# Visual Document Analysis: Towards a Semantic Analysis of Large Document Collections

Dissertation zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften
an der Universität Konstanz im Fachbereich
Informatik und Informationswissenschaft

vorgelegt von
Daniela Oelke

Tag der mündlichen Prüfung: 13. Juli 2010

Referenten: Prof. Dr. Daniel A. Keim, Universität Konstanz
Prof. Dr. Oliver Deussen, Universität Konstanz
Prof. Dr. Gerhard Heyer, Universität Leipzig

# Abstract

Large amounts of data are only available in textual form. However, due to the semi-structured nature of text and the impressive flexibility and complexity of natural language the development of automatic methods for text analysis is a challenging task.

The presented work is centered around a framework for analyzing document (collections) that takes the whole document analysis process into account. Central to this framework is the idea that most analysis tasks do not require a full text understanding. Instead, one or several semantic aspects of the text (called quasi-semantic properties) can be identified that are relevant for answering the analysis task. This permits to targetly search for combinations of (measurable) text features that are able to approximate the specific semantic aspect. Those approximations are then used to solve the analysis task computationally or to support the analysis of a document (collection) visually.

The thesis discusses the above mentioned framework theoretically and presents concrete application examples in four different domains: literature analysis, readability analysis, the extraction of discriminating and overlap terms, and finally sentiment and opinion analysis. Thereby, the advantages of working with the above mentioned framework are shown. A focus is put on showing where and how visualization techniques can provide valuable support in the document analysis process. Novel visualizations are introduced and common ones are evaluated for their suitability in this context. Furthermore, several examples are given how good approximations of semantic aspects of a document can be found and how given measures can be evaluated and improved.

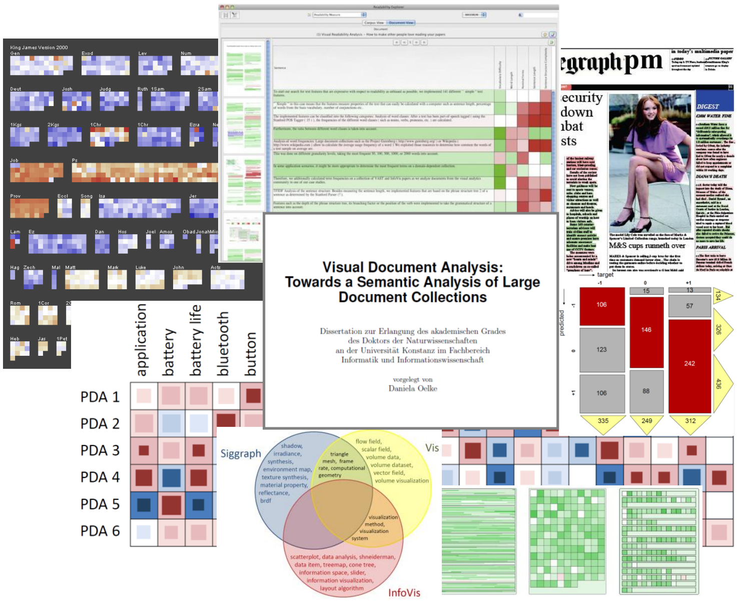
*Figure 1: Graphical summary, showing frequent terms of the dissertation. The image was generated with the tool Wordle (www.wordle.net).*

# Zusammenfassung

Viele Daten sind nur in textueller Form verfügbar. Da Text zu den semi-strukturierten Datentypen gehört und natürliche Sprache sich durch erstaunliche Flexibilität und Komplexität auszeichnet, stellt die Entwicklung von automatischen Methoden eine herausfordernde Aufgabe dar.

Der vorliegenden Arbeit liegt ein Framework zur Analyse von Dokumenten(kollektionen) zugrunde, das den gesamten Analyseprozess berücksichtigt. Die zentrale Annahme des Frameworks ist, dass die meisten Analyseaufgaben kein vollständiges Textverständnis erfordern. Stattdessen können ein oder mehrere semantische Aspekte identifiziert werden (genannt quasi-semantische Maße), die relevant für die Bearbeitung einer Analyseaufgabe sind. Das erlaubt es, gezielt nach Kombinationen von (messbaren) Texteigenschaften zu suchen, die in der Lage sind, den spezifischen semantischen Aspekt zu approximieren. Diese Approximation wird dann verwendet, um die Analyseaufgabe maschinell zu bearbeiten oder um Unterstützung durch Visualisierungstechniken anzubieten.

Die Doktorarbeit diskutiert das oben genannte Framework theoretisch und präsentiert konkrete Anwendungsbeispiele aus vier verschiedenen Domänen: Literaturanalyse, Lesbarkeitsanalyse, Extraktion von diskriminierenden und überlappenden Termen, sowie Stimmungs- und Meinungsanalyse. Hierbei werden die Vorteile aufgezeigt, die eine Arbeit mit dem Framework mit sich bringt. Ein Schwerpunkt wird darauf gelegt, wo und wie Visualisierungstechniken gewinnbringend im Analyseprozess eingesetzt werden können. Neue Darstellungsarten werden vorgestellt und bewährte Techniken auf ihre Tauglichkeit in diesem Kontext untersucht. Darüber hinaus werden mehrere Beispiele dafür gegeben, wie gute Approximationen von semantischen Aspekten gefunden werden können und wie vorhandene Maße evaluiert und verbessert werden können.



Visual Document Analysis:
Towards a Semantic Analysis of Large Document Collections

Dissertation zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften
an der Universität Konstanz im Fachbereich
Informatik und Informationswissenschaft

vorgelegt von
Daniela Oelke

King James Version 2000
application
battery
battery life
bluetooth
button
PDA 1
PDA 2
PDA 3
PDA 4
PDA 5
PDA 6
Siggraph
shadow, irradiance, synthesis, environment map, texture synthesis, material property, reflectance, brdf
triangle mesh, frame rate, computational geometry
flow field, scalar field, volume data, volume dataset, vector field, volume visualization
Vis
visualization method, visualization system
scatterplot, data analysis, shneiderman, data item, treemap, cone tree, information space, slider, information visualization, layout algorithm
InfoVis
M&S cups runneth over
DIANA'S DEATH
PARIS ARRIVAL
target
predicted
-1
0
+1
106
15
13
57
146
123
242
106
88
134
326
436
335
249
312

*Figure 2: Collage verschiedener Visualisierungstechniken aus der Dissertation.*

# Acknowledgements

First and foremost, I wish to thank my supervisor Prof. Daniel A. Keim for the opportunity to work in his group and for his enduring support and motivation. I can indeed say that I learned a lot about research from him in the last years. His guidance significantly formed the course of this thesis.

I also would like to thank Prof. Oliver Deussen and Prof. Gerhard Heyer, not only for taking the time to serve in the thesis committee, but much more for the fruitful discussions, their valuable advise and the encouragement in the past months and years.

Thinking of my colleagues, I remember fruitful joint work, interesting discussions, and very helpful assistance. You all shaped the friendly, relaxing, but yet productive atmosphere in our group which I really enjoyed.

I also would like to mention all the student workers who did not only help with the implementation but often contributed with fresh and valuable ideas to the projects. Furthermore, I am grateful to our DBVIS support team for providing very reliable services which really made life easier.

I appreciated being an associated PhD student in the Graduiertenkolleg (PhD Graduate Program) “Explorative Analysis and Visualization of Large Information Spaces”. Thanks to all members of the GK for widening my horizon on computer science research and for providing valuable comments wherever possible.

One of the outstanding stages during my PhD time was my visit to the HP labs in Palo Alto which gave me a chance to see industrial research at first hand. Thanks to Ming Hao and Umeshwar Dayal for their support and collaboration which made this visit possible.

Finally, I am truly grateful for all the people outside the university who shared their time and life with me and thereby indirectly provided non research-oriented but not less important support for the thesis or simply made life more enjoyable.

# Contents

# 1 Motivation

## Contents

Books, newspaper articles, patents, service reports, protocols, . . . - large amounts of written information are not available in a structured form but as text. According to a study of the University of Berkeley that was published in 2003 [13], about $1075 \cdot 10^7$ pages of original office documents are produced and printed per year. The same study states that worldwide there are about 25,276 different newspaper publications, 80,000 mass market and trade periodicals, 37,609 scholarly periodicals, 40,000 newsletters and about 950,000 new books each year. All the above mentioned numbers refer to unique, original publications (no copies). Furthermore, about 31 billion e-mails are sent daily. The above mentioned study reported in 2003 that it is expected that this number would double by 2006. Recent (unverified web-)studies [3] even estimate the number of e-mails per day in 2008 to be around 210 billion. Another digital text source, the surface web, was estimated to contain about 167 terabyte of data in 2003 (20-50 TB in 2000, which means that the value at least tripled in those three years). The deep web was estimated to be about 400 to 450 times larger which refers to 66,800 and 91,850 TB of data. Of course this does not only comprise textual content and all the numbers are just estimates. But what is sure is that the internet greatly simplifies publishing your own thoughts and ideas, and that this opportunity is willingly taken by more and more people. This gives rise to a huge interesting and freely available source of information.

Luckily, for nobody there is a need to read through *all* of the above mentioned text sources. But everybody knows the feeling of having to discard information that could be valuable, because you do not have the time to read everything you want. Since a lack of information can result in wrong decisions and manually evaluating the available sources is often not feasible, more and more companies are becoming interested in technology that supports and speeds up document analysis processes. This is reflected in the emergence of lots of new companies and commercial activities in this sector that provide help to get value out of the often underutilized textual data resources.

## 1.1 Example scenarios for document analysis tasks

In this section, different analysis scenarios are introduced. They are taken from different areas of life to show how widespread document analysis tasks are[1]. Furthermore, the need for automatic support in document analysis should become clear as most of the tasks require working on large amounts of textual data and reading everything manually is a very time-consuming task.

The examples were chosen in a way that as many different tasks as possible become obvious. Chapter 2 refers back to them to exemplify the framework. They are *not* chosen with respect to what is already possible but rather should show how challenging and demanding document analysis can be. Later chapters discuss how realistic solving those tasks automatically is and point out open research issues. Additionally, the chapters 4 to 7 present concrete solutions for some of the scenarios to show the applicability of the proposed framework.

### 1.1.1 Example scenarios for document analysis tasks of companies

In companies data that comes in textual form ranges from office documents, contracts, and patents, over call center notes, customer feedback and service reports, to e-mails and letters (just to name a few examples). Managing and analyzing this large data source is not always easy and thus, often this kind of data is underutilized. In addition to that, companies are also interested in news about their competitors that can be found on web pages, in newspapers or in press releases. In the following, some examples for typical document analysis tasks in companies are given:

**Analysis of Customer Feedback**
Many companies collect customer feedback in online stores or on their web pages. They would like to know what their customers like and what they complain about. Analyzing the data with respect to those questions can help to improve products and services and to keep customers satisfied or gain new ones. Of course, this kind of opinion analysis is not only interesting for companies but also for customers. According to [105], 81% of the Internet users look for information about services and products in the Internet and read what other customers posted about it. For them, getting an idea of the advantages and disadvantages of a product can help to make the right choices.

**Finding out the current market buzz**
Besides analyzing feedback that has been directly addressed to a company, it is also important to know the current market buzz about the company and its competitors. The past has shown that rumors can be the death of a company or at least can seriously harm it. Nowadays, rumors are not only spread by word of mouth or in newspapers but also

[1] Please note that the assignment into one specific area is not always unambiguous. Many of the introduced document analysis tasks are present in several areas.

in the Internet (e.g. in blogs, forums, and on web pages). Knowing about new rumors in time enables a company to react quickly and avert serious harm. Similarly, knowing about its competitors situation enables a company to adapt its own strategy accordingly. Related to this is the need of politicians to know what their voters think about them. Undoubtedly, the vast size and fast evolution of the internet makes it difficult to manually keep track of what is going on.

**Response management**
Each day many e-mails in which customers ask for support or request some information have to be processed. Much time can be saved if those e-mails are automatically forwarded to the responsible case worker. Standard requests are answered automatically with templates.

### 1.1.2 Example scenarios for document analysis tasks of researchers

Since publications and proposals play a vital role in scientific environments, researchers are a community for which it is common to be confronted with large amounts of documents. In the following, some examples of document analysis tasks of researchers are given.

**Browsing through large paper collections**
Among the frequently recurring tasks of a researcher is to search for papers that are related to the own working area or that include concepts that could be used to improve one's own approach. This requires reviewing large paper collections and also includes keeping track of current trends in research.

**Assessment of papers and proposals**
Researchers are often asked to assess the work of other researchers in their area. In this case the submitted publications or proposals have to be checked for completeness with respect to the cited related work. Furthermore, copying text from other publications without explicitly marking it would be considered as plagiarism and is not allowed. A high quality publication or proposal introduces novel ideas and approaches and describes them in a clear, understandable manner. Additionally, it is expected that the publications are in concern with the conventions of the specific community.

### 1.1.3 Example scenarios for document analysis tasks of literary scholars

For linguists and literary scholars natural language text itself is the subject, they are doing research on. So far, most of the work is done manually. Automatic support would permit to take more data into account. However, some of their tasks are challenging with respect to automatization. Below some examples for document analysis tasks in literature analysis are listed.

**Analysis of novels**
Literature scholars analyze the stories and the writing style of novels in detail and with respect to many criteria. This includes, for example, the analysis of the behavior of the

figures, their relationship to each other and the development of the story in general. Besides this, the text is also analyzed with respect to attributes that are more difficult to grasp such as the question what the writer wanted to achieve. Does he or she comment on problems of society? What kind of literary allusions are used?

**Literary Quality**

Also very challenging and often quite arguable is the analysis of a book with respect to its literary quality. In figure 1.1 some novelists and literary scholars are cited whose statements cast some light on what could be analyzed in a book in order to assess its literary quality. For sure, many more criteria could be mentioned here. The quotations that are listed here have been chosen because they provide some interesting examples of how complex and far-reaching literature analysis can be.

What is wonderful about great literature is that it transforms the man who reads it towards the condition of the man who wrote.
*E. M. Forster*

Books can be dangerous. The best ones should be labeled “This could change your life.”
*H. Exley*

The best effect of any book is that it excites the reader to self activity.
*T. Carlyle*

The worth of a book is to be measured by what you can carry away from it.
*J. Bryce*

A good book has no ending.
*R.D.Cumming*

*Figure 1.1: Quotations of writers and literary scholars on the assessment of literary quality*

**Determining the age of the target audience of a book**

One of the tasks of a librarian is to decide what age group a book is suitable for. To make this decision, different aspects have to be taken into account such as how easy the book is to read and how complex the story is. Furthermore, questions like what kind of emotions are aroused by the book or what age the average reader has to be to understand the topic that is discussed could play a role.

**Authorship attribution**

Given several text samples with known authorship and a text with unknown authorship, the task of Authorship Attribution is to predict the author of the text with unknown authorship or to state how probable it is that all given texts have been written by the same person. This requires an analysis of the writing style of the text samples. If used in forensic authorship attribution, it is especially important to search for aspects in the writing style that are difficult to control consciously to unmask possible fakes. Besides its usage in a court proceeding in which the authorship of some text is disputed (e.g. of an offensive e-mail), authorship attribution also plays an important role in literature analysis, e.g. when a previously unknown old poem has been found that some people assign to a famous poet but others appeal against.

### 1.1.4 Example scenarios for document analysis tasks of Internet users

So far, most examples were in some way or another related to the work of companies, institutions or researchers. However, nowadays we are also privately confronted with an abundance of textual information. Especially the Internet provides easy access to textual resources from all over the world.

**Assessing the quality of Internet content**
According to a survey in December 2008 [105], 79% of the American adults use the internet. Among other things, 83% of them stated that they use the internet to look for health and medical information. The same amount of people research information about a hobby or interest and 73% say that they search for news in the internet. One of the problems when using the internet as a source of information is that most of the content has not been peer-reviewed or otherwise checked for quality standards. It is the task of the user to assess how trustworthy the available content is. This is something that has to be learned and different people will have different strategies (e.g., checking the author if possible, the source of information (well known newspaper versus blog), quality of writing etc). Most people are able to develop an intuition after a while how trustworthy the presented content is. However, thorough examination is often difficult and takes a lot of time.

## 1.2 Special characteristics of the data type “text”

In the previous section many examples for document analysis tasks are given. In many cases the data is available and the analysis question can be clearly formulated. But still, often there is no good way of supporting the task computationally so far. There are several reasons why it is challenging to process text automatically. Some of the problems are caused by the vast amount of words that exist, of which many of them have several meanings. Furthermore, many language rules are not strict and invariant but allow for much flexibility which makes natural languages so powerful. This is aggravated by the fact that humans incorporate much additional knowledge when interpreting a text besides processing the meaning of the words and the order they are put in. Farghaly puts it this way: *“When humans interact using language, they subconsciously make use of their knowledge of the world, situation, linguistic context, interlocutor, and common sense.”* ([34], page 6).

The following subsections will give some examples of what is special about the data type text. This illuminates, why it is difficult to automatically process text and what challenges algorithms are confronted with.

### 1.2.1 Linguistic levels of natural language

In order to understand a text, the reader or listener has to be able to process the different linguistic levels of text:
(definitions are taken from [68])

- *Phonetics and Phonology* - knowledge about linguistic sounds.
  Knowledge about those two aspects is only important if an audio signal with text has to be processed or the correct acoustic sound has to be produced.

- *Morphology* - knowledge of the meaningful components of words.
  Morphology comprises the knowledge about the rules by which words are formed and how to break them down into the smallest components that carry meaning (e.g., “I'm” → “I am” or “dogs” → “dog + plural 's' ”).

- *Syntax* - knowledge of the structural relationship between words.
  Syntax is the knowledge about the rules for the arrangement of words into phrases and of phrases into sentences.

- *Semantics* - knowledge of meaning.
  With respect to semantics *lexical semantics* and *compositional semantics* can be distinguished. Where the first one denotes the meaning of words, the second one describes the fact that often the context of a word has to be taken into account to correctly understand its meaning (e.g. “what exactly constitutes 'Western Europe' as opposed to Eastern or Southern Europe, what does 'end' mean when combined with 'the 18th century' ” ([68], p. 37))

- *Discourse* - knowledge about linguistic units larger than a single utterance.
  In a normal text, it is not possible to process every sentence separately. In order to get a coherence between the sentences, writers refer to previous sentences or omit words that are clear from the context. E.g., in the sentence “I like them.” it is not clear what “them” refers to if the context cannot be taken into account.

- *Pragmatics* - knowledge of the relationship of meaning to the goals and intentions of the speaker.
  Usually, not only a fact is expressed with an utterance but the speaker has some intention with what he or she says. Exactly the same sentence may therefore convey different things in different situations. Consider, for example, the sentence “It is cold in here”. It is possible to understand the semantics of the sentence without knowing the context it is in. But in order to know what the intention of the speaker is, the situation that the speaker is in has to be known (e.g. the window could be open and this could be a request to close it). Pragmatics is therefore also defined as “the branch of linguistics which seeks to explain the meaning of linguistic messages in terms of their context of use” ([94], page 137).

Since this thesis is only about the analysis of written text, knowledge about phonetics and phonology is not required to process the data. (Exceptions may be some special analysis task of linguists in which written text is analyzed with respect to pronunciation.) However, all other levels of language are important to answer the questions of section 1.1.

### 1.2.2 Ambiguity of natural language

Another reason why it is difficult to teach a computer how to truly understand natural language is its inherent ambiguity. Because ambiguities in natural language are such a common problem in automatic text analysis, many text analysis models and algorithms are centered around resolving them [68]. Part-of-speech tagging, word sense disambiguation, speech act interpretation or probabilistic parsing are examples for linguistic algorithms that are designed to help resolving ambiguities.

A text can be ambiguous with respect to *Syntactic Ambiguity* and *Semantic Ambiguity*. For example in the sentence “Stolen painting found by tree”, it is impossible to tell (without knowledge of the world), if the painting has been found near the tree or if it was the tree that found the painting [62]. This is known as syntactic ambiguity. In sentences that are syntactically ambiguous it cannot be told for sure which parts of the sentence refer to each other. There are several valid ways to syntactically parse the sentence. An example for semantic ambiguity is the following: The sentence “Iraqi head seeks arms” contains several words that are ambiguous with respect to their meaning. “Head” could either be a job description meaning that this person is a chief or could denote a specific part of the body. Similarly, “arms” could be interpreted as weapons or again as a part of the body. Semantic ambiguity arises when a word has more than one meaning.
In both cases (syntactic and semantic ambiguity), context knowledge can help to find the right answer. The sentences above exemplify that *“the meaning of a text is not derived from just the meaning of the words, phrases, and sentences that comprise it, but rather from the situation in which it is used.”* ([34], page 6).

Text analysis systems are not only afflicted with domain dependency but can also benefit from it. If it is known which domain a text belongs to, the space of possible resolutions of ambiguities can be greatly narrowed.

### 1.2.3 Noise in natural language data

Another challenge when parsing natural language text is that it can be noisy. Besides misspellings or inadvertently made grammar mistakes, there are occasions in which the usage of “noisy” language is accepted or even part of the community’s self-definition. This is for example true for user-generated content in the Internet. Farghaly states in [34] that the language in the Internet is characterized by *“incoherence, misspellings, truncation of words and sentences, and violation of basic grammatical and punctuation rules. Moreover, this language continues to change and, in the process, develops its own conventions and symbols.”*. Furthermore, second language errors are frequently to be expected and sometimes capitalization rules are completely ignored. Similarly, SMS messages that are used to communicate with written text via mobile phones are characterized by many abbreviations and omissions of words due to the necessity to keep them as short as possible.

As stated above, users of the Internet tend to develop their own conventions with respect to language rules and usage. A similar development can be observed when comparing texts of different ages. Some of the words that we use today have not been known in former days and vice versa. Other words slightly altered their meaning. Additionally, it can be observed that grammar rules change over time. Thus, systems that are trained on modern languages are not necessarily able to work correctly on older documents (e.g. from the Middle Ages). Even more difficult is to work with text of different ages at the same time (which could be necessary, for example, to analyze the development of languages over time).

Finally, the analysis of spoken language has to be mentioned as a special challenge. Usually, spoken dialogues are not well structured and incomplete sentences are very common. Sometimes, spoken language also exists in written form, e.g. verbatim protocols of a meeting of parliament.

### 1.2.4 Text as semi-structured data type

Most of the classical data analysis and data mining algorithms were developed for structured data. Structured data is data that can be put into a relational database easily, because its semantic entities can be grouped together into relations or classes, and entities in the same group can be described with the same attributes. Due to the large amount of different words (i.e. entities) that exist, its inherent ambiguity and the different linguistic levels a text consists of, it is not possible to convert text into a fully structured format without losing information. Text is therefore sometimes called unstructured. However, given the definition that unstructured data does not follow any rule, is not predictable and that no pattern or sequence or specific format is recognizable, textual data also clearly does not fall into this category. Instead, it is often called semi-structured. Although it is not fully structured, still the following observations apply:

1. Words cannot be put in an arbitrary order but are constraint by grammar rules. Equally, in each language, constraints exist that restrict possible orders of letters within a word.

2. Context plays an important role in interpreting a text and narrows the space of possible words and meanings of the following sentences (and therefore makes text somehow predictable).

3. Many documents contain formatting information that conveys structural information such as setting apart the title or author names, separating paragraphs, or putting text into tables etc.

Because of the big differences that exist with respect to argument 3 in the list above, [37] suggests to further distinguish between “free-format or weakly structured text documents” (e.g. plain text documents) and the real “semi-structured text documents” (e.g. XML documents or Latex files) that have strong typographical, layout, or markup indicators.

In this thesis I am going to use the term *“text”* to denote weakly structured text documents and *“document”* for textual data that additionally contains some layout or other structural information. Furthermore, the term *“document”* is used as a hypernym for all kinds of textual data. Similarly, the term *“document analysis”* is used where the general character of the analyzed textual data set is to be emphasized. *“Text analysis”*, same as *“text”*, is used when the analysis of the textual content of a document is central to the analysis (and not the meta-data that might come with it) or if it is referred to *“text”* as a special data type.

## 1.3 Outline and main contributions of the thesis

The thesis at hand contributes to the state-of-the-art in the following respects:

- The presented work is centered around a framework for document analysis that is based on the assumption that most analysis tasks do not require a full text understanding. Instead, one or several semantic aspects of the text (called quasi-semantic properties) can be identified that are relevant for answering the analysis task. Chapter 2 provides a theoretic discussion of the advantages and disadvantages of working with such a framework. Furthermore, properties of quasi-semantic measures are identified. The four application examples that are presented in the rest of the thesis exemplify document analysis processes that are based on the suggested framework and show its applicability and advantages in real-world scenarios.

- A special focus is put on the usage of visualization within this process. The application chapters show different examples where and how visualization techniques can support the analysis. This includes the development of a novel visualization technique that permits to analyze documents in detail with respect to a specific property (chapter 4), the application of visual analysis to understand the functional principles of the used algorithms and measures better (chapters 4, 6, and 7), and finally the evaluation of the automatic algorithms in order to identify improvement potential (chapter 7). One of the advantages that are gained by working with the framework are the comprehensibility of the measures that are used. Chapters 5 and 7 present visual interfaces that are optimized with respect to providing this transparency to the user in the analysis step.

- Another big challenge when working with the framework is to find good approximations for the quasi-semantic properties that are necessary to solve a certain analysis task. Chapter 5 exemplifies the feature engineering process in the context of readability analysis. In chapter 6 a novel technique for term extraction is proposed that focuses on extracting terms that discriminate several document collections from each other. Other chapters, such as the Literature Analysis in chapter 4 and Opinion Analysis in chapter 7 make use of existing measures that are proposed in related work and evaluate and improve the given techniques.

***What this thesis is NOT about:*** Some of the above mentioned challenges that document analysis comes with are *not* treated in this thesis. This includes processing audio-signals of spoken text or dealing with colloquial language. Furthermore, document analysis questions that go beyond the semantic level of the text, dealing with the goals and intentions of the speaker, respectively the effect that the text has on the reader, are not considered in the application examples.

***Outline:*** Chapter 2 introduces and defines the above mentioned quasi-semantic properties, the research framework, and the resulting research questions. Furthermore, the consequences of working with such a framework are discussed and the usage of visualization techniques is motivated. Chapter 3 presents related work in the area. Chapters 4 to 7 present concrete examples for document analysis tasks with quasi-semantic properties. In some application scenarios the focus is set on the approximation of quasi-semantic properties, whereas others are centered around the analysis and evaluation step, using measures that are proposed in related work. Visualizations are employed in all application chapters, illustrating their value in different steps of the process. Note that the application chapters follow a common structure: First, the research and application context is defined. Next, the quasi-semantic measures are introduced, and novel techniques are presented and evaluated. In each chapter, one or several case-studies demonstrate the applicability of the proposed approaches in real-world scenarios. The application chapters conclude with a summary, discussion, and future work. Chapter 8 summarizes the thesis by placing the work into a broader context and discussing visions for the future.

***Parts of the thesis were published in:***

1. Daniel A. Keim, Daniela Oelke: Literature Fingerprinting: A New Method for Visual Literary Analysis, *IEEE Symposium on Visual Analytics Science and Technology (VAST 2007)*, 2007.

2. Daniela Oelke, Peter Bak, Daniel A. Keim, Mark Last, Guy Danon: Visual evaluation of text features for document summarization and analysis, *IEEE Symposium on Visual Analytics Science and Technology (VAST 2008)*, 2008.

3. Daniel A. Keim, Florian Mansmann, Daniela Oelke, Hartmut Ziegler: Visual Analytics: Combining Automated Discovery with Interactive Visualizations, *Discovery Science*, 2008.

4. Daniela Oelke, Ming Hao, Christian Rohrdantz, Daniel A. Keim, Umeshwar Dayal, Lars-Erik Haug, Halldór Janetzko: Visual Opinion Analysis of Customer Feedback Data, *IEEE Symposium on Visual Analytics Science and Technology (VAST 2009)*, 2009.

5. Daniel A. Keim, Daniela Oelke, Christian Rohrdantz: Analyzing document collections via context-aware term extraction, *14th International Conference on Applications of Natural Language to Information Systems (NLDB 2009)*, 2009.

6. Daniel A. Keim, Miloš Krstajić, Peter Bak, Daniela Oelke, Florian Mansmann: Methods for interactive exploration of large-scale news streams, *Proceedings NATO Advanced Research Workshop on Web Intelligence and Security*, 2009.

7. Miloš Krstajić, Peter Bak, Daniela Oelke, Martin Atkinson, William Ribarsky, Daniel A. Keim: Applied visual exploration on real-time news feeds using polarity and geo-spatial analysis, *6th International Conference on Web Information Systems and Technologies (WEBIST 2010)*, 2010.

8. Daniela Oelke, David Spretke, Andreas Stoffel, Daniel A. Keim: Visual Readability Analysis: How to make your writings easier to read, *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST 2010)*, 2010 (to appear).

# 2

# Bridging the Gap: Towards Answering Quasi-semantic Questions

## Contents

This chapter introduces and discusses the research framework that the work in the thesis is based upon (section 2.1). The second part of the chapter is dedicated to quasi-semantic properties that are the central concept in the framework. Section 2.2 explains what quasi-semantic properties and the corresponding quasi-semantic questions are and illuminates their characteristics.

The contribution of this chapter is an in-depth theoretical discussion of the introduced framework and the concept of quasi-semantic properties. This theoretical knowledge is vital for designing effective algorithms and visualizations. The following application chapters exemplify working with the framework and show how the identified challenges can be met in practice.

## 2.1 Research framework

After a definition of terms, section 2.1.2 details the central ideas of the framework. This is followed by a fictive example of what a complete document analysis process that is based on the framework might look like (section 2.1.3). Next, research challenges are identified that result from working with the framework (section 2.1.4). Finally, section 2.1.5 discusses the role of visualization within the process.

### 2.1.1 Definition of terms

Central to our document analysis process is the notion of quasi-semantic properties. As mentioned above, most document analysis questions refer to a specific semantic aspect of a document. This aspect is what we call a quasi-semantic property of a document.
Why do we call those properties *"quasi-semantic"*? As explained in section 1.2.1 the linguistic definition of "semantic" is centered around the meaning of words and sentences. Understanding the meaning of the text is necessary for many of the analysis tasks that were introduced in section 1.1. But additionally, those document analysis questions often refer to other aspects of a text such as the effect that it has on the reader or its readability. This is not covered by the usual linguistic definition of the term "semantic". To account for this, we call those properties "quasi-semantic".
In the following the term quasi-semantic property and some related terms are defined:

- **Quasi-semantic Property (QSP)**
→ *The semantic aspect that we want to measure (the concept, the ideal).*
In practice, it is important to specify this as narrow and concrete as possible to restrict the considered domain as far as possible.

- **Quasi-semantic Question (QSQ)**
→ *An analysis question that focuses on a quasi-semantic property.*
Thus, this is very closely related to the quasi-semantic property itself. There are several reasons why we additionally require formulating a quasi-semantic question. First, many semantic aspects of a text cannot be unambiguously formulated in a single noun phrase. Consider for example the quasi-semantic properties *consistency* or *quality*. Depending on the application scenario, those terms will refer to a range of different aspects of a text. Formulating quasi-semantic questions helps to concretize what the quasi-semantic property is referring to. A quasi-semantic question serves as a more detailed description of a quasi-semantic property. Besides this, those kind of questions can be formulated from a user perspective. Thus, the quasi-semantic question helps to bridge the gap between the user's perspective on the task and the algorithmic view on it.

- **Quasi-semantic Measure (QSM)**
→ *An approximation of the quasi-semantic property as we measure it.*
While the term "quasi-semantic *property*" refers to an ideal, a quasi-semantic *measure* is what an algorithm effectively extracts (technical view).

- **Text Feature (TF)**
→ *Any feature of a text (e.g. statistical, syntactical, structural or quasi-semantic).*
We consider the term "text feature" as a hypernym for all kinds of features. This includes features without semantic meaning but also quasi-semantic measures. Because the features without semantic meaning are often statistical ones and therefore easier to measure, we also call them "simple or low-level text features". However, counter-examples of text features without semantic meaning, that are quite difficult

to measure, exist (e.g., features that require the detection of structural elements). Furthermore, it should be noted that every “simple text feature” could be considered a quasi-semantic measure in another application scenario in which this is already the approximation for a quasi-semantic property. The distinction between “simple text features” and “quasi-semantic measures” therefore can only be made with respect to a specific application context.

- **Analysis Question**
  → *The question that has to be answered to solve the user’s analysis task.*
  This does not necessarily have to be formulated in the form of a question but could also be a description of an analysis task. To solve the task, the related quasi-semantic measures are used. Often there is a close relation to standard analysis tasks such as classification, clustering, network analysis etc. What is special about those analysis tasks is that they are centered around at least one quasi-semantic property.

Example:
Let us assume that a teacher would like to organize the books in the school library according to the age of the target group that they are suitable for. In this scenario the quasi-semantic question would be: *“For which age is this book suitable for?”*, the quasi-semantic property would be *age suitability*, and a quasi-semantic measure would be a mapping of text features and other quasi-semantic measures that is able to estimate the age suitability of the book and produces a result value that specifies the age (or an age range) that the book is suitable for. This measure could be made up of several text features such as readability, the complexity of the topic, the amount of violence etc. Finally, an analysis question that uses this quasi-semantic measure could be to provide an overview of the library’s book collection with respect to the age suitability or to search for authors that wrote books for a wide range of age groups.

### 2.1.2 The research framework

Figure 2.1 shows a visual depiction of our framework. It consists of three main steps: the input or preparation step, the approximation of quasi-semantic properties with quasi-semantic measures, and the quasi-semantic analysis. In every step of the process the user may be involved. All three steps can be evaluated to ensure a high quality of the process.

In the *input or preparation step* all relevant material is provided and the real-world document analysis task is specified in a way that it can be processed within the framework. This implies the transformation of the task into analysis questions and the identification of the related quasi-semantic properties and quasi-semantic questions (see section 2.1.1 for a definition of terms). Recall that our basic assumption is that most analysis tasks do not require a full text understanding but refer to certain semantic aspects of the text that we call quasi-semantic properties (see section 2.2 for a formal definition and examples).

Next, the *approximation of the quasi-semantic properties with quasi-semantic measures* follows. In this step (combinations of) text features are chosen that approximate the necessary quasi-semantic properties and in this way make them measurable. Thereby, text features may be low-level features (with no semantic meaning) or other quasi-semantic

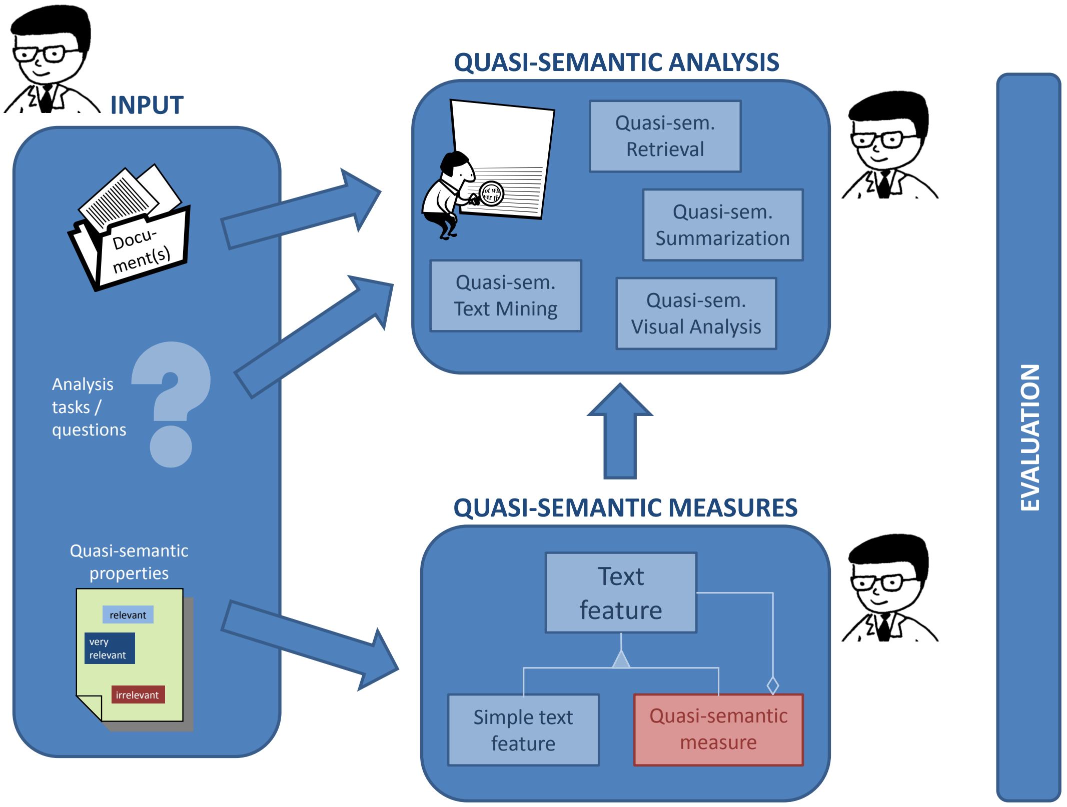


*Figure 2.1: Research framework*

measures. The user may be included in the process of the selection and combination of text features to get a high quality approximation of the quasi-semantic properties.

Finally, the *quasi-semantic analysis* can be performed. Depending on the kind of analysis task, the respective analysis method has to be chosen (e.g., retrieval, summarization, text mining, or visual analysis). We call this *quasi-semantic* analysis because the analysis is based on the extracted quasi-semantic properties. This is done by analyzing the document(s) automatically with respect to the given properties and analysis task and / or representing the document(s) in a way that the user can grasp and interpret the relevant properties and analysis results quickly. The latter is especially important if the machine would not be able to derive the solution automatically, e.g. because the interpretation requires knowledge of the world or the user is not interested in the properties that are *in* the data but rather in the effect that they cause (see also the examples that are given in section 2.2.2).

The framework is based on the assumption that complex features can be approximated with (a combination of) low-level features. This idea is not new but is a standard practice in many fields of data mining and especially in 3D retrieval. However, it is not so common in the area of document analysis so far. One goal of the thesis is to show the advantages and challenges that consequently applying the idea to the field of document analysis comes with. Furthermore, from the perspective of visual document analysis, it is especially important to build measures that are understandable by the user (which can be ensured by using quasi-semantic measures instead of low-level features in the approximation). This allows us to include the user in the process, wherever solving a problem fully computationally is not feasible up to date. Because of the complexity and flexibility of natural language, incorporating the user to bridge the gap between the automatic methods and the real-world scenario is often necessary.

The above introduced framework comes with the following advantages:

1. **Restriction of the domain**
   By identifying the semantic aspect(s) that the analysis question refers to, the domain that we are working on is restricted. As stated above, one of the biggest challenges in document analysis is to cope with the high complexity and the inherent ambiguity of natural language. Restricting the domain results in a decrease of possibilities and thus can greatly facilitate the task. The fact that the quasi-semantic measures can be a combination of other quasi-semantic measures allows to further divide the task until it is computationally solvable.

2. **High transparency and good integration of the user**
   Whenever possible, quasi-semantic properties are approximated with other (existing) quasi-semantic measures instead of low-level features. Because the employed quasi-semantic measures are approximating a semantic aspect of the text themselves, they are easy to understand for the user. Thereby, a high transparency of the automatic analysis process is achieved. This comes with many advantages: First, the analyst does not have to trust the algorithm blindly but is able to follow and control the automatic analysis process. Second, this allows us to integrate the user in the analysis step which is especially important if the document analysis questions require some interpretation or the usage of knowledge of the world. Besides this, integrating the user is also important for finding a good approximation of a quasi-semantic property. It is therefore indispensable to provide transparency in the process in any case in which we want the user to bridge the gap between the capabilities of the machine and the requirements of the real-world tasks.

3. **High reusability**
   The highly modular structure of the framework makes it easy to include existing approaches. Quasi-semantic properties that were developed for another task (or related work from other researchers) can easily be incorporated. Furthermore, an application of a quasi-semantic measure often results in a structured form which allows using analysis techniques that have been developed for structured data. Caution should be exercised, however, when the quasi-semantic measures are to be used in a completely different application scenario or with a different text genre. In this case, the measure does not necessarily capture the same semantic aspect (see chapter 7, for example).

4. **Computability**
   All of the above mentioned advantages result in being able to measure or at least to approximate even complex semantic aspects of a text that otherwise would not be measurable. Moreover, thanks to the fact that most measures are based on statistical characteristics this can be done efficiently and thus, it becomes possible to analyze even large document collections with respect to a specific quasi-semantic property of the text.

### 2.1.3 Example for a complete, fictive analysis process

Chapters 4 to 7 of the thesis present real-world examples for working with the framework. However, most of the time one or several aspects of the framework are detailed instead of focusing on the complete process. In this section, an example for a complete, fictive analysis process that is based on the framework is given.

Web 2.0 enables Internet users to actively participate in creating content. Think about web portals that collect information about a specific topic. Because everybody can contribute information, not every article will be of high quality. Ideally, every article should be reviewed by an expert. However, this is time-consuming and human resources are expensive.

Let us imagine the following fictive scenario to exemplify the document analysis process that is proposed above: The providers of a Wikipedia-like portal[1] decide to support their voluntary reviewers with a system that permits to identify articles that do not meet the Wikipedia standard quickly. Let us further assume that we are asked to help them with the task.

In this case, our first step would be to enquire detailed information about the analysis task. During the conversation, it becomes clear that the Wikipedia standard includes the following properties: (1) The article must not contain subjective statements. (2) There is a special structure that is required. Each article must start with a generally understandable abstract and should be clearly structured. (3) The text should be written in a way that is easy to understand and follows the rules of written English.

Together with the provider, we would identify that the quasi-semantic question in this case is *"How good is the quality of an article?"* and therefore, the quasi-semantic property could be called *article quality*. Furthermore, we can record that this quasi-semantic property is defined by three other quasi-semantic properties, namely *subjectivity*, *wiki-structure*, and *writing style*.

Next, those quasi-semantic properties have to be approximated by a quasi-semantic measure. Exemplarily, the search for a measure that approximates the quasi-semantic property *subjectivity* will be described. We decide to employ an algorithm for subjectivity analysis that was proposed by Rilloff and Wiebe in 2003 [112]. The approach works as follows: First, two high precision classifiers are used to identify a set of definitively subjective and objective sentences. The first classifier identifies sentences that are clearly subjective and the second one searches the remaining sentences for clearly objective statements. Both classifiers are based on a set of so-called subjective clues such as specific lexical items, single words, and n-grams. If a sentence contains at least two strong subjective clues, it is classified as subjective. Sentences that contain none or at most one weak subjective clue are considered as being objective. As only strong and obvious subjective clues are used (that are domain-independent), the precision of the classifiers is high whereas their recall is low. In the next step, syntactic templates are used to extract patterns that might be subjective clues as well. Using the previously extracted sentences as a benchmark, their discrimination power can be assessed by computing a confidence value. Phrases that show a high discrimination power are applied to the remaining, so far unclassified sentences and are used to improve the high precision classifiers.

Theoretically, the algorithm could be directly applied to our data set. However, in the paper we read that the algorithm was developed for and tested on customer reviews

[1] Our fictive portal is similar to www.wikipedia.com

only. Our wiki-articles are a different application domain. Many text analysis algorithms cannot be transferred directly to another domain. We therefore have to evaluate the results and the way the algorithm works carefully. How many iterations are necessary when the algorithm is applied to wiki-articles? And how do we have to choose the confidence thresholds in the pattern selection process to get reliable results?
We decide to employ a visualization technique to learn about the process. Using a pixel-oriented technique, a subset of articles is visualized by displaying every word as a single pixel. Words that belong to the same sentence are visually grouped to highlight sentence borders. Next, positions that contain a subjective clue are highlighted. Thereby, different colors may be used to distinguish the different subjective clues or to differentiate between the different iterations of the process. This gives us an idea how well those predefined, manually selected strong subjective clues work in our application context. For example, it might be that some subjective clues can only be found in specific categories or that some categories do not have subjective sentences at all according to our algorithm. Both cases would suggest that there are categories that are different from the rest with respect to expressing subjectiveness and that need some special consideration. Furthermore, it might be interesting to see how subjective sentences are typically distributed within an article. Finally, the development of the extraction process from one iteration to the next might help to identify an appropriate stop criterion.

Similarly, quasi-semantic measures for the two other text properties have to be identified. To analyze the *writing style* of an article, simple text features such as number of spelling errors, grammar checks, or readability measures might be included. Furthermore, by comparing different articles to each other, commonalities in the writing style as well as outliers might be identified. Regarding the quasi-semantic property *wiki-structure*, measuring the average length of a passage might help or searching for key terms that hint at a well-structured composition of the article. Supposedly, the provider would also be able to tell us what different sections an article might or should contain. Of course, the above mentioned measures are just ideas that would have to be assessed for their usefulness.

As soon as we are able to approximate each quasi-semantic property, we can develop a tool that helps the analyst to identify articles that need some polishing quickly. Combining the three measures into a single one is not favorable in this case, as the three cases need to be treated separately when revising the article. We therefore suggest the provider to develop an application that visualizes the outcome of the algorithm in detail. In an overview, the whole corpus is displayed and each article is visualized as a thumbnail that contains information about the range of observed values for each of the quasi-semantic properties. This permits the analyst to spot documents that might be interesting to look at quickly. Single documents can be zoomed into interactively. In the detail level, the values are shown separately for each sentence and the full text is displayed as well, allowing the analyst to revise the text directly if necessary.

As time goes by, our analyst might get the impression that bad values in the category *writing style* also point to subjectiveness, even in cases in which the *subjectivity* value is quite low. A reason for this might be that our *writing style* measures also take the distribution of personal pronouns into account and that a frequent usage of the pronouns “I” and “we” points to subjectivity. This assumption could be evaluated and, if it turns out to be true, used to improve the *subjectivity* measure. Furthermore, we could use the ground-truth data that results from the work of the analyst to further improve the other measures by searching for correlations with different simple text features respectively combinations of text features.

### 2.1.4 Challenges and research questions

The above example shows some of the challenges that our document analysis process comes with. First, it shows that the right choice of algorithms and measures is not only dependent on the application task but also on the specific domain and the text genre that the techniques are used for. Furthermore, some quasi-semantic properties are dependent on the personality or the personal background of the reader. (Imagine the task of browsing a document collection where the property *interestingness* means something different for each person).

This calls for a deep understanding of the applied algorithms and an efficient and transparent feature engineering process. One of the research questions is therefore how we can find good approximations for a quasi-semantic property. This is even aggravated if no ground-truth data exists, like in the example scenario, which is true for many document analysis tasks. Furthermore, we have to be able to find out if a specific quasi-semantic measure is domain-dependent and where applicable, be able to adapt it accordingly to a new domain or text genre with as few effort as possible.

Some semantic aspects are too complex to find good computational approximations. And even if we do have a good approximation, often, there still exists a gap between the computational efforts and the analysis goals. The analysis process therefore has to be designed in a way that the user can be incorporated to bridge the semantic gap. The necessary transparency of the features are often an inherent property of our quasi-semantic measures (see section 2.1.2). However, if we want the user to be able to do the final interpretation of the results, it is essential that we design the user interface in a way that he is able to effectively evaluate the outcome of the algorithm. Another research question is therefore, how an appropriate interface has to look like.

Next, we need to be able to assess how well a quasi-semantic measure is able to approximate a quasi-semantic property. This is necessary to find out if an approximation does meet our requirements. Furthermore, it is also important to choose an appropriate analysis technique. Knowledge about the quality of the process helps to design algorithms and visualization techniques in a way that they are noise-tolerant enough respectively allow the analyst to control the process. This means that research has to be done with respect to the evaluation of a quasi-semantic measure, but also the design of noise tolerant analysis methods and visualizations.

In some cases, it will be possible to use standard feature engineering and analysis methods. But the above examples show that document analysis tasks often confront us with special challenges. However, the final goal will always be to reuse as much as possible and generalize wherever possible. It certainly would not be efficient to start all over again for every new application scenario. Thus, another research question could be formulated as identifying the potential of generalization for the different steps of the process.

### 2.1.5 The role of visualization

Due to the large complexity and flexibility of natural language, fully automatic processing of documents is not possible for every task. Understanding a text properly often requires knowledge of the world and interpretation. Furthermore, in many analysis tasks creativity and the ability to detect previously unknown patterns is needed. On the other hand, to process large amounts of text, the computational speed and the storage capabilities of modern computers are indispensable. A tight cooperation between the human and the machine is therefore mandatory to analyze large textual datasets.

In the past, visualization has proved as a very powerful means of integrating the human into an automatic process. The human visual system is very powerful, being able to grasp more than a million measures immediately. Low level properties of a visualization (such as orientation, color, texture, and movement patterns) can be perceived at an instance allowing to process millions of measurements at the same time. Furthermore, it is known that humans are very proficient in detecting patterns in a visual scene. When looking at a picture, automatically regions and simple patterns are identified. In contrast to a computer, the human does not need an explicit description of what those patterns look like. This enables the user to find previously unknown structures in the data. [147]

The tight combination of visualization techniques and automatic algorithms to enable an effective and efficient collaboration between the human and the machine in the analysis process is called *Visual Analytics*. The term Visual Analytics thereby denotes the whole *"interactive process that involves collecting information, data preprocessing, knowledge representation, interaction, and decision making"* [72]. It is important to note that this process is not a one-way road but an iterative process with feedback loops between the different steps of the pipeline. The human input triggers and guides the next steps of the automatic analysis. Ideally, at the end of the process gained knowledge can be used to improve the algorithms and may eventually even lead into a fully automatic analysis of the data.

In our case, an effective collaboration between the human and the machine is especially important in the following two steps of the process: To find appropriate measures to approximate a quasi-semantic property and to analyze documents with respect to specific quasi-semantic properties. In the first case, the challenge is to select appropriate features and find the combination that best approximates the desired quasi-semantic property. Often, available features and algorithms are like a "black box" for the user (and even for the expert). By means of visualization, the feature space can be made perceivable or the operation mode of an algorithm is made transparent. This allows the user to take meaningful steps into steering and controlling the feature engineering process. (See [117, 116] for examples of successful visual feature engineering and the detection of optimal parameter settings by means of visualization.)
In the analysis step, it is the human's ability to interpret the results (using his or her background knowledge) that is especially important. There are many situations in which an automatic annotation with respect to a quasi-semantic property is possible, but the final sensemaking of the results is not. Visualization techniques allow the user to derive insight from the data, come up with new hypotheses, and draw conclusions.

One of the goals of this thesis is to investigate how visualization techniques can help in the different steps of the document analysis process. Several novel visualization techniques are introduced and reviewed with respect to where in the process they are profitable. The different application examples demonstrate the usefulness of the techniques. Furthermore, it is shown that in some cases it is possible to use standard visualization techniques while in other cases techniques are necessary that meet the special requirements that document analysis comes with.

## 2.2 Quasi-semantic questions and properties

Quasi-semantic properties are a central element of the framework that is presented in section 2.1.2. The underlying assumption is that for most questions it is not necessary to understand the text fully in a way that we as humans do. This is an essential observation, because it enables algorithms to focus only on the relevant semantic aspects of the text which can significantly reduce the complexity. Farghaly points out in [34], page 6, that *"Restricting the domain of an NLP application usually results in a dramatic improvement in the accuracy, coverage, and overall performance of that application."* He arguments that the reason for this is that *"The amount of ambiguity is reduced and real world knowledge that needs to be incorporated in the system becomes manageable."*

In this section, the characteristics of quasi-semantic properties are theoretically discussed. Knowing about these characteristics has an important impact on the design and implementation of a system. First, for some of the example scenarios in section 1.1, the relevant analysis questions together with the corresponding quasi-semantic properties and questions are identified (section 2.2.1). This provides a rich source of examples that cover the different characteristics of quasi-semantic properties. These characteristics are listed and discussed in the following section 2.2.2. Afterwards a more formal definition of quasi-semantic measures is given (section 2.2.3) and their network character is illustrated (section 2.2.4).

### 2.2.1 Quasi-semantic questions and properties in the example scenarios

In section 1.1 several example scenarios were introduced. In the following, for some of these scenarios the corresponding quasi-semantic properties and questions as well as the analysis tasks are identified[2]. The examples will then be used in the following section to discuss some of the characteristics of quasi-semantic properties.

#### QSQs and QSPs for the example scenarios of companies:

<u>Analysis of Customer Feedback</u>

*Quasi-semantic properties:* Quasi-semantic questions

1. *Attribute:* What attributes of the product do customers frequently comment on?
2. *Sentiment:* What sentiments are expressed in the text?
3. *Opinion on attribute:* What opinions do the customers express on a specific attribute?
4. *Strength of opinion:* How strong do they like or dislike an attribute?

[2] Please refer to section 1.1 for a detailed explanation of the scenarios. The corresponding QSPs, QSQs, and analysis tasks for scenarios that are not given here can be found in appendix A.1.

Related analysis questions:

- Summary of the customers' opinions on a specific product.
- What are the most severe problems of a product according to the customers' opinion?
- Are there subgroups of people with similar opinions?

Finding out the current marked buzz

*Quasi-semantic properties:* Quasi-semantic questions

1. *Topic / Concept*: What do others say about me / my competitors?
2. *Rumor*: How likely is it that the message is true?
3. *Prominence*: How prominent is the message? (Does a single blog post it that only very few people read or is it among the top news of the day?)
4. *Sentiment*: Is the statement in total positive or negative?
5. *Impact*: What impact does it have? (How do people react to it?)

*Related analysis questions:*

- Summary of the main discussions.
- How many other sources report about the same issue?
- Does the message spread and how fast does it spread?
- What are currently the top ten rumors?
- How long do rumors in average stay alive?

### QSQs and QSPs for the example scenarios of researchers:

Browsing through large paper collections

*Quasi-semantic properties:* Quasi-semantic questions

1. *Concepts and their central phrases:* What is the paper about?
2. *Technical Depth:* Is the presented approach of technical depth?
3. *Interestingness:* Is it interesting for me?

*Related analysis questions:*

- Give me all documents of the collection that are related to what I do (e.g. that are similar to a given paper).
- Summarize the content of the paper in a few sentences.
- Represent the whole paper collection in an expressive way that facilitates browsing.
- Label all documents with my personal interestingness score.

Assessment of papers and proposals

*Quasi-semantic properties:* Quasi-semantic questions

1. *Concepts:* What do they propose?
2. *Interestingness / Utility:* Is the approach useful or interesting for the community?
3. *Novelty:* Is the proposed approach novel?
4. *Non-Obviousness:* Is the proposed approach non-obvious?
5. *Readability:* Is the text well readable?
6. *Structure:* How is the document structured?
7. *Information Density:* Do they repeat themselves over and over again or is it written compactly?
8. *Technical Soundness:* Is the proposed approach technically sound?

*Related analysis questions:*

- Given the concepts, find related approaches that already have been published.
- Find out what the writing conventions of the specific community are and compare them to the structure of the given document.
- Represent the results of the above QSPs in a way that the strengths and weaknesses of the proposal / paper can easily be seen.
- Find the appropriate class for the proposed approach in a given taxonomy of research proposals.

### QSQs and QSPs for the example scenarios of literary scholars:

Analysis of Novels

*Quasi-semantic properties:* Quasi-semantic questions

1. *Figure:* Which different figures exist?
2. *Popularity:* How popular is the figure?
3. *Problem of society:* What problems of societies are thematized or hinted at?
4. *Central message:* What is the general message that the authors wants to convey?
5. *Message hints:* Where are messages hidden between the lines?

*Related analysis questions:*

- In which part of the novel is which figure most prominent?
- Which figures have a relationship to each other?
- How does the network of figures look like?
- How similar are the thematized problems of societies to what is discussed in our society today?
- Which messages are only hinted at and which are clearly expressed?

Literary Quality

*Quasi-semantic properties:* Quasi-semantic questions

1. *Self-activity:* Where is the reader encouraged to self-activity?
2. *Persuasiveness:* How persuasively is the book written?
3. *Understandability:* Does the reader understand what is written?
4. *Issue Stream:* Which issues are discussed in the book and which parts of it talk about the specific issue?

*Related analysis questions:*

- Has every issue that is being raised in the book been resolved in the end?
- Does the book have an effect on the reader?

QSQs and QSPs for the example scenarios of Internet users:

Assessing the quality of Internet content

*Quasi-semantic properties:* Quasi-semantic questions

1. *Publisher:* Who published the page?
2. *Readability, Writing Style, Consistency:* How well has the text been written?
3. *Degree of consent:* Is the content in concert with the information that other web pages publish?
4. *Completeness:* Has the topic been discussed completely or do other sources, that discuss the topic, frequently mention an issue that has not been raised here?

*Related analysis questions:*

- How probable is it that the content of the page is trustworthy?
- Which additional aspects are raised in other web pages that discuss the topic?
- Does the web page represent an outlier opinion or the common consent on the topic?

### 2.2.2 Some notes on the nature of quasi-semantic properties and the related analysis questions

Each of the above mentioned scenarios could be discussed in detail. Yet, this is beyond the scope of this thesis. What is more interesting is the question what we can learn about the nature of quasi-semantic properties in general from the above mentioned scenarios. In the following, some notes on general observations about analysis questions and quasi-semantic properties are given.

1. *Properties that go beyond semantics*
   Some of the above mentioned text properties are special in the sense that they do not measure a property that is *in* the text, but rather an "effect" that is caused by the text. This is true, for example, for the properties *impact*, *self-activity*, and *induced emotions*. Thus, they can only be called a quasi-semantic property, if the term "semantic" is understood broadly. This is reasonable, because usually there is some internal evidence in the text that can be used to define a measure that approximates the effect that is caused. In the scenario of finding out the current market buzz to which the quasi-semantic property *impact* belongs, this could be measuring the number of people who called the web page or the number of web pages that link to the respective page. The number of web pages that pick up the discussed topic could hint at the impact of the page, too. Ultimately, looking for some concrete evidence is also what some human analyst would have to do when solving the task "manually".

2. *Quasi-semantic properties and personality*
   There are quasi-semantic properties that are highly dependent on the personality or the personal background of the reader. For example, the property *interestingness* in the scenario of browsing large document collections is correlated to what I am working on and what my personal preferences are. For the properties *non-obviousness* and *persuasiveness*, this is at least partly true since my personal background knowledge and my personal convictions will play a role in the decision whether something is convincing or obvious for me. On the other hand, those two properties also show that there is often a non-personal part that can be approximated without taking the personality of the reader into account. In the case of *persuasiveness*, this includes measuring if there are any contradictions in the argumentation and if the writing style is appropriate. Similarly, for many other properties that can be considered as being well approximable at least a weak influence of personality could be proven as well. Likewise, for some properties knowledge about the writer will improve the results, because it facilitates the correct interpretation of the text. To sum up, there is a continuum for quasi-semantic properties between being highly dependent on the personality and independent of the personality.

3. *Quasi-semantic properties and knowledge of the world*
   When interpreting text, humans constantly make use of their knowledge of the world. This includes lots of background knowledge, but also the knowledge about feelings or experiences in relationships. Some of the quasi-semantic properties require such knowledge of the world. For example, in the scenario of the analysis of novels an examination of the text with respect to problems of society that are hinted at would require such background knowledge. The same is true if the characteristics of a figure in the novel are to be listed, especially if those characteristics are not explicitly mentioned, but can only be inferred from the way that the person acts. When knowledge of the world is required in document analysis tasks, often ontologies are employed. This may help to diminish some of the problems, but others (like the analysis of the figure's characteristics) remain unsolved. This is why considering knowledge of the world has to be mentioned as one of the big challenges for the automatic support of document analysis tasks.

4. *How difficult is it to approximate a quasi-semantic property?*
   There are big differences with respect to the complexity of the above mentioned quasi-semantic properties. As mentioned above, special characteristics such as the necessity of knowledge of the world, asking for the effect that is caused by the text or the influence of the personality of the reader represent special challenges. Besides that, we can distinguish between properties that can be measured directly and others that are approximated with a combination of several other quasi-semantic measures. Furthermore, the existence of ground-truth data eases finding a good approximation for a quasi-semantic property, because this permits to use some of the methods that have been developed for classical feature selection tasks. Unfortunately, in many of the document analysis tasks no ground-truth data is available and creating it would be very cost-intensive or even impossible since no generally agreed on solution exists.

5. *Hierarchy levels the properties are defined on*
   Many quasi-semantic properties can be calculated on different hierarchy levels of the text (e.g. word level, sentence level, paragraphs, document, etc). However, some of them are only defined on a certain hierarchy level. For example, the property vocabulary richness is only defined on a relatively large block of text, because it is impossible to assess the vocabulary richness of a writer from a single sentence. On the other hand, there are properties such as proper names that are only defined on word or phrase level. This may aggravate analysis tasks that have to take different properties into account that are not defined on the same hierarchy levels.

6. *Reduction to standard analysis questions*
   Almost all of the above mentioned analysis questions could be reduced to a "standard" analysis task such as clustering, classification, network analysis, time series analysis, information retrieval, summarization etc. In some cases, this also means that as soon as a measure has been found that approximates the quasi-semantic property, standard data analysis techniques can be applied. However, this is not always the case. One reason for that might be that the different quasi-semantic properties that have to be taken into account are not defined on the same hierarchy level (see note above). Another one could be that the analysis task requires knowledge of the world or interpretation and therefore the analysis has to be done in a way that the user can be incorporated.
   Like in standard data analysis, depending on the specific task either visual or automatic methods can be applied (or a combination of both).

### 2.2.3 Formal definition of quasi-semantic measures

As defined above, a quasi-semantic measure is a computationally measurable approximation of a quasi-semantic property. This quasi-semantic measure is made up of one or several text features, which may be simple features (e.g. sentence length) or quasi-semantic measures themselves. The combination of several text features can either be a mathematical one (e.g. the weighted sum of different features) or a heuristic that makes use of several features. Below a formal definition of a quasi-semantic measure is given.

$$TF_{new}(X, L, D) = f(X, \{TF_i(X, L', D)\}, D)$$

where i $\neq$ new

$TF_i$ = simple text feature | quasi-semantic text feature
X = the textual content (hierarchically structured)
$L, L'$ $\in \{l_1, l_2, \ldots, l_n\}$ - the set of hierarchy levels that the text should be annotated on
$D$ $= \{d_1, d_2, \ldots, d_{n_d}\}$ - the set of additional data (dictionaries etc.)

Hierarchy levels can be, for example, the token level, phrase level (e.g. noun phrases), sentence level, paragraph level, section levels etc. Not every text feature is defined on every level. For example, a measure that estimates the vocabulary richness of a text will not work on single tokens but needs larger samples of text. On the other hand, even for a text feature that can work on every level, the user might not be interested in the values for all levels. This is why the set of levels ($L$) has to be given as an input of the function. Note that the levels of the text feature that is the output of the function do not have to be the same than the ones that the text features it is comprised of work on (which is why $L'$ is needed).
Some features need external data such as dictionaries to work. In the function above, this is taken into account with the parameter $D$. As can be seen, the definition of a quasi-semantic measure is recursive in the way that the text feature(s) on the right hand side of the equation could also be a quasi-semantic measure themselves. Note that the format of the text features may be single values or terms (as in the case of age suitability or trustworthiness), a set or vector of values or terms (e.g., in keyword extraction), or a set of tuples (e.g., function word frequencies).

### 2.2.4 Networks of quasi-semantic measures

Interestingly, some quasi-semantic properties show up over and over again in the above mentioned examples. Those quasi-semantic properties are central for analyzing a document. Consider, for example, the quasi-semantic property *concept*. Understanding where a new concept is introduced and what it is all about is a basic task when reading a document. As stated above, quasi-semantic properties that cannot be measured directly are approximated with a combination of other text features. In general, it is preferable to use quasi-semantic measures in the combination instead of low-level text features. This facilitates understanding the measure for the user. Technically, all quasi-semantic measures can be described with low-level text features only, because of the recursive definition of quasi-semantic measures (see figure 2.1).

Given a concrete analysis scenario, it is possible to draw a network of quasi-semantic measures showing which measures are build upon which others. Figure 2.2 shows a (fictive) network of quasi-semantic measures to exemplify the idea. The graphic is to be read as follows: A → B means that the quasi-semantic measure A uses the quasi-semantic measure B as part of its definition. For example, for measuring how consistently a text is written, measurements of the writing style, certain authorship characteristics (which is closely related to the writing style, therefore building on it itself), and finally the introduced

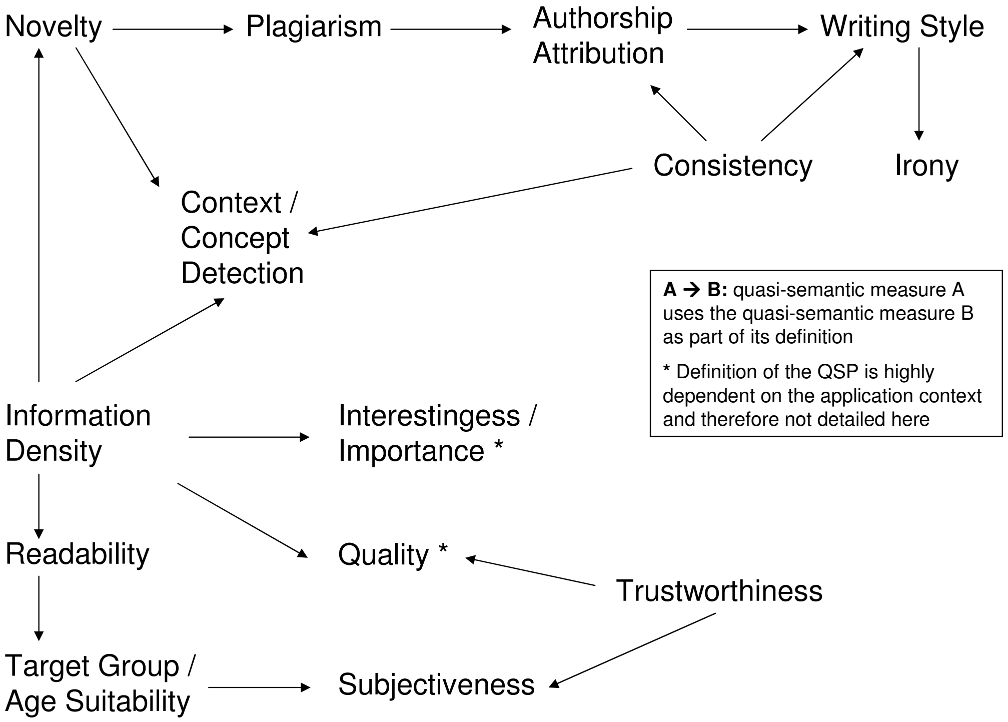

Novelty
Plagiarism
Authorship Attribution
Writing Style
Consistency
Irony
Context / Concept Detection
A → B: quasi-semantic measure A uses the quasi-semantic measure B as part of its definition
* Definition of the QSP is highly dependent on the application context and therefore not detailed here
Information Density
Interestingess / Importance *
Readability
Quality *
Trustworthiness
Target Group / Age Suitability
Subjectiveness

*Figure 2.2: Excerpt from a network of quasi-semantic measures. Central to the framework proposed in section 2.1.2 is the idea of approximating complex quasi-semantic properties with other text features. Ideally, existing quasi-semantic measures are used to increase the transparency of the measure for the user.*

concepts might be taken into account. Note that for some quasi-semantic measures such as *quality* no outgoing arrows were drawn, because what constitutes the *quality* of a text is highly application-dependent (and basically could be everything).

# 3 Related Work in Computational Document Analysis

## Contents

Computational Document Analysis as a research area comprises many different aspects such as structure analysis, information extraction, information retrieval, document clustering and classification, machine translation, summarization etc., but also the more linguistic tasks of analyzing the different linguistic levels of text such as morphology, syntax, semantics and so on.

On the computational side both, fully automatic and semi-automatic approaches exist. Sometimes, the automatic techniques are enhanced with (interactive) visualizations. Besides computational linguists working in the field, more and more machine learning specialists are engaged in document analysis. Depending on the task and the data, linguistic or statistical approaches, or even a combination of both, performs best. Similarly, a wide range of applications can be thought of, some of which were discussed in the previous chapters (see sections 1.1 and 2.2.1).

In the following, related work in the context of computational document analysis is reviewed. Section 3.1 compares different approaches for semantic document analysis. Furthermore, examples for other approaches that are based upon what we call a quasi-semantic measure are given. The next 4 sections focus on related work for the application chapters. Section 3.2 reviews existing techniques for document visualization. Note that papers that present literature analysis properties are directly addressed in chapter 4, because we do not contribute to this area but use existing measures. Section 3.3 discusses related work in the context of readability analysis. Next, techniques for automatic term extraction and the extraction of product attributes are presented (section 3.4). This is followed by an overview of related work for sentiment and opinion analysis in section 3.5. The chapter concludes with a short evaluation of commercial text analysis products in section 3.6.

## 3.1 Semantic document analysis based on quasi-semantic properties

The work that is presented in this thesis is based on the visual analysis framework that is introduced in section 2.1.2. Alternative approaches for semantic document analysis exist. Section 3.1.1 compares the different approaches.

A central concept of the introduced framework are the so-called quasi-semantic measures that approximate a semantic aspect of the text. This is not a commonly used term in literature, but there is related work that reports on measures that could be called quasi-semantic in our terminology. Section 3.1.2 presents some of this work.

Finally, section 3.1.3 comments on the impact of linguistic research on this work.

### 3.1.1 Approaches for semantic document analysis

Four fundamentally different approaches for semantic document analysis exist. Figure 3.1 schematically compares the variants.

Semantics is the branch of linguistic research that aims at capturing the meaning of linguistic expressions in formal structures (called meaning representations). *Computational Semantics* deals with the problem of creating those representations automatically. It is based on the assumption that the semantics can be inferred from the syntactic structure of the sentences. The transformation into the formal representation is done automatically, but requires manually defined knowledge bases and rules that describe how a syntactic parse tree can be translated into the corresponding meaning representation. [68]

The *Semantic Web* approach is similar in that it is also based on ontologies and employs inference and reasoning to defer knowledge from textual data. However, instead of inferring the semantics from the sentences directly, the Semantic Web approach requires that the text is enriched with meta-data that captures the (relevant parts of the) meaning of the text. This implies that the human involvement is not a one-time effort, but must be done separately for every new document that is generated. [5]

*Artificial Intelligence and Text Mining* approaches completely rely on machine-learning techniques. In contrast to the before mentioned techniques, they aim at developing fully automatic methods that allow machines to comprehend textual data. So far, it could not be shown that it is possible to capture the complete semantics of the text with those techniques. However, text mining has been successfully applied to specific aspects and tasks that can be computationally modeled. [92, 37]

*Visual Analytics* is based on a tight cooperation between the human and the machine. Thereby, visualization serves as an interface between the human and the machine. This permits to provide the machine interactively with the necessary knowledge of the world without shifting the major work load to the human. Instead, the goal is to let the machine support the user as good as possible with the time-consuming parts of the task. The interaction between the human and the machine permits to profit from both's strengths and extends the possibilities of semantic document analysis compared to techniques that are solely based on automatic methods or require extensive human involvement. [133, 71]

Note that the four approaches do not only differ with respect to the techniques that they use and the human involvement, but also with respect to their goals. Whereas computational semantics and also some artificial intelligence approaches aim at fully capturing the semantics of a text, the semantic web approach as well as visual document analysis concentrate on the semantic aspects of a document that are necessary to solve a specific application task. Although computational semantics is the only approach whose origin is clearly located in linguistics, the other techniques also incorporate linguistic knowledge where helpful.

### 3.1.2 Quasi-semantic measures in related work

As was discussed above, one of the fundamental differences between the different techniques that analyze the semantics of a document is that some of them try to fully formalize the inherent semantics whereas others, as we do, focus on the specific semantic aspect that is relevant for answering an analysis tasks. In the following, some examples for approaches that model a semantic aspect of a text are reviewed. In our framework process, the proposed techniques could be applied as a quasi-semantic measure.

Mandic and Kerne [88] measure the level of *intimacy* in e-mails. The proposed measure is based on several text features such as the context (i.e. professional vs. casual), emotional tone, capitalization and formality of language. Furthermore, the number of recipients are taken into account. The authors suggest to use visualization techniques to analyze a user's emails with respect to this property.

*Irony* in user-generated web content is detected in [20]. The authors compare eight linguistic patterns (such as heavy punctuation marks, positive interjections, the usage of emoticons etc.) with respect to their expressiveness in analyzing ironic statements.

*Emotions* in web news are measured in [163, 48]. Both techniques employ a sentiment dictionary to determine the degree of expressed emotions such as sadness, joy, anticipation, surprise etc.

Several papers discuss measures for *web information credibility* (e.g., [151, 114]). Depending on the proposed approach, the methods take different text features into account such as the opinions that are expressed in the blog, the appeal of the page, the lengths of the articles, the usage of emoticons, the number of comments of readers on the page etc.

Agarwal et al. [2] identify *influential bloggers*. A blogger is considered as being influential if its posts are recognized by many other blogs (measured as the number of referring posts), if they have a large number of comments, if they present novel ideas (assuming that this is reflected by a low number of references to other blogs), and finally, if the blog posts are rather long (something that the authors interpret as hinting at a certain eloquence).

Moturu et al. [98] analyze how trustworthy web articles are that share advice. Two other high-level properties (in our terminology QSPs) are used, namely quality and credibility, to estimate the amount of *trust* that is associated with the blog by its readers. To measure *quality*, a combination of features is used including external links (showing

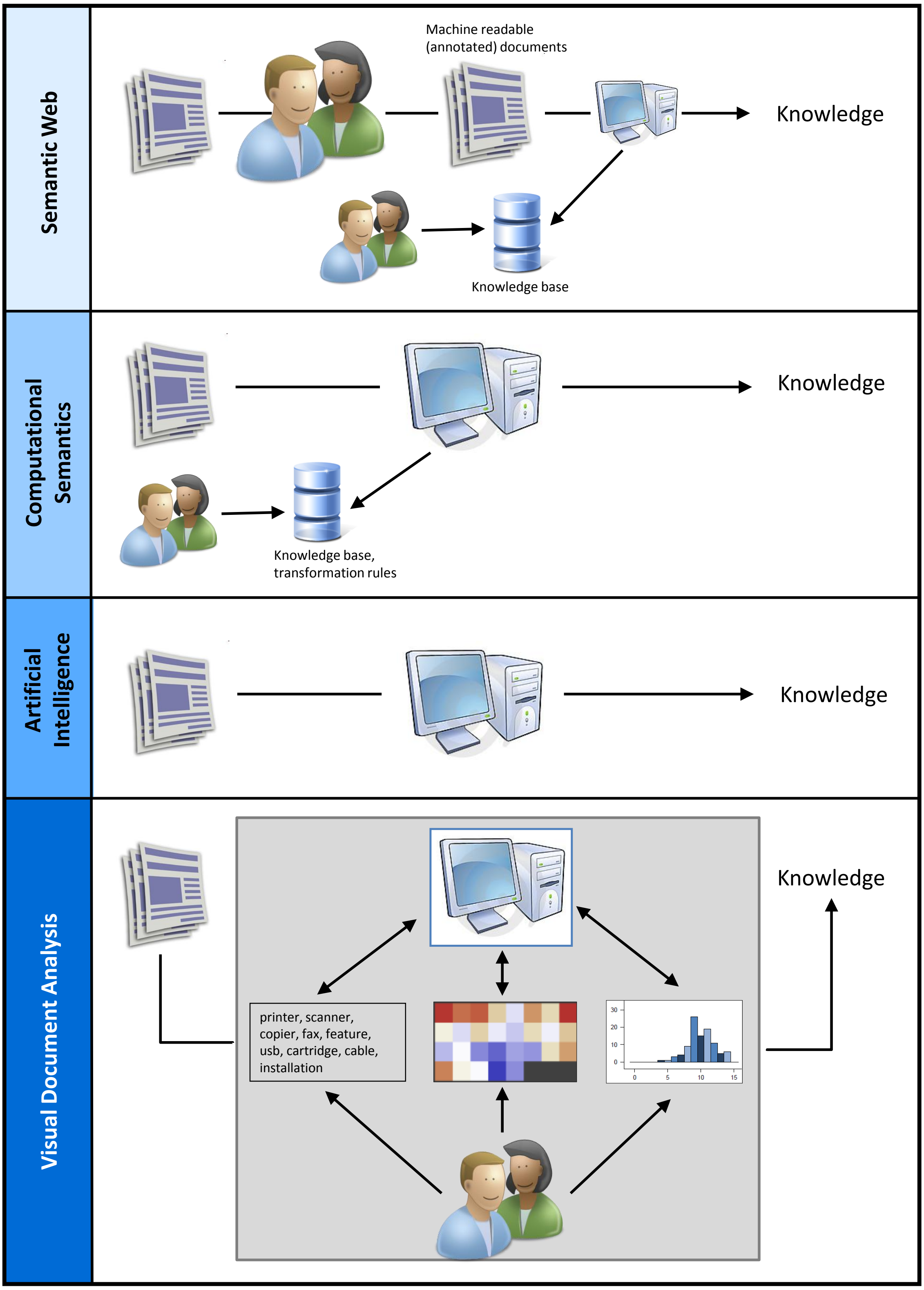


*Figure 3.1: Comparison of different approaches for semantic document analysis with a focus on the involvement of the human in the process.*

that the text is substantiated with external evidence), internal links, and the size of the post (assuming that a longer text corresponds to the willingness of the author to put a larger effort in the analysis). *Credibility* is approximated by the number of friends and the connectedness of the author in a social network (given that this information is available). Furthermore, the number of journal entries and the number of replies to them are taken into account.

### 3.1.3 Linguistic research in document analysis

The term "Semantics" denotes the linguistic branch that studies linguistic meaning. Thereby, a focus is put on the development of a formal representation that encapsulates the semantics of the text. "Computational Semantics" deals with computational approaches to defer the linguistic meaning of a sentence from the meaning of its syntactic constituents ([94], chapter 5). In both cases, an emphasis is put on a high quality and completeness, rather than efficiency. This implies that so far only few applications are based on this technology. Examples include first approaches of question answering systems and verifications of sentences with respect to semantic plausibility. Note that there is a trade-off between being as exact as possible in the representation and providing a flexible and machine readable (and producible) format. Broader document analysis tasks such as *"Which product attributes are mentioned positively / negatively in the customer feedback data"* would require a comprehensive description of the semantics or extensive postprocessing with external dictionaries.

More generally speaking, computational linguistics deals with the processing of all levels of natural language with a computer. Standard tasks include word-sense disambiguation, anaphora resolution, text segmentation, parsing, or part-of-speech tagging for which computational approaches have been and are developed. Besides this, there are more advanced and more application-oriented tasks such a natural language generation, text-to-speech synthesis, the generation of ontologies etc. Among the application contexts which are closely related to linguistic research are information extraction, question answering, machine translation, text summarization, spoken dialogue systems and many more. [94, 68]

Especially the basic technologies can be considered as central components of all kinds of document analysis tasks (both, for linguistically-based and statistically-based processing). The work that is presented in this thesis exploits several linguistic tools such as software for part-of-speech tagging, parsing, morphological base form reduction, and segmentation. Others, such as anaphora resolution techniques for example, would be valuable resources as well to improve the algorithms. However, the current state of development is not far enough to employ them in real-world applications.

## 3.2 Related work for document visualization

One point of focus of this thesis is to investigate how and where document visualization techniques can support the analysis process. In the following, existing document visualization approaches are reviewed and set into context.

### 3.2.1 Detailed document analysis with respect to a text property

In chapter 4 a visualization technique is presented that permits to analyze a document in detail with respect to a text property. Closest related to this *Literature Fingerprinting* technique are the visualizations that were introduced in [11, 57, 30].

Seesoft [11] has been designed for the visual analysis of program code, whereas the intention of TileBars [57] is to provide a compact and meaningful representation of Information Retrieval results. With respect to the objective the authors had in mind, the FeatureLens technique, presented in [30], is probably the one that is closest related to Literature Fingerprinting, being designed to explore interesting text patterns, find meaningful co-occurrences of them, and identify their temporal evolution. None of the three techniques makes use of the structural information of the documents. Yet, [57, 30] are able to view the documents on different resolution levels (without taking the document structure into account). However, in both techniques documents are represented as single line without any wrap. Large documents therefore cannot be displayed in full detail, e.g. on word level without extensive use of scrolling. Consequently, this results in the fact that only features can be visualized for which a meaningful aggregation exists. Both techniques so far only have been applied for (n-grams of) term frequency, a feature for which this property holds true. Finally, Seesoft contrasts with the others by its semantic zoom that finally leads to a view in which the underlying text is directly readable. Most of the other techniques are only able to display the text in a separate window on request.

Beyond that the visualization technique Ink Blots of Abbasi and Chen [1] and the system Compus of Fekete and Dufournaud [35] have to be mentioned in the context of text feature visualization. The visualization technique that is used in Compus is similar to the one used for Literature Fingerprinting. But in contrast to all the previous techniques, Compus visualizes a multitude of features at once in one single visualization by accepting much overplotting. This becomes possible, because Compus is specialized on manually annotated XML documents. Their analysis is based on the structural information that is included in the XML documents and the annotations. Many of these features are binary and do not change their values over large parts of the document. Others are very sparse and concentrated on small sections of the document and can therefore be printed on top of the larger features without losing too much information. However, this still results in pictures that are hard to interpret. This is why the authors suggest to allow selecting a subset of features manually that can be displayed simultaneously without getting too much clutter. The Ink Blot technique also visualizes multiple features at once and accepts overplotting but follows a different visualization strategy. Abbasi and Chen use blots to mark the occurrence of an observed feature directly in the text. With the help of a decision tree, they select the features that are most discriminating for a specific class. The color of the blots signals whether the frequent existence of the feature is characteristic for a specific class or not. So the resulting visualizations can be used to decide whether a new document belongs to a specific class or not or to detect outliers in an otherwise consistent set of documents. Both [1] and [35] cannot cope with features that provide values for each single text unit (at least not without giving up their claim to visualize multiple attributes at once).

### 3.2.2 Visualizations for single documents

Whereas the above mentioned visualizations all focus on showing the development of certain text properties across a document, a couple of techniques exist that discard the positional information. They take a more global view on a document by analyzing word frequencies and/or exploring the relationship between certain entities.

A famous representative in this category are tag clouds (also called text or word clouds). Typically, the most frequent terms are arranged in alphabetical order and the font size of each word is mapped to its frequency in the text like in TagCrowd [119]. The popular tool Wordle [143] abandons the alphabetical ordering and focuses on a tightly packed cloud instead. Thereby, words do not necessarily have to be arranged horizontally (although this can be enforced). Some word clouds additionally use color to encode a specific property of the words such as their part-of-speech class (see word clouds in [144], for example). TextArc [101] is a special example of a word cloud, because the positions of the words within the document are taken into account as well. First, words are lined up in an elliptical shape in their order of appearance. Second, words that occur more often and are spread across the document are drawn into the middle of the ellipse. Interaction techniques permit to display lines that show the original positions of those words within the text. Color and word size are used to encode the frequency of the words.

In the Word Tree [150] visualization additionally the context that the words appear in is shown. After choosing a word or a phrase, all sentences that include this phrase are extracted and arranged in a tree structure that groups sentences with common start phrases together.

In contrast to the above mentioned approaches, [140] additionally permits to explore the connectivity between the words. In the Phrase Net graph each node represents a word and each edge links two words that are connected by a user-specified relation such as "X *at* Y". Arc Diagrams [149] focus on displaying repetition in a document by connecting repeating subsequences with semicircles. Thereby, the thickness of the line encodes the length of the subsequences, whereas the height of the semicircle is determined by the distance between the two phrases.

### 3.2.3 Visualizations for document corpora

In addition to this, a variety of techniques for the visualization of document corpora exists. In contrast to Literature Fingerprinting and our interface for readability analysis, those approaches do not visualize a single document in detail but illustrate the relations of the documents among each other.

WebSOM [82], Galaxies and ThemeScape of IN-SPIRE$^{TM}$ [154], as well as [40] are examples for techniques that are based on dimensionality-reduction methods. The documents are represented in a high-dimensional space (e.g. with TFIDF vectors) and then mapped into two dimensions, resulting in document landscapes or maps.

ThemeRiver [56] is an example for a technique that explores the development of certain topics over time. A river metaphor is used to display the change in the distribution of the topics. Approaches that go along similar lines are Meme-tracking that was introduced in [86] and the news stream visualization in [80]. [38] and [59] also show the development of topics over time but use a line chart to visualize the trends. [59] additionally visualizes co-occurrence networks. Both techniques employ automatic methods to identify terms that show an interesting behavior over time.

Parallel Tag Clouds [24] display each document by a vertically arranged tag cloud. Next, the tag clouds of multiple documents are set beside each other. The comparison of different clouds is visually supported by colored bands. In [113], Rose et al. visualize the emergence and fall of story events using a technique that is similar to the Parallel Tag Clouds but additionally displays some context information.

Thiel et. al [132] visualize topic shift within conference proceedings. The extracted topics and the years are modeled as nodes in a network. A topic node is connected to a year node if the corresponding term appears in the proceedings of the specific year. The resulting graph is layouted with force-directed techniques. Spreading activation methods are used to facilitate the exploration of the network.

Relationships between documents are also displayed in [22, 162]. In this case, the nodes of the network represent papers and the connections are based on citations, building a large graph of scientific literature. Many extensions and variants for co-citation analysis exist.

The history flow visualization [142] represents the change from one version to another in a collaboratively generated document collection such as Wikipedia. Instead of visualizing topic change, metadata such as information about who has edited a certain passage is displayed. Other tools that are mainly based on metadata are Jigsaw [127] and the PatViz system [79]. Jigsaw provides many possibilities to analyze the relationships between documents and entities, e.g. by using graph visualizations and scatterplot views. PatViz is a system for patent analysis. Depending on the analysis task, users can choose between different visual representations such as a map that shows where the patents come from, a depiction of their distribution across a classification system, or network representations. Both tool can be considered as classical visual analytics tools that get their strength from the combination of automatic and visual methods.

In [130], we propose the usage of thumbnails (called Document Cards) that are composed of a mixture of key terms and images to represent a document. In a case study with papers, we could show that in combination with the title and the author names a compact but expressive visual representation is created, allowing to display a whole document corpus on a single screen.

Finally, a couple of techniques for displaying information retrieval results exist. Among them are the TileBars that were already introduced in section 3.2.1. Furthermore, InfoCrystal [125] and MedioVis [49] have to be mentioned. InfoCrystal displays the search results for a query with multiple search terms in an abstracted venn diagram, showing how many documents contain a specific subset of query terms. In contrast to this, MedioVis provides detailed information about the retrieved documents. A core component of the tool is the so called HyperGrid, a table representation that is enhanced with powerful semantic zooming facilities.

## 3.3 Related work for readability analysis

Several well known formulas to measure readability exist. Among the most popular ones are the Flesch-Kincaid Readability Test [75], Flesch Reading Ease [39], SMOG [91], the Coleman-Liau-Index [23], and Gunning Fog [50]. It is common to all these measures that they are solely based on statistical properties of the text, such as word length (either measured as the number of syllables or the number of characters), the number of words in a sentence / paragraph, and the number of easy and hard words. A word is considered as "hard", if it consists of three or more syllables or alternatively, if it is not contained in a list of easy words. The most severe disadvantage of these methods is that the calculated value does not permit to conclude what exactly has to be changed to improve the readability of the text. Nevertheless, those measures are commonly found in commercial tools for readability analysis (see e.g., RocketReader [109], or Readability Studio [110]).

Other approaches take more aspects of readability into account. For example, [58] and [118] consider the syntactic complexity with the help of features like the depth of the parse tree of a sentence or the number of sentences in passive voice. In both papers the vocabulary usage is taken into account with a statistical language model to avoid the need for a vocabulary list, same as [26] and [121] do. The difficult problem of measuring how coherent a text is, is tackled in [12]. Their approach is based on the assumption that the distribution of entities can be used to defer information about the local coherence of the text. Additionally, [106] takes discourse relations into account to measure text coherence and show that they are good predictors of readability (comparing them to several other readability features). However, their method requires that the discourse annotation is given, because so far, it cannot be determined automatically. [21] analyzes if syntactical surface statistics are good predictors for sentence fluency. Their study suggests that these features indeed correlate with fluency, but nevertheless the result indicates that they are not a good predictor for text quality.

In contrast to the above mentioned methods, we do not make assumptions about what features might be good predictors for readability. We prefer to start with a high number of features and let automatic algorithms decide what the best predictors are. Furthermore, our goal is to provide the user with a tool that guides the improvement of the text which results in the special requirement that we need features that are semantically understandable and that the development of the values across the document must be perceivable (to identify the passages that are in need of refinement).

## 3.4 Related work for discriminating and overlap terms

*Term extraction* is one of the central techniques for all kinds of document analysis. Applications that depend on term extraction methods include automatic translation, information retrieval, text mining, keyword lists (e.g. for libraries) etc. Depending on the community and the application area, also other names are used such as *keyword extraction*, *automatic term recognition*, *terminology extraction*, *term acquisition*, and *automatic indexing*. Related to this is *entity extraction* (also called *entity recognition / detection / identification*) that deals with the extraction of events and entities such as person names, location etc. *Information extraction* or *relationship extraction* additionally mines relationships between the entities (e.g. dove is bird). Note that the term *information extraction* is sometimes used as a hypernym for all kinds of techniques that extract structured information from unstructured textual data.

In chapter 6, a novel approach for term extraction is presented. Its special characteristic is that it optimizes the result with respect to the extraction of terms that discriminate several document collections from each other. Furthermore, we also determine terms that discriminate multiple collections from one or several others.

### 3.4.1 Automatic term extraction

Many methods exist that are dedicated to the extraction of terms out of document collections. One reason for this is that what constitutes a *relevant* term is highly application-dependent. In the following, term extraction methods of different communities and application areas are reviewed.

Approaches for keyword extraction often originate from the information retrieval field like e.g. the prominent TFIDF method [124, 115]. An extensive survey on that can be found in [69]. But also in text mining research, keyword extraction methods play a role [36, 90]. Usually, there is a measure that permits to score terms with respect to a document or a document collection and a certain number of top scored terms are then extracted.

Named entity recognition aims at extracting proper names, that have a certain semantic category, in order to construct semantic lexica [111, 25]. Typical examples for such categories are names of persons, companies or locations.

Among the term extraction approaches are some that extract domain specific terms comparing an analysis corpus of a certain domain with a reference corpus. The reference corpus is chosen in a way that it is as broad and universal as possible and can either be a general language corpus [17, 155] or composed of several other domain corpora [141]. Another approach takes a large collection of heterogeneous newspaper articles as a reference corpus [31]. Those approaches are useful for example to support terminology extraction or ontology construction.

Methods for term acquisition are often based on linguistic patterns. TERMINO [84], e.g., is based on generative grammar rules, while LEXTER [16] employs lexico-syntactic patterns to identify relevant terms. A major focus of term acquisition techniques is to build novel term databases for a specific task or to keep them up to date.

Methods for labeling are mainly used for visualization tasks. Usually, they extract very few terms that describe the documents that constitute a certain area of a visualization. In the ThemeScape$^{TM}$ visualization [153], a common TFIDF approach is used for the labeling of the distinct document clusters. A similar labeling approach is done in the WEBSOM visualization [65], where the relative frequencies of terms in the different nodes of the self-organizing map are compared [83, 9].

Our method that is introduced in chapter 6 is similar to the approaches that do domain specific term extraction, because we also compare the scores of a certain term for different domains/classes. Yet, in contrast to those methods we compare several class corpora with and to each other instead of using a general reference corpus for comparison. Furthermore, we use a novel measure called TFICF which is an adaption of the popular TFIDF measure to assess the importance of a term within a class. This allows us to determine discriminating terms for single classes or sets of classes in the concrete context of other interesting classes. By doing so, we are able to figure out the topical coherences and distinctions among a whole set of particular classes and thus satisfy a very specific information need.

In [120] a term frequency inverse cluster frequency value is calculated to get feature vectors of previously attained clusters of document paragraphs. In contrast to our approach, the cluster simply can be seen as a concatenation of all of its documents so that actually there is no difference to the common TFIDF formula.

Our approach is also situated in the context of contrastive summarization [85] and comparative text mining [161] which is a subtask of contextual text mining [93]. Contrastive summarization has a rather narrow application field, it only regards the binary case (two classes) and is focused on opinion mining. Having reviews of two products, the aim is to automatically generate summaries for each product, that highlight the difference in opinion between the two products.

While the fundamental idea of comparative text mining is closely related to our work, the outcome of the cross-collection mixture model proposed in [161] is rather orthogonal to our approach. The process is subdivided in two steps: (1) themes that are common to all collections are identified, (2) for each discovered theme, it is analyzed what is common among all collections and how the collections differ from each other. Whereas this kind of analysis is based on the themes common to *all* classes, our method does explicitly not account for those themes, but for themes that discriminate one or several classes from the remaining ones.

### 3.4.2 Extraction of product attributes

In the application example that is presented in chapter 7, we use our technique to extract frequent product attributes from customer reviews. In that domain, specific algorithms for the identification of attribute terms exist that are reviewed below.

In [66, 67] the Apriori algorithm is used to search for frequent features (where a feature is defined as a set of terms that occur frequently together in a sentence). Subsequently, two pruning steps are applied to refine the result of this association mining algorithm. The first one checks if the terms of a feature set are sufficiently close together. The second one discards features that are a subset of another feature phrase and do not appear at least $k$ times alone. The authors additionally propose to identify infrequent attributes by assuming that the closest noun phrase to an opinion word must be an attribute as well if none of the previously identified attributes can be found in the sentence. Their evaluation shows that this further increases the recall, but significantly diminishes the precision values. In contrast to this approach, Popescu et al. [107] consider all noun phrases as attributes whose frequency is above a certain threshold. The list of attributes is then further filtered by calculating the PMI score (Point-wise Mutual Information) between each phrase and discriminator phrases (such as “is a scanner” or “scanner has” etc. in case of reviews on scanners). The PMI scores are calculated on a set of web documents containing the product name. The paper reports a 22% increase in precision with 3% loss of recall compared to the results reported in [66]. In [158], Yi et al. propose and compare two different approaches. One of them is based on a mixture language model and the other one applies the likelihood-ratio test. They report better results when using the likelihood-ratio test. Titov and McDonald introduce the concept of a Multi-Aspect Sentiment model in [134] that is based on an adaption of Latent Dirichlet Allocation to extract rated attributes (here called aspects) from reviews. The approach of Kim and Hovy [74] is based on FrameNet [41], an online lexical database which consists of 800 semantic frames. The idea is to label each sentence that contains an opinion-bearing term with semantic roles and defer the attribute and opinion holder in the sentence from these categories. The semantic roles (such as agent, speaker, patient, topic etc.) are assigned by a classifier that is trained with (annotated) example sentences from FrameNet. The decision which semantic roles are the attribute and opinion holder in the sentence, is done with the help of a manually defined mapping table.

## 3.5 Related work for sentiment and opinion analysis

Within the context of opinion analysis three main tasks can be distinguished: (1) Detecting whether a text is subjective or objective, (2) detecting whether the general opinion in a text is positive, neutral or negative, and (3) additionally analyzing what has been commented on positively or negatively. Whereas (1) is usually denoted as Subjectivity Analysis, for (2) and (3) two different terms exist, namely Sentiment Analysis and Opinion Analysis that are more and more used interchangeably in literature. According to [103], the term *sentiment analysis* is more popular among communities with a focus on natural language processing, whereas information retrieval communities and researchers that are strongly associated with Web search prefer the name *opinion mining*. Furthermore, there is a tendency to use the term *opinion mining* for (3) which is also the way it is used in this thesis.

Sentiment and opinion analysis algorithms can be subdivided according to the level that the algorithm is working on. Algorithms that work on document level are based on the assumption that each document comments on a single object. Each document is labeled with one of the three classes *positive*, *neutral*, or *negative*. Alternatively, each sentence or phrase can be analyzed separately. Sometimes, a two-step approach is used which classifies the sentences first into objective or subjective and afterwards refines the labeling of the subjective ones as being positive or negative. Finally, attribute-based opinion mining works on the most detailed level by additionally analyzing *what* has been commented on positively or negatively.

In the following, exemplarily several algorithms for each category are reviewed. Thereby, a focus is put on those approaches and tasks that are closest related to the work in chapter 7. A more comprehensive overview on opinion analysis techniques can be found in [103]. Note that the main contributions of our work in the area of opinion analysis are not in developing a novel automatic approach. Instead, we show how visualization can help to analyze a set of reviews with respect to the quasi-semantic property opinion. Furthermore, we evaluate a popular opinion analysis algorithm in detail and show how this information can be used to improve the algorithm systematically.

### 3.5.1 Subjectivity and sentiment analysis

In literature, two fundamentally different approaches for building a sentence classification model can be found:

- *supervised learning approaches* that expect a set of preclassified sentences as training data to be given

- and *unsupervised learning approaches* that often require a dictionary of positive and negative opinion signal words as input.

In the following, for each of the two categories some examples are given.

### Supervised learning approaches

In [104] and [27], standard techniques for supervised opinion analysis are evaluated. Both studies compare different features and classification methods with respect to their effectiveness for classifying documents as positive or negative. Interestingly, their results are not in every respect consist with each other. Whereas in the test of Pang et al. [104] the usage of unigrams outperforms bigrams, [27] shows that there are cases where the opposite is true.

Other researchers proposed to employ a two-step approach that first classifies sentences into subjective or objective and afterwards determines the polarity score of a document using only the subjective sentences. In [160], Hong and Hatzivassiloglou present some experiments in which news articles are classified into subjective or objective using a Naive Bayes classifier with a bag-of-words feature vector on document-level and a combination of different features such as part-of-speech tags, positive / negative word counts, etc. on sentence level. Subsequently, an average per word log-likelihood score is calculated for each subjective sentence to determine its polarity (positive / negative).
Pang and Lee [102] take the context that the sentence is in into account as well instead of analyzing each sentence individually. For example, they assume that neighboring sentences are likely to belong to the same class. Together with other constraints, this results in an optimization problem which is solved by modeling the constraints in a graph and searching for the minimum cut. After labeling each sentence as either subjective or objective, the overall polarity of the document is determined by means of standard classification algorithms that work on unigram feature vectors.

In [103] more variants can be found that incorporate additional information in the classification process such as relationships between product features or discourse participants.

### Unsupervised learning approaches

Central to most unsupervised approaches is the usage of a sentiment lexicon consisting of two lists with positive and negative opinion signal words, respectively. A high quality of this dictionary is crucial for the performance of the algorithms. Hatzivassiloglou and Wiebe [55] present an algorithm that enhances a given seed list with additional opinion signal words by assuming that adjectives that are connected with certain conjunctions such as “and” will most likely have the same orientation. A similar approach is presented by Ding et al. in [28]. In addition to [55], they also take the attribute that the opinion word refers to into account. This way, a more subtle distinction for opinion words whose orientation is dependent on the context they are used in can be made.

Turney [137] follows a slightly different approach for determining the polarity of a document. First, two-word phrases are extracted with the help of syntactic patterns. Examples of the pattern rules include specifications like this: “If an adverb is followed by an adjective and the third word is anything but a noun, then extract the adverb and the adjective as a potential opinion bearing phrase”. In a second step, the semantic orientation of the extracted phrases is determined by calculating the Pointwise Mutual Information (PMI) to the words “excellent” and “poor” and subtracting the two values from each other. Thereby, the PMI serves as a measure of the degree of statistical dependence between two words. To find out how often two words are close to each other in a document, the *NEAR* operator of the AltaVista search engine was used. Finally, the average orientation of all the extracted phrases is calculated to classify a document either as positive or negative.

Besides lexicon-based approaches, other unsupervised techniques for subjectivity / sentiment classification exist. Among them is [112] that labels sentences as either subjective or objective using a bootstrapping approach. First, with the help of high-precision classifiers, automatically some benchmark sentences are identified. This classification process is based on manually collected lexical items, such as single words and n-grams that are good subjective clues. To increase the recall, those sentences are then mined for additional patterns that are good discriminators between the subjective and objective sentences, allowing to make another parse through the so far unlabeled data with a pattern-based classifier. Furthermore, the newly classified sentences are used to iteratively improve the high precision classifiers.

### 3.5.2 Attribute-based opinion mining

Attribute-based opinion mining is often accomplished by two successive steps: First, the attributes (sometimes also called features or aspects), that have been commented on, are identified. Secondly, the respective opinion that has been expressed on them is detected. Related work on the extraction of attributes that are frequently commented on is discussed in section 3.4. In the following, the different approaches for detecting the opinions about an attribute are reviewed.

The approach that closest resembles the algorithm that we use in chapter 7 is [66]. After extracting attributes that are frequently commented on (see section 3.4), the opinion words are identified. Each attribute is considered a potential opinion. To determine the orientation of an attribute, WordNet's [156] synsets are used. To take negation into account, the orientation of an attribute that is preceded or followed by a negation signal word is inverted. Next, the polarity of each sentence is determined. If a majority of opinion words is positive or negative, this is assumed to be the prevalent orientation for the whole sentence. If the number of positive opinion words equals the number of negative ones, for each attribute in the sentence only its closest opinion word is taken into account. In rare cases, in which still no decision can be made, the orientation of the preceding sentence is decisive. Furthermore, some heuristics help to treat sentences correctly that contain a "but"-clause (phrases starting with *but, however*, etc.). In [28] an extension is introduced that takes the distance of an opinion word to the attributes in the sentences into account. Furthermore, an approach for automatic enhancement of a list of opinion words with context-dependent terms is presented. Finally, [29] tackles the challenging problems of identifying implicit attributes and distinguishing between comments that refer to the analyzed product and the ones that compare the product with another one.

The approach of Popescu et al. [107] is based on extraction rules that formulate syntactic dependencies (e.g.: if [Subject = attribute, Predicate, Object] then the object is a potential opinion as in the sentence "Lamp has problems"). After extracting all phrases that match those manually specified rules, their semantic orientation has to be identified. This is done by means of relaxation labeling, an unsupervised classification technique. The necessary graph is build up by connecting two nodes (which represent word, attribute, or sentence-tuples) if some neighborhood constraint applies. Examples for such neighborhood constraints include a connection with specific conjunctions and disjunctions, WordNet-supplied synonymy, morphological relationships etc. One of the advantages of

the approach is that it permits to take into account that some opinion words change their polarity depending on the attribute they are used with (e.g. “A *long* battery life time” is good, but a “A *long* response-time” is bad).
Similarly, [158] uses a sentiment pattern database to mine subjective sentences.

The technique that is presented in [134] is different in the way that it employs the user given scores instead of working only on the free text itself. Users are expected to enrich their reviews with detailed numerical ratings of the different (predefined) attributes. The presented technique is then used to mine the free text to find out more about the reasons for the specific rating of an attribute. Thereby, Multi-Grain Latent Dirichlet Allocation is used to find words that are closely related to a specific attribute.

Zhuang et al. [164] present an approach that uses the output of a dependency parser to mine attribute-opinion pairs. First, all sentences are parsed for candidate attributes and opinion words using a keyword list. Second, with the help of dependency relation templates, the attribute-opinion pairs are classified as forming a valid attribute-opinion pair or not. Examples for dependency relation templates include the following rule: Node of type adjective is connected to a noun in the dependency graph by the relationship “modifies”. In this case, the noun would be the attribute and the adjective the opinion word. Again, negation signal words may invert the orientation of an opinion word. The templates are automatically extracted from a training data set and can be applied to any other previously unknown dataset afterwards. Additionally, Zhuang et al. identify two domain specific heuristics for mining implicit attribute-opinion pairs - a category that is generally hard to detect.

### 3.5.3 Sentiment analysis and domain dependency

It is known that sentiment analysis cannot be considered as being domain independent. Two main reasons for this can be discerned: First, different text genres tend to express opinions differently. Whereas reviews directly refer to what the author likes or dislikes about an item or an event, a newspaper article often addresses opinions more subtle. Second, many sentiment analysis algorithms are based on a dictionary of positive and negative terms. However, a term that may be positive in one domain does not necessarily exhibit the same orientation in another domain. For example, the term “unpredictable” may serve as a positive opinion signal word in the context of analyzing movie reviews but would not be positive when reporting about a car’s steering abilities [137].

In [8], Aue and Gamon qualitatively analyzed domain-dependency. They reported differences of up to 38% when training an SVM classifier on a different domain (e.g. using movie reviews as training data and web survey data as test data). When the analyzed domains are closer to each other (e.g. movie and book reviews), the observed differences were significantly smaller.

Different approaches to deal with the problem are proposed in literature. These include:

- *Working on domain-independent features only.* E.g., [157] is classifying blog data but training the classifier on movie and product reviews. To overcome the gap between the two domains, only features that are frequent in both domains are used, assuming that those features are domain-independent.

- *Generation of sentiment dictionaries with domain-dependent terms.* While the above described approach might work for document level sentiment analysis, attribute-based opinion analysis requires techniques that are able to classify single sentences or even phrases correctly and therefore cannot afford missing opinion signal words. Alternative approaches propose techniques for the automatic enrichment of a sentiment dictionary with domain-dependent opinion signal words (see e.g. [28, 78]) to be able to apply the techniques to different domains.

- *Hybrid approaches with little domain-dependent training data.* Several approaches aim at building classifiers that can adapt to a new domain with just little domain-dependent, labeled training data e.g. by additionally extracting new sentiment terms from unlabeled data of the new domain [43], or by employing a general domain-independent sentiment dictionary in parallel [4].

### 3.5.4 Visual opinion analysis

Visualization of opinions has not yet been a major focus of research in the area. Nevertheless, since the output of the algorithms is usually a long table with assignments, there certainly is a need for effective visualization techniques.

In [19] the authors suggest to use traditional bar charts to visualize the distribution of positive and negative comments for each attribute. As they take the strength of opinion into account, values between +3 and -3 can exist. Furthermore, the attributes are manually classified into a hierarchy and stacked bar charts are used to visualize the ratings for higher-level attributes. Similar to this, [87] also uses bars but does not consider the strength of opinions. Instead, there is one bar per attribute (and product) and the vertical displacement shows the percentage of positive and negative comments, respectively. The length of the bar is proportional to the number of reviews that commented on the specific attribute.

Pulse [44] clusters reviews according to topic and then calculates the average opinion per cluster. The result is displayed in a treemap with color being mapped to the average opinion. In a text field at the bottom of the tool, the sentences that are most indicative of the observed sentiment in the selected cluster are shown.

The BLEWS system introduced in [45] analyzes blogs with respect to their political orientation. The average emotional sentiment ("level of emotional charge") of a set of articles is visualized as a glow around the bars that represent the number of documents that link to a specific news article. The length of the bars is used to encode how many of the blogs that cite the article belong to the liberal camp and how many to the conservative one.

Morinaga et al. [96] display characteristic phrases for the group of positive or negative sentences in a 2D scatterplot. Squares in the scatterplot represent product names, whereas circles mark the positions of opinion-oriented terms. The terms are selected by automatically determining the most discriminative terms from a set of subjective sentences that are extracted from reviews about the product. The positions of the terms in the scatterplot show the correspondence between the terms and the products.

Wensel and Sood [152] determine the emotions that are associated with important topics in a blog. This information is then visualized with the help of *EmoMeters*, small symbols, similar to a tachometer that show for each topic how positive or negative it is perceived according to the posts. Furthermore, the authors suggest to use line charts (one per topic) to show the changes in topical emotion over time.

In [145], we presented a technique that visualizes the development over time of RSS feeds that report on the U.S. elections. This work is similar to [87] in that we also use the vertical deflection of bars to encode the opinion that is expressed. However, in our case one bar represents one document instead of the summary of all sentences talking about a specific attribute of a product. Moreover, in our visualization the development over time is central, something that is completely omitted in most of the above mentioned approaches for sentiment analysis / opinion mining. [146] reports on an extension of the technique.

[163] presents a geo-sentiment visualization. In the paper, Zhang et al. analyze news articles with respect to the categories joy vs. sadness, acceptance vs. disgust, anticipation vs. surprise, and fear vs. anger. First, a 4D-feature vector with the before mentioned dimensions is created for every article. Next, a summary feature vector is generated for each news site by averaging the values of all the articles from this publisher. This summary vector is then visualized as a line graph that shows the development of the values for this news site over time. Finally, each line chart is placed on a map on the news site's location.

Similarly, Gregory et al. [48] do not only display positive or negative sentiment but also other aspects such as pleasure, pain, power conflict etc. The detected emotions are visualized in an adapted rose plot. A bar below each rose shows the number of documents that are represented in the plot.

## 3.6 Commercial text analysis products

To find out what the state-of-the-art in commercial text analysis software is, we inspected the webpages of companies that offer software with text analysis capabilities. Most of the 39 companies are listed on the KDnuggets webpage (`http://www.kdnuggets.com/software/text.html`, accessed on July 1st, 2009); some additional tools have been found elsewhere. A list of the reviewed companies can be found in appendix A.2. Below a summary of the main topics that are covered by the different products is given. Please note that most companies keep their description of what they do very high-level. Information about the product is usually only given from a user-perspective which means that just by reading the webpages it cannot be judged how advanced and reliable the provided technology is. However, what can be concluded from the evaluation is which features are most often advertised by the companies and thus also what is considered as state-of-the-art by customers. Furthermore, it can be assumed that companies try to offer features that are esteemed by customers and sell well. What in general cannot be deduced from the given data basis is which features would sell well but cannot be offered by the companies so far (and are thus not advertised on the webpages). On the other hand, an interesting observation was that small companies are usually much more visionary in what they claim to be able to do than large, established companies such as SAP or IBM are. Their ideas tend to be more innovative and therefore give an idea where the actual development in the commercial sector might go to.

If someone was asked what text analysis is all about and the only source of information for this person were the webpages of the companies offering text analysis products, he or she would probably answer that text analysis is mainly about information extraction, categorization/clustering and information retrieval techniques. Maybe sentiment analysis and summarization would also be added to the list, as they are growing more and more important in this sector. The two big parts of information extraction (from the perspective of the commercial tools) are named entity extraction (14)[1] and relation extraction (10). Relations are usually detected on entity level, but sometimes also on document level. Additionally some companies offer keyword extraction techniques (4) in their products. Related to that but further into the direction of a semantic understanding of the text are the approaches for concept detection (9). Clustering and classification techniques are offered by 14 of the companies. Usually, the techniques are applied to the document level, only in a few cases the extracted entities or concepts are clustered or categorized. The companies that offer information retrieval technology (13) promote their tools with the functionality to query the search engine with natural language questions (6) or other advanced search techniques (5) (e.g. searching on the extracted concepts or the automatic suggestion of additional search terms). Some of them also offer Desktop Search functionalities (3). Sentiment analysis was offered by 7 of the companies and summarization techniques were incorporated in 8 products. Visualization techniques were provided by 9 products. Mostly, this refers to rather standard techniques such as scatterplots, box plots, bar charts etc. Often, the extracted information is combined with additional (numerical) meta-data that allows the usage of basic visualization techniques. Among the more advanced visualizations are document landscapes (provided by SWAPit (Fraunhofer FIT)) and a visualization that shows some similarity to Sunburst [128] (Eaagle text mining software). Other text analysis features in the tools include document comparison, spell checking, disambiguation, all kinds of natural language processing techniques (e.g. part-of-speech tagging, phrase structure grammar parsing etc.), readability analysis and tools to prepare documents for the semantic web.

[1] The numbers in brackets denote the number of products that advertised the specific feature. Please consider those numbers as a “lower bound” for the real numbers. If a company provided basis functionality in an area (e.g. standard visualizations) but did not consider it to be the core functionality of their product and thus did not clearly advertise it, we might have missed it.

# 4 Quasi-semantic Property I: Literature Analysis Properties

## Contents

COMPUTER-ASSISTED literary criticism is a rather young field in literature analysis. Typically, researchers in literary studies use computers only to collect data that is afterwards analyzed conventionally. Yet, there are some cases in which the computer has already proven useful, e.g., for the analysis of prosody and poetic phonology or for comparing an author's revisions (from version to version). Computer-assisted studies have also been performed in the context of sequence analysis in the past, such as assigning quoted passages to speakers and locating them in the sequence of the text [18].

Visualization is not often used in the context of literature analysis. Commonly, a text is read sequentially and then analyzed by the researcher bit by bit. However, there are literature analysts, such as Franco Moretti, that are an exception to the rule. In his book "Graphs, Maps, Trees" [95] Moretti coins the term *"distant reading"*. The idea is to *"reduce the text to a few elements, and abstract them from the narrative flow, and construct a new, artificial object"* [95]. Moretti uses standard diagrams, maps and evolutionary trees to visualize and analyze meta data about a book collection or certain aspects of a text (such as the names and locations of villages that are mentioned in stories, for example). He advocates distant reading because it allows a researcher to take far more documents into account than this would be possible with standard literature analysis techniques. Furthermore, he points out that focusing on a certain text property and visualizing it appropriately permits to detect structures that otherwise would not become visible.

The work that is presented in this chapter follows a similar track. We also concentrate on certain text properties and visualize them to allow the user to see interesting patterns at a glance. Novel visualization techniques are proposed that permit to analyze the distribution of the values across the text. Thereby, new analysis questions can be addressed that could not be answered with the techniques that are used by Moretti. Furthermore, the text can be analyzed on a much more detailed level - something that was called for by a reviewer of Moretti's book [131] who deemed it proper to take *"the particulars that make the study of literature critical"* [131] into account.

The chapter is structured as follows: After identifying the research challenges from a computer science perspective (in section 4.1), a novel text visualization, the Literature Fingerprinting technique, is introduced (section 4.2). The following two sections present application examples: In section 4.3, the technique is applied in the context of authorship attribution, which is followed by an analysis of the novel *Following the Equator* of Mark Twain and the Bible in section 4.4. The chapter concludes with a summary of the presented work and a discussion of the applicability of the technique in different scenarios.

## 4.1 Research and application context

### Analysis tasks and quasi-semantic questions

The case studies in this chapter are centered around two application scenarios: Authorship attribution and literature analysis in general. Authorship attribution asks for how probable it is that a given piece of writing has been written by a specific person. Thereby, it relies on samples with known authorship from the set of putative authors. To determine the true author, we have to ask what the characteristic and distinct elements in the writings are. Thus, the quasi-semantic property is *writing style*. Authorship attribution techniques, like many other data mining methods, highly depend on an appropriate choice of the features that are used in the analysis. We show that the comparison of the visualizations for different measures leads to insights about the discrimination power of the different measures for authorship attribution.

Our work in the context of literature analysis is in line with Moretti who suggests to concentrate the analysis on a specific aspect of the text and uses visualization to provide a new perspective on the data. Depending on the specific analysis task, different quasi-semantic properties are relevant. An interesting aspect here is that even with rather simple measures (from a computational perspective), we are able to detect interesting patterns and characteristics in the text.

### Research focus of the chapter

Chapter 1 discusses some of the special characteristics of the data type "text". One of them is its sequential nature. Text is not just a set of words, but the order of the words is meaningful as well. This implies that the behavior of the values across the text can have an impact on the analysis. Furthermore, documents are organized with respect to a logical structure such as grouping of words into sentences, paragraphs, sections, or chapters. Those are natural aggregation levels for text analysis and provide valuable information for the user.

The main research contribution in this chapter is the development of a visualization technique that accounts for the above mentioned particularities of textual data. We show that averaging the values over the text can lead to a smoothing of passages with an unusual trend, camouflaging interesting patterns. In contrast to related work (see section 3.2.1), we focus on the scalability and compactness of the representation, allowing to display even a large data set on a high resolution level. This is even more important if not only a single document, but a document collection is to be analyzed. Another research challenge is posed by the fact that some measures provide a value for every single text unit, whereas others only sparsely populate the space. We analyze the texts on different hierarchy levels (by calculating one value per sentence, paragraph, chapter, or text block). Furthermore, by incorporating information about the document structure, it becomes possible to locate interesting patterns within the document. By visualizing the results of the detailed literature analysis together with their position in the text, even local analyses become possible.

Note that the research focus in this chapter is *not* on approximating quasi-semantic properties. In the application examples, we use measures that are commonly used by literature analysts and have been proposed in related work.

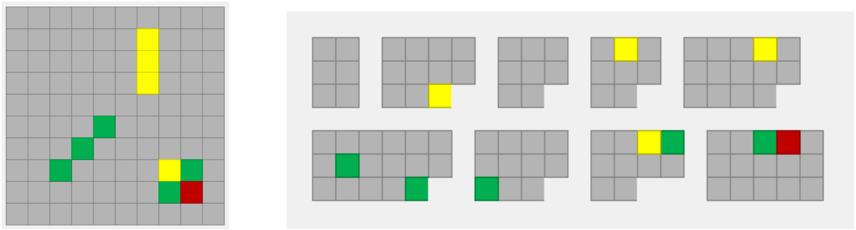
## 4.2 Literature Fingerprinting technique

As specified above, an appropriate visualization that permits to analyze a document (collection) in detail must be scalable, incorporate available information about the document structure, and permit to identify the position of a calculated feature value within the text. A special challenge is posed by data that only sparsely populates the space. In this section, a novel visualization technique, called Literature Fingerprinting, is introduced (section 4.2.1) and several enhancements for sparse data are proposed (section 4.2.2). Finally, section 4.2.3 presents our tool that permits to interactively explore a document with respect to different features.

### 4.2.1 Basic literature fingerprint

The basic idea of literature fingerprinting is to represent documents by a pixel-based visualization in which each pixel represents one unit of text. Pixels are arranged from left to right and top to bottom which results in a compact and scalable visualization. The color of each pixel is mapped to its feature value and therefore permits to analyze the behavior of the feature values across the text in detail. Although simple, this is an effective visualization since the order of the text blocks is very important and the alignment corresponds to the standard reading direction. To display the pixels, we experimented with different shapes such as plain and rounded rectangles, squares with beveled borders and circles. It turned out that the perception of a trend is easiest when displayed on a closed area with no borders visible. For the comparison of discrete values, the other shapes are more useful.

If a hierarchy has been defined on the text (made up of chapters, pages of the book, paragraphs, etc.), the pixels are visually grouped according to that hierarchy. Thereby, the structure of the text can easily be perceived and patterns that discern one passage of the other become obvious. Figure 4.1 exemplifies why grouping the pixels according to the



(a) Sequential alignment (from left to right and top to bottom)

(b) Hierarchical grouping (within each group sequential alignment as in fig. 4.1(a))

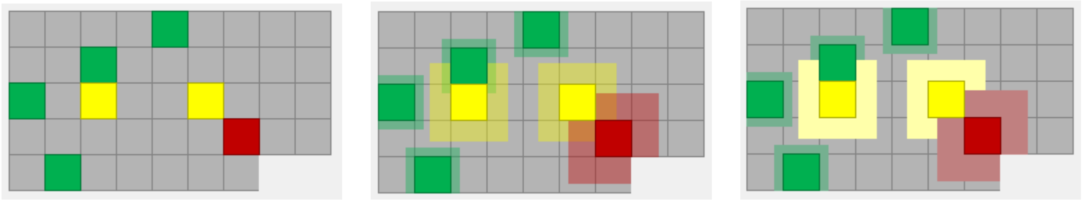
*Figure 4.1: Comparison of sequential alignment and hierarchical grouping. In figure 4.1(a) artifacts can be seen that are not present anymore in figure 4.1(b).*

underlying structure is important. It is easy to spot interesting patterns in figure 4.1(a). However, those patterns are not present anymore in figure 4.1(b) in which the hierarchical structure of the same (artificially created) text is taken into account. As can easily be seen, the patterns in the former graphic are artifacts that are perceived because of the human habit to search for two-dimensional patterns if the data is presented in a two-dimensional space. To avoid the problems of figure 4.1(a) in which neighboring cells do not necessarily represent text units that are close together in the document, we also experimented with other orderings such as the Z-order [97] or the Hilbert curve [61]. However, it turned out that such a layout is too hard to interpret for the user and therefore impedes following the course of the values across the text.

### 4.2.2 Enhancements for sparse data

The basic literature fingerprinting technique works nicely if a value exists for each or at least for most of the text units. However, there are text features whose values are only sparsely-distributed across the text. An example for such a scenario is marking the names of the different characters in a novel to analyze their distribution across the text. If displayed on word level, only few pixels will be colored making them visually less salient. In the following, some ideas and preliminary tests are presented how to deal with this problem. An evaluation of the proposed enhancements is left for future work.

1. *Visual boosting of rare colors*
   A halo is added to pixels that encode a rare value. The size of the halo is inversely proportional to the frequency of the value across the text. This means that rare colors are visually boosted to ensure they are salient enough to be spotted. The frequency of a value can either be measured globally (across the whole text) or locally (e.g. within each sentence separately). Halos are only painted over "empty" pixels to avoid camouflaging other values. As halo color we either use a brighter version of the pixel color or increase the alpha value of the color to get a semi-transparent ring. Using transparency comes with the advantage that even in overlap areas all halos are visible but may lead to blended colors that are difficult to interpret. Figure 4.2 exemplifies the different variants.



(a) Without visual boosting (b) Boosting with semi-transparent halos (c) Boosting with opaque halos

*Figure 4.2: Visual boosting of rare colors. The size of the halo around a pixel is inversely proportional to the frequency of the value across the text. Halos may be painted with semi-transparent or opaque colors. As the halos may overlap each other, the painting order is important.*

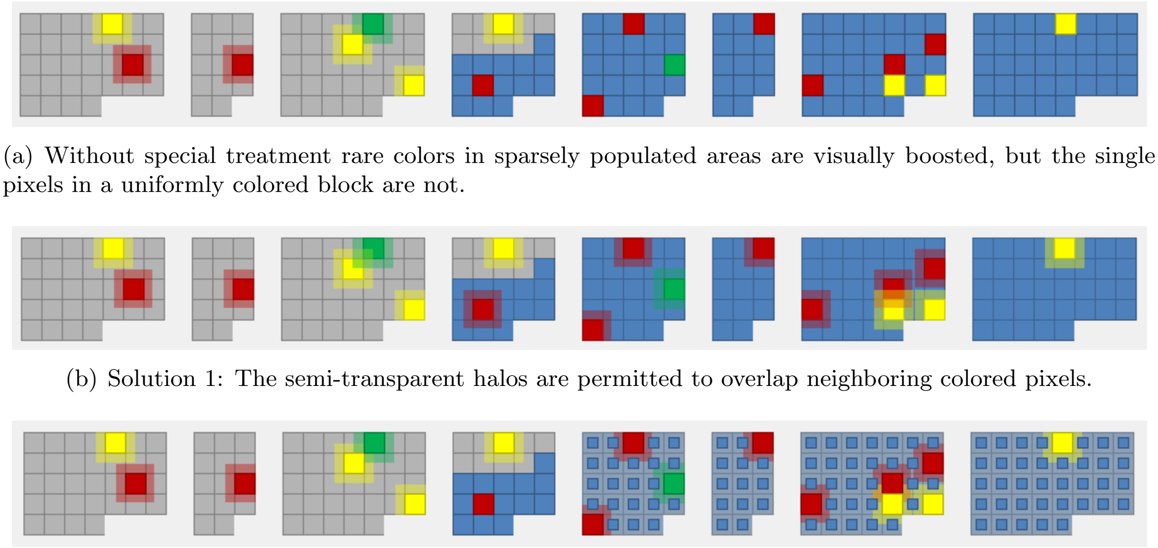
2. *Painting order of halos*
   In areas in which several colored pixels are close to each other, their halos may overlap. In this case, the painting order is important. This is especially true when non-transparent halos are used but also for transparent halos, since the last color will be the most salient one. By painting the halos last whose value is least frequent in the sentence, we ensure that rare colors get the most halo space. If the frequency of two colors in the sentence is the same, the pixel with more neighbors is painted last. We consider colored pixels and non-existent border pixels as "neighboring pixels". Since we do not overpaint those pixels with the halo, they diminish the space of the halo around a pixel (which means that this pixel is more in need for the available space than other pixels are). These heuristics are already considered in figure 4.2.

3. *Boosting rare colors among a uniformly colored block*
   If several text feature are displayed at once, the following situation may occur: There are big uniformly colored text blocks that are interrupted from time to time by a single pixel in another color (see fig. 4.3(a) for example). In this situation using halos as described above does not help, because we do not allow a halo to overpaint neighboring pixels if they are colored themselves. Two possibilities to deal with the problem are: 1.) Allowing semi-transparent halos to overlap with neighboring pixels (without painting a halo around the pixels of the closed area). 2.) Additionally or alternatively decreasing the size of the pixels if the frequency of a color is above a certain threshold in a sentence (e.g. if at least 80% of the pixels in the sentence are painted in the same color). Figures 4.3(b) and 4.3(c) show examples.

Ultimately, deciding on what the most appropriate technique is, is highly dependant on the analysis task that needs to be fulfilled.

### 4.2.3 Interactive analysis tool

To ease the analysis, we set up a framework in which all the measures are implemented and that allows us to explore a text interactively. Figure 4.4 shows a screenshot of the tool. As can be seen, multiple documents can be loaded at a time and therefore compared to each other or - as in this case - one document can be displayed multiple times to compare different measures. For a preannotated document, it is possible to switch between different



(a) Without special treatment rare colors in sparsely populated areas are visually boosted, but the single pixels in a uniformly colored block are not.

(b) Solution 1: The semi-transparent halos are permitted to overlap neighboring colored pixels.

(c) Solution 2: If a color fills more than X% of a block, the size of its pixels is decreased.

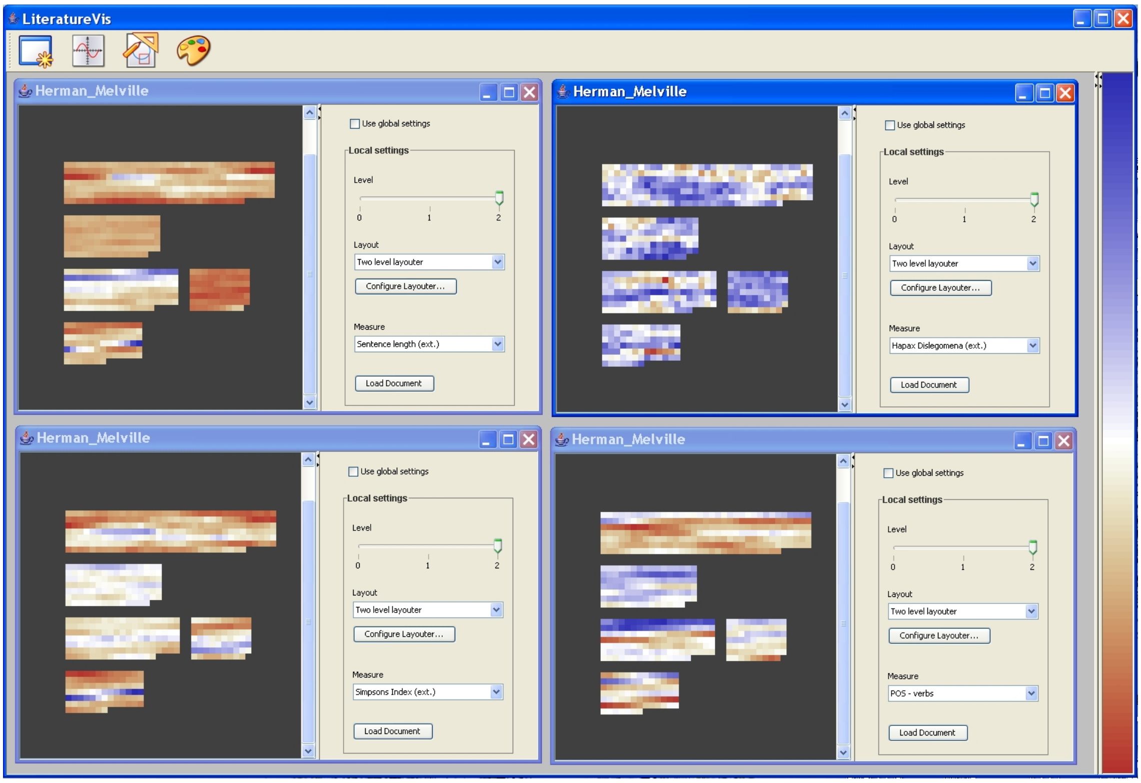
*Figure 4.3: Large, uniformly colored blocks bear a special challenge. If halos are only painted over empty cells, no boosting is performed for rare colors within such uniform blocks. The figure exemplifies two ideas how the problem could be addressed.*

measures at runtime, allowing to interactively explore the document. Interaction with a defined hierarchy is possible and comes in several ways. First, the elements can be grouped according to the hierarchy (e.g. in fig. 4.8 all the points of one chapter are grouped together and the chapters themselves are grouped into books). Secondly, the text can be displayed on different resolutions by choosing the hierarchy level the values should be shown for. Finally, choosing the normalization, the colormap, and the way the elements are displayed, rounds out the interaction capabilities.

## 4.3 Application: Authorship Attribution

The goal of authorship attribution is to determine the authorship of a text when it is unknown by whom the text has been written or when the authorship is disputed. Authorship attribution can also be used when there is doubt whether the person that claims to have written the text is really the creator. One example for such a doubtful situation is the assignment of the 15th book of the series of the Wizard of Oz. The book was published after the death of its author L. Frank Baum and was said to have been only edited by his successor Ruth Thompson who wrote the next books of the series. However, some literature specialists think that Ruth Thompson also wrote the 15th book and that the attribution to Baum was only due to commercial motives to ease the transition from one author to the next without losing sales. See [15] for an interesting analysis on the problem.

In some respects authorship attribution could also be seen as a classification task where the different authors are the categories the unknown samples have to be assigned to. To train the algorithms, a well-defined set of putative authors is needed. As many texts as



□ Use global settings
Layout: Two level layouter
Measure: Sentence length (ext.)

□ Use global settings
Layout: Two level layouter
Measure: Hapax Dislegomena (ext.)

□ Use global settings
Layout: Two level layouter
Measure: Simpsons Index (ext.)

□ Use global settings
Layout: Two level layouter
Measure: POS - verbs

*Figure 4.4: Screenshot of the framework that we implemented to enable an efficient and interactive analysis of texts.*

possible of the putative authors with known authorship are collected and the stylistic traits of those texts are extracted. To get reliable results, enough texts of the potential writers with known authorship have to be available as basis for attributing the text in doubt to one of them. Key for successfully determining the authorship of the text is to choose the features in a way that they effectively discriminate between the different authors. Then it is possible to compare the traits of the disputed text to the traits of the texts with known authorship which hopefully results in a clear voting for one of the authors. Authorship attribution has also been named stylometry, because the classification is based on the distinct stylistic traits of a document and is independent of its semantic meaning.

### 4.3.1 Proposed measures for authorship attribution

Different features for authorship attribution have been proposed. They can roughly be classified into three groups: statistical measures, vocabulary measures, and syntax measures. In [63], a comprehensive survey on features for literature analysis with a focus on authorship attribution can be found. A special requirement for authorship attribution features is that they must not be controllable consciously.
In the following, some examples for authorship attribution measures are given:

**Statistical measures**:
Examples for statistical measures include counting the average number of syllables of a

word or the frequency of certain expressions. Instead of working on the words directly, it is also possible to analyze the proportions of certain *parts of speech (POS)* (such as nouns, verbs, adjectives ...) in the text. By this, the degree of formality of a text can be measured or the style of a text can be compared to its translation in another language. For authorship attribution, the (average) *sentence length* may be used as an indicator of authorship. We use this measure in our case studies, because the results are promising. Nevertheless, in the context of authorship attribution it can be problematic since the length of the sentences is consciously controllable by an author. The measure is therefore not meaningful if the text has been edited by someone else. It has been shown that the distribution of sentence length is a more reliable marker for authorship than the average sentence length [122]. Yet, it is also more difficult to evaluate. Here our technique proves useful, because the visualization of the results allows an effective comparison of the distribution.

**Vocabulary measures**

Vocabulary measures are based on the assumption that authors (and their texts) differ from each other with respect to vocabulary richness (how many words are in the vocabulary of the author and is he/she able to use his/her vocabulary by applying new words as the text proceeds) and with respect to word usage (which words are preferred if several can be applied).

To measure the characteristic word usage of an author, the *frequencies of specific words* are counted. The success of this method highly depends on the appropriate choice of words for which the frequencies are compared. Good results have been reported for function words such as "the, and, to, of, in ..." as the set of words. According to [15], function words have the advantage that writers cannot avoid using them, which means that they can be found in every text and almost every sentence. Furthermore, they have little semantic meaning and are therefore among the words that are least dependent on context. With the exception of auxiliary words, they are also not inflected, which simplifies counting them. Finally, the choice of specific function words is mainly done unconsciously which means that it is an interesting measure for authorship attribution.

Measures of vocabulary richness are mainly based on the evaluation of the number of tokens and different types. In the following, let $N$ denote the number of tokens (that is the number of word occurrences which form the sample text, i.e. the text length), $V$ the types (the number of lexical units which form the vocabulary in the sample, i.e. the number of different words), and $V_r$ the number of lexical units that occur exactly $r$ times. A simple measure for vocabulary richness is the *type-token ratio (R)* defined as

$$R = \frac{V}{N}.$$

This measure has one severe disadvantage, namely its dependency on the length of the text. A more sophisticated method to measure vocabulary richness is the *Simpson's Index (D)* that calculates the probability that two arbitrarily chosen words will belong to the same type. $D$ is calculated by dividing the total number of identical pairs by the number of all possible pairs:

$$D = \frac{\sum_{r=1}^{\infty} r(r-1)V_r}{N(N-1)}.$$

While the Simpson's Index takes the complete frequency profile into account, there are also measures that focus on just one specific part of the profile. For example, [63] reports that Honoré suggested a measure that tests the tendency of an author to choose between a word used previously or utilizing a new word instead, which can be calculated as

$$R = \frac{100logN}{1 - V_1/V}$$

. The measure is based on the number of *Hapax Legomena ($V_1$)* of a text, that means the number of words that occur exactly once. The method is said to be stable for texts with $N > 1300$. Similar to this, the *Hapax Dislegomena ($V_2$)* (the words that occur exactly twice) can be used to characterize the style of an author. According to [63], Sichel found that the proportion of hapax dislegomena ($V_2/V$) is stable for a particular author for $1,000 < N < 400,000$. At first this seems counterintuitive but with increasing text length not only more words appear twice but also words that formerly occurred twice now occur three times and therefore left the set of hapax dislegomena.
Many other methods to measure the vocabulary richness exist. The interested reader should consult [63] for a deeper investigation of the topic.

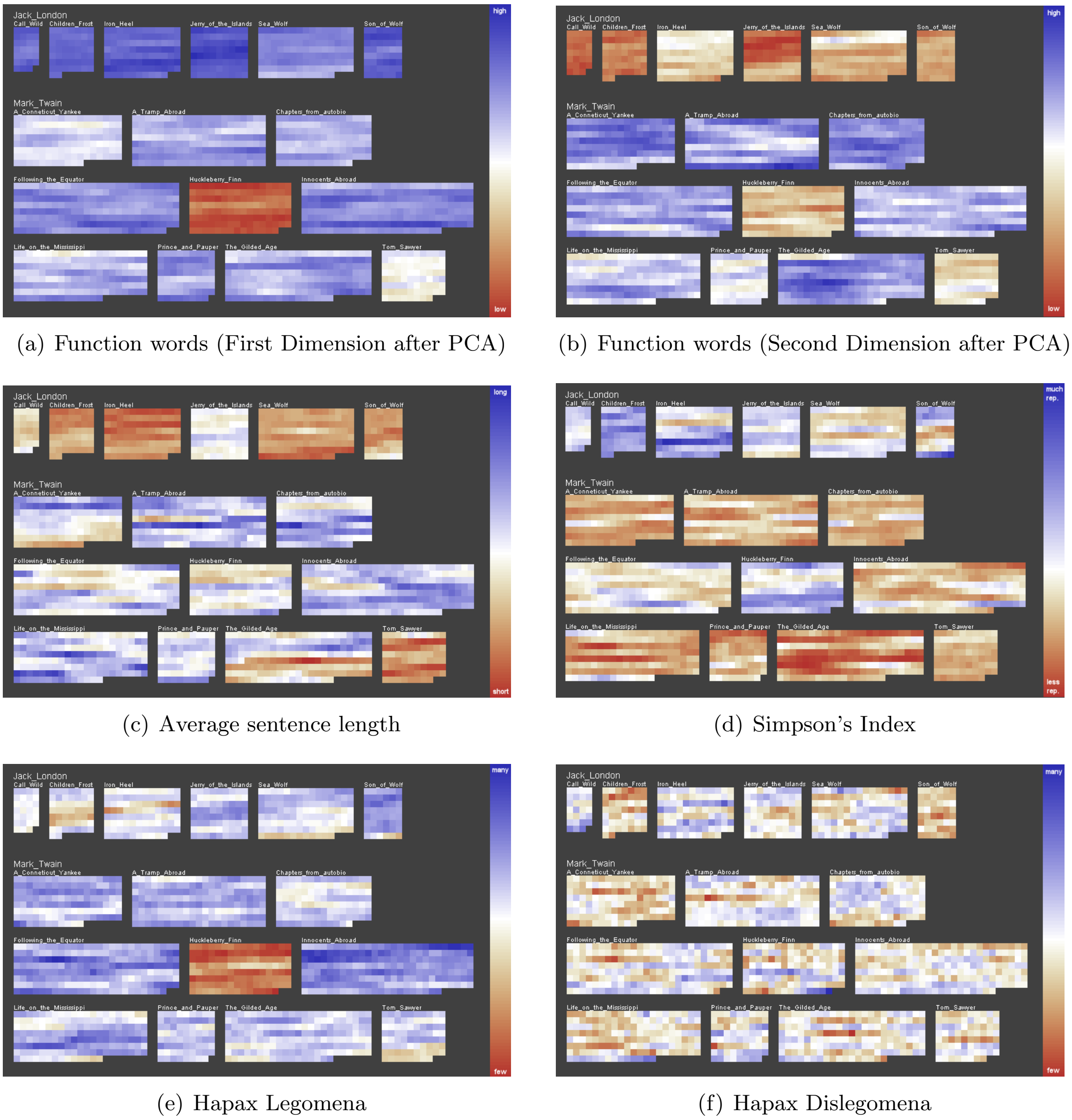
**Syntax measures**
Syntax-based measures analyze the syntactical structure of the text and are based on the syntax tree of the sentences. As the syntactical structure contains additional information, syntax measures have a high potential in literature analysis and have already been used in some projects. In [139], an experiment is reported in which a new syntax-based approach was tested against some word-based methods and was shown to beat them. In another approach [123], the authors build up syntax trees and develop different methods to analyze the writing style, the syntax depth, and functional dependencies by evaluating the trees. Note that – to a certain extend – the usage of function words also takes the syntax into account, because some function words mark the beginning of subordinate clauses or connect main-clauses. Thus, they allow inferences about the sentence structure without analyzing the syntax directly.

### 4.3.2 Case study with literature of Mark Twain and Jack London

In the following, we will present the results of a study with literature of Mark Twain and Jack London. Our goal was to test the existing literature analysis measures and see whether our detailed visual representation leads to new insights.

In our study we used the following texts, that are all publicly available from Project Gutenberg [51]:

- Jack London:
  - *The Call of the Wild*
  - *Children of the Frost*
  - *The Iron Heel*
  - *Jerry of the Islands*
  - *The Sea-Wolf*
  - *The Son of the Wolf.*



(a) Function words (First Dimension after PCA)

(b) Function words (Second Dimension after PCA)

(c) Average sentence length

(d) Simpson's Index

(e) Hapax Legomena

(f) Hapax Dislegomena

*Figure 4.5: Fingerprints of books of Mark Twain and Jack London. Different measures for authorship attribution are tested. If a measure is able to discriminate between the two authors, the visualizations of the books that are written by the same author will equal each other more than the visualizations of books written by different authors. It can easily be seen that this is not true for every measure (e.g. Hapax Dislegomena). Furthermore, it is interesting to observe that the book Huckleberry Finn sticks out in a number of measures as if it is not written by Mark Twain.*

- Mark Twain:
  - *A Connecticut Yankee in King Arthur's Court*
  - *A Tramp Abroad*
  - *Chapters From My Autobiography*
  - *Following the Equator*
  - *The Adventures of Huckleberry Finn*
  - *The Innocents Abroad*
  - *Life on the Mississippi*
  - *The Prince and the Pauper*
  - *The Gilded Age: A Tale of Today*
  - *The Adventures of Tom Sawyer.*

We preprocessed the texts by removing the preamble and other Gutenberg specific parts of the document. Furthermore, short forms were replaced with the corresponding long forms (e.g. isn't → is not). Afterwards we used the Stanford part-of-speech tagger to annotate the texts [126]. For that we had to remove the chapter titles, since the tagger is only able to cope with complete sentences (though it is fault-tolerant with some grammatical errors). Finally, we split the documents into blocks with a fixed number of words each, to be able to show the behavior of the feature values across the text. The number of words per block can be chosen arbitrarily. For this experiment, we set the number of words per block to 10,000. Similar results are obtained for a wide variation of this number as long as the blocks are not too small ($> 1,000$), since some literature analysis measures will provide unstable results when applied to short texts. To obtain a continuous and split-point independent series of values, we overlap the blocks with the neighboring blocks by about 9,000 words. This results in a soft blending of the values instead of hard cuts and therefore enables the user to easily follow the development of the values across the text (even if no hierarchy is defined on the text). As visual representation of the results we depict each text block as a colored square and line them up from left to right and top to bottom.

Since function word analysis is known as one of the most successful methods for discriminating the texts of different authors, we started our analysis with this measure. We took a list of 52 function words that was also used in [15]. For each text block, a feature vector was calculated by counting the frequency of each of the function words, resulting in a 52-dimensional feature vector. We then applied principal component analysis (PCA) to the feature vectors to linearly transform the data to a new coordinate system in which the first dimension accounts for the largest variance, the second dimension for the second largest variance and so on. Figure 4.5(a) shows the values of the first dimension. We use a bipolar, interactively adjustable colormap to map the values to color. If a measure is able to discriminate the two authors, the books of one author will be mainly in blue and the books of the other one will be mainly in red. It is obvious that this is not the case here. What sticks out immediately is Mark Twains *The Adventures of Huckleberry Finn.* This novel seems to differ more from all the other writings of Mark Twain than the writings of the two authors differ from each other. If we visualize the second dimension of the transformed function word vectors, we can see that the books of the two authors now separate from each other (figure 4.5(b)) - again with the exception of *Huckleberry Finn* (and this time also the book *The Adventures of Tom Sawyer*), which we would rather attribute to London than to Twain if its authorship was unknown. To analyze the strange behavior of *Huckleberry Finn*, we tested other features such as Sentence length, Simpson's Index, the

Hapax Legomena measure of Honoré, and the Hapax Dislegoma ratio (see section 4.3.1 for an introduction of the measures). Figures 4.5(c) - 4.5(f) show the visualizations for the different measures. In fig. 4.5(e) *Huckleberry Finn* again clearly stands apart. The Simpson's Index shown in fig. 4.5(d) would again mislead us to attribute the book to Jack London, whereas in 4.5(c) it nicely fits to all the other books of Mark Twain. Finally, the Hapax Dislegoma shown in 4.5(f) seems to have no discriminative power and is therefore not useful for the analysis. Taking all analysis measures into account, it is clear that there is something special about Mark Twain's *The Adventures of Huckleberry Finn*. The reasons for the exceptional behavior cannot be answered by our analysis. The potential explanations range from language particularities such as the southern accent of the novel which may irritate some of the measures over the editing of the text in Project Gutenberg to the surprising speculation that a ghost writer was involved in the creating of the novel.

On the more general side, the figures show that not every measure is able to discriminate between the books of Mark Twain and those of Jack London, and this is also true if the novel *Huckleberry Finn* is excluded from the study. In fig. 4.5(f) (Hapax Dislegomena), we do not see much of a difference between the texts at all. The statement of Sichel that the proportion of Hapax Dislegomena in a text is specific for an author [63] cannot be verified, at least not for these two authors. In contrast to this, the sentence length measure (see fig. 4.5(c)) allows a very nice discrimination between the two authors. Mark Twain's books in average have longer sentences than Jack London's books. Only one novel per writer, namely *Jerry of the Islands* of Jack London and *The Adventures of Tom Sawyer* of Mark Twain break ranks and may be attributed to the other author. The second PCA dimension of the function word vector (fig. 4.5(b)) and the Simpson's Index (fig. 4.5(d)) also provide very nice results. Based on the Simpson's Index, we can observe a trend to a higher vocabulary richness (less repetition) in the writings of Mark Twain than in the books of Jack London.

## 4.4 Application: Literature Analysis

In the following, two pieces of literature are analyzed. Although very simple measures are used, interesting observations can be made. From a research perspective, the first study shows the superiority of a detailed representation over solely working with an overall score for a document. In the second application, we demonstrate the scalability of the method and the advantage of taking the logical structure of the document into account. Furthermore, an example for a sparse data set is given and the value of the enhancements of the basic literature fingerprinting technique (see section 4.2.2) is illustrated.

### 4.4.1 Findings in analyzing the novel *Following the Equator*

When looking at figure 4.5, it becomes apparent that some books are less homogenous with respect to the authorship attribution feature than others. In the following, we will look in detail at two books, whose average sentence length is about the same. The images in Figure 4.6 show the result of splitting the text into overlapping text blocks of 10,000 words each (with an overlap of 9,000 words) and calculating the average sentence length

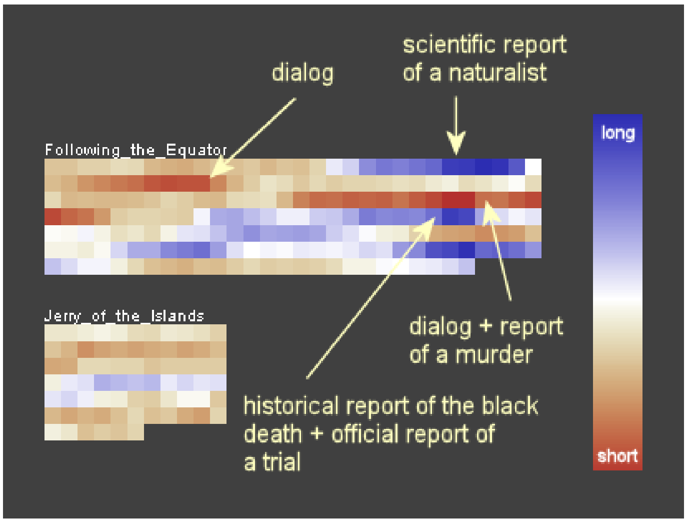


*Figure 4.6: The figure shows the fingerprints of two novels that almost have the same average sentence length. In the detailed view, the different structure of the two novels is revealed. The inhomogeneity of the travelogue Following the Equator can be explained with the alternation of dialogs, narrative parts and quoted documents.*

block-wise. The visual fingerprints reveal that the structure of the two books is totally different despite their identical overall average values. While the average sentence length in *Jerry of the Islands* of Jack London does not differ much across the novel (and thus the total average value would be meaningful), there are significant variations in *Following the Equator* of Mark Twain. *Following the Equator* is a non-fiction travelogue that Mark Twain wrote as an account of his tour of the British Empire in 1895. In fig. 4.6, some passages stick out as they are in dark blue respectively dark red. Taking a closer look at the text reveals the reasons: The long stripe in dark blue in the first line, for example, represents a passage, in which Mark Twain quotes the scientific text of a naturalist with rather complex and long sentences. On the other hand, in the dark red passages in the second and third line Mark Twain noted some conversations that he had during his travel with the short sentences of spoken language. The second dialog is directly followed by the quotation of a written report about a murder. One would rather expect such a report as being characterized by long sentences, but it is not. This is probably why Twain himself utters his surprise about the text in his book. He says about it:

> *"It is a remarkable paper. For brevity, succinctness, and concentration, it is perhaps without its peer in the literature of murder. There are no waste words in it; there is no obtrusion of matter not pertinent to the occasion, nor any departure from the dispassionate tone proper to a formal business statement."* [138]

The dark blue area in the forth line is due to a historical report of the black death and an official report of the trail.

Note that the inhomogeneous structure of the book could only be revealed by a detailed representation of the book.

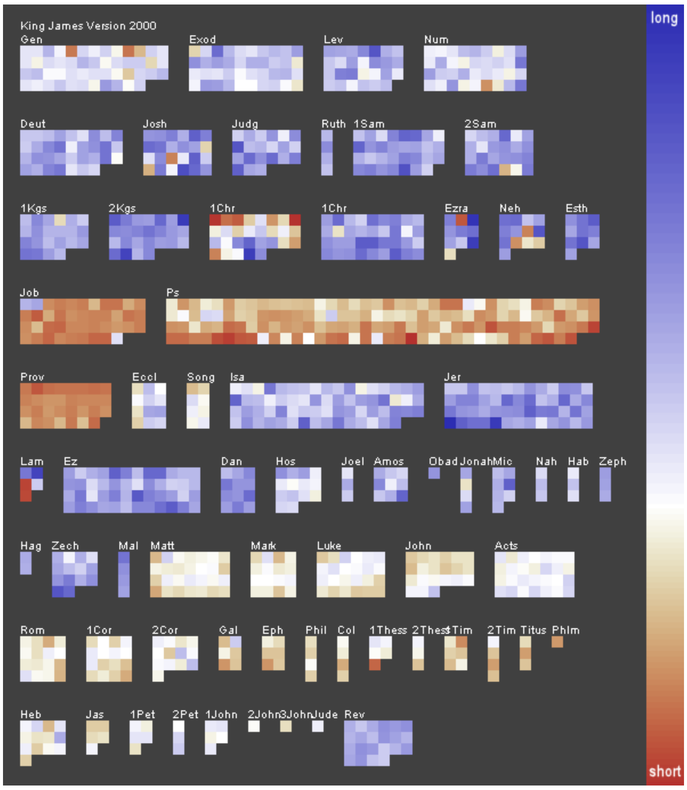


*Figure 4.7: Visual Fingerprint of the Bible. Each pixel represents one chapter of the bible and color is mapped to the average verse length. Interesting characteristics such as the generally shorter verses of the poetry books, the inhomogeneity of the 1. Book of Chronicles or the difference between the Old Testament and the New Testament can be perceived.*

### 4.4.2 Findings in analyzing the bible

In a second study, we analyzed the visual fingerprint of the bible. In this case, we used the existing hierarchy of the text to define the blocks. While every text has an inherent syntactical hierarchy consisting of words and sentences, most texts also have a hierarchy given by the author of the text consisting of chapters, sections, subsections, and paragraphs. In the case of the bible, in addition there is a segmentation into verses. These man-made split points take the discourse structure of the text into account and therefore provide valuable information for the analysis.

#### Analysis of the average verse length

In figure 4.7, each pixel represents one chapter of the bible and the chapters are grouped to books. The color is mapped to the average verse length of the chapters. Using this measure, the books of poetry, also called the books of wisdom *Job, Psalms, and Proverbs*, immediately stick out as the average verse length in those books is constantly shorter than in most other books. Secondly, we can clearly see the split into Old and New Testament. The average verse length of books in the New Testament is in general significantly shorter than the average verse length of the books in the Old Testament. There is one clearly visible exception to this rule, namely the *Book of Revelations*. The fingerprint of the

*Book of 1. Chronicles* reveals that this book is very inhomogeneous with respect to the average verse length. The reason is that the book contains different types of text, namely genealogical trees, precise instructions of how to build the temple, and narrative passages. In the fingerprint, not only whole books with interesting characteristics stick out; there are also some chapters that are quite different than the rest of the book they are in. This is true, for example, for the second chapter in the *Book of Ezra* that lists the people that come back from exile which results in a chapter with extraordinary short verses.

Drilling down to the next level (from chapter to verse level), we can discern details that were camouflaged before. In figure 4.8, each pixel represents a single verse. The verses are grouped into chapters and the chapters are grouped into books. Again, verse length is mapped to color. In the detailed view, we may be able to get additional new insights. Note, for example, that a lot of chapters in *Job* start with a short verse. In the *Book of Job*, Job and his friends take turns at giving long monologues. As the chapter borders were drawn in between two speeches, most of the chapters start with a short verse like *"Then Job / Bildad / Zophar / Eliphaz answered and said"*. Another interesting observation is the clear division of *Nehemiah 10* into two parts. The first one, the one with the short verses, consists of a list of persons that signed a treaty whereas the second part is a historical report. In the coarse representation of fig. 4.7, we were not able to discern that because the average value of the whole chapter is not much different from other chapters. Another interesting observation is the regular pattern of *Numbers 7*, which appears odd. Looking into the text, we find reports how each tribe offers its dedication for the tabernacle. Since the offerings of the tribes were all similar, almost the same text is used for every tribe and therefore the text is repeated twelve times.

### Analysis of the distribution of the person names in the bible

Figure 4.9 shows a word cloud with the most frequent person names of the (German) bible. The more frequent a name is, the bigger is its font size in the graphic. Furthermore, the position of each term was chosen in a way that persons which often co-occur are close to each other in the visualization. Note that only person names that occur more than 50 times in the bible or are mentioned in at least 10 different books are shown. Names that also denote a tribe or district in Israel were excluded. Interestingly, some names of fairly infamous persons such as *Joab* can be found in the list as well. Joab was the commanding officer of David. The reason why *Joab* got such a high frequency score is that his story is tightly interweaved with the story of King David. Thus, this character profits from the detailed description of the life of King David. To distinguish between names that are only frequent in a specific context and those that are wide spread, color was used to visualize in how many books of the bible a name can be found (see color legend below the figure).

Although simple, the visualization is quite expressive. It is obvious that *Mose*, *David*, and *Jesus (Christus)* are the three names that are most frequent in the bible. Furthermore, other key players such as *Jakob* or *Abraham* can be identified that have a high frequency and can be found in many books. However, something that the graphic does not tell is *where* in the text those terms can be found. To view the distribution of the names across the bible, the Literature Fingerprinting technique is better suited. Displaying all the names at once (with a different color for each of them) would not be advisable. According to [148], different studies report that humans are not easily able to distinguish more than 6-12 colors. In the following example only the three most frequent person names are displayed. Figure 4.10 shows the distribution of the terms *Mose*, *David*, and *Jesus Christus* across the bible.

Each pixel represents one verse. If a verse contains the term *Mose*, it is colored in blue. Occurrences of *David* are colored in yellow and the terms *Jesus* and *Christus* are highlighted in green. If a verse contains more than one of those terms, it is marked in red. It is easy to see that each name has its own focal point in the bible. However, occurrences of *David* or *Mose* can also be found in the sections in which *Jesus Christ* is most frequent. While those verses are difficult to spot in figure 4.10, the halos in figure 4.11 significantly improve their visibility. Furthermore, the many single hits for *David* in the middle of the bible become now apparent. The chapters that start with a single verse containing the term *David* are the psalms written by him in which his name is mentioned at the beginning of the text. The halo sizes in figure 4.11 were determined locally. This means that the size of the halo is inversely proportional to the number of pixels in the chapter with the same color. If a color covers more than 30% of a chapter, no halo is used at all for this color, because it is already visually salient enough.

## 4.5 Summary and future work

In this chapter, a novel visualization technique for document analysis was presented. Instead of displaying only a single value per document, the data is shown in detail allowing to follow the development of the values across the text. Several application examples in this chapter such as a visual evaluation of measures for authorship attribution, a detailed analysis of novels and an analysis of the bible proved the importance of this characteristic. Another advantage of the technique is its scalability and the incorporation of the document structure. This permits to display even larger amounts of text without losing the orientation and minimizes the danger of creating artifacts. Furthermore, the document can be shown on different resolution levels. Additionally, several enhancements of the technique were presented that help to deal with the special challenges that a sparse data set comes with.

The technique is not restricted to the area of Literature Analysis but is useful in every document analysis task that requires a detailed analysis of the development of features across a document. The wide applicability of the technique becomes also apparent in the following chapters in which the technique is applied in different contexts. Chapter 5 makes use of the technique as an overview representation esteeming its compactness and scalability. In the context of sentiment and opinion analysis (chapter 7), the visualization is employed in the evaluation process that requires detailed insight into the data to understand and improve the algorithm and for the analysis of a news article. We can observe that within the scope of the research framework that is presented in section 2.1.2, the technique already proved useful in almost every step of the process, be it the feature engineering, the analysis of the document with respect to specific properties, or the evaluation.

In future work, we would like to develop the technique further in the following directions:

- In section 4.3, we compared different features to each other. The technique was well suited for this task, because we were not interested in a comparison of single units of the document, but only in the overall impression. However, comparing several features to each other would be difficult if the exact position of the values within the document is an issue. With the current technique, one graphics per feature would have to be generated forcing the user to targetly compare specific regions to each other. Ideally, the visual representation should facilitate and support this special task. For sparse data, it can be possible to use the current technique by simply plotting everything into one visualization. This was done in the application example in section 4.4.2 in which the position of three person names is shown in the bible. But still, open research question are how to deal with the situation if two values have to be displayed at the same position, how to represent the data on higher aggregation levels, and how to encode more than just three or four features at the same time (as color might not be the right choice). If the data is not sparse, a more rigorous enhancement of the visualization technique is necessary.

- Literature Fingerprinting is a scalable technique that permits to present large documents on a single screen (e.g. the whole bible on word level). However, the scalability has not yet been pushed to a limit. In many application tasks, the exact position of the values is not important, but knowing a tentative location is enough. This invites local compression of the graphics. For sparse and categorical data, the pixel placement techniques that are used in geographical data visualization might be an interesting solution. If the data is continuous and each pixel is filled, compression techniques that are used for scaling pictures down could be exploited.

- The proposed technique permits to select the granularity that the document is displayed on. This gradual zoom could be changed into a continuous one in the future. If a measure exists that rates the different areas for their interestingness, the envisioned smart zoom could be implemented in a way that uninteresting regions are aggregated earlier than the interesting ones. This is based on the assumption that for some regions the detailed representation is more important than for others. One advantage of this extension is that the scalability of the technique is further increased without losing interesting information. Research challenges in this case are to identify the interesting measures for the different tasks and to develop the visualization further in a way that it can deal with displaying different resolutions at the same time.

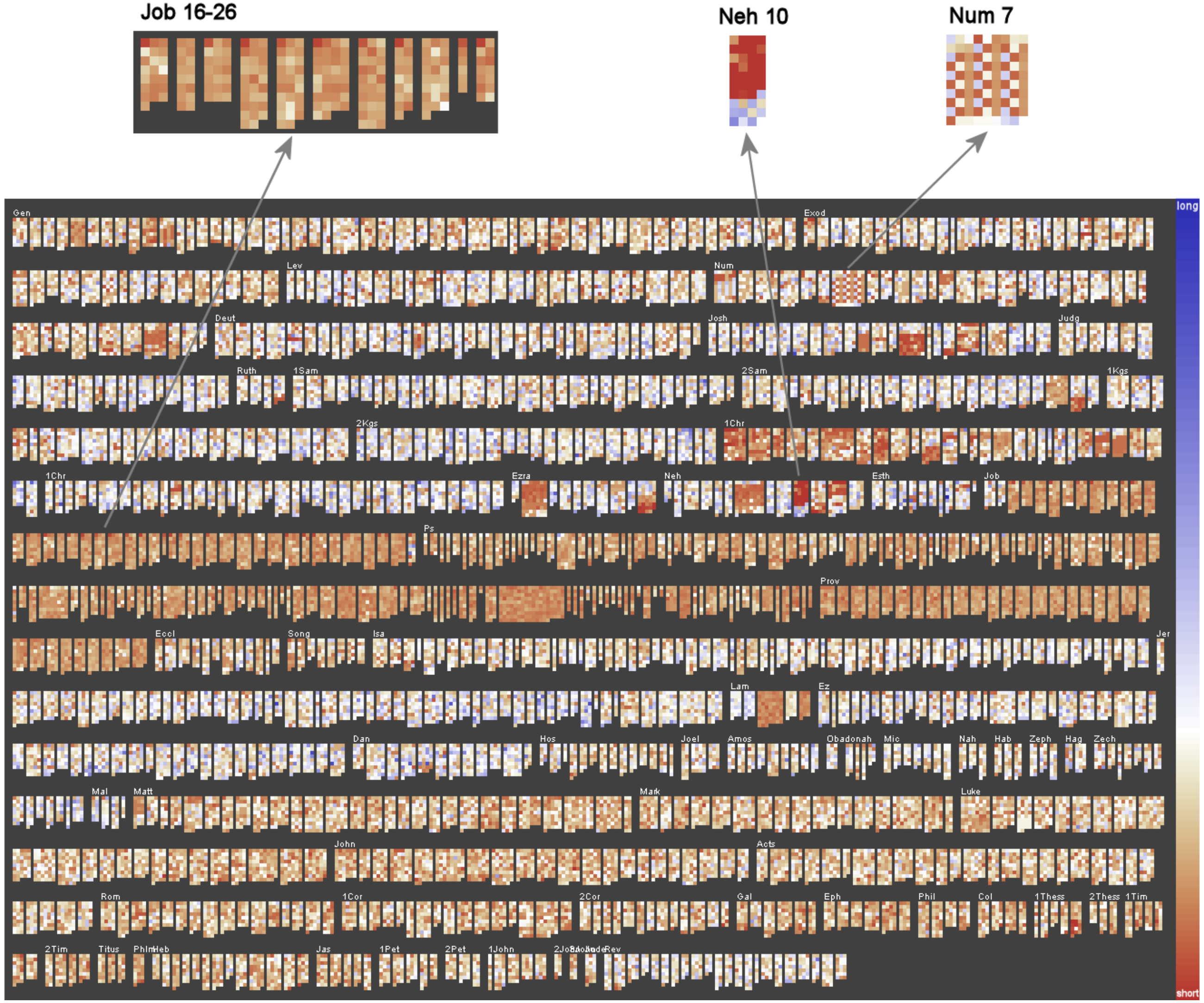


*Figure 4.8: Visual Fingerprint of the Bible. More detailed view on the bible in which each pixel represents a single verse and verses are grouped to chapters. Color is again mapped to verse length. The detailed view reveals some interesting patterns that are camouflaged in the averaged version of fig. 4.7.*

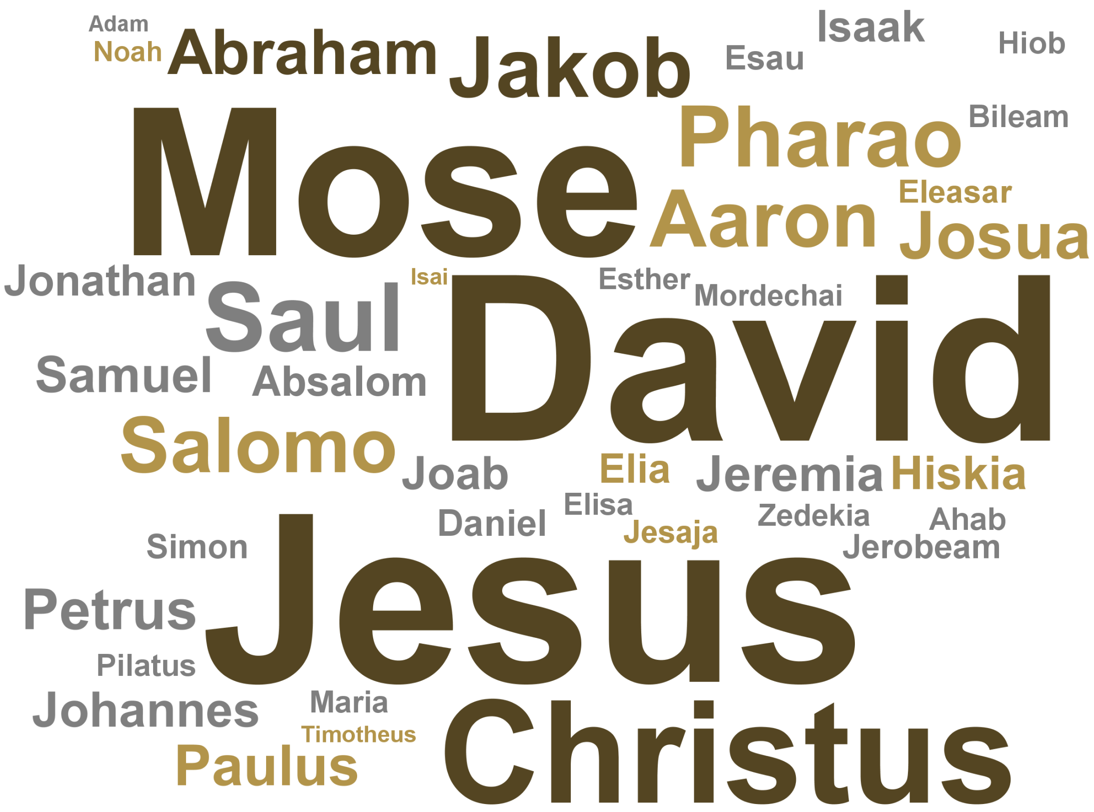


*Figure 4.9: Word cloud of the most frequent names in the bible. The font size is mapped to the frequency of the term. Color represents the number of books the name is in. (Brown = name in more than 20 books, golden = name in more than 10 books, silver = name in 10 or less books.) Furthermore, names that often co-occur are closer together. The graphic was generated with a tool of the working group of Prof. Deussen (University of Konstanz).*

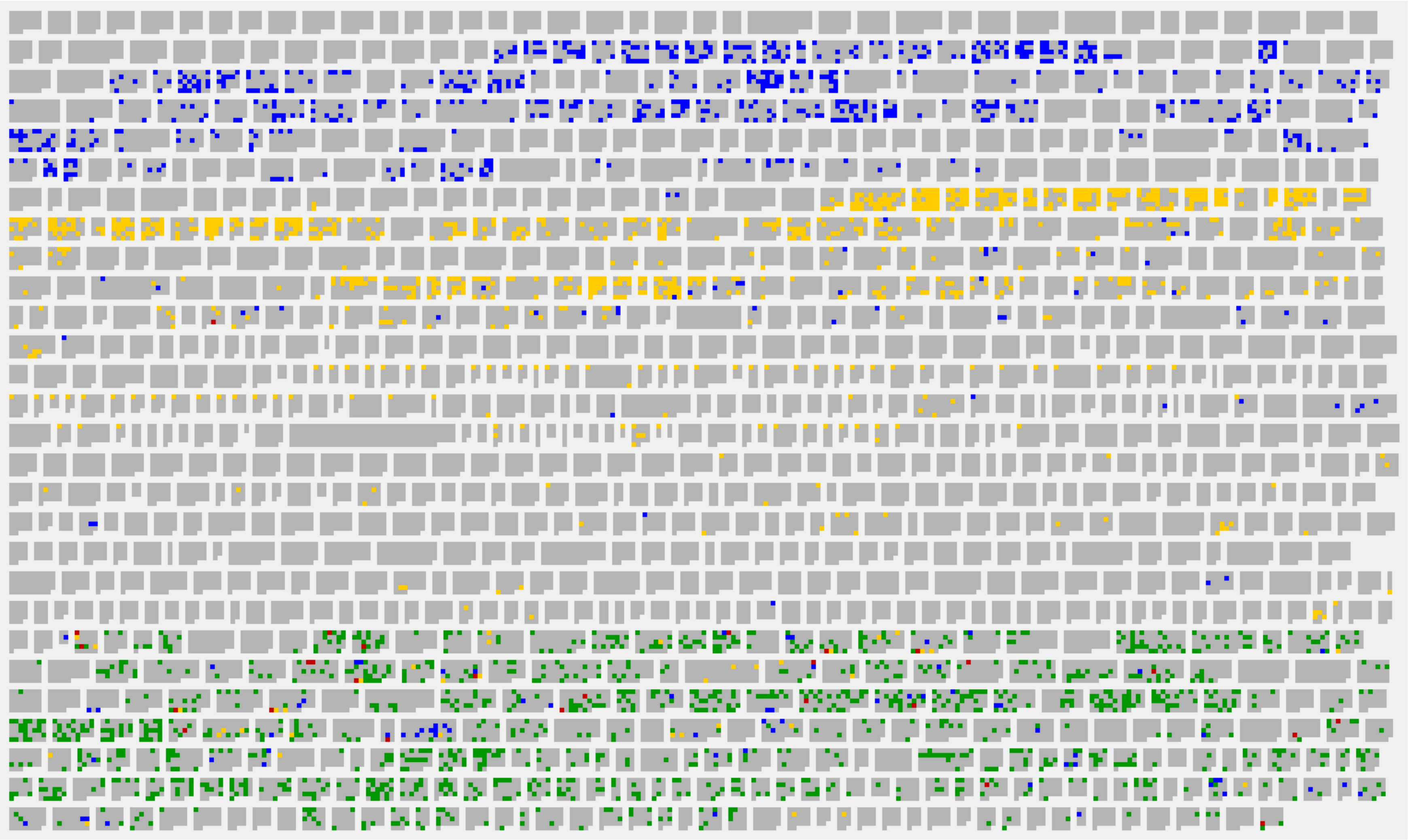


*Figure 4.10: "Mose", "David", and "Jesus Christus" are the most frequent names in the (German) bible. This figure shows their distribution across the bible books. Verses that contain the term "Mose" are colored in blue, occurrences of "David" are colored in yellow and the terms "Jesus" and "Christus" are highlighted in green. If a verse contains more than one of those terms, it is marked in red.*

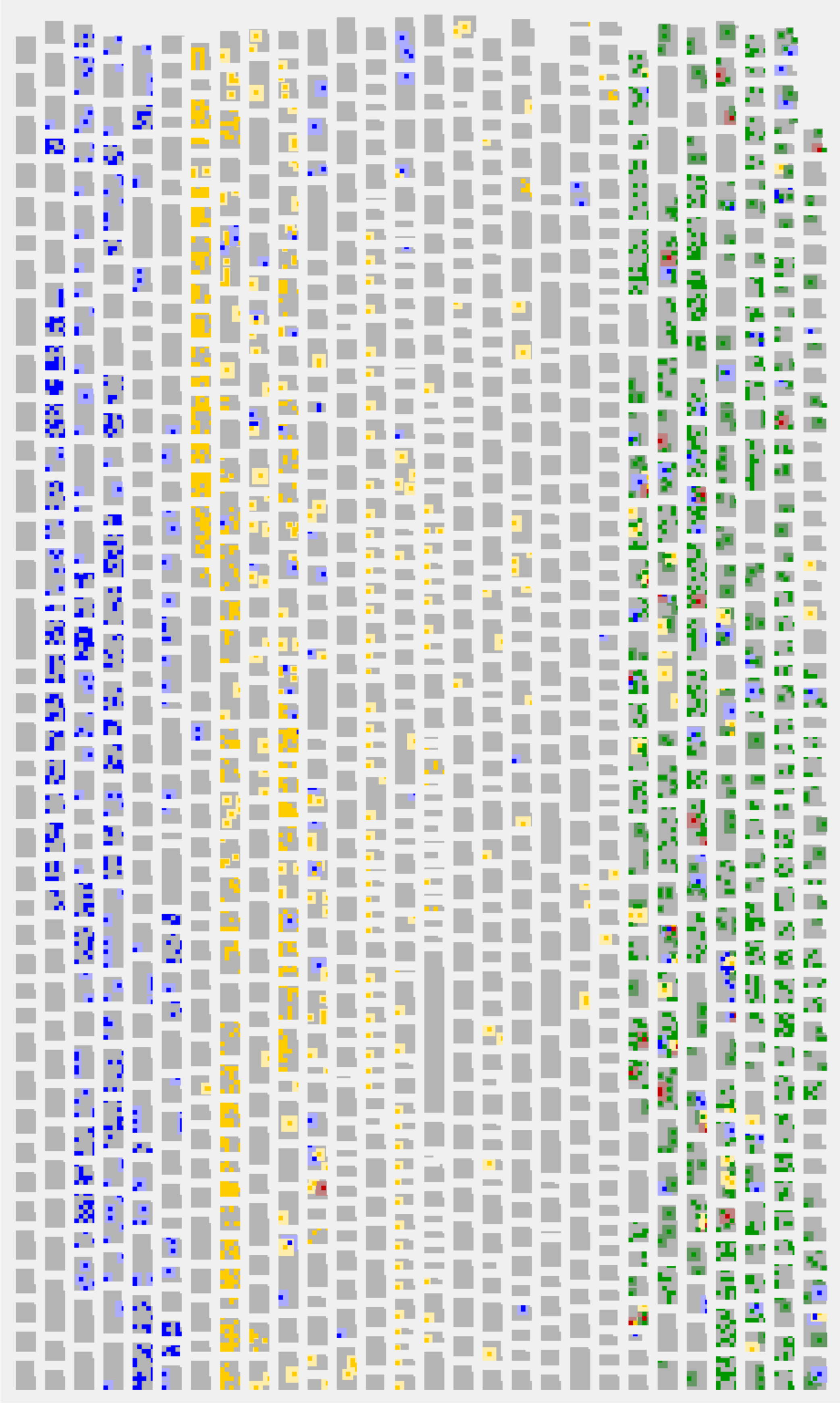


*Figure 4.11: Same as figure 4.10, but this time rare colors are enhanced with the help of halos.*

# 5 Quasi-semantic Property II: Readability

## Contents

A common challenge when producing a text is to write it down in a way that it is easy to read and understand by the target community. This includes aspects like ensuring contextual coherency or avoiding unknown vocabulary, difficult grammatical structures, or misspellings etc. Application scenarios range from writing a paper over wordings of the laws to newspaper articles etc.

The vision that underlies this chapter is to build a tool that supports the user in revising a draft-version of a text. In addition to showing which paragraphs and sentences are difficult to read and understand, we want to help the user understand *why* this is the case.

A special challenge in our application scenario is issued by the need for features that are expressive predictors of readability, and additionally are semantically understandable. Furthermore, they must allow for a detailed analysis of the text with respect to the reasons for the observed difficulties. After an introduction of the research and application context (section 5.1), section 5.2 discusses how we find appropriate features from a large set of candidates using a semi-automatic feature selection approach. Following this, section 5.3 introduces the VisRA tool. The tool is designed in a way that it is easy to see the behavior of the features across the document but at the same time identify the single paragraphs and sentences that are most in need of being revised. Visualization techniques support the user in the analysis process and are employed to convey the information about why a sentence or paragraph is difficult to read and/or understand effectively. Finally, the case studies in section 5.4 show the wide applicability of our tool.

## 5.1 Research and application context

The research that is done in this chapter is motivated by a typical scenario in a research institution: A paper, thesis, or proposal was written (often in a team with multiple authors) and needs to be revised before it is submitted. In this case, several properties need to be checked: Is it well readable? Is it consistent? Are all the new or uncommon terms explained? Is it well structured? etc.

### Analysis tasks and quasi-semantic questions

This chapter focuses on the quasi-semantic property *readability* of a text. The quasi-semantic question in this case would be *"Is the text well readable?"*.

Two basic aspects of readability can be distinguished: linguistic and contentwise difficulties. Consider e.g. the sentence *"I think, therefore I am"*. It is not difficult to understand the sentence in terms of vocabulary or grammar, but contentwise, it requires some deeper thoughts. Additionally, contextual coherence and consistency, but also the print layout of a page influence how well readable a document is. In this work, we concentrate on features that measure linguistic and also partly contentwise appropriateness.

Analysis tasks in the above scenario include the exploration of a text with respect to *where* there are passages that are difficult to read and *why* this is the case. This results in the requirement that the measure that is used to approximate the quasi-semantic property must be comprehensible for the user. Furthermore, the tool must be designed in a way that gives the user access to this transparency of the measure.

### Research focus of the chapter

The chapter has two main research foci:

1. Finding an appropriate measure that approximates the quasi-semantic property *readability*. In contrast to related approaches (see section 3.3), our application scenario requires that the different semantic aspects are transparently covered by the measure to support the user in revising the text.

2. The design of a visual interface that supports the user in analyzing a document with respect to where and why the text is difficult to read.

Starting with a large amount of (simple) text features, a meaningful subset has to be determined that is able to approximate the different aspects of readability. We make use of ground-truth data that covers examples for very difficult and very easy to read documents. This enables us to use standard feature engineering approaches for the preselection of candidate features. A special requirement is to approximate the quasi-semantic property with features that are semantically understandable by the user. This disallows the usage of standard readability measures that are mainly based on statistical features (see section 3.3) and poses a special challenge in the feature engineering process. Furthermore, the different causes of bad readability must be covered by the measure. Note that these requirements are typical when working with the introduced framework. Many quasi-semantic properties, such as consistency, interestingness, or quality (see section 2.2.1 for more examples) require to take multiple aspects into account and make them transparent for the user.

The second part of the chapter presents a visual interface that is designed in a way that a detailed analysis of the document with respect to the above mentioned properties becomes possible. Thereby, the challenge of presenting details but at the same time providing an overview is met. Furthermore, it is taken into account that different types of documents may require a different presentation. This can also be caused by the lack or availability of information about the logical and physical document structure. Finally, it is discussed in which scenarios the presented interface is applicable.

## 5.2 Finding semantically rich readability features

In the feature engineering process, our goal was to search as unbiased as possible for text features that are expressive with respect to readability. We therefore implemented 141 different text features which can be classified into the following categories:
(Please refer to appendix A.3 for a complete list of features.)

- *Features that are based on word classes:* After a text has been part-of-speech tagged (using the Stanford POS Tagger [136]), the frequencies of the different word classes (such as nouns, verbs, pronouns, etc.) are calculated. Furthermore, the ratio between different word classes is taken into account.

- *Features that are based on word frequencies:* Large document collections such as the Project Gutenberg (`http://www.gutenberg.org/`) or Wikipedia (`http://www.wikipedia.com`) permit to calculate the average usage frequency of a word. We exploited those resources to determine how common the words of a text sample on average are. This was done on different granularity levels, taking the most frequent 50, 100, 500, 1000, or 2000 words into account. In some application scenarios, it is more appropriate to determine the most frequent terms on a domain-dependent collection. The ratio behind this is that even words that are difficult to understand in general may be well-known within a specific community and therefore appropriate to use in such a context. Since we analyze documents from the visual analytics community in two of our case studies, we additionally calculated term frequencies on a collection of VAST and InfoVis papers of previous years.

- *Features that analyze the sentence structure:* Besides measuring the sentence length, we implemented features that are based on the phrase structure tree[1] of a sentence as determined by the Stanford Parser [77]. Features such as the depth of the phrase structure tree, its branching factor or the position of the verb were implemented to take the grammatical structure of a sentence into account.

[1] A phrase structure tree is a hierarchical representation of a sentence that is build according to the nesting of its (sub)phrases.

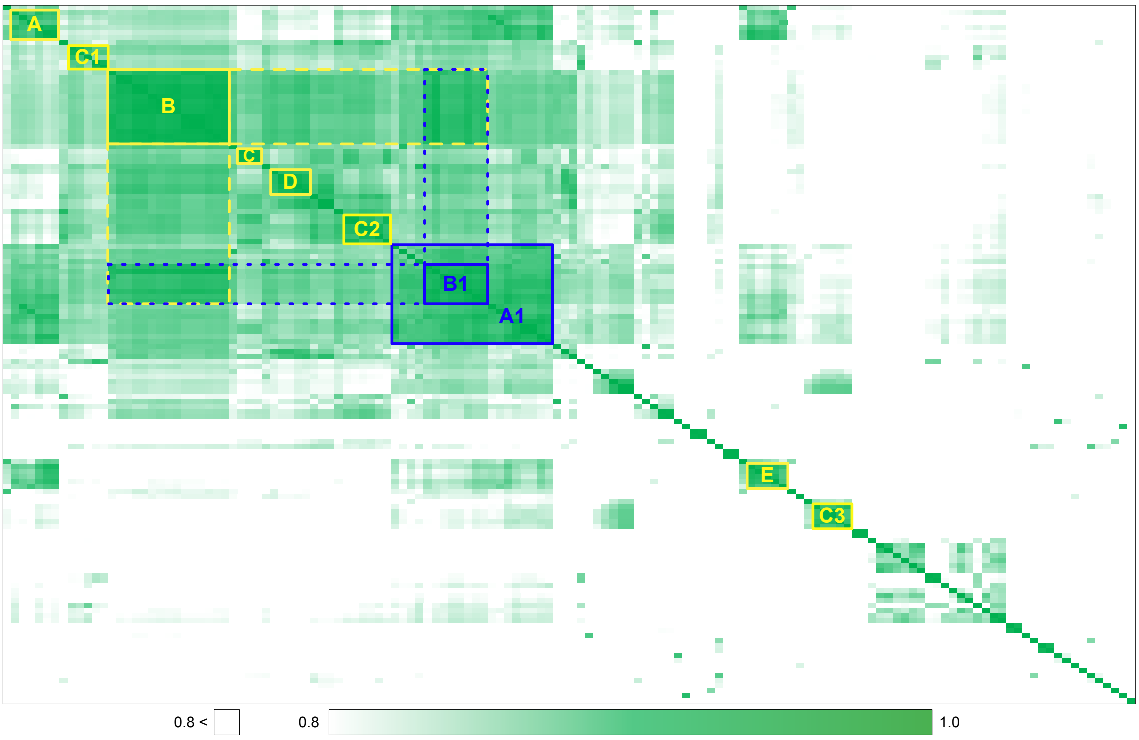


*Figure 5.1: Correlation matrix of the features remaining after removing the ones with low expressiveness. As can be seen, some features are highly correlated to each other measuring the same aspect of readability.*

- *Others:* In addition to the aforementioned features, several other features were implemented, e.g. measuring the number of quotations in a text or the number of sentences in passive voice.

The selection of appropriate features is performed in a two step process. First, the feature set is reduced by removing all features that only show a low expressiveness with respect to the text property *readability*. Second, a set of semantically meaningful, non-redundant features is being determined.

### 5.2.1 Step 1: Removing features with low expressiveness with respect to readability

Using a ground-truth data set of text samples that are very easy respectively very difficult to read, features that show no or only a very low expressiveness with respect to readability are filtered out. The necessary ground-truth data set is compiled of a collection of books for children (most of them are rated as being suitable for children aged 4 to 6) and the work program of the FP7 initiative[2]. Please refer to appendix A.4 for a list with the documents. The documents are split into text samples of about 1000 words each. Next, the 65 samples that are rated by the Flesch Reading Ease Measure [39] and the easiest respectively most difficult ones are chosen to be a part of the training data set. For each

[2] FP7 stands for the *Seventh Framework Programme for Research and Technological Development* of the European Union, whose work programs are generally agreed on as being difficult to read.

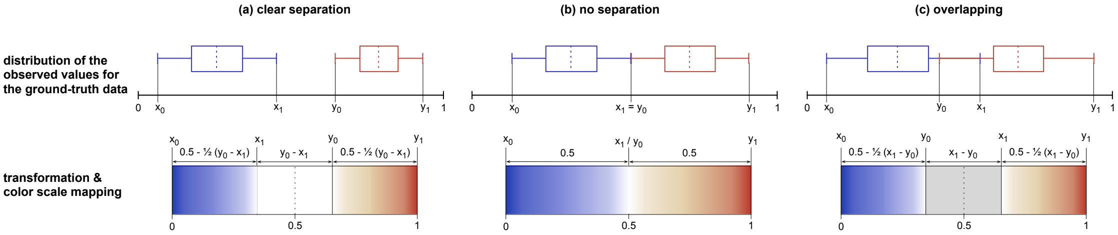


*Figure 5.2: Normalization of the feature values is done relatively to the values that we observed for our ground-truth data set. The graphic shows the formulas and color scales for the three different cases that are possible.*

of the 141 features and 130 text samples a normalized value between 0 and 1 is calculated, resulting in a 130 dimensional vector for each feature. To determine the discrimination power of each feature, the Pearson Correlation Coefficient is calculated assuming that the ideal feature should rate all FP7 documents as 1 and the samples that are taken from children's literature as 0. Only features that score at least 0.7 in this test are kept (which is about 40% of all features).

### 5.2.2 Step 2: Selecting semantically meaningful, non-redundant features

After filtering out all features that show a low discrimination power with respect to the two classes, we select appropriate features that a) are semantically meaningful and b) are non-redundant (i.e. do not measure the same aspect of readability). Using again the Pearson Correlation Coefficient, the correlation factors between all possible feature pairs are calculated. To detect features that highly correlate with each other, we resort the rows and columns of the resulting correlation matrix with the help of a hierarchical clustering algorithm. Furthermore, the cells of the matrix are colored according to the value that they represent (starting with values $\geq 0.8$, see color scale in figure 5.1). Next, the clusters are manually inspected to find out which semantic aspect they measure. For each cluster, one feature is chosen as a representative. If there is no common semantic aspect, the feature is chosen that is easiest to understand. "Easy to understand" in this case means that the feature must be consciously controllable when writing a text, allowing an analyst to improve the readability of a sentence with respect to this feature.

In figure 5.1, clusters from which features were chosen are marked in red. Cluster $B_1$ was dismissed because of its strong correlation to cluster $B$ (see overlap area of dashed lines). The same is true for $A_1$ which correlates with $A$. Interestingly, the clusters $C$, $C_1$, $C_2$, and $C_3$ contain features that are semantically similar (different variants of measuring nominal forms), but despite of this, no strong correlation can be perceived. Features that are not distinguishable on a semantic level do not help the user when refining a text. We therefore decided to choose one feature from each cluster but to present only the one with the highest score to the user. Cluster $D$ summarizes features that measure how common the used vocabulary is (in comparison to a reference corpus). Finally, cluster $E$ contains features that measure the sentence structure complexity.

### 5.2.3 Resulting feature set

Finally, the following features were selected:

- *Word Length*: Measured as the average number of characters in a word.
- *Vocabulary Complexity*: Measured as the percentage of terms that are not contained in a list of common terms. These terms are either defined as the 1000 most frequent terms in a large document collection of the specific language (the so-called basic vocabulary of the language)[3] or are determined from a set of documents of the specific domain (in this case VAST/InfoVis papers).
- *Nominal Forms*: This is a combined measure (see section 5.2.2) consisting of features that take the noun/verb ratio and the number of nominal forms (i.e. gerunds, nominalized words (ending with *ity, ness, etc.*) and nouns) into account.
- *Sentence Length*: Measured as the number of words in a sentence.
- *Sentence Structure Complexity*: Measured as the branching factor in the phrase structure tree of a sentence. This measure is related to the one proposed in [159]. It follows the assumption that the mental complexity of processing a sentence is increased if parts of the sentence are interrupted by subordinate sentences or parenthesis. In this case, the brain is forced to remember incomplete parts of the sentence.

All features are normalized with respect to sentence length and mapped between 0 and 1. We use the values that we observed for our ground-truth data set to determine the normalization factors for each feature. Figure 5.2 shows the three cases that are possible: (a) The values of the easy-to-read samples are clearly separated from the values of the difficult ones. (b) There is no separation at all between the two classes. (c) The observed values overlap each other, meaning that there is a range of values for which we cannot decide the class the text unit belongs to.

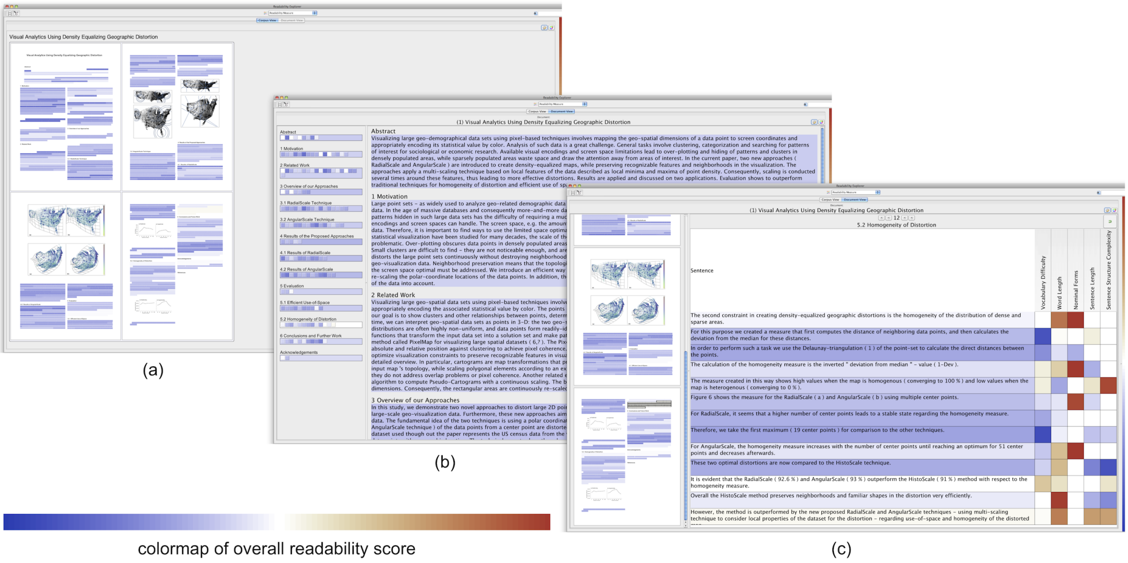
The feature values are normalized in a way that the interval size for both classes is the same (e.g. one class between 0 and 0.4 and the other class between 0.6 and 1). The distance between the observed values of the two classes is accounted for by the size of the gap between the two intervals (see graphics and formulas in figure 5.2).

For the values of the easy-to-read samples a color scale from light green (fairly easy) to dark green (very easy) is used. Similarly, values in the interval of the difficult samples are colored in shades of red. Values in between the two intervals are colored in white if there is a clear separation between the two classes, and in grey if both classes overlap (see color scales in figure 5.2).

### 5.2.4 The readability measure

Central to the concept of our tool is to provide the user with a detailed view allowing him or her to determine why a specific sentence is difficult to read. However, in the overview representations we still need a single value for each section or paragraph that guides the user to the sections that need a closer inspection. We therefore calculate the average of the different features as an overall readability score.

[3] As an English word list we use [52] (based on Project Gutenberg), our German word list is [53] (calculated on a corpus of news articles).


Visual Analytics Using Density Equalizing Geographic Distortion
(a)
(1) Visual Analytics Using Density Equalizing Geographic Distortion
Abstract
1 Motivation
2 Related Work
3 Overview of our Approaches
(b)
colormap of overall readability score
Readability Explorer
Readability Measure
Corpus View
Document View
(c)

(1) Visual Analytics Using Density Equalizing Geographic Distortion

12

5.2 Homogeneity of Distortion

| Sentence | Vocabulary Difficulty | Word Length | Nominal Forms | Sentence Length | Sentence Structure Complexity |
|---|---|---|---|---|---|
| The second constraint in creating density-equalized geographic distortions is the homogeneity of the distribution of dense and sparse areas. | | | | | |
| For this purpose we created a measure that first computes the distance of neighboring data points, and then calculates the deviation from the median for these distances. | | | | | |
| In order to perform such a task we use the Delaunay-triangulation ( 1 ) of the point-set to calculate the direct distances between the points. | | | | | |
| The calculation of the homogeneity measure is the inverted " deviation from median " - value ( 1-Dev ). | | | | | |
| The measure created in this way shows high values when the map is homogenous ( converging to 100 % ) and low values when the map is heterogenous ( converging to 0 % ). | | | | | |
| Figure 6 shows the measure for the RadialScale ( a ) and AngularScale ( b ) using multiple center points. | | | | | |
| For RadialScale, it seems that a higher number of center points leads to a stable state regarding the homogeneity measure. | | | | | |
| Therefore, we take the first maximum ( 19 center points ) for comparison to the other techniques. | | | | | |
| For AngularScale, the homogeneity measure increases with the number of center points until reaching an optimum for 51 center points and decreases afterwards. | | | | | |
| These two optimal distortions are now compared to the HistoScale technique. | | | | | |
| It is evident that the RadialScale ( 92.6 % ) and AngularScale ( 93 % ) outperform the HistoScale ( 91 % ) method with respect to the homogeneity measure. | | | | | |
| Overall the HistoScale method preserves neighborhoods and familiar shapes in the distortion very efficiently. | | | | | |
| However, the method is outperformed by the new proposed RadialScale and AngularScale techniques - using multi-scaling technique to consider local properties of the dataset for the distortion - regarding use-of-space and homogeneity of the distorted | | | | | |

*Figure 5.3: Screenshot of the VisRA tool on three different aggregation levels. (a) Corpus View (b) Block View (c) Detail View. To display single features, the colormap is generated as described in section 5.2.3 and figure 5.2.*

## 5.3 Visual interface for analyzing documents with respect to readability

Figure 5.3 shows a screenshot of the VisRA tool. Three different views are available: The Corpus View (figure 5.3(a)), the Block View (figure 5.3(b)), and the Detail View (figure 5.3(c)).

### 5.3.1 The corpus view

The corpus view (see figure 5.3(a)) serves as an overview representation. In this view, each document is represented by a rounded rectangle whose color is mapped to the overall document score. Within such a document thumbnail, the development of the feature values across the document is indicated by an embedded visualization. Some of these visualizations make use of the internal structure of the document (e.g. chapters and sections) and/or the physical layout of the pages. If no structure is available, the document is split into equal-sized blocks of text whose size may be determined by the user. Depending on the type of document (corpus) that is to be analyzed, the user can choose between three different embedded representations (see figure 5.4):

- *Structure Thumbnails*: If the structure and the print layout of the document(s) are known, structure thumbnails can be employed (see figure 5.3(a) and 5.4(a)), including as many details as possible.

- *The Seesoft representation*: If the print layout is unknown, a representation like the one suggested in [11], which represents each sentence as a line whose length is proportional to the sentence length, may be suitable (figure 5.4(b)).

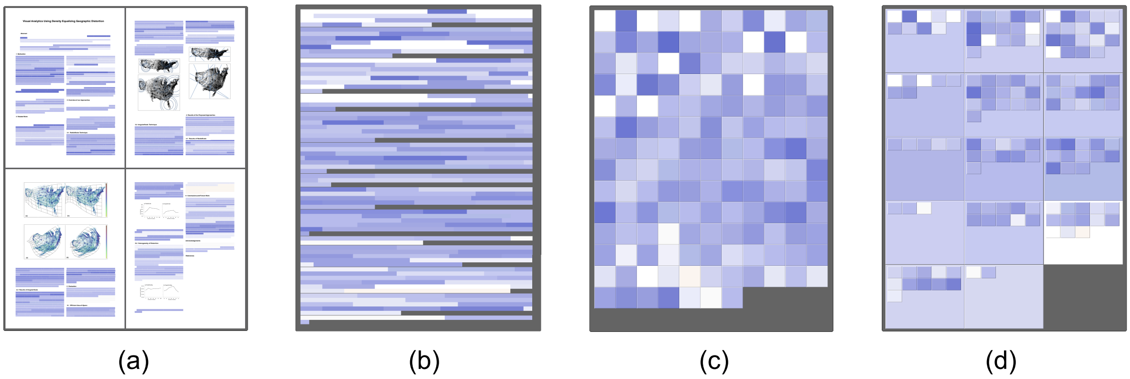

(a) (b) (c) (d)

*Figure 5.4: Embedded representations: (a) Structure Thumbnails, (b) Seesoft representation and (c, d) Literature Fingerprinting representation.*

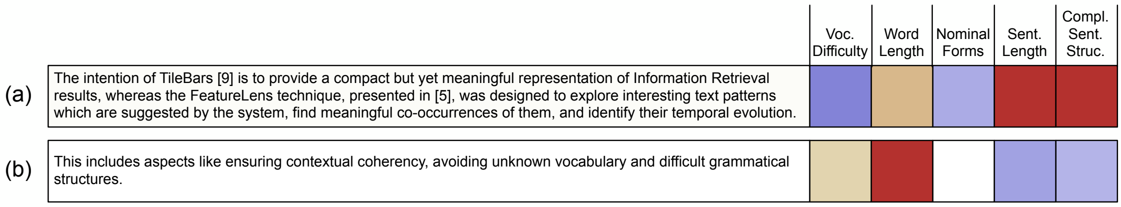
- *The Literature Fingerprinting representation*: As suggested in [73], each text unit (e.g. a section/block or a sentence) is represented by a single square that is colored according to the calculated feature value. The size of the squares is chosen in a way that the whole document can be displayed at once (see figure 5.4(c)). If enough space is available, big rectangles are used instead of squares to visualize the blocks and the sentence level is shown within them using small squares to depict a sentence (figure 5.4(d)). This technique is the most scalable one of the three, allowing to provide an overview even for large documents, respectively to show several documents at once on the screen.

### 5.3.2 The block view

In this intermediate level, complete blocks or sections are displayed and are colored with the overall score of this section / block (see figure 5.3(b)). In contrast to the corpus view, the text is already readable in this view, allowing the user to choose the section that is most in need of being revised. Both, the block view and the detail view offer a navigation panel at the left which can be used to locate the position of the displayed text in the document and to select a specific region for further analysis. Again, the user can choose between two different representations, the Structure Thumbnails (see figure 5.3(c)) and the Literature Fingerprinting technique (see figure 5.3(b)). Depending on the type of analysis task, the size of the document, and the available information about the physical and logical document structure (see section 5.3.1 for an explanation of the two techniques) either one of them is better suitable.

### 5.3.3 The detail view

In the detail view, each sentence is displayed separately (see figure 5.3(c)). The background color of a sentence is set to its overall readability score. Alternatively, the user can choose to have only one of the features displayed. Next to each sentence, the values for each feature are shown separately permitting to investigate the reasons why a sentence was classified as difficult. For this step, the color scales of figure 5.2 are used, meaning



| | | Voc. Difficulty | Word Length | Nominal Forms | Sent. Length | Compl. Sent. Struc. |
|---|---|---|---|---|---|---|
| (a) | The intention of TileBars [9] is to provide a compact but yet meaningful representation of Information Retrieval results, whereas the FeatureLens technique, presented in [5], was designed to explore interesting text patterns which are suggested by the system, find meaningful co-occurrences of them, and identify their temporal evolution. | 🟦 | 🟫 | 🟦 | 🟥 | 🟥 |
| (b) | This includes aspects like ensuring contextual coherency, avoiding unknown vocabulary and difficult grammatical structures. | 🟫 | 🟥 | | 🟦 | 🟦 |

*Figure 5.5: Two example sentences whose overall readability score is about the same. The detail view reveals the different reasons why the sentences are difficult to read.*

that colors are assigned relative to the values that were observed for the very easy and very difficult text samples in the ground-truth dataset. Hovering over one of the cells, triggers the highlighting of the parts of the sentence that contribute to the feature value in the sentence. For example, for the feature *Vocabulary Difficulty* all the words that were classified as difficult are underlined. Additionally, the sentences of a section can be sorted according to the readability score or one of the features. This is very helpful if the user's task is to increase the readability of the document, because sentences that are most in need of being revised are presented first. To help the user to locate a sentence within the section after resorting, the position of the sentence within the document is highlighted with a blue border in the navigation panel as soon as the user hovers over a specific sentence (see figure 5.3 (c)).

## 5.4 Application: Revision and analysis of document (corpora) with respect to readability

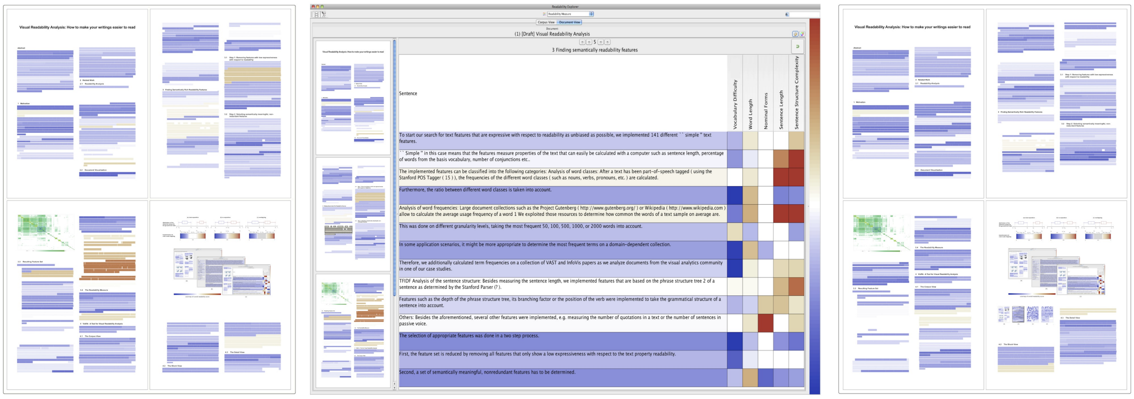
In the following, several case studies are presented that show the wide range of applicability of our tool.

### 5.4.1 Advantage of detailed insight over a single score

Figure 5.5 shows two example sentences whose overall readability score is about the same. Only the detail view reveals that there are different reasons why the sentences are difficult to read. In figure 5.5(a), our tool detects a complex sentence structure whereas in figure 5.5(b) the high percentage of gerunds (verbs acting as nouns) is complicating the sentence. This exemplifies that the details that our tool provides are a clear benefit in the refinement process.

### 5.4.2 Revision of a paper

We also used the tool to revise one of our own papers [100]. Figure 5.6(a) shows the structure thumbnails of the first four pages of the paper. The physical and logical structure of the paper was automatically extracted using the technique described in [129]. Lines with meta-data, such as the names of the authors, their affiliations, keywords, etc., are

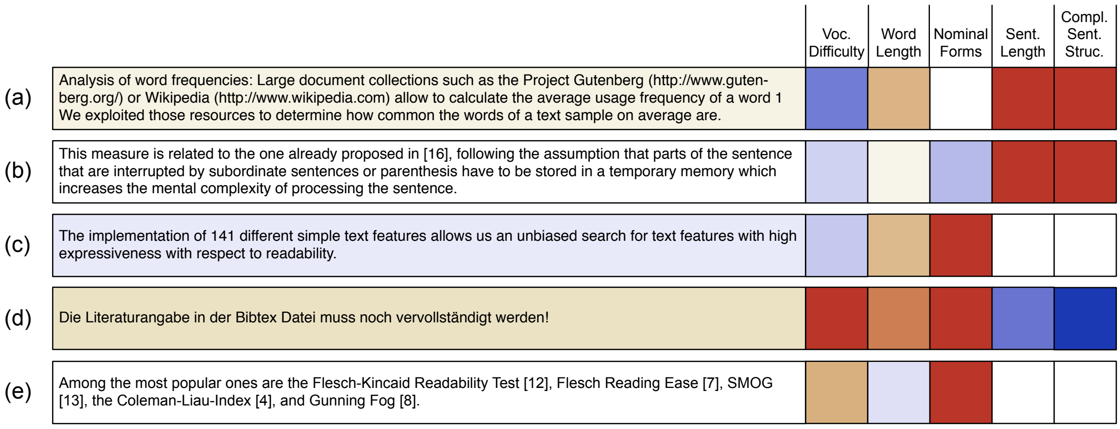


Visual Readability Analysis: How to make your writings easier to read
Readability Explorer
Readability Measure
Corpus View
Document View
Document:
(1) [Draft] Visual Readability Analysis
5
3 Finding semantically readability features

| Sentence | Vocabulary Difficulty | Word Length | Nominal Forms | Sentence Length | Sentence Structure Complexity |
|---|---|---|---|---|---|
| To start our search for text features that are expressive with respect to readability as unbiased as possible, we implemented 141 different `` simple " text features. | 🟦 | 🟫 | ⬜ | ⬜ | 🟫 |
| `` Simple " in this case means that the features measure properties of the text that can easily be calculated with a computer such as sentence length, percentage of words from the basis vocabulary, number of conjunctions etc.. | 🟦 | ⬜ | ⬜ | 🟧 | 🟥 |
| The implemented features can be classified into the following categories: Analysis of word classes: After a text has been part-of-speech tagged ( using the Stanford POS Tagger ( 15 ) ), the frequencies of the different word classes ( such as nouns, verbs, pronouns, etc. ) are calculated. | ⬜ | 🟫 | ⬜ | 🟥 | 🟥 |
| Furthermore, the ratio between different word classes is taken into account. | 🟦 | 🟫 | ⬜ | 🟦 | 🟦 |
| Analysis of word frequencies: Large document collections such as the Project Gutenberg ( http://www.gutenberg.org/ ) or Wikipedia ( http://www.wikipedia.com ) allow to calculate the average usage frequency of a word 1 We exploited those resources to determine how common the words of a text sample on average are. | 🟦 | 🟫 | ⬜ | 🟥 | 🟥 |
| This was done on different granularity levels, taking the most frequent 50, 100, 500, 1000, or 2000 words into account. | 🟫 | 🟦 | ⬜ | ⬜ | 🟦 |
| In some application scenarios, it might be more appropriate to determine the most frequent terms on a domain-dependent collection. | 🟦 | 🟫 | 🟦 | ⬜ | ⬜ |
| Therefore, we additionally calculated term frequencies on a collection of VAST and InfoVis papers as we analyze documents from the visual analytics community in one of our case studies. | 🟦 | ⬜ | ⬜ | 🟫 | 🟫 |
| TFIDF Analysis of the sentence structure: Besides measuring the sentence length, we implemented features that are based on the phrase structure tree 2 of a sentence as determined by the Stanford Parser (? ). | ⬜ | ⬜ | ⬜ | 🟫 | 🟧 |
| Features such as the depth of the phrase structure tree, its branching factor or the position of the verb were implemented to take the grammatical structure of a sentence into account. | 🟦 | 🟦 | 🟫 | 🟫 | 🟫 |
| Others: Besides the aforementioned, several other features were implemented, e.g. measuring the number of quotations in a text or the number of sentences in passive voice. | ⬜ | 🟫 | 🟥 | ⬜ | 🟫 |
| The selection of appropriate features was done in a two step process. | 🟦 | 🟦 | ⬜ | 🟦 | 🟦 |
| First, the feature set is reduced by removing all features that only show a low expressiveness with respect to the text property readability. | 🟦 | ⬜ | ⬜ | ⬜ | ⬜ |
| Second, a set of semantically meaningful, nonredundant features has to be determined. | 🟦 | 🟫 | 🟦 | 🟦 | 🟦 |

(a) (b) (c)

*Figure 5.6: Revision of one of our own papers. (a) The first four pages of the paper as structure thumbnails before the revision. (b) Detail view for one of the sections. (c) Structure thumbnails of the same pages after the revision.*



| | | Voc. Difficulty | Word Length | Nominal Forms | Sent. Length | Compl. Sent. Struc. |
|---|---|---|---|---|---|---|
| (a) | Analysis of word frequencies: Large document collections such as the Project Gutenberg (http://www.gutenberg.org/) or Wikipedia (http://www.wikipedia.com) allow to calculate the average usage frequency of a word 1 We exploited those resources to determine how common the words of a text sample on average are. | 🟦 | 🟫 | ⬜ | 🟥 | 🟥 |
| (b) | This measure is related to the one already proposed in [16], following the assumption that parts of the sentence that are interrupted by subordinate sentences or parenthesis have to be stored in a temporary memory which increases the mental complexity of processing the sentence. | 🟦 | ⬜ | 🟦 | 🟥 | 🟥 |
| (c) | The implementation of 141 different simple text features allows us an unbiased search for text features with high expressiveness with respect to readability. | 🟦 | 🟫 | 🟥 | ⬜ | ⬜ |
| (d) | Die Literaturangabe in der Bibtex Datei muss noch vervollständigt werden! | 🟥 | 🟧 | 🟥 | 🟦 | 🟦 |
| (e) | Among the most popular ones are the Flesch-Kincaid Readability Test [12], Flesch Reading Ease [7], SMOG [13], the Coleman-Liau-Index [4], and Gunning Fog [8]. | 🟫 | 🟦 | 🟥 | ⬜ | ⬜ |

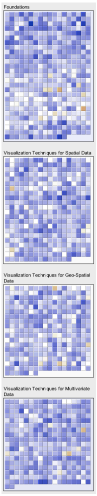
*Figure 5.7: Examples for different reasons of difficulties that were found while revising our own paper with the VisRA tool. The detailed view reveals for each sentence what causes the difficulty. (a) A forgotten period. (b) Long and complex sentence structure. (c) Large number of nominal forms. (d) German comment that we forgot to delete. (e) Many terms that are uncommon in the VAST community.*

automatically filtered out. Section titles are presented in the flow of the document but are excluded from the analysis. The remaining sentences are colored according to their overall readability score. As can be seen, the readability of the paper is already quite good, but some passages clearly need a revision. Figure 5.6(b) shows section 3 of the paper in the detail view. The fifth sentence from the top seems to need some revision as it is colored in red (for an enlarged version see figure 5.7(a)). We find out that the difficulty of the sentence is primarily caused by the fact that we forgot to set a period after the inserted footnote. By hovering over the sentence, it is highlighted in blue in the navigation panel at the left, which makes it easier to find it in the paper.

Figure 5.7 (b)-(e) show some more examples for problems that can be found with the tool. (b) In this case, the sentence was too long and its structure too complex. We split it into several separate ones and dissolved the nested sentences. (c) The main difficulty of this sentence was that we had nominalized several verbs and adjectives. We reformulated the sentence in such a way that wherever possible the verb and adjective forms were used. Although this lengthens the sentence, it can be processed easier by the brain, because fewer words need to be transformed back into their original form [14]. (d) We found a comment in German that we forgot to delete. (e) Interestingly, only a few sentences could be found that are difficult with respect to the used vocabulary in previous VAST proceedings[4]. This confirms that the VAST conference is the proper venue at which to present our research. In addition to pointing us to some sentences in German (sentences registered as using uncommon words compared to the previous VAST papers), one of the sentences in the related work section was highlighted. Since the average VAST paper does not talk about readability measures, it cannot be expected that the terms used are known by the respective community, which means that they should be introduced properly. Figure 5.6(c) shows the first four pages of the paper after the revision.

### 5.4.3 Revision of a large document

When revising a large document such as a book, our thumbnail representation would not be scalable enough. Consequently, several visualization techniques can be chosen on every level of the tool, depending on the size of the document and the availability of information about its logical and physical structure. The figure at the right shows a screenshot of four chapters of a new book on data visualization like it is shown in the navigation panel. A total of about 170 pages are displayed, whereby each of the pixels represents one sentence of the book. It is easy to see that the book is very well written with respect to readability. Only a few sentences stand out as being difficult to read. Further investigation revealed that some of those sentences talk about an application domain to which the introduced visualization was applied. Our vocabulary difficulty feature registers this as an accumulation of many words that are uncommon in the visualization community. Additionally, the tool revealed some long sentences that might have better been split into two sentences.

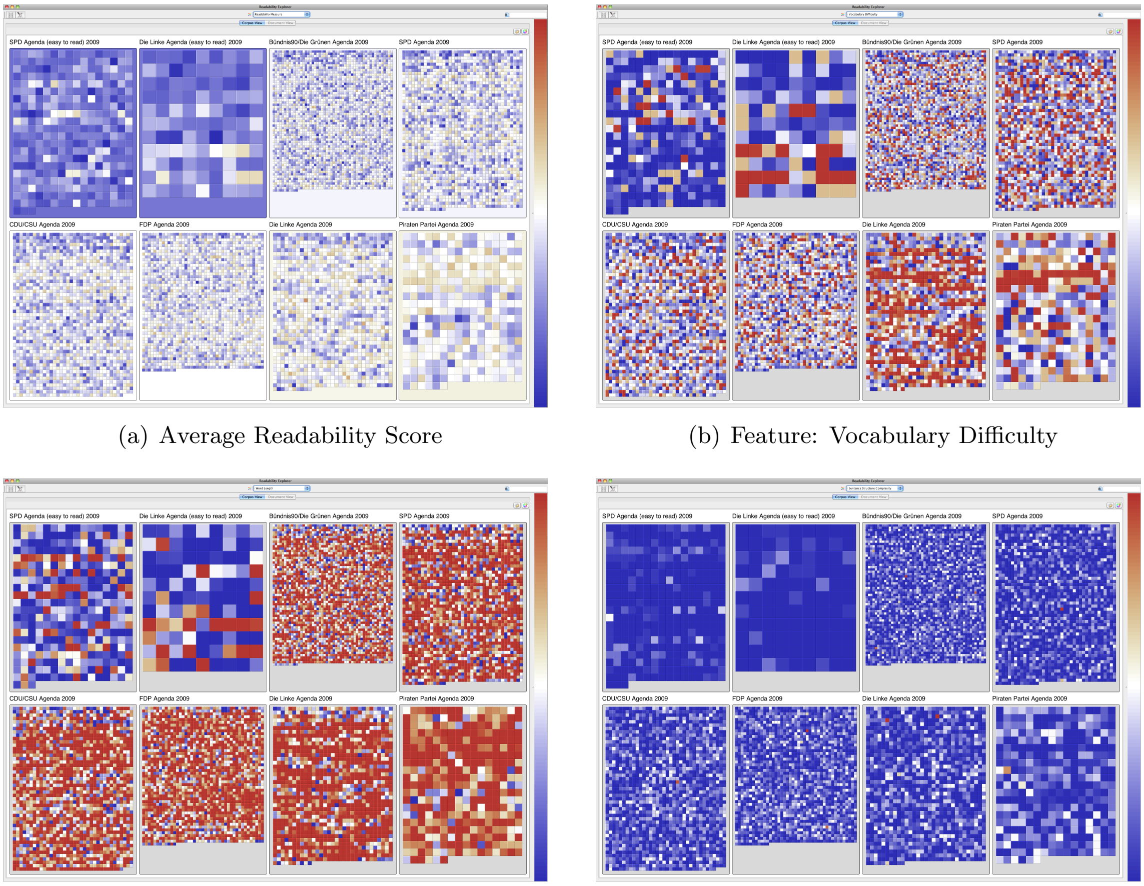

Foundations
Visualization Techniques for Spatial Data
Visualization Techniques for Geo-Spatial Data
Visualization Techniques for Multivariate Data

### 5.4.4 Analyzing a corpus with election agendas

The VisRA tool cannot only be used for refining single documents, but also for a comparative analysis of several documents with respect to the different aspects of readability. Figure 5.8 shows eight election agendas from the elections of the German parliament in 2009. As an embedded visualization, we chose the Literature Fingerprinting technique on sentence level. This allows us to display the large data set on one screen, while still providing the necessary details.

In figure 5.8(a) the average readability score is mapped to color. It can easily be seen that two of the election agendas are significantly shorter and easier to read than the rest of the documents (first two documents in the first row). Those are special versions that are provided by the parties *SPD* and *Die Linke* for people that are less proficient in reading.

[4] VAST is short for *IEEE Conference on Visual Analytics Science and Technology*, the conference that the paper was submitted to


Readability Explorer
SPD Agenda (easy to read) 2009
Die Linke Agenda (easy to read) 2009
Bündnis90/Die Grünen Agenda 2009
SPD Agenda 2009
CDU/CSU Agenda 2009
FDP Agenda 2009
Die Linke Agenda 2009
Piraten Partei Agenda 2009

(a) Average Readability Score

(b) Feature: Vocabulary Difficulty

(c) Feature: Word Length

(d) Feature: Sentence Structure Complexity

*Figure 5.8: Visual Analysis of eight election agendas from the elections of the German parliament in 2009.*

Interestingly, the normal election agenda of *Die Linke* (third one in the last row) is the second most difficult one.

At first, we were surprised to see that this agenda is rated as comparably difficult to read, since the target group of *Die Linke* is traditionally the working class. A more detailed analysis with respect to the different aspects of readability revealed some of the reasons for this. Figure 5.8(b) shows how the sentences are rated with respect to the vocabulary difficulty. To determine if a word is common, the dictionary of the University of Leipzig is employed. Frequencies in this dictionary are based on a large corpus of news articles. Closer analysis of the election agenda of *Die Linke* revealed that a high number of socialistic terms were used in the text. This terminology is not common in German newspapers. As mentioned earlier, two election agendas were intended to be easy to read. Strikingly, these two agendas also contain difficult vocabulary. The detail view reveals that in those documents long words are broken up by inserting a dash ("-"). These words are most often compound words and characteristic to the German language (e.g. in genitive constructions). They are often broken up by dashes or hyphens in order to allow for better comprehension. However, these words cannot be found in the list of most frequent terms (since they are spelled differently now from the words provided in the vocabulary list) and thus they are classified by the algorithm as uncommon. Long words are avoided at all costs in the special election agendas that are written in a easy to read language. This fact is reflected by the visualization of the average word length that is depicted in figure 5.8(c). It also explains the significant differences between the easy-to-read election agendas and the more difficult ones. Finally, figure 5.8(d) displays the feature sentence structure complexity. Obviously, all election agendas are well-formulated with respect to this property. Only single sentences are highlighted for which a revision might have been advisable.

## 5.5 Summary and future work

In this chapter, we introduced a tool for visual readability analysis that supports the writer in refining a document, and thereby to increase its readability. Special consideration was given to the selection of features that are non-redundant and semantically understandable. This is reflected in the design of the tool that provides insight into the data at several levels of detail. At the highest resolution, for every single sentence the values of the different features are displayed instead of only visualizing the average score. Several different overview representations account for differences in the size of the documents and the knowledge about the physical and logical structure of the document.

With the semi-automatic feature engineering approach that is presented above, we could identify features that are expressive predictors of readability. By clustering the feature vectors for the ground-truth data set, we could filter out correlated features. Furthermore, this permitted to detect different aspects of readability and cover them in the final measure. By reviewing the detected clusters manually, we could ensure that the selected features are semantically understandable by the user.

The approach that was described above is based on two assumptions: First, we assume that features that discriminate well between easy and difficult to read *paragraphs* will also be able to discriminate easy and difficult to read *sentences*. Experiences in other fields of document analysis (e.g. authorship attribution) suggest that this is not necessarily true. Second, by only working with very difficult and very easy to read documents in the feature engineering step, we implicitly assume that it is possible to linearly interpolate the feature values between those two extremes. However, theoretically it is possible that a feature is well able to discriminate easy and difficult features but does not distinguish the ones with an average readability value from the difficult ones. Figure 5.8(c) suggests that this might be the case for the word length feature. Verifying the two assumptions and if necessary adapt the approach might further improve the results.

Furthermore, the measures could be improved by taking combinations of features into account to measure an aspect of readability. Rudimentarily, this was already done for the nominal forms feature. Instead of a combination, it could also be valuable to select several features per aspect and automatically choose the best one depending on another text property (e.g. the length of the sentence). Similarly, a more advanced combination of the different measures to one overall readability score might be applied.

From an application perspective, it would be interesting to approximate additional quasi-semantic properties. For example, it might be helpful to include features that measure how appropriate the writing style of a document is or how well it is structured. Both measures are dependent on the domain or on the community, for which the document is written. Additionally, they would be asking for a calculation that compares the document to others in the same context. Furthermore, it would be valuable to take measures into account that work on the discourse level and assess the consistency of the text.

Since the tool is build in a way that any set of features can be displayed, incorporating new features is easy. This invites using the visual interface in different scenarios. Basically, the technique would be useful for any application in which a detailed analysis of a document with respect to several features in parallel is required. Another advantage is the possibility to adapt the overview representation to the specific type of document, the analysis task, and the meta-information that is available.

There is also improvement potential in the visual representation. So far, the block view does only display the overall readability score for a paragraph. Additional information such as the distribution of the values in the next lower level might be beneficial. And finally, we envision to enhance the tool with a natural language generation component that is able to provide a written summary of the results. Although, this can be considered as challenging in the general case, the restricted domain with respect to what could have to be said should make it feasible in this application scenario.

# 6 Quasi-semantic Property III: Discriminating and Overlap Terms

## Contents

Term extraction is the task of automatically extracting terms from a document (collection) that are considered as interesting or central for the text. Many text analysis techniques rely on such terms, such as document retrieval, document clustering, summarization, and text mining, but also approaches in the context of machine translation, thesaurus construction, or knowledge organization.

The definition of *interestingness* may differ from domain to domain. Typical examples include approaches that extract terms which are very frequent in a document (collection), terms that are much more frequent in a reference corpus, or terms that are not part of everyday speech but are considered as technical terms. In some cases, classes (or clusters) of documents can be distinguished and topical differences and similarities among those classes are of interest.

In this chapter, an approach is introduced that helps analyzing a set of classes of documents with respect to the question what discriminates one class of documents from the rest. In addition to those discriminating terms, the technique also determines so-called overlap terms that discriminate a subset of the classes from the remaining ones (section 6.2). A detailed analysis and evaluation of the algorithm and the properties of the extracted terms is presented in section 6.3. The algorithm is applied to proceedings of several conferences in section 6.4 and is used to extract product attributes from customer reviews in the next chapter. Finally, a summary is given and future work is discussed.

## 6.1 Research and application context

Our research in the area of term extraction was driven by two application scenarios: First, the need to extract frequently commented on product attributes from customer reviews and second the comparison of different document corpora. Both cases have in common that we are interested in the terms that discriminate two or more sets of documents from each other. The latter application scenario differs from the first one in the way that not only the terms that discriminate one corpus from another one are important but also the terms that discriminate two or more document collections together from one or several others (overlap terms).

### Analysis tasks and quasi-semantic questions

In the context of opinion analysis, the attributes that are frequently commented on by customers are needed to answer the question *what* particularly the people liked or disliked about a product. With attributes we refer to certain characteristics of the entity of interest which are frequently mentioned when this entity is evaluated. For a product those attributes may be components, properties or features that are important for customers when evaluating it. Having such attributes allows a detailed analysis of the opinions that are expressed in the text instead of only looking at the general sentiment.
Thus, the related analysis question in this case is: *"Which attributes do customers frequently comment on in customer reviews?"* and consequently the related quasi-semantic property would be the *product attribute*.

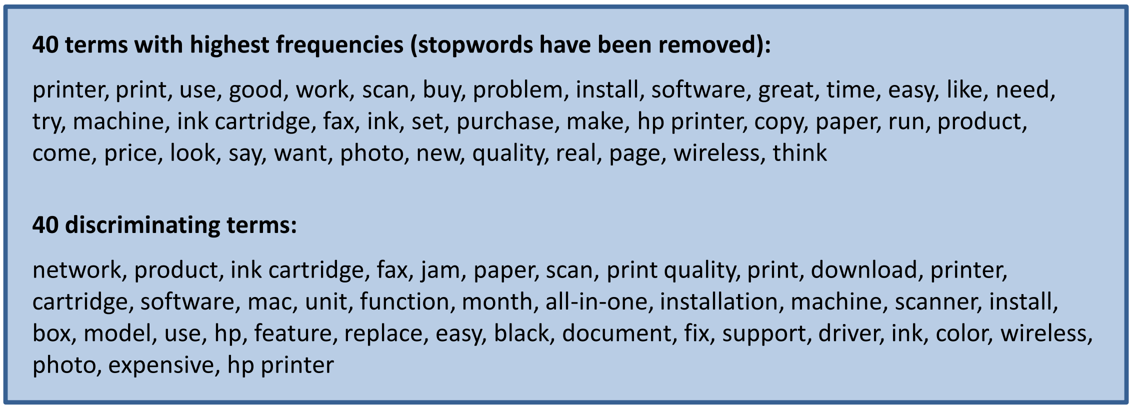
In the context of customer feedback analysis, in most cases no more than 30 attributes are needed. But still, there are reasons for using an automatic algorithm. First, for an unknown product the analyst might not know which components or features of the product are important. The algorithm provides a set of attributes that were frequently commented on, which means that a significant number of people that already used the product think that those attributes are worth to be taken into account. Secondly, even if the analyst knows what he is looking for, he might not know which terms are commonly used by the community the reviews were written by. When looking for a monitor, do you have to search for comments on the "size of the monitor", "the screen size" or for "monitor diagonal"? Our algorithm automatically detects the most common terminology that is shared by many reviewers.

In the second application example of comparing different document corpora to each other, we are interested in questions like: *"How does one document collection differ from the others?"* and additionally: *"What do the document collections have in common and what distinguishes them from each other?"*. The related quasi-semantic properties in this case are *discriminating terms* (answering the question what discriminates one document collection from another / several others) and secondly *overlap terms* (to find out what discriminates a set of document collections from one or several others and thus to find out what they have in common).

### Research focus of the chapter

Central to the proposed algorithm is the notion of determining what an interesting term is with the help of a counter-balance class. This idea is not new but is used by other term extraction algorithms as well. What is special in our scenario is that

1. we do not only use the reference class to find interesting terms but the property of



**40 terms with highest frequencies (stopwords have been removed):**

printer, print, use, good, work, scan, buy, problem, install, software, great, time, easy, like, need, try, machine, ink cartridge, fax, ink, set, purchase, make, hp printer, copy, paper, run, product, come, price, look, say, want, photo, new, quality, real, page, wireless, think

**40 discriminating terms:**

network, product, ink cartridge, fax, jam, paper, scan, print quality, print, download, printer, cartridge, software, mac, unit, function, month, all-in-one, installation, machine, scanner, install, box, model, use, hp, feature, replace, easy, black, document, fix, support, driver, ink, color, wireless, photo, expensive, hp printer

*Figure 6.1: 40 most frequent terms (top) compared to the Top-40 discriminating terms. It can easily be seen that the list of discriminating terms is more dense with respect to the question what the customers frequently comment on in customer reviews on printers. The list of the most frequent terms also contains many terms that are typically used in reviews but do not convey the desired information (e.g. need, like, good, etc).*

being discriminative is a central part of our definition of interestingness (hence, this is what we optimize our algorithm for)

2. we are additionally interested in terms that discriminate *multiple* document collections against one or several others.

In this chapter, a novel term extraction method is presented that is especially tailored to these needs. Hence, considering the framework of section 2.1.2, the focus of the chapter is on finding a measure that approximates a given quasi-semantic property. Furthermore, the measure is evaluated in detail. The analysis gives insight into what properties the extracted terms have and leads to a better understanding of the algorithms. A comparison with alternative approaches permits to position the algorithm within the wide landscape of term extraction methods. Advantages and disadvantages of the algorithm are discussed and indicate in which situation applying the extracted terms might be useful. Besides this rather general analysis of the algorithm, it is evaluated how well the extracted terms approximate the quasi-semantic property *product attribute* in the context of customer review analysis.

## 6.2 Automatic extraction of discriminating and overlap terms

A straightforward way to automatically extract attributes out of textual data sources (such as reviews) would be to take the most frequent words and filter out stop words according to a given stop word list. Using printer reviews from amazon.com, this results in the list of the 40 most frequent terms that is shown in the upper part of figure 6.1. The problem that comes along with this approach is that not only words describing product attributes like “print” or “software” are frequent but also typical review terms like “great”, “like” or “need”. Widely used stop word lists contain only very general terms like conjunctions, determiners, pronouns etc. and thus are not suitable to separate the printer terms from the rest. We have to apply a special term filtering that extracts the printer terms while it

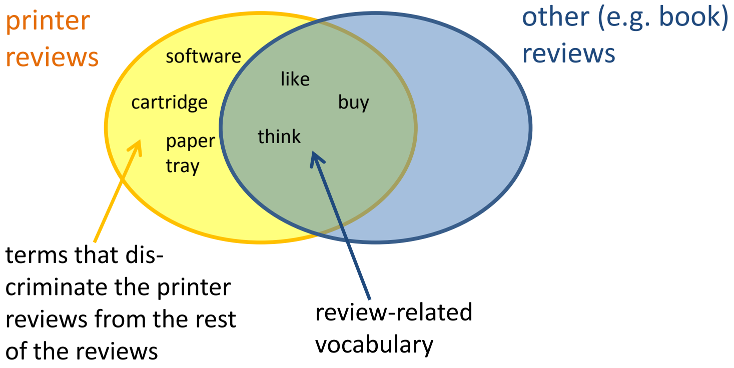

printer reviews
other (e.g. book) reviews
software
cartridge
paper tray
like
buy
think
terms that discriminate the printer reviews from the rest of the reviews
review-related vocabulary

*Figure 6.2: To extract product attributes, we make use of the fact that the review-related vocabulary is used by all kind of reviews, whereas the attributes that we are looking for are predominantly found in the class of printer reviews.*

does not consider the review terms. For this purpose, we developed a novel discrimination-based term extraction method: We consider the set of reviews that we are interested in (e.g. printer reviews) to be a special class of text documents ("printer review class") and compare it to a set of reviews from amazon.com (e.g. book reviews) which we consider to be the counter-balance class ("book review class"). Now, the aim is to find the terms that are much more important within the class of printer reviews than within the class of book reviews and at the same time characteristic for the class. We make use of the fact that both classes share the review-related vocabulary and extract the terms that discriminate the printer review class against the counter-balance class of book reviews. Figure 6.2 illustrates the basic concept. By our definition a term discriminates one class from another if it is much more important within this class than within the other one. Furthermore, we require discriminating terms to be characteristic for the class, i.e. frequent and well-distributed across the documents. In order to measure the importance of terms for a class, we weight terms according to a novel extension of the TFIDF-measure, our "Term Frequency Inverse Class Frequency" (TFICF). We determine then the set of terms that discriminate the printer review class against the counter-balance class considering the TFICF term scores. Note that sometimes it can be helpful to use more than one class as a counter-balance class. This is the case when there are several undesired aspects to be filtered out and there is no single counter-balance class that contains all of those aspects. Furthermore, there has to be more than one counter-balance class in application contexts in which we are interested in the overlap between two or more classes. This is the case in our second application scenario in which we use the technique to find out what different conferences have in common and how they differ from each other (see section 6.4).

### 6.2.1 Term frequency inverse class frequency (TFICF) - Our importance measure

In order to find terms that discriminate one class of documents from one or several others, we need a measure that calculates the importance of this term for the class in comparison to the other classes. The most popular approach for term scoring, TFIDF (term frequency inverse document frequency) [115, 124], is not suitable in this case. This is due to the fact that the TFIDF value determines an importance value for a certain term with respect to a document within a document collection. What we need is an importance value for a certain term with respect to a document class. We therefore introduce the TFICF, which is an extension of the classic TFIDF measure. The formula for TFICF is composed of two factors: a term frequency value (TF) and an inverse class frequency value (ICF) (see equation 6.1).

Let $C$ be the set of classes $c_j$ with j $\in \{1, \ldots, |C|\}$ ($|C|$ denotes the number of classes in $C$), $D_j$ is the set of documents $d_{jk}$ in class $c_j$ with k $\in \{1, \ldots, |D_j|\}$, and $t$ is a term.

$$TFICF(t, c_j) = TF(t, c_j) \cdot ICF(t) \tag{6.1}$$

The TF value reflects the relative frequency of a term within a class as in the TFIDF measure. It is calculated by dividing the overall frequency of a term among the documents of a collection $c$ by the overall number of tokens in the collection (see equation 6.2).

$$TF(t, c_j) = \frac{\sum_{k=1}^{|D_j|} \mathit{freq}(t, d_{jk})}{\sum_{k=1}^{|D_j|} |d_{jk}|} \tag{6.2}$$

where $|D_j|$ denotes the number of documents in class $c_j$, $|d_{jk}|$ denotes the number of tokens in document $d_{jk}$, and $\mathit{freq}(t, d_{jk})$ is the frequency of the term t in document $d_{jk}$.

The ICF value takes into account in how many classes the term is present. In contrast to the standard IDF formula, our ICF formula has to operate on multiple classes of documents instead of a single class. A straightforward application of the IDF formula would be to say that a term $t$ is an element of a class $c$, if it occurs in at least one of the corresponding documents. However, that means that outlier documents (whose content is untypical for the collection) get a high influence on the result. We therefore propose to define that term $t$ is only considered element of a class $c$ if at least $X$ percent of the documents $D$ (where $X$ is a user-defined parameter) contain the term (see equation 6.3). Alternatively, the percentages of documents in each class that contain the term may be summed up (see equation 6.4). In section 6.3 we compare our method to a similar one that does not take the distribution of the terms across the class into account. The experiments show that considering the distribution across the documents is especially important in the scenario of comparing different paper collections to each other and less important for the extraction of product attributes (see section 6.3 for details).

$$ICF_1(t) = log(\frac{|C|}{|\{c \in C : \frac{|\{d \in c : t \in d\}|}{|\{d \in c\}|} > X\}|}) \tag{6.3}$$

$$ICF_2(t) = log(\frac{|C|}{\sum_{j=1}^{|C|} \frac{|\{d \in c_j : t \in d\}|}{|\{d \in c_j\}|}}) \tag{6.4}$$

Additionally, the distribution of the terms across each class could be measured more explicitly by weighting the TF-values with a factor that measures this distribution. To get terms that are well spread across the different documents e.g. the inverse standard deviation of the frequency of the terms in the class or the $\chi^2$ significance value might be used (both are suggested in [36] as term weights). Another example for such a distribution weight is the term relevance weight which was defined by Salton & Buckley [90] as *"the proportion of relevant documents in which a term occurs divided by the proportion of non-relevant items in which the term occurs"*. (The notion of being "relevant" or "irrelevant" would be easy to specify in our case, as we could take the documents in the class that we want to extract the terms for as relevant and all the others as irrelevant.) However, our tests showed that this does not lead to further improvement in our application scenarios. Only in a situation in which we are specifically searching for terms that are unequally distributed, the additional parameter proved useful. It is therefore not further detailed here.

### 6.2.2 Determining discriminating and overlap terms

The TFICF measure provides a term weight that is comparable among several classes. For each term, we get one value per class that allows us to compare the importance of the term in the two classes. We now define that a term is discriminating for one of these classes if its score is significantly higher for this class than its scores for the other classes. To determine the discriminating terms for a class, we use a threshold called discrimination factor by which a score for one class must outnumber the scores of all other classes (see definition 6.1).

**Definition 6.1 (Discriminating terms)**

> *A term $t$ is discriminating for a single class $C_k$ if:*
> $\forall i \in \{1 \dots n\} \backslash k$ :
> *TFICF*$(t, c_k) >$ *discrimination-factor* $\cdot$ *TFICF*$(t, c_i)$.

Besides using the technique to extract terms that discriminate *one* class from the others, the approach can also be employed to compare a set of document classes by means of discriminating and overlap terms. A term is considered as being discriminating for one class against the other classes if it is characteristic for the class *and* serves as a distinguishing feature between the classes. Technically speaking this is the case if the equation of definition 6.1 holds. The same approach can be applied to determine if a term is discriminating for the overlap of several classes. This is precisely the case if the lowest term score for one of the overlap classes outnumbers the highest term score of the remaining classes at least by the threshold factor (see definition 6.2).

**Definition 6.2 (Overlap Terms)**

> *For the overlap area of several classes* $\{C_k, C_l, \cdots, C_m\}$ *a term $t$ is discriminating if:*
> $\forall i \in \{1 \cdots n\} \setminus \{k, l, \cdots, m\}$:
> *min(weighted_tficf*$(t, C_k)$, *weighted_tficf*$(t, C_l)$, $\cdots$, *weighted_tficf*$(t, C_m)$)
> $>$ *discrimination-factor* $\cdot$ *weighted_tficf*$(t, C_i)$.

Note that our approach extracts only terms that are characteristic *and* discriminating for a class. This means that we do not necessarily get all the terms that could be considered as keywords of the document collection but only the ones that additionally discriminate the specific class from the other documents in the corpus.

### 6.2.3 Preprocessing and parameter tuning

#### Preprocessing

Like in many text mining applications careful preprocessing is valuable. In our case we applied a base form reduction algorithm [81] to all words in order to get singular forms for nouns and infinitive forms for verbs. In addition, we used a sentence splitter and POS-tagger ([136], [135], [126]) and an NP-chunker ([108], [47]) to identify nouns respectively noun phrases. This allows us to focus only on nouns and noun phrases if this is desired. Numbers and short strings with less than 3 characters were deleted in the preprocessing step, since they often correspond to punctuation marks or special characters that do not need to be considered.

One interesting advantage of our method is that we do NOT use any stopword lists. Frequent stopwords like “the” or “and” are automatically ignored with very high probability, because their ICF values become 0. Stopwords with a lower frequency in a regular case should not appear considerably more often in one class than in the others and thus are filtered out.

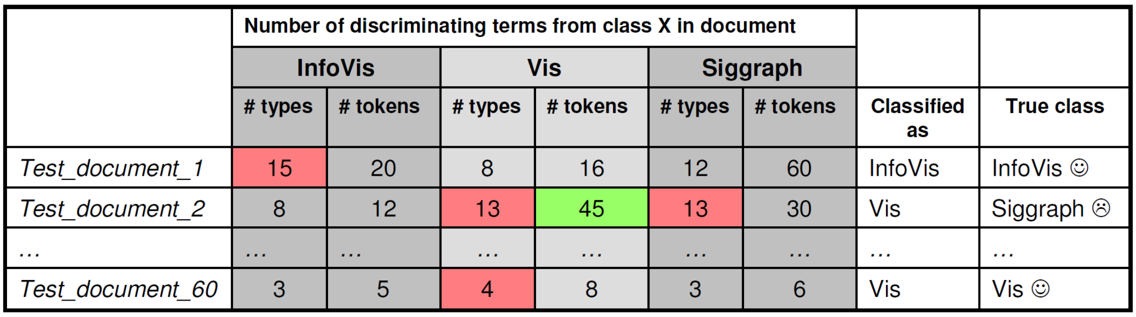
#### Parameter Tuning

Our algorithm for determining the discriminating and overlap terms has two parameters: A minimum percentage and the discrimination factor. The minimum percentage is used to specify the minimum number of documents of a class that must contain the term to allow it to be chosen as discriminative. Without that parameter all terms that only occur in one class would most certainly be considered as being discriminative (no matter how often they occur in the class, because $X > 0 * factor$ would always be true). The minimum percentage can easily be set by the user (e.g. 0.2 if at least 20% of the documents shall contain a term). In contrast to this, the discrimination factor threshold is not an intuitive parameter. However, our experiments showed that reasonable thresholds lie typically in the interval between 1.5 and 5.0 and that the result is quite robust to changes of this factor. In our implementation, the exact threshold and minimum percentage is set by using a dynamic slider, which allows the user to get the desired amount of discriminating terms.

## 6.3 Evaluation of the QSP “Discriminating terms”

Central to our approach is the extraction of terms that discriminate one class from another. To evaluate our method, we therefore analyzed how well the terms extracted from a real dataset are able to discriminate documents of one class from several others. Furthermore, we tried to learn more about the characteristics of the extracted terms by looking at the frequency that those terms have in their classes and analyzing how well they are spread across the different documents of a class.

However, knowing that the extracted terms are well able to discriminate different classes does not necessarily mean that they are a good approximation for the quasi-semantic property *product attribute* as we defined it in section 6.1. We therefore secondly analyzed the usefulness of the extracted terms under the assumption that they are to be used as attributes when analyzing customer comments.In both evaluation scenarios, we



| | Number of discriminating terms from class X in document | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | InfoVis | | Vis | | Siggraph | | | |
| | # types | # tokens | # types | # tokens | # types | # tokens | Classified as | True class |
| *Test_document_1* | 15 | 20 | 8 | 16 | 12 | 60 | InfoVis | InfoVis ☺ |
| *Test_document_2* | 8 | 12 | 13 | 45 | 13 | 30 | Vis | Siggraph ☹ |
| … | … | … | … | … | … | … | … | … |
| *Test_document_60* | 3 | 5 | 4 | 8 | 3 | 6 | Vis | Vis ☺ |

*Figure 6.3: This table exemplifies the classification process. For each conference 15 terms were extracted from the 100 training documents. These terms are then used to classify 60 previously unknown test documents. For each document, we count how many of the terms can be found at least once in the document (columns "# types"). The document is then assigned to the class that it shares most terms with. If no unambiguous decision can be made using this number (like for Test_document_2 in the example), multiple occurrences of the terms are counted as well (columns "# tokens") to decide which class the document belongs to. We can then compare the classification results to the given ground-truth and build the confusion matrices that are shown in figure 6.4 for each term extraction method.*

compared our approach to several alternative approaches for term extraction.

Finally, the sensitivity of the method with respect to the size of the collection and the choice of the counter-balance class is evaluated.

### 6.3.1 Evaluation of the discrimination power of the extracted terms

To evaluate how well the extracted terms are able to discriminate one class of documents from the others, we used the extracted terms in a classification task. The classes were the three conferences InfoVis (Information Visualization), Siggraph (Computer Graphics), and Vis (Scientific Visualization). Each class was represented by 100 recent papers of the conference. For each of the three document collections, we used 4 different methods to extract (in average) 15 terms per class (the different methods are described in detail below). The extracted terms were then used to classify a set of 60 test documents (20 of each class) that were different from the training set. Each of the 60 documents was assigned to the class that it shared most discriminating terms with (counting each extracted term at most once). If there was more than one winning class, the document was assigned to the class that contained more discriminating terms taking multiple occurrences of the terms into account as well. If the document still could not be assigned unambiguously, it was assigned to the class of ambiguous documents. (Figure 6.3 exemplifies the classification process.) In this classification task a method performs best if it extracts terms that discriminate a class from the others but yet also chooses terms that are characteristic for the class they have been extracted for (i.e. that they are shared by many documents of the specific class instead of being only significant for a small subset of documents of the class).

### Employed term extraction methods

We used the following four methods for term extraction:

- **TFIDF average:** Given the training corpus of 300 documents, for each document and each term in the corpus a TFIDF (Term Frequency Inverted Document Frequency) value was calculated.[1] The TFICF measure that is proposed in this chapter is an extension of the famous TFIDF measure. It is therefore reasonable to compare our approach to this existing and well-established measure. While the calculation of the term frequency (TF) is the same for both methods (see equation 6.2), the TFIDF multiplies it with the inverted *document* frequency (IDF). We used the following formula in the evaluation to calculate the IDF value:

  $$idf(t) = log(\frac{|D|}{|\{d \in D : t \in d\}|})$$

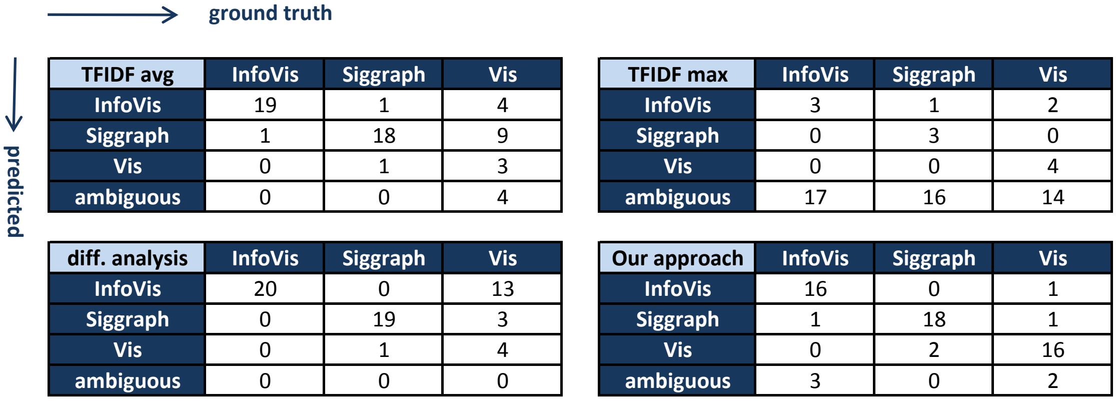
  with $D$ being the set of all documents (no matter which class they belong to),
  $d =$ a specific document of the collection,
  $t =$ a specific term.
  $|\ |$ is used to refer to the number of elements in a set.

  After calculating the TFIDF values for every term, the documents were sorted into classes and for each class the average TFIDF of each term was calculated. Next, the terms were sorted according to their average value. Finally, for each class the 15 top terms were chosen.

- **TFIDF max:** The second method is very similar to the first one. The only difference is that instead of calculating the average TFIDF value, the maximum TFIDF value of the class is chosen for each term. Then, again the terms are sorted according to their TFIDF values and the 15 top terms for each class were chosen. We included this method, too, since it has been proposed in several other publications ([36], [132]).

- **Differential Analysis:** This technique extracts technical terms from a corpus by comparing for each term the probability of its occurrence in the given corpus to a general reference corpus [155, 60] (this corresponds to what we call a counter-balance class). We used the author's terminology extraction tool (TE) that is part of the ASV Toolbox [6] with its default settings to extract the terms for our experiments. The method is similar to our approach as both search for terms that are more important for the analysis corpus than for a reference corpus. The main difference between the two methods is the measure that is used to determine the importance of a term. The algorithm for Differential Analysis uses a measure that is based on the likelihood-ratio-test whereas our method is based on an extension of the TFIDF. Note that the tool permits to replace the general reference corpus with a user-given one. In our experiments, we replaced the given corpus with the papers of the two conferences that we wanted our terms to discriminate against. This way we could make sure that the differences in the results are not only caused by using a different reference corpus. Additionally, we also did the same tests with the general reference corpus that comes with the tool to compare both outcomes.

[1] An alternative to the described approach would be to calculate the TFIDF values separately for each class instead of using the whole set of 300 documents. Our experiments showed that the results are almost the same (neither better nor worse).


ground truth
predicted

| TFIDF avg | InfoVis | Siggraph | Vis |
|---|---|---|---|
| InfoVis | 19 | 1 | 4 |
| Siggraph | 1 | 18 | 9 |
| Vis | 0 | 1 | 3 |
| ambiguous | 0 | 0 | 4 |

| TFIDF max | InfoVis | Siggraph | Vis |
|---|---|---|---|
| InfoVis | 3 | 1 | 2 |
| Siggraph | 0 | 3 | 0 |
| Vis | 0 | 0 | 4 |
| ambiguous | 17 | 16 | 14 |

| diff. analysis | InfoVis | Siggraph | Vis |
|---|---|---|---|
| InfoVis | 20 | 0 | 13 |
| Siggraph | 0 | 19 | 3 |
| Vis | 0 | 1 | 4 |
| ambiguous | 0 | 0 | 0 |

| Our approach | InfoVis | Siggraph | Vis |
|---|---|---|---|
| InfoVis | 16 | 0 | 1 |
| Siggraph | 1 | 18 | 1 |
| Vis | 0 | 2 | 16 |
| ambiguous | 3 | 0 | 2 |

*Figure 6.4: Confusion matrices for the four different methods classifying 60 documents.*

- **Our approach:** To extract terms with the approach that is introduced above (using $ICF_1$, see equation 6.3), we set the parameter values as follows: The minimum percentage was set to 0.11 (that means that more than 10% of the documents have to contain the term) and the discrimination factor to 2.0. Since our method does not extract a given number of terms but automatically determines the number of terms that well discriminate one class from the others we do not have exactly 15 terms per class but 14 terms for InfoVis, 15 for Vis and 16 for Siggraph.

### The evaluation result

The following accuracy values were calculated for the four methods (accuracy = number of correctly classified documents divided by the total number of documents): TFIDF avg: 0.67 (0.71), TFIDF max: 0.17 (0.77), Differential analysis: 0.72 (0.72), our approach: 0.83 (0.91)[2]. Interestingly, the accuracy values for the Differential analysis were slightly better when using the general corpus instead of the two other conferences as reference corpus: 0.77 (0.78).

Figure 6.4 shows the result in more detail. It can be seen that using the TFIDF max approach almost 80% of the documents could not be classified unambiguously. The results for the other 3 techniques are more meaningful. It can easily be seen in the confusion matrices that all the methods performed well on the classes InfoVis and Siggraph but that TFIDF average and the Differential Analysis (with both reference corpora) had problems with the class Vis. An explanation for that might be that the Vis conference is thematically somehow in between the two other conferences.[3]

Our assumption was that the closer the classes are related to each other, the more important it is that the applied method is able to find terms that are well spread across the different documents of the class and clearly discriminating.

[2] Values in brackets result from ignoring ambiguous documents in the accuracy calculation.

[3] For completeness we also tested if the performance of the TFIDF values is increased when more terms are used. However, our experiments showed that the accuracy becomes even worse. (The test was done with 100 terms and TFIDF avg as well as TFIDF max as extraction methods.)

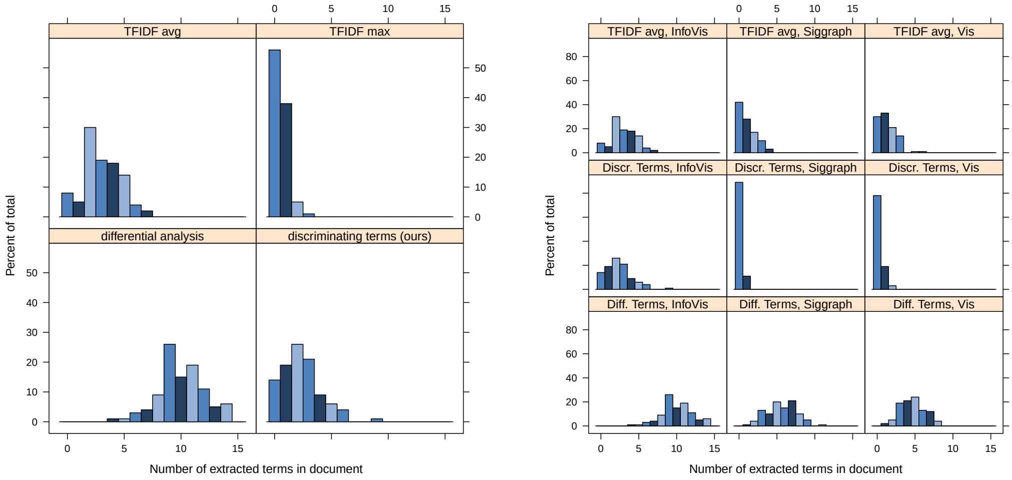


*Figure 6.5: Analysis of the distribution of the terms, comparing the three methods TFIDF avg, TFIDF max and Differential Analysis to our method (Discriminating Terms). Left: Distribution across the documents of the class that the terms were extracted for (InfoVis). The height of each bar in the graphic represents the number of documents in the training corpus that contain k extracted terms (with k being mapped to the x-axis). Right: Distribution across the documents of the class InfoVis compared to the distribution of the documents across the other two classes that the terms were not extracted for.*

## Detailed analysis

In order to get some deeper insight into how the different algorithms select their terms, we conducted a more extensive evaluation, where we also analyzed the distribution of the extracted terms visually. For each document in the training corpus, we counted how many of the terms that were extracted for its class are contained in the document (for the class InfoVis, e.g., this corresponds to column *InfoVis, # types* in figure 6.3). The left graphic of figure 6.5 shows the histogram of these values for the documents of the class InfoVis. The height of each bar in the graphic represents the number of documents in the training corpus that contain exactly $k$ extracted terms. Obviously, the distribution for TFIDF max falls apart. More than 90% of the documents contain only 1 or even 0 of the extracted terms! That means that the method extracts many terms that can only be found in very few documents of the class (which means that they cannot be considered as characteristic for the whole class). The three other methods show distributions that are similar to each other. The right graphic of figure 6.5 reveals the difference between those three methods. This time not only the distribution of the terms across the class that the terms were extracted for has been analyzed (again InfoVis) but also the distribution across the two other classes. As can be seen, our approach (middle row) clearly favors terms that are very infrequent among the documents of the other classes. The distribution of the terms that the Differential Analysis extracted was the one that surprised us most since at first sight the diagrams for the 3 classes look very similar[4]. Closer analysis (by producing a cumulated diagram) revealed that the peak of the distribution of the terms across the

[4] To generate this histogram, the papers of the two conferences were used as a reference corpus. The picture with the general reference corpus looks almost the same.

documents of Siggraph and Vis is slightly shifted to the left. Because our classification task was designed in a way that even one more term was enough to determine the class it belongs to, this slight movement of the distribution helps the method in passing the test.

Figure 6.5 suggests that there is a trade-off between extracting terms that are characteristic for the class (frequent and shared by many documents) and extracting terms that are discriminative in the sense that they are not characteristic for the other classes. The histograms show that Differential Analysis prefers terms that are very characteristic for the class, even if they are only slightly less characteristic for the other classes. In contrast to this, our method optimizes the term extraction with respect to getting the most discriminative terms and accepts that the terms are less characteristic for their own class than they could be. Which method is best highly depends on the application context. For analyzing the differences between conferences (see also section 6.4), a method that focuses on what discriminates the different classes (as our method does) is preferable. If we wanted to find key terms that describe a conference best, Differential Analysis might be the better choice.

### 6.3.2 Evaluation of the usefulness of the terms as product attributes

To evaluate if the extracted terms seem reasonable as product attributes, we conducted a small user study. For the evaluation scenario, the top-40 terms according to frequency were compared to the top-40 terms extracted by our discrimination-based approach. In both cases, we used a set of printer reviews as a corpus (see Figure 6.1). For each of the terms, the participants of the user study had to decide whether it is a printer attribute of which they would want to know if users generally liked or disliked it before buying a particular printer. Those are precisely terms that should be extracted by an automatic method. In order to avoid any bias, the terms extracted by both approaches were merged and the resulting list was ordered alphabetically. Thus, the participants did not know by which method a term was originally extracted. As participants of the user study, five rather experienced printer owners were recruited.

Figure 6.6 depicts the result of the study. On the x-axis the number of users is listed that have voted for an extracted term. The y-axis indicates how many terms were identified by at least x users as useful attributes. A number of 5 users implies that it was an unanimous vote. An interesting outcome of the user study was that users have quite varying preferences on attribute terms. For 31 out of the 40 terms that our method extracted, at least one participant thought that they were useful printer attributes. For the 40 top-frequency terms, only 21 terms were found to be useful by at least one user. In total, our method clearly outperforms the standard frequency-based method by a significant margin (at least 44% more relevant attributes).

Something that we could not assess in our user study is whether all necessary terms were extracted by the algorithm or if there are terms missing that should have been extracted as well.

For completeness table 6.1 additionally shows the terms that were extracted by the two other methods that were evaluated in the previous section. As can be seen, the term list that has been extracted with TFIDF average (using the same method than in the section 6.3.1) contains much noise, but the terms that have been extracted with the Differential Analysis seem quite reasonable. The list contains even less review-related terms (such as “easy” or “expensive”). This suggests that for the task of extracting

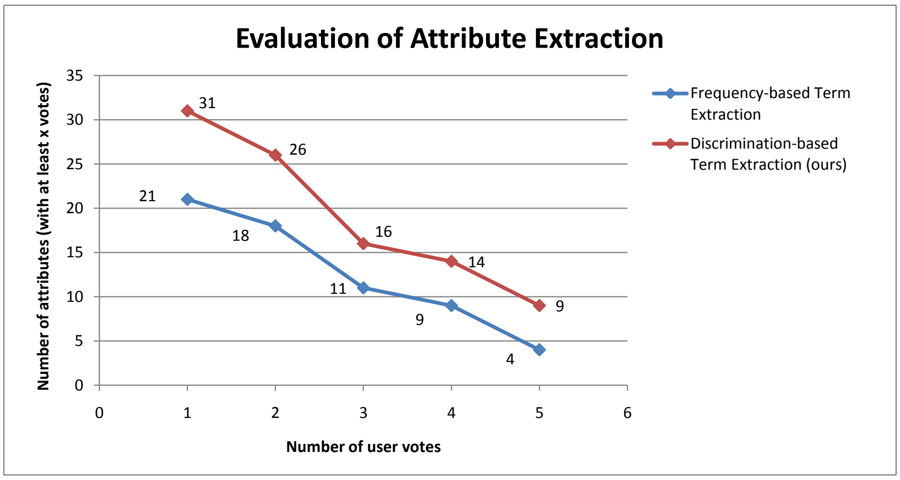


*Figure 6.6: Results of the user-study. On the x-axis the number of users is listed that have voted for an extracted term. The y-axis indicates how many terms were identified by at least x users as useful attributes. A number of 5 users implies that it was an unanimous vote. For each individual user vote threshold our method finds at least 44% more useful attributes than the top-frequency method.*

product attributes optimizing to the discriminative power of the terms is not as important as for the comparison of different conferences. The reason for this might be that reviews are usually very short documents and not everybody is commenting on everything.

In general, it has to be said that judging how useful a given list of attribute terms is for opinion analysis is very difficult. As can be seen in our user study, for many terms this is arguable. Another problem is that none of the methods is able to cope with the existence of synonyms or the usage of abbreviations. Furthermore, topics are not always addressed directly but sometimes only paraphrased or it is only referred to a superordinated topic. As will be shown in the next chapter, the current approach provides us with terms that are reasonable for our application scenarios. However, the above mentioned disadvantages suggest that working with concepts instead of single terms as product attributes could be a valuable direction in the future.

### 6.3.3 Sensitivity with respect to the size of the collection

Since our approach is based on statistics, it is likely to be dependent on a larger size of documents. We therefore tested the influence of the size of the document collections on the attribute extraction. Given a class of camera reviews and a collection of reviews on Harry Potter books containing each 1000 documents, we gradually reduced the number of documents in each class. The following characteristics could be observed: Keeping the parameter settings stable (discrimination factor = 5 and minimum percentage = 3%), the number of extracted terms increases when the number of reviews decreases. Figure 6.7(a) graphically shows the results of the experiment. The rationale behind this is that given a smaller number of reviews also a smaller number of terms is needed to pass the threshold which means that single reviews get a stronger influence (e.g. 3% of 1000 reviews = 30, but 3% of 100 reviews = 3, meaning that some outliers are enough to let a term

| |
|---|
| **Our approach:**<br>network, product, ink cartridge, fax, jam, paper, scan, print quality, print, download, printer, cartridge, software, mac, unit, function, month, all-in-one, installation, machine, scanner, install, box, model, use, hp, feature, replace, easy, black, document, fix, support, driver, ink, color, wireless, photo, expensive, hp printer |
| **Differential analysis:**<br>hp, printer, ink, print, printing, photo, scanner, cartridges, quality, mac, scan, epson, fax, vista, paper, software, scanning, prints, amazon, xp, wireless, photos, copier, machine, cartridge, usb, cd, canon, works, printers color, tech, tray, installation, computer, ethernet, copying, copy, setup, feeder |
| **Most frequent terms (after stopword removal):**<br>printer, print, use, good, work, scan, buy, problem, install, software, great, time, easy, like, need, try, machine, ink cartridge, fax, ink, set, purchase, make, hp printer, copy, paper, run, product, come, price, look, say, want, photo, new, quality, real, page, wireless, think |
| **TFIDF average (after stopword removal):**<br>printer, print, software, computer, ink, wireless, buy, photo, quality, good, scanner, install, time, great, photos, install, easy, price, all-in-one, page, cable, function, purchase, make, setup, Vista, Windows, feature, find, jam, real, tray, program, driver, tech, fast, Epson, copier, XP, option |

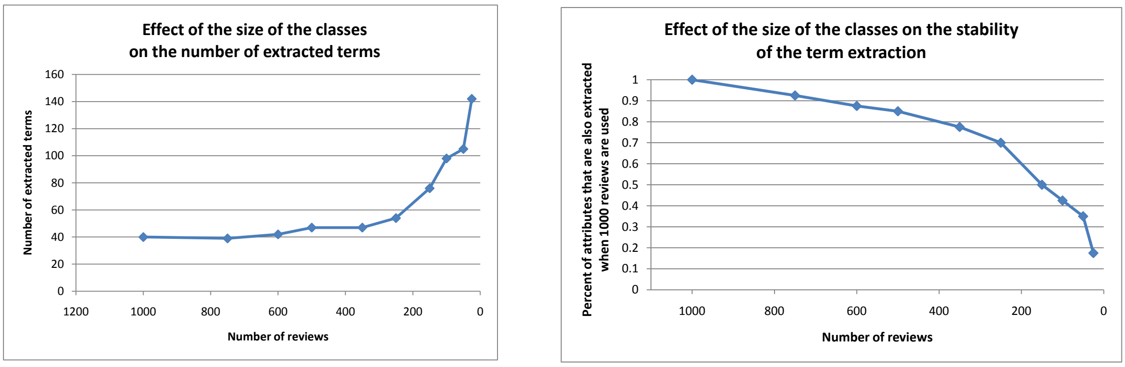
*Table 6.1: The 40 top product attributes from printer reviews using different extraction methods.*

pass through). Similarly, taking the terms that we extracted with our largest collection as a "gold standard", we can observe that smaller collection sizes also result in stronger deviations from this term set. Figure 6.7(b) illustrates the decreasing consensus with the gold standard. The results of our experiments suggest that when analyzing reviews around 200-300 documents are necessary to get stable results. This number is consistent with our impression of the quality of the terms that are extracted. Using less than 200 documents, the extracted terms start to get more and more arbitrary even if we adjust the parameter settings in a way that the number of extracted terms is the same for all document collection sizes[5]. Note that the results of this experiment cannot necessarily be carried over to other document collections as reviews are typically very short. In our case-study with papers (see section 6.4), we could easily obtain meaningful results with a significantly smaller number of documents.

### 6.3.4 Sensitivity with respect to the choice of the counter-balance class

To assess the sensitivity of the approach with respect to the counter-balance class, we compared the results of the method when using different counter-balance classes to extract product attributes from camera reviews. While the discrimination factor was always set to 5, the minimum percentage was chosen in a way that about 30 terms were extracted for each of the counter-balance classes. As counter-balance classes, we used reviews of the following products: Harry Potter books, books about camera usage, HP printers, navigation systems, and rain boots. Table A.4 shows the extracted attributes. The following

[5] The full list of extracted terms can be found in the appendix A.5



(a) Number of extracted terms: It can easily be seen that below 200 documents the number of extracted terms increases significantly.

(b) Stability of the terms: Setting the terms that were extracted with 1000 reviews as "gold standard", the further the number of documents is decreased the stronger the deviation is that can be observed.

*Figure 6.7: Results of an experiment in which the number of documents in the counter-balance class and the class that we want to extract the attributes from was gradually reduced starting with 1000 documents in both classes.*

observations can be made: The words that are included in all attribute lists or at least in four of the five lists are: *image stabilization, hd video, viewfinder, battery life, fuji, image quality, iso, mp, nikon, picture quality, slr, and video.* Besides that, the word lists show some differences. Deciding on what the best word list is might be arguable. The results that we got when using Harry Potter reviews were the ones that satisfied us most. However, other word lists might be appropriate as well.

Some word lists contain words like "vacation", "trip", or "beach". The reason for this is that reviewers tend to write about what they bought the camera for, too, and not only about the attributes of the camera. If we do not want to ask the user to filter out such terms manually, a second counter-balance class could be applied whose documents talk about the topic "vacation". Finally, it can be observed that brand names of cameras can be found in all the word lists. As they are usually not interesting as attributes, they should be filtered out automatically (e.g. with the help of a list with brand names).

The experiment confirms that a careful selection of the counter-balance class is important when using the technique to extract product attributes. The fact that the related technique of Differential Analysis[6] provides us with good terms as well suggests that using a bigger, more general corpus instead of a single class of reviews as we currently do might alleviate the problem in the case of review analysis.
In other application examples, such as the comparison of different conferences (see section 6.4), the choice of the counter-balance class is obvious. Because we are specifically interested in terms that discriminate one set of documents from the other(s), using a general corpus would impair the results. Compared to the application scenario of extracting product attributes, the task of comparing different conferences to each other is closer to the original purpose of this special term extraction technique.

[6] Please refer to section 6.3.1 for an explanation of the technique. Table 6.1 shows the product attributes that were extracted from a set of printer reviews using the technique.

| **Counter-balance class** | **Attributes** |
|---|---|
| Harry Potter | panasonic, button, battery life, video, sony, photographer, pic, battery, user, picture quality, hd video, slr, memory card, auto mode, zoom, flash, fuji, lcd, viewfinder, mode, photo, auto, color, manual, shoot camera, mp, research, lcd screen, iso, amazon verified purchase, image quality, image stabilization, nikon |
| Camera Books | canon powershot, olympus, panasonic, pocket, battery life, quality picture, video, purse, cable, battery, picture quality, performance, menus, hd video, memory card, resolution, screen, beach, software, fuji, view finder, lcd, kodak, mp, lcd screen, sound, research, image quality, image stabilization, charger |
| HP Printer | canon powershot, olympus, battery life, dslr, video, focus, subject, megapixel, purse, sony, photography, hd video, slr, sunlight, auto mode, beach, zoom, fuji, view finder, shutter speed, viewfinder, auto, lens, vacation, shoot camera, iso, amazon verified purchase, image stabilization, nikon, charger |
| Navigation systems | olympus, zoom quality, lense, nikon, quality picture, image quality, image stabilization, kodak, photography, lens, canon cameras, shutter speed, canon powershot, panasonic, picture quality, color quality, camera size, megapixel, mp, slr, aa battery, viewfinder, focus, tripod, shoot camera, iso, view finder, hd video, fuji, dslr |
| Rain boots | battery, auto mode, video, lens, option, resolution, mode, photographer, canon, control, battery life, image, hd video, result, software, slr, nikon, image stabilization, mp, screen, lcd, camera, model, sony, iso, vacation, focus, zoom, picture quality, memory card, image quality, flash, viewfinder |

*Table 6.2: Extracted attributes from camera reviews using different counter-balance classes. The parameter "discrimination factor" was always set to 5. The minimum percentage was chosen in a way that approx. 30 terms are extracted.*

## 6.4 Application: Comparison of proceedings of different conferences

As an example, the technique was used to learn about the topical differences and similarities of scientific conferences by analyzing 100 recently published papers of their proceedings. We compared a set of 9 different conferences of the following areas:

- Information Retrieval (SIGIR),
- Database and Data Storage (VLDB and SIGMOD),
- Database and Natural Languages (NLDB),
- Knowledge Discovery and Data Mining (KDD),
- Visual Analytics (VAST),
- Information Visualization (InfoVis),
- Visualization (VIS), and
- Computer Graphics (SIGGRAPH).

The application was motivated by the following questions: If we take different conferences in the computer science area, can we detect automatically by processing the papers published in these conferences: (a) How they differ from each other? (b) What single conferences focus on or what makes them special? (c) What several conferences have in common, respectively what distinguishes them from the other conferences?

As an introductory example, figure 6.8 shows the results for a comparison of the three conferences Vis, Siggraph, and InfoVis. With such a small set of conferences we can use venn diagrams as an intuitive visualization technique to illustrate the results. In the venn diagram each circle represents one conference. All three conferences deal with graphical representations and visualization. Yet, each conference has its own specific orientation in the field. In an outer section of the diagram that is unique for one of the conferences, the terms that discriminate this conference from all the others are displayed. For the Vis conference and the comparison to Siggraph and InfoVis, those terms are {flow field, scalar field, volume data, volume dataset, vector field, volume visualization}. Furthermore, the terms are shown that are shared by two conferences and discriminate them against the third conference in the overlap regions of the diagram. Apparently, there is no overlap of the Siggraph and the InfoVis conference. While this might not be surprising for an expert in the area of these conferences (as the Vis conference is topically somewhere in between Siggraph and InfoVis), it provides quite useful information to non-experts without requiring substantial reading efforts. The overlap area of all three conferences remains empty, because our approach only extracts discriminating terms and in this case there is nothing to discriminate against.

The results of processing the proceedings of all 9 conferences can be found in figure 6.9. On the left side of the figure, the set of conferences is listed for which a set of terms is discriminating. A conference is contained in this set if its corresponding matrix entry is marked in a blue color tone. The more conferences a set contains, the darker is the blue. The corresponding terms can be found on the right side. The combinations of conferences that do not appear, simply do not jointly discriminate against the others in a certain topic.

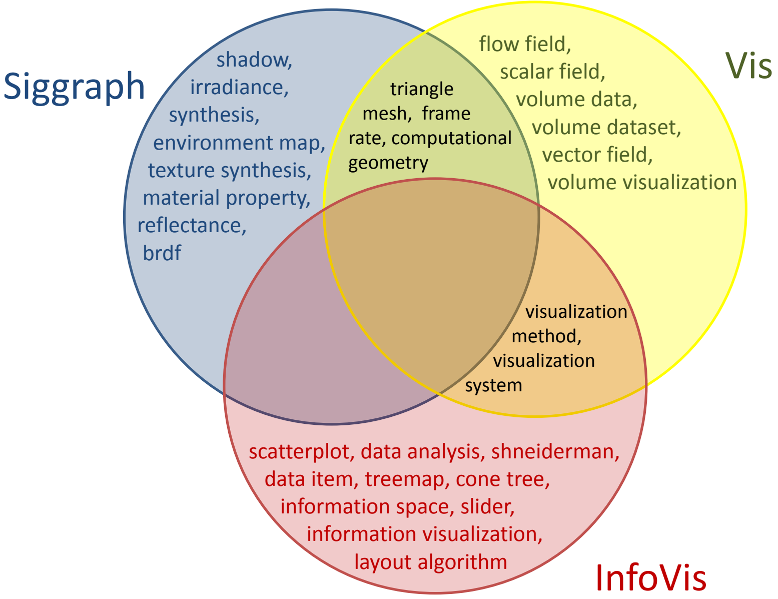
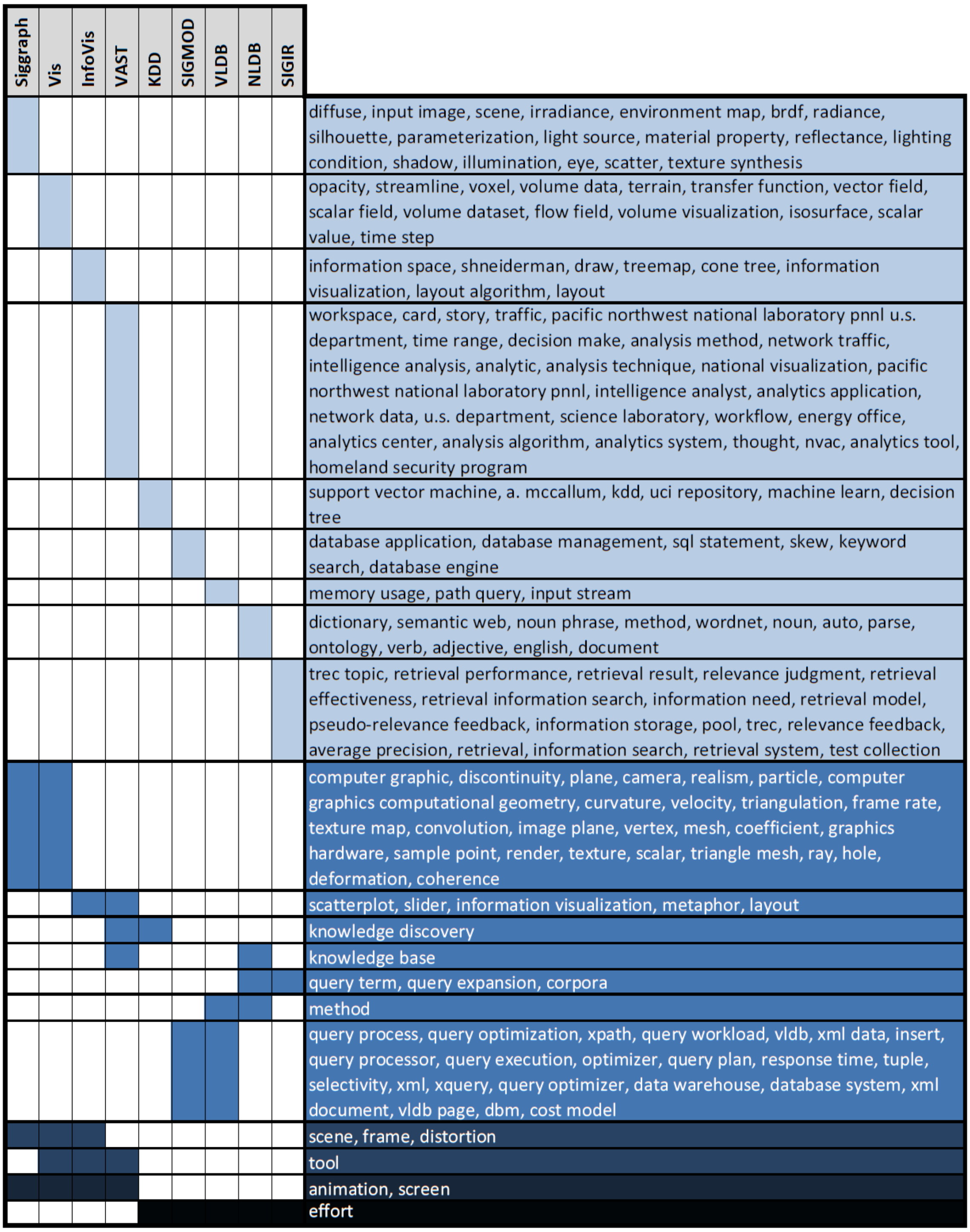


*Figure 6.8: Discriminating and overlap terms for the three conferences Siggraph, Vis, and InfoVis (generated with about 100 papers of each conference). Terms in the overlap areas are shared by two classes and discriminate them against the third one, while the rest of the terms discriminates one specific class against the others.*

As can be seen, the discriminating terms of the NLDB relate very much to natural language. Database-related vocabulary does not appear in the list, because it is also covered by other conferences and thus not discriminating for NLDB in this context. NLDB has a discriminating overlap with the SIGIR conference, because only those two frequently deal with *query terms* and *corpora*. In contrast, everything related to information or document retrieval apparently is significantly more covered by the papers of the SIGIR conference. NLDB has also small discriminating overlaps with VLDB and VAST, but there is no overlap with the conferences that focus on visualization and computer graphics.
Also the other extracted terms for overlaps between two or more conferences fit nicely and are reasonable: E.g. SIGGRAPH and VIS share a lot of computer graphics vocabulary, and InfoVis and VAST the topic of visualizing information. SIGGRAPH, VIS and InfoVis still share some vocabulary related to graphical representations, while VIS, InfoVis and VAST all deal with the development of tools. Finally, while SIGMOD and VLDB are both database conferences that share many database-related topics our method reveals that there are also differences in topic coverage. The term *database management*, for example, only occurs in the SIGMOD term list, while VLDB papers seem to focus more on topics such as *memory usage*.
One particularity of our method is that if a term is important for every class, then it is not extracted: Although e.g. NLDB surely shares topic terms such as *algorithm* or *data* with the visualization conferences, they are not extracted, as all the other considered conferences also contain these topics. Within the context of the selected conferences such terms are not of interest, because they do not provide any discrimination power. Another interesting issue is that some proper names appear in result sets. This is an indication that certain persons and institutions seem to have strong influences on specific conferences.



| Siggraph | Vis | InfoVis | VAST | KDD | SIGMOD | VLDB | NLDB | SIGIR | |
|---|---|---|---|---|---|---|---|---|---|
| ■ | | | | | | | | | diffuse, input image, scene, irradiance, environment map, brdf, radiance, silhouette, parameterization, light source, material property, reflectance, lighting condition, shadow, illumination, eye, scatter, texture synthesis |
| | ■ | | | | | | | | opacity, streamline, voxel, volume data, terrain, transfer function, vector field, scalar field, volume dataset, flow field, volume visualization, isosurface, scalar value, time step |
| | | ■ | | | | | | | information space, shneiderman, draw, treemap, cone tree, information visualization, layout algorithm, layout |
| | | | ■ | | | | | | workspace, card, story, traffic, pacific northwest national laboratory pnnl u.s. department, time range, decision make, analysis method, network traffic, intelligence analysis, analytic, analysis technique, national visualization, pacific northwest national laboratory pnnl, intelligence analyst, analytics application, network data, u.s. department, science laboratory, workflow, energy office, analytics center, analysis algorithm, analytics system, thought, nvac, analytics tool, homeland security program |
| | | | | ■ | | | | | support vector machine, a. mccallum, kdd, uci repository, machine learn, decision tree |
| | | | | | ■ | | | | database application, database management, sql statement, skew, keyword search, database engine |
| | | | | | | ■ | | | memory usage, path query, input stream |
| | | | | | | | ■ | | dictionary, semantic web, noun phrase, method, wordnet, noun, auto, parse, ontology, verb, adjective, english, document |
| | | | | | | | | ■ | trec topic, retrieval performance, retrieval result, relevance judgment, retrieval effectiveness, retrieval information search, information need, retrieval model, pseudo-relevance feedback, information storage, pool, trec, relevance feedback, average precision, retrieval, information search, retrieval system, test collection |
| ■ | ■ | | | | | | | | computer graphic, discontinuity, plane, camera, realism, particle, computer graphics computational geometry, curvature, velocity, triangulation, frame rate, texture map, convolution, image plane, vertex, mesh, coefficient, graphics hardware, sample point, render, texture, scalar, triangle mesh, ray, hole, deformation, coherence |
| | | ■ | ■ | | | | | | scatterplot, slider, information visualization, metaphor, layout |
| | | | ■ | ■ | | | | | knowledge discovery |
| | | | ■ | | | | ■ | | knowledge base |
| | | | | | | | ■ | ■ | query term, query expansion, corpora |
| | | | | | | ■ | ■ | | method |
| | | | | | ■ | ■ | | | query process, query optimization, xpath, query workload, vldb, xml data, insert, query processor, query execution, optimizer, query plan, response time, tuple, selectivity, xml, xquery, query optimizer, data warehouse, database system, xml document, vldb page, dbm, cost model |
| ■ | ■ | ■ | | | | | | | scene, frame, distortion |
| | ■ | ■ | ■ | | | | | | tool |
| ■ | ■ | ■ | ■ | | | | | | animation, screen |
| | | | | ■ | ■ | ■ | ■ | ■ | effort |

*Figure 6.9: On the left side the set of conferences is listed for which a set of terms is discriminating. A conference is contained in this set if its corresponding matrix entry is marked in a blue color tone. The more conferences a set contains, the darker is the blue. The corresponding terms can be found on the right side. The combinations of conferences that do not appear simply do not jointly discriminate against the others in a certain topic.*

## 6.5 Summary and future work

In this chapter, a novel measure for term extraction was proposed that is optimized with respect to extracting terms that discriminate a document collection from a counter-balance class. Furthermore, the approach permits to determine overlap terms that discriminate multiple document collections from several others. The method does not extract terms that are central concepts in the collection if they occur with the same frequency in the counter-balance class. Instead, the focus is put on the discriminative aspect. An in-depth evaluation showed that the proposed approach can hold up to these requirements. Furthermore, the application to a scenario in which the proceedings of different conferences were compared to learn about their differences showed the usefulness of the technique in a real-world scenario.

Like in the other application chapters, visualization is used to support the analysis process. However, the visualizations in this chapter are rather basic. Especially in the last step of the pipeline (e.g. for analyzing the document collection), a more intuitive visualization would be beneficial. The application of venn diagrams (figure 6.8) seems promising. But with more than 3 collections, drawing a venn diagram in which terms can be plotted becomes difficult. Maybe the less restrictive Euler diagrams might be a solution, as often not all possible overlaps between the collections exist.

The evaluation showed that the technique is well suited to approximate the quasi-semantic properties *discriminating terms* and *overlap terms* as defined above. However, it also indicated that the results of the measure are satisfiable, but not yet ideal when used to approximate the quasi-semantic property *product attribute*. One of the challenges in this domain is that the documents (reviews) are usually very short. Furthermore, the terminology is not as well defined and carefully chosen as in the domain of scientific publications. In future work, a technique could be developed that takes semantic concepts into account instead of focusing on single terms. The approach should be able to deal with clusters of semantically-related terms (such as “monitor diagonal”, “diagonal”, “size of the monitor” etc.) and consider them as a single concept. This would alleviate the above mentioned problems and could further improve the results.

# 7
# Quasi-semantic Property IV: Sentiment and Opinion

## Contents

Generally, data analysis techniques are used to mine collections of *facts*, e.g. sales figures that are collected in databases. However, there are many analysis tasks for which *subjective information* is relevant as well. Examples include an analysis what the voters think during a political election and the analysis of customer opinions about a specific product. A rich source of subjective information is available in textual form. Despite of this, standard techniques tend to ignore subjective content or do not make a distinction between facts and opinions (e.g. in information retrieval, subjective and objective statements are treated the same). However, for some document analysis questions this distinction is indispensable. It is therefore not surprising that the development of opinion and sentiment analysis techniques was a hot topic in the last years (see section 3.5).

This chapter is structured as follows: After an introduction and examination of the problem (section 7.1), automatic algorithms for sentiment and opinion detection are introduced (section 7.2). Following this, a detailed evaluation of the algorithms is presented (section 7.3). With the help of visual analysis techniques, the strengths and weaknesses of the algorithms are examined and the methods are systematically improved. Most related work does not go beyond the automatic detection of sentiment and opinion. However, this leaves the user with a large collection of data that is now structured but still too extensive to study manually. Sections 7.4 and 7.5 are dedicated to the visual analysis of the mined data. In several application examples, the applicability and usefulness of the visualizations is shown. The chapter concludes with a summary in section 7.6.

## 7.1 Research and application context

Many different document analysis scenarios exist in which we are interested in the opinions that are directly or indirectly expressed in a text. Examples include the following:

- **Example No. 1: Customer feedback** Companies would like to know what their customers like or dislike about their products. Knowing this helps them to tailor their marketing strategies, improve their products, and optimize their business strategies for the future. For potential customers, the same information is interesting. They can learn from the experiences of other customers and make sure that they find the product that fits their needs best. Thanks to the internet, large resources of customer feedback on all different kinds of products are freely available as product reviews or forum posts. However, mining this large data collection manually is time-consuming and therefore only a small amount of documents can be taken into account if no automatic support is given.

- **Example No. 2: Politics** Politicians are interested in what their citizens think. This is especially true before an election, but also in between e.g. in situations in which decisions have to be made that are controversially discussed in the population. Furthermore, in order to make the right decisions, it is important to know how a topic is discussed in other parts of the world. Polls can illuminate the opinions of the citizens of a country, but alternatively, the information could also be extracted from newspapers (assuming that they mirror and influence the public opinion) or by reading blogs or twitter news.
  On the other hand, for citizens it is sometimes difficult to tell what certain politicians stand for. This is even more true if what they say is not in harmony with what they do. Reviewing news articles of the past months and years that report about what they said and did can cast light on their actual positions.

- **Example No. 3: Newspaper bias** Newspapers report about what happens in the world. However, they are not completely neutral in doing so. Instead, there is a bias with respect to which news they write about but also with respect to how they write about the events. Since newspapers have a significant influence on forming the public opinion, knowing their general orientation is interesting. Similarly, reports in newspapers often reflect the view of the country they are produced in.

It is easy to see that all of the above mentioned examples somehow refer to the opinion that is expressed in textual data. However, at a second glance some important differences can be recognized. First, in example 1 we can expect that the opinions that we are looking for are directly and clearly expressed in the text. The main purpose of writing a review about a product is to express an opinion about it. Other than that, opinions in newspapers are generally expressed more subtle. In this case, not only subjective statements about an event are interesting, but also to observe which facts they are writing about and which they discard. In example 2 a third way of expressing an opinion can be found. In this case, what a politician claims to do and believe has to be compared to the actions that are reported on. Such a comparison requires a high degree of interpretation and knowledge of the world and is therefore not easy to accomplish computationally. Another difference between the scenarios is that sometimes, we are interested in the general sentiment that is expressed in a document about a specific topic, whereas in other cases details are required about *what* exactly is seen as positive or negative (compare e.g. news bias analysis with customer feedback analysis). And finally, we can distinguish between document analysis tasks that work on a set of documents that are known as being mostly subjective reports (such as customer reviews) and others that process data that is a mixture between facts and subjective content (e.g. newspapers).

Making the distinction between the above mentioned characteristics is important, because it affects the design of the algorithms. Later in the chapter it is shown that algorithms that were designed for one category cannot necessarily be employed for another category, too. It is therefore important to carefully specify the quasi-semantic property that is needed.

Note that there is no clear distinction in literature between the terms “sentiment (analysis)” and “opinion (analysis)”. We use the term “sentiment” (and “sentiment analysis”), when we refer to a general subjective view that is expressed on something and that either has a positive or a negative tendency. In contrast to this, the term “opinion” is used to denote a more differentiated positive or negative view on a subject. Opinion analysis additionally asks for *what* the author likes or dislikes. It is therefore also called “feature-based or attribute-based opinion analysis” where “attribute” refers to details of the subject that are liked or disliked (e.g. components or features of a product or actions of a politician). A more detailed review of those terms in related work can be found in [103] section 1.5.

### Analysis tasks and quasi-semantic questions

In this chapter, solutions for customer feedback analysis (as specified in Example 1 above) are discussed. With respect to the above mentioned categories, customer feedback analysis is an example for a task in which we can focus on data that is mostly subjective in its nature and in which the opinions are directly and clearly expressed. A challenge is that we are not only interested in the general sentiment that is expressed in the documents, but we also ask for *what* customers liked or disliked.

The analysis questions include:

- What is the general trend in the opinions on a specific product with respect to the attributes that are frequently commented on?
- How do different products compare to each other with respect to the expressed opinions?
- What are the most severe problems of a product according to the customers’ opinion?
- Are there subgroups of people with similar opinions?

Hence, the quasi-semantic properties that we need to measure and their related quasi-semantic questions are:

- *Product attribute* - What features or parts of the product do customers frequently comment on?
- *Sentiment* - What kind of sentiment(s) is (are) expressed in a document?
- *Opinion* - Are the mentioned attributes commented on positively or negatively?

The extraction of product attributes was already discussed in chapter 6. Techniques for sentiment and opinion analysis will be presented in this chapter. Additionally, the introduced techniques will be applied to news articles. We show that using the method for extracting the quasi-semantic property *sentiment* that was successfully applied to the customer feedback data, provides interesting insight with respect to the comparison of newspaper articles. However, when applied to news articles this technique does not extract the quasi-semantic property *sentiment* in the classical sense but something that rather could be seen as a *context polarity* of the topic. This means that our measurements are not necessarily based on the subjective content of a document but rather compare the documents with respect to their choice of news about a topic with a positive or negative connotation. This allows us to answer the analysis questions *How do the newspapers of different countries talk about certain topics?* and more specifically: *Do they mention the topic in a negative or a positive context?*

### Research focus of the chapter

In the previous years, already much research has been done in the context of sentiment and opinion analysis. This thesis contributes to the state-of-the-art in the following aspects:

- **Visual evaluation and improvement of the sentiment and opinion analysis methods**
  This chapter presents an in-depth evaluation of the used techniques for sentiment and opinion detection. Visual Analysis Methods are applied in the evaluation process. We are able to show that this helps to gain a deep understanding of the algorithms and allows a systematic examination of the strengths and weaknesses of the algorithms.

- **Improvement of a popular opinion analysis method**
  By consequently learning from the systematic evaluation of the algorithms, we are able to improve a popular opinion analysis method. Furthermore, we assess how incorporating linguistic knowledge about the grammatical structure of a sentence in the measure affects the output of the algorithm.

- **Domain-dependency of the quasi-semantic measure**
  A quasi-semantic measure that has been developed in one application context cannot necessarily be applied directly to another application context. This becomes apparent in one of the application scenarios, where also possible explanations for

this phenomenon are discussed. Furthermore, it is also shown that using the technique in another context (here: news data analysis) can result in interesting findings even if this measures a slightly different quasi-semantic property.

- **Visual analysis of the detected opinions**
  While there are many related approaches that automatically extract an opinion from customer feedback data, only few techniques exist for analyzing the documents with respect to the extracted properties. In this chapter, we are going to introduce a novel visual analysis method called summary reports that provides a compact but yet sufficiently detailed overview to compare the reviews of several products to each other. Furthermore, the Literature Fingerprinting technique that is introduced in chapter 4 is applied in the context of news feed analysis. Finally, we explore a set of customer reviews with respect to subgroups with similar opinions.

## 7.2 Automatic algorithms for sentiment and opinion detection

Section 7.2.1 presents a simple heuristic for detecting the sentiment of a text. Section 7.2.2 explains the basic algorithm for opinion detection that is employed in this thesis. Part of this process is the mapping between attributes and sentiment. Sections 7.2.3 and 7.2.4 present two different heuristics for performing this mapping: one is statistically-based, the other one incorporates linguistic knowledge about the sentence structure.

### 7.2.1 Algorithm for sentiment detection

To get a sentiment score for each article, a basic sentiment analysis algorithm was applied. With the help of two lists with opinion signal words (one with negative signal words and the other with positive ones) from the General Inquirer Project [46], each word is classified as positive, negative or neutral[1]. We count the number of positive signal words in an article and subtract the number of negative signal words from it. To improve the accuracy of the method, negation is taken into account. This is done by inverting the value of a word if in a maximum distance of $X$ words a negation signal word is found (such as “no”, “not”, “without”, ...). In this case, the parameter $X$ (the maximum distance to the negation signal word) was set to 3, a value that experimentally proved as minimizing the failures. For alternative approaches please refer to section 3.5.

### 7.2.2 Attribute-based opinion analysis

Attribute-based opinion analysis is often done as a two-step process (in our case three steps):

[1] Note that the list contains signal words of all parts of speech. That means, not only opinion-bearing adjectives, but also nouns, verbs, etc. (e.g. “catastrophe”, “to like”).

**Step 1:** First, the opinion signal words and the attributes that are commented on are identified. In our experiments, we use the algorithm that is presented in chapter 6 to compile a list of attributes that are frequently commented on. This list is then used to identify mentions of attributes in the reviews. Note that alternative approaches for detecting attributes exist that could easily be substituted in our process (see section 3.4). As an opinion word list, we use the General Inquirer dictionary that was already introduced in the previous subsection.

**Step 2:** After annotating all the occurrences of attributes and opinion signal words in the text, a mapping between attributes and opinion words is performed. This means that we determine for each attribute in a sentence which opinion word(s) refer to it, classifying each attribute as positive, neutral, or negative. Thus, we need to combine the quasi-semantic measure *product attribute* with the text feature *opinion signal word* that was already used in the previous subsection. Note that it is important to take the category "neutral" into account as well, because in real-world data sets an attribute can also be mentioned in a non-evaluative way. In the next subsections, two different ways of performing the mapping are introduced.

**Step 3:** In our application scenario, we do not only want to know whether an attribute was mentioned positively or negatively in a specific sentence. Instead, we are interested in the overall opinion that was expressed about the attribute in the review. If an attribute is mentioned several times in the review, the majority vote of the sentence polarities for this attribute is determined to get its opinion value on the review-level.

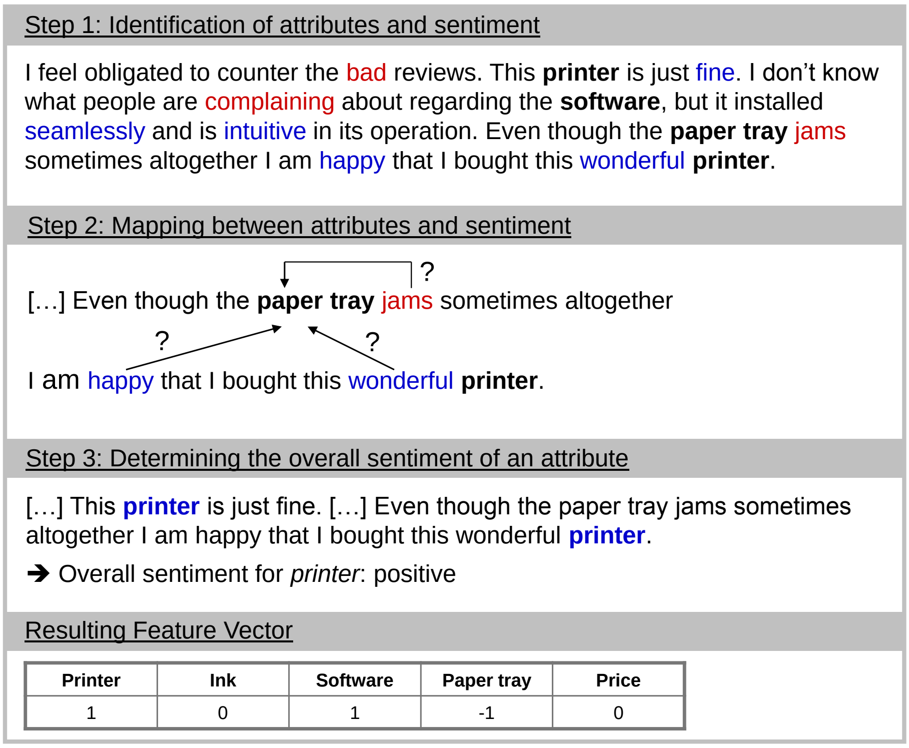
**Result:** As a result, we get a feature vector for each review that summarizes the expressed opinions on the individual attributes. For each attribute, there is one feature dimension in the vector. The corresponding value of the vector for a particular attribute's dimension indicates whether the attribute was mentioned positively (+1), negatively (-1) or neutrally / not at all (0).

Figure 7.1 exemplifies the process.

### 7.2.3 Statistically-based mapping

As a statistically-based approach to perform the mapping, we use a method that is similar to the one presented in [66, 28]. (The main difference is that we use an additional cutoff value and need attribute scores on document level.) The algorithm is based on the assumption that the closer an opinion signal word is to an attribute, the higher is the probability that it refers to the attribute. The opinion score for each attribute is therefore calculated by building a weighted sum of all opinion signal words that are in the same sentence than the attribute. The weight is determined by a weighting function that takes the distance to the attribute into account. An example for such a weighting function is given in equation 7.1.

$$\textit{rd-weight(A,o)} = \begin{cases} 1 & \textit{if dist(A,o)} \leq \textit{cutoff/2}, \\ 0.5 & \textit{if cutoff/2} < \textit{dist(A,o)} \leq \textit{cutoff}, \\ 0 & \textit{else.} \end{cases} \tag{7.1}$$



Step 1: Identification of attributes and sentiment

I feel obligated to counter the bad reviews. This **printer** is just fine. I don't know what people are complaining about regarding the **software**, but it installed seamlessly and is intuitive in its operation. Even though the **paper tray** jams sometimes altogether I am happy that I bought this wonderful **printer**.

Step 2: Mapping between attributes and sentiment

[…] Even though the **paper tray** jams sometimes altogether

I am happy that I bought this wonderful **printer**.

Step 3: Determining the overall sentiment of an attribute

[…] This **printer** is just fine. […] Even though the paper tray jams sometimes altogether I am happy that I bought this wonderful **printer**.

➔ Overall sentiment for *printer*: positive

Resulting Feature Vector

| Printer | Ink | Software | Paper tray | Price |
|---|---|---|---|---|
| 1 | 0 | 1 | -1 | 0 |

*Figure 7.1: The graphic above illustrates the different steps in the opinion analysis process. Attributes are highlighted in bold face. Opinion signal words are colored in blue if they are positive and in red if they are negative.*

Instead of a step function like the one above, other weighting functions, e.g. a gaussian function could be used as well. Our experiments showed that gaussian functions provide equally good results given that a cutoff-value is used that defines how many words before and after the attribute are taken into account at most. In our application, the cutoff threshold was set to 4. Experiments showed that this cutoff value is especially important in long sentences, because it prevents errors that are caused by very distant opinion signal words that are incorrectly mapped to the attribute. Instead of using a cutoff-value, [28] proposes to determine sentence segments by searching for so called *BUT*-words/phrases (e.g. "but", "except that", …) that tend to invert the sentiment that is expressed in a sentence. Within a sentence segment, all the opinion signal words are taken into account and weighted by equation 7.2:

$$rd\text{-}weight(A,o) = \frac{1}{dist(A,o)} \tag{7.2}$$

In order to get an opinion value for an attribute A and a sentence S equation 7.3 is applied. The weighted polarity values for each opinion signal word o within S are summed up. The polarity value of an opinion signal word is either +1 or -1 depending on whether it is contained in the positive or negative word list. Thereby, negation is taken into account in the way described in section 7.2.1, inverting the polarity of an opinion signal word.

$$opinion\text{-}score(A,S) = \sum_{o \in S} rd\text{-}weight(A,o) \cdot polarity(o) \tag{7.3}$$

If an attribute A gets a positive opinion score, the sentence is interpreted as talking positively about the attribute. Likewise, if this sum is negative, the sentence is supposed to talk in a negative manner about the attribute. If the sum is equal to 0, the polarity of the closest opinion signal word is decisive. Note that the opinion-score value cannot be used to determine the strength of the opinion, because the distance or the number of opinion signal words in a sentence do not necessarily correlate with intensity of opinion.

### 7.2.4 Linguistically-based mapping

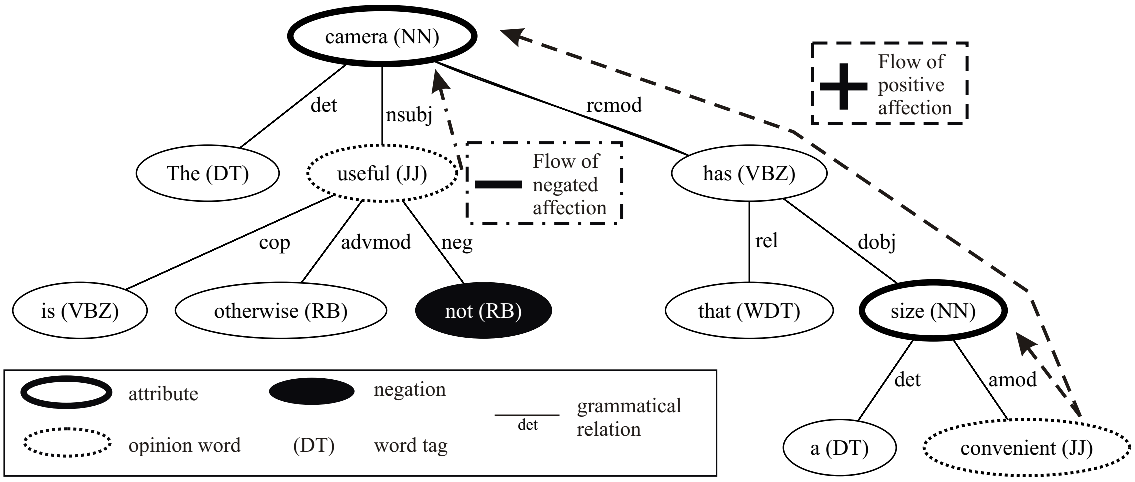


*Figure 7.2: Output of the dependency parser for the input sentence "The camera that has a convenient size is otherwise not useful."*

Additionally to the previously presented technique, we tried to gain as much as possible from including linguistic knowledge. It has to be assumed that the approach of measuring the distance between an attribute and an opinion signal words fails in sentences with a complex nesting of subordinate clauses. We therefore use a dependency graph built by the Stanford Parser [77, 76] to analyze the grammatical structure (incl. negations) of the input sentences. In figure 7.2 an example output of the parser for the sentence *"The camera that has a convenient size is otherwise not useful."* is depicted. Different line types and shadings represent properties that have to be determined. First, we check which of the nodes (representing each of the tokens) is an opinion word and retrieve its polarity from a given opinion word list. In figure 7.2, all opinion words are painted as nodes with a dashed line. Afterwards, we mark each edge (representing a grammatical relationship between two tokens) that is negating or passing affections on. For instance, determiners would not let affections pass through, whereas the relative clause modifier ("rcmod") would[2].
After building our representation of the sentence structure, we traverse the graph starting from the opinion words and propagate their polarity through the graph. The influence of an opinion word decreases with increasing graph distance and the polarity is negated whenever

[2] Complete list of passing edges: conj (conjunct), amod (adjectival modifier), nsubj (nominal subject), dobj (direct object), prep (prepositional modifier), rcmod (relative clause modifier), xcomp (clausal complement with external subject), nn (noun compound modifier), dep (dependent), cop (copula), ccomp (clausal complement with internal subject). The following edges negate the opinion: neg (negation modifier), conj_neg (negating conjunction), conj_but (but conjunction).

negating edges are passed. Note that this traversal is done for each of the opinion words and consequently the polarities weighted by distance are summed up. Considering the previously mentioned example sentence, we would propagate a positive polarity from the token "convenient" to the attributes "size" and "camera". Additionally, a negated positive polarity would be propagated from "useful" to the attributes "camera" and "size". As the influence of the opinion words decreases with increasing graph distance, the resulting polarity for "camera" would be negative and for "size" positive.

To be able to work with compound nouns as well, we group tokens together which are directly connected in the dependency graph via a "NN"- (compound noun) or "amod"- (modifier) edge. The affection of those compound attributes is the average affection of all their tokens.

To the best of our knowledge, this is the first attempt to use the result of a dependency parser to measure the distance between an opinion signal word and an attribute. However, it is not the first approach that attempts to improve sentiment analysis techniques with a dependency parser. E.g., Ng et al. [99] use the output of the parser to get special n-grams that represent adjective-noun, subject-verb, or verb-object relations for their document-level sentiment analysis. Zhuang et al. [164] parse a sentence for candidate attributes and opinion words using a keyword list and subsequently, with the help of dependency relation templates, classify the attribute-opinion pairs as forming a valid attribute-opinion pair or not. Templates, such as *attribute modifies noun*, are automatically extracted from a training data set. While [99] states that the results are similar to the ones achieved with traditional n-gram feature vectors, Zhuang et al. report on significant improvements over comparative methods without a dependency parser. In contrast to [164] (which is also an attribute-based method), we use the dependency tree to measure the distance between two words in a way that takes the syntactic structure of the sentence into account but do not specify any templates.

## 7.3 Evaluation and improvement of the QSPs "Sentiment" and "Opinion"

In the following, a detailed evaluation of the sentiment and opinion analysis algorithms is presented. We show how visualization can help to understand algorithms and derive targeted improvement strategies where necessary. In section 7.3.1, we demonstrate how the Literature Fingerprinting technique that is presented in chapter 4 can help to assess the sentiment detection algorithm of section 7.2.1. Following this, section 7.3.2 presents an in-depth evaluation of the mapping strategy. With the help of expressive visualization techniques, the strengths and weaknesses of the employed algorithm are discovered. Systematically, suggestions for improving the algorithm are derived and tested to assess their potential to increase the accuracy of the algorithm. Within the scope of this analysis, we also evaluated the benefit of using the output of the dependency parser to calculate the distance between the opinion signal word and an attribute as suggested in section 7.2.4.

### 7.3.1 Visual evaluation of the sentiment detection algorithm

In the following, the sentiment analysis technique that is presented in section 7.2.1 is evaluated. Departing from section 7.2.1, we apply the technique to the sentence level.

As a benchmark data set, reviews from amazon.com on a digital camera are used. Each sentence was manually rated with respect to the expressed attitude towards the camera (positive, neutral or negative). We use the Literature Fingerprinting technique that is introduced in chapter 4 to provide detailed insight into the strengths and weaknesses of the algorithm. In order to show how visualization techniques can support the evaluation and improvement of a feature, we start with a very basic algorithm that does not take negation and nouns as opinion words into account. Furthermore, the visual representation is used to uncover correlations to other features (in this case *sentence length*).

Figure 7.3 shows our benchmark data set and the result of the automatic classification. Each squared pixel represents one sentence and color is mapped to the assigned class (green = positive, white = neutral, and red = negative). The color gradations in figure 7.3 can be interpreted as how sure the algorithm is about its rating. The pixels are grouped into the three classes. Ideally, a perfect feature would only have green pixels in the first group, white ones in the second group and red ones in the last group. Such grouping allows us to analyze whether a feature has particular problems with one of the classes. In the example in figure 7.3 we can see that the class of positive statements is the easiest one for the algorithm, whereas there are more errors in the sections of the neutral and negative statements. Within each class the pixels are sorted by the length of the sentences. As can be seen by the decreasing confidence of the feature for classifying positive or negative sentences, there is a weak correlation between this property and the feature value. Opposing to this, in neutral sentences this correlation is negative. In case of the positive and negative statements this means that the algorithm is more confident about its decision (more correct ones and higher (darker) values), whereas in the section of the neutral statements the length of the sentences is negatively correlated to the classification accuracy. This can be explained by the fact that the probability that opinion words appear is higher in longer sentences.

To get hints for further improvements, we pointedly analyzed the sentences that were wrongly classified by the algorithm. In figure 7.3 some pixels have been annotated with the underlying text. In the annotation, words that appear in the list of positive opinion words are colored in green and the negative ones are colored in red to enable an understanding of the decision of the algorithm and reveal the problems. While analyzing the results, it can be seen that there are some systematic errors. Reason for this is that negation is not taken into account, and nouns are not included in the list of opinion words. To improve the feature, two extensions were evaluated: First, negation is taken into account by inverting the value of a word if one of the $X$ preceding words is a negation signal word (such as “no”, “not”, “without” ...). We set the parameter $X$ (the maximum distance to the negation signal word) experimentally to 3 minimizing the failures. Second, we added nouns with negative / positive connotations (such as “problem”, “error”, “advantage”) to our list of opinion words.

In the following, we are going to evaluate whether and how those extensions result in an improvement of the classification of the sentences. Figure 7.4 visualizes the changes that occur when negation respectively nouns are taken into account. In this visualization, all sentences whose values did not change from one version to the next one are colored in white respectively in yellow if their classification is still wrong. For the rest of the pixels we distinguished between minor and major improvements by highlighting them in light green and dark green, respectively. Correspondingly, minor deteriorations are highlighted in light red and major deteriorations in dark red. We speak of a minor improvement if

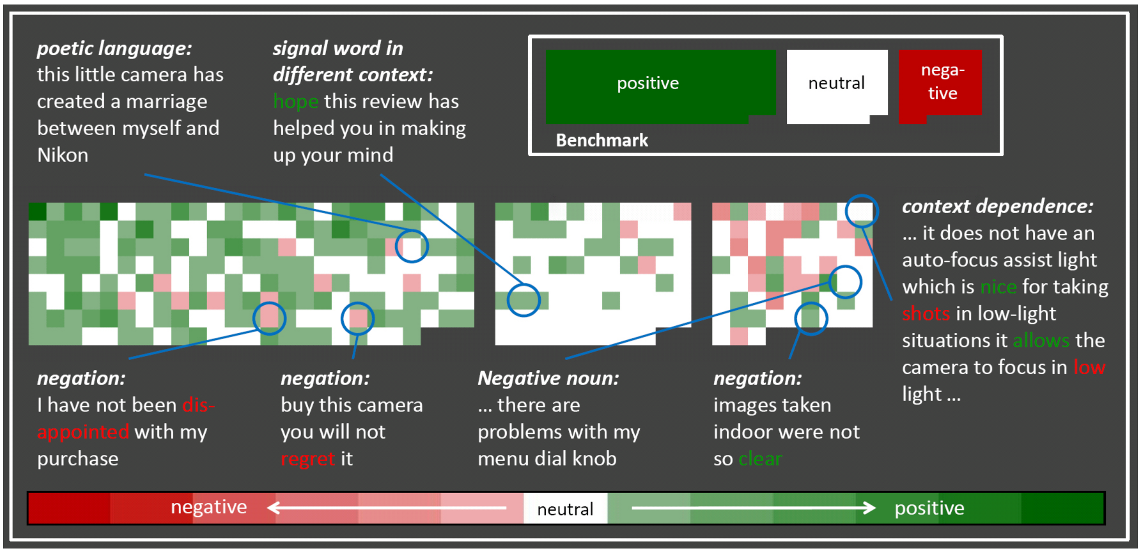


*Figure 7.3: Visualization of product reviews for a digital camera. Each pixel represents one sentence. The sentences are grouped into positive, neutral, and negative statements (left, middle, right as shown in the benchmark visualization above). The sentences are sorted by their length allowing to analyze whether the classification accuracy correlates with this property. Color is mapped to the classification result of the algorithm. The visualization has been annotated with comments on some of the wrongly classified statements. The feature evaluated here is based on lists of positive and negative words. In the annotation, words that appear in the list of positive opinion words are colored in green and the negative ones are colored in red to enable an understanding of the decision of the algorithm and reveal the problems.*

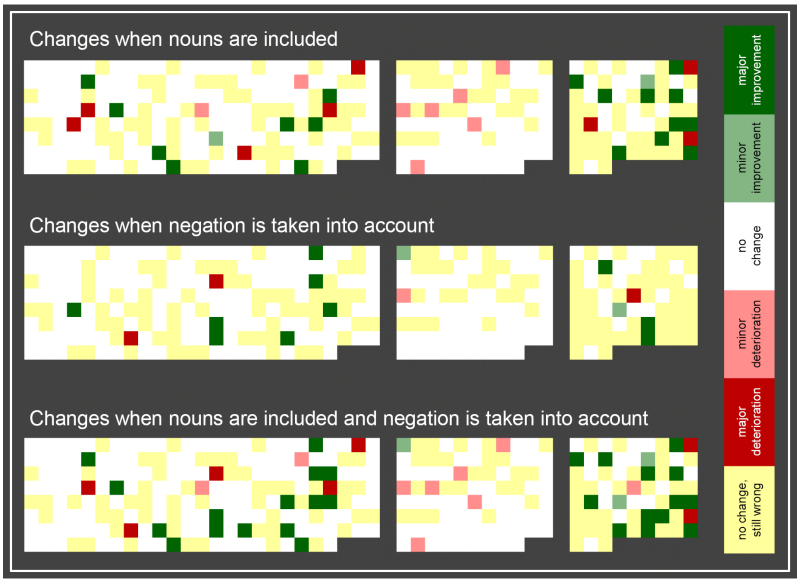


*Figure 7.4: Visualization of the changes that occur when the feature is extended by adding nouns to the list of opinion words, taking negation into account, or by using both extensions at once. It can easily be seen that the class of negative statements profits most from the changes, but that a decrease in the classification accuracy of the class of neutral statements has to be accepted.*

the prediction was moved one step in the correct direction on the scale negative - neutral - positive, but the classification is still wrong (Example: Sentence in benchmark is defined

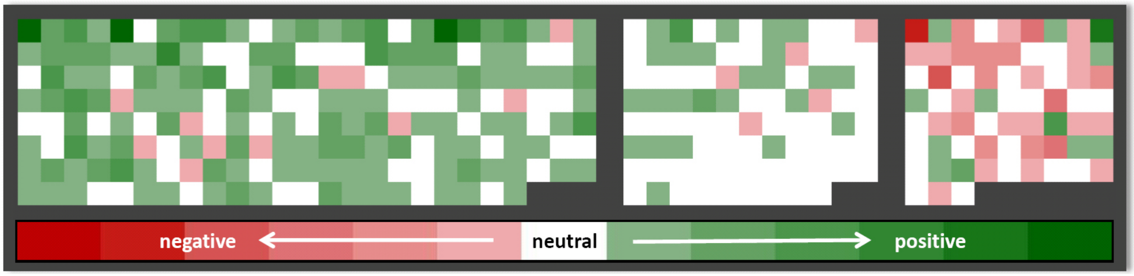


*Figure 7.5: Visualization of the classification results when the feature has been extended by taking negation and nouns into account. In comparison to figure 7.3, it can be seen that especially the class of negative statements profited from the changes.*

as negative, but in the first version it was wrongly classified as positive. If in the second version the sentence became neutral, this would lead to a minor improvement. If the same sentence was correctly classified as negative in the second version, this would lead to a major improvement). One can easily see in figure 7.4 that both extensions result in an improvement. Especially the class of the negative statements seems to profit from the enhancements. However, it is also obvious that we introduced some new mistakes. This is particularly true for the class of the neutral statements which did not profit from the changes.

Finally, the third visualization in figure 7.4 shows the changes when both extensions are combined. As can be seen, some of the errors that were introduced by one of the extensions could be eliminated by the combination of both. Figure 7.5 visualizes the classification result when both extensions are used. Compared to figure 7.3, especially the section with the negative statements has profited from our changes. However, there are still some mistakes in all three classes. Analyzing them again as in figure 7.3, reveals that there are different kinds of mistakes, some of which could easily be fixed. First of all, we recognized that our list of positive / negative words is not complete. Those words could easily be added. Furthermore, the list could be extended by context-dependent opinion words respectively the ones that do not have a positive or negative connotation in our context could be removed (like "shoot"). Other mistakes would require the usage of advanced natural language processing (NLP) algorithms, e.g. to detect change in context. (e.g. if someone compares an old camera to this new one and the *old* camera is mentioned in a negative way). Even more difficult are the cases in which the text is written in slang or in which no opinion words are used at all and knowledge about the context is required to interpret the sentence correctly (like *"You got to have flash on to get it even though your room is well lit"*).

### 7.3.2 Identification of the improvement potentials in attribute-based opinion analysis

We manually annotated 548 reviews for a DVD player to do a more extensive study. In total 996 mentions of attributes were classified as positive, negative, or neutral. This allows us to analyze systematically which properties might influence the result. Furthermore, neutral sentences are taken into account as well, which has rarely been done in related work so far ([66, 107, 158], for example, work on subjective sentences only). Since neutral sentences frequently contain attribute names, too, in a real-world scenario the algorithm has to be able to detect their neutrality even if opinion words are present as in the sentence “I bought a notebook and a wonderful printer” where the attribute “notebook” was not commented on. Only an evaluation that takes neutral sentences into account as well is able to provide detailed insight into the performance of the algorithm. Otherwise, there is a risk for tuning the algorithm with respect to its ability to discern negative attributes from positive ones to the detriment of correctly identifying neutral attributes.

In the following, we are going to

1. assess the accuracy that can be achieved with a state-of-the-art methodology (taking neutral mentions of attributes into account as well)
2. analyze error-sources in detail and in dependence of different variables like the number of opinion words or the length of a sentence
3. come up with targeted improvement strategies for the existing methods and test them for their effectiveness in terms of yielding better accuracies.

Thereby, visual analysis methods are employed to easy the analysis of the effect of different variables.

### Employed method for opinion analysis

To perform the mapping between opinion words and attributes, we employed the method described in section 7.2.3 using the inverse distance weighting that was introduced in [28]. As an opinion word dictionary, we used the list of the General Inquirer Project [46] in a version that was revised by Nancy Chinchor.

### Performance of the statistically-based approach

We get an overall accuracy of 55.1% when working with the Simple Word Distance Mapping. Our biggest class contains 37.4% of the data, so the value is clearly above the majority vote. But still, the result is not fully satisfactory.

The confusion matrix in figure 7.6 (left) reveals some of the problems of the technique. In this mosaic plot visualization [54, 42], the width of each column is scaled proportionally to the number of cases in this target category. Furthermore, within each column, the height of a box is determined by the number of cases in this cell of the matrix. The numbers in each cell show the exact values. Row and column sums are represented by triangles at the sides of the matrix.

This representation permits to quickly spot some distinctive characteristics. First, it is easy to see that category “-1” (negative attributes) is the one that the algorithm had most problems with. Only about 32% of the negative attributes could be retrieved (recall). On the other hand, the precision of the attributes that were classified as -1 was pretty high with 79.1% (compared to 44.8% for class “0” and 55.5% for class “1”). We can conclude that we miss lots of negative attributes, but if we classify one as negative, this is a strong indicator that it indeed is negative.

In column 2, we observe that almost 6 times as many cases are misclassified as +1 than as -1. Furthermore, in total we predict about 1.4 times too often +1 and about 2.5

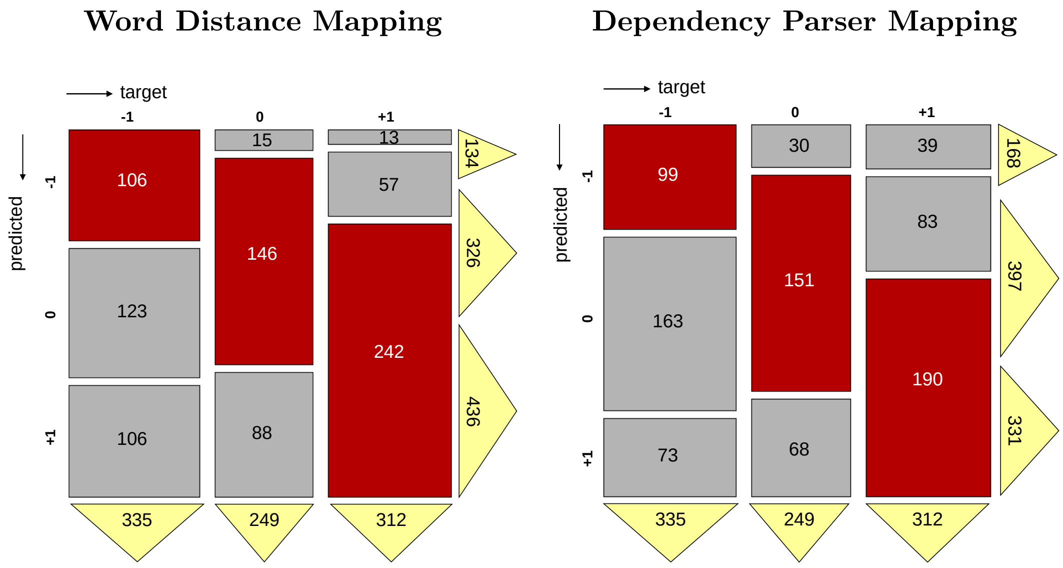


*Figure 7.6: Visually enhanced versions of the confusion matrices showing the results obtained with the simple word distance mapping (left) and the dependency parser mapping (right).*

times too less -1. Given the fact that our opinion word dictionary contains more negative than positive terms (5431 negative vs. 3076 positive), this is an interesting observation that calls for a deeper investigation.

In an application scenario in which the general trend of an attribute in a set of reviews is to be analyzed, missing a comment is less problematic than misclassifying it (swapping -1 and +1). The big number of negative attributes that were misclassified as +1 therefore bothers us. Furthermore, the imbalanced misclassifications of neutral attributes (much more often positive than negative) are a problem.

### Evaluation according to number of opinion words and sentence length

When breaking the evaluation down into many subcategories, more detailed insights are possible. In figure 7.7 such an in-depth evaluation is provided: The quality of the results is assessed (1) in dependence of the amount of opinion words that could be detected within the considered sentence and (2) in dependence of the respective sentence length.

When considering the influence of the amount of opinion words present in a sentence, several interesting observations can be made (see figure 7.7 (1)): It is obvious that sentences that do not contain any opinion words will necessarily be evaluated as neutral, because no indications for polarity can be detected. However, it is interesting that only about half of those sentences actually are neutral which is revealed by the low precision of neutral attributes in sentences with 0 opinion words. On the other hand, the recall of neutral sentences becomes very low as soon as at least one opinion word can be detected in a sentence.

The most interesting observation is that the more opinion words a sentence contains, the more likely an attribute within it is evaluated as positive. This can easily be tracked by looking at the development of recall and precision values. While the precision of negative attribute assignments grows with the amount of detected opinion words, the precision of

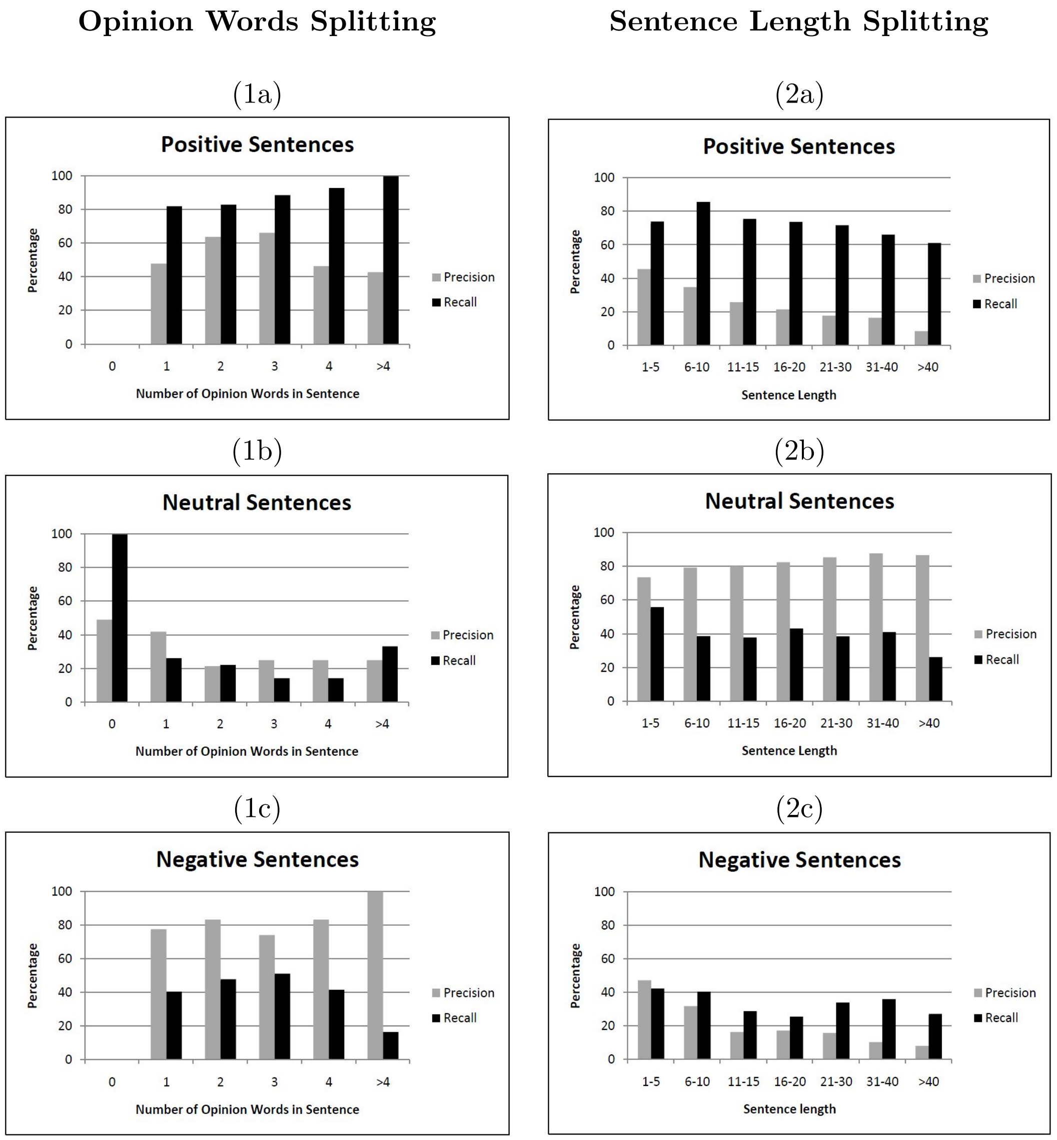


*Figure 7.7: A detailed presentation of evaluation results is provided, separately showing the performance for positive (a), neutral (b) and negative (c) assignments. Furthermore, the performance of opinion assignments can be analyzed in dependence of the number of opinion words present in a sentence according to the graphs on the left hand side (1). Accordingly, on the right hand side (2), a result analysis in dependence of the sentence length is enabled.*

.

positive attribute assignments is decreasing. The recall values, in contrast, behave exactly the other way around.

With respect to the sentence length (see figure 7.7 (2)), the most salient issue is that with increasing sentence length the precision values for both positive and negative sentences decrease.

### Exploring and assessing improvement potentials

The above described observations reveal the weaknesses of the current approach and suggest some improvements.

***PROBLEM 1: Wrong mapping between attributes and opinion words***
Our mapping strategy is based on the simple assumption that the closer an opinion word is to an attribute the more likely it will refer to it. However, it can be shown with simple example sentences (like the one in figure 7.2) that this assumption does not always hold. We therefore tried to further improve the overall accuracy by using an opinion-to-attribute mapping that involves linguistic knowledge.

**Idea: Dependency parser mapping** To take the sentence structure into account, the output of a dependency parser was used as described in section 7.2.4. As can be seen in the obtained confusion matrices in figure 7.6, unexpectedly the results were worse using this more sophisticated mapping. Both methods have in common, that they perform worst predicting "-1". The most frequent error was classifying a "-1" as a "0", which was even more often happening with the dependency parser mapping. Further analysis showed that this is due to the fact that we do not propagate the sentiment over every edge but only over selected ones. This is why in comparison to the statistical method, a few negative opinions are missed and some errors are shifted from positive to neutral. However, follow-up experiments showed that propagating across all edges does in total deteriorate the results. For the neutral opinions, (second column) the dependency parser approach performs slightly better and has less bias towards positive assignments. Finally, for the positive opinions (third column) a big difference can be perceived: The word distance mapping is much better.
One characteristic of the results with word distance mapping is that there is a strong tendency to predict "+1". The dependency parser mapping is better in the sense that it does not have such a strong unidirectional tendency.
In a further in-depth analysis where results were split according to sentence length, it could be observed that for very long sentences the dependency parser method worked better. Apparently, when having a complex sentence structure, e.g. involving relative clauses, the simple word distance measure has problems which can be solved with the dependency parser. Because such sentences are very infrequent in our data, word distance mapping still is the better overall approach. Nevertheless, a combination of both mapping methods, applying the dependency mapping only for sentences with a complex syntactical structure, would be feasible.

***PROBLEM 2: Detecting negative sentiment***
The evaluation of the confusion matrix in figure 7.6 suggests that we have some problems detecting negative sentiments. To find out more about the problem, we investigated the distribution of the opinion words in our test corpus and the role of negation in detail.

**Negation** In our example exactly 23 times the polarity of negative words had been inverted because of preceding negation words and also 23 times this was the case for positive words. Considering the fact, that the overall frequency of detected positive opinion words was much higher (455 positives vs. 187 negatives), they are apparently much less probably negated. Changing the size of the negation window, however, did not lead to better overall results.

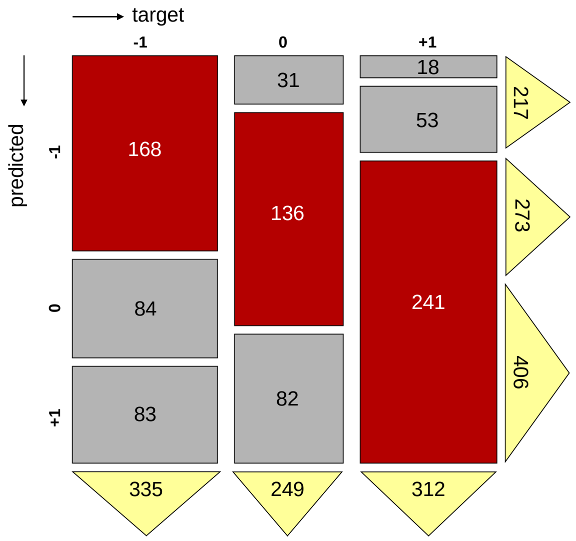


*Figure 7.8: The resulting confusion matrix when negated verbs are taken into account. The method performs far better with respect to the detection of negative opinions without noteworthy negative effects on the detection of neutral and positive opinions.*

**Opinion lists** In total, 136 different positive opinion words were found at least once and 86 different negative opinion words. The total amount of positive opinion words found was 455 while the total amount of negative opinion words was only 187. It is evident that many more and also more different positive opinion words have been found although (1) the negative word list contains many more words than the positive list and (2) the benchmark annotation contains slightly more negatively connoted attributes than positive ones. The problem might be solved by adding more negative opinion words to the corresponding list. On the other hand, the negative opinion word list already is much larger than the positive list, which brings up the idea of searching opinions that are not stated by the explicit usage of opinion words. When skimming through the test data in search for reasons, we could observe that often negated verbs appeared in such sentences. Basically, reviewers expressed their inability to do things with the product that should be possible. In addition, they complained about the product or manufacturer not acting as supposed. Due to the generality of these negated verbs they had not been included in the opinion word list. Consequently, a simple heuristic was designed to take negated verbs into account.

**Idea: Incorporating negated verbs** For every verb within the negation window an opinion value of “-0.5” was introduced, with verbs being identified according to their part-of-speech tag. By assigning them only half of the usual “-1” for negative opinions, their influence was limited, which yielded better results than counting them as full negative opinion signals. Thus, negated verbs became mainly decisive in formerly neutral sentences, which is in concordance with the observation that assigning “0” to negative attributes was the most frequent error.

It was surprising to see the strong beneficial impact this simple heuristic had (figure 7.8). The main goal was reached: A lot more “0” became “-1” (first column) while the good recall for positive opinions (second column) stayed practically the same and the recall of neutral opinions was barely affected (second column).

***PROBLEM 3: Detecting neutral sentiment when opinion words are present***
When analyzing the performance with respect to the number of opinion words (see above), we observe that as soon as a sentence contains at least one opinion word, the recall is very low for neutral sentences. This is caused by the fact that our algorithm will always perform a mapping if at least one opinion word can be found within the predefined window around the attribute. With a cutoff value of 10, the current window size is pretty big taking into account large parts of a sentence. When opinion words can be found, the algorithm will only predict neutral, if the weights of positive and negative opinion words erase each other which is rarely the case if their distance to the attribute is taken into account.

**Idea: Modification of the cutoff value** As outlined previously, the rather large cutoff value of 10 in many cases forces a decision for a positive or negative polarity instead of leaving it to neutral. However, shrinking the cutoff value is not a good option because the overall accuracies deteriorate with lower cutoff values: 54.576% for cutoff 9, 52.679% for cutoff 7, 50.335% for cutoff 5 and 46.205% for cutoff 3. On the other hand the accuracy does not increase noteworthy with larger cutoff values. Thus, we can conclude that for our test data there is no real improvement potential in varying the cutoff value.[3]

***PROBLEM 4: Missed opinion words***
When reviewing the sentences in which an attribute was commented on whose sentiment was completely missed by our algorithm, it became evident that despite of its huge sizes, the utilized opinion word list did not contain all of the appearing opinion words. In some cases this was due to the domain dependency of opinion words: The adjective "long" e.g. is negative when it refers to a loading time and positive when a battery lasts long. In other cases, informal, colloquial and sometimes vulgar expressions were not contained in our list. Another problem that called our attention was that negation words in many cases contained typos. Instead of "don't" i.e. a considerable amount of people wrote "dont" or "don"t".

**Idea: Manual enhancement of the opinion word list** All missing opinion words that were not domain-dependent were added to our lists, resulting in 20 new negative and 64 new positive entries. In addition, the verb "to stop" was interpreted as negation signal word for verbs (e.g. "stop working" is negative).
Together with the previously applied improvements these changes pushed the overall accuracy to 63.1%. The significant improvement suggests that revising the opinion word list is a one-time effort that is worth undertaking (either manually or by using some of the automatic methods that were pointed to in the related work section). It has to be expected that additionally taking domain-dependent opinion words into account would further push the results.

***PROBLEM 5: Missed attributes***
Out of 996 opinion assignments in the benchmark 100 refer to an implicit attribute. That means they can not be detected automatically on sentence-basis because the product attribute does not appear in the respective sentence. This may be caused by the usage of a synonym or an anaphora as well as a circumscription of the attribute. In our test corpus, 58 times an implicit attribute was mentioned negatively, 42 times positively. We do not

[3] Note that the values would change in a scenario in which the class of the neutral attributes is bigger. In this case, it certainly would be favorable for the overall accuracy to reduce the cutoff value but would result in missing many positive and negative attributes.

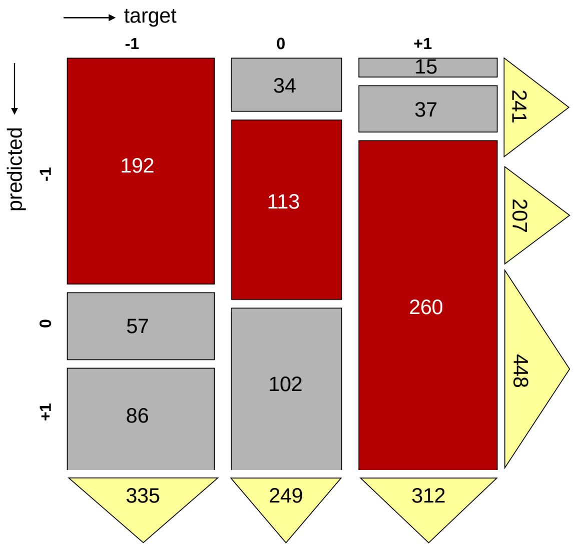


*Figure 7.9: The resulting confusion matrix when all the suggested improvements are included.*

have ideas at hand how to deal with the problem of implicit attributes. However, it is interesting to see how often they occur.

***Overall improvement***

In total, we were able to improve the accuracy values by systematically learning from the in-depth analysis from 55.1% to 63.1%. Much of the improvement was gained by improving the detection of sentiment. Especially, the identification of negative comments was significantly improved by interpreting negated verbs as negative opinion signal words. Furthermore, our experiments suggest that including linguistic knowledge about the sentence structure (by using the result of a dependency parser) does not improve the mapping in the general case, but only for complex sentences. Figure 7.9 shows the resulting confusion matrix. In an experiment that is based on subjective sentences only (as is usually done in related work), we get accuracy values of more than 80%.

## 7.4 Application: Visual analysis of customer review data

The online version of the Encyclopedia Britannica defines the term *public opinion* as

> *"an aggregate of the individual views, attitudes, and beliefs about a particular topic, expressed by a significant proportion of a community. Some scholars treat the aggregate as a synthesis of the views of all or a certain segment of society; others regard it as a collection of many differing or opposing views."* [33]

This definition indicates that a visual analytics system that is dedicated to supporting the analysis of documents with respect to the opinion that is expressed in them needs to fulfill the following requirements: (a) a "significant portion" of the community has to be considered, asking for efficient automatic and visual support in the process, (b) it is the global view that is interesting ("the aggregate") and not first and foremost the single opinion, but on the other hand (c) it has to be taken account that the public opinion is composed of "a collection of many differing or opposing views" which should be reflected in the visual representation.

In the following, several application examples are presented that visually explore a set of documents with respect to the opinion that is expressed in them. Thereby, the above mentioned requirements are taken into account by designing systems that are scalable and show a global view on the data, but yet provide as much details as possible.

### 7.4.1 Visual summary reports for review analysis

With the rapid growth of Internet technologies, there are large numbers of customer reviews on the websites. [103] reports that *"81% of Internet users (or 60% of Americans) have done online research on a product at least once"*. Furthermore, they state that customers are willing to invest significantly more for a 5-star-rated product than a 4-star-rated product. Hence, reviews can have a large impact on the profit margin that a company is able to realize with a specific product. While the internet users were generally satisfied with their online product research, *"at the same time, 58% also report that online information was missing, impossible to find, confusing, and/or overwhelming"* [103].

Often customers are asked to give a total score (see e.g. the webpage of amazon.com). Yet, this score does not necessarily reveal the product's true quality and may provide misleading recommendations. An attribute of a product that was important for customer A and therefore had an important impact on the total score that this customer gave might be irrelevant for customer B. Thus, the latter does not mind if this feature is not available in the product or is deficient. Similarly, it is not enough for a company to know which of their products customers liked best or least. In order to learn from the feedback and be able to improve the products, they need to know which attributes of the product their customers liked and disliked.

The technique that was introduced in 7.2.2 transforms the semi-structured data into a structured format. Remember that the approach generates a feature vector for each review that holds detailed information about which attributes were liked or disliked by a customer. However, if we stopped here (as many related approaches do), this would leave the user with thousands of feature vectors. There is clearly a need for supporting the last step of the analysis process - the interpretation of the results. Such an interpretation cannot be done automatically, because what is considered as important is differs from user to user. In the following, our visual summary reports are introduced - a compact representation of thousands of reviews that yet provides enough detail to derive a comprehensive insight.

#### Summary reports

Each line in the summary report represents one group of reviews (e.g. all the reviews for one product or brand). The table structure contains one column per attribute. For each attribute extracted by our automatic algorithm, it is shown whether it belongs to the category of attributes with a positive tendency (blue) or the category with a negative tendency (red). The size of the inner rectangles is determined by the percentage of reviews that comment on the attribute signaling the importance that the analyst should give to this attribute in his or her evaluation. Color is mapped to the percentage of positive or negative opinions, respectively. Using our automatic analysis method, we calculate the average percentage of positive comments per attribute and use this as a threshold. Attributes whose percentage of positive comments is above that threshold exhibit a positive tendency compared to the other attributes (color = blue). The ones that are below the threshold show a negative tendency (color = red). The stronger the positive or negative tendency is the darker the color value becomes. The intervals for the four shades of blue/red tones are determined by the quantiles of the set of positive or negative attributes.

By recalculating the threshold value for each data set instead of using a fixed one, we compensate for the shift that is caused by the fact that some products are generally commented on more positively than others. Please note that the basis for the calculation of the threshold and the quantiles is always the whole set of product reviews that are displayed to ensure comparability across the different lines.

The approach that is most similar to our work is presented in [87]. The authors suggest to use traditional bar charts to visualize how many positive respectively negative statements exist within the document corpus. The advantage of our technique is that it is much more scalable with respect to the number of attributes and the number of products that can be displayed. Furthermore, our matrix-based visualization simplifies the comparison between both different attributes and different products.

### Application to reviews from amazon.com

Figure 7.10 shows a visual summary report of reviews from amazon.com on three different printers (in total 1876 reviews were analyzed). For printer 1 we additionally show the result for two different printer types separately. This allows a detailed analysis of strengths and weaknesses of specific printer families. It can be seen that there are some attributes that the customers are generally satisfied with. This is true for the general attribute *printer* but also for other attributes such as *copy, photo, print or quality.* The latter ones suggest the assumption that the quality of the prints is not that much of a distinguishing factor between the different brands. On the other hand, we found that there is a lot of negative feedback for the attributes *cartridge*, *ink*, *month*, *software* and *unit.*

Of course, such attribute terms can only be considered as hints that point at certain problems that the analyst should have a closer look at. This is especially true for terms such as *month* or *unit* which are not self-explanatory. In analyzing the data set, the size of the inner rectangle in combination with the color gives an idea to the analyst of how severe the problem might be. For example, in Figure 7.10 the *software* column sticks out because of the large size of the inner rectangles (signaling that many customers commented on it) in combination with the dark red colors (which means that a relatively large number of customers was dissatisfied with this aspect). For a customer it will be of special interest to observe the differences in the evaluation of the three printers. Strengths and weaknesses of the specific printers as seen by other customers become easily visible in the summary report visualization.

A second example is shown in figure 7.11. In this case, reviews on PDAs were analyzed. For some attributes, clear differences in the average rating can be perceived. This is, for example, the case for the attributes *keyboard* and *map.* Some features are only present in a subset of the products. If an attribute was rated zero times for a product, the corresponding cell is colored in white. For example, the GPS navigation system *tomtom* only seems to be included in two of the analyzed PDAs. For those two products, the attribute was

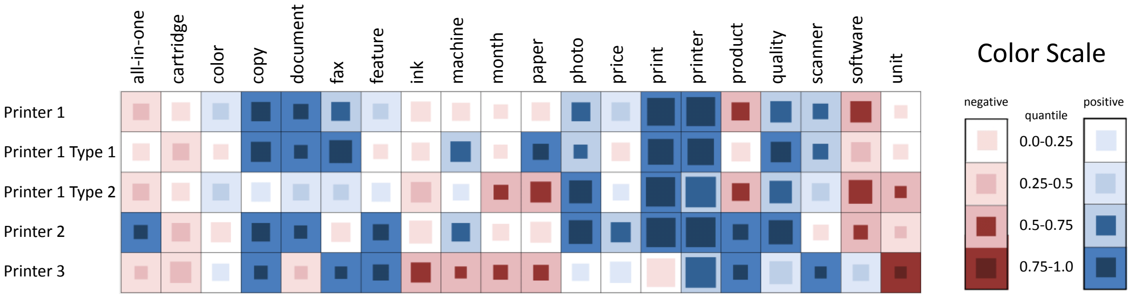


*Figure 7.10: Summary Report of printers: Each row shows the attribute performances of a specific printer. Blue color represents comparatively positive user opinions and red color comparatively negative ones (see color scale). The size of an inner rectangle indicates the amount of customers that commented on an attribute. The larger the rectangle the more comments have been provided by the customers.*

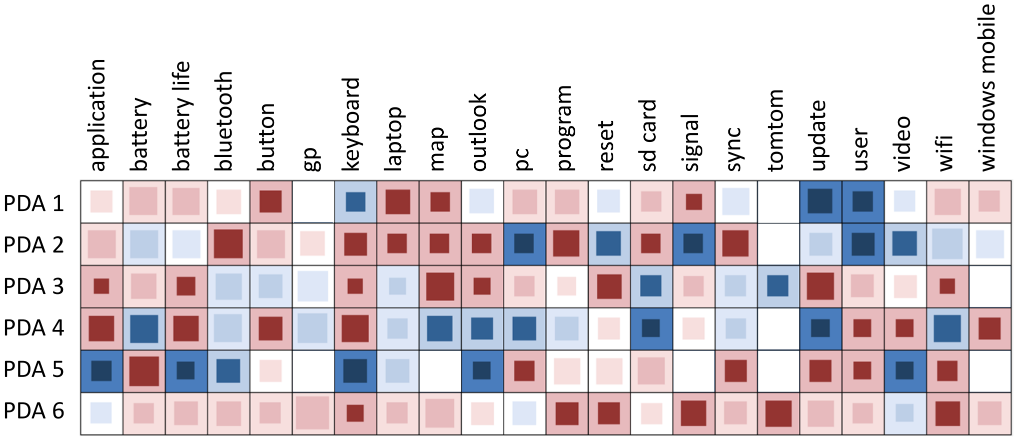


*Figure 7.11: Summary Report of reviews on PDAs. Note that the size of the inner rectangles was normalized with a square root function. If an attribute was not commented on for one of the products, its cell is left white. The color scale is the same as in figure 7.10.*

commented on many times with a very contrasting overall result. Besides analyzing the differences between the products with respect to certain attributes, comparing different products (rows) across all lines can be interesting as well. For example, it can be perceived that PDA 6 is the one that got the lowest overall rating. The line above it (PDA 5) is eye-catching, because its rating is often reverse compared to the other lines. Attributes that were rated positively for this product are often rated negatively for the other products and vice versa.

### 7.4.2 Cluster analysis on reviews

For marketing purposes, it is interesting to see which subgroups of customers with similar opinions exist. Our second application example in the context of opinion analysis explores feedback data with respect to groups of customers with a similar opinion. To the best of our knowledge currently no technique exists that analyzes and visualizes customer feedback with respect to clusters of reviews in which similar opinions are expressed.

To find the different groups of customer opinions, we apply a hierarchical clustering algorithm (Complete Linkage) and then project each cluster representative (on a user-selected hierarchy level) in 2D space using multi-dimensional scaling as a dimensionality reduction method. Each cluster is then visualized using a thumbnail image that depicts for each attribute the percentage of reviews in the cluster that commented on it, split up into negative and positive comments. The number of reviews that the cluster contains is mapped (non-linearly) to the size of the thumbnails and to the grey tone of the corresponding voronoi cell.

Figure 7.12 shows an example in which clusters of customers of printer 1 are shown. The largest cluster is the one that summarizes all the reviews that did not criticize any attribute but were satisfied with the printer. On the other hand there is also a group of customers that disliked most attributes (cluster 2). More interesting however are the clusters that summarize the reviews with a rather differentiated opinion. For example, cluster 3 aggregates the reviews that had an overall positive tone but in which also some critical aspects (mostly because of the scanner) are mentioned. Finally, clusters 4-6 show user groups with a clearly differentiated opinion about the product.

### Distance function for the clustering algorithm

Key for the detection of expressive clusters is the use of a meaningful distance function that is able to measure the similarity of reviews. In the following, we derive a first heuristic for such a distance function. Note that it could easily be replaced with alternatives in the future.

Given a feature vector r = ($i_1$, $i_2$, ..., $i_n$) for each review as described in section 7.2.2, we need a distance function that is able to discern the similarity of two reviews with respect to the opinion that is expressed in them. The similarity between two reviews is increased if they both comment on an attribute and both agree in their opinion about this attribute. Likewise, if an opposite opinion is expressed about an attribute in two reviews, the similarity value between those two reviews has to decrease. This leads us to the first part of our distance function that counts how often the two reviews state opposing opinions on an attribute (see equation 7.4):

$$f(r_1, r_2) = \sum_{k \in K} g(i_k^{(r_1)}, i_k^{(r_2)}), \tag{7.4}$$

where

$$K = \{i : i^{(r_1)} \neq 0 \wedge i^{(r_2)} \neq 0\}$$

$$\text{and } g(i_k^{(r_1)}, i_k^{(r_2)}) = \begin{cases} 1 & \text{if } \operatorname{sign}(i_k^{(r_1)}) \neq \operatorname{sign}(i_k^{(r_2)}), \\ 0 & \text{else.} \end{cases}$$

So far, the positions in the feature vectors in which at least one of the reviews does not comment on the attribute do not contribute to the distance. Can they be ignored? Consider the following two feature vectors:

```
R1:  [ +1  0 +1  0 +1  0 +1  0 +1  0 ]
R2:  [ +1 -1  0 -1  0 -1  0 -1  0 -1 ]
```

In this pair of feature vectors, only the first attribute is commented on by both reviewers. As they both mention the attribute positively, the above introduced distance function would consider the two reviews as stating a very similar opinion (distance = 0). However, it is obvious that those reviews do not express a similar opinion on the product. Thus, we also have to take the attributes into account that only one of the reviewers commented on. We do so by calculating for both feature vectors the percentage of positively mentioned attributes, taking only the attributes into account that the other reviewer did not comment on, since the ones that both commented on already contribute to the first part of our distance function. The same is done with the negatively mentioned cases. By setting the values in relation to each other, we measure the difference in their general opinion about the remaining attributes (see equation 7.5).

$$h(r_1, r_2) = \frac{1}{2} \cdot (|\frac{L_{1+}}{L_1} - \frac{L_{2+}}{L_2}| + |\frac{L_{1-}}{L_1} - \frac{L_{2-}}{L_2}|), \tag{7.5}$$

where

$$\begin{aligned} L_1 \;\; &= |\{i \; : \; i^{(r_1)} \neq 0 \; \wedge \; i^{(r_2)} = 0\}| \\ L_{1+} &= |\{l \in L_1 \; : \; l > 0\}| \\ L_{1-} &= |\{l \in L_1 \; : \; l < 0\}| \\ L_2 \;\; &= |\{i \; : \; i^{(r_2)} \neq 0 \; \wedge \; i^{(r_1)} = 0\}| \\ L_{2+} &= |\{l \in L_2 \; : \; l > 0\}| \\ L_{2-} &= |\{l \in L_2 \; : \; l < 0\}| \end{aligned}$$

Finally, both parts of the distance function are weighted with the number of attributes that contribute to it, leading to the following distance function (equation 7.6):

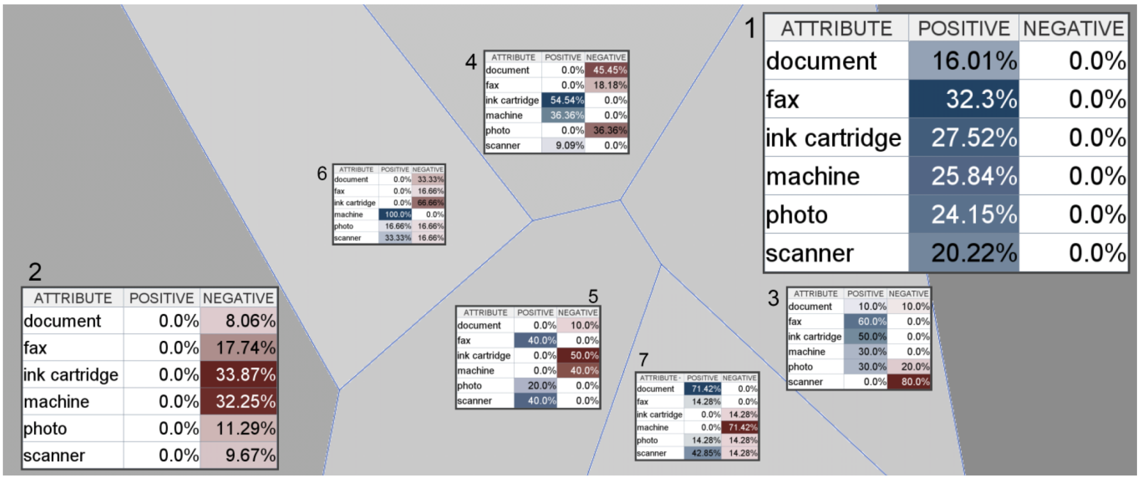
$$dist(r_1, r_2) = \frac{|K|}{|K| + L_1 + L_2} \cdot f(r_1, r_2) + \frac{L_1 + L_2}{|K| + L_1 + L_2} \cdot h(r_1, r_2) \cdot w \tag{7.6}$$

where the factor $w$ permits to balance the influence of the two aspects of the distance function.

Please note that the cluster analysis that is presented in this chapter has to be considered as a proof-of-concept. A proper evaluation of the distance function is still to be done.

## 7.5 Application: Visual analysis of news data

In the following, the sentiment detection algorithm of section 7.2.1 is applied to news data. It turned out that the algorithm does not measure the same text property as it does for customer reviews. This is partly because of our opinion dictionary that considers terms like *to kill* or *to accomplish* as bearing an opinion (which is strictly speaking not true as they may be associated with positive or negative issues but do not at first convey an opinion). On the other hand, it has to be considered that news articles express opinions differently than customer reviews. Whereas opinions are explicitly mentioned in reviews and there are not many side issues discussed, newspapers often express their opinion more subtle, e.g. by reporting about certain facts, while others are disregarded.



1

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 16.01% | 0.0% |
| fax | 32.3% | 0.0% |
| ink cartridge | 27.52% | 0.0% |
| machine | 25.84% | 0.0% |
| photo | 24.15% | 0.0% |
| scanner | 20.22% | 0.0% |

2

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 0.0% | 8.06% |
| fax | 0.0% | 17.74% |
| ink cartridge | 0.0% | 33.87% |
| machine | 0.0% | 32.25% |
| photo | 0.0% | 11.29% |
| scanner | 0.0% | 9.67% |

3

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 10.0% | 10.0% |
| fax | 60.0% | 0.0% |
| ink cartridge | 50.0% | 0.0% |
| machine | 30.0% | 0.0% |
| photo | 30.0% | 20.0% |
| scanner | 0.0% | 80.0% |

4

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 0.0% | 45.45% |
| fax | 0.0% | 18.18% |
| ink cartridge | 54.54% | 0.0% |
| machine | 36.36% | 0.0% |
| photo | 0.0% | 36.36% |
| scanner | 9.09% | 0.0% |

5

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 0.0% | 10.0% |
| fax | 40.0% | 0.0% |
| ink cartridge | 0.0% | 50.0% |
| machine | 0.0% | 40.0% |
| photo | 20.0% | 0.0% |
| scanner | 40.0% | 0.0% |

6

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 0.0% | 33.33% |
| fax | 0.0% | 16.66% |
| ink cartridge | 0.0% | 66.66% |
| machine | 100.0% | 0.0% |
| photo | 16.66% | 16.66% |
| scanner | 33.33% | 16.66% |

7

| ATTRIBUTE | POSITIVE | NEGATIVE |
|---|---|---|
| document | 71.42% | 0.0% |
| fax | 14.28% | 0.0% |
| ink cartridge | 0.0% | 14.28% |
| machine | 0.0% | 71.42% |
| photo | 14.28% | 14.28% |
| scanner | 42.85% | 14.28% |

*Figure 7.12: Scatterplot of customer reviews on printers: Seven main opinion clusters have been identified and mapped in a 2D space, each represented by one thumbnail. The more reviews a cluster contains, the larger its thumbnail is displayed. Positive opinions are highlighted in shades blue, the negative ones accordingly in red. The color brightness is mapped to the percentage of reviews within a cluster that share a certain opinion.*

Thus, we can conclude that a measure that has been successfully applied in one domain (or application context) cannot necessarily be used in another domain as well. Nevertheless, our analysis provided us with interesting results. If applied to news articles, the sentiment detection algorithm measures a property that could be called the *context polarity* of a news article. This means that we determine whether the context that something was mentioned in can be considered as being positive or negative.

### 7.5.1 Visualizing 4 weeks of *The Telegraph*

Figure 7.13 shows the title pages of 4 weeks of *The Telegraph*, an English newspaper. Shades of red and green are used to highlight the number of positive or negative opinion signal words in the sentences. Note that this very simple algorithm does not take negation into account.

Keeping the layout and including the pictures in the representation, greatly facilitates the analysis. On some days almost all the news on the title page are negative, whereas on other days reporting is mixed. Interestingly, the Queen was always mentioned in a positive context in those four weeks.

### 7.5.2 Visual exploration of news feeds using polarity and geo-spatial analysis

Excess amount of information is generated each day on the internet, making processing of the content very difficult for the individual. Global news agencies, such as The Associated Press (AP), Reuters and Agence France-Presse (AFP), provide media companies with news reports from all over the world. This content is then duplicated, enriched with commentary and opinion. Additionally, news are filtered according to importance or interest of the editorial team.

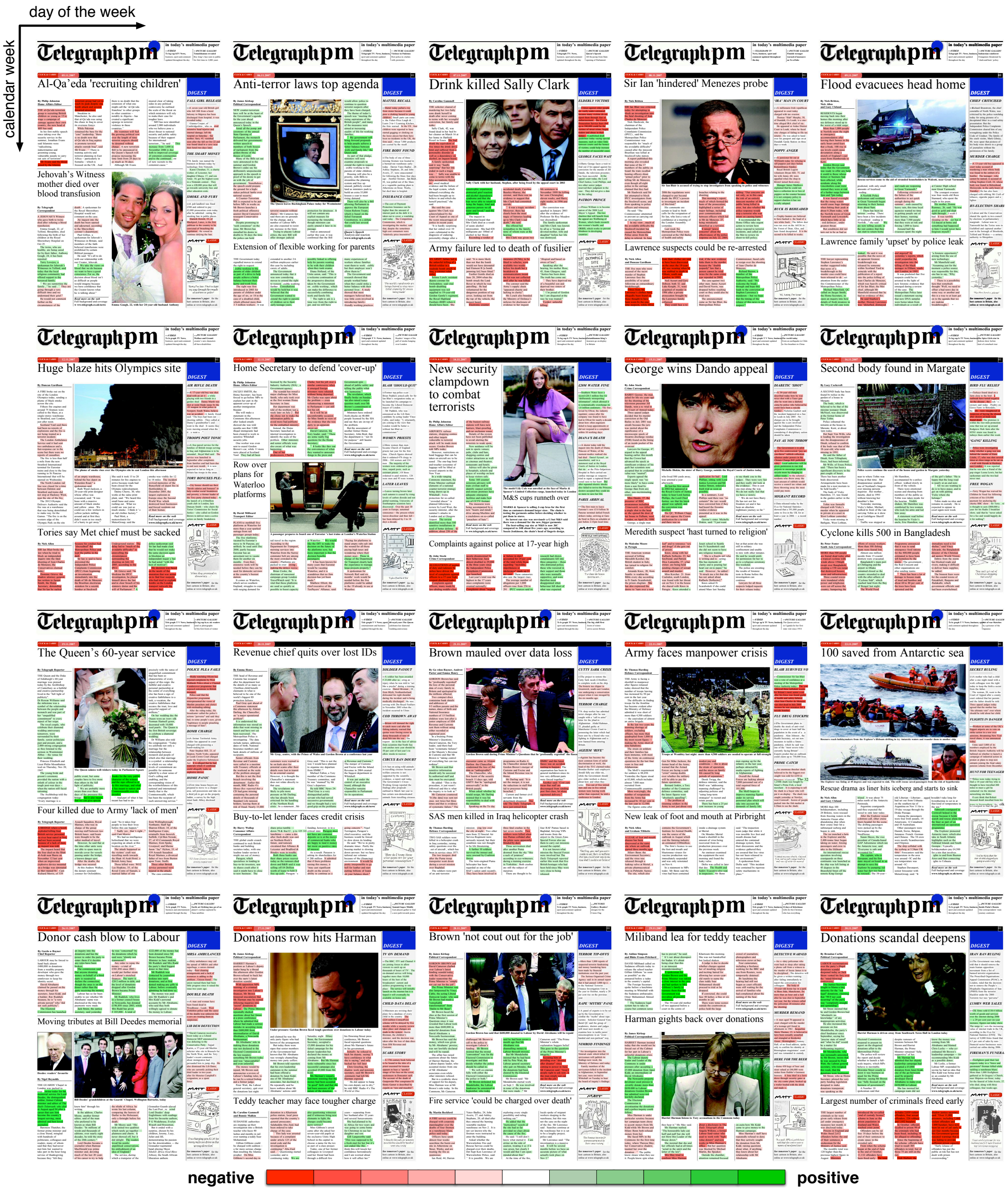

day of the week
calendar week
Al-Qa'eda 'recruiting children'
Jehovah's Witness mother died over blood transfusion
Anti-terror laws top agenda
Extension of flexible working for parents
Drink killed Sally Clark
Army failure led to death of fusilier
Sir Ian 'hindered' Menezes probe
Lawrence suspects could be re-arrested
Flood evacuees head home
Lawrence family 'upset' by police leak
Huge blaze hits Olympics site
Tories say Met chief must be sacked
Home Secretary to defend 'cover-up'
Row over plans for Waterloo platforms
New security clampdown to combat terrorists
M&S cups runneth over
Complaints against police at 17-year high
George wins Dando appeal
Meredith suspect 'hast turned to religion'
Second body found in Margate
Cyclone kills 500 in Bangladesh
The Queen's 60-year service
Four killed due to Army 'lack of men'
Revenue chief quits over lost IDs
Buy-to-let lender faces credit crisis
Brown mauled over data loss
SAS men killed in Iraq helicopter crash
Army faces manpower crisis
New leak of foot and mouth at Pirbright
100 saved from Antarctic sea
Rescue drama as liner hits iceberg and starts to sink
Donor cash blow to Labour
Moving tributes at Bill Deedes memorial
Donations row hits Harman
Teddy teacher may face tougher charge
Brown 'not cout out for the job'
Fire service 'could be charged over death'
Miliband lea for teddy teacher
Harman gights back over donations
Donations scandal deepens
Largest number of criminals freed early
negative
positive

*Figure 7.13: Positive/Negative Sentence Highlighting. The visualization shows the title pages of The Telegraph newspaper of November, 2007. Each row represents a calendar week (from week 44 to 48) and each column a day of the week (from Monday to Friday).*

In the following, the visual exploration of a news feed from the Europe Media Monitor (EMM) [7, 32] is presented. The Europe Media Monitor is a news aggregation system which monitors over 2500 news sources, collecting 80,000 - 100,000 news articles per day in 42 languages. Clearly, some automatic support is needed to make use of this rich information resource.

One of the questions when analyzing the news feed is how the different groups (e.g. countries) discuss different topics. Do they share the same opinion? Are there clear differences between some countries? Does it depend on the topic how much they agree with each other? Which special observations can be made? What is challenging in this case, is that we cannot say clearly, what we are looking for. The fact that our dataset is not static, but that we are working with a data stream, aggravates the problem. Knowing what would be interesting to look at today does not necessarily mean that this would also be a good view for tomorrow's news. This calls for a detailed visualization that makes as few as possible assumptions in advance. We decide to apply the Literature Fingerprinting technique on several resolution levels to analyze the data. Furthermore, the geographic aspect is taken into account by using a geo-visualization.

### Visual polarity analysis

Figure 7.14 shows a Literature Fingerprint for about three weeks (May 11th - May 28th 2009) of English newspaper articles from all over the world. In the left column each pixel represents the set of news articles for a single country. A block of pixels contains all the articles that belong to a specific topic. (Our topics are *agriculture*, *security*, *sports*, *swine flu*, and *terrorism*). Color is mapped to the average sentiment score of the articles that are represented by the pixel (see color scale at the right). In the right column, the data is shown on an aggregated resolution level. In this case, each pixel represents a single article and the articles are first grouped according to the country they belong to and then according to the topic they report on. Again, color is mapped to the sentiment score but this time it represents the score for a single article.

Looking at the left column of figure 7.14, it is easy to see that there are clear differences between the topics with respect to their fundamental tone. While *security* and *terrorism* show a negative trend for most countries, the opposite is the case for *agriculture* and *sports*. Articles that report about *swine flu* in total seem to be in the middle of the scale. While some topics show big differences in sentiment between the single countries, *agriculture* is a topic that is almost homogeneously seen as positive.

The advantage of the representation in the right column is that not only an average score is depicted. In the higher resolution level, it can also be seen how many articles contributed to the average value and how homogeneous the reporting is with respect to the sentiment that is expressed.

In the last line of the right column, three kinds of patterns can be perceived: countries for which almost all pixels are colored in shades of red (negative), countries which are homogeneously shaded in blue or green, and finally countries in which all colors of our color scale occur. Among the ones whose articles are homogeneously classified as talking negatively about terrorism are Australia, Croatia, and the Cayman Islands (see enlarged depiction at the bottom of the figure). A closer analysis shows that their articles report primarily on terrorism in other countries. Of course, those terroristic activities are clearly

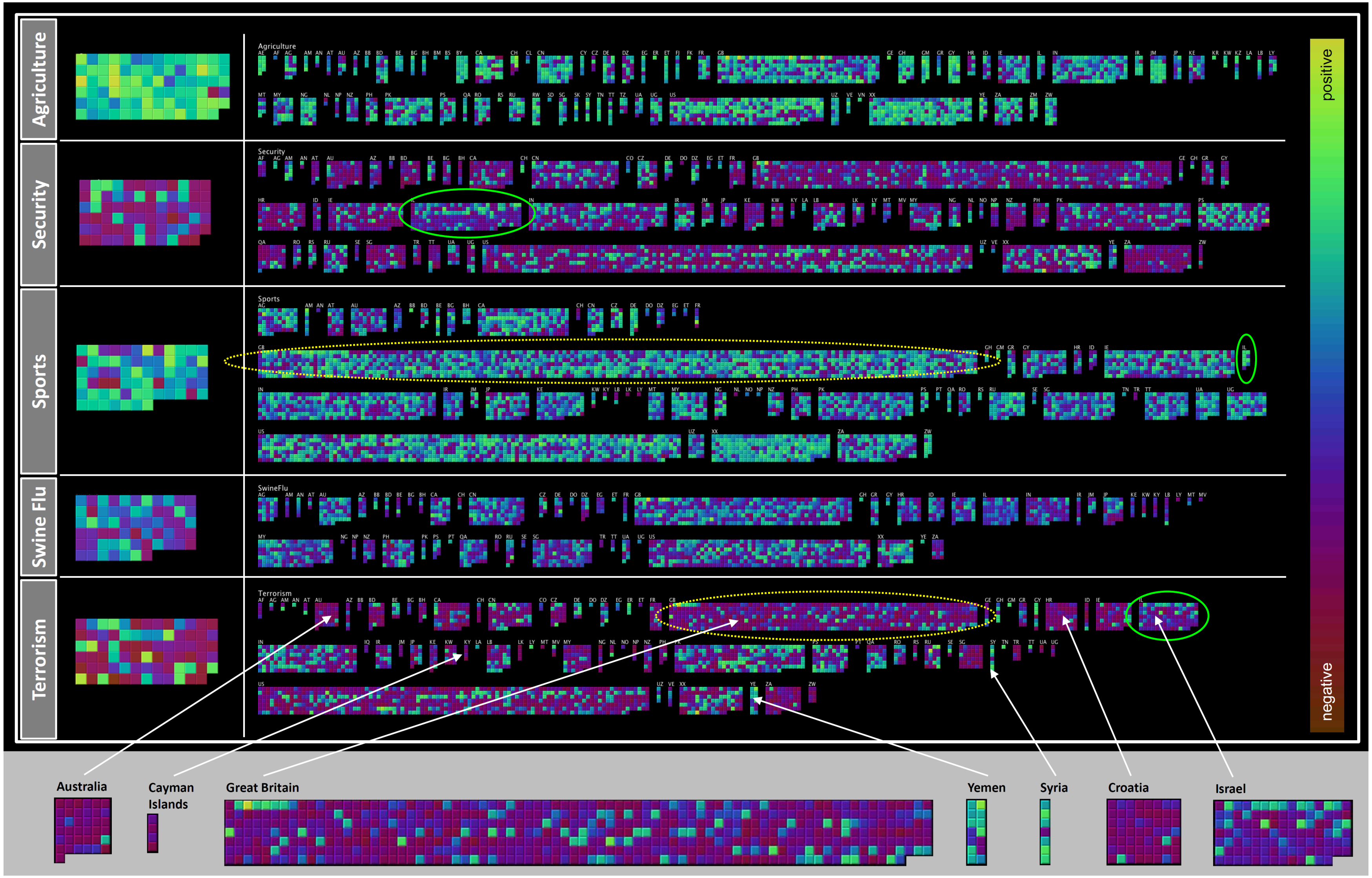


*Figure 7.14: Context Polarity Analysis on news articles. In the right column, each pixel represents a single article. Its color is mapped to the calculated polarity score. Articles that were published in the same country are grouped together into blocks of pixels. In the left column, each pixel represents the average polarity score for a set of articles of the same country. We display five different news categories.*

damned. Countries in which a concrete danger of terroristic acts exists, usually show a multi-colored picture in our visualization with an in total negative tendency (see e.g. Great Britain or Israel). The reason for this is that also political speeches or activities (such as cooperations with other countries) that talk about fighting against terrorism are included. This also nicely exemplifies how our algorithm works. The latter articles are dominated by security-related terms, measures against terrorism, and optimistic perspectives for the future and thus our algorithm classifies them as positive, because the connotation of those terms is positive. This means that the used algorithm would not distinguish between an article that agrees with those political speeches and another one that cites them but afterwards disassociates itself from the message. Finally, we were surprised to see that almost all articles in this category of the Syrian Arab Republic and Yemen are clearly classified as positive. Reading through the articles revealed that in those days the foreign ministers of the Islamic countries met. Among other things, they discussed ways to preserve Islamic values and the Islamic culture despite of experienced terroristic activities. For the participating countries this was the major topic in those days and the optimistic tone of the conference (also praising much their own countries efforts and perspectives) explains the large amount of positive reports in the terrorism category.

Finally, we can also analyze the articles across topics. It is interesting to see that Great Britain has about twice as many articles in the category *sports* than in the category *terrorism* (see yellow dotted circles). Opposite to that, Israel has only very view articles in the *sports* category compared to the amount of articles in the categories *terrorism* and *security* (see green circles in figure 7.14).

### Geo-spatial analysis of the articles with respect to polarity

Our news feed also provides us with information about geographic attributes. First, the news origin refers to a news agency located in a country or state, from which the news was published. This information is automatically mapped to the location of the news agency. Second, requiring more sophisticated tagging, is the location of the news' topics themselves. For the purpose of geographically tagging the location of a news item, the full text article is scanned for city, state, and country names. Consequently, one news item could have more than one location, when more distinct places are mentioned. In practice, however, the majority of news items has only one annotated geo-location.

In collaboration with a colleague working in the area of geo-spatial analysis, a visual representation of the polarity results with respect to the geographical information was generated. As a visualization technique, a pixel-based approach was used that avoids overlap by arranging the pixels circularly around the original location. Furthermore, the map is distorted with the help of a cartogram algorithm to give more space to highly populated regions [10, 70]. As the technique is not a contribution of the thesis, it is not further detailed here.

Figure 7.15 shows the resulting plot. The data were obtained in the time period between May 11 and June 7, 2009. The figure represents spatial analysis of news feeds showing the origin (upper left, where the news were published) and location (upper right, where the event mentioned in the news took place) for topics belonging to two categories: Security and Terrorism. The news originate mainly in Europe and in the US, and are reporting on the US, Europe, but also a lot on the Middle East and Asia. Spatial analysis of news feeds, showing the polarity score of topics related to security and terrorism (bottom

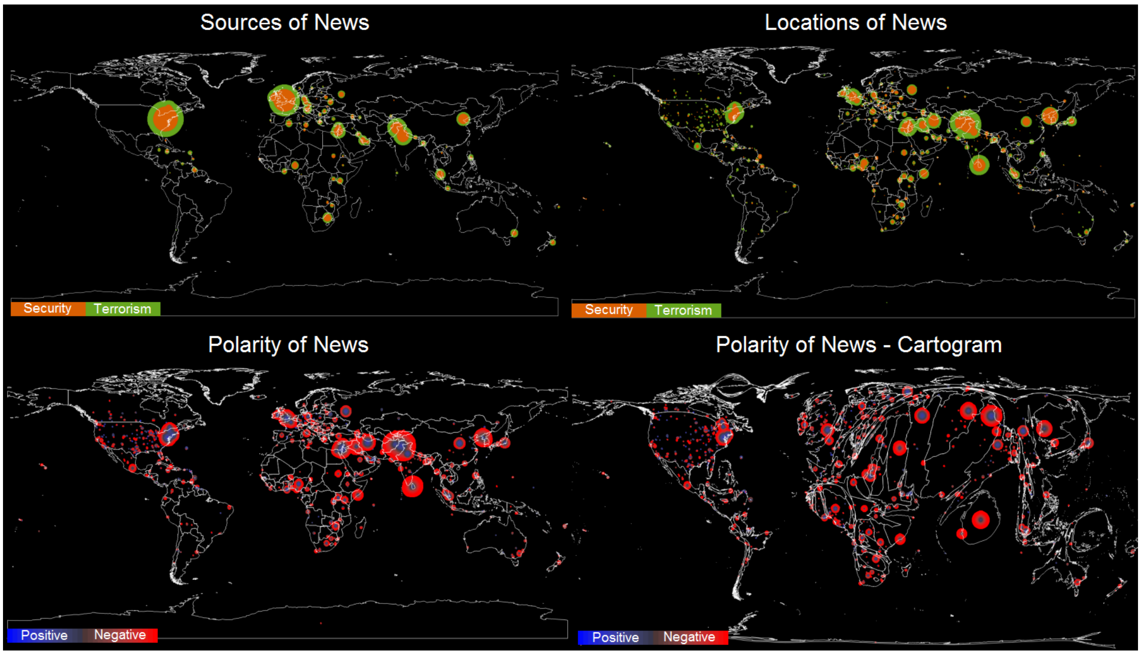


*Figure 7.15: Spatial analysis of news collected from May 11, 2009 until June 7, 2009. The sources (upper left) and the locations (upper right) of news are represented for two categories - security (orange) and terrorism (green). The polarity of the news articles (lower left) is shown using a bi-polar color map, reaching from blue (positive polarity) to red (negative polarity) on a logarithmic scale. This polarity map is also shown in a Cartogram representation, enhancing regions of importance, where the number of news items corresponds to the area of each country*

left), is shown using a bi-polar color map, having red for negative, and blue for positive news with increasing intensity. The news mainly report on Middle East, Central Asia (especially on the events in Sri Lanka) and North Korea in the particular time period. Although the majority of these news is negative in their tonality, there are some positive reports on successes in the fight on terrorism. The Cartogram representation enhances the area of these important locations.

## 7.6 Summary and future work

This chapter dealt with document analysis tasks that are centered around the quasi-semantic properties *sentiment* and *opinion*. To approximate these properties, we used measures that were proposed in related work. An in-depth evaluation of the measures revealed improvement potential which allowed us to develop the methods further. The usage of visualization techniques in the evaluation process proved as very beneficial, because it eased to comparison of the results of different experiments. In the future, the visualization of the confusion matrices might be further enhanced by adding information about the flow of the data from one run to the next. Figure 7.16 depicts two mockups for a single column in the confusion matrix that show how this extended version might look like.

In the presented case studies, customer feedback data and collections of news articles

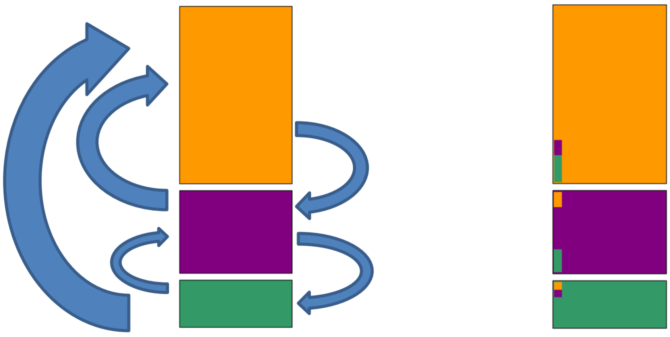


*Figure 7.16: Two mockups for a single column in the confusion matrix that show how an extension with flow information could look like. Left: Arrows are used to show the changes in the assignment. The thickness of the arrows represents the amount of data that changes its class from one run to the other. Right: The same data is visualized by adding small rectangles at the side that show where the data originally comes from. Note that there is no horizontal flow in the confusion matrix (if the target class is assigned to the x-axis).*

were analyzed. We could show that the novel summary reports are a useful means of comparing different products with respect to their strengths and weaknesses. Although, the technique was developed in the context of opinion analysis, it could be applied to all kind of data sets that are available in a structured, table-like format and require an aggregation and later comparison of subsets of the data. Furthermore, it was argued that cluster analysis of customer reviews requires a special distance function and a first heuristic for such a function was proposed. One of the next steps should be the analysis of opinions over time - a topic that is already worked on by some colleagues (see e.g. [145]). The analysis of news revealed that a quasi-semantic measure that was developed for one domain cannot necessarily be applied to another domain. Nevertheless, applying the sentiment measure to this new domain let to interesting results because it approximates a quasi-semantic property that could be considered as the *context polarity* that a topic is mentioned in. Future work includes the development of a method that measures opinions in news articles directly.

It is worth noting that some of the visualizations that were used in this chapter are not restricted to the domain of document analysis. As soon as the semi-structured textual data is transformed into a structured format, it is often possible to use existing visualization (and data mining) techniques.

# 8

# Concluding Remarks

**Contents**

This chapter concludes the thesis by putting the presented work into the larger context of semantic document analysis that is based on visual analysis methods. Thereby, the contributions of the thesis are summarized and further research questions are identified. Note that the future work that specifically relates to one of the application examples was already discussed in the summary sections of the previous chapters.

## 8.1 Summary, Discussion, Open Issues

The availability and accessibility of digitalized text documents is constantly growing. Automatic support in the analysis of these resources is mandatory if they are to be utilized. Although much research has been done in the last decades to process text computationally, still a major problem is to capture the semantic content of a text. Existing approaches differ with respect to (a) their goals (completely accessing the semantics vs. only the aspects that are relevant for a specific analysis task), (b) the employed method (linguistically vs. statistically based), and (c) the role of the human (fully automatic vs. human effort and computational processing clearly separated vs. a tight interaction between the human and the machine).

The work in this thesis is centered around a framework that could be classified as (a) approximating only the semantic aspects that are relevant for the task, (b) being (mostly) statistically based (while employing linguistic basic technology), and (c) advocating for a tight interaction between the human and the machine by means of visual analysis techniques. Working with such a framework raises several questions:

1. Are visual analysis techniques a suitable means to answer document analysis questions that require (at least partly) a semantic understanding?

2. Can higher-level text properties be approximated with statistical low-level features and should this approach be consequently followed in document analysis?

3. How can the above mentioned techniques, that are already popular in other domains, be applied to document analysis tasks? Or more precisely:

   (a) How can we find good approximations with low-level features for specific semantic aspects of a document?

   (b) Do we need novel visualization techniques for document analysis or can we use the proven ones of other domains?

4. How can we assess where visualization techniques are necessary and which ones are well suited?

In the following, the thesis at hand is reviewed with respect to which of these questions were answered, what remains an open issue and which additional questions were raised while working with the framework.

### 8.1.1 Visual analysis techniques as a means of semantic document analysis (Question 1)

Visual analysis techniques were successfully applied to several application examples in the thesis. This did not only show that it is possible to employ visual analysis techniques for document analysis tasks, but also illustrated some of the advantages that a combination of visual and automatic methods comes with.

In chapter 4, the Literature Fingerprinting technique was introduced that permits to visualize a document in detail. Fully automatic methods are often based on a single score per document only. With this compact, pixel-based representation, it becomes possible to analyze the development of the values across the text and thus, perceive trends and patterns in the data that otherwise would have been camouflaged. Once new hypothesis have been generated and verified, they can be fed back to improve the automatic analysis methods (see section 7.3.1, e.g., were the technique was used to evaluate and improve the quasi-semantic measure for sentiment analysis).

Chapter 5 dealt with readability analysis. In this case, transparency was one of the core requirements, because we did not only want to show the user if a certain passage in the text is difficult to read but also why this is the case. This meant that we needed (a) a measure that approximates the different aspects of readability separately and is understandable by the user and (b) a visualization that conveys this detailed view to the user. The case studies showed that this combination can be a powerful means in revising a document.

In chapter 7, visualization techniques were used to present the results of the automatic analysis process to the user. Instead of feeding the system with extensive knowledge of the world, the interpretation of the data is left to the user. This is a clear profit in this case, because what must be considered as *important* is highly dependent on the personal preferences and the specific context of the task - and thus would be very difficult to model explicitly. Furthermore both, chapters 6 and 7 combine visual and automatic methods in the evaluation process. The generated hypothesis are verified and then used to improve the automatic measures.

Does that mean that visual analytics should *always* be used in *every* document analysis process? The thesis at hand did not treat this question. However, it has to be pointed out that the concept of visual analytics in general does not assume that every analysis task needs this combination of visual and automatic method. Because human resources are expensive, they should only be utilized if the machine lacks necessary knowledge that would be difficult or impossible to model.

### 8.1.2 Approximation of higher-level text properties with statistical low-level features (Question 2)

The application examples in the thesis (especially chapters 5 and 7) as well as examples in related work (see section 3.1.2) showed that approximating a quasi-semantic property with low-level features is possible.

Furthermore, we could show the value of using semantically understandable features in the approximation. The resulting transparency of the measure is especially important when a close collaboration between the human and the machine is required. However, this poses additional requirements and challenges on the feature engineering process.

Regarding the second part of the question that asks whether such an approach should be consequently followed in document analysis processes, the following can be said: Approximating higher-level properties with a combination of low-level features has already proven powerful in other domains such as 3D Multimedia Retrieval. It permits to capture semantic aspects that would otherwise not be measurable. Besides, statistical approaches are usually more efficient than approaches that generate linguistic representations of the semantics. On the other hand, such an approach implies that only certain semantic aspects of a document are measured. There may be tasks that require a comprehensive representation of the inherent semantics. Currently, the linguistic branch of computational semantics and the more machine-learning based approaches of computer science are clearly separated. Investigating how and where those two approaches can profit from each other could be a valuable goal of future research.

### 8.1.3 Finding *good* approximations (Question 3a)

Chapter 5 exemplified a feature engineering process that takes into account that the resulting measure should be semantically understandable for a user. In chapter 7 existing measures for sentiment and opinion analysis were evaluated and improved. And finally, a novel method for the extraction of discriminating and overlap terms was introduced in chapter 6.

All three cases exemplify the process of finding good approximations for quasi-semantic properties. Both, readability analysis and opinion analysis could recourse on ground-truth data, something that generally eases the feature engineering process. To develop the term extraction method, no such ground-truth data was available. However, in this case detailed information about the properties of the extracted terms were given that could be used to evaluate and optimize the measure in the specific direction.

Quasi-semantic properties for which neither ground-truth data nor a clear specification is available pose additional challenges on the feature engineering process. This is for example the case if the quasi-semantic property depends on the personality or the previous knowledge of the user. The quasi-semantic property *non-obviousness* can be considered as an example in which the previous knowledge of the user plays a role. Depending on how much someone already knows, the presented content will be considered as non-obvious or obvious. In such cases, no universally agreed on ground-truth exists. Interaction between the user and the machine is therefore mandatory in the feature engineering process of such properties.

Future work should also investigate whether a generalization in the feature engineering process is possible. As was shown in section 2.2.4, quasi-semantic measures often build on other quasi-semantic measures. This property was exploited in chapter 7 where the attribute extraction method of chapter 6 was incorporated in the opinion measure. However, our experiments also showed that a measure that was developed for one domain cannot necessarily be employed in other domain (see section 7.5). This calls for a detailed analysis of how and where generalization would be possible.

### 8.1.4 Requirements on the employed visualization techniques (Question 3b)

Applying an automatic document analysis technique can result in a transformation of the semi-structured text into a structured dataset. In this case, the usage of standard visualization techniques, that were developed for structured data, may be possible. Examples in this thesis include the summary reports of section 7.4.1. Although specifically designed for the opinion analysis task, they could easily be used for other kind of data that is available in a tabular format. The technique is useful in all kinds of situations in which multiple continuous and normalized values for each item exist and several lines should be summarized and presented in a way that they can be compared to each other. Similarly, most of the examples of visual feature engineering employed standard visualization techniques.

On the other hand, certain properties of textual data can be identified that justify the need for special visualization techniques for document analysis. First, in some cases the document structure is important. Techniques such as Literature Fingerprinting (chapter 4) permits to follow the development of the feature values across the document and to recognize the position of outlier values within the document. Second, words are very expressive and thus convey information that numbers or a restricted amount of symbols never can. The popularity of word clouds reflects the need for techniques that account for this additional information. Especially challenging for the design of effective visualizations is the fact that words cannot be perceived preattentively. Besides, relationships between the different (term) sets may have to be displayed. This is for example the case in chapter 6. In the presented case study, we analyze the extracted discriminating and overlap terms of conference proceedings. For each document collection, a set of terms exists and additionally the relationships between two or multiple collections need to be illustrated. When only few collections were compared to each other, venn diagrams were used to display the extracted terms and the overlaps between the conferences. For larger data sets, an extended table representation was employed that intuitively presents the result of the automatic algorithm, but requires some effort in reading. In the future, more advanced visualizations in that area could be developed.

### 8.1.5 Assessing the need and value of visualization (Question 4)

When answering question 1, we already pointed out that not all document analysis tasks require the combination of visual and automatic methods. This raises the question when visualization techniques should be incorporated in the process. In general, it can be said that analysis processes profit from the use of visualization, whenever a task cannot be solved fully automatically and the knowledge and capabilities of the human are required.

Regarding the question what determines the value of visualization, two aspects can be identified: First, is the data displayed in a way that the necessary information can easily be perceived and that the display is not misleading? And second, does the usage of visualization techniques outperform the approaches that try to solve the task fully automatically? Both questions are not restricted to the domain of document analysis but must be asked in all domains in which visual analysis methods are applied. How visualization techniques can and should be evaluated is currently a hot topic in the visualization community. In the past, two main approaches were followed: User studies were conducted to verify or falsify the hypothesis that a certain visualization is not misleading and does effectively support the analysis process. Second, visualizations can be evaluated quantitatively if the desired properties of the visualization are computationally measurable. (See e.g. [143, 64] compared to [89].)

In conclusion, it can be said that there is a growing interest in computational support for document analysis. The thesis at hand could show that visual document analysis is a promising means of tackling the task. There are many interesting and practically relevant research questions that raise the expectation that a further development in the area will be seen in the upcoming years.

# A
# Appendix

**Contents**

## A.1 Quasi-semantic questions and properties in the example scenarios

In the following, details for the example scenarios of section 1.1 that are not discussed in section 2.2.1 are given.

### QSQs and QSPs for the example scenarios of companies:

Response management

*Quasi-semantic properties:* Quasi-semantic questions

1. *Topic / Concept:* Which topics are covered in the email?

*Related analysis questions:*

- How similar is the mail compared to what we previously answered? (searching for a good template)

QSQs and QSPs for the example scenarios of literature scientists:

Determining the age of the target audience of a book

*Quasi-semantic properties:* Quasi-semantic questions

1. *Violence level:* Does it contain much violence?
2. *Readability:* How easy is it to read?
3. *Induced emotions:* What emotions are aroused when reading it?
4. *Complexity:* How complex is the story?
5. *Age Suitability of Concepts:* Are the topics that are discussed suitable for the age of the target audience?

*Related analysis questions:*

- Prediction of the age (age interval) of the ideal target audience.

Authorship Attribution

*Quasi-semantic properties:* Quasi-semantic questions

1. *Writing style:* What are the characteristic and distinct elements in the writing style of the author?

*Related analysis questions:*

- How probable is it that a given piece of writing has been written by a specific person that we have samples with known authorship from?
- How probable is it that two given pieces of writing have been written by the same person?

## A.2 Commercial text analysis software

| **Company** | **URL** |
|---|---|
| ActivePoint | http://www.activepoint.com/ |
| AEROTEXT | http://www.lockheedmartin.com/products/AeroText/-index.html |
| Arrowsmith software | http://arrowsmith.psych.uic.edu/arrowsmith_uic/-index.html |
| Attensity | http://www.attensity.com/en/index.php |
| Basis Technology | http://www.basistech.com/ |
| Clarabridge | http://www.clarabridge.com/ |
| ClearForest | http://www.clearforest.com/ |
| Compare Suite | http://comparesuite.com/ |

*Continued on next page*

| **Company** | **URL** |
|---|---|
| Connexor Machinese | http://www.connexor.com/ |
| Copernic Summarizer | http://www.copernic.com/ |
| Crossminder | http://www.crossminder.com/ |
| Cypher | http://www.monrai.com/products/cypher |
| dtSearch | http://www.dtsearch.com/ |
| Eaagle text mining software | http://www.eaagle.com/ |
| Entrieva | http://www.lucidmedia.com/ |
| Expert System | http://www.expertsystem.net/ |
| File Search Assistant | http://www.aks-labs.com/products/-files_search_assistant.htm |
| IBM | http://www-01.ibm.com/software/data/infosphere/-warehouse/unstructured-data-analysis.html |
| Intellexer | http://www.intellexer.com/ |
| ISYS:desktop | http://www.isys-search.com/ |
| IxReveal | http://www.ixreveal.com/ |
| Leximancer | https://leximancer.thecustomerinsightportal.com/ |
| Lextek Onix Toolkit | http://www.lextek.com/ |
| Linguamatics | http://www.linguamatics.com/welcome/software/I2E.html |
| L&C | http://www.landcglobal.com/ |
| Megaputer Intelligence | http://www.megaputer.com/ |
| NewsFeed Researcher | http://newsfeedresearcher.com/ |
| Power Text Solutions | http://www.powertextsolutions.com/ |
| Readability Studio | http://www.oleandersolutions.com/readabilitystudio.html |
| Readware | http://www.readware.com |
| recommind MindServer | http://www.recommind.com/ |
| SAS Text Miner | http://www.sas.com/technologies/analytics/datamining/-textminer/ |
| SPSS PASW Text Analytics | http://www.spss.com/software/modeling/text-analytics/ |
| Fraunhofer FIT | http://www.fit.fraunhofer.de/projects/prozesse/-swapit_de.html |
| temis | www.temis.com |
| VantagePoint | http://www.thevantagepoint.com/ |
| VisualText | http://www.textanalysis.com/ |
| WordStat | http://www.provalisresearch.com/wordstat/Wordstat.html |
| Xanalys | http://www.xanalys.com/ |

## A.3 Complete list of text features

Complete list of text features that were taken into account in the readability analysis.

| **Text feature** | **Description** |
|---|---|
| Word | computes amount of words (punctuation marks excluded) |
| Word-Syllable | computes amount of syllables for each words (punctuation marks excluded) |
| Word-Monosyllable | marks and computes amount of monosyllable words (words which consist of one syllable only, punctuation marks excluded) |
| Word-Polysyllable | marks and computes amount of polysyllable words (words which consist of 3 or more syllables, punctuation marks excluded) |
| Word-Character | computes amount of characters for each words (punctuation marks excluded) |
| Sentence | computes amount of sentences |
| Word Length (Syllables per Token) | computes word length in syllables per tokens |
| Word Length (Characters per Token) | computes word length in characters per tokens |
| Word Length (Syllables per Word) | computes word length in syllables per words (punctuation marks excluded) |
| Word Length (Characters per Word) | computes word length in characters per words (punctuation marks excluded) |
| Phrase Length (Tokens per Phrase) | computes phrase length in tokens |
| Phrase Length (Phrases per Phrase) | computes phrase length in phrases |
| Clause Length (Tokens per Clause) | computes clause length in tokens |
| Clause Length (Phrases per Clause) | computes clause length in phrases |
| Clause Length (Clauses per Clause) | computes clause length in clauses |
| Sentence Length (Tokens per Sentence) | computes sentence length in tokens |
| Sentence Length (Token-Syllables per Sentence) | computes sentence length in syllables of tokens |
| Sentence Length (Token-Characters per Sentence) | computes sentence length in characters of tokens |
| Sentence Length (Words per Sentence) | computes sentence length in words (punctuation marks excluded) |
| Sentence Length (Word-Syllables per Sentence) | computes sentence length in syllables of words (punctuation marks excluded) |
| Sentence Length (Word-Characters per Sentence) | computes sentence length in characters of words (punctuation marks excluded) |
| Adverbial | marks and computes amount of place and time adverbials |
| Adverbial Class | determines adverbial class (place, time, none) |
| Place Adverbial | marks and computes amount of place adverbials |
| Time Adverbial | marks and computes amount of time adverbials |

*Continued on next page*

| Text feature | Description |
|---|---|
| Nominal Form Class | classifies tokens into one of following classes: 'nominalization', 'gerund', 'other', 'none' |
| Nominal Form | nominal forms (nominalizations, gerunds or other nouns) |
| Nominalization | nominalizations |
| Gerund | gerunds |
| Other Noun | other nouns |
| Passive | passive constructions |
| Pronoun | personal and impersonal pronouns |
| Personal Pronoun | personal pronouns |
| First Person Pronoun | first person pronouns |
| Second Person Pronoun | second person pronouns |
| Third Person Pronoun | third person pronouns |
| Impersonal Pronoun | impersonal pronouns |
| Demonstrative Pronoun | demonstrative pronouns |
| Indefinite Pronoun | indefinite pronouns |
| it Pronoun | it pronouns |
| Intensifier | intensifiers (emphasizer, amplifier or downtoner) |
| Amplifier | amplifiers |
| Downtoner | downtoners |
| Emphasizer | emphasizers |
| Verb Class | classifies tokens into one of following classes: 'public', 'private', 'suasive', 'seem_appear', 'none' |
| Verb Class | words which are contained in one of a verb class (public, private, suasive, seem/appear) |
| Public Verb | public verbs |
| Private Verb | private verbs |
| Suasive Verb | suasive verbs |
| Seem/Appear Verb | seem/appear verbs |
| Part-of-Speech Category Class | classifies tokens into one of the following classes: 'noun', 'pronoun', 'verb', 'adjective', 'adverb', 'other' |
| Noun | marks and computes amount of nouns (gerunds included, requires full parsing) |
| Noun (Gerunds excluded) | marks and computes amount of nouns (gerunds excluded) |
| Proper Noun | marks and computes amount of proper nouns |
| Noun + Proper Noun | marks and computes amount of nouns (gerunds included, requires full parsing) and proper nouns |
| Noun (Gerund excluded) + Proper Noun | marks and computes amount of nouns (gerunds excluded) and proper nouns |
| Verb | marks and computes amount of full verbs (gerunds excluded, requires full parsing) |
| Verb (possible gerund included) | marks and computes amount of verbs (possible gerunds included) |
| Verb + Modal | marks and computes amount of verbs (gerunds excluded, requires full parsing) and modals |
| Verb (possible gerund included) + Modal | marks and computes amount of verbs (possible gerunds included) and modals |

*Continued on next page*

| Text feature | Description |
|---|---|
| Adjective | marks and computes amount of adjectives |
| Adverb | marks and computes amount of adverbs |
| Adjective + Adverb | marks and computes amount of adjectives and adverbs |
| Pronoun | marks and computes amount of pronouns (personal, possessive or (possessive) wh-pronoun) |
| Wh-Word | marks and computes amount of wh-words (wh-determiner, (possessive) wh-pronoun or wh-adverb) |
| Modal | marks and computes amount of modals |
| Interjection | marks and computes amount of interjections |
| Conjunction | marks and computes amount of conjunctions (connective of words, phrases or clauses) |
| Particle | marks computes amount of particles (token that are not assignable to any of the other pos categories) |
| Determiner | marks computes amount of determiners/articles |
| Cardinal Number | marks and computes amount of cardinal numbers |
| Pre/Postposition | marks and computes amount of prepostions and postpositions |
| Foreign-Language Material | marks and computes amount of foreign-language material |
| Non-Word | marks and computes amount of non-words |
| Composition | marks and computes amount of compositions (e.g. An- [und Abreise]) |
| Noun Phrase | marks and computes amount of noun phrases |
| Verb Phrase | marks and computes amount of verb phrases |
| Adjective Phrase | marks and computes amount of adjective phrases |
| Adverb Phrase | marks and computes amount of adverb phrases |
| Adjective/Adverb Phrase | marks and computes amount of adjective/adverb phrases |
| Pre/Adpositional Phrase | marks and computes amount of pre- (for English) and ad- (for German) positional phrases |
| Conjunction Phrase | marks and computes amount of conjunction phrases |
| Quantifier Phrases | marks and computes amount of quantifier phrases |
| Unlike Coordinated Phrases | marks and computes amount of unlike coordinated phrases |
| Wh-Phrases | marks and computes amount of wh-phrases |
| Fragments | marks and computes amount of fragments |
| Parentheticals | marks and computes amount of parentheticals |
| Sentence/Simple Declarative Clause | marks and computes amount of simple declarative clauses (for PTTS) or sentences (for STTS), i.e. one that is not introduced by a (possible empty) subordinating conjunction or a wh-word and that does not exhibit subject-verb inversion |
| Inverted Declarative Sentences | inverted declarative sentence (determined by its clausal category), i.e. one in which the subject follows the tensed verb or modal. |
| Subordinate Clauses | clause introduced by a (possibly empty) subordinating conjunction (determined by its clausal category) |
| Direct Questions | direct question (determined by its clausal category) introduced by a wh-word or a wh-phrase |
| Inverted Yes/No Questions / Main Clauses | inverted yes/no question, or main clause of a wh-question (determined by its clausal category) |

*Continued on next page*

| Text feature | Description |
|---|---|
| Easy/Hard/Complex Word Class | classifies tokens into one of following classes: ’easy’, ’hard’, ’complex’ |
| Easy Word | words with less than 3 syllables |
| Hard Word | words with more than 2 syllables |
| Complex Word | words with more than 2 syllables, no proper nouns, no compound words |
| Basic Vocabulary Word | marks and computes amount of basic vocabulary words |
| Function Word | marks and computes amount of function words |
| Academic Word | marks and computes amount of academic words |
| General Service Word | marks and computes amount of general service words |
| Connective | marks and computes amount of connectives |
| Conjunct | marks and computes amount of conjuncts |
| Most 50 Frequent Word (default) | words which are contained in the default most 50 frequent word list |
| Most 50 Frequent Word (Project Gutenberg, English) | words which are contained in the most 50 frequent ’Project Gutenberg’ word list |
| Most 50 Frequent Word (Wikipedia, German) | words which are contained in the most 50 frequent ’Wikipedia’ word list |
| Most 50 Frequent Word (Uni Leipzig Lexicon, German) | words which are contained in the most 50 frequent ’Uni Leipzig Lexicon’ word list |
| Most 100 Frequent Word (default) | words which are contained in the default most 100 frequent word list |
| Most 100 Frequent Word (Project Gutenberg, English) | words which are contained in the most 100 frequent ’Project Gutenberg’ word list |
| Most 100 Frequent Word (Wikipedia, German) | words which are contained in the most 100 frequent ’Wikipedia’ word list |
| Most 100 Frequent Word (Uni Leipzig Lexicon, German) | words which are contained in the most 100 frequent ’Uni Leipzig Lexicon’ word list |
| Most 500 Frequent Word (default) | words which are contained in the default most 500 frequent word list |
| Most 500 Frequent Word (Project Gutenberg, English) | words which are contained in the most 500 frequent ’Project Gutenberg’ word list |
| Most 500 Frequent Word (Wikipedia, German) | words which are contained in the most 500 frequent ’Wikipedia’ word list |
| Most 500 Frequent Word (Uni Leipzig Lexicon, German) | words which are contained in the most 500 frequent ’Uni Leipzig Lexicon’ word list |
| Most 1000 Frequent Word (default) | words which are contained in the default most 1000 frequent word list |
| Most 1000 Frequent Word (Project Gutenberg, English) | words which are contained in the most 1000 frequent ’Project Gutenberg’ word list |

*Continued on next page*

| **Text feature** | **Description** |
|---|---|
| Most 1000 Frequent Word (Wikipedia, German) | words which are contained in the most 1000 frequent 'Wikipedia' word list |
| Most 1000 Frequent Word (Uni Leipzig Lexicon, German) | words which are contained in the most 1000 frequent 'Uni Leipzig Lexicon' word list |
| Most 2000 Frequent Word (default) | words which are contained in the default most 2000 frequent word list |
| Most 2000 Frequent Word (Project Gutenberg, English) | words which are contained in the most 2000 frequent 'Project Gutenberg' word list |
| Most 2000 Frequent Word (Wikipedia, German) | words which are contained in the most 2000 frequent 'Wikipedia' word list |
| Most 2000 Frequent Word (Uni Leipzig Lexicon, German) | words which are contained in the most 2000 frequent 'Uni Leipzig Lexicon' word list |
| No Most Frequent Word (default) | words which are not contained in the default most 2000 frequent word list |
| No Most Frequent Word (Project Gutenberg, English) | words which are not contained in the most 2000 frequent 'Project Gutenberg' word list |
| No Most Frequent Word (Wikipedia, German) | words which are not contained in the most 2000 frequent 'Wikipedia' word list |
| No Most Frequent Word (Uni Leipzig Lexicon, German) | words which are not contained in the most 2000 frequent 'Uni Leipzig Lexicon' word list |
| Branching Factor | computes branching factor of phrase structure tree |
| Total Phrase Structure Tree Depth | computes maximum depth of phrase structure tree |
| First Verb Occurrence (Token) | computes first occurrence of a verb in phrase structure tree measured in number of tokens |
| First Verb Occurrence (Word) | computes first occurrence of a verb in phrase structure tree measured in number of words (punctuation marks excluded) |
| First Verb Occurrence (Depth) | computes first occurrence of a verb in phrase structure tree measured in tree depth |
| First Verb Phrase Occurrence (Depth) | computes first occurrence of a verb phrase in phrase structure tree measured in tree depth |
| Yngve Complexity | computes Yngve complexity of tokens and sentences |
| Noun Phrase Complexity | computes complexity of noun phrases by checking number of adjectives/adverbs |
| Sentence Nesting | computes degree of sentence nesting |
| Automated Readability Index | computes Automated Readability Index |
| Flesch-Kincaid Readability Measure | computes Flesch-Kincaid Readability Measure |
| Flesch Readability Measure | computes Flesch Readability Measure |
| Simple Measure of Gobbledygook | computes Simple Measure of Gobbledygook |
| Coleman-Liau Index | computes Coleman-Liau Index |

*Continued on next page*

| Text feature | Description |
|---|---|
| FORCAST formula | computes FORCAST formula |
| Gunning Fog Index | computes Gunning Fog Index |
| Linsear Write Readability Metric | computes Linsear Write Readability Metric |
| Simpson's Index (Word) | measures the chance that two words arbitrarily chosen from a text sample will be the same |
| Simpson's Index (Stem) | measures the chance that two stems arbitrarily chosen from a text sample will be the same |
| Honore's R (Word) | Honore's R is supposed to "test the propensity of an author to choose between the alternatives of employing a word used previously or employing a new word" (Holmes, 1994). R = ( 100 * log(number of words) )/( 1 - words that occurs exactly once / number of words ) |
| Honore's R (Stem) | Honore's R is supposed to "test the propensity of an author to choose between the alternatives of employing a word used previously or employing a new word" (Holmes, 1994). R = ( 100 * log(number of stems) )/( 1 - stems that occurs exactly once / number of stems ) |
| Hapax Dis Legomena (Word) | word that occurs exactly twice / number of words |
| Hapax Dis Legomena (Stem) | stem that occurs exactly twice / number of stems |
| Hapax N Legomena (Word) | word that occurs exactly n times / number of words |
| Hapax N Legomena (Stem) | stem that occurs exactly n times / number of stems |
| Yule's K-Characteristic (Word) | measures the likelihood of two words, chosen at random from the text, being the same. |
| Yule's K-Characteristic (Stem) | measures the likelihood of two stems, chosen at random from the text, being the same. |
| Default Type-Token-Ratio (Word) | default type-token-ratio : #unique words / #tokens |
| Default Type-Token-Ratio (Stem) | default type-token-ratio : #unique stems / #tokens |
| Corrected Type-Token-Ratio (Word) | corrected type-token-ratio : #unique words / (2 * sqrt(#tokens)) |
| Corrected Type-Token-Ratio (Stem) | corrected type-token-ratio : #unique stems / (2 * sqrt(#tokens)) |
| Index of Guiraud (Word) | Index of Guiraud : #unique words / sqrt(#tokens) |
| Index of Guiraud (Stem) | Index of Guiraud : #unique stems / sqrt(#tokens) |
| Index of Herdan (Word) | Index of Herdan : log(#unique words) / log(#tokens) |
| Index of Herdan (Stem) | Index of Herdan : log(#unique stems) / log(#tokens) |
| Uber's Index (Word) | Uber's Index : pow(log(#tokens),2) / (log(#tokens) - log(#unique words)) |
| Uber's Index (Stem) | Uber's Index : pow(log(#tokens),2) / (log(#tokens) - log(#unique stems)) |
| Stop Words | stop words |
| No Stop Words | DS |
| Words with n or more Characters | words with n or more characters |

*Continued on next page*

| **Text feature** | **Description** |
|---|---|
| Words with less than n Characters | words with less than n characters |
| Token Node Filter (default) | classifies tokens into “enabled” or “disabled” depending on particular condition |
| Token Node Filter (Enabler) | classifies tokens into “enabled” or “disabled” depending on particular condition |
| Token Node Filter (Disabler) | classifies tokens into “enabled” or “disabled” depending on particular condition |
| Term Frequency (absolute) | absolute term frequency |
| Term Frequency (absolute, log weighted) | absolute log weighted term frequency |
| Term Frequency (relative) | relative term frequency |
| Term Frequency (relative, log weighted) | relative log weighted term frequency |
| Inverse Document Frequency | Computes inverse document frequency. This is a corpus feature transformer which means the transformation process could be very time- and resource-intensive depending on size of corpus |
| Term Frequency - Inverse Document Frequency (default) | default tf-idf (requires a corpus feature transformer) |
| Term Frequency (absolute) - Inverse Document Frequency | tf-idf using absolute term frequency (requires a corpus feature transformer) |
| Term Frequency (absolute, log weighted) - Inverse Document Frequency | tf-idf using absolute log weighted term frequency (requires a corpus feature transformer) |
| Term Frequency (relative) - Inverse Document Frequency | tf-idf using relative term frequency (requires a corpus feature transformer) |
| Term Frequency (relative, log weighted) - Inverse Document Frequency | tf-idf using relative log weighted term frequency (requires a corpus feature transformer) |
| Unique, Numerical LEVEL Identifier | Unique, numerical identifier for LEVEL tags |
| Quotation | classifies tokens into one of following classes: ’unquoted’ (not quoted words), ’quoted’ (words within quotes), ’quote’ (the quotes self) |
| Noun (+ Proper Nouns) / Verb (+ Auxiliaries) Ratio | computes noun/verb ration (noun: nouns, proper nouns and gerunds; verb: verbs and auxiliaries) |
| Noun / Verb (+ Auxiliaries) Ratio | computes noun/verb ration (noun: nouns and gerunds (proper nouns excluded); verb: verbs and auxiliaries) |
| Noun (+ Proper Nouns) / Verb (no Auxiliaries) Ratio | computes noun/verb ration (noun: nouns, proper nouns and gerunds; verb: verbs (auxiliaries excluded)) |
| Noun / Verb (no Auxiliaries) Ratio | computes noun/verb ration (noun: nouns and gerunds (proper nouns excluded); verb: verbs (auxiliaries excluded)) |
| Noun (+ Proper Nouns, no Premodifiers) / Verb (no Auxiliaries) Ratio | computes noun/verb ration (noun: nouns, proper nouns and gerunds (premodifiers of other nouns excluded); verb: verbs (auxiliaries excluded)) |

*Continued on next page*

| Text feature | Description |
|---|---|
| Noun (no Premodifiers) / Verb (no Auxiliaries) Ratio | computes noun/verb ration (noun: nouns and gerunds (proper nouns and premodifiers of other nouns excluded); verb: verbs (auxiliaries excluded)) |

## A.4 Benchmark dataset for readability analysis

| Title | Author(s) | Number of tokens | Flesch Score [39] |
|---|---|---|---|
| FP7 (B1) Cooperation (T1) Health - Work Programme 2009 | European Commission | 13201 | 10.79 |
| FP7 (B1) Cooperation (T2) Food, Agriculture and Fisheries, and Biotechnology - Work Programme 2009 | European Commission | 21748 | 9.04 |
| FP7 (B1) Cooperation (T3) Information and Communication Technologies - Work Programme 2007-2008 | European Commission | 13219 | 7.23 |
| FP7 (B1) Cooperation (T3) Information and Communication Technologies - Work Programme 2009-2010 | European Commission | 20472 | 11.60 |
| FP7 (B1) Cooperation (T4) Nanosciences, Nanotechnologies, Materials and New Production Technologies - Work Programme 2009 | European Commission | 13399 | 3.47 |
| FP7 (B1) Cooperation (T5) Energy - Work Programme 2009 | European Commission | 9115 | 14.38 |
| FP7 (B1) Cooperation (T6) Environment - Work Programme 2009 | European Commission | 11709 | 7.82 |
| FP7 (B1) Cooperation (T7) Transport - Work Programme 2008 | European Commission | 21943 | 8.39 |
| FP7 (B1) Cooperation (T8) Socio-Economic Sciences and Humanities - Work Programme 2009 | European Commission | 8410 | 9.84 |
| Campfire Girls in the Allegheny Mountains | Stella M, Francis | 38840 | 67.32 |
| Mrs. Budlong's Christmas Presents | Rupert Hughes | 15317 | 73.08 |
| A Kidnapped Santa Claus | L. Frank Baum | 4327 | 68.46 |
| Little Stories for Little Children | Anonymous | 1458 | 91.18 |
| Astronomy for kids | kidsastronomy.com | 15409 | 70.31 |
| Geology for kids | kidsgeo.com | 10398 | 62.35 |
| Geography for kids | kidsgeo.com | 30616 | 63.40 |
| Biology for Kids | kids-biology.com | 20125 | 72.42 |

*Continued on next page*

**Table A.3 – continued from previous page**

| **Title** | **Author(s)** | **Number of tokens** | **Flesch Score [39]** |
|---|---|---|---|
| Children's Day | null | 1447 | 90.98 |
| Little Polar Bear | null | 985 | 85.05 |
| Mick and Mo | null | 1011 | 83.74 |
| Pirate Girl | null | 1069 | 80.48 |
| Short Football | null | 1409 | 88.36 |
| Super Easy Reading | Ron Chang Lee (Rong-Chang Li) | 12645 | 94.33 |
| The Summer Holidays: A Story for Children | Amerel | 10765 | 81.48 |
| Grimms' Fairy Tales | The Brothers Grimm | 121017 | 80.03 |
| The Adventures of Peter Pan | James M. Barrie | 58605 | 79.91 |
| The Adventures of Pinocchio | C. Collodi - Pseudonym of Carlo Lorenzini | 49805 | 84.30 |
| Cinderella | Anonymous | 1126 | 68.04 |
| Cinderella; or, The Little Glass Slipper and Other Stories | Anonymous | 22542 | 82.25 |

## A.5 Evaluation of the attribute extraction (additional material)

Additional material for section 6.3.3, showing the extracted terms for different sizes of the document collection.

| **Number of documents** | **Attributes** |
|---|---|
| ca. 1000 | olympus, panasonic, button, battery life, dslr, aa battery, video, megapixel, sony, photographer, pic, photography, battery, user, picture quality, hd video, slr, zs3, memory card, auto mode, beach, zoom, flash, fuji, shutter speed, lcd, viewfinder, photo, color, manual, shoot camera, mp, research, lcd screen, iso, amazon verified purchase, image quality, image stabilization, nikon, menu |
| ca. 750 | olympus, panasonic, button, battery life, dslr, aa battery, video, megapixel, sony, photographer, pic, photography, battery, size, user, picture quality, slr, memory card, auto mode, beach, zoom, flash, fuji, shutter speed, lcd, viewfinder, photo, color, manual, shoot camera, mp, lcd screen, research, iso, model, amazon verified purchase, image quality, image stabilization, nikon |

*Continued on next page*

| Number of documents | Attributes |
|---|---|
| ca. 600 | olympus, button, battery life, dslr, aa battery, video, megapixel, sony, photographer, pic, photography, battