{R}^k$ , respectively. The  $k$  elements in  $\mathbf{y}_g$  measure to which extent design element  $g$  possesses these factors. Similarly, the elements in  $\mathbf{x}_u$  measure to which extent user  $u$  prefers these factors. High correspondence between factors of user  $u$  and factors of design element  $g$  indicate high utility. Correspondence between user and design factors is modeled as inner product such that

$$s_{ug} \approx \mathbf{x}_u^T \mathbf{y}_g \quad (9.1)$$

for all known utility-scores  $s_{ug}$ . In matrix notation, Eq. 9.1 takes the form

$$S \approx X \cdot Y,$$

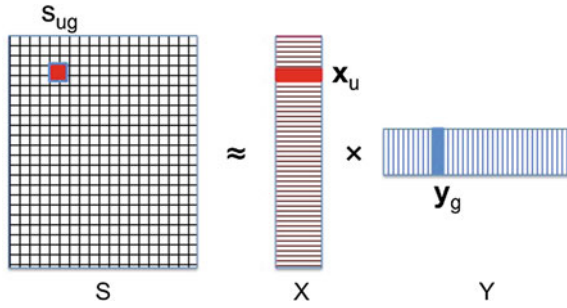
where  $X$  is the user matrix and  $Y$  is the game design element matrix. The rows  $\mathbf{x}_u^T$  of  $X$  and the columns  $\mathbf{y}_g$  of  $Y$  describe the users  $u$  and design elements  $g$ , respectively. Figure 9.7 illustrates how the user-design matrix  $S$  is factorized by low-rank matrices  $X$  and  $Y$ .

Figure 9.8 shows a fictitious example of how the six users and seven game design elements from Table 9.9 are associated to vectors from the two-dimensional vector space  $\mathbb{R}^2$ . The latent factors are inferred from the utility-score patterns hidden in the user-design matrix  $S$ . In this example, the two discovered factors refer to the preferences according to the player typology proposed by Bartle [2]. In practice, however, there may be additional ( $k > 2$ ) or different factors, which may not be interpretable for humans.

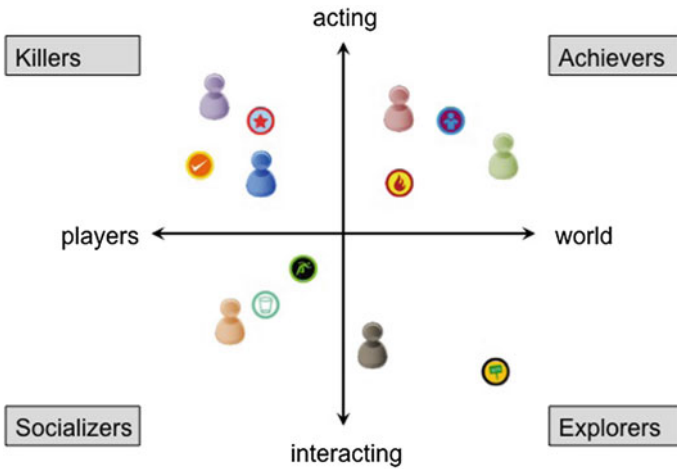
After all users and all game design elements have been embedded into the joint latent factor space  $\mathbb{R}^k$ , missing values  $s_{ug}$  of the sparse matrix  $S$  can be predicted in a straight forward way by

$$\hat{s}_{ug} = \mathbf{x}_u^T \mathbf{y}_g$$





**Fig. 9.7** Matrix factorization. Approximate the user-design matrix  $S$  by low-rank matrices  $X$  and  $Y$



**Fig. 9.8** A simplified illustration of the latent factor space generated by matrix factorization. The latent factors refer to the preferences indicated by the  $x$ - and  $y$ -axis. The six users and seven design elements of Table 9.8 are embedded into the factor space. According to Bartle’s player typology [2], users fall into one of the four categories *achiever*, *explorer*, *socializer*, and *killer*. Similarly, the features of the design elements refer to characteristics of the user categories

in order to complete matrix  $S$ .

To learn the embeddings into the factor space  $\mathbb{R}^k$ , we need to solve the following basic problem:

$$(P) : \min_{\mathbf{x}_*, \mathbf{y}_*} \sum_{(u, g) \in \mathcal{P}} (s_{ug} - \mathbf{x}_u^T \mathbf{y}_g)^2,$$

where  $\mathcal{P}$  is the set of all pairs  $(u, g)$  for which  $s_{ug}$  is known.

## 9.5 Conclusion and Outlook

In this chapter, we address the challenge of applying gamification in a workplace environment from two directions. First, we present a user study that focused on determining users' perception of gamification and their actual interaction with a gamified system. We conclude from this initial study that there is a relationship between the perceived and the actual role of gamification principles in a workplace environment.

Under the assumption that different users experience the same game design elements differently, we then focus on automatically identifying play-personas. This will allow us to create gamified systems that adapt the application of gamification elements based on users' types. In this context, we define the gamification problem as the problem of assigning each user a game design element such that their expected contribution to achieve some pre-specified goal is maximized.

One way to assign design elements to users is by means of customer segmentation. In marketing theory, segmentation aims at identifying customer groups in order to better match the needs and wants of customers. For games these customer segments correspond to different player types based on character theory. Once a user is classified into a customer segment, an appropriate design element for that segment is selected and assigned to that user. The hardest part of this approach is to design categories that correspond to various dimensions describing characteristic features of users such as the multiple motivations of varying degrees existing simultaneously across users and user types.

In order to avoid assignments of design elements to users via the indirection of customer segments and user types from marketing and character theory, respectively, we aim at learning a predictive model based on statistical principles that directly classify users to game design elements. Based on user interaction with game design elements, we suggest to solve the learning problem by means of matrix factorization. The latent factors discovered by a matrix factorization model may be interpreted as characteristic properties of game design elements. User factors describe to which extent a user prefers such characteristic features. Thus, the latent factors can be regarded as a computerized alternative to the aforementioned customer segments and user types.

Aiming to keep the gamification model simple, we ignore time dynamics of user preferences leaving this issue open for future research. In addition, learning classifiers based on user behavior characteristics is a second issue for future research. The main challenges consist in constructing a useful utility function when using matrix factorization and generating useful behavior features when learning classifiers. Due to lack of publicly available data, empirical evaluations are currently not possible. Therefore, this contribution aims at directing the design process of gamification to a more principled way based on statistical grounds.

**Acknowledgments** The research leading to these results was performed in the CrowdRec project, which has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n° 610594.

## References

1. A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, Steering user behavior with badges, in *Proceedings WWW* (Rio de Janeiro, Brazil, 2013), pp. 95–106
2. R. Bartle, Hearts, clubs, diamonds, spades: players who suit MUDs. *J. Virtual Environ.* (1996)
3. E.J. Candès, B. Recht, Exact matrix completion via convex optimization. *Found. Comput. Math.* **9**(6), 717–772 (2009)
4. L.-T. Cheng, S. Shami, C. Dugan, M. Muller, J. DiMicco, J. Patterson, S. Rohal, A. Sempere, W. Geyer, Finding moments of play at work, in *Workshop on Gamification: Using Game Design Elements in Non-Gaming Contexts*, pp. 2–5 (2011)
5. C. Cortes, V. Vapnik, Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
6. S. Deterding, D. Dixon, R. Khaled, L. Nacke, From game design elements to gamefulness: defining Gamification, in *Proceedings of the International Academic MindTrek Conference*, pp. 9–15 (2011)
7. D. Dixon, Player types and gamification, in *Workshop on Gamification at CHI2011*, pp. 12–15 (2011)
8. R. Farzan, J.M. DiMicco, When the experiment is over: deploying an incentive system to all the users, in *Persuasive Technology* (ACM, 2008)
9. R. Farzan, J.M. DiMicco, B. Brownholtz, Spreading the honey: a system for maintaining an online community, in *Proceedings of the ACM GROUP'09*, pp. 31–40 (2009)
10. R. Farzan, J.M. DiMicco, D.R. Millen, Results from deploying a participation incentive mechanism within the enterprise, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 563–572 (2008)
11. J. Hamari, J. Koivisto, Social motivations to use Gamification: an empirical study of gamifying exercise, in *Proceedings of the ECIS'13*, pp. 1–12 (2013)
12. J. Hamari, Transforming homo economicus into homo Ludens: a field experiment on Gamification in a utilitarian peer-to-peer trading service. *Electron. Commer. Res. Appl.* **12**(4), 236–245 (2013)
13. J. Hamari, J. Koivisto, H. Sarsa, Does Gamification work?—A literature review of empirical studies on Gamification, in *Proceedings of the 47th Hawaii International Conference on System Sciences* (2014)
14. J. Hamari, J. Tuunanen, Player types: a metasynthesis, in *Transactions of the Digital Games Research Association* (2014)
15. K. Huotari, J. Hamari, Defining Gamification: a service marketing perspective, in *Proceedings of the 16th International Academic MindTrek Conference*, pp. 17–22 (2012)
16. P.B. Kantor, L. Rokach, F. Ricci, B. Shapira, in *Recommender Systems Handbook* (Springer, New York, 2011)
17. R. Khaled, It's not just whether you win or lose: thoughts on Gamification and culture. in *Workshop on Gamification: Using Game Design Elements in Non-Gaming Contexts*, pp. 1–4 (2011)
18. F. Khatib, F. DiMaio, Foldit Contenders Group, Foldit Void Crushers Group, S. Cooper, M. Kazmierczyk, M. Gilski, S. Krzywda, H. Zabranska, I. Pichova, J. Thompson, Z. Popović, M. Jaskolski, D. Baker, Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nat. Struct. Mol. Biol.* **18**, 1175 – 1177 (2011)
19. Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems. *IEEE Comput.* **42**(8), 42–49 (2009)
20. D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization. *Nature* **401**(6755), 788–91 (1999)
21. J.J. Lee, J. Hammer, Gamification in education: what, how, why bother? *Acad. Exch. Q.* **15**(2), 2 (2011)
22. M. Meder, T. Plumbaum, F. Hopfgartner, Perceived and actual role of Gamification principles, in *First Workshop on Crowdsourcing and Gamification in the Cloud (CGCloud) in Conjunction with the 6th IEEE/ACM International Conference on Utility and Cloud Computing* (2013)

23. M. Meder, T. Plumbaum, F. Hopfgartner, Daiknow: a gamified enterprise bookmarking system. in *Proceedings of the 36th European Conference on Information Retrieval, ECIR'14* (Springer, 2014), pp. 759–762
24. M. Moore, C. Dugan, M. Muller, D.R. Millen, W. Geyer, B. Brownholtz, The Dogear game: a social bookmark recommender system, in *Proceedings of the ACM GROUP'07*, pp. 387–390 (2007)
25. I. Mosca, +10! Gamification and de Gamification. *GIAMIE* **1**(1) (2012)
26. J. Nielsen, Participation inequality: encouraging more users to contribute (2006)
27. S. Nikkila, S. Lin, H. Sundaram, A. Kelliher, Playing in taskville: designing a social game for the workplace. in *Workshop on Gamification: Using Game Design Elements in Non-Gaming Contexts*, pp. 1–4 (2011)
28. F. Ortega, J.M. Gonzalez-Barahona, G. Robles, On the inequality of contributions to Wikipedia, in *Proceedings of the International Conference on System Sciences*, pp. 304–304 (2008)
29. D. Robinson, A preliminary taxonomy of Gamification elements for varying anticipated commitment, in *Proceedings of the ACM CHI 2013 Workshop on Designing Gamification: Creating Gameful and Playful Experiences* (2013)
30. A. Said, B. Jain, S. Narr, T. Plumbaum, Users and noise: the magic barrier of recommender systems, in *User Modeling, Adaptation, and Personalization* (Springer, Heidelberg, 2012)
31. O. Stewart, D. Lubensky, J.M. Huerta, Crowdsourcing participation inequality: a SCOUT model for the enterprise domain, in *Proceedings of the ACM SIGKDD Workshop on Human Computation* (2010), pp. 30–33
32. J. Thom, D. Millen, J. DiMicco, Removing Gamification from an enterprise SNS, in *Proceedings of the CSCW'12*, pp. 1067–1070 (2012)
33. V. Vapnik, in *The Nature of Statistical Learning Theory* (Springer, New York, 2000)
34. L. von Ahn, L. Dabbish, Labeling images with a computer game, in *Proceedings of the CHI'04*, vol. 6(1), pp. 319–326 (2004)
35. J. Yang, M.R. Morris, J. Teevan, L.A. Adamic, M.S. Ackerman, One Microsoft way. Culture matters: a survey study of social Q and A behavior, in *International AAAI Conference on Weblogs and Social Media*, pp. 409–416 (2011)
36. N. Yee, Motivations for play in online games. *J. CyberPsychol. Behav.* **9**, 772–775 (2007)
37. N. Yee, N. Ducheneaut, L. Nelson, Online gaming motivations scale: development and validation, in *Proceedings of the CHI'12*, pp. 2803–2806 (2012)

# Part III

## Sensor-Based Knowledge Acquisition and Signal Processing Services

### Overview

Traditionally, most information that was recorded in history was conveyed in textual form. Think, for example, of the works of great philosophers, novelists, or historians who relied on the written word to express their view points, to express their creativity, or to report what they witnessed. Without any doubt, information in human-readable textual form is the most commonly used means of sharing information. At the same time, however, information is often shared in non-textual form. Early art work such as cave paintings dating back over 30,000 years indicate that mankind has always relied on many other forms to convey information. With the invention of photography and photographic filmmaking, the share of non-textual documents that was used to distribute information increased significantly. In the early days of the twentieth century, when photography started to take off, the adage “a picture is worth a thousand words” was used to express the notion that complex ideas can easily be conveyed in a still visual image. From a computational point of view, interpreting non-textual material such as images or films is a nontrivial task. The main challenge is to bridge the so-called semantic gap, i.e., the difference between humans’ interpretation of the information that is depicted in such material, and the representation of this information that can be processed by a machine. The challenge of interpreting non-textual data even increases when considering non-visual material such as sensor readings. With the introduction of sensors and sensor systems, an ever-increasing amount of data is created that conveys detailed information in the form of digital or optical signals. Analyzing this data for the provision of information services requires advanced methods in the fields of data mining and machine learning. In the final part of this book, we present four different scenarios where data in non-textual form is created and present approaches to exploit this data.

The first scenario, presented in Chap. 10, addresses sustainable energy consumption in smart home environments. More precisely, Spiegel presents a framework for heating control and scheduling that considers occupancy information and, therefore, allows for the reduction of residential energy consumption. His work demonstrates how to use aggregated energy signals, measured by a smart meter, to

identify the habits of the residents in terms of presence. He proposes to employ energy disaggregation techniques to make inference about the use of certain household appliances, which indicate the physical presence of the occupants. This approach is also referred to as non-intrusive load monitoring, which has the benefit that it refrains from using an additional sensor infrastructure.

Focusing on personal media consumption, Acar et al. present in Chap. 11 an approach for identifying a certain type of pattern present in Hollywood movies or user generated videos. This pattern is “violence.” These movies or videos contain two modalities (audio and visual), each modality being directed to a different sense of the media consumer (hearing and seeing), therefore allowing an “immersion,” which does not actually exist, or is limited to textual information. Detecting violent content in movies and videos is one application which neatly illustrates the meaning of “bridging the semantic gap.” Acar et al. achieve this by extracting meaningful features from the data, and by classifying those features using advanced machine learning techniques. They also present a user interface designed to allow the consumer to browse data and search for “violent” scenes.

The scenario that is outlined in Chap. 12 appears in the context of the automotive industry, where the objective is to optimize vehicle engines with regard to exhaust emission. Spiegel presents a data mining approach that was developed in cooperation with researchers and engineers from one of the leading car manufacturers, who aim to run emission simulations based on operational profiles that characterize recurring driving behavior. In order to obtain real-life operational profiles, the automotive engineers collect sensor data from test-drives for various combinations of driver, vehicle, and route. Such measurements can also be considered as high-dimensional time series, where each dimension represents the progression of a certain physical quantity, such as the engine temperature, during car drive. Spiegel’s proposed approach is able to identify time series representatives that best comprehend the recurring temporal patterns contained in a corresponding dataset. He applies this approach to determine operation profiles that comprise frequently recurring driving behavior patterns, but his introduced model can also be used for time series datasets from other domains.

The final scenario of this section, presented in Chap. 13, focuses on the related topic of traffic optimization. The ever-increasing urbanization has significantly aggravated the traffic situation in megacities. Common travel habits, such as using a vehicle, are challenged by factors like severe congestion, insufficient parking availabilities, or present fuel prices. Furthermore, growing traffic entails an increased level of noise and greenhouse gases and thus affects residents even more. Acar et al. present an approach to utilize means of transportation in a more effective and sustainable fashion in order to increase the quality of life in cities and to contribute to global environmental objectives. They describe a travel assistance system that proposes intermodal traveling options which are tailored to drivers’ needs. Different information channels are integrated in the system. One of these channels is information derived from video analysis.

# Chapter 10

## Optimization of In-House Energy Demand

Stephan Spiegel

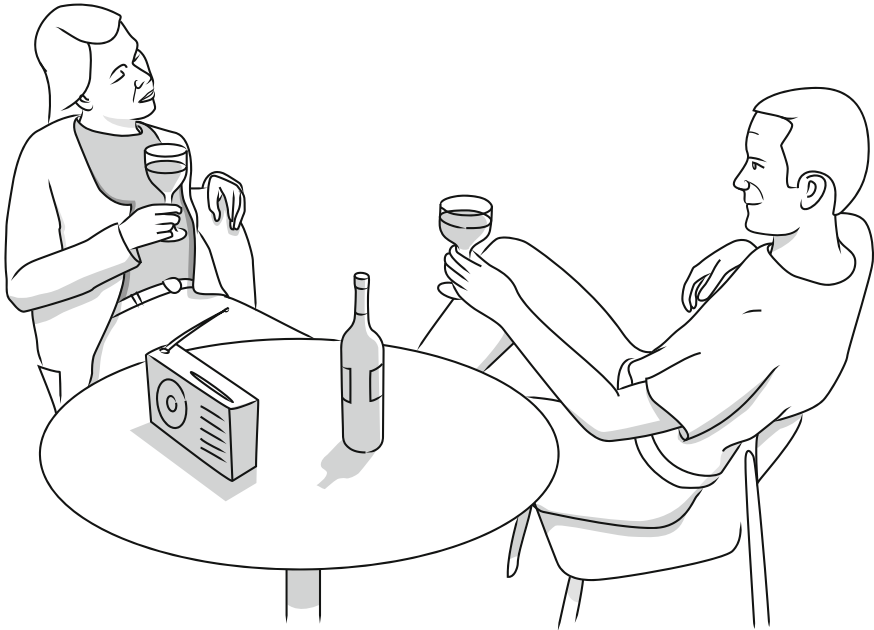
**Abstract** Heating control is of particular importance, since heating accounts for the biggest amount of total residential energy consumption. Smart heating strategies allow to reduce such energy consumption by automatically turning off the heating when the occupants are sleeping or away from home. The present context or occupancy state of a household can be deduced from the appliances that are currently in use. In this chapter, we investigate energy disaggregation techniques to infer appliance states from an aggregated energy signal measured by a smart meter. Since most household devices have predictable energy consumption, we propose to use the changes in aggregated energy consumption as features for the appliance/occupancy state classification task. We evaluate our approach on real-life energy consumption data from several households, compare the classification accuracy of various machine learning techniques, and explain how to use the inferred appliance states to optimize heating schedules.

### Sustainable Energy: The Early Adopter Scenario

Steven and his wife Suzanne love to relax in their garden behind the house on warm summer evenings. Usually, they enjoy dinner with a glass of wine on their garden terrace, talking about the kids, Clara and Carl, or listening to the latest news from the local radio station. Yesterday evening there was a radio broadcast about the advance of smart meters and their potential to reduce the energy consumption in residential homes. Suzanne was excited about the idea of saving energy by themselves, especially since the power market has continuously raised prices over the last couple of years. Steven has always been fond of Suzanne's commitment to sustainable living and suggested to contact Ralph, an old schoolmate of him, who runs his own little business in the IT sector and also is a trained electrician.

---

S. Spiegel (✉)  
Technische Universität Berlin, Berlin, Germany  
e-mail: stephan.spiegel@dai-labor.de



On the next day Steven called Ralph and was surprised to learn that heating accounts for the biggest amount of total residential energy consumption. Ralph told him that some households save up to 30% energy by installing an advanced heating control, which allows to setup heating schedules. But Steven argued that it would be difficult to set up a heating schedule for his family, since the timetable of his kids is irregular, his wife's company allows her to telecommute up to four times per month, and he sometimes has to do shift work. Ralph contemplated Steven's arguments for a while and eventually suggested to install a presence sensing infrastructure, which can detect movements and forwards presence states to the heating control. However, after weighing the cost for such a sensor infrastructure against the potential energy savings, they both abandoned this course of action. Having discussed the advantages and limitations of available solutions, Steven told Ralph about the radio broadcast which had advertised smart metering technology and came up with the idea of using energy consumption measurements for the detection of presence states. Ralph thought it was a brilliant idea and said that he would do some research on smart metering technology.

A week had passed when Ralph finally called back and told Steven that he had come up with an advantageous solution for automatically detecting presence states from smart meter measurements. Although Steven had never before heard of something like energy disaggregation, he could understand that Ralph planned to implement an algorithm that utilizes energy consumption readings to infer appliance use, which in turn could be used to deduce presence states. Ralph explained that the use of an individual electronic device causes a specific energy consumption profile, which can be used to deduce certain activities in a household. According to Ralph,



the task of the energy disaggregation algorithm would be to recognize and classify the individual appliances consumption profiles within the aggregated energy signal coming from the smart meter. Ralph assured Steven that this solution does not require additional sensor infrastructure and enables recommendations for optimized heating schedules based on automatically deduced presence states.

Steven was very pleased with Ralph's proposal and informed his wife Suzanne about the good news. In about three months they would have a new energy-aware heating control system, which will help them to cut future energy bills without any effort. Suzanne was relieved by the idea that she will never again have to worry about the kids going out of the house without turning the heating off or her coming home early in winter finding the living room in a state of severe cold. Steven told her that it would even be possible to access and monitor the heating control from remote via her smart phone, which may be convenient when they are on longer vacation with their kids. As usual both Steven and Suzanne were enjoying dinner together on their garden terrace, talking about new family projects and listening to the local radio station, which this time was broadcasting a debate about environmentally friendly vehicles and new ways of transportation.

## 10.1 Introduction

The main goal of our study is to provide a framework for heating control and scheduling which considers the occupancy states of residential homes. Since most solutions for occupancy state identification involve complex sensor infrastructure and costly hardware which cause high usage barrier [3, 9, 14, 17], we aim to use given information from available electricity smart meters. We propose to employ energy disaggregation to infer appliance usage which is, as we will show, beneficial to occupancy state identification. In the following, we briefly introduce the value of appliance usage information, before we explain how we use this information for the purpose of heating control.

In the context of domestic environments, consumers vastly underestimate the energy used for heating and overestimate the energy used for appliances that replace manual labor tasks [4]. Numerous studies have identified that consumers get a better understanding of their energy use by clear, concise, and direct feedback about appliance-specific consumption information [13, 19, 23].

In regard to power grid operators and power suppliers, knowledge about the energy consumption on appliance level is critical to the development of power system planning, load forecasting, billing procedures, and pricing models [4, 19]. In addition, the identification of electric appliances in domestic environments is important, because the increasing number of renewable energy sources in the power grid requires electric utilities to be able to quickly react to changes in supply and demand [18].

The growing need for accurate and specific information about domestic energy consumption on device level has led to numerous studies on appliance load monitoring [1, 4, 10, 22, 23]. Existing solutions for appliance load monitoring can

be classified into two primary techniques [4, 21]: distributed direct sensing and single-point sensing.

Distributed direct sensing typically requires a current sensor to be installed in-line with every device and is therefore often referred to as intrusive load monitoring. Although intrusive load monitoring easily achieves a consumption breakdown, deploying a large number of sensors in the residential environment quickly leads to high cost and discouraging high usage barrier [21].

Single-point sensor systems are easier to deploy and are typically subsumed under the concept of nonintrusive load monitoring (NILM) [21]. Energy disaggregation is the task of using an aggregated energy signal, such as that coming from a single-point sensor or rather whole-home power monitor, to make inferences about the different loads of individual appliances [10]. However, single-point sensor systems require knowledge about the household devices and their electrical characteristics [21]. The challenges in energy disaggregation are mainly due to appliances with similar energy consumption, appliances with multiple settings, parallel appliance activity, and environmental noise [19]. Recent studies [8, 10–13, 20] have shown that machine learning techniques represent a suitable solution to recognize appliances in such dynamic and unpredictable environments.

In this work, we consider energy disaggregation techniques to derive occupancy states from appliance usage data in order to use this information in smart heating control strategies [9]. Heating control is of particular importance, since heating accounts for the biggest amount of total residential energy consumption and recent studies have shown that up to 30% of the total energy can be saved by turning the heating off when the occupants are asleep or away [14]. Existing work on the inference of occupancy states in residential environments includes statistical classification of aggregated energy data [9], hot water usage [3] as well as human motion and activity [17]. Our own approach to infer occupancy states differs in that we consider appliance usage, which gives more detailed information about the present context in a household and the devices which suggest user activity. Furthermore, our proposed framework does not require any additional infrastructure, and, therefore, is more likely to be accepted by residents.

For the evaluation of our approach, we consider the REDD dataset [10], which consists of whole-home and device-specific electricity consumption for a number of real houses over the period of several month. In our experiments, we compare the performance of different models for the appliance/occupancy state classification task. We use cross-validation (training on all houses and leave-one-out for testing) to evaluate how well the different models generalize. Our results suggest that the Naive Bayes classifier is suitable for the prediction of occupancy/appliance states and fits the problem of real-time heating control.

The rest of the chapter is structured as follows. In Sect. 10.2, we give some background on recent advances in energy disaggregation. Section 10.3 introduces the formal notation of our appliance state classification task. Our proposed framework for heating control and scheduling by means of energy disaggregation techniques is described in Sect. 10.4. The experimental design and results on our approach are presented in Sect. 10.5. A practical application for our approach, named SOE, is

demonstrated in Sect. 10.6. Eventually, we conclude our study and give an outlook on future work in Sect. 10.7.

## 10.2 Background

Energy disaggregation, also referred to as nonintrusive load monitoring, is the task of using an aggregated energy signal, such as that coming from a whole-home power monitor, to make inferences about the different individual loads of the system [10]. This approach is seen as an intermediate between existing electricity meters (which merely record whole-home power usage) and fully energy-aware home appliance networks, where each individual device reports its own consumption [18].

For a thorough evaluation of various energy disaggregation mechanisms under real-world conditions, a comprehensive collection of power consumption data is needed [18]. Most approaches to energy disaggregation have been supervised, in that the model is trained on individual device power signals [23]. The vast majority of supervised disaggregation approaches have evaluated the trained models on the same devices but in new conditions [1].

Research on energy disaggregation has been encouraged by publicly available datasets such as REDD [10], which contains information about the power consumption of several different homes on device level, and, therefore, allows cross-validation for individual appliances. Experiments on the REDD dataset have shown that the Factorial Hidden Markov Model (FHMM) is able to disaggregate the power data reasonably well [10]. In that case, the disaggregation task is framed as an inference problem and the performance of energy disaggregation is evaluated considering the percentage of energy correctly classified.

Although FHMMs have shown to be a powerful tool [5] for learning probabilistic models of multivariate time series, the combinatorial nature of distributed state representation makes an exact algorithm for inferring the posterior probabilities of the hidden state variables intractable. Approximate inference can be carried out using Gibbs sampling or variational methods [5]. Recent work [8] on energy disaggregation presents different FHMM variants which incorporate additional features and better fit the probability distribution of the state occupancy durations of the appliances.

Another work [19] proposes Artificial Neural Networks (ANNs) for appliance recognition, because they (i) do not require prior understanding of appliance behavior, (ii) are capable of handling multiple states, and (iii) are able to learn while running. The results show that after training the ANN with generated appliance signatures, the proposed system is able to recognize the previously learned appliances with relatively high accuracy, even in demanding scenarios. To tune the ANN, the authors suggest to use the generated signatures to create a training dataset with all possible combinations of appliance activity. Comparing the disaggregation performance for different ANN algorithms, additional work [11] suggests to employ back-propagation rather than the radial-base-function.

In another study [21], the authors propose a disaggregation algorithm that consists of several consecutive steps including normalization, edge detection via thresholding and smoothing techniques, extraction of power-level and delta-level consumption, matching of known appliances from a signature database with extracted delta vectors, and labeling of recognized devices. The proposed system does not require setup or training, because the user is able to label appliance signatures via her smart phone. In that case, the appliance signatures are based on apparent, reactive, real, and distortion power measured by the smart meter.

The classification of household items based on their electricity usage profile over a fixed time interval is discussed in yet another study [13]. The authors consider the time series classification problem of identifying device types through daily or weekly demand profiles. The proposed approach concentrates on bespoke features such as mean, variance, kurtosis, skewness, slope, and run measures. The experiments show that classification using the bespoke features performs better than classification using the raw data. However, the nature of similarity captured strongly depends on the features extracted.

In a similar work [18], the authors present an appliance identification approach based on characteristic features of traces collected during the 24h of a day. The extracted features include temporal appliance behavior, power consumption levels, shape of the consumption, active phase statistics, and noise level characteristics. Each resulting feature vector is annotated by the actual device class and used to train the underlying model of the selected classifier. Among various tested classifiers, the Random Committee algorithm performs best in categorizing new and yet unseen feature vectors into one of the previously trained device types. Additional work [11] demonstrates that the solution from any single-feature, single-algorithm disaggregation approach could be combined under a committee decision mechanism to render the best solution.

Yet another work [20] presents a nonintrusive appliance load monitoring technique based on integer programming. Since the overall load current is expressed as a superposition of each current of the operating appliance, the monitoring problem can be formulated as an integer quadratic programming problem by expressing the operating conditions as integer variables. Besides that the proposed method does not require relearning when a new appliance is installed in the house, it is furthermore able to distinguish between different device modes and some-type appliances that operate simultaneously.

To monitor the states of multiple appliances via electricity consumption measurements, another work [12] introduces the Bayes filter approach, which computes the posterior distribution over the current state given all observations to date. Since the state transition of an appliance is a continuous process, the authors employ a sliding window to take the temporal factor into consideration and extract the past records of data to be features. The estimated states are represented as binary strings, where each bit denotes the on/off state of one individual appliance. According to the results, the Bayes filter outperforms the KNN, Naive Bayes, and SVM classifier.

Leveraging recent advances in device and appliance power supplies, another series of studies [4, 6] extends the energy disaggregation approach by using high-frequency

sampling of voltage noise, which provides an additional feature vector that can be used to distinguish more accurately between energy usage signatures. Appliances conduct a variety of noise voltage back onto the home's power wiring, yielding measurable noise signatures that are easily detectable using appropriate hardware. An important advantage of voltage noise signatures is that any electrical outlet inside the home can be used as a single installation point.

### 10.3 Notation

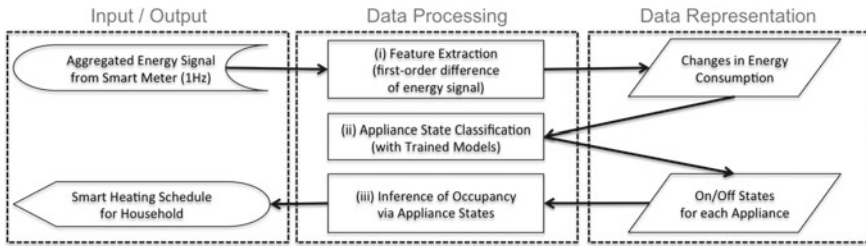
Since different devices tend to draw different amounts of power, which are consistent over time, total power is a reasonable feature to use for classification [4]. Most devices have predictable current consumption and can be categorized according to the magnitude of real/reactive power. Given a household with  $N$  devices, the power consumption of an individual appliance  $i \in \{1, \dots, N\}$  over a period of  $T$  time points can be expressed as:  $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \dots, y_T^{(i)}\}$ . Usually, we only observe the sum of all power outputs at each time:  $\bar{y}_t = \sum_{i=1}^n y_t^{(i)}$ , with  $t = 1, \dots, T$ .

Given the aggregated power signal, most research on energy disaggregation [1, 22, 23] aims at inferring the individual device consumption. Since we aim to infer the context or rather occupancy states in residential environments in order to optimize heating control, we are mainly interested in the ON/OFF states of individual appliances  $s_t^{(i)}$ , where  $s_t^{(i)} = 1$  if device  $i$  is turned "on" at time point  $t$ , and  $s_t^{(i)} = 0$  otherwise. The appliance state identification task can be framed as an inference problem. Given an aggregated power signal  $\bar{y}_1, \dots, \bar{y}_T$ , we intend to compute the posterior probability  $p(s_t^{(i)} | \bar{y}_t)$  of individual appliance states  $s_t^{(i)}$  for each device  $i = 1, \dots, N$  and each time point  $t = 1, \dots, T$ .

Due to the fact that the aggregated power signal is super-imposed and unnormalized, and, therefore, unsuitable for the appliance state identification, we consider the changes in power consumption as features, which can be derived by the first-order difference of the power signal  $\Delta y_t^{(i)} = y_t^{(i)} - y_{t-1}^{(i)}$  for  $t = 2, \dots, T$ . Thus the appliance state identification task could also be formulated as a classification problem, where a certain change in power consumption categorizes a device into either "ON" or "OFF" state.

### 10.4 Framework and Algorithms

Figure 10.1 shows a flowchart of our proposed framework for heating control and scheduling by means of energy disaggregation. The input for our heating control framework is an aggregated energy signal, such as that coming from a smart meter in a residential home. In the first step (i) we extract features from the energy signal, i.e. changes in consumption, which can be used to categorize the individual electrical



**Fig. 10.1** Framework for heating control and scheduling by means of energy disaggregation techniques

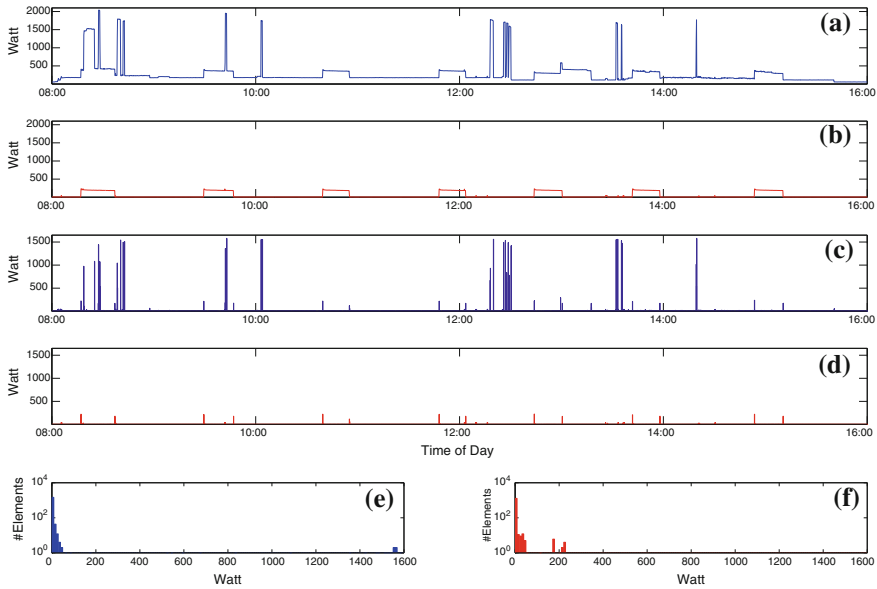
devices. Subsequently, (ii) we use the extracted features as input for the appliance state classification. For the sake of simplicity, Fig. 10.1 assumes that the individual appliance models were trained on other households prior to the classification task. Given the classified ON/OFF states for each appliance, we can eventually (iii) infer the occupancy state of the respective household and recommend optimized heating schedules.

In the following subsections, we describe the (i) feature extraction, (ii) appliance state classification, and (iii) inference of occupancy in more detail.

### 10.4.1 Feature Extraction

Given the overall energy consumption of a household and the energy consumption of the individual appliances in this household, we aim to build a model for each appliance in order to estimate its ON/OFF states in a previously unknown environment or household. Since an appliance can be either turned ON or OFF, the device state identification can be formalized as a two class problem. For the training of an individual appliance model, we consider the changes in power consumption that classify the respective device states. In our approach, the input for the classification model are two distributions of power changes, which represent the features that characterize one or the other class.

Figure 10.2 illustrates the feature extraction process on the basis of real-life measurements from the REDD data set, in particular the energy consumption of (a) House 1 and (b) its refrigerator for a sample time frame of 8 h. We can see that (a) the overall energy consumption is the sum of (b) the Refrigerator's energy consumption and the energy consumption of other appliances. Given this information, we can derive the changes in energy consumption by the first-order difference of the power signals. This step is often referred to as edge detection, since the stable periods in the signal are filtered out. The edges or changes in power consumption of the overall energy signal and the Refrigerator signal are shown in Fig. 10.2c, d, respectively. Knowing which edges specify (d) the activity of the Refrigerator, we can easily separate the changes in energy consumption that categorize other devices by considering all the



**Fig. 10.2** Energy consumption of **a** House 1 and **b** its Refrigerator over an interval of 8 h. Plot **c** and **d** show the changes in power consumption for House 1 and its Refrigerator. The distribution of power changes that classify the Refrigerator’s ON/OFF states are illustrated in Plot **e** and **f**

edges in (c) the overall energy signal which do not belong to the Refrigerator. The distribution of the edges that classify the Refrigerator’s ON/OFF states are illustrated in Fig. 10.2e, f. These distributions can serve as training input for most probabilistic models.

### 10.4.2 Appliance State Classification

In this study, we aim at evaluating the appliance state classification task by means of various machine learning techniques, including Naive Bayes (NB) classifier, Factorial Hidden Markov Model (FHMM), Classification Tree (CT), and One-Nearest-Neighbor (1NN) classifier.

We selected these models based on their complementary characteristics and degree of popularity regarding the energy disaggregation task. Table 10.1 shows typical characteristics of the considered machine learning techniques [16], although the

**Table 10.1** Characteristics of algorithms

	NB	FHMM	CT	1NN
Fitting speed	Fast	Fast	Fast	Fast
Prediction speed	Fast	Fast	Fast	Medium
Memory usage	Low	Low	Low	High
Easy to interpret	Yes	No	Yes	No

characteristics strongly depend on the underlying algorithm and the problem. Therefore, Table 10.1 should be considered as a guide for an initial choice of models.

The NB classifier is a simple probabilistic model based on applying Bayes' theorem with strong independence assumptions, which has been applied for appliance and occupancy recognition in various studies [9, 12, 13, 18]. Speed and memory usage of the NB classifier are good for simple distributions, but can be poor for large datasets [16].

The FHMM is a statistical model in which the system under study is assumed to be a Markov process with unobserved or hidden states. FHMMs have been successfully applied to the energy disaggregation problem [8, 10, 23]; however, their complexity increases with the number of states and the length of the Markov chain [5, 8].

CTs map observations about an item to conclusions about the item's target value, meaning the predicted outcome is the class to which the data belongs. Decision tree learning has been proven to be applicable to appliance identification on metering data in a couple of recent studies [1, 18].

The INN classifier is often regarded as the simplest straw man or baseline approach [7], and has been considered for the energy disaggregation task in several studies [12, 13, 23]. INN usually has good performance in low dimensions, but can have poor predictions in high dimensions. For linear search, INN does not perform any fitting [16].

### ***10.4.3 Inference of Occupancy***

We assume that there exists a direct relationship between appliance usage and occupancy states in residential homes. For instance, if the lighting is turned ON, we usually know that the residents are at home, unless someone forgot to turn OFF the lighting. Hence, lighting may be a straightforward indicator for occupancy states, enabling us to verify manually adjusted heating schemes and recommend optimized heating schedules.

However, heating control is much more complex, because the usage of certain appliance actually requires to decrease the temperature. For example, when residents turn ON the oven or stove, the temperature in the kitchen rises automatically, and we can reduce heating to save energy, instead of just opening the window. In case the heating control system would have knowledge about the installation points of all devices, one could even use the appliance states to control the temperature in individual rooms.

The knowledge of individual appliance states furthermore allows us to infer devices that are unrelated to occupancy. For instance, the refrigerator automatically switches between ON and OFF state every few minutes, no matter if the residents are at home or not. The same is true for devices in standby mode or appliances such as the smoke alarm or electronic panels which are constantly drawing power. Therefore, by just looking at the overall energy consumption of a household it is impossible to distinguish between occupancy states.



The accuracy of the appliance state classification and the implications for heating control will be scrutinized in the following section.

## 10.5 Empirical Evaluation

The goal of our evaluation is twofold: (i) we investigate which of the considered machine learning models is most accurate for the the appliance state classification task; and (ii) we assess the use of the identified appliance or rather occupancy states for heating control.

### 10.5.1 Energy Data

We consider the REDD dataset [10], which comprises electricity consumption measurements from six household at the granularity level of individual devices, and represents to date one of the largest and richest publicly available collections of power consumption data [2]. There are approximately 20 consecutive days of measurements available for each house, providing data from the two main phases and each individual circuit at 1 Hz frequency rate. Measured appliances include main consumers such as Air Conditioning, Dishwasher, Disposal, Electrical Heating, Microwave, Oven, Refrigerator, Stove, Washer/Dryer as well as other miscellaneous electronics and outlets (see Table 10.2).

### 10.5.2 Experimental Design

In our empirical evaluation, we compare the classification accuracy of the introduced machine learning models (see Table 10.1) on the REDD data set. Strictly speaking, we assess the appliance state classification accuracy for all considered models on a granularity level of individual devices. The training of the respective models is done on appliance-specific consumption measurements of one particular device for all households but one. The aggregated electricity consumption signal of the left-out household is then used for testing the performance of the trained models for each individual device. This evaluation principle is also commonly known as cross-validation with leave-one-out.

### 10.5.3 Classification Accuracy

Table 10.2 illustrates the classification accuracy per (a) household and (b) appliance for all examined models, including Naive Bayes (NB), Factorial Hidden Markov Model (FHMM), Classification Trees (CT), and One-Nearest-Neighbor

**Table 10.2** Cross-validation of trained models

	NB	FHMM	CT	INN
(a) Classification accuracy of device states per household averaged over all appliances				
House 1	0.8429	0.8414	0.8319	0.7615
House 2	0.9310	0.9300	0.9224	0.8062
House 3	0.9275	0.9200	0.8908	0.7213
House 4	0.8645	0.8616	0.8746	0.7038
House 5	0.9864	0.9854	0.9839	0.7638
House 6	0.8131	0.7873	0.7752	0.6050
<b>MEAN</b>	<b>0.8942</b>	<b>0.8876</b>	<b>0.8798</b>	<b>0.7269</b>
(b) Classification accuracy of device states per appliance averaged over all households				
Air conditioning	0.9315	0.9248	0.9300	0.9138
Bathroom GFI	0.9328	0.9275	0.9324	0.9134
Dishwasher	0.9541	0.9493	0.9551	0.9134
Disposal	0.9955	0.9818	0.9970	0.9918
Electrical heating	0.8863	0.8856	0.8620	0.8895
Electronics	0.8875	0.7970	0.7404	0.0991
Furnace	0.8216	0.8211	0.7294	0.5216
Kitchen outlets	0.7902	0.7915	0.7070	0.1775
Lighting	0.7751	0.7737	0.8006	0.7611
Microwave	0.9516	0.9473	0.9526	0.9279
Miscellaneous	0.9242	0.9295	0.9296	0.7237
Outdoor outlets	0.9982	0.9995	0.9997	0.9996
Oven	0.9754	0.9804	0.9815	0.9811
Refrigerator	0.7834	0.7872	0.7952	0.7898
Smoke alarm	0.9729	0.9629	0.9738	0.6234
Stove	0.9346	0.9288	0.9363	0.8330
Subpanel	0.9808	0.9807	0.9815	0.9811
Unknown outlets	0.9578	0.9558	0.9555	0.3432
Washer/Dryer	0.9287	0.9256	0.9297	0.8763
<b>MEAN</b>	<b>0.9148</b>	<b>0.9079</b>	<b>0.8994</b>	<b>0.7505</b>

(INN) classifier. The classification results present the performance of the trained models in an unknown environment or rather before unseen household.

The results in Table 10.2a show the classification accuracy of device states per household averaged over all appliances. For instance, the NB model achieved an accuracy of 0.8429 for House 1, meaning that the model was trained on House 2–6 and tested on the previously unknown House 1, where 84.29% of all device states were classified correctly. However, as illustrated in Table 10.2a the classification accuracy of each model varies with the household, which is due to the fact that the examined households use appliances of different manufacturers with dissimilar energy profiles.

Table 10.2b presents the classification accuracy of device states per appliance averaged over all households. For example, the results show that the NB model is able to classify the device states of the Air-Conditioning with an average accuracy of 93.15 %, taking the mean of House 1–6. In general, all models achieved a relatively high classification accuracy for appliances with distinctive energy profiles, such as the Dishwasher or Oven, but performed less well on appliances with changes in consumption that can easily be confused with other devices, like the Refrigerator or Lighting.

By taking the mean over all results for (a) each household and (b) each appliance per model, shown in the bottom row of Table 10.2a, b respectively, we can easily see that on average the NB model achieved the highest classification accuracy, closely followed by FHMM and CT. Although the INN classifier shows relatively high classification accuracy for several individual appliances, it is unable to correctly classify the device states of others, and, therefore, achieve the lowest average classification performance.

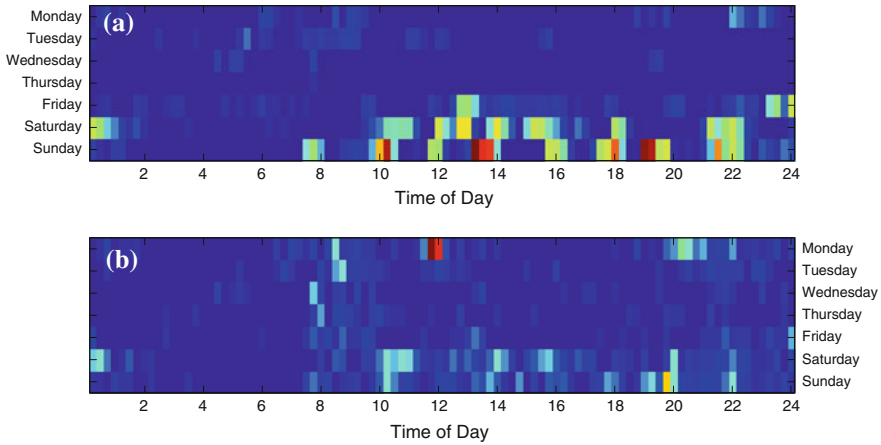
### 10.5.4 Heating Control

In this subsection, we discuss how the classified ON/OFF device states can be used for heating control and scheduling. Since the Naive Bayes (NB) model achieved the highest average accuracy on classifying device states of appliances in an unknown household (see Table 10.2), we will consider the NB approach in our following exemplification.

Figure 10.3 shows the (a) observed and (b) estimated ON/OFF states for the Washer/Dryer in House 1 over a period of 4 weeks, where every quarter of an hour aggregates the device activities that occurred during the same weekday and time of day. By illustrating the (a) observed activity of the Washer/Dryer, which constitutes our ground truth, we see that this appliance is mostly used on Fridays and weekends. The (b) estimated activity of the Washer/Dryer, inferred from the overall energy consumption of House 1 by the trained NB model, shows similar behavior patterns for weekends, but predicts false ON states for Mondays.

By taking a closer look at the confusion matrix of observed and estimated ON/OFF device states for the Washer/Dryer in House 1, shown in Table 10.3, we are able to gain a better understanding of the estimated appliance activity. Table 10.3 reveals the percentage of true positives (TP) or true ON states, true negatives (TN) or true OFF states, false positives (FP) or false ON states, and false negatives (FN) or false OFF states. Although the NB model achieves a high classification accuracy  $[(TP + TN)/(TP + TN + FP + FN) = 96.83 \%$ ], the percentage of falsely classified states  $[FP + FN = 3.17 \%$ ] is not negligible, explaining the mistaken Washer/Dryer activity estimated for Mondays (see Fig. 10.3b). The FP and FN estimates imply heating during absence and cooling during occupancy, respectively.

The cause of falsely classified states can also be explained with help of Fig. 10.2. By examining the distribution of ON and OFF states of the refrigerator in House 1,



**Fig. 10.3** Observed and estimated ON/OFF states for the Washer/Dryer in House 1 over the period of 4 weeks, illustrating the actual and predicted device activities in the time from April 25 to May 15 2011, where every quarter of an hour aggregates the activities that occurred during the same weekday and time of day. The transition from *blue*, to *yellow*, to *red* colored areas illustrates low, moderate, and high device activity. **a** Observed ON/OFF states of Washer/Dryer. **b** Estimated ON/OFF states of Washer/Dryer

**Table 10.3** Confusion matrix of observed and estimated ON/OFF device states for the Washer/Dryer in House 1, where Accuracy = TP + TN = 96.83 %.

	Observed ON	Observed OFF
Estimated ON	True positive (TP) = 0.59 %	False positive (FP) = 1.14 %
Estimated OFF	False negative (FN) = 2.03 %	True negative (TN) = 96.24 %

shown in Fig. 10.2e, f respectively, we can see there is a significant overlap of changes in power consumption that are caused by both the Refrigerator and other devices. According to Fig. 10.2e, f, changes in power consumption that range from around 1–50 W occur at times when the refrigerator is turned ON as well as when it is turned OFF, leading to an inaccurate appliance model.

In order to decrease the number of FP and FN device states one could orchestrate the trained appliance models or consider additional features that distinguish the appliances more accurate. However, this goes beyond the scope of this study, but could be part of future work.

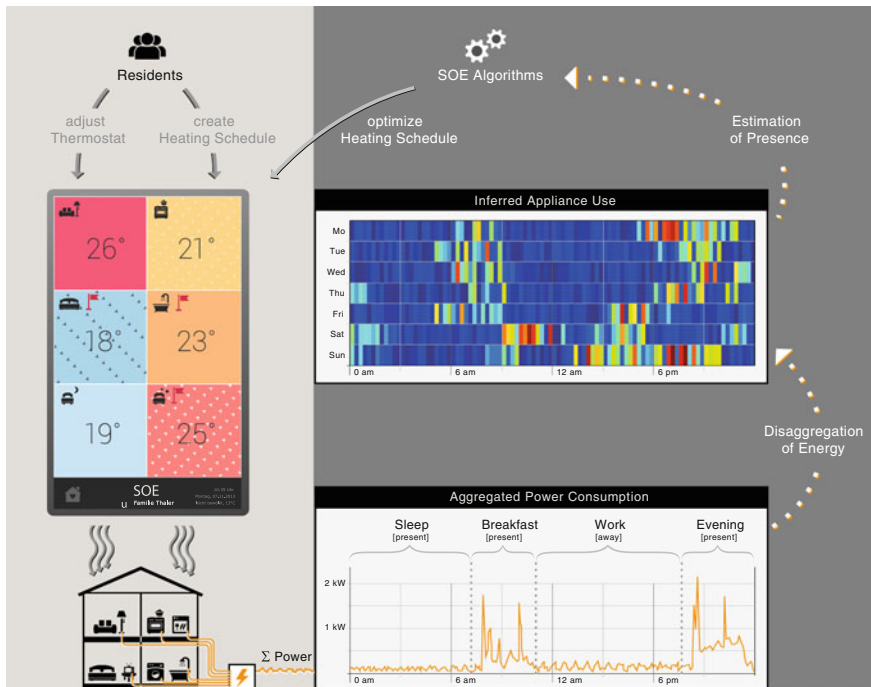
A more thorough evaluation of the heating schedules would require datasets that comprise information about actual occupancy states in the residential homes and preferences of the residents in regard of temperature settings.

## 10.6 Application

Having explained our approach, we are now in the position to present SOE, a single-agent heating control system, that proposes optimized heating schedules that aim to reduce the residential energy consumption. SOE computes the optimized heating schedules based on manual adjustments of the residents and automatically determined occupancy states. In addition, SOE enables the residents to monitor and control their heating from remote using mobile devices.

In order to build a practical application we embedded the implementation of our trained appliance model in our SOE agent using the Matlab to Java compiler.<sup>1</sup> The SOE agent [15] is responsible for the heating control in a home and has access to the aggregated energy signal using Smart Message Language (SML)<sup>2</sup> and the Multi Utility Communication (MUC)(see footnote 2) interface.

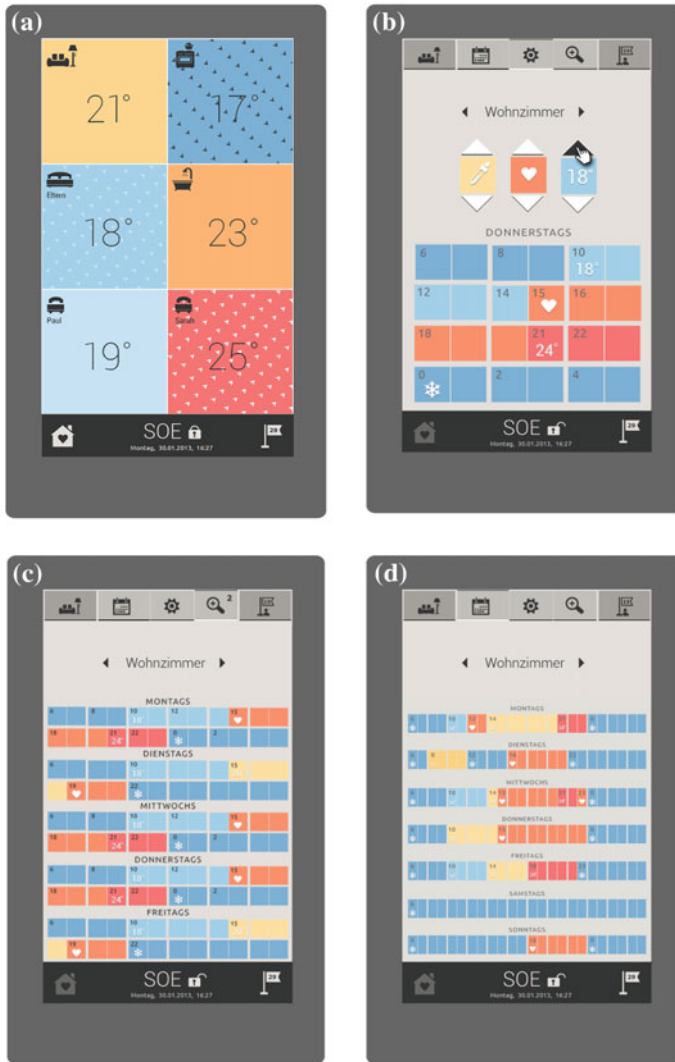
Figure 10.4 illustrates the overall architecture of a SOE agent [15]. The residents are enabled to adjust the thermostat and create heating schedules for each



**Fig. 10.4** Overall architecture of SOE heating control system. The aggregated energy signal is disaggregated using a Naive Bayes classifier to infer appliance usages. From such usage the occupants presence is estimated and used to optimize the heating schedule. The whole system can be controlled by the residents using tablets and smartphones

<sup>1</sup> [www.mathworks.de/products/javabuilder/](http://www.mathworks.de/products/javabuilder/).

<sup>2</sup> [www.vde.com/en/fnn/extras/sym2/Infomaterial/Pages/default.aspx](http://www.vde.com/en/fnn/extras/sym2/Infomaterial/Pages/default.aspx).



**Fig. 10.5** Graphical user interface (GUI) of SOE heating control system, showing the temperature settings for **a** all rooms at current time, **b** one individual room for a specific weekday, **c** a single room for all workdays, and **d** one individual room for the entire week. The manually created heating schedules are compared against the automatically optimized scheme in order to give recommendations for possible energy savings at idle time intervals

individual room (refer to Fig. 10.5). Given the aggregated power consumption of the household, our implemented energy disaggregation component is able to classify individual appliance states. The inferred appliance use is subsequently employed to infer presence and to propose optimized heating schedules to the residents.

The SOE heating control system aims at integrating the user as an essential part of the heating control process. At this point, we want to address questions concerning various aspects of human computer interaction. This includes the usability and acceptance of the developed system with regard to different user groups and/or environments. Users are able to access the system using mobile devices and control the heating process in a fine-grained manner (refer to Fig. 10.5).

In case that the manually created and automatically optimized heating schedules differ from each other, the SOE agent will provide recommendations for possible adaptations. These suggestions are shown as notifications, whereas the user can either accept the recommended adaptations or reject the automatically generated heating schedule to manually conduct changes. This is of utter importance, because the number of falsely classified appliance states is not negligible. False estimates imply heating during absence or cooling during presence, and are, therefore, undesired.

In our future work, we intend to reduce the number of false estimates and in consequence to improve the appliance classification by using acceptance/rejection as reward/punishment signal for reinforcement learning strategies. In order to demonstrate the system outside of our showroom, we use a common notebook to simulate the smart home and an iPad to show the SOE application.

## 10.7 Conclusion and Future Work

In this work, we reviewed recent advances in energy disaggregation and adopted established appliance identification strategies to infer occupancy states for smart heating control and scheduling. Our proposed approach to appliances state identification considers the changes in power consumption as characteristic to classify the individual devices. In our evaluation, we have shown that the Naive Bayes classifier is able to achieve relatively high accuracy on the appliance state identification task, even in unknown environments or households. Furthermore, we explained how to use the information about identified appliances to infer occupancy states in residential homes. We exemplified the idea of occupancy-based heating schedules and discussed the problem of falsely identified appliance states.

The main advantage of our proposed framework for heating control and scheduling is its simplicity in that we refrain from implementing new infrastructure in residential homes, but use given information from available electricity smart meters. This approach will eventually lead to higher acceptance rates among residents and provides alternative avenues for novel heating control strategies.

In addition, we demonstrated SOE, a smart heating control system, which integrates the discussed energy disaggregation algorithms to infer appliances states that indicate presence. Our implementation of the SOE provides insights into practicality and usability, which are valuable for the intended deployment in real estates.

Since our appliance state identification strategy can replace sensing infrastructure that is used to identify occupancy states in residential homes, it would also be interesting to compare the energy savings provided by our approach with the performance of

existing frameworks, such as the smart thermostat [14]. However, this would require datasets that comprise information about actual occupancy states in the residential homes and preferred temperature settings.

Our proposed approach to appliance state identification can furthermore be beneficial for other applications. Recent studies [2] have shown that the availability of smart meter data alone is often not sufficient to achieve high load disaggregation accuracies. Future work could combine the knowledge of total energy consumption with additional information about sequences of events, such as ON/OFF states for each individual appliance, to improve the accuracy of certain disaggregation algorithms [2] that use such events along with smart meter data.

**Acknowledgments** This work was funded by the Federal Ministry of Economic Affairs and Energy (BMWi) under funding reference number KF2392312-KM2. The presented SOE application was developed by Veit Schwartz, Stephen Prochnow, and Marie Schacht.

## References

1. K.C. Armel, A. Gupta, G. Shrimali, A. Albert, Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy* **52**(C), 213–234 (2013)
2. C. Beckel, W. Kleiminger, T. Staake, S. Santini, Improving device-level electricity consumption breakdowns in private households using on/off events. *SIGBED Rev.* **9**(3), 32–38 (2012)
3. P.J. Boait, R.M. Rylatt, A method for fully automatic operation of domestic heating. *Energy Build.* **42**(1), 11–16 (2010)
4. J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, S. Patel, Disaggregated end-use energy sensing for the smart grid. *IEEE Pervasive Comput.* **10**(1), 28–39 (2011)
5. Z. Ghahramani, M.I. Jordan, Factorial hidden Markov models. *Mach. Learn.* **29**(2–3), 245–273 (1997)
6. S. Gupta, M.S. Reynolds, S.N. Patel, Electrisense: single-point sensing using EMI for electrical event detection and classification in the home, in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pp. 139–148 (2010)
7. E.J. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Min. Knowl. Discov.* **7**(4), 349–371 (2003)
8. H. Kim, M. Marwah, M.F. Arlitt, G. Lyon, J. Han, Unsupervised disaggregation of low frequency power measurements, in *SDM*, pp. 747–758 (2011)
9. W. Kleiminger, C. Beckel, S. Santini, Opportunistic sensing for efficient energy usage in private households, in *Proceedings of the Smart Energy Strategies Conference 2011* (2011)
10. J.Z. Kolter, M.J. Johnson, REDD: a public data set for energy disaggregation research, in *Proceedings of SustKDD Workshop on Data Mining Applications in Sustainability* (2011)
11. S.J. Lian, S. Ng, G. Kendell, J. Cheng, Load signature study—part i: basic concept, structure, and methodology. *IEEE Trans. Power Deliv.* **25**(2), 551–560 (2010)
12. G.-Y. Lin, S.-C. Lee, J.Y.-J. Hsu, Sensing from the panel: applying the power meters for appliance recognition, in *Proceedings of the 14th Conference on Artificial Intelligence and Applications* (2009)
13. J. Lines, A. Bagnall, P. Caiger-Smith, S. Anderson, Classification of household devices by electricity usage profiles, in *IDEAL*, pp. 403–412 (2011)
14. J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, K. Whitehouse, The smart thermostat: using occupancy sensors to save energy in homes, in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 211–224 (2010)



15. M. Luetzenberger, T. Kuester, T. Konnerth, A. Thiele, N. Masuch, A. Hessler, J. Keiser, M. Burkhardt, S. Kaiser, S. Albayrak, JIAC V—a MAS framework for industrial applications (extended abstract), in *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, ed. by T. Ito, C. Jonker, M. Gini, O. Shehory, pp. 1189–1190 (2013)
16. MathWorks (TM), Statistics Toolbox: Supervised Learning Workflow and Algorithms (R2013a)
17. V. Pallotta, P. Bruegger, B. Hirsbrunner, Smart heating systems: optimizing heating systems by kinetic-awareness, in *Proceedings of 3rd International Conference on Digital Information Management* (2008)
18. A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, R. Steinmetz, On the accuracy of appliance identification based on distributed load metering data, in *Proceedings of 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability* (2012)
19. A.G. Ruzzelli, C. Nicolas, A. Schoofs, G.M.P. O’Hare, Real-time recognition and profiling of appliances through a single electricity sensor, in *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 279–287 (2010)
20. K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, K. Ito, Nonintrusive appliance load monitoring based on integer programming, in *Proceedings on SICE Annual Conference* (2008)
21. M. Weiss, A. Helfenstein, F. Mattern, T. Staake, Leveraging smart meter data to recognize home appliances, in *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, pp. 190–197 (2012)
22. M. Zeifman, K. Roth, Nonintrusive appliance load monitoring: review and outlook. *IEEE Trans. Consum. Electron.* **57**(1), 76–84 (2011)
23. A. Zoha, A. Gluhak, M. Ali Imran, S. Rajasegarar, Non-intrusive load monitoring approaches for disaggregated energy sensing: a survey. *Sensors* **12**(12), 16838–16866 (2012)

# Chapter 11

## Detecting Violent Content in Hollywood Movies and User-Generated Videos

Esra Acar, Melanie Irrgang, Dominique Maniry and Frank Hopfgartner

**Abstract** Detecting violent scenes in videos is an important content understanding functionality, e.g., for providing automated youth protection services. The key issues in designing violence detection algorithms are the choice of discriminative features and learning effective models. We employ low and mid-level audio-visual features and evaluate their discriminative power within the context of the MediaEval Violent Scenes Detection (VSD) task. The audio-visual cues are fused at the decision level. As audio features, Mel-Frequency Cepstral Coefficients (MFCC), and as visual features dense histogram of oriented gradient (HoG), histogram of oriented optical flow (HoF), Violent Flows (ViF), and affect-related color descriptors are used. We perform feature space partitioning of the violence training samples through  $k$ -means clustering and train a different model for each cluster. These models are then used to predict the violence level of videos by employing two-class support vector machines (SVMs). The experimental results in Hollywood movies and short web videos show that mid-level audio features are more discriminative than the visual features, and that the performance is further enhanced by fusing the audio-visual cues at the decision level.

### Babysitting

Nowadays our children are submerged by connected equipment, whether this is at school, at home, or even in the car. Notable examples of such equipment include TV, cable or satellite set-top boxes, tablets or the smartphones of the parents, when those let their children play with it.

---

E. Acar (✉) · M. Irrgang · D. Maniry · F. Hopfgartner  
Technische Universität Berlin, Berlin, Germany  
e-mail: esra.acar@tu-berlin.de

M. Irrgang  
e-mail: melanie.irrgang@campus.tu-berlin.de

D. Maniry  
e-mail: dmaniry@cs.tu-berlin.de

F. Hopfgartner  
e-mail: frank.hopfgartner@tu-berlin.de

Think of tablets. Children use it for learning to read or to count; finding where “Wally” is; watching funny videos on streaming websites. And for TVs, Video-On-Demand (VOD) services such as Netflix, Hulu, or Amazon Prime Instant, allow them to watch all sorts of video content, including not only “educational” content, but also—and children of course are not aware of it—content which is harmful for them. Parents can always try to track these “unwanted” contents. But sometimes, when they are out of home, they have to rely on babysitters for taking care of their children.

Occasionally, Clara works as a babysitter. A conscientious one! Not the sort of babysitter who would put a child in front of the TV or in front of an iPad and focus on talking for hours on the phone with her friends until the parents finally come back and pay her.

Today, Saturday, Clara is babysitting Ben, the 9-year-old child of her neighbors David and Rose. These are celebrating the promotion which Rose just obtained, so they invited many guests in a restaurant, where the parents of Clara are also present.



The dinner is over. “Hey, what would you like to do, Ben?”, asks Clara. Ben is not very talkative. He points to the TV with his finger. “Aha, so you want to watch TV, huh? Okay.” adds Clara. Ben noticed that the famous 1990 movie *Home Alone* which he heard of recently, is aired on the VOD service. This is a well-known children comedy movie starring child Hollywood star Macaulay Culkin.

Clara then says: “Hey buddy, I know that movie very well, you know! I agree to watch it with you, but only if we skip the bad scenes, the scenes which are not good for a cute child like you!” She remembers that some scenes are very funny but also very violent. Especially, the scenes where the burglars first attempt to enter the house. They got badly shot by a toy rifle. Then, one of them gets his scalp burned... Or inside

the house, when one of them is hit by a can of paint, or when he walks barefoot on crushed glass and gets hurt. Really funny, but really violent...

Now, with the violent movie content detector plug-in, finding and skipping violent parts has never been made so easy. Clara just has to turn on the violence detection plug-in on the Smart TV and quickly browse the movie before letting Ben watch it. Thanks to the plug-in, she can easily recognize violent parts while watching the movie and jump to the next nonviolent scene when necessary. Before a violence detector could offer such possibilities, Smart TV users were provided information mainly based on the Electronic Program guide (EPG).

In their most common forms, EPG information contains a short description of the TV program and a parental guidance (PG) rating for the whole program (referring to the recommended minimal age of the spectator). Unlike an EPG, the plug-in works by analyzing directly the audio and visual content to discover scenes which a spectator would consider as violent. The advantage is that, instead of discarding a whole movie based solely on a PG rate, the plug-in can label the movie scene-by-scene or even shot-by-shot.

At the end of the movie, Clara noticed that the plug-in was really helpful. She thinks, "Wow, almost all violent scenes were found by that app. The kid did not see any of those violent scenes."

Clara checks what time it is: "Only 7? What else am I gonna do with the kid until the parents are back?" Indeed, Ben is allowed to stay awake until 9 pm on Saturday evenings. He likes to play games on the tablet—crushing objects, throwing birds away and many other games—or watch funny clips on Youtube or Dailymotion—videos sent by his schoolmates via popular social networks. Clara knows that the tablet belongs to the parents and that they do not have any parental control filter installed on it. Therefore, she is not very much pleased with the idea of tablet. However, something suddenly pops in her mind. "Oh yes, that violence detection plug-in on the TV has an equivalent application for the tablet. I remember that it can also be downloaded for the tablet from the Internet", she remembers.

She installs it, sets violent scene detection on, and hands the tablet to Ben. Convinced by the performance of the plug-in on the TV, she is confident that Ben risks nothing when such a feature is turned on. He can play with it until it is time to go to bed...

## 11.1 Introduction

As the amount of available multimedia content becomes more and more abundant, the use of automatic multimedia analysis solutions in order to find relevant semantic search results or to identify illegal content present on the World Wide Web has reached a critical importance. In addition, the advances in digital media management techniques have facilitated delivering digital videos to consumers. Therefore, accessing online movies through services such as VOD has become extremely easy. As a result, parents are not able to constantly and precisely monitor what their

children watch. Children are, consequently, exposed to movies, documentaries, or reality shows which have not necessarily been checked by parents, and which might contain inappropriate content. Violence constitutes one example of such inappropriate content. Psychological studies have shown that violent content in movies has harmful impacts, especially on children [4]. As a consequence, there is a need for automatically detecting violent scenes in videos, where the legal age ratings are not available.

Like any other research challenge, tackling the problem of violence detection begins with establishing a framework, in particular adopting a definition of violence to work with. Since the concept of violence is highly subjective (i.e., person-dependent)—not everybody would indeed evaluate a particular scene of a movie as violent, one of the challenges within the context of multimedia violent content detection is to properly delimit the boundaries of what can be designated as a “violent” scene. In our work, we aim at sticking to the two definitions of violence as described in [8]: the objective and subjective points of view. *Objective violence* is defined as “physical violence or accident resulting in human injury or pain,” whereas *subjective violent* scenes are “those which one would not let an 8-year-old child see because they contain physical violence.”

In this context, the MediaEval Violent Scene Detection (VSD) task [9] provides a consistent evaluation framework to the research community and enables different approaches to be evaluated by using the same violence definitions and a standardized annotated dataset. Detailed description of the task, the dataset, the ground truth, and evaluation criteria are given in the chapter by Demarty et al. [9]. The task ascribes to a use case from the company *Technicolor*.<sup>1</sup> The French producer of video content and entertainment technologies adopted the aim of helping users to select movies that are suitable to watch with their children.

In spite of the existence of institutions, the task of which is to assign a recommended age to movies in France, the ratings determined by those institutions are not as strict and differentiated as the ones from the German *Freiwillige Selbstkontrolle der Filmwirtschaft* (FSK). One explanation of the sources of discrepancies is the fact that employees from the film-making industry are allowed to participate in the recommendation process in France.

A lot of movies labeled as FSK 16 (i.e., recommended for an audience of age higher than 16 years old) in Germany are released without restrictions in France.<sup>2</sup> There also is no equivalence for the FSK 6 label in France, where movies are recommended only for the age of 0, 12, 16 and 18.<sup>3</sup> This seems to be the main motivation behind the introduction of the challenge, for which one additional sub-task in 2013 is to detect scenes with subjective violence. Another illustration of differences between countries is the age-rating labels used in the USA, where one example of label is “NC-17.” The latter does not mean that audience should be at least 17 years old, but that audience

---

<sup>1</sup> <https://research.technicolor.com/rennes/>.

<sup>2</sup> <http://fsf.de/jugendmedienschutz/international/filmfreigaben/>.

<sup>3</sup> <http://www.fsk.de/>.

should not be 17 or under 17, while, for instance, FSK 16 means audience should be at least 16. This can also be a source of confusion among consumers.

The degree of violence one is able or willing to bear might vary strongly even within a group of persons of identical age. That is probably why parents should get from *Technicolor* information which is not limited to rating only but also a preview of the most violent scenes, in order to help them decide if the movie is adequate to be watched by their child.

Next to the issue of definition, another important step in the task of movie violent content detection is the representation of movie segments. Many of the existing works (e.g., [5, 14]) proposed for violence detection represent videos using low-level representations, especially for the representation of audio signals. Inferring abstract representations is more suitable than directly using low-level features in order to bridge the semantic gap between the features and high-level human perception of violence. However, high-level semantics are more difficult to detect and state-of-the-art detectors are far from perfect. Therefore, the use of mid-level representations may help modeling video segments one step closer to human perception.

This chapter aims at assessing the discriminative power of mid-level audio-visual features to model violence in Hollywood movies. We also investigate the effects of combining mid-level audio-visual features with low-level audio-visual features for the detection of violent content and show that promising results are obtained by fusing audio-visual cues at the decision level.

The chapter is organized as follows. Section 11.2 explores the recent developments and reviews methods which have been proposed in the literature in order to detect violence in movies. In Sect. 11.3, we introduce our method and the functioning of its various components. We provide and discuss evaluation results obtained on Hollywood movies in Sect. 11.4. In Sect. 11.5, we present our browser-based visualization tool which provides an intuitive way of using our solution. Concluding remarks and future directions to expand our current approach are presented in Sect. 11.6.

## 11.2 Related Work

Although video content analysis has been studied extensively in the literature, violence analysis of movies or of user-generated videos is restricted to a few studies only. We discuss here some of the most representative ones which use audio and/or visual cues. A difficulty arises regarding the definition of violence. In some of the works presented in this section, the authors do not explicitly state their definition of violence. In addition, nearly all papers in which the concept is defined consider a different definition of violence; therefore, whenever possible, we also specify the definition adopted in each work discussed in this section.

First, we briefly discuss uni-modal (i.e., based exclusively on the audio or visual modality) violence detection methods. Giannakopoulos et al. [13] define violent scenes as those containing shots, explosions, fights and screams, whereas nonviolent content corresponds to audio segments containing music and speech. Frame-level

audio features both from the time and the frequency domain such as energy entropy, short-time energy, zero crossing rate (ZCR), spectral flux, and roll-off are employed. A polynomial SVM is used as the classifier. The main issue with this work is that audio signals are assumed to have already been segmented into semantically meaningful nonoverlapping pieces (i.e., shots, explosions, fights, screams, music and speech).

In their chapter [7], de Souza et al., similarly to other works related to violence detection, adopt their own definition of violence, and designate violent scenes as those containing fights (i.e., aggressive human actions), regardless of the context and the number of people involved. Their approach is based on the use of Bag-of-Words (BoW), where local Spatial-Temporal Interest Point Features (STIP) are used as the feature representation of video shots. They compare the performance of STIP-based BoW with SIFT-based BoW on their own dataset, which contains 400 videos (200 violent and 200 nonviolent videos). The STIP-based BoW solution has proven to be superior to the SIFT-based one.

Hassner et al. [18] present a method for real-time detection of breaking violence in crowded scenes. They define violence as sudden changes in motion in a video footage. The method considers statistics of magnitude changes of flow-vectors over time. These statistics, collected for short frame sequences, are represented using the Violent Flows (ViF) descriptor. ViF descriptors are then classified as either violent or nonviolent using a linear SVM. The authors also introduce a new dataset of crowded scenes on which their method is evaluated. According to the presented results, the ViF descriptor outperforms the Local Trinary Patterns (LTP) [38], histogram of oriented gradient (HoG) [23], histogram of oriented optical flow (HoF) [23] descriptors as well as the histogram of oriented gradient and optical flow (HNF) descriptor [23]. The ViF descriptor is also evaluated on well-known datasets of videos of noncrowded scene such as the Hockey dataset [28] and the ASLAN dataset [22] in order to assess its performance in action-classification tasks of “non-textured” videos (i.e., noncrowded). With small vocabularies, the ViF descriptor outperforms the LTP and STIP descriptors, while with larger vocabularies, STIP outperforms ViF. However, this performance gain comes with a higher computational cost.

In [35], Xu et al. propose to use Motion SIFT (MoSIFT) descriptors to extract a low-level representation of a video. Feature selection is applied on the MoSIFT descriptors using kernel density estimation. The selected features are subsequently summarized into a mid-level feature representation based on a BoW model using sparse coding. The method is evaluated on two different types of datasets: crowded scenes [18] and noncrowded scenes [28]. Although Xu et al. do not explicitly define violence, they study fights or sudden changes in motion as violence-related concepts. The results show that the proposed method is promising and outperforms HoG-based and HoF-based BoW representations on both datasets.

Second, we review multimodal methods, which constitute the most common type of approach used in violent content detection in videos, and which consist in fusing audio and visual cues at either feature or decision level. Aiming at detecting horror, Wang et al. [5] apply Multiple Instance Learning (MIL; MI-SVM [3]) using color, textual, and MFCC features. The authors do not explicitly state their definition of horror. Therefore, assessing the performance of their method and identifying the

situations on which it properly works is difficult. Video scenes are divided into video shots, where each scene is formulated as a bag and each shot as an instance inside the bag for MIL. Color and texture features are used for the visual representation of video shots, while MFCCs are used for the audio representation. More specifically, mean, variance, and first-order differential of each dimension of MFCCs are employed for the audio representation. As observed from their results [5], using color and textural information in addition to MFCC features slightly improves the performance.

Giannakopoulos et al. [14], in an attempt to extend their approach based solely on audio cues [13], propose to use a multimodal two-stage approach. In the first step, they perform audio and visual analysis of segments of one-second duration. In the audio analysis part, audio features such as energy entropy, ZCR, and MFCCs are extracted and the mean and standard deviation of these features are used to classify scenes into one of seven classes (violent ones including shots, fights and screams). In the visual analysis part, average motion, motion variance, and average motion of individuals appearing in a scene are used to classify segments as having either high or low activity. The classifications obtained in this first step are then used to train a  $k$ -NN classifier.

In [15], a three-stage method is proposed. In the first stage, the authors apply a semi-supervised cross-feature learning algorithm [37] on the extracted audio-visual features such as motion activity, ZCR, MFCCs, pitch, and rhythm features for the selection of candidate violent video shots. In the second stage, high-level audio events (e.g., screaming, gun shots, explosions) are detected via SVM training for each audio event. In the third stage, the outputs of the classifiers generated in the previous two stages are linearly weighted for final decision. Although not explicitly stated, the authors define violent scenes as those which contain action and violence-related concepts such as gunshots, explosions, and screams. The method was only evaluated on action movies. However, violent content can be present in movies of all genres (e.g., drama). The performance of this method in genres other than action is, therefore, unclear.

Lin and Wang [24] train separate classifiers for audio and visual analysis and combine these classifiers by co-training. Probabilistic latent semantic analysis is applied in the audio classification part. Spectrum power, brightness, bandwidth, pitch, MFCCs, spectrum flux, ZCR, and harmonicity prominence features are extracted. An audio vocabulary is subsequently constructed by  $k$ -means clustering. Audio clips of one-second length are represented by the audio vocabulary. This method also constructs mid-level audio representations with a technique derived from text analysis. However, this approach presents the drawback of only constructing a dictionary of 20 audio words, which prevents having a precise representation of the audio signal of video shots. In the visual classification part, the degree of violence of a video shot is determined by using motion intensity, the (non-)existence of flame, explosion, and blood appearing in the video shot. Violence-related concepts studied in this work are fights, murders, gunshots, and explosions. This method was also evaluated only on action movies. Therefore, the performance of this solution in genres other than action is uncertain.

Chen et al. [6] proposed a two-phase solution. According to their violence definition, a violent scene is a scene that contains action and blood. In the first phase,



where average motion, camera motion, and average shot length are used for scene representation and SVM for classification, video scenes are classified into action and nonaction. In the second phase, using the “Viola-Jones” face detector, faces are detected in each keyframe of action scenes and the presence of blood pixels near detected human faces is checked using color information. The approach is compared with the method of Lin and Wang [24] because of the similar violence definitions, and is shown to perform better in terms of precision and recall.

Ding et al. [11] observe that most existing methods identify horror scenes only from independent frames, ignoring the context cues among frames in a video scene. In order to consider contextual cues in horror scene recognition, they propose a Multiview MIL ( $M^2IL$ ) model based on a joint sparse coding technique which simultaneously takes into account the bag of instances from the independent view and from the contextual view. Their definition of violence is very similar to the definition in [5]. They perform experiments on a horror video dataset collected from the Internet and the results demonstrate that the performance of the proposed method is superior to other existing well-known MIL algorithms.

The works discussed in the following paragraphs employ the same definitions of violence (i.e., *objective* and/or *subjective* violence) adopted in the MediaEval 2013 VSD task. Penet et al. [29] propose to exploit temporal and multimodal information for objective violence detection at video shot level. In order to model violence, different kinds of Bayesian network structure learning algorithms are investigated. The proposed method is tested on the dataset of the MediaEval 2011 VSD Task. Experiments demonstrate that both multimodality and temporality add valuable information into the system and improve the performance in terms of MediaEval cost function [9]. The best-performing method achieves 50 % false alarms and 3 % missed detection, ranking among the best submissions to the MediaEval 2011 VSD task.

In [21], Ionescu et al. address the detection of objective violence in Hollywood movies. The method relies on fusing mid-level concept predictions inferred from low-level features. The mid-level concepts used in this work are gory scenes, presence of blood, firearms and cold weapons (for the visual modality); presence of screams and gunshots (for the audio modality); and car chases, presence of explosions, fights, and fire (for the audio-visual modalities). The authors employ a bank of multilayer perceptrons featuring a dropout training scheme in order to construct 10 violence-related concept classifiers. The predictions of these concept classifiers are then merged to construct the final violence classifier. The method is tested on the dataset of the MediaEval 2012 VSD task and ranked first among 34 other submissions, in terms of precision and F-measure.

In [16], Goto and Aoki propose a violence detection method which is based on the combination of visual and audio features extracted at the segment level, using machine learning techniques. Violence detection models are learned via multiple kernel learning. The authors also propose mid-level violence clustering in order to implicitly learn mid-level concepts without using manual annotations. The proposed method is trained and evaluated on the MediaEval 2013 VSD task using the official MediaEval metric Mean Average Precision at 100 (MAP@100). The results show

that the method outperforms the approaches which use no external data (e.g., Internet resources) in the MediaEval 2013 VSD task.

Derbas and Quénot [10] explore the joint dependence of audio and visual features for violent scene detection. They first combine the audio and the visual features and then determine statistically joint multimodal patterns. The proposed method mainly relies on an audio-visual BoW representation. The experiments are performed in the context of the MediaEval 2013 VSD task. The obtained results show the potential of the proposed approach in comparison to methods which use audio and visual features separately, and to other fusion methods such as early and late fusion.

## 11.3 The Violence Detection Method

In this section, we discuss (1) the representation of video segments, and (2) the learning of a violence model, which are the two main components of our method.

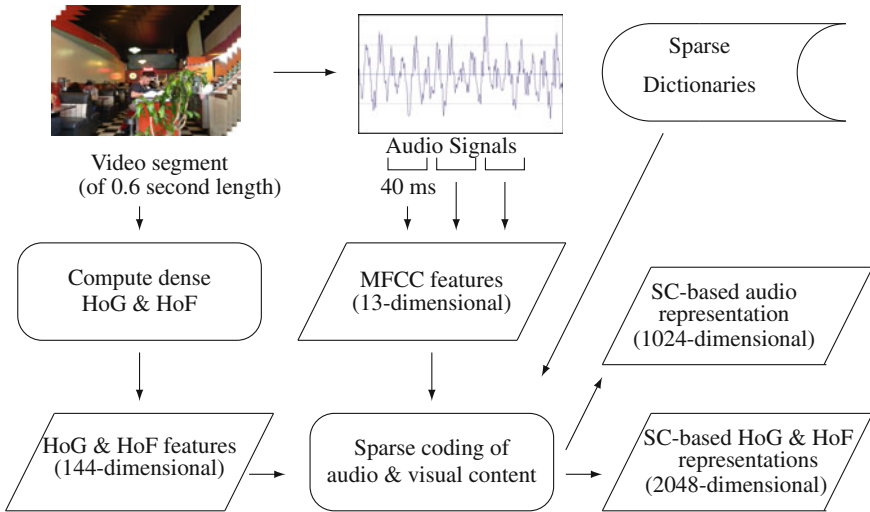
### 11.3.1 Video Representation

Sound effects and background music in movies are essential for stimulating people's perception [33]. Therefore, the audio signals are important for the representation of videos. Visual content of videos provides complementary information for the detection of violence in videos. We represent the audio content using mid-level representations, whereas the visual content is represented at two different levels: low-level and mid-level.

#### 11.3.1.1 Mid-Level Audio Representation

Mid-level audio representations are based on MFCC features extracted from the audio signals of video segments of 0.6 s length as illustrated in Fig. 11.1. In order to generate the mid-level representations for video segments, we apply an abstraction process which uses an MFCC-based Bag-of-Audio Words (BoAW) approach with sparse coding (SC) as the coding scheme.

The construction of the SC-based audio dictionary is illustrated in Fig. 11.2. We employ the dictionary learning technique presented in [26]. The advantage of this technique is its scalability to very large datasets containing millions of training samples which makes the technique well suited for our work. In order to learn the dictionary of size  $k$  ( $k = 1,024$  in this work) for sparse coding,  $400 \times k$  MFCC feature vectors are sampled from the training data (experimentally determined figure). In the coding phase, we construct the sparse representations of audio signals by using the LARS algorithm [12]. Given an audio signal and a dictionary, the LARS algorithm returns sparse representations for MFCC feature vectors. In order to generate the final sparse representation of video segments, which is a set of MFCC feature vectors, we apply the *max-pooling* technique.



**Fig. 11.1** The generation process of SC-based audio and visual representations for video segments. Each video segment is of length 0.6s. Separate dictionaries are constructed and used for MFCC, HoG and HoF to generate 1,024-dimensional representations. Each HoG and HoF descriptor is 144-dimensional

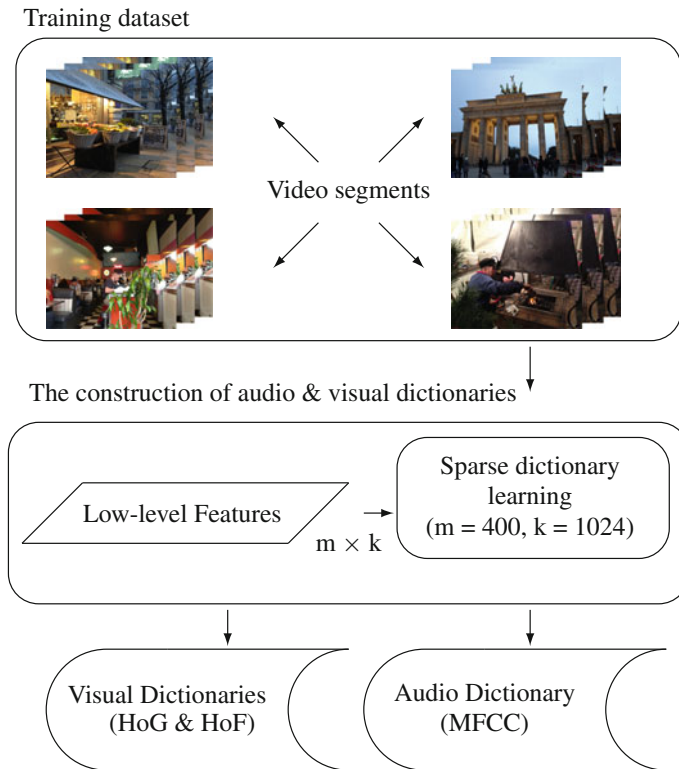
### 11.3.1.2 Low-Level Visual Representation

Film-makers usually make use of motion in order to elicit some particular perception in the audience [33]. Therefore, we use motion-related descriptors for the visual representation of video segments. One of the motion descriptors is ViF which is an efficient motion descriptor. We computed a ViF descriptor for each video segment to represent statistics of flow-vector magnitude changes over time. For a detailed explanation of the computation of this descriptor, the reader is referred to [18].

In addition to motion information, static content of video frames is also important for evoking some particular perception in the audience [33]. We, therefore, also use static content representations in our work. More specifically, we employ affect-related static visual descriptors. Inspired by the work presented in [25], we compute mean and standard deviation of saturation, brightness, and hue in the HSL color space. We also compute the colorfulness of the keyframe of video segments using the method in [17], where the keyframe is deemed to be the frame in the middle of a video segment.

### 11.3.1.3 Mid-level Visual Representation

Mid-level visual representations are based on HoG and HoF features extracted from the visual content of video segments of 0.6s length. HoG and HoF descriptors are densely sampled and computed for subvolumes of video segments (HoG descriptors

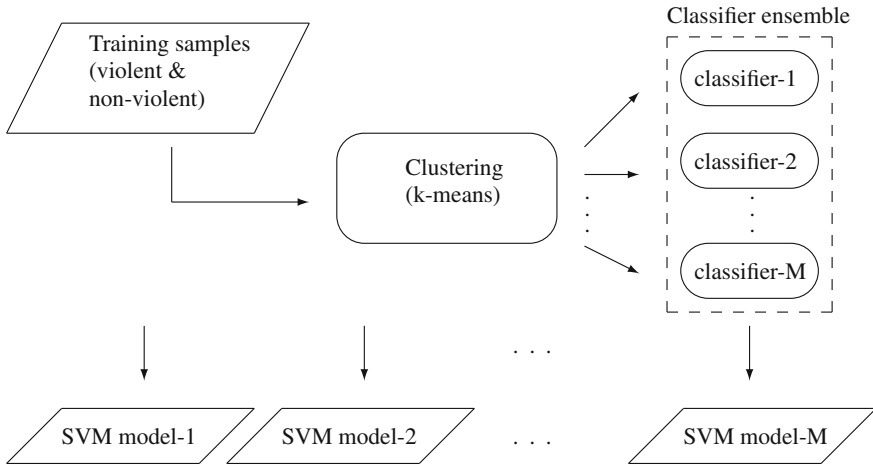


**Fig. 11.2** The generation of audio and visual dictionaries with sparse coding. Each video segment is of length 0.6 s. Low-level features are MFCCs, densely sampled HoG and HoF descriptors

are subsampled every 6 frames and HoF descriptors are subsampled every 2 frames as recommended in [32]). The Horn-Schunk method [20] is applied to compute optical flow vectors which are used for the extraction of HoF descriptors. The resulting HoG and HoF descriptors are subsequently used to generate mid-level HoG and HoF representations separately, which is illustrated in Fig. 11.1. The construction of the SC-based HoG and HoF dictionaries is illustrated in Fig. 11.2.

### 11.3.2 Violence Detection Model

“Violence” is a concept, which can be expressed in diverse manners. For instance, both explosions and scream scenes are labeled as violent according to the definition that we adopted. However, these scenes might highly differ from each other in terms of audio-visual appearance depending on their characteristics of violence. Therefore, instead of learning a unique model for violence detection, learning multiple models constitutes a more judicious choice. This justifies that we first perform feature space

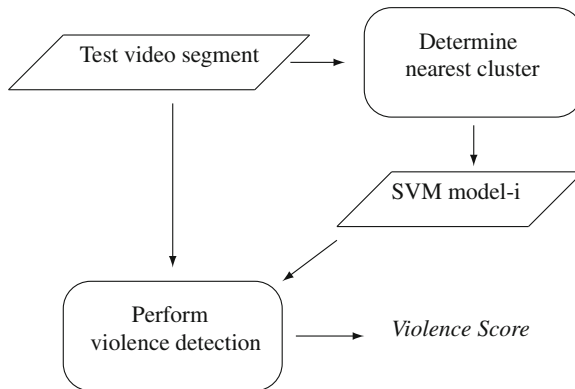


**Fig. 11.3** The generation of violence detection models with feature space partitioning

partitioning by clustering video segments of 0.6 s length in our training dataset and learn a different model for each violence subconcept (i.e., cluster). We use two-class SVMs in order to learn violence models. An overview of the generation of violence detection models is presented in Fig. 11.3.

In the learning step, the main issue is the problem of imbalanced data. This is caused by the fact that, in the training dataset, the number of nonviolent video shots is much higher than the number of violent ones. This phenomenon results in the learned boundary being too close to the violent instances. Consequently, the SVM tends to classify every sample as nonviolent. Different strategies to “push” this decision boundary toward the nonviolent samples exist. Although more sophisticated methods dealing with the imbalanced data issue have been proposed in the literature (see [19] for a comprehensive survey), we choose, in the current framework, to perform random undersampling to balance the number of violent and non-violent samples (with a balance ratio of 1:2). This method proposed by Akbani et al. [2] appears to be particularly adapted to the application context of our work. In [2], different under and oversampling strategies are compared. According to the results, SVM with the undersampling strategy provides the most significant performance gain over standard two-class SVMs. In addition, the efficiency of the training process is improved as a result of the reduced training data and, hence, training is easily scalable to large datasets similar to the ones used in the context of our work.

In the test phase, the main challenge is to combine the classification results of the violence models. We perform a classifier selection to solve this. More specifically, we first determine the nearest cluster to a video segment of the test set using Euclidean distance measures. Once the classifier for the video sample is determined, the output of the chosen model is used as the final prediction for that video sample. An overview of the test phase of our method is presented in Fig. 11.4.



**Fig. 11.4** An overview of the test phase of our classification method. Each video segment is of length 0.6 s

## 11.4 Performance Evaluation

The experiments presented in this section aim at comparing the discriminative power of low and mid-level audio-visual representations and feature space partitioning through clustering. A direct comparison of our results with other works discussed in Sect. 11.2 is not straightforward due to the differences in the definition of “violence” in published works. However, we compare our method with one of the methods in the MediaEval VSD task of 2013 which also sticks to the same “violence” definition and provides evaluation results at the video segment level.

### 11.4.1 Dataset and Ground Truth

We used two different types of dataset in our experiments: (1) a set of 24 Hollywood movies which were the movies of the MediaEval 2013 VSD task (the “Hollywood movie dataset”), and (2) a set of 86 short YouTube web videos under Creative Commons licenses which were the short web videos of the MediaEval 2014 VSD task (the “Web video dataset”). The 24 movies of the Hollywood movie dataset are from different genres (ranging from extremely violent movies to movies without violence). Each movie is split in a multitude of video segments, where each video segment is of length 0.6 s. In total, the Hollywood movie dataset consists of 289,699 video segments, where each video segment is labeled as violent or nonviolent.

A total of 17 movies from the Hollywood set are dedicated to the training process: *Armageddon*, *Billy Elliot*, *Eragon*, *Harry Potter 5*, *I am Legend*, *Leon*, *Midnight Express*, *Pirates of the Caribbean 1*, *Reservoir Dogs*, *Saving Private Ryan*, *The Sixth Sense*, *The Wicker Man*, *The Bourne Identity*, *The Wizard of Oz*, *Dead Poets Society*, *Fight Club* and *Independence Day*. The remaining 7 movies—*Fantastic*

**Table 11.1** The characteristics of training and test movies of the Hollywood movie dataset (The number of movies and video segments, the number and percentage of violent and nonviolent video segments)

Dataset	Movies	Video segments	Violent	Nonviolent
Train	17	201,216	24,517 (12 %)	176,699 (88 %)
Test	7	88,483	11,594 (13 %)	76,889 (87 %)
Total	24	289,699	36,111 (12.5 %)	253,588 (87.5 %)

*Four 1*, *Fargo*, *Forrest Gump*, *Legally Blond*, *Pulp Fiction*, *The God Father 1* and *The Pianist*—serve as the test set for the main task which is to detect violence in Hollywood movies. In terms of number of video segments, the training set (17 movies) consists of 201,216 video segments and the test set (7 movies) consists of 88,483 video segments. Table 11.1 presents the main characteristics of the dataset in more detail. The movies of the training and test sets were selected in such a manner that both training and test data contain movies of variable violence levels (extreme to none). On average, in both datasets, around 12.5 % of segments are annotated as violent.

The ground truth of the Hollywood dataset was generated by nine human assessors, partly by developers and partly by potential users. Violent movie segments are annotated at the frame level. Automatically generated shot boundaries with their corresponding key frames are also provided for each movie. A detailed description of the Hollywood dataset and the ground truth generation are given in [9]. For the generalization task which is to detect violence in short web videos, the ground truth was created by several human assessors<sup>4</sup> who followed the subjective definition of violence as explained in Sect. 11.1. A detailed description of the Web video dataset and the ground truth generation are given in [31].

### 11.4.2 Experimental Setup

We employed the MIR Toolbox v1.4<sup>5</sup> to extract the MFCC features (13-dimensional). Frame sizes of 40 ms without overlap are used to align with the 25 fps frames. The Matlab toolbox<sup>6</sup> provided by Uijlings et al. [32] was used to extract dense HoG and HoF features. Features are extracted as explained in Sect. 11.3.

We employed the SPAMS toolbox<sup>7</sup> in order to compute sparse codes which are used for the generation of the mid-level audio and visual representations.

<sup>4</sup> Annotations were made available by *Fudan University*, *Vietnam University of Science*, and *Tech-nicolor*.

<sup>5</sup> <https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/>.

<sup>6</sup> <http://homepages.inf.ed.ac.uk/juijling/index.php?page=software/>.

<sup>7</sup> <http://spams-devel.gforge.inria.fr/>.

The VLFeat<sup>8</sup> open source library is used to perform  $k$ -means clustering ( $k = 10$  in this work).

We trained the two-class SVMs with an RBF kernel using libsvm<sup>9</sup> as the SVM implementation. Training was performed using audio and visual features extracted at the video segment level. SVM parameters were optimized by fivefold cross-validation on the training data.

### 11.4.3 Evaluation Metrics

We used two different evaluation metrics in our evaluation: (1) *average precision at 100 (AP@100)* which is the official metric used in the MediaEval 2013 VSD task, and (2) *average precision (AP)* which is the official metric used in the MediaEval 2014 VSD task. Although the AP@100 metric is no longer the official metric of the MediaEval VSD task, we think that providing a ranked list of violent video shots to the user is still important for our use case. Additionally, including the AP@100 metric allows a comparison with potential other works which would present their results based on AP@100 solely.

### 11.4.4 Results and Discussion

Table 11.2 reports the mean AP and AP@100 metrics on the Hollywood movie dataset. We observe that the mid-level audio representation based on MFCC and sparse coding provides promising performance in terms of average precision and outperforms all other representations that we use in this work. We also note that the performance is further improved by fusing these mid-level audio cues with low and mid-level visual cues at the decision level by linear fusion.

Table 11.3 reports the mean AP and AP@100 metrics on the Web video dataset. We observe results which are similar to the ones obtained on the Hollywood movie dataset (Table 11.2). We used the same violence detection models which were trained using the 17 Hollywood movies, and evaluated these models on short Web videos. The results in terms of AP and AP@100 are still encouraging and even demonstrate superior results compared to the ones obtained on the Hollywood movie dataset. Therefore, we can conclude that our violence detection method generalizes particularly well to other types of video content not used for training the models. Another interesting observation is that affect-related color features seem to provide better results in terms of AP metrics on the Web video dataset in comparison to the Hollywood movie dataset. One final remark is that the linear fusion of audio-visual

---

<sup>8</sup> <http://www.vlfeat.org/>.

<sup>9</sup> <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.



**Table 11.2** The Mean Average Precision (MAP) and MAP@100 of our method with different video representations (i.e., *audio-only* where we use MFCC-based mid-level audio representations, *visual-only* where HoG- and HoF-based mid-level features and ViF descriptors are used, *affect-only* where we use affect-related color features, and *multimodal* where the previous three representations are linearly fused at the decision level) on the Hollywood movie dataset

Dataset	MAP	MAP@100
<i>Audio-only</i>	0.363	0.476
<i>Visual-only</i>	0.327	0.439
<i>Affect-only</i>	0.209	0.140
<i>Multimodal</i>	0.422	0.539

**Table 11.3** The Mean Average Precision (MAP) and MAP@100 of our method with different video representations (i.e., *audio-only* where we use MFCC-based mid-level audio representations, *visual-only* where HoG- and HoF-based mid-level features and ViF descriptors are used, *affect-only* where we use affect-related color features, and *multimodal* where the previous three representations are linearly fused at the decision level) on the Web video dataset

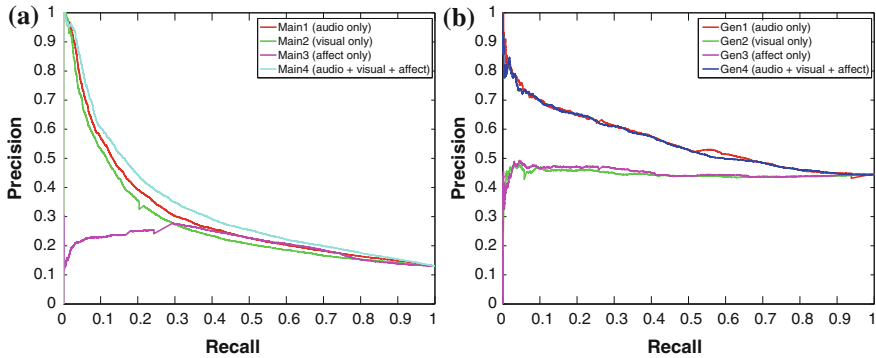
Dataset	MAP	MAP@100
<i>Audio-only</i>	0.582	0.582
<i>Visual-only</i>	0.478	0.478
<i>Affect-only</i>	0.495	0.495
<i>Multimodal</i>	0.567	0.567

features does not help improving the performance in the case of short web videos. This might be the consequence of the weights used in the fusion process, as more weight is given to *visual-only* representations. On the contrary, affect-related features perform better in web videos.

In order to allow the interested reader to have a closer look at the evaluation results, we provide movie-specific MAP@100 values of our best-performing method which is the one with the multimodal representation (Table 11.4). One significant point

**Table 11.4** The movie-specific MAP@100 values of our best-performing method (i.e., the one with a *multimodal* representation) on the Hollywood movie dataset. Each video segment is of length 0.6 s

Movie name	MAP@100	# of Violent video segments
<i>Fantastic Four 1</i>	0.615	2,102
<i>Fargo</i>	0.416	1,426
<i>Forrest Gump</i>	0.548	1,139
<i>Legally Blond</i>	0.000	0
<i>Pulp Fiction</i>	0.992	3,720
<i>The God Father 1</i>	0.290	987
<i>The Pianist</i>	0.910	2,220



**Fig. 11.5** Precision-Recall curves **a** on the Hollywood movie dataset (Area Under Curve (AUC) for Main1: 0.2985, Main2: 0.2757, Main3: 0.2066, and Main4: 0.3255), and **b** on the Web video dataset (AUC for Gen1: 0.5606, Gen2: 0.4447, Gen3: 0.4500, and Gen4: 0.5571) using different representations (Main/Gen1: *audio-only*, Main/Gen2: *visual-only*, Main/Gen3: *affect-only* and Main/Gen4: *multimodal*)

**Table 11.5** The Mean Average Precision (MAP) and MAP@100 of our best-performing method (i.e., the one with a *multimodal* representation), the work of Penet et al. [30] and an SVM-based unique violence detection model (i.e., no feature space partitioning) on the Hollywood movie dataset

Method	MAP	MAP@100
Our method (multimodal)	0.422	0.539
Penet et al. [30]	0.353	0.448
SVM-based unique violence detection model	0.257	0.356

which can be inferred from the overall results is that the average precision variation of the method is high for movies of varying violence levels.

In Fig. 11.5, the precision-recall (PR) curves of our method with different video representations are provided. As seen from the resulting PR curves, our method performs better on short web videos (Fig. 11.5b).

Table 11.5 provides a comparison of our best performing method (i.e., the one with a *multimodal* representation) in terms of MAP and MAP@100 metrics with the method introduced in [30] and an SVM-based unique violence detection model (i.e., a model where no feature space partitioning is performed). We can conclude that our method provides promising results and more importantly, outperforms the SVM-based detection method where the feature space is not partitioned and all violent and nonviolent samples are used to build a unique model.

Finally, we can observe from the overall results provided in this section that our method performs better when violent content is better expressed in terms of audio features (a typical example would be a gun shot scene). This is an indication that we need more discriminative visual representations for detecting violent content in movies and short web videos to further improve the performance of our method.

## 11.5 Application

We present in this section a browser-based visualization tool that allows users to explore movies and online videos based on the detected violence levels. In this tool, currently, only the objective violence definition of the MediaEval VSD task is used to model violence. The system offers the visualization of annotations and results of the MediaEval 2013 VSD task [9] and can interactively download and analyze content from video hosting sites such as YouTube.

The development and evaluation of VSD creates the need for a detailed visualization to assess the strengths and weaknesses of algorithms. Our visualization tool [27] consists of three parts: the *Ranked List* view shows the results on the test set of the MediaEval 2013 VSD task, the *Annotations* view shows the annotations of the MediaEval 2013 VSD training set and the *Online Analysis* carries out our analysis pipeline [1] to arbitrary online videos.

### 11.5.1 The Method

Among the plurality of audio features, MFCCs are shown to be indicators of the excitement level of video segments [36]. Therefore, we employ them as low-level audio features. For the representation of video segments, we use mid-level audio features based on MFCCs in a BoAW scheme. We apply the BoAW approach with two different coding schemes; as an alternative to sparse coding (introduced in Sect. 11.3), we also carried out vector quantization. We train a pair of two-class SVMs in order to learn violence models using both mid-level feature representations. Normally, in a basic SVM, only class labels or scores are output. The class label results from thresholding the score, which is not necessarily a probability measure. The scores output by the SVM are converted into probability estimates using the method explained in [34].

### 11.5.2 Ranked List

The user first selects the run (algorithm and parameters), of which the results will be visualized. The user can also select a specific test movie or the whole test set. The *Ranked List* view (Fig. 11.6) then shows the thumbnails of all segments with an overlay of the violence score (i.e., the probability of violence), time information and a notice whether the classification matches the ground truth. If a segment is classified as violent, the thumbnail is highlighted with an orange frame around it. This enables the user to interpret the results easily and quickly. A click on the thumbnail plays the given segment without leaving the *Ranked List* view. The user can sort the list by the violence scores returned by the algorithm, or can sort it by time to see the classification results chronologically from the beginning to the end of the movie. We also added a button which, when pressed, jumps to a random part of the list to enable a more dynamic exploration experience.



Fig. 11.6 *Ranked List* view of our visualization tool

### 11.5.3 Annotations

The training set of the MediaEval 2013 VSD task provides annotations for 18 Hollywood movies. The annotations mark the presence of audio, visual and audio-visual concepts such as explosions, gunshots, screams, blood, fights, car chases, fire, firearms, cold weapons and gore. The user can query any or all movies for any of these concepts (e.g., show all segments with fire in *Saving Private Ryan*). The annotations are then displayed in a view (Fig. 11.7) similar to the one of the *Ranked List*.

### 11.5.4 Online Analysis

The *Online Analysis* (Fig. 11.8) executes our VSD pipeline to any video hosted by YouTube (or any other site supported by the youtube-dl script). After the user entered the URL, the video is downloaded, transcoded, and split into segments. The MFCC feature vectors of the audio of each segment are subsequently computed and used to build mid-level features with sparse coding and vector quantization as explained in [1]. Both mid-level feature representations are used to classify the segment and produce two violence scores. Even though our methods only use audio features, the *Online Analysis* pipeline can be applied to any method using audio, visual, or audio-visual features. In addition to the *Ranked List* view, the *Online Analysis* produces a

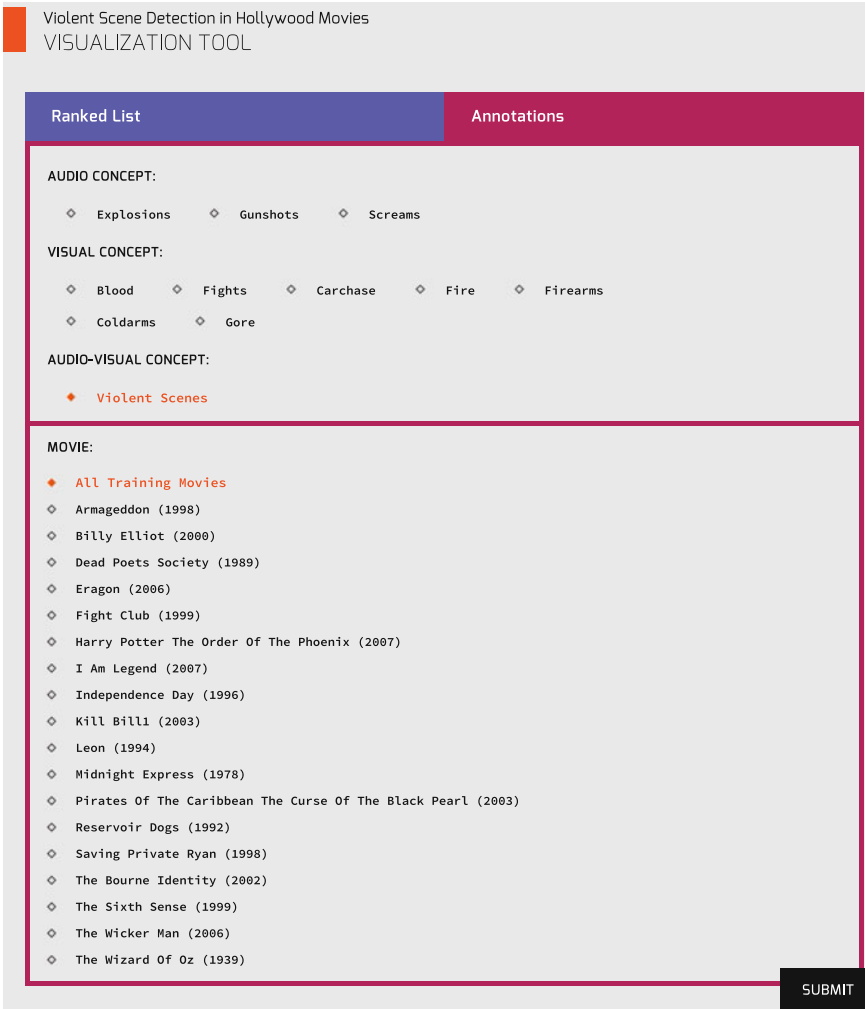
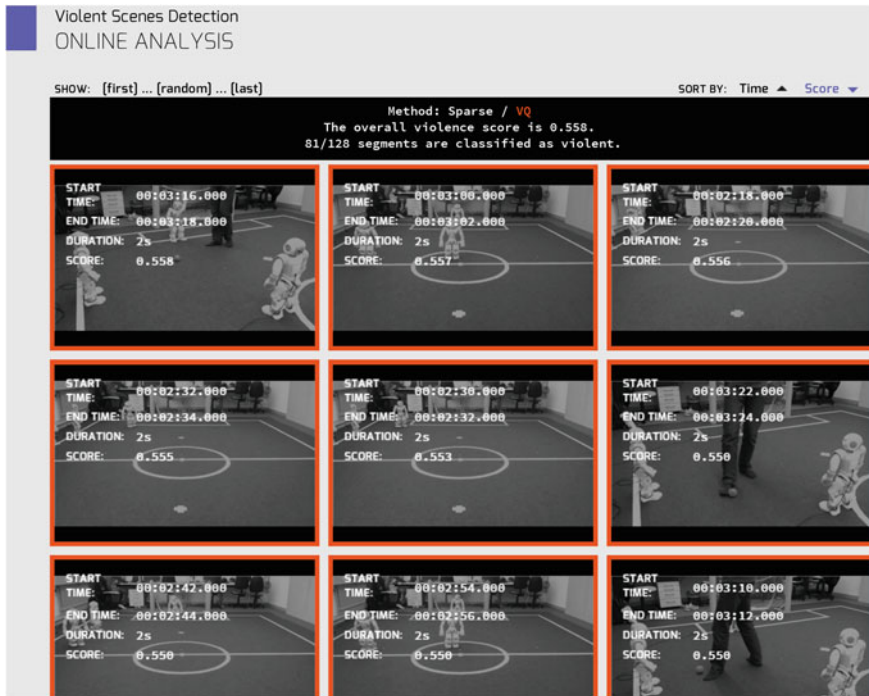


Fig. 11.7 Annotations view of our visualization tool

summary box with the maximum violence score and the number of violent segments. The results are cached so that a query of a previously seen video can return the results immediately without downloading or classifying again.

### 11.5.5 Discussion

On one hand, our audio-based method is able to suitably detect violent content such as fights and disasters with explosions. Video segments which contain no excitement



**Fig. 11.8** Screenshot of *Online Analysis* view of our visualization tool (showing classification results)

(e.g., containing a man giving a speech or featuring strong music in the background) are also easily classified as nonviolent. On the other hand, the method wrongly classifies a video segment as violent when the segment contains very strong sounds or exciting moments such as a plane taking off or a bell ringing loudly. The most challenging violent segments to detect are the ones which are “violent” according to the objective definition of violence given in the MediaEval VSD task, but which actually contain only actions such as self-injuries, or other moderate actions such as an actor pushing or hitting slightly another actor. Our method is also unable to detect violent video segments which are “violent” according to the objective definition of violence, but which contain no audio cues exploitable for the identification of violence (e.g., a man bleeding). More detailed discussion on the performance of our method is given in [1].

## 11.6 Conclusions and Future Work

In this chapter, we presented an approach for the detection of violent content in movies and short web videos at the video segment level. We employed low and mid-level audio-visual features to represent videos. The mid-level audio and visual

representations are based on BoW where we first extract audio features (MFCC) and visual features (dense HoG and HoF), and subsequently apply sparse coding on each feature descriptor separately. We used ViF and affect-related color features as low-level visual representation of videos.

Since “violence” is a very diverse concept, we first performed feature space partitioning through clustering video segments instead of learning a unique violence detection model. We then learned a different model for each violence subconcept. In order to combine the classification results of the violence models in the test phase, we performed a classifier selection. More specifically, we labeled a video segment with the output of the classifier whose cluster center is closest to the video segment in terms of Euclidean distance.

To demonstrate the wide applicability of our solution, we evaluated our method on two different datasets: one dataset of Hollywood movies and one dataset of user-generated videos.

We showed that the mid-level audio representation based on MFCC and sparse coding provides very promising performance in terms of AP and AP@100 metrics and also outperforms visual representations that we used in this work. We also fused these mid-level audio cues with low and mid-level visual cues at the decision level using linear fusion for further improvement and achieved better results than unimodal video representations in terms of the AP metrics.

Different from Hollywood movies, user-generated videos are more challenging, since they are not professionally edited, e.g., in order to enhance dramatic scenes. We also demonstrated the performance of our system on the challenging web video dataset which contains short web videos from YouTube. The evaluation results on the short web videos were similar to the ones on the Hollywood movie dataset and hence, showed that our violence detection method generalizes well to different types of video content.

We observed from the overall evaluation results that our method performs better when violent content is better expressed in terms of audio features. Hence, as a future work, we need to extend/improve our visual representation set with more discriminative representations. Another possibility for future work is to further investigate the feature space partitioning concept and optimize the distribution or number of subconcepts in order to enhance the classification performance of our method.

**Acknowledgments** The research leading to these results has received funding from the European Community FP7 under grant agreement number 261743 (NoE VideoSense). We would like to thank *Technicolor* (<http://www.technicolor.com/>) for providing the ground truth, video shot boundaries, and the corresponding keyframes which have been used in this work. Our thanks also go to *Fudan University* and *Vietnam University of Science* for providing the ground truth of the Web video dataset.

## References

1. E. Acar, F. Hopfgartner, S. Albayrak, Detecting violent content in Hollywood Movies by mid-level audio representations, in *CBMI 2013* (IEEE 2013)
2. R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets. *Mach. Learn.: ECML* **2004**, 39–50 (2004)
3. S. Andrews, I. Tsochantaris, T. Hofmann, Support vector machines for multiple-instance learning. *Adv. Neural Inf. Process. Syst.* **15**, 561–568 (2002)
4. B.J. Bushman, L.R. Huesmann, Short-term and long-term effects of violent media on aggression in children and adults. *Arch. Pediatr. Adolesc. Med.* **160**(4), 348 (2006)
5. L.-H. Chen, H.-W. Hsu, L.-Y. Wang, C.-W. Su, Horror video scene recognition via multiple-instance learning, in *ICASSP* (2011)
6. L.-H. Chen, H.-W. Hsu, L.-Y. Wang, C.-W. Su, Violence detection in movies, in *2011 Eighth International Conference on Computer Graphics, Imaging and Visualization (CGIV)* (IEEE, 2011), pp. 119–124
7. F.D.M. de Souza, G.C. Chávez, E.A. do Valle, A. de A. Araujo. Violence detection in video using spatio-temporal features, in *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (IEEE, 2010), pp. 224–230
8. C.-H. Demarty, B. Ionescu, Y.-G. Jiang, V.L. Quang, M. Schedl, C. Penet, Benchmarking violent scenes detection in movies, in *2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)* (IEEE, 2014), pp. 1–6
9. C.-H. Demarty, C. Penet, M. Schedl, B. Ionescu, Vu L. Quang, Yu-G. Jiang, The MediaEval 2013 affect task: violent scenes detection, in *Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, 18–19 October 2013
10. N. Derbas, G. Quénot, Joint audio-visual words for violent scenes detection in movies, in *Proceedings of International Conference on Multimedia Retrieval* (ACM, 2014), p. 483
11. X. Ding, B. Li, W. Hu, W. Xiong, Z. Wang, Horror video scene recognition based on multi-view multi-instance learning, in *Computer Vision-ACCV 2012* (Springer, 2013), pp. 599–610
12. B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression. *Ann. Stat.* **32**(2), 407–499 (2004)
13. T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, S. Theodoridis, Violence content classification using audio features. *Adv. Artif. Intell.* **3955**, 502–507 (2006)
14. T. Giannakopoulos, A. Makris, D. Kosmopoulos, S. Perantonis, S. Theodoridis, Audio-visual fusion for detecting violent scenes in videos. *Artif. Intell.: Theor. Model. Appl.* **6040**, 91–100 (2010)
15. Y. Gong, W. Wang, S. Jiang, Q. Huang, W. Gao, Detecting violent scenes in movies by auditory and visual cues. *Adv. Multimed. Inf. Process.-PCM* **2008**, 317–326 (2008)
16. S. Goto, T. Aoki, Violent scenes detection using mid-level violence clustering. *Comput. Sci.* (2014)
17. D. Hasler, S.E. Suesstrunk, Measuring colorfulness in natural images, in *Electronic Imaging 2003*. International Society for Optics and Photonics, pp. 87–95 (2003)
18. T. Hassner, Y. Itcher, O. Kliper-Gross, Violent flows: real-time detection of violent crowd behavior, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, 2012), pp. 1–6
19. H. He, E.A. Garcia, Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
20. B.K. Horn, B.G. Schunck, Determining optical flow, in *1981 Technical Symposium East*. International Society for Optics and Photonics, pp. 319–331 (1981)
21. B. Ionescu, J. Schlüter, I. Mironica, M. Schedl, A naive mid-level concept-based fusion approach to violence detection in Hollywood Movies, in *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval* (ACM, 2013), pp. 215–222
22. O. Kliper-Gross, T. Hassner, L. Wolf, The action similarity labeling challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(3), 615–621 (2012)



23. I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008* (IEEE, 2008), pages 1–8
24. J. Lin, W. Wang, Weakly-supervised violence detection in movies with audio and video based co-training. *Adv. Multimed. Inf. Process.-PCM* **2009**, 930–935 (2009)
25. J. Machajdik, A. Hanbury, Affective image classification using features inspired by psychology and art theory, in *Proceedings of the International Conference on Multimedia* (ACM, 2010), pp. 83–92
26. J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.* **11**, 19–60 (2010)
27. D. Maniry, E. Acar, F. Hopfgartner, S. Albayrak, A visualization tool for violent scenes detection, in *Proceedings of ACM Conference on Multimedia Retrieval, ICMR'14* (ACM, 2014), pp. 522–523
28. E.B. Nievas, O.D. Suarez, G.B. García, R. Sukthankar, Violence detection in video using computer vision techniques, in *Computer Analysis of Images and Patterns* (Springer, 2011), pp. 332–339
29. C. Penet, C.-H. Demarty, G. Gravier, P. Gros, Multimodal information fusion and temporal integration for violence detection in movies, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2012), pp. 2393–2396
30. C. Penet, C.-H. Demarty, G. Gravier, P. Gros et al., Technicolor/inria team at the mediaeval 2013 violent scenes detection task. *MediaEval 2013 Working Notes* (2013)
31. M. Sjöberg, B. Ionescu, Y.-G. Jiang, V.L. Quang, M. Schedl, C.-H. Demarty. The MediaEval 2014 affect task: violent scenes detection, in *Working Notes Proceedings of the MediaEval 2014 Workshop*, Barcelona, Spain, 16–17 October 2014
32. J.R.R. Uijlings, I. Duta, N. Rostamzadeh, N. Sebe, Realtime video classification using dense HOF/HOG, in *Proceedings of International Conference on Multimedia Retrieval* (ACM, 2014), p. 145
33. H.L. Wang, L.F. Cheong, Affective understanding in film. *IEEE Trans. Circuits Syst. Video Technol.* **16**(6), 689–704 (2006)
34. W. Ting-Fan, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.* **5**, 975–1005 (2004)
35. L. Xu, C. Gong, J. Yang, Q. Wu, L. Yao, Violent video detection based on MoSIFT feature and sparse coding, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2014), pp. 3538–3542
36. M. Xu, N.C. Maddage, C. Xu, M. Kankanhalli, Q. Tian, Creating audio keywords for event detection in soccer video, in *ICME'03* (IEEE 2003)
37. R. Yan, M. Naphade, Semi-supervised cross feature learning for semantic concept detection in videos, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1 (IEEE, 2005), pp. 657–663
38. L. Yeffet, L. Wolf, Local trinary patterns for human action recognition, in *2009 IEEE 12th International Conference on Computer Vision* (IEEE, 2009), pp. 492–497

# Chapter 12

## Discovery of Driving Behavior Patterns

Stephan Spiegel

**Abstract** Given a set of time series, our goal is to identify prototypes that cover the maximum possible amount of occurring subsequences regardless of their order. This scenario appears in the context of the automotive industry, where the objective is to determine operational profiles that comprise frequently recurring driving behavior patterns. This problem can be solved by clustering, however, standard distance measures such as the dynamic time warping distance might not be suitable for this task, because they aim at capturing the cost of aligning two time series rather than rewarding pairwise occurring patterns. In this work, we propose a novel time series distance measure, based on the theoretical foundation of recurrence plots, which enables us to determine the (dis)similarity of multivariate time series that contain segments of similar trajectories at arbitrary positions. We use recurrence quantification analysis to measure the structures observed in recurrence plots and to investigate dynamical properties, such as determinism, which reflect the pairwise (dis)similarity of time series. In experiments on real-life test drives from Volkswagen, we demonstrate that clustering multivariate time series using the proposed recurrence plot-based distance measure results in prototypical test drives that cover significantly more recurring patterns than using the same clustering algorithm with dynamic time warping distance.

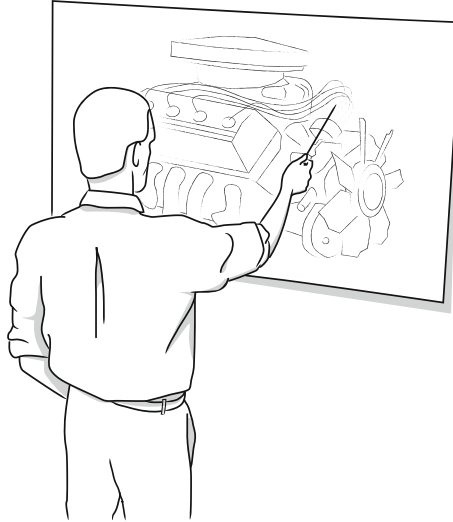
### Reduced Emissions: The Drive Green Scenario

Since Steven entered preliminary school at the early age of five, he always loved to spend the Sunday afternoons in solitary reading books about theoretical and applied mathematics. None of this has changed in the last 30 years and nobody in his family was surprised about his recent decision to work as an associate researcher at one of the leading car manufacturers. Although Steven is passionate about modern technology, his father thought him to respect and preserve nature. When Steven was a teenager, he and his dad often went hiking to watch birds at one of the small lakes in the forest

---

S. Spiegel (✉)  
Technische Universität Berlin, Berlin, Germany  
e-mail: stephan.spiegel@dai-labor.de

of the nearby mountains. On their hikes, Steven was told a lot about the local wildlife and everything his father had learned from the scouts when he was a boy. At that time, Steven decided that he would dedicate his adult life to preserve that untouched nature for his own children and the generations after. He soon realized that he could use his natural talent for mathematics to solve environmental problems and to contribute to society. His current position in a research and development department for internal combustion engines allows Steven to take an active part in reducing the emission of greenhouse gases, which have long been known to cause global warming.



Recent political debates about climate change have led to profound environmental regulations that limit the maximum permissible emission for vehicles on European roads. To avoid severe sanctions, the automotive industry has to ensure that their newly developed engines go below the allowed limit of exhaust fumes. However, automobile manufacturer face the problem that a nationwide survey of their car fleet would result in exorbitant cost and effort. Instead of that, Steven proposed to draw samples from test drives which characterize specific combinations of driver, vehicle, and route. In further investigations, the derived operational profiles could be used to simulate typical driving behavior and to spot-check against the newly introduced emission regulations. Although the top management is not fully convinced, Steven has strong support from his own rows and was invited to present his detailed proposal to the decision-making department.

Steven explains that he aims at deriving characteristic operational profiles of their new vehicle fleet by means of several controlled test drives. His idea is to record multiple engine parameters during test drives to discover driving behavior patterns which typically occur under certain circumstances. The main challenge is to develop an algorithm that is able to efficiently and effectively compare the high-dimensional measurements with regard to co-occurring temporal patterns. Test drives with a high number of typical driving behavior patterns may than be used for engine simulations

and emission evaluations. Due to the fact that Steven's approach is less expensive and time consuming than a nationwide survey, the decision committee felt positive about his idea and assigned him to lead the research project.

Leading a team of researchers in developing more efficient and environment-friendly combustion engines does not only mean a real breakthrough in Steven's career, but also a huge success on a personal level. Since his childhood, he always dreamed of finding a way to use own skills to do something for the benefit of the nature his father taught him to love. This is a unique opportunity for Steven to make a positive impact on the environment of future generations. He wants his children and grandchildren to enjoy and experience nature in the same way as he did as a kid. With this in mind, Steven accepts the challenge of his lifetime.

## 12.1 Introduction

Clustering of times series data is of pivotal importance in various applications [9] such as, for example, seasonality patterns in retail [13], electricity usage profiles [17], DNA microarrays [26], and fMRI brain activity mappings [39]. A crucial design decision of any clustering algorithm is the choice of (dis)similarity function [1, 14]. In many clustering applications, the underlying (dis)similarity function measures the cost of aligning time series to one another. Typical examples of such functions include the DTW and the Euclidean distance [4, 10, 27].

Alignment-based (dis)similarity functions, however, seem not to be justified for applications, where two time series are considered to be similar, if they share common or similar subsequences of variable length at arbitrary positions [2, 16, 28, 40]. A real-life example for such an application comes from the automotive industry, where test drives of vehicles are considered to be similar, if they share similar driving behavior patterns, i.e., engine behavior or drive maneuvers, which are described by the progression of multiple vehicle parameters over a certain period of time [33, 35]. In this scenario, the order of the driving behavior patterns does not matter [32], but the frequency with which the patterns occur in the contrasted time series.

Recent work [5] on time series distance measures suggests to neglect irrelevant and redundant time series segments, and to retrieve subsequences that best characterize the real-life data. Although subsequence clustering is a tricky endeavor [12], several studies [2, 7, 16, 28, 40] have demonstrated that in certain circumstances ignoring sections of extraneous data and keeping intervals with high discriminative power contributes to cluster centers that preserve the characteristics of the data sequences. Related concepts that have been shown to improve clustering results include time series motifs [2, 16], shapelets [28, 40], and discords [7].

In this contribution, we propose to adopt recurrence plots (RPs) [18, 21, 22] and related recurrence quantification analysis (RQA) [19, 20, 23] to measure the similarity between multivariate time series that contain segments of similar trajectories at arbitrary positions in time [32]. We introduce the concept of joint cross recurrence plots (JCRPs), an extension of traditional RPs, to visualize and investigate

multivariate patterns that (re)occur in pairwise compared time series. In dependence on JCRPs and known RQA measures, such as determinism, we define a **Recurrence** plot-based (RRR) distance measure, which reflects the proportion of time series segments with similar trajectories or recurring patterns, respectively.

In order to demonstrate the practicability of our proposed recurrence plot-based distance measure, we conduct experiments on both synthetic time series and real-life vehicular sensor data [32, 33, 35]. The results show that, unlike commonly used (dis)similarity functions, our proposed distance measure is able to (i) determine cluster centers that preserve the characteristics of the data sequences and, furthermore, (ii) identify prototypical time series that cover a high amount of recurring patterns.

The rest of the chapter is organized as follows. In Sect. 12.2, we state the general problem being investigated. Related work is discussed in Sect. 12.3. Subsequently, we introduce traditional recurrence plots as well as various extensions in Sect. 12.4. Recurrence quantification analysis and corresponding measures are discussed in Sect. 12.5. Our proposed recurrence plot-based distance measure and respective evaluation criteria are introduced in Sect. 12.6. Possible ways to reduce the computational complexity of our introduced distance measure are offered in Sects. 12.7 and 12.8. Our experimental results are presented and discussed in Sect. 12.9. In addition, Sect. 12.10 presents BestTime, a platform-independent Matlab application with graphical user interface, which enables us to find representative that best comprehend the recurring temporal patterns contained in a certain time series dataset. Finally, we conclude with future work in Sect. 12.11.

## 12.2 Problem Statement

Car manufacturers aim to optimize the performance of newly developed engines according to operational profiles that characterize recurring driving behavior. To obtain real-life operational profiles for exhaust simulations, Volkswagen (VW) collects data from test drives for various combinations of driver, vehicle, and route.

Given a set  $\mathcal{X} = \{X_1, X_2, \dots, X_t\}$  of  $t$  test drives, the challenge is to find a subset of  $k$  prototypical time series  $\mathcal{Y} = \{Y_1, \dots, Y_k\} \in \mathcal{X}$  that best comprehend the recurring (driving behavior) patterns found in set  $\mathcal{X}$ . Test drives are represented as multivariate time series  $X = (x_1, \dots, x_n)$  of varying length  $n$ , where  $x_i \in \mathbb{R}^d$  is a  $d$ -dimensional feature vector summarizing the observed measurements at time  $i$ . A *pattern*  $S = (x_s, \dots, x_{s+l-1})$  of  $X = (x_1, \dots, x_n)$  is a subsequence of  $l$  consecutive time points from  $X$ , where  $l \leq n$  and  $1 \leq s < s+l-1 \leq n$ . Assuming two time series  $X = (x_1, \dots, x_n)$  and  $Y = (y_1, \dots, y_m)$  with patterns  $S = (x_s, \dots, x_{s+l-1})$  and  $P = (y_p, \dots, y_{p+l-1})$  of length  $l$ , we say that  $S$  and  $P$  are *recurring patterns* of  $X$  and  $Y$  if  $d(S, P) \leq \epsilon$ , where  $d : X \times X \rightarrow \mathbb{R}^+$  is a (dis)similarity function and  $\epsilon$  is a certain similarity threshold. Note that recurring patterns of  $X$  and  $Y$  may occur at arbitrary positions and in different order.

Since we aim to identify  $k$  prototypical time series that (i) best represent the set  $\mathcal{X}$  and (ii) are members of the set  $\mathcal{X}$ , one can employ the  $k$ -medoid clustering algorithm.

## 12.3 Related Work

The main goal of clustering is to organize unlabeled data into homogeneous groups that are clearly separated from each other. In general, clustering involves the clustering algorithm, the similarity or rather distance measure, and the evaluation criterion. Clustering algorithms are categorized into partitioning, hierarchical, density-based, grid-based, and model-based methods. All of these clustering algorithms can be applied for static and temporal data [14]. In the following, we discuss important considerations, common pitfalls, successful applications, and recent developments in time series clustering.

**Time Series Clustering.** Unlike static data, temporal data evolves over time and therefore requires special handling. One could either modify the existing clustering algorithms to handle time series data or convert the time series into a form that can be directly clustered. The former approach works with the raw time series, and the major modification lies in replacing the distance/similarity measure. The latter approach converts the raw time series either into feature vectors or model parameters, and then applies conventional clustering algorithms. Thus, time series clustering approaches can be categorized into raw-data-based, feature-based, and model-based methods [14].

**Time Series Representation.** In this study, we mainly focus on clustering methods that work with raw data, in particular multivariate time series with same sample rate. Clustering time series only differs from conventional clustering in how to compute the similarity between data objects [14]. Therefore, the key is to understand the unique characteristics of the time series and then to design an appropriate similarity measure accordingly. For instance, Meesrikamolkul et al. [25] have proposed a novel method which combines the widely used k-means clustering algorithm with the Dynamic Time Warping distance measure, instead of the traditional Euclidean distance, to study sequences with time shifts. Unlike before, the new method determines cluster centers that preserve the characteristics of the data sequences.

**Distance/Similarity Measures.** Besides Euclidean distance and Dynamic Time Warping distance, commonly used similarity measures include Minkowski distance, Levenshtein distance, Short Time Series distance, Pearson correlation coefficient, cross-correlation-based distances, probability-based distance functions, and many others. The choice of similarity measure depends on whether the time series is discrete-valued or real-valued, uniform or nonuniform sampled, univariate or multivariate, and whether the data sequences are of equal or unequal length [14].

**Distortions and Invariance.** Furthermore, the choice of the time series distance measure depends on the invariance required by the domain. The literature [1] has introduced techniques designed to efficiently measure similarity between time series with invariance to (various combinations of) the distortions of warping, uniform scaling, offset, amplitude scaling, phase, occlusions, uncertainty, and wandering baseline. Recent work [32] has proposed an order-invariant distance which is able to determine the (dis)similarity of time series that exhibit similar subsequences at arbitrary

positions. The authors demonstrate that order invariance is an important consideration for domains such as automotive engineering and smart home environments [33, 35], where multiple sensors observe contextual patterns in their naturally occurring order, and time series are compared according to the occurrence of these multivariate patterns.

**Evaluation Criterion.** Evaluation criteria for clustering are distinguished between known ground truth and unknown ground truth [14]. In case of known ground truth, the similarity between known clusters and obtained clusters can be measured. The most commonly used clustering quality measure for known ground truth is the Rand Index or minor variants of it [40]. In contrast, without prior knowledge the clusters are usually evaluated according to their within-cluster similarity and between-cluster dissimilarity [14]. Various validity indices have been proposed to determine the number of clusters and their goodness. For instance, the index  $I$  has been found to be consistent and reliable, irrespective of the underlying clustering technique and data dimensionality, and furthermore has been shown to outperform the Dunn and David-Bouldin index [24].

**Realistic Assumptions.** The majority of publicly available time series datasets were preprocessed and cleaned before publishing. For instance, the UCR archive [9] contains only time series with equal length, which are mostly snippets of the original data that were retrieved manually. The publication of perfectly aligned patterns of equal length has led to a huge amount of time series classification and clustering algorithms that are not able to deal with real-world data, which contains irrelevant sections. Hu et al. [5] suggest to automatically build a data dictionary, which contains only a small subset of the training data and neglects irrelevant sections and redundancies. The evaluations show that using a data dictionary with a set of retrieved subsequences for each class leads to higher classification accuracy and is several times faster than the compared strawman algorithms. However, one needs to be careful about how to retrieve subsequences, for reasons explained in the following.

**Subsequence Clustering.** Keogh and Lin [12] state that the clustering of time series subsequences is meaningless, referring to the finding that the output does not depend on input, and the resulting cluster centers are close to random ones. In almost all cases the subsequences are extracted with a sliding window, which is assumed to be a quirk in clustering. To produce meaningful results the authors suggest to adopt time series motifs, a concept highly related to clusters. Their experiments demonstrate that motif-based clustering is able to preserve the patterns found in the original time series data [12].

**Time Series Motifs.** Motifs are previously unknown, frequently occurring patterns, which are useful for various time series mining tasks: such as summarization, visualization, clustering and classification of time series [2, 16]. According to the definition [16] a time series motif is a subsequence that comprises all non-trivial matches within a given range. Since the naive (brute-force) approach to motif discovery has quadratic complexity, Lin et al. [16] introduce a new motif discovery algorithm that provides fast exact answers, and faster approximate answers, achieving a speedup of one to two orders of magnitude. In order to reduce the num-

ber of possible candidates of motifs, Chiu et al. [2] propose to omit consecutive subsequences that resemble each other. Furthermore, the set of subsequences in each motif should be mutually exclusive, because otherwise the motifs would be essentially the same. Although normalization techniques are commonly applied to compare time series with different offset and amplitude, Chiu et al. [2] state that these are important characteristics that might prove to be useful to distinguish motifs, because after normalization most subsequences correspond to almost the same upward or downward trend and become indistinguishable.

**Time Series Shapelets.** Most existing methods for time series clustering rely on distances calculated on the shape of the signals. However, time series usually contain a great amount of measurements that do not contribute to the differentiation task or even decrease cluster accuracy. Hence, to cluster time series, we are generally better off ignoring large sections of extraneous data and keeping intervals with high discriminative power. Recent work [28, 40] proposes to use local patterns, so called shapelets, to cluster time series databases. According to the definition [40], a shapelet is a time series snippet that can separate and remove a subset of the data from the rest of the database, while maximizing the separation gap or rather information gain. Although the experiments demonstrate that shapelet-based clustering gives better results than statistical-based clustering of the entire time series, finding optimal shapelets is a nontrivial task, and almost certainly harder than the clustering itself [40]. However, the results underline the importance of ignoring some data to cluster time series in real-world applications under realistic settings.

**Time Series Discords.** Different from motifs or shapelets, time series discords are subsequences of longer time series that are most unusual or rather maximally different to all the rest of the time series subsequences. Keogh et al. [7] have shown that time series discords are particularly attractive as anomaly detectors because they only require one intuitive parameter, namely the length of the subsequences. Furthermore, discords have implications for the time series clustering, cleaning, and summarization.

**Time Series Prototypes.** To sum up, the concepts that may possibly be adapted to identify time series prototypes (as described in our problem statement in Sect. 12.2) include motifs [2, 16] and shapelets [28, 40]. However, in both cases this would require major modifications of the existing algorithm. A straightforward approach to solve the stated problem is presented in the following sections.

## 12.4 Recurrence Plots

Recurrence plots (RPs) are used to visualize and investigate recurrent states of dynamical systems or rather time series [23, 31]. Even though RPs give very vivid and impressive images of dynamical system trajectories, their implicit mathematical foundation is deceptively simple [18]:



$$R_{i,j}^x(\epsilon) = \Theta(\epsilon - \|x_i - x_j\|) \quad x_i \in \mathbb{R}^d, \quad i, j = 1 \dots n \quad (12.1)$$

where  $x$  is a time series of length  $n$ ,  $\|\cdot\|$  a norm and  $\Theta$  the Heaviside function. One of the most crucial parameters of RPs is the recurrence threshold  $\epsilon$ , which influences the formation of line structures [21]. In general, the recurrence threshold should be chosen in a way that noise corrupted observations are filtered out, but at the same time a sufficient number of recurrence structures are preserved. As a rule of thumb, the recurrence rate should be approximately one percent with respect to the size of the plot. For quasiperiodic processes, it has been suggested to use the diagonal line structures to find the optimal recurrence threshold. However, changing the threshold does not preserve the important distribution of recurrence structures [23].

A general problem with standard thresholding methods is that an inappropriate threshold or laminar states cause thick diagonal lines, which basically corresponds to redundant information. Schultz et al. [31] have proposed a local minima-based thresholding approach, which can be performed without choosing any particular threshold and yields in clean RPs of minimized line thickness. But this approach comes with some side effects, e.g., bowed lines instead of straight diagonal lines.

Furthermore, it is important to discuss the definition of recurrences, because distances can be calculated using different norms [18]. Although the  $L_2$ -norm is used in most cases, the  $L_\infty$ -norm is sometimes preferred for relatively large time series with high computational demand [23].

Although traditional RPs only regard one trajectory, we can extend the concept in a way that allows us to study the dynamics of two trajectories in parallel [22]. A cross recurrence plot (CRP) shows all those times at which a state in one dynamical system occurs in a second dynamical system. In other words, the CRP reveals all the times when the trajectories of the first and second time series,  $x$  and  $y$ , visits roughly the same area in the phase space. The data length,  $n$  and  $m$ , of both systems can differ, leading to a nonsquare CRP matrix [19, 21].

$$CR_{i,j}^{x,y}(\epsilon) = \Theta(\epsilon - \|x_i - y_j\|) \quad x_i, y_j \in \mathbb{R}^d, \quad i = 1 \dots n, \quad j = 1 \dots m \quad (12.2)$$

For the creation of a CRP, both trajectories,  $x$  and  $y$ , have to present the same dynamical system with equal state variables because they are in the same phase space. The application of CRPs to absolutely different measurements, which are not observations of the same dynamical system, is rather problematic and requires some data preprocessing with utmost carefulness [21].

In order to test for simultaneously occurring recurrences in different systems, another multivariate extension of RPs was introduced [22]. A joint recurrence plot (JRP) shows all those times at which a recurrence in one dynamical system occurs simultaneously with a recurrence in a second dynamical system. With other words, the JRP is the Hadamard product of the RP of the first system and the RP of the second system. JRPs can be computed from more than two systems. The data length of the considered systems has to be the same. [19, 21].

$$JR_{i,j}^{x,y}(\epsilon^x, \epsilon^y) = \Theta(\epsilon^x - \|x_i - x_j\|) \cdot \Theta(\epsilon^y - \|y_i - y_j\|) \quad (12.3)$$

$$x_i \in \mathbb{R}^{d1}, \quad y_j \in \mathbb{R}^{d2}, \quad i, j = 1 \dots n$$

Such joint recurrence plots have the advantage that the individual measurements can present different observables with different magnitudes or range. They are often used for the detection of phase synchronization [19, 21].

Since this work aims at clustering test drives, which involves pairwise (dis)similarity comparisons of multivariate time series, we propose a combination of joint and cross recurrence plot, namely (JCRP) joint cross recurrence plot. A JCRP shows all those times at which a multivariate state in one dynamical system occurs simultaneously in a second dynamical system.

$$JCR_{i,j}^{x,y}(\epsilon^1, \dots, \epsilon^k) = \Theta(\epsilon^1 - \|x_i^1 - y_j^1\|) \times \dots \times \Theta(\epsilon^k - \|x_i^k - y_j^k\|) \quad (12.4)$$

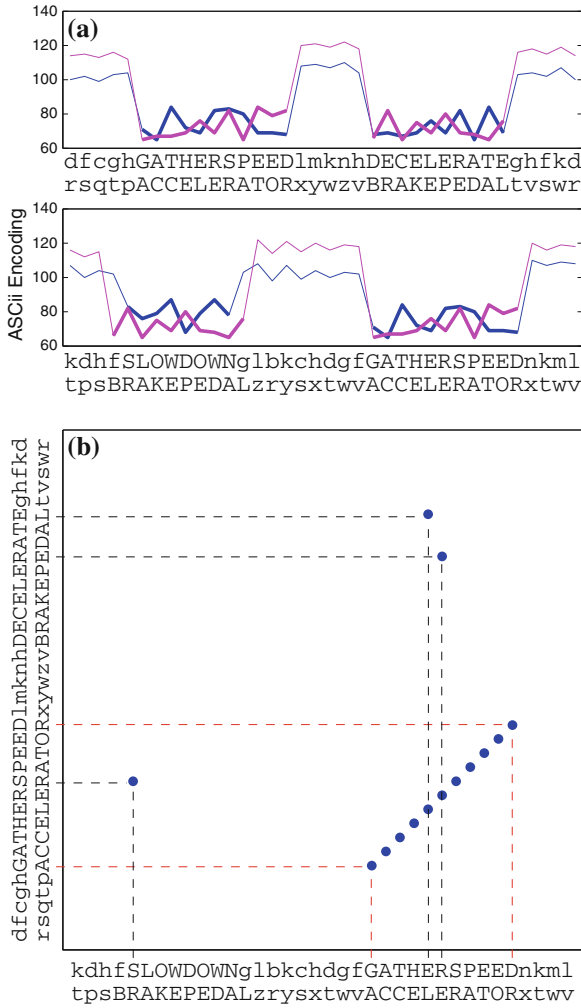
$$x_i, y_j \in \mathbb{R}^d, \quad i = 1 \dots n, \quad j = 1 \dots m$$

For the creation of a JCRP both trajectories,  $x$  and  $y$ , need to have the same dimensionality or number of parameters  $d$ , but can have different length,  $n$  and  $m$ . We shall see that JCRPs are very useful, because they enable us to compare two multivariate systems with the same set of observables that can have different magnitudes. In other words, the introduced JCR notation allows us to determine an  $\epsilon$ -threshold for each individual parameter, which is advantageous for observables with different variance. A toy example for JCRPs is given in the following:

$$x = \begin{cases} \text{dfcghGATHERSPEEDlmknhDECELERATEghfk} \\ \text{rsqtpACCELERATORxyzvwBRAKEPEDALtvsr} \end{cases}$$

$$y = \begin{cases} \text{kdhfSLOWDOWNglbkchdgfGATHERSPEEDnkm} \\ \text{tpsBRAKEPEDALzrysxtwvACCELERATORxtwv} \end{cases}$$

Assume two multivariate time series  $x$  and  $y$  which comprise the speed and accelerator signal recorded during different car drives. Both time series contain multivariate states or rather string sequences that occur in both systems, as demonstrated in Fig. 12.1a. The corresponding JCRP of  $x$  and  $y$ , as illustrated in Fig. 12.1b, shows the times at which a multivariate state occurs simultaneously in both systems. Furthermore, the diagonal line structure in Fig. 12.1b reveals that both trajectories run through a similar region in phase space for a certain time interval. With other words, both systems contain the same multivariate pattern, which represents that the driver hits the ‘ACCELERATOR’ pedal and the vehicle simultaneously ‘GATHERSPEED’. In Sect. 12.5, we discuss how to interpret single recurrence points and diagonal line structures, and explain how to use them to define a distance measure for time series with certain distortions or invariance.



**Fig. 12.1** **a** ASCII decimal encoding of two multivariate time series  $x$  and  $y$  which contain the same pattern or string sequence at different positions in time. **b** Joint cross recurrence plot (JCRP) of time series  $x$  and  $y$ , introduced in **a**, with  $\epsilon = 0$ . The diagonal line structure in the recurrence plot indicates the existence and position of a co-occurring multivariate pattern. The single recurrence points can be considered as noise

## 12.5 Recurrence Quantification

Recurrence quantification analysis (RQA) is used to quantify the structures observed in recurrence plots [21]. RQA is grounded in theory, but possesses statistical utility in dissecting and diagnosing nonlinear dynamic systems across multiple fields of science [38]. The explicit mathematical definition to distinct features in recurrence

plots enables us to analyze signals that are multivariate, nonlinear, nonstationary, and noisy.

The global (large-scale) appearance of a RP can give hints on stationarity and regularity, whereas local (small-scale) patterns are related to dynamical properties, such as determinism [38]. Recent studies have shown that determinism, the percentage of recurrence points that form lines parallel to the main diagonal, reflects the predictability of a dynamical system [21].

Given a recurrence matrix  $R$  with  $N \times N$  entries generated by any of the introduced recurrence plot variations, such as our proposed JCRP, we can compute the determinism  $\text{DET}(\epsilon, l_{\min})$  for a predefined  $\epsilon$ -threshold and a minimum diagonal line length  $l_{\min}$  as followed [19, 21]:

$$\text{DET}(\epsilon, l_{\min}) = \frac{\sum_{l=l_{\min}}^N l \cdot P(\epsilon, l)}{\sum_{i,j=1}^N R_{i,j}(\epsilon)} \quad (12.5)$$

$$P(\epsilon, l) = \sum_{i,j=1}^N \left\{ \begin{aligned} &(1 - R_{i-1,j-1}(\epsilon)) \\ &\times (1 - R_{i+l,j+l}(\epsilon)) \\ &\times \prod_{k=0}^{l-1} R_{i+k,j+k}(\epsilon) \end{aligned} \right\} \quad (12.6)$$

where  $P(\epsilon, l)$  is the histogram of diagonal lines of length  $l$  with respect to a certain  $\epsilon$  neighborhood.

In general, processes with chaotic behavior cause none or short diagonals, whereas deterministic processes cause relatively long diagonals and less single, isolated recurrence points [21, 37]. In respect to JCRPs, diagonal lines usually occur when the trajectory of two multivariate time series segments is similar according to a certain threshold. Since we aim to measure the similarity between time series that contain segments of similar trajectories at arbitrary positions, which in turn cause diagonal line structures, we propose to use determinism as a similarity measure. According to the introduced JCRP approach, a high DET value indicates high similarity or rather a high percentage of multivariate segments with similar trajectory, whereas a relatively low DET value suggests dissimilarity or rather the absence of similar multivariate patterns.

However, data preprocessing like smoothing can introduce spurious line structures in a recurrence plot that cause high determinism value. In this case, further criteria like the directionality of the trajectory should be considered to determine the determinism of a dynamic system, e.g., by using iso-directional and perpendicular RPs [19, 21, 23]. In contrast to traditional recurrence plots, perpendicular recurrence plots (PRPs) consider the dynamical evolution of only the neighborhoods in the perpendicular direction to each phase flow, resulting in plots with lines of the

similar width without spreading out in various directions. Removing spurious widths makes it more reasonable to define line-based quantification measures, such as divergence and determinism [3]. Another solution is to estimate the entropy by looking at the distribution of the diagonal lines [23]. The entropy is based on the probability  $p(\epsilon, l)$  that diagonal lines structures with certain length  $l$  and similarity  $\epsilon$  occur in the recurrence matrix [19, 21], and can be computed as follows:

$$\text{ENTR}(\epsilon, l_{\min}) = - \sum_{l=l_{\min}}^N p(\epsilon, l) \ln p(\epsilon, l) \quad (12.7)$$

Recurrence plots (RPs) and corresponding recurrence quantification analysis (RQA) measures have been used to detect transitions and temporal deviations in the dynamics of time series. Since detected variations in RQA measures can easily be misinterpreted, Marwan et al. [20] have proposed to calculate a confidence level to study significant changes. They formulated the hypothesis that the dynamics of a system do not change over time, and therefore the RQA measures obtained by the sliding window technique will be normally distributed. Consequently, if the RQA measures are out of a predefined interquantile range, an observation can be considered significantly. Detecting changes in dynamics by means of RQA measures obtained from a sliding window have been proven to be useful in real-life applications such as comparing traffic flow time series under fine and adverse weather conditions [37].

Since recurrence plot-based techniques are still a rather young field in nonlinear time series analysis, systematic research is necessary to define reliable criteria for the selection of parameters, and the estimation of RQA measures [23].

## 12.6 Recurrence Plot-Based Distance

According to our formalization of joint cross recurrence (JCR) in Eq. 12.4 and the denotation of the determinism (DET) in Eq. 12.5, we can define our RecuRRence Plot-based (RRR) distance measure as follows:

$$\text{RRR}(\epsilon, l_{\min}) = 1 - \text{DET}(\epsilon, l_{\min}) \quad (12.8)$$

Since the DET value ranges from 0 to 1, depending on the proportion of diagonal line structures found in a JCR plot, the RRR distance is 0 if the trajectory of both dynamical systems is identical and 1 if there are **no** similar patterns at any position in time.

Although our proposed RRR distance measure can be used as a subroutine for various time series mining tasks, this work primarily focuses on clustering. Our aim is to group a set of  $t$  unlabeled time series  $T$  into  $k$  clusters  $C$  with centroids  $Z$ . In order to evaluate the performance of the time series clustering with respect to our RRR distance, we suggest to quantify the number of similar patterns that recur

within the established clusters. Therefore, we define the following cluster validation index:

$$E(k) = \frac{1}{t-k} \sum_{z \in \{Z\}} \sum_{c \in \{C_z \setminus z\}} \text{RRR}(z, c) \quad (12.9)$$

According to our problem setting, the more patterns occur jointly when comparing each centroid  $z \in \{Z\}$  with all objects  $c \in \{C_z \setminus z\}$  of the corresponding cluster, the lower  $E$ , the better our clustering, and the more characteristic are the corresponding prototypes.

Furthermore, we are going to evaluate the clustering of time series according to the index  $I$  [24], whose value is maximized for the optimal number of clusters:

$$I(k) = \left( \frac{1}{k} \cdot \frac{E(1)}{E(k)} \cdot D_k \right)^p \quad (12.10)$$

The index  $I$  is a composition of three factors [24], namely  $1/k$ ,  $E(1)/E(k)$ , and  $D_k$ . The first factor will try to reduce index  $I$  as the number of clusters  $k$  increases. The second factor consists of the ratio of  $E(1)$ , which is constant for a given dataset, and  $E(k)$ , which decreases with increase in  $k$ . Consequently, index  $I$  increases as  $E(k)$  decreases, encouraging more clusters that are compact in nature. Finally, the third factor,  $D_k$  (which measures the maximum separation between two clusters over all possible pairs of clusters), will increase with the value of  $k$ , but is bounded by the maximum separation between two points in the dataset.

$$D_k = \max_{i,j=1}^k \|z_i - z_j\| \quad (12.11)$$

Thus, the three factors are found to compete with and balance each other critically. The power  $p$  is used to control the contrast between the different cluster configurations. Previous work [24] suggests to choose  $p = 2$ .

The index  $I$  has been found to be consistent and reliable, irrespective of the underlying clustering technique and data dimensionality, and furthermore has been shown to outperform the Dunn and David-Bouldin index [24].

## 12.7 Dimensionality Reduction

As with most problems in computer science, the suitable choice of representation greatly affects the ease and efficiency of time series data mining [15]. Piecewise Aggregate Approximation (PAA), a popular windowed averaging technique, reduces a time series  $x$  of length  $n$  to length  $n/r$  by dividing the data into  $r$  equal sized frames. The mean value of the data falling within a frame is calculated and a vector of these values becomes the data-reduced representation.

$$x_i = \frac{r}{n} \sum_{j=\frac{n}{r}(i-1)+1}^{\frac{n}{r}i} x_j \quad (12.12)$$

$$i = 1 \dots r, \quad j = 1 \dots n$$

The PAA dimensionality reduction is intuitive and simple, yet has been shown to rival more sophisticated dimensionality reduction techniques like Fourier transforms and wavelets [15]. Having transformed a time series database into PAA, we can apply our proposed recurrence plot-based time series distance measure on the reduced representation. Since the computational complexity of our RRR distance measure is quadratic in the length  $n$  of the time series, reducing the original time series to  $r$  dimensions leads to a performance improvement of factor  $(n/r)^2$ . In our experiments on the real-life vehicular data we use a compression rate of  $n/r = 10$ , which correspond to a speedup of two orders of magnitude or rather 100 times less matrix entries to compute. However, this approach comes with the cost of missing recurrences [23].

## 12.8 Adjustment Window Condition

Another approach to reduce the computational complexity of our proposed recurrence plot-based (RRR) time series distance measure is to constrain the number of cells that are evaluated in the distance matrix [30]. Constraints have been successfully applied to the Dynamic Time Warping (DTW) distance to create tight lower bounds which allow to prune similarity calculations [8, 11]. The two most commonly used constraints are the Itakura Parallelogram [6] and the Sakoe-Chiba Band [29], which both speed up calculations by a constant factor, but still lead to quadratic complexity if the window size  $w$  is a linear function of the time series.

Given the formal definition of (joint) cross recurrence (see Eqs. 12.2 and 12.4), the Sakoe-Chiba Band is an adjustment window condition which corresponds to the fact that time-axis fluctuations in usual cases never causes a too excessive timing difference [29]:

$$|i - j| \leq w \quad (12.13)$$

$$\forall x_i, y_j \in \mathbb{R}^d, \quad i = 1 \dots N, \quad j = 1 \dots M$$

In general, constraints work well in domains where time series have only a small variance, but perform poorly if time series are of events that start and stop at radically different times [30]. Since this study considers time series that exhibit recurring patterns at arbitrary positions, we refrain from applying constraints for the data under study.

## 12.9 Evaluation

The goal of our evaluation is to assess how well the RRR distance is suited for: (i) calculating the similarity between time series with order-invariance (in Sect. 12.9.1), (ii) clustering time series that contain similar trajectories at arbitrary positions (in Sect. 12.9.2), and (iii) identifying prototypical time series that cover as much as possible patterns which co-occur in other sequences of the dataset (in Sect. 12.9.3).

### 12.9.1 Order-Invariance

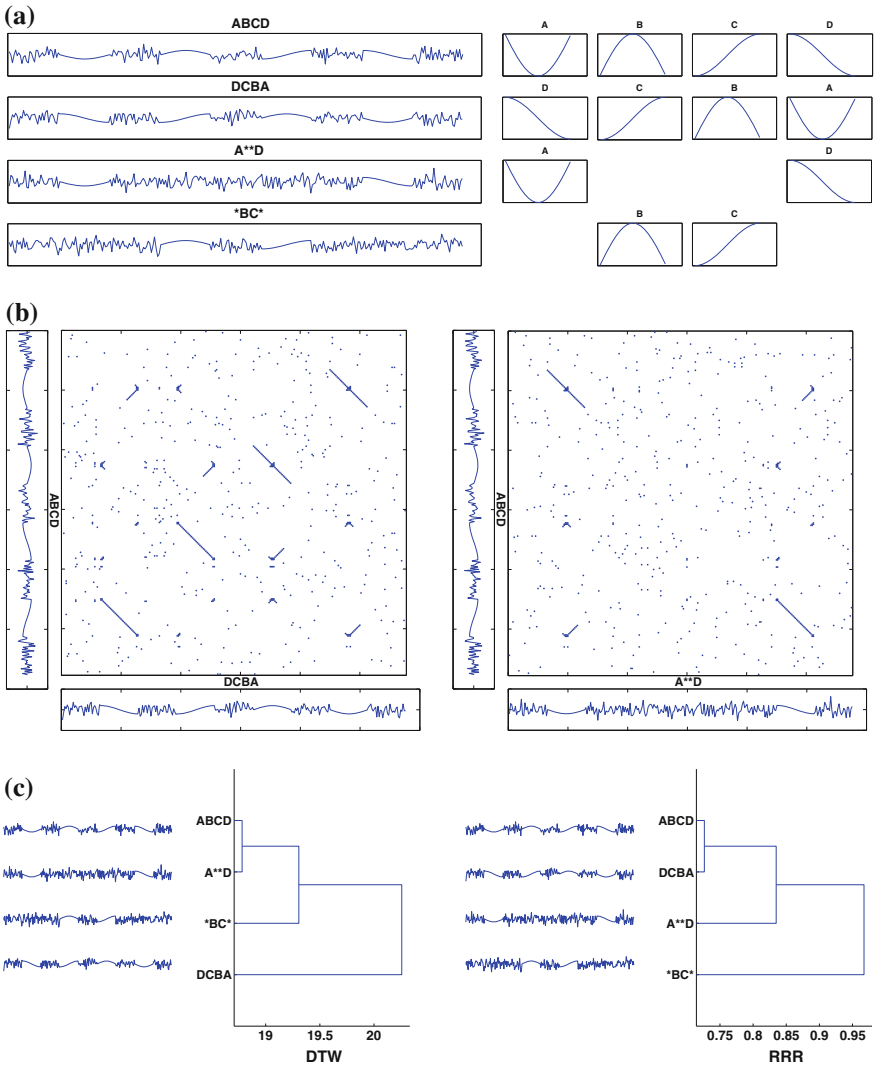
In this section, we demonstrate the practicality of our proposed RRR distance on a sample dataset of synthetic time series. As illustrated in Fig. 12.2a, we consider four different normally distributed pseudorandom time series with artificially implanted sinus patterns. The first two time series comprise the same subsequences in reverse order, whereas the last two time series contain a subset of the artificially implanted signals.

Figure 12.2b illustrates the cross recurrence plot (CRP) of time series ABCD and DCBA as well as ABCD and A\*\*D introduced in Fig. 12.2a. Lines parallel to the main diagonal (from upper left to bottom right corner) indicate similar subsequences in both time series. The percentage of recurrence points that form diagonal lines is much higher in the CRP of the time series ABCD and DCBA than in the CRP of the pair ABCD and A\*\*D. As discussed in Sect. 12.6, we quantify the local small-scale structures in the recurrence plots by means of the determinism DET (refer to Eq. 12.5).

Figure 12.2c shows a direct comparison of *Dynamic Time Warping* and our introduced RRR distance measure. As expected, the hierarchical cluster tree generated by means of DTW indicates a relatively small distance between the time series ABCD, A\*\*D and \*BC\*, because they exhibit similar subsequences at the same positions. However, DTW treats the time series DCBA as an outlier, because the artificially implanted patterns occur in reverse order and cross-alignment is prevented. In contrast, the RRR measure considers the time series ABCD and DCBA as most similar, as the order of the matched patterns is disregarded. Furthermore, the dendrogram generated by means of RRR reveals that the time series A\*\*D and \*BC\* are dissimilar to ABCD and DCBA, which is due to the fact that the overlap of same or similar subsequences is relatively small ( $\leq 50\%$ ).

The results presented in Fig. 12.2 serve to demonstrate that the proposed RRR distance measure is able to handle time series with order-invariance. In the following, we investigate the capability of our RRR measure to cluster time series which exhibit same or similar subsequences at arbitrary positions in time.



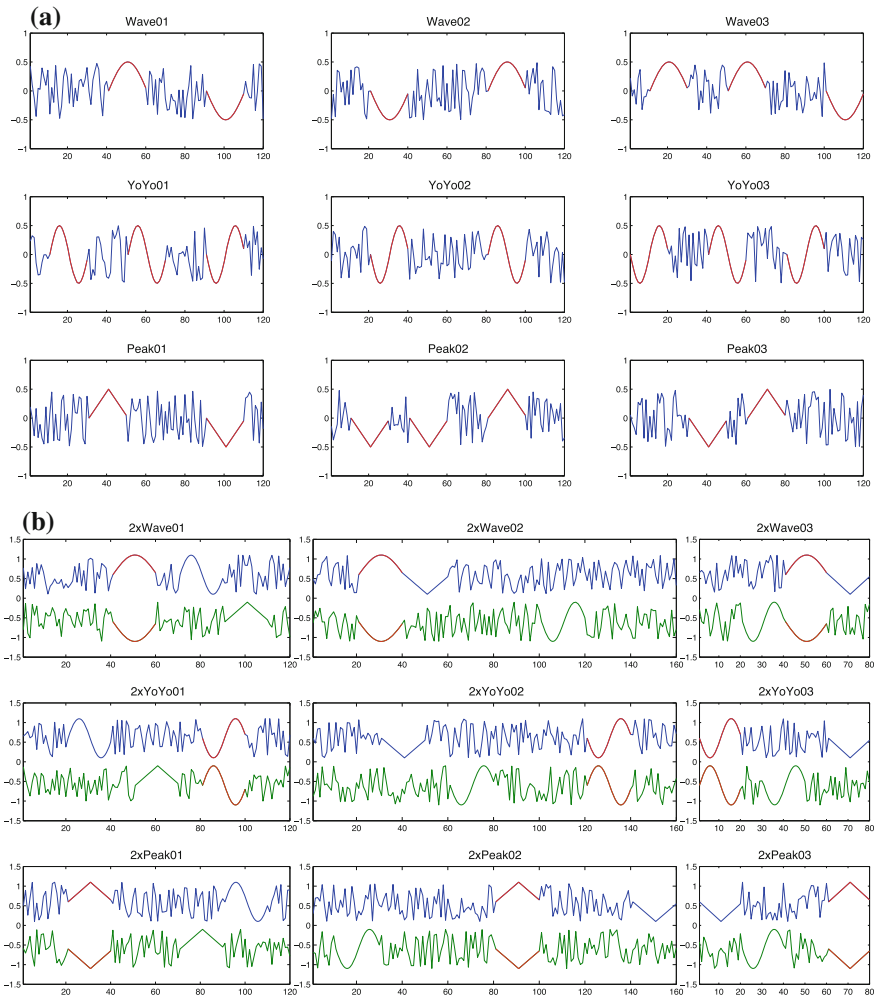


**Fig. 12.2** **a** Sample dataset of normally distributed pseudorandom time series (named as ABCD, DCBA, A\*\*D and \*BC\*, illustrated *left*) with artificially implanted sinus patterns (labeled as A–D, presented in their occurring order on the *right*). **b** Cross Recurrence Plot (CRP) of synthetic time series ABCD and DCBA (*left*) as well as ABCD and A\*\*D (*right*) introduced in **a**. Note that the main diagonal runs from *upper left* to *bottom right*. **c** Agglomerative hierarchical cluster tree (dendrogram) of synthetic time series data (introduced in **a**) according to the DTW distance (*left*) and our proposed RRR distance (*right*), where the x-axis reveals the distance between the time series being merged and the y-axis illustrates the corresponding name and shape of the signal

### 12.9.2 Synthetic Data

This controlled experiment aims at visualizing the clustering results of the proposed RRR distance measure compared to the DTW distance.

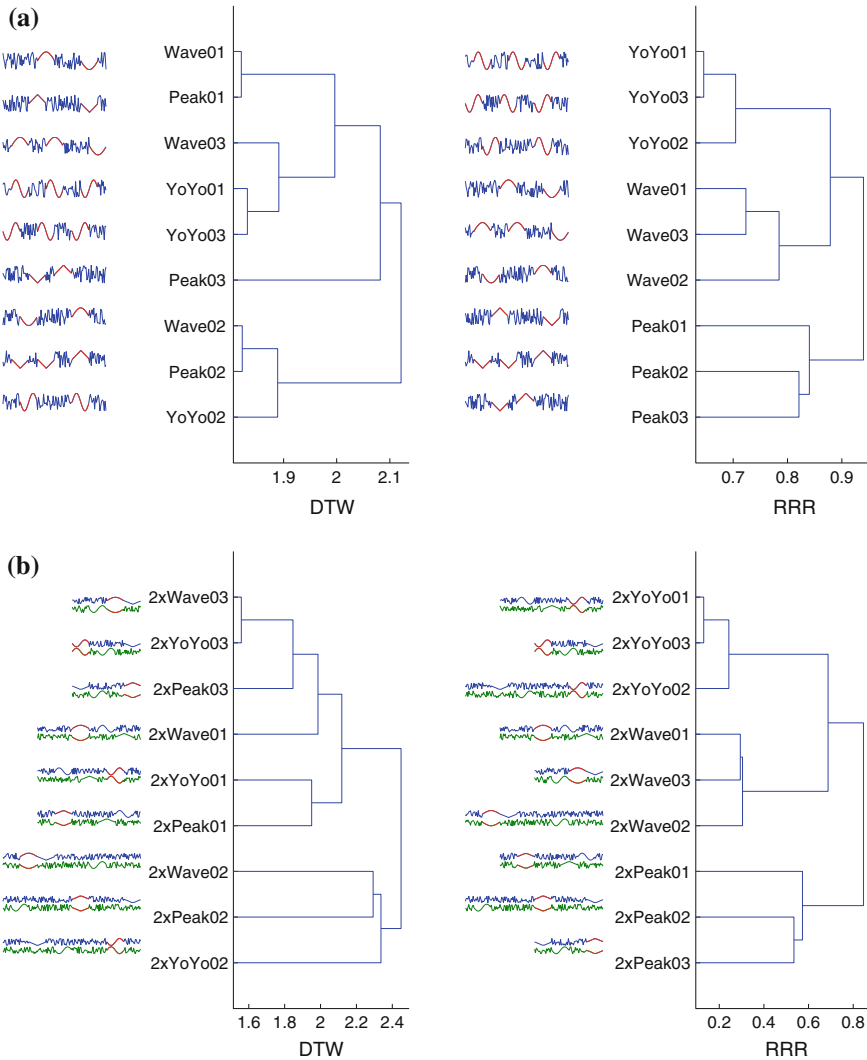
We generated a labeled dataset, which consists of nine time series from three different categories, called Wave, YoYo, and Peak. Each category comprises three time series characterized by multiple occurrence of the same artificial patterns at arbitrary positions. The dataset consists of univariate time series of equal length, as shown in Fig. 12.3. To visualize the clustering results of the RRR and DTW



**Fig. 12.3** Univariate **a** and multivariate **b** synthetic time series with artificially implanted patterns (red color) at arbitrary positions, where each time series belongs to one of three groups (Wave, YoYo, and Peak)

distance, we applied agglomerative hierarchical clustering with complete linkage on the synthetic dataset.

Figure 12.4 illustrates the generated hierarchical cluster trees for both examined distance measures on the synthetic time series. The first observation to be made is that RRR perfectly recovers the cluster structure provided by the ground truth, given



**Fig. 12.4** Cluster tree (dendrogram) of univariate **a** and multivariate **b** synthetic time series (introduced in Fig. 12.3) according to the DTW and RRR distance. The x-axis reveals the distance between the time series being merged and the y-axis illustrates the corresponding name and shape of the time series

our knowledge that there are three categories. In contrast, the DTW distance fails and assigns time series of different categories to the same cluster at an early stage. The second observation to be made is that RRR is able to recover the ground truth even if a large portion of the time series is noisy. The DTW distance, however, groups time series into the same clusters, if they have globally a similar shape. Therefore, the noisy parts of the time series supersede or superimpose the relevant recurring patterns.

### 12.9.3 Real-Life Data

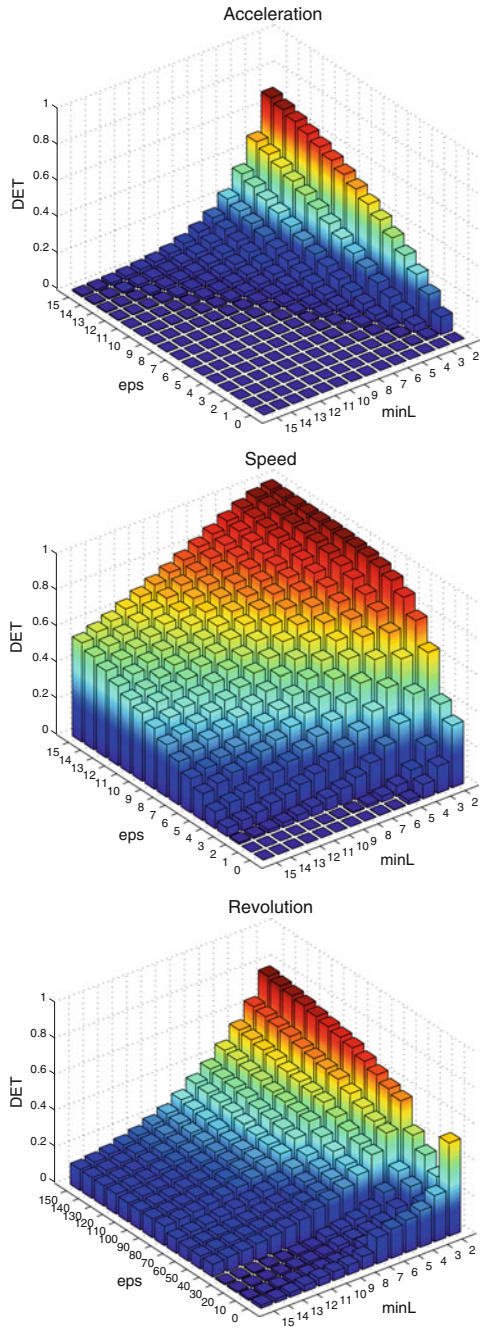
This experiment aims at assessing the time series prototypes identified by the proposed RRR distance measure compared to the DTW distance.

For our evaluation, we consider the VW DRIVE dataset, which consists of 124 real-life test drives recorded by one vehicle operated by seven different individuals. Test drives are represented as multivariate time series of varying length and comprise vehicular sensor data of the same observed measurements. Since we aim to identify operations profiles that characterize recurring driving behavior, we exclusively consider accelerator, speed, and revolution measurements, which are more or less directly influenced by the driver. The complete VW DRIVE dataset contains various other measurements, such as airflow and engine temperature, and can be obtained by mailing the first author of this paper.

To measure the (dis)similarity of the VW DRIVE time series using our proposed RRR distance, we first need to determine the optimal similarity threshold  $\epsilon$  and pattern length  $l_{\min}$  for each of the considered measurements, such that a considerable amount of the recurring patterns is preserved.

Figure 12.5 shows the determinism value for the accelerator, speed, and revolution signal in regard to different parameters settings. We can observe that for all considered signals the DET value decreases with increasing pattern length  $l_{\min}$  and decreasing similarity threshold  $\epsilon$ . Furthermore, Fig. 12.5 reveals that the speed signal is highly deterministic, meaning that the same patterns occur frequently, whereas the acceleration and revolution signal are less predictable and show more chaotic behavior.

Since we aim to analyze all signals jointly by means of the proposed joint cross recurrence plot (JCRP) approach, we have to choose a pattern length or rather minimum diagonal line length  $l_{\min}$  that is suitable for all signals. In general, we are looking for relatively long patterns with high similarity. In other words, we aim to find a parameter setting with preferably large  $l_{\min}$  and small  $\epsilon$  which results in a DET value that is above a certain threshold. To preserve the underlying characteristics or rather recurring patterns contained in examined data, at least 20% of the recurrence points should form diagonal line structures, which corresponds to  $\text{DET} \geq 0.2$ . Based on this criterion, we choose  $l_{\min} = 5$  and  $\epsilon = 14/2/40$  for the accelerator, speed, and revolution signal, respectively. Note that the individual signals were not normalized,



**Fig. 12.5** Determinism (DET) value for changing similarity threshold  $\epsilon$  and minimum diagonal line length  $l_{\min}$  for accelerator, speed, and revolution signal; based on the cross recurrence plots (CRPs) of 10 randomly selected pairs of tours from our DRIVE dataset. Note that the DET was averaged

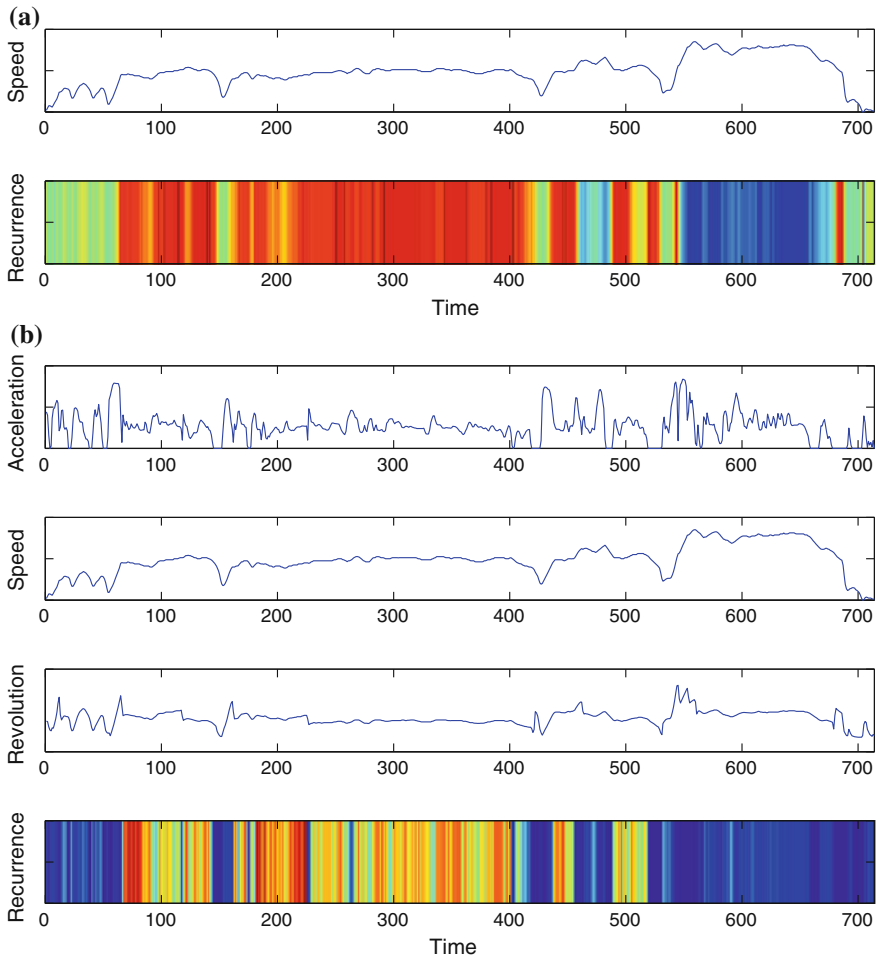
(a) Speed		(a) Speed		(b) Acceleration, Speed, and Revolution		(b) Acceleration, Speed, and Revolution			
k	I_RRR	E_RRR	I_DTW	E_DTW	I_RRR	E_RRR	I_DTW	E_DTW	k
1	-	0.5441	-	0.7041	-	0.7959	-	0.8737	1
2	<b>1.0000</b>	<b>0.5168</b>	0.1162	0.6794	<b>1.0000</b>	<b>0.7393</b>	0.7775	0.8622	2
3	0.8778	0.5034	0.6904	0.6602	0.7820	0.7203	0.9088	0.8405	3
4	0.6431	0.4952	0.7548	0.6474	0.5558	0.7064	0.8585	0.8413	4
5	0.4647	0.4924	0.4438	0.6474	0.3883	0.6992	<b>1.0000</b>	<b>0.8407</b>	5
6	0.3479	0.4909	<b>1.0000</b>	<b>0.6480</b>	0.2821	0.6934	0.9746	0.8420	6
7	0.2687	0.4888	0.2993	0.6479	0.2141	0.6910	0.2529	0.8452	7
8	0.2151	0.4892	0.1894	0.6493	0.1679	0.6897	0.3100	0.8482	8
9	0.1751	0.4866	0.1189	0.6507	0.1362	0.6855	0.3955	0.8478	9
10	0.1469	0.4862	0.1271	0.6524	0.1131	0.6837	0.2119	0.8534	10
11	0.1254	0.4838	0.3730	0.6530	0.0960	0.6818	0.2624	0.8545	11
12	0.1078	0.4823	0.1184	0.6544	0.0825	0.6784	0.4089	0.8528	12
13	0.0947	0.4817	0.1616	0.6518	0.0717	0.6781	0.2517	0.8576	13
14	0.0838	0.4804	0.2449	0.6531	0.0635	0.6755	0.2453	0.8574	14
15	0.0745	0.4805	0.2988	0.6598	0.0565	0.6746	0.2941	0.8603	15
16	0.0672	0.4803	0.2365	0.6570	0.0508	0.6718	0.2753	0.8588	16
17	0.0609	0.4780	0.1862	0.6507	0.0462	0.6674	0.1106	0.8535	17
18	0.0557	0.4774	0.1761	0.6569	0.0422	0.6687	0.2091	0.8622	18
19	0.0514	0.4751	0.3307	0.6603	0.0387	0.6687	0.1336	0.8596	19
20	0.0473	0.4756	0.0899	0.6579	0.0358	0.6667	0.1036	0.8563	20

**Fig. 12.6** Evaluation of RRR and DTW distance for clustering **a** univariate and **b** multivariate time series of our DRIVE dataset. We compare the index  $E$  for the number of clusters  $k$  where the (normalized) index  $I$  reaches its maximum. The results are based on 1,000 runs of  $k$ -medoids clustering with random initialization

wherefore the  $\epsilon$ -threshold represents the accelerator pedal angle, kilometers per hour, and rotations per minute.

To identify prototypical time series using RRR and DTW distance respectively, we applied  $k$ -medoids clustering with random initialization. For evaluation purpose, we computed index  $I$  and  $E$  for a varying number of  $k$  prototypes. The results of index  $I$  were normalized in a way that the highest value, which indicates the optimal number of clusters, equals one. Since index  $E$  is a sum of RRR values (see Eq. 12.9) and  $RRR = 1 - DET$ , the lower  $E$ , the higher the average DET value, and the more recurring (driving behavior) patterns are comprised of the prototypes identified by the respective distance measure.

Figure 12.6 shows the empirical results for clustering univariate and multivariate time series of the VW DRIVE dataset using RRR and DTW distance, respectively. Since the VW DRIVE dataset consists of ‘only’ 124 test drives recorded by one and the same vehicle, the optimal number of clusters for both RRR and DTW distance is rather small. However, the proposed RRR distance is able to find cluster configurations with lower index  $E$  values or rather prototypes with higher amount of recurring patterns than the DTW distance. In case of univariate time series (a), in particular speed measurements, RRR and DTW achieved an index  $E$  value of around 0.52 and 0.65 for the optimal number of clusters, which corresponds to a determinism value of 0.48 and 0.35, respectively. In the multivariate case (b), RRR and DTW reached an index  $E$  value of around 0.74 and 0.84 for the optimal number



**Fig. 12.7** Medoid time series of biggest cluster (with  $k = 2$ ) found by our RRR distance measure for **a** univariate and **b** multivariate case. The intervals highlighted in *red color* indicate patterns that frequently recur in the time series objects of the corresponding cluster, whereas intervals in *blue* indicate low recurrence

of clusters, which corresponds to determinism value of 0.26 and 0.16, respectively. As might be expected, the results for the univariate time series are better than for the multivariate case, because the search space expands and the probability of recurring patterns decreases with an increasing number of dimensions or measurements, respectively. In both cases, however, our RRR distance performs about 10 % better than the compared DTW distance, meaning that the identified prototypes contain 10 % more recurring (driving behavior) patterns.

Figure 12.7 shows the prototype or rather medoid time series of the biggest cluster found by the k-medoids algorithm (for  $k = 2$ ) in combination with our RRR distance

measure. In the univariate case (a) the medoid contains a high amount of patterns that recur in the time series objects of the corresponding cluster, making it an excellent prototype. As expected, in the multivariate case (b) the medoid time series contains less and shorter intervals of recurring patterns.

## 12.10 Application

Having introduced our recurrence plot-based distance measure, we are eventually in the position to present BestTime, a platform-independent Matlab application with graphical user interface, which enables us to find representatives that best comprehend the recurring temporal patterns contained in a certain time series dataset. Although BestTime was originally designed to analyze vehicular sensor data and identify characteristic operational profiles that comprise frequent behavior patterns [32], our extended version [36] can be used to find representatives in arbitrary sets of single- or multi-dimensional time series of variable length.

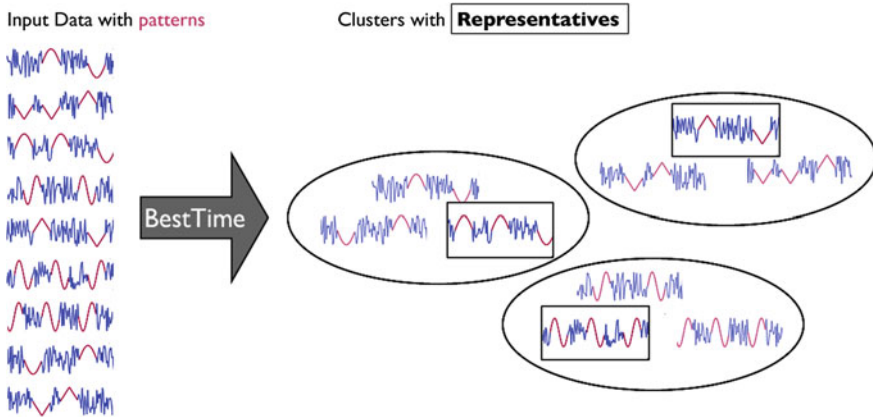
As described above, our approach to find representatives in time series datasets is based on agglomerative hierarchical clustering [14]. We define a representative as the time series that is closest to the corresponding cluster center of gravity [25]. Since we want a representative to comprehend the recurring temporal patterns contained in the time series of the respective cluster, we need a distance measure that accounts for similar subsequences regardless of their position in time [32].

However, as mentioned before, traditional time series distance measures, such as the Euclidean distance (ED) and Dynamic Time Warping (DTW), are not suitable to match similar subsequences that occur in arbitrary order [1, 4]. Hence, we proposed to employ Recurrence Plots (RPs) and corresponding Recurrence Quantification Analysis (RQA) [21, 38] to measure the pairwise (dis)similarity of time series with similar patterns at arbitrary positions [34]. Above, we introduced a novel recurrence plot-based distance measure, which is used by our BestTime tool to cluster time series and find representatives.

In the following, we briefly describe the operation of our BestTime application and illustrate the data processing for a small set of sample time series, see Figs. 12.8 and 12.9. Please feel free to download our BestTime tool [36] to follow the stepwise operating instructions given below.

**Input Data.** BestTime is able to analyze multivariate time series with same dimensionality and of variable length. Each individual time series needs to be stored in an independent csv (comma separated values) file, where rows correspond to observations and columns correspond to variables. Optionally, the first row may specify the names of the variables. The user selects an input folder that should contain all time series in specified csv format. A small set of sample time series that we use as input is illustrated in Fig. 12.8.





**Fig. 12.8** Given a set of time series with previously unknown patterns, we aim to cluster the data and find a representative (highlighted) for each group

**Minimum Number of Observations.** Depending on the application, the user can optionally reduce the size of the dataset by specifying the minimum length of the time series which should be considered for further processing.

**Data Reduction Rate.** Since the computational complexity of our distance calculations is quadratic in the length of the time series, we offer the possibility to reduce the length via piecewise aggregate approximation [4]. Given a time series of length  $n$  and a reduction rate  $r$ , the approximate time series is of length  $n/r$ .

**Minimum Pattern Length.** As described in Sect. 12.9, the predetermined minimum pattern length  $l_{\min}$  directly influences the time series similarity. This parameter strongly depends on the application and needs to be chosen by a domain expert.

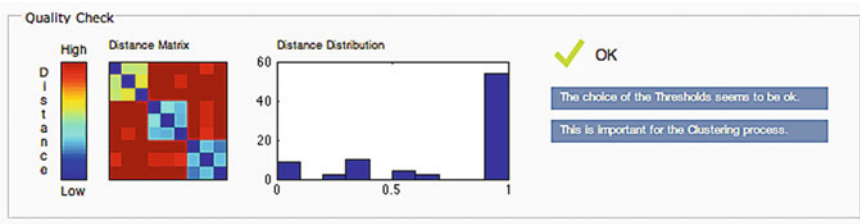
**Variable Selection.** In case of time series datasets with multiple dimensions, the user interface of our tool offers the possibility to select the variables that should be considered for further analysis.

**Similarity Threshold.** This parameter is usually very sensitive and directly influences the clustering result. Since it may be challenging to determine an appropriate similarity threshold  $\epsilon$  for each variable, our tool can alternatively recommend (estimated) thresholds.

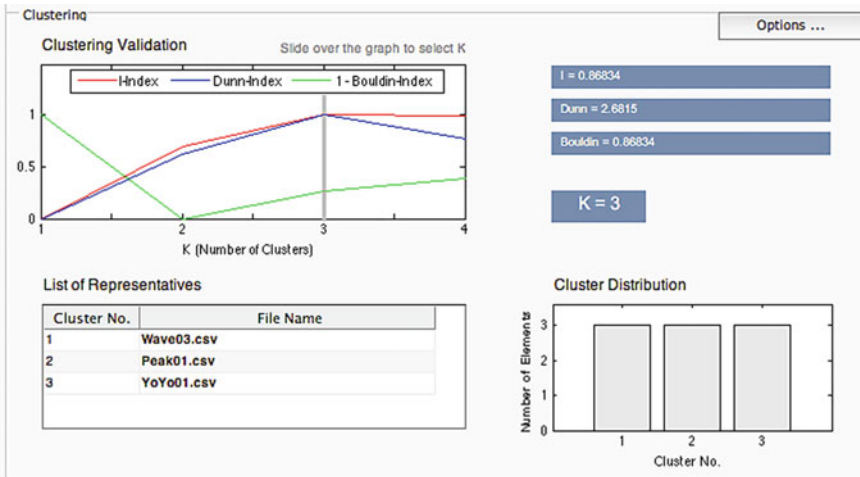
**Parallel Computing.** Calculating the distance matrix is costly for large datasets. However, this step is fully parallelized and runs almost  $n_{\text{CPU}}$ -times faster than serial processing. Up to 12 parallel workers are supported.

**Quality Control.** Our tool presents a colored plot of the computed distance matrix and a histogram of the distance distribution in order to ensure appropriate parameter settings as well as clusters that preserve the time series characteristics. Since both plots are updated iteratively during distance calculations, we can abort computation anytime the preview suggests undesired results. For the distance matrix, a high variance in the distances/colors indicates an appropriate parameter setting, and a low variance in the distances/colors may result in poor clustering. In general,

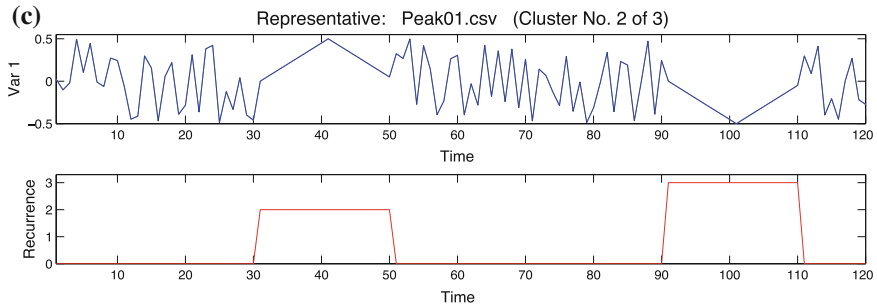
(a)



(b)



(c)



**Fig. 12.9** BestTime operation and data processing for finding representatives in time series datasets, exemplified on sample time series introduced in Fig. 12.8. **a** Visualization of computed distance matrix and distance distribution, which are used to ensure both appropriate parameter settings and clusters that preserve the time series characteristics. **b** Clustering results which show various validation indexes for a changing number of clusters, the list of identified representatives for a selected number of clusters, and the cardinality of the individual clusters. **c** Detailed view of a representative and its corresponding pattern frequency with regard to the selected cluster

good clustering results can be achieved when the distances do not accumulate at either end of the interval (all close to zero or one). Figure 12.9a shows the quality control for our sample dataset.

**Clustering Validation.** To support the user in choosing an optimal number of  $k$  clusters or representatives, our tool validates the cluster goodness for changing  $k$  according to three cluster validation indexes. Figure 12.9b shows the cluster validation for our sample dataset.

**Cluster Distribution.** The clustering may result in groups of different size. Our tool illustrates the cluster distribution to identify outliers and emphasize prominent groups with expressive representatives. For our sample dataset all clusters have the same size, see Fig. 12.9b.

**List of Representatives.** Since we aim at finding representatives, our tool does not only show a list of identified candidates as illustrated in Fig. 12.9b, but also allows to visualize the time intervals or patterns that co-occur in other time series of the same cluster, see Fig. 12.9c.

Please note that we provide supplementary online material [36], which includes our BestTime tool for finding time series representatives, real-life testing data, a video demonstration, and a technical report.

## 12.11 Conclusion and Future Work

This work is a first attempt to solve time series clustering with nonlinear data analysis and modeling techniques commonly used by theoretical physicists. We adopted recurrence plots (RPs) and recurrence quantification analysis (RQA) to measure the (dis)similarity of multivariate time series that contain segments of similar trajectories at arbitrary positions and in different order.

Strictly speaking, we introduced the concept of joint cross recurrence plots (JCRPs), a multivariate extension of traditional RPs, to visualize and investigate recurring patterns in pairwise compared time series. Furthermore, we defined a recurrence plot-based (RRR) distance measure to cluster (multivariate) time series with order invariance.

The proposed RRR distance was evaluated on both synthetic and real-life time series, and compared with the DTW distance. Our evaluation on synthetic data demonstrates that the RRR distance is able to establish cluster centers that preserve the characteristics of the (univariate and multivariate) sample time series. The results on real-life vehicular data show that, in terms of our cost function, RRR performs about 10 % better than DTW, meaning that the determined prototypes contain 10 % more recurring driving behavior patterns.

In addition, we have introduced BestTime, a Matlab tool, which implements our RRR distance to find time series representatives that best comprehend the recurring

temporal patterns in a corresponding dataset. Although BestTime was originally designed to analyze vehicular sensor data [32], our extended version [36] can be used to find representatives in arbitrary sets of single- or multi-dimensional time series of variable length.

Worthwhile future work includes (1) the investigation of RQA measures which quantify recurring patterns with uniform scaling, (2) the application of speed-up techniques for RP computations, and (3) the formalization/analysis of a RP-based distance metric.

**Acknowledgments** The proposed recurrence plot-based distance measure for clustering multivariate time series was developed in cooperation with the Volkswagen AG, Wolfsburg. Thanks to Bernd Werther and Matthias Pries (from the Volkswagen AG) for their contribution of expert knowledge and their help in recording vehicular sensor data. The presented BestTime application was developed in cooperation with David Schultz at DAI-Labor.

## References

1. G.E.A.P.A. Batista, X. Wang, E.J. Keogh, A complexity-invariant distance measure for time series, in *SDM*, pp. 699–710 (2011)
2. B.Y. chi Chiu, E.J. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, in *KDD*, pp. 493–498 (2003)
3. J.M. Choi, B.H. Bae, S.Y. Kim, Divergence in perpendicular recurrence plot; quantification of dynamical divergence from short chaotic time series. *Phys. Lett. A* **263**(4–6), 299–306 (1999)
4. H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E.J. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures. *PVLDB* **1**(2), 1542–1552 (2008)
5. B. Hu, Y. Chen, E.J. Keogh, Time series classification under more realistic assumptions, in *SDM* (2013)
6. F. Itakura, Minimum prediction residual principle applied to speech recognition. *IEEE Trans. Acoust. Speech Signal Process.* **23**(1), 67–72 (1975)
7. E.J. Keogh, J. Lin, A.W.-C. Fu, Hot Sax: efficiently finding the most unusual time series subsequence, in *ICDM*, pp. 226–233 (2005)
8. E.J. Keogh, C.A. Ratanamahatana, Everything you know about dynamic time warping is wrong, in *KDD* (2004)
9. E.J. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, C.A. Ratanamahatana, The UCR time series classification/clustering homepage (2011), [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/)
10. E.J. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Min. Knowl. Discov.* **7**(4), 349–371 (2003)
11. E.J. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **7**(3), 358–386 (2005)
12. E.J. Keogh, J. Lin, Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowl. Inf. Syst.* **8**(2), 154–177 (2005)
13. M. Kumar, N.R. Patel, J. Woo, Clustering seasonality patterns in the presence of errors, in *KDD* (2002)
14. T.W. Liao, Clustering of time series data—a survey. *J. Pattern Recognit.* **38**(11), 1857–1874 (2005)
15. J. Lin, E.J. Keogh, S. Lonardi, B.Y. chi Chiu, A symbolic representation of time series, with implications for streaming algorithms, in *SIGMOD*, pp. 2–11 (2003)
16. J. Lin, E.J. Keogh, S. Lonardi, P. Patel, Finding motifs in time series, in *KDD* (2002)

17. J. Lines, A. Bagnall, P. Caiger-Smith, S. Anderson, Classification of household devices by electricity usage profiles, in *IDEAL*, pp. 403–412 (2011)
18. N. Marwan, *Encounters with Neighbours: Current Developments of Concepts Based on Recurrence Plots and their Applications*. Ph.D. thesis, University of Potsdam (2003)
19. N. Marwan, M. Romano, M. Thiel, Recurrence plots and cross recurrence plots. [www.recurrence-plot.tk](http://www.recurrence-plot.tk)
20. N. Marwan, S. Schinkel, J. Kurths, Recurrence plots 25 years later—gaining confidence in dynamical transitions. *Europhys. Lett.* **101**(2), (2013)
21. N. Marwan, M. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems. *Phys. Rep.* **438**(5–6), 237–329 (2007)
22. N. Marwan, A historical review of recurrence plots. *Eur. Phys. J. Spec. Top.* **164**(1), 3–12 (2008)
23. N. Marwan, How to avoid potential pitfalls in recurrence plot based data analysis. *Int. J. Bifurc. Chaos* **21**(4), 1003–1017 (2011)
24. U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(12), 1650–1654 (2002)
25. W. Meesrikamolkul, V. Niennattrakul, C.A. Ratanamahatana, Shape-based clustering for time series data, in *PAKDD*, pp. 530–541 (2012)
26. C.S. Moeller-Levet, F. Klawonn, K.-H. Cho, O. Wolkenhauer, Fuzzy clustering of short time-series and unevenly distributed sampling points, in *LNCS, Proceedings of the IDA2003*, pp. 28–30 (2003)
27. T. Rakthanmanon, B.J.L. Campana, A. Mueen, G. Batista, M.B. Westover, Q. Zhu, J. Zakaria, E.J. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in *KDD*, pp. 262–270 (2012)
28. T. Rakthanmanon, E.J. Keogh, Fast-shapelets: a scalable algorithm for discovering time series shapelets, in *SDM* (2013)
29. H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition. *Trans. Acoust. Speech Signal Process.* **26**(1) (1978)
30. S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space. *J. Intell. Data Anal.* **11**(5), 561–580 (2007)
31. A.P. Schultz, Y. Zou, N. Marwan, M.T. Turvey, Local minima-based recurrence plots for continuous dynamical systems. *Int. J. Bifurc. Chaos* **21**(4), 1065–1075 (2011)
32. S. Spiegel, S. Albayrak, An order-invariant time series distance measure—position on recent developments in time series analysis, in *Proceedings of 4th International Conference on Knowledge Discovery and Information Retrieval (KDIR)* (SciTePress, 2012), pp. 264–268
33. S. Spiegel, J. Gaebler, A. Lommatzsch, E. De Luca, S. Albayrak, Pattern recognition and classification for multivariate time series, in *Proceedings of the 5th International Workshop on Knowledge Discovery from Sensor Data, SensorKDD'11* (ACM, New York, 2011), pp. 34–42
34. S. Spiegel, B.-J. Jain, S. Albayrak, A recurrence plot-based distance measure, in *Springer Proceedings in Mathematics—Translational Recurrences: From Mathematical Theory to Real-World Applications* (2014). To appear
35. S. Spiegel, B.-J. Jain, E. De Luca, S. Albayrak, Pattern recognition in multivariate time series—dissertation proposal, in *Proceedings of 4th Workshop for Ph.D. Students in Information and Knowledge Management (PIKM)*, CIKM'11 (ACM, 2011)
36. S. Spiegel, D. Schultz, M. Schacht, S. Albayrak, Supplementary onlinematerial—besttime App, test data, video demonstration. Technical report: [www.dai-lab.de/spiegel/besttime.html](http://www.dai-lab.de/spiegel/besttime.html) (2013)
37. E.I. Vlahogianni, M.G. Karlaftis, Comparing traffic flow time-series under fine and adverse weather conditions using recurrence-based complexity measures. *J. Nonlinear Dyn.* **69**(4), 1949–1963 (2012)
38. C.L. Webber, N. Marwan, A. Facchini, A. Giuliani, Simpler methods do it better: success of recurrence quantification analysis as a general purpose data analysis tool. *Phys. Lett. A* **373**(41), 3753–3756 (2009)

39. A. Wismueller, O. Lange, D.R. Dersch, G.L. Leinsinger, K. Hahn, B. Puetz, D. Auer, Cluster analysis of biomedical image time-series. *Int. J. Comput. Vis.* **46**(2), 103–128 (2002)
40. J. Zakaria, A. Mueen, E.J. Keogh, Clustering time series using unsupervised-hapelets, in *ICDM*, pp. 785–794 (2012)

# Chapter 13

## Intermodal Mobility Assistance for Megacities

Esra Acar, Marco Lützenberger and Marius Schulz

**Abstract** In this chapter, we present an approach to utilize means of transportation in a more effective and sustainable fashion in order to increase the quality of life in cities and to contribute to global environmental objectives. We describe a travel assistance system that proposes intermodal traveling options which are tailored to drivers' needs. Different information channels are integrated in the system. One of these channels is information derived from video analysis. This analysis is based on a macroscopic approach using particular features extracted from video snapshots periodically captured from static traffic surveillance cameras. The video analysis approach is evaluated on 254 highway traffic videos of the UCSD dataset and achieves an accuracy of 94.90%. Finally, running at 15 frames per second on average, the approach is also appropriate for real-time video analysis, without requiring a special purpose computer. In addition, a routing system based on a dynamically changing map has been developed in order to provide fast and reliable routing solutions. It integrates the information from all channels into one world view and takes these into account when searching for routes through a city.

### A Drive to Work

“Suzanne! Hurry up, we are already late,” Steven shouted. Every morning the same routine! Every morning after breakfast, Steven would go outside to get the car ready. And every morning, he had to wait for his wife who suddenly remembered something ‘urgent’ that she had to do before coming out as well. “You know I have this meeting at work this morning, so stop stalling and get out here! I also have to drop off Clara at her office, so that will take even longer today!” Steven starred at the front door

---

E. Acar (✉) · M. Lützenberger · M. Schulz  
Technische Universität Berlin, Berlin, Germany  
e-mail: esra.acar@tu-berlin.de

M. Lützenberger  
e-mail: marco.luetzenberger@dai-labor.de

M. Schulz  
e-mail: marius.schulz@dai-labor.de

and angrily pushed the horn to support his demand to hurry up. The little sign next to the doorbell caught his eyes: “Here loves, fights and argues the Marks Family!” He started smiling. Although he was angry, he appreciated these arguments with his wife. “It’s things like this that make family life worth living,” he thought. Being dependent on the car was one thing he and Suzanne expected when they decided to buy this house in the suburbs to raise their family. And since his wife was strictly against buying a second car—“We have to head to the same direction anyway, so let’s share the car and save the planet”—he had no choice but to drive her to work every morning. Over the years, their daily arguments when driving to work together had become an important part of their social interaction that neither of them would like to miss. “Sorry honey, I just had to check if I closed the window in the bathroom.” Suzanne sat down on the front seat and buckled up. “Laura told me that there were several burglaries in the neighborhood, so I don’t want to make it too easy for them...” Steven grumbled and started driving. He remembered that he left the window open in the morning, so there was nothing really he could complain about this time.



“So Clara, do you know already what you have to do today at work?” Suzanne turned around to her daughter who was sitting on the back seat. Steven switched off the radio since he also was curious about what Clara had to do today at the news agency that she joined a short time ago. So far, she seemed quite excited about her internship so both Steven and Suzanne were happy as she finally found something that she seemed to like doing for a living. Clara enjoyed the attention she got from her parents and started talking about her new job.

After listening in for a while, Steven noticed some warning sites along the road. “Oh no, look at that!” he mourned. “Looks like they started a construction site here today. Oh my, do you see that traffic jam over there!?” annoyed, Steven stared at the radio which was still switched off. He probably should not have done that earlier since he probably would have been informed by the traffic announcement service of his favorite radio station that he should avoid taking this route today. Sometimes, these announcements could be really annoying. Especially, when they informed about traffic situations that are not relevant for him at all, such as slow traffic on the opposite direction or 200 km from his location. Still, they are better than not being informed



at all, like this time. What he would really need is a personalized driver assistance system that would tailor the information based on his own needs. A few weeks ago, he viewed this automotive show on TV where they presented the latest technological advances in the automobile industry. They featured premium cars that used computer vision technology to assist their drivers. The car was able to recognize speed limit signs or approaching vehicles, warn and react to obstacles on the street, and had many other features that completely changed the driving experience. In the episode, they also revealed that they are now working on connecting the sensors of the cars with a central traffic coordination system. They argued that such system would be able to notice any type of traffic anomalies by interlinking the data streams from cars, surveillance cameras, and many other sources. Pointing out the benefit of the system, they explained that this information could be fed right back into the navigation system of cars, which can then adjust the current route using real-time information. “How great would it be to have such a system and a luxury car,” Steven sighed. He knew that unless they would win in the lottery, they would never be able to afford such a car. At least not as long as Carl and Clara would be able to financially stand on their own feet.

The thought about Clara woke him up from his daydream. She was still talking about her new job, and he really wanted to hear how she liked her new job. Maybe, soon, he would finally be able to afford his dream car...

## 13.1 Introduction

Due to ever increasing urbanization, the traffic situation in megacities is increasingly getting worse. Consequently, using vehicle has become an everyday challenge due to several factors. One of these factors is sudden traffic disturbances. This issue is one of the problems addressed by the Intermodal Mobility Assistance for Megacities (IMA) system. The aim of the system is to enable its users to get access to traffic-related information which is updated in real time. This information is subsequently used by the route planning component of the IMA system to indicate the best path at any time.

From a scientific point of view, the calculation of routes from a given source to a given target location is referred to as the best-first search problem [22]. There are many algorithms that can be used to solve this problem; two of the most popular ones are the Dijkstra [7] and the A\* [22, pp. 97–101] algorithm.

Yet, our traffic and transport networks have changed, and in parallel, our possibility to gain information about the current traffic situation has evolved proportionally. Currently, there are many different channels that can be used to retrieve information about traffic-related data, such as congestion or accidents, or more general data, such as departure and arrival times of public transports. Traffic information is provided by many different sources. These sources include traffic monitoring cameras, user data (e.g., accident or traffic jam reports), third party data (e.g., obtained from public transportation companies), but also data from a multitude of mobile devices and

sensors (e.g., mobile phones, *GPS*<sup>1</sup> units). This data can be used to provide more efficient routing that also accounts for real-time aspects. However, due to the presence of multiple sources, the data to be processed is distributed and highly heterogeneous. Thus, in order to make use of the data (e.g., for routing purposes), it is necessary to preprocess it and to bring different information channels together in a loosely coupled system architecture.

One of the information channels for the IMA system is the traffic congestion level analysis results based on surveillance video analysis. The principal objective of surveillance or monitoring is to identify “suspicious,” “intriguing,” “abnormal,” or “unusual” actions or events which are susceptible to represent significant threat to the public. In a broader sense, these actions or events constitute obstacles to the smooth progress of daily life. Thanks to the rapid reduction of hardware cost over the past years, the deployment of closed-circuit television (CCTV) systems in surveillance scenarios is nowadays a commonplace measure. In addition, rapid developments in computer vision and machine learning, and the consideration that automated video surveillance systems have positively impacted the society has incited members of the scientific community to work on the realization of such surveillance and monitoring systems.

Monitoring of urban roads and highways for traffic safety purposes is one typical use case illustrating the applications of automated surveillance systems. Information is collected on anomalies such as traffic congestion in order to inform drivers about unexpected traffic disturbances. Therefore, automated surveillance video analysis is of increasing relevance for intelligent transport systems (ITSs) [4].

Challenges arise in traffic surveillance video analysis. First of all, the data being analyzed is generally of poor quality (i.e., low-resolution image data). Second, it is also expected that analysis performs well in different operating conditions such as evolving weather conditions. Therefore, video analysis techniques are required to be robust. Third, traffic monitoring objectives generally require real-time processing, which further constrains the complexity of the proposed approaches [4]. In addition to surveillance video data, ITSs usually combine a variety of other sensor data gathered from different sources in order to provide more robust traffic flow analysis methods [4].

Another important information source of the IMA system is live traffic data. We refer to live data as data that describes the status of the entire traffic system. This channel includes data on congestion, accidents, construction sites, or general warnings. There are many different ways to retrieve live traffic data. In this work (see Sect. 13.3.1.3), we focus on a connection to the *Microsoft* traffic and map service *Bing Maps*.<sup>2</sup>

Merging the stream of information coming from different sources and using pieces of information in a routing application is another important functionality of the IMA system. The collected traffic data needs to be interpreted and integrated into a model of the area in which the system is operating. This model is subsequently used to

---

<sup>1</sup> *Global Positioning System*.

<sup>2</sup> Bing Maps Website: <http://www.bing.com/maps/>.

find appropriate routes through the city, with a focus on avoiding congestion and traffic jams.

The aim of this chapter is twofold. First, we present a mechanism that allows combining different and heterogeneous information channels. For this purpose, we make use of the agent paradigm [26] as well as of multiagent system architectures [25].

Second, we describe selected information channels in more detail. Our intention is to outline a system architecture that can be used to integrate, preprocess, and utilize heterogeneous traffic information from different channels and to show how these channels have to be implemented in order to make the approach work.

## 13.2 The IMA System—An Overview

The IMA system provides solutions to its users for effective transportation. The aim of the system is to increase the quality of life in megacities. While IMA is a complex software application, which comprises dozens of services, components, apps, front-ends, etc., the focus of this chapter is on IMA's routing capability (*IMA routing*). IMA routing is principally implemented through two fundamental components, namely: (1) the *Route Planning* component and (2) the *Video Analysis* component. In this section, we elaborate on the architecture of our system. In order to do so, it is necessary to provide details about the framework that was used for its implementation, which can be found in Sect. 13.2.1. Subsequently, in Sect. 13.2.2, we describe the architecture of our system.

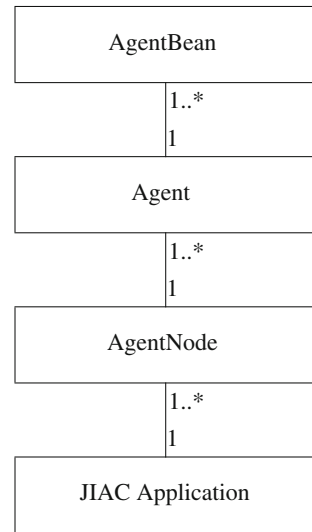
### 13.2.1 IMA—A Distributed System

The IMA system aims to integrate different services, which are not necessarily running on the same hardware. Thus, from an engineering perspective, the IMA application is a distributed system, which can be considered as multiagent system [25]. Agent-oriented software engineering [12] is a common approach to develop such distributed systems. The advantage of applying an agent-oriented view for the development is the presence of a comprehensive set of methodologies, mechanisms, and tools, which significantly eases the engineering process.

One particular tool, which facilitates the development of agent-based systems is the *Java-Based Intelligent Agent Componentware*, or *JIAC* [16]. The development of JIAC was geared toward reliability and robustness [17]. As both characteristics are vital for our system (which is meant to be deployed in a real-life scenario), JIAC appeared to be a judicious choice as the basis of our implementation.

The JIAC architecture is a concept that builds on multiple agents, with each agent serving as an intelligent service execution container. Thus, a multiagent system can work as a flexible and dynamic service execution platform that corresponds to the

**Fig. 13.1** JIAC basic concepts and their structural relationships



principles of service oriented architectures. In order to achieve this, JIAC includes dedicated concepts that cover the whole range, from single service components to a whole application. These concepts, depicted in Fig. 13.1, are as follows:

- The *AgentBean* is the central building block of JIAC agents and applications. All other concepts and elements are built from this. An *AgentBean* is basically a Java class that can be plugged into an agent.
- An *Agent* in JIAC is an autonomous entity dedicated to a particular role. The capabilities of the agent are implemented with *AgentBeans* and can be made accessible to other agents as services.
- An *AgentNode* is a runtime container that holds and manages all agents that are hosted on a single computer. The *AgentNode* mainly provides management and infrastructure functions and is responsible for managing access to physical resources.
- A *JIAC Application* is the sum of all *AgentNodes* that can communicate with each other. Thus, this constitutes a physically distributed environment that encompasses all agents that are able to interact with each other.

### 13.2.2 The IMA System and Its Architecture

In a nutshell, the functionality of the IMA system is to propose tailored routes between one or more locations in a major city.

In doing so, IMA accesses a large pool of information channels and services in order to outperform, in terms of quality, available solutions. As an example, consider

the sudden and unpredictable occurrences of high congestion. Contemporary solutions are not able to account for such incidents. On the contrary, the IMA system integrates live traffic data from three categories of sources in order to account for this particular problem. These sources are as follows: First, data from on-board units, deployed on selected vehicles, is considered. These on-board units provide real-time data about the current traffic flow at selected locations. Second, video channels are analyzed in real-time. The gathered information is directly forwarded to the IMA system and included in the route computation. Finally, traffic data coming from an online platform, namely *Microsoft Bing*, is being used.

IMA seamlessly integrates these three channels and uses their information for the route computation process. Given a certain quality of input data, the resulting routes are highly precise and reflect the current traffic situation much better than available solutions.

For monitoring purposes, the IMA system provides a graphical user front-end, which interactively shows the current traffic situation on an animated map.

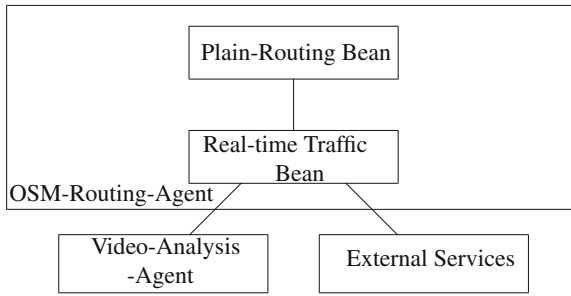
IMA routing was implemented within two distinct JIAC agents with access to external services, in compliance with the agent paradigm. In addition, in conformity with the philosophy of JIAC, fundamental agent-functionality was implemented within encapsulated and reusable AgentBeans. The entire system comprises the following components:

- *The OSM Routing Agent*
  - *The Plain Routing Bean*
  - *The Real-time Traffic Bean*
- *The Video-Analysis-Agent*
- *External Traffic Services*

The *Plain Routing Bean* represents the core of the IMA routing system and provides several capabilities to find appropriate routes through the city, utilizing the underlying topology. In order to compute routes that reflect the current traffic situation, the Plain Routing Bean uses information which is provided by the Real-time Traffic Bean.

The *Real-time Traffic Bean* receives and processes traffic data from connected input channels. Information related to traffic incidents is stored in a volatile fashion in order to reflect the dynamic nature of traffic environments. Available information, however, is directly used for the route computation, such that the resulting routes are tailored to the current traffic situation. Traffic-related data is also forwarded to the graphical user interface of the IMA application, which visually displays the current state of the managed traffic system. Combined together, the Plain Routing Bean and the Real-time Traffic Bean form the *OSM Routing Agent*.

The second agent in our system is the *Video-Analysis-Agent*. This agent comprises only one AgentBean. The Video Analysis Agent analyzes video streams from traffic surveillance cameras located at busy main roads in the city. The data gathered from this analysis gives some indication of the traffic situation at this specific road. This



**Fig. 13.2** The system architecture of the IMA Routing Service application

information is also sent to the Real-time Traffic Bean and used by the Plain Routing Bean for the computation.

Sources classified under “External Traffic Services” can include any kind of service that provides traffic-related information. The only requirement for these services is that they implement the interface which standardizes the access done by the Real-time Traffic Bean. According to MacKenzie et al. [18], JIAC seamlessly integrates the service paradigm. As a consequence, the presented application is able to retrieve data from any (web-)service which complies with this common standard. In order to substantiate the functionality of our approach, we developed a set of exemplary External Traffic Services and integrated those into the system. Each service was implemented as a separate JIAC software agent, in order to make these sources independent from each other. The separation of the functionalities not only increases the maintainability, but also the scalability of the IMA system. In the remainder of this chapter, one of these exemplary services is presented in more detail. This service provides information about traffic incidents, such as construction sites and traffic congestion.

Finally, for monitoring and debugging purposes, we developed a graphical user interface which displays the traffic information received by the *Real-time Traffic Bean* on a map of the city and provides an easy way to get routes from the system. The assembly of the IMA routing system as well as the connections between all relevant components is illustrated in Fig. 13.2.

In the following, we present the fundamental components of the IMA routing system in more detail.

### 13.3 The IMA Routing System

As mentioned above, the IMA routing system comprises two essential agents, namely the OSM Routing Agent and the Video-Analysis Agent, as well as support for external services. In this section, we present related work and elaborate on the implementation details of both agents.

### 13.3.1 The OSM Routing Agent

The OSM Routing Agent implements the fundamental routing functionality of the IMA Routing System. The agent aims at providing a fast and reliable path finding solution, which is able to integrate the gathered traffic data into its worldview in real-time and to find efficient routes, which are tailored to the current traffic situation.

Due to the volatile nature of traffic environments, it was necessary to store available traffic information in a manner which accounts for dynamic changes. We designed this data structure in a graph-like fashion, including subgraphs in order to store changing states of the system. Another issue was to account for the connection to external services (and to respectively store their information). Such mechanism was included as well. In order to explain implementation details, we continue by presenting related work. Subsequently, we present the OSM Routing Agent in detail.

#### 13.3.1.1 Related Work

For the implementation we mainly applied two mechanisms, a specialization of a *best-first search* algorithm, namely *A\* search* [22, pp. 97–101], and the *Open-StreetMap*, or *OSM* framework [21], which provides free map material for traffic systems all over the world.

*A\* search* [22, pp. 97–101] is an algorithm which can be used to find paths through graph structures. *A\* search* evaluates nodes of the graph by combining the costs to reach a given node and the cost to get from this node to the target location. To this end, *A\** “expands,” i.e., starts at a source location and explores the graph, by respectively moving to the neighbor node with the lowest sum of costs that are necessary to reach the node and expected costs to reach the target location from this node. While the former costs have been calculated during the expansion, the latter costs are estimated. This estimation significantly determines the quality of *A\** search. In the case of route finding, one particular heuristic is commonly applied, namely the beeline distance. We applied this heuristic, such that the assessment of the quality of a node  $n$  complies with:

$$\text{estimated\_costs}(n) = \text{cost}(s, n) + \text{beeline}(n, t),$$

where  $s$  is the source location,  $\text{beeline: node} \times \text{node} \rightarrow \mathbb{R}$  is the function which returns the geographical distance between two nodes, and  $t$  is the source location. We implemented this algorithm to be compatible with OSM data.

Map material, which is provided by OSM, is available in several formats. The most common format being *XML*.<sup>3</sup> The syntax of XML-based OSM maps is rather simple. Basically, these files contain only two categories of entry types, namely *node* and *way*. While nodes can be considered as elementary building blocks, ways are basically collections of nodes. Based on this mechanism, entire traffic systems can

---

<sup>3</sup> XML is the abbreviation for *Extensible Markup Language*.

be described. Both, nodes and ways can be extended by custom tags. This mechanism allows to enrich map data with semantic information. By means of tagging, arbitrary information can be stored in OSM maps, e.g., different way-types (roads, sidewalks, bicycle ways, seaways, metro lines, bus lines, etc.) or different points of interest, such as shops, car parks, bus stops, telephone booths, restaurants, among others.

Both, OSM and A\* search were used to implement the core functionality of our OSM Routing Agent. As illustrated in Fig. 13.2, the OSM Routing Agent comprises two JIAC AgentBeans. We describe these agent components below in more detail.

### 13.3.1.2 The Plain Routing Bean

The *Plain Routing Bean* is central to the IMA routing system. It is the only component which has direct access to the OSM-based map data. Therefore, routing was implemented directly within this AgentBean. The map itself is parsed from an OpenStreetMap input-file (based on XML) and translated into a more efficient internal representation.

The parsing process was implemented with the help of the *Apache Xerces2 Java Parser*<sup>4</sup> as follows: Firstly, every node of the future search graph is extracted from the XML-file. In doing so, only nodes that are not required for the routing process are neglected (e.g., traffic lights or shop locations). Secondly, ways with the additional description “highway” (identified by tag with the same name) are loaded from the file. These ways and all remaining nodes are used to instantiate the digraph, which is finally used for the routing. What happens next is that nodes which do not belong to a highway are removed from the model (these nodes are not reachable). The resulting data structure is tailored to the use in IMA. In order to avoid future import processes the optimized data structure is persistently stored within a separate file.

Based on this mechanism, we were able to decrease the file size for an OSM-based representation of Berlin (Germany) by more than 50 %, from roughly 130 MB to about 61 MB. This optimization was necessary, especially in the light of the complexity of OSM data. As an example, the original OSM map for Berlin currently contains 573.397 nodes, while only 165.254 of these are located in a highway structure. Based on our more efficient representation it was possible to decrease the time which is required for the parsing by roughly 50 %.

Every time the system is started, available maps (in the optimized format) are parsed anew. Information in these files are used to instantiate the internal routing graph. This graph is designed as a digraph, where the weighted edges represent the streets with their respective length. In order to access each node and edge quickly, both are stored in a hash map where they are indexed by their respective OSM identifiers, such that nodes only contain references to their neighbor nodes and edges that connect these neighbors. Routing is done by means of A\* search. In fact, the search algorithm is included in a modular fashion, such that data which is required by the algorithm is provided in a standardized way, which is described by an interface. This mechanism

---

<sup>4</sup> The Apache Xerces Project website: <http://xerces.apache.org/>.



allows to easily swap the applied search algorithm and to compare the efficiency of different routing solutions. Based on our experiences, we integrated A\* search as it perfectly complies with our requirements.

Another benefit of using the OSM data is the semantic enrichment of the provided data. The parsing process is not limited to retrieving information on length and geographical positions of the streets and junctions, but also stores information about speed limits, street names and street types. This enables the routing algorithm to distinguish between different means of transportation, e.g., vehicles, bicycles, or regular walking, to name a few.

As mentioned above, routing is done by means of A\* search. We implemented the algorithm to calculate path costs based on the required time to reach a given target location (fastest way). Estimates about the cruising speed of vehicles are done based on the speed limits which are provided by the OSM framework. In the case that routes are done on foot, we use fixed speed values.

As a utility, all street names are also parsed and stored in memory. This makes it possible to search for street names and to retrieve associated nodes on the graph, rather than specifying an exact geographical coordinate. To find street names quickly, available data is stored in a dedicated data structure, namely a “ternary search tree.” This structure allows us to search for strings in a space and time efficient manner, as retrieving a specific street can be managed with almost constant complexity.<sup>5</sup> Both the search tree and the search graph which is used for the routing are instantiated simultaneously.

A major challenge was the integration of the traffic data into the map. The main requirement was the possibility to dynamically include information about traffic incidents (e.g., congestion or accidents) and to account for the volatile nature of these incidents. The premise was to include this information without instantiating the graph structures again. We approached this problem by parsing the traffic data into a separate graph which contains only those parts of the map which are directly affected. Whenever the search algorithm arrives at positions in the graph where additional information is available, the newly created graph fragment is favored over the static structure. This mechanism also accounts for situations in which additional information has to be stored while the *Plain Routing Bean* is operating.

The traffic situation itself is represented by a number of incidents. We distinguish between *Point-Incidents* and *Edge-Incidents*. The Point-Incidents are traffic disturbances which are limited to particular locations or small area on the map, e.g., smaller construction sites and erroneous traffic lights. These incidents are integrated into the map by searching for all nodes within a certain range of this point. All edges reaching to one of these nodes are then assigned the *Incident-Coefficient* which determines the extent of the delay that occurs if one travels on affected edges. These are stored in the above-mentioned sub-graph. Contrary, Edge-Incidents are these incidents which affect a segment of a road and are processed as follows: an Edge-Incident is given by a start and an end location—provided as nodes. The application uses these start

---

<sup>5</sup> More precisely, retrieving a street name has a time complexity in  $O(l)$ , where  $l$  is the number of characters in the longest street name.

and end nodes in order to determine edges which connect these nodes. These are the edges which are affected by the incident. The problem to determine the edges between two nodes in a graph is a specialization of best-first search problem, thus, our A\* search algorithm fits its purpose, again. When the edges were determined, the process is the same as in the case of Point-Incidents.

Following the JIAC philosophy, the above-presented features are provided to other system components in compliance with the service metaphor. Each functionality is offered as JIAC *actions*. Actions which are specific for the IMA routing can be grouped into the following two categories:

- searching for the fastest path, and
- searching for the shortest path.

Both categories require information about the selected mean of transport. Possible options are: car, bicycle, and foot (in the system, the latter option is referred to as “Pedestrian”). Searching for the fastest path is available in two different modes, namely:

- searching without traffic information, and
- searching including traffic information.

Information about traffic incidents can be added to the system by using another JIAC action.

### 13.3.1.3 The Real-Time Traffic Bean

The Real-time Traffic Bean provides functionality to merge traffic data from different sources and to make them available for the Plain Routing Bean. The Plain Routing Bean and the Real-time Traffic Bean work hand in hand, thus, we plugged both beans into the same agent, the OSM Routing Agent.

While fundamental routing functionality was implemented within the Plain Routing Bean, the Real-time Traffic Bean accounts for the latest information about the managed traffic system. The bean communicates with all services which provide traffic related data and accepts routing requests. It also manages the connection to the graphical user interface.

The procedure of processing traffic data is the same for every service. After data has been received, the data is converted into a consistent format and subsequently forwarded to the Plain Routing Bean. The main problem of converting traffic-related incidents into an appropriate format was the need for one common representation. This common representation was necessary in order to maintain the separation between the particulars of traffic incidents and the Plain Routing Bean.

Data from external services is actively polled by the Real-time Traffic Bean. For this purpose, compatible services have to implement an interface, which requires the implementation of actions which can be used to retrieve available data.

As an example for an external service, we implemented the *Bing Traffic Data Service*. The Real-time Traffic Bean was designed to actively retrieve information from this service in a 15 min interval, in order to have the latest traffic updates.

Another feature of the Real-time Traffic Bean is its connection to the graphical user interface. Since communication in multiagent systems is asynchronous, we selected the *WebSocket Protocol* (described in RFC6455 by the “Internet Engineering Task Force”<sup>6</sup>) for transmitting data between the web-based graphical user interface and the IMA Routing System. Thus, the Real-time Traffic Bean can be considered as a web socket server, handling incoming routing requests and actively pushing the traffic situation whenever it changes.

The actions provided by the OSM Routing Agent can be grouped into two categories:

- routing-specific actions, and
- traffic-specific actions.

Capabilities that were implemented for connecting the routing system to the graphical user interface fall in none of these categories.

The category of routing-specific actions contains all actions which facilitate the separation between the Real-time Traffic Bean and the Plain Routing Bean. Incoming routing requests are converted and forwarded to the Plain Routing Bean. When the route has been calculated, it is forwarded to the Real-time Traffic Bean and returned to the requesting service. Also, the status of the Plain Routing Bean is checked to ensure that it is fully functional and ready to receive requests.

In compliance with the categories for the Plain Routing Bean actions, routing actions for the Real-time Traffic Bean can be grouped into the following categories:

- searching the shortest path, and
- searching the fastest path.

The Real-time Traffic Bean does not offer actions for routing without the use of traffic information, as this is not the purpose of the bean. Traffic-specific actions offer the interfaces for other services. With these interfaces traffic information may be pushed to the Real-time Traffic Bean.

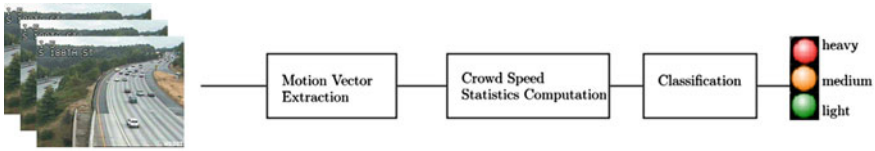
Currently the bean provides actions for the *Video Analysis* agent. We explain this component below.

### ***13.3.2 The Video Analysis Agent***

The *Video Analysis (VA)* component is implemented as a JIAC agent, namely the Video-Analysis-Agent. The aim of the VA component is to support the routing component of the IMA system (i.e., the Real-time Traffic Bean, see Sect. 13.3.1.3) by analyzing traffic flow based on video data gathered from traffic surveillance cameras

---

<sup>6</sup> Internet Engineering Task Force website: <http://tools.ietf.org/html/rfc6455/>.



**Fig. 13.3** Overview of our traffic video analysis approach for traffic congestion level estimation

(i.e., by monitoring traffic jams). The problem that is addressed by the VA component is to discover a traffic jam in the viewing range of a single camera. We perform a global (i.e., holistic, or macroscopic) analysis making use of particular features extracted from videos. The VA component receives periodically a video snapshot of 5 s length. Visual features are extracted in order to generate motion descriptors. The snapshot is classified into one of the three traffic categories *light*, *medium* and *heavy* using the motion descriptors. An overview of our holistic approach for the analysis of traffic videos is given in Fig. 13.3. The classification result of the current snapshot is passed to the routing component. In the following, we briefly review methods which have been proposed to analyze traffic surveillance videos, then introduce our method, and finally provide and discuss evaluation results on the UCSD highway traffic dataset [5].

### 13.3.2.1 Related Work

Computer vision has been successfully applied to the analysis of traffic scenes with the aim of determining traffic congestion, counting vehicles, identifying license plates, and detecting incidents among others. The survey by Buch et al. [4] provides a comprehensive review of solutions used to tackle those problems. Our aim is to use computer vision for congestion level estimation in traffic videos. Taking into account the relevant literature, we observed that the methods for traffic video analysis for congestion estimation can be classified into two main categories. Namely, those based on object detection and tracking, and those based on a global (i.e., holistic, or macroscopic) analysis making use of particular features extracted from the videos.

In the methods of the first category, the focus is on detecting and tracking individual objects (e.g., a single vehicle) in order to infer higher level traffic information (e.g., counting or congestion level estimation). Such a chain of operation—detection, tracking, and inference—appears as a natural way of processing, as this is how a human operator would interpret a scene appearing in a video.

Lien and Tsai [14] have proposed an approach falling into this first category. Their method uses vehicle detection and tracking to count vehicles and detect traffic jams. Detection is performed in a multistep fashion, where frame differencing is used to extract moving regions in the video, followed by morphological processing, and completed by a dual (short-term and long-term) background modeling. Orientation histogram of motion vectors of extracted moving objects is finally analyzed to

precisely detect vehicles. Tracking is achieved via Kalman filtering. Their solution is used to count vehicles and detect jams. However, according to their results, traffic jam detection is limited to identifying stationary vehicles, and the videos used for testing only show fluid traffic. Therefore, it is not clear if the method in [14] is scalable to congestion analysis in challenging situations.

In a similar manner, Wu et al. [27] use motion information for detecting vehicles. In addition to motion, they use edges located in those regions showing motion. An analysis of this edge map using heuristic rules results in the determination of individual vehicles. Vehicles are subsequently tracked. The authors have applied their algorithm to the determination of congestion levels. However, this approach shows serious limitations, in particular the detection of vehicles, which is based on heuristics. For instance, the authors make assumptions about the size of vehicles. Consequently, this method is not easily scalable, as it needs to be calibrated for each camera or scene setup.

A more advanced solution to vehicle detection and classification was proposed by Buch et al. who developed a method which makes use of motion information [3]. In their work, Buch et al. use the Stauffer-Grimson Gaussian Mixture Model to extract the moving objects (blobs) appearing in the video. Individual blobs are verified against 3-D wire frame models in order to classify vehicles.

However, although object detection and tracking have nowadays reached an unprecedented level of performance, detection and tracking are, generally speaking, not yet fully appropriate and operational in the context of traffic video analysis where videos are often lower in resolution, and where vehicles often appear occluded due to viewpoints. For instance, segmentation of individual vehicles might be problematic or tracking might fail because of such occlusions. Therefore, an increasing number of researchers of the traffic video analysis community have looked for alternatives represented by the methods of the second category.

One example is the solution proposed by Lee and Bovik [13], who adopted an approach which does not require individual detection and tracking of vehicles. Their solution is based on the global analysis of optical flow of traffic videos, followed by a statistical analysis of flow regions to extract meaningful information. Optical flow is estimated via a robust gradient-based solution [2] which allows a representation of different traffic lanes appearing in a traffic video. The subsequent statistical analysis consists in a histogram analysis of flow vectors. However, the approach was only illustrated in the context of flow anomaly detection. For instance, no determination of congestion levels was demonstrated.

Sobral et al. [24] have proposed a method based on vehicle crowd density estimation and tracking, which does not rely on individual vehicle detection or tracking. First, the extraction of moving blobs by background subtraction is used to determine crowd density. Second, crowd tracking by the Kanade-Lucas-Tomasi (KLT) tracker [23] is performed to estimate the speed of vehicle crowds. The analysis is limited to a region of interest (ROI) of size 190 by 140 pixels. The resulting crowd density and speed are concatenated into a feature vector. Classification of feature vectors is performed via various classifiers including k-nearest neighbors (k-NN), support vector machines (SVM) and neural networks (NN), and returns the level of

congestion (light, medium, or heavy). The rationale behind the use of density and speed is that heavy congestion results from high density and low speed, while light congestion is synonym of lower density and higher speed. However, the authors have not provided any information regarding the applicability of their method to scenes showing more than one traffic direction, and the results only illustrate one-way traffic situations.

A similar solution based on density and speed has been developed by Hu et al. [10]. The lane where vehicles are present is determined by aggregating the location of the moving blobs over time. A variant of the KLT [15] is used to track corner points extracted in the regions covered by moving blobs. Classification is achieved via Fuzzy Logic. Similarly to [24], the authors only demonstrated this approach on one-way situations.

In an attempt to develop an alternative to optical flow analysis, Derpanis and Wildes [6] have suggested using a holistic solution derived from the field of image texture analysis. The authors represent image dynamics using spatiotemporal orientation decomposition. The features extracted from the videos are fed into a k-NN classifier, which classifies the level of congestion of a given scene as light, medium, or heavy. The k-NN classifier was trained using annotated videos. However, as noted by the authors themselves, this approach fails in distinguishing scenes with similar dynamics (for instance, confusion exists between empty road and stopped road situations).

Another work based on optical flow analysis, but in the context of human action recognition in video surveillance applications, is the one by Martinez et al. [20]. In this work, dense optical flows using a method which captures multiscale information [19] is computed. These optical flows are then used to build histogram-based descriptors. An SVM is used to classify actions. Although not originally designed for traffic video analysis, this work provides a general motion-based descriptor which might be used for traffic scene analysis.

As an alternative to optical flow, Albiol and Mossi [1] have proposed to detect corner points appearing in moving regions in order to estimate queue lengths. The idea is that salient points such as corners normally result from the presence of cars and that asphalt regions do not contain significant amounts of corners. Consequently, a high number of corners is an indication of a densely occupied road. Motion detection is used to obtain moving regions, and corners inside those regions are determined via the Harris corner detector [9]. In their work, Albiol et al. also estimate the length of the queues using a perspective estimation method. However, this method might fail if the road is also highly textured.

### 13.3.2.2 Video Representation

The detection of moving objects is an important step for the video-based traffic monitoring. Therefore, the first step was to separate moving object(s) from image background using image segmentation according to features of the moving object(s). There are three basic approaches for moving object detection which are *background*



**Fig. 13.4** Illustration of the Farneback dense optical flow vectors for sample video frames of different traffic congestion levels in the UCSD dataset—**a** light traffic **b** medium traffic **c** and heavy traffic

*modeling, frame differencing* and *optical flow*. Background modeling is more suitable to model slowly changing backgrounds. Frame differencing, on the other hand fails in the situations with slowly moving objects and produces many “holes” in detected region(s), if object(s) exhibit poor texture. We decided to employ optical flow methods for the detection of moving object(s). Optical flow is the apparent motion of brightness patterns in images caused by the relative motion between an observer and the scene. The main concern about optical flow is its time-consuming computation. However, recent works claim that optical flow can be computed fast and accurately. We explored state-of-the-art sparse and dense optical flow extraction methods to detect moving objects. In our final implementation, in order to extract dense optical flow vectors we decided to use the Farneback method [8] due to its superior classification performance. The optical flow vectors are low-level features and provide a rough overview of the traffic flow in the scene. Therefore, they are processed to construct meaningful motion descriptors (i.e., mid-level features) based on their statistics. In Fig. 13.4, sample video frames of different traffic levels are shown with the corresponding dense optical flow vectors.

### 13.3.2.3 Traffic Flow Analysis Model

As the final step of VA, we train a multiclass SVM in order to learn a traffic flow analysis model. This model is trained to classify traffic flow extracted from videos into one of the following three categories: *heavy*, *medium*, or *light*. Training of the model was performed using the UCSD dataset which is introduced in Sect. 13.3.2.4.

### 13.3.2.4 Video Dataset and Ground Truth

We currently use the UCSD highway traffic dataset which contains 254 highway traffic videos of daytime highway traffic in Seattle (Washington, USA) in order to build our traffic analysis model. The videos are of 5 s length with  $320 \times 240$  resolution recorded at 10 frames per second (fps) and collected from a single static

**Table 13.1** Characteristics of the UCSD Highway Traffic Dataset

Level	Description	No. of videos
Light	Free-flow traffic. Low number of vehicles at high speed	165
Medium	Average number of vehicles at reduced speed	45
Heavy	Stop-and-go traffic. High number of vehicles at very low speed	44

traffic surveillance camera over two days. In the dataset, each video is manually annotated as *light* (i.e., free-flowing traffic), *medium* (i.e., traffic at reduced speed), or *heavy* (i.e., stopped or very slow speed traffic) traffic. The dataset is challenging, since a multitude of weather conditions are represented (e.g., clear, raining, and overcast). Table 13.1 presents the main characteristics of the dataset in more detail.

### 13.3.2.5 Experimental Setup

OpenCV Java<sup>7</sup> is used to implement the extraction of optical flow vectors and to generate the traffic flow analysis SVM model. Motion descriptors are constructed based on optical flow vectors as explained in Sect. 13.3.2.2. We trained the multiclass SVM model with an Radial Basis Function (RBF) kernel. SVM parameters were optimized by fivefold cross-validation on the training data.

We adopted the same training and testing methodology as [5, 24]. We repeated the tests four times with different training and test samples, where in each repetition the dataset was split with 75 % for training and cross-validation and 25 % for testing.

### 13.3.2.6 Results and Discussion

In this section, we present the classification performance of our method together with the related confusion matrices. We also compared our SVM-based method with other classification schemes such as k-NN, Naive Bayes, and AdaBoost learning methods. The classification performance of each method is shown in Table 13.2. We achieved 94.90 % classification accuracy on average with motion vector-based representations and multiclass SVM on the UCSD dataset. In addition, the multiclass SVM outperformed other classification methods such as k-NN, Naive Bayes, and AdaBoost.

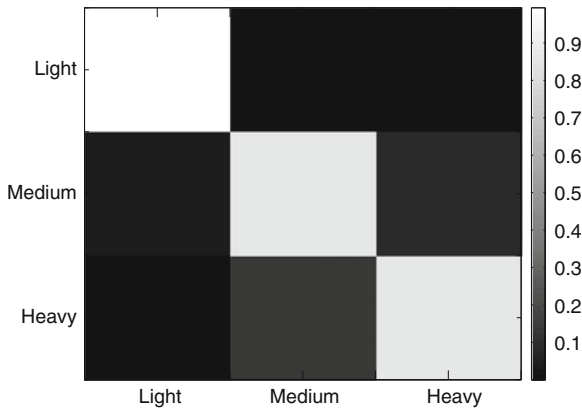
In Fig. 13.5, the confusion matrix of the classification results of our method for the UCSD dataset is illustrated. The confusion matrix represents the performance of our method with motion vector-based representations using a multiclass SVM. The detailed definition of the labels presented in Fig. 13.5 is given in Sect. 13.3.2.4. As illustrated in Fig. 13.5, *medium–heavy* pairs are the most confused congestion-level label pairs. This result suggests that there is a need to incorporate additional visual

<sup>7</sup> <http://opencv.org/opencv-java-api.html>.



**Table 13.2** Classification accuracy (in percent) of k-NN, Naive Bayes, AdaBoost, and multiclass SVM on the UCSD dataset

Method	Trial-1	Trial-2	Trial-3	Trial-4	Average
multiclass SVM	96.83	90.63	96.88	95.24	94.90
k-NN	93.65	87.5	93.75	92.06	91.74
Naive Bayes	71.43	79.69	75.00	77.78	75.98
AdaBoost	73.02	70.31	73.44	73.02	72.45

**Fig. 13.5** Confusion matrix of the classification results of our method on the UCSD traffic highway dataset (Mean accuracy: 94.90 %)**Table 13.3** Classification accuracy (in percent) of our method, Sobral et al. [24], Chan and Vasconcelos [5] on the UCSD dataset

Method	Trial-1	Trial-2	Trial-3	Trial-4	Average
Chan et al. [5]	N/A	N/A	N/A	N/A	95.00
Our method	96.83	90.63	96.88	95.24	94.90
Sobral et al. [24]	95.20	95.30	93.80	93.70	94.50

features for the representation of video segments to be able to better discriminate between *medium* and *heavy* traffic.

Table 13.3 provides a comparison of our approach with the solutions developed by Sobral et al. [24], Chan and Vasconcelos [5] in terms of classification accuracy. Since we adopted the same evaluation strategy as [5, 24], the results are directly comparable. Our method which uses dense optical flow-based motion descriptors and multiclass SVM demonstrates comparable performance and very promising results on the UCSD dataset. In addition to its classification accuracy, our method processes and analyzes 15 frames per second on average, without using any special purpose hardware. This makes our approach suitable for real-time video analysis.



**Fig. 13.6** Sample frames of video snapshots misclassified by our method. *Blue*-bordered frames are of *heavy* traffic videos misclassified as *medium* traffic. *Red*-bordered frames are of *medium* traffic snapshots misclassified as *heavy* traffic. *Green*-bordered frames are of *medium* traffic snapshots misclassified as *light* traffic. *Yellow*-bordered frames are of *light* traffic videos misclassified as *medium* traffic

In Fig. 13.6, sample frames of misclassified video snapshots are presented. Our method was unable to properly discriminate 13 out of 254 video snapshots. We observe that construction of motion descriptors on the video snapshot level instead of the frame level is one promising possibility to further proceed in order to reduce confusion between different traffic situations.

### 13.4 Scenario Simulation

In this section, we present a scenario where we explain the advantages of using different information sources such as surveillance video analysis in the IMA system. As in many other huge cities, living in Berlin, Germany can be sometimes exhausting. Every “Berliner” spends around 70 min on average in the traffic per day [11]. This means that an efficient and intelligent routing solution is expected to have a big impact in our daily lives. It also explains why we need a system like IMA.

Imagine the typical scenario of a lady in Berlin. This person possesses a bicycle and a car. She avoids using her car though. The city is extensively covered by roads equipped with sections reserved for bicycles. Cycling around the city is, therefore, quite safe and convenient. She takes the same route every day. Therefore, she already knows the location of construction sites and sections with a higher risk of traffic jams on her way. However, today is different than other days. Outside, it is raining heavily, so she would like to get to work by car.

In Fig. 13.7, the shortest route for her commute is presented. However, this shortest route is currently blocked by construction sites (denoted by exclamation mark signs along the way on the map). As can be seen from Fig. 13.7, there is also a static traffic surveillance camera installed along her route (depicted by a camera sign).

In Fig. 13.8, we present the route suggested by the IMA system, planned when considering only information about traffic incidents. The system discards the ways

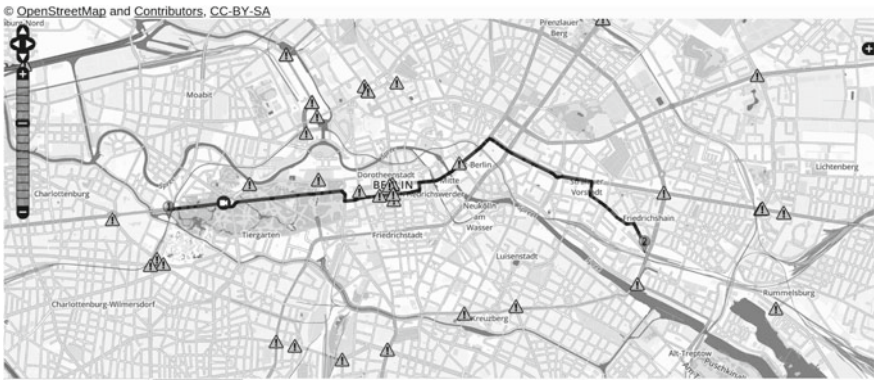


Fig. 13.7 Illustration of the shortest route from Marchlewskistraße to Ernst-Reuter-Platz in Berlin (traffic incidents and the position of surveillance cameras are shown)



Fig. 13.8 Same start and end point as in Fig. 13.7 (standard traffic information enabled during routing)



flow analysis model on the UCSD dataset, and noted a performance of 94.90% average accuracy over four trials. This is the demonstration that our holistic video analysis approach is very promising.

We plan to extend the current video feature set based on dense optical flow vectors by integrating more sophisticated motion descriptors such as *vehicle crowd density* using background subtraction and/or optical flow methods, and *trajectory-based features* by tracking sampled points in the optical flow fields.

The task of finding the best possible way through the city with a focus on integrating the gathered traffic information has been addressed by the *PlainRoutingBean*. This component works with the A\*-algorithm on a Digraph representing the streets of Berlin. The response time of this component (i.e., the time required by the *PlainRoutingBean* to receive a request, calculate a route, and return the result) always lies below 400 ms, and between 50 and 100 ms in most of the cases. This good performance is achieved at the expense of considerable memory consumption. It takes around 3 Gbyte of RAM in order to run smooth. Reducing this memory consumption is one possible suggestion for future optimization. We also expect that the response time can be reduced by several milliseconds. In the current implementation, only little effort has been put into performance optimization, as the attention was centered on the development of the functionality itself. This could also be addressed through future improvements.

The *RealtimeTrafficBean* currently supports live traffic data only. A mechanism for storing and loading traffic situations is planned. This enables the components to plan future routes with a realistic view on the traffic situation for the given time. In addition, integrating more interfaces for different service providers into this bean (e.g., real-time data from cars) is another possibility.

**Acknowledgments** This work is funded by the *Federal Ministry of Education and Research (BMBF)* under funding reference number 01IS12049.

## References

1. A. Albiol, J.M. Mossi, Video-based traffic queue length estimation, in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, (IEEE, 2011), pp. 1928–1932
2. M.J. Black, P. Anandan, The Robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.* **63**(1), 75–104 (1996)
3. N. Buch, J. Orwell, S.A. Velastin, Detection and classification of vehicles for urban traffic scenes, in *5th International Conference on Visual Information Engineering, 2008. VIE 2008*, (IET, 2008), pp. 182–187
4. N. Buch, S. A Velastin, J. Orwell, A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **12**(3), 920–939 (2011)
5. A.B. Chan, N. Vasconcelos, Probabilistic kernels for the classification of auto-regressive visual processes, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1 (IEEE, 2005), pp. 846–851
6. K.G. Derpanis, R.P. Wildes, Classification of traffic video based on a spatiotemporal orientation analysis, in *2011 IEEE Workshop on Applications of Computer Vision (WACV)* (IEEE, 2011), pp. 606–613

7. E.W. Dijkstra, A note on two problems in connexion with graphs. *Numerische Mathematik* **1**(1), 269–271 (1959)
8. G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in *Scandinavian Conference on Image Analysis*, pp. 674–679 (2003)
9. C. Harris, M. Stephens, A combined corner and edge detector, in *Alvey vision conference*, vol. 15, (Manchester, 1988), p. 50
10. S. Hua, J. Wua, L. Xub, Real-time traffic congestion detection based on video analysis. *J. Inf. Comput. Sci.* **9**(10), 2907–2914 (2012)
11. H. Jahn, J. Krey, Berliner verkehr in zahlen. Technical report, Senatsverwaltung für Stadtentwicklung und Umwelt (2014)
12. N.R. Jennings, M. Wooldridge, Agent-oriented software engineering. *Artif. Intell.* **117**, 277–296 (2000)
13. J. Lee, A.C. Bovik, Estimation and analysis of urban traffic flow, in *2009 16th IEEE International Conference on Image Processing (ICIP)*, (IEEE, 2009), pp. 1157–1160
14. C.-C. Lien, M.-H. Tsai, Real-time traffic flow analysis without background modeling. *J. Inf. Technol. Appl.* **5**(1) (2011)
15. B.D. Lucas, T. Kanade et al., An iterative image registration technique with an application to stereo vision, in *IJCAI*, vol. 81, pp. 674–679 (1981)
16. M. Lützenberger, T. Küster, T. Konnerth, A. Thiele, N. Masuch, A. Heßler, J. Keiser, M. Burkhardt, S. Kaiser, S. Albayrak, JIAC V—a MAS framework for industrial applications (extended abstract), in *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems, Saint Paul, Minnesota*, ed. by T. Ito, C. Jonker, M. Gini, O. Shehory, pp. 1189–1190 (2013)
17. M. Lützenberger, T. Küster, T. Konnerth, A. Thiele, N. Masuch, A. Heßler, J. Keiser, M. Burkhardt, S. Kaiser, J. Tonn, M. Kaisers, S. Albayrak, A multi-agent approach to professional software engineering, in *Engineering Multi-Agent Systems – First International Workshop, EMAS 2013, St. Paul, MN, USA Paul, MN, USA May 6–7, 2013, Revised Selected Papers*, Lecture Notes in Artificial Intelligence, vol. 8245, ed. by M. Cossentino, A.E.F. Seghrouchni, M. Winikoff (Springer, Berlin, 2013), pp. 158–177
18. C.M. MacKenzie, K. Laskey, F. McCabe, P.F. Brown, R. Metz, B.A. Hamilton, Reference model for service oriented architecture 1.0, October 2006, <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>. Accessed 16 May 2014
19. A. Manzanera, Local jet feature space framework for image processing and representation, in *Seventh International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, (IEEE, 2011), pp. 261–268
20. F. Martínez, A. Manzanera, E. Romero, A motion descriptor based on statistics of optical flow orientations for action classification in video-surveillance, in *Multimedia and Signal Processing* (Springer, New York, 2012), pp. 267–274
21. F. Ramm, J. Topf, S. Chilton, *OpenStreetMap: Using and Enhancing the Free Map of the World*, 1st edn. (UIT Cambridge Ltd., Cambridge, 2010)
22. S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*. 2nd edn. Artificial Intelligence, (Prentice Hall, 2003)
23. J. Shi, C. Tomasi, Good features to track, in *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR'94*, (IEEE, 1994), pp. 593–600
24. L.O.A. Sobral, L. Schnitman, F. De Souza, Highway traffic congestion classification using holistic properties, in *IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)* (2013)
25. K.P. Sycara, Multiagent systems. *AI Mag.* **19**(2), 79–92 (1998)
26. M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice. *Knowl. Eng. Rev.* **10**(2), 115–152 (1995)
27. B.-F. Wu, C.-C. Kao, J.-H. Juang, Y.-S. Huang, A new approach to video-based traffic surveillance using fuzzy hybrid information inference mechanism. *IEEE Trans. Intell. Transp. Syst.* **14**(1), 485–491 (2013)

# Index

## A

- A\* search, 353
- Agent, 102, 114, 120
  - broker agent, 108, 115–118, 120
  - crawling agent, 116, 117
  - desktop agent, 117
  - LDAP agent, 108
  - mail agent, 117
  - OSM routing agent, 351, 353
  - plain routing bean, 351, 354
  - real-time traffic bean, 351, 356
  - search agent, 108, 116–120
  - video analysis agent, 351, 357
- Agent-oriented software engineering, *see* AOSE
- Agent-oriented software engineering (AOSE), 349
- Aggregated power consumption, 277
- Appliance state classification, 279
- Appliance states, 277
- Application
  - activity assistant, 93
  - bestTime, 337
  - DAIKnow, 250
  - enterprise Bookmarking, 250
  - gender identity disorder (GID), 84
  - health information system, 84
  - IMA routing system, 351
  - news aggregator, 40
  - nutrition assistant, 92
  - open recommendation platform, 157, 160, 161
  - PIA enterprise, 113
  - prevention service (PS), 92
  - semantic movie recommender, 140
  - sentiment tracker, 63
  - SERUM, 184

standard operating environment (SOE), 285

Approximation, 263

## B

- Background modeling, 360
- Bag-of-audio words, 299
- Bag-of-visual words, 300
- Bag-of-words, 31
- Bigrams, 31
- Bookmarking, 250

## C

- Changes in power consumption, 277, 278
- Chunking, 18
- Cold-start problem, 127, 194, 195, 198, 234
- Collection representation, 104
- Collection selection, 105, 120
- Constraint satisfaction, 216
- Contextual factors, 163, 167, 176
- Cooking session, 92
- CORI, 109

## D

- Definition
  - gamification, 243
- Delta-idf, 31, 32
- Design
  - gamification, 244
- Determinism, 325
- Dimensionality reduction, 328
- Discourse markers, 33
- Distributed information retrieval, 104, 108
- DLS, *see* document level security

Document level security, 104, 107  
 Dynamic topic hierarchy, 10, 11

**E**

Edge algebras, 135  
 Edge weight scaling models, 134  
 Energy disaggregation, 275  
 Enterprise, 101, 106, 116, 118–120, 247, 250  
   environment, 102, 103, 105  
   gamification, 245  
   search, 102, 105  
   search system, 102, 103  
   users, 106  
 Entropy, 326  
 Exercises, 93  
 Experiment, 247  
 Explicit information, 190  
 Exploration exploitation trade-off, 159

**F**

Feature vectors, 61  
 Federated search, *see* distributed information retrieval  
 Frame differencing, 361  
 Freebase, 130

**G**

Gamification, 266  
   definition, 243  
   design, 244, 252, 260, 261, 264  
   design problem, 260, 261, 264  
   enterprise, 245  
   experiment, 247, 255  
   history, 243  
   model, 261  
   user, 245, 256  
 Google search appliance, 104  
 Grails, 140  
 Graph-based knowledge representation, 85, 129, 134  
 GSA, *see* google search appliance

**H**

Heating control, 277, 285  
 HetRec, 132  
 Hierarchical clustering, 139, 329, 332  
 Holistic video analysis, 358, 360

**I**

Idf, 31, 32

Implicit information, 190  
 Index-I, 327  
 Information overload, 126, 152, 153, 215, 217  
 Internet movie database (IMDB), 130

**J**

Java-Based intelligent agent component-ware, *see* JIAC  
 Java-based intelligent agent componentware (JIAC), 113, 349  
   agent, 350  
   AgentBean, 350  
   AgentNode, 350  
   application, 350  
 Joint cross recurrence plot, 323

**K**

Knowledge aggregation, 129

**L**

LDAP, *see* lightweight directory access protocol  
 Learning, 265  
   problem, 261, 264  
   regression, 264  
 Lemmatization, 18  
 Lightweight directory access protocol, 102, 117  
 Linked open data cloud, 129  
 Living lab, 157  
 Low-rank approximation, 138

**M**

MAS, *see* multi-agent system  
 Matrix, 262, 264  
 Matrix factorization, 264  
 Maximum entropy, 65, 68  
 Maximum entropy classifier, 60  
 Metric, 109  
   accuracy, 229  
   click through rate, 162, 175, 176  
   normalized discounted cumulative gain, 109, 111  
   order-invariant distance, 319  
   precision, 109, 229  
   recall, 109  
   recurrence plot-based distance, 326  
 Micro-blogging, 51  
 MLSA corpus, 33  
 MovieLens, 132



Multi-agent system, 105, 108, 113, 120, 349  
 Multilingual data, 128  
 Multilingual search, 85

## N

Naïve Bayes, 60, 65  
 Naïve merger, 110  
 Named entity recognition, 18  
 Named-entity recognition (NER), 184  
 National endowment for democracy (NED), 184  
 Natural language processing, 53  
 Natural language toolkit, 57  
 Negation patterns, 32  
 News aggregator, 8  
 Noise reduction approaches, 136  
 Nonintrusive load monitoring, 275  
 Nutrition, 92

## O

Occupancy states, 280  
 Online shopping, 215  
 Ontology, 84, 129, 188, 190  
   health ontology, 84  
   user behavior ontology, 183, 184, 188–192  
 OpenCV, 362  
 Openstreetmap, *see* OSM  
 Openstreetmap (OSM), 353  
 Opinion mining, *see* sentiment analysis  
   corpus, 35  
   features, 31, 33  
 Optical flow, 361  
   farneback, 361  
 Order-invariance, 320, 329  
 Overlay models, 226

## P

Parsing, 58  
 Part-of-speech tagging, 18, 32  
 Personalization, 182, 190  
 Piecewise aggregate approximate, 327  
 Popularity bias, 128, 163, 164, 167, 176  
 Prediction, 264

## Q

Query refinement, 142  
 Questionnaire, 248, 257  
 Quotation  
   speaker, 12

  definition, 12  
   direct quotations, 13  
   holder, 12  
   indirect quotations, 13  
   quoted speech, 13  
 Quotation extraction, 11, 16, 25  
   corpus, 21  
   direct quotation extraction, 19  
   from news, 12  
   indirect quotation extraction, 20  
   quotation holder extraction, 21  
   reporting verb detection, 19  
 Quotation marks normalization, 16

## R

Real-time processing, 157, 159, 176  
 Recommender system, 127, 152, 153, 163, 164, 167, 186, 190, 193, 198, 199, 216–218, 264  
   clustering, 139  
   collaborative filtering, 127, 154, 159, 170, 193, 194, 196, 198, 202, 216, 217  
   constraint-based, 216, 219  
   content-based filtering, 154, 159, 173, 218  
   ensembles, 140, 230  
   explanations, 137  
   graph-based recommendation, 188  
   k-nearest neighbor, 170, 230, 231  
   knowledge-based, 219  
   matrix factorization, 170, 264  
   memory-based, 137  
   model-based, 138  
   most popular, 168  
   most recent, 169  
   Naïve Bayes, 230, 231  
   random, 169  
   semantic-based algorithms, 137  
   text-based, 139  
 Recurrence plots, 321  
 Recurrence quantification analysis, 324  
 Relatedness models, 134  
 Reported clause, 12  
 Reporting clause, 12  
 Reporting clue, 13  
 Reporting phrase, 13  
 Reporting verb, 13  
 Repository, 101, 102, 106–108, 119  
 Resistance distance, 136  
 Resource description framework (RDF), 129  
 Result merging, 105, 108, 111, 112  
 Round robin, 109, 110

**S**

Secure enterprise search, 104  
 Semantic knowledge resources, 129  
 Semantic queries, 141  
 Semantic reasoning, 58  
 Semantic search, 85  
 Semantic web, 188  
 Sentence detection, 17  
 Sentiment analysis, 12, 25, 40, 60  
   classification, 30  
   corpus, 34, 35  
   features, 31, 33  
   polarity classification, 30  
   sentiment features, 31  
   subjectivity detection, 30  
 Sentiment lexicon for German, 32  
 Sentiment types, 34  
 Sentiment words, 32  
 Sentimentwortschatz (SentiWS), 32  
 SES, *see* secure enterprise search  
 Shortest path algebra, 136  
 Single sign-on, 107  
 Singular value decomposition, 138  
 Software agent, *see* agent  
 Sparsity, 163, 167, 262, 264  
 Speech  
   direct speech, 13  
   indirect speech, 13  
   reported speech, 13  
 Stemming, 31, 58, 139  
 Stop words, 31, 139  
 Superimposed signals, 277  
 Support vector machines, *see* SVM  
 Support vector machines (SVM), 261, 302, 361  
   radial basis function kernel, 305, 362

**T**

Tagging, 58, 250  
 Tf, 31  
 Tf-delta-idf, 31, 32  
 Tf-idf, 31, 32  
 Time series  
   clustering, 319  
   discords, 321  
   distance measures, 319  
   motifs, 320  
   prototype, 335, 336

  shapelets, 321  
 Tokenization, 58  
 Toolboxes  
   libsvm, 305  
   MIR, 304  
   SPAMS, 304  
   VLFeat, 305  
 Topic detection and tracking, 8, 10  
 Tracking, 256  
 Triples, 129  
 Twitter, 49, 53, 57

**U**

Unigrams, 31  
 User  
   gamification, 245, 256  
   interaction, 256  
 User behavior, 184–186, 188–190  
 User modeling, 85, 184, 190  
 User preferences, 218  
 User profile, 152, 185–188, 193–196, 198–200, 202  
 Utility, 262

**V**

Valence shifters, 33  
 Vehicular sensor data, 333  
 Video representation  
   affect-related color features, 300  
   dense HoF, 300  
   dense HoG, 300  
   MFCC, 299  
   Violent flows (ViF), 300  
 Violence detection, 299  
   classifier selection, 302  
   feature space partitioning, 301

**W**

Warping constraint, 328  
 Web ontology language (OWL), 129, 191, 192  
 Weblogs, 52  
 Weighted MinMax, 109  
 Weighted path algebra, 136  
 Wordnet, 58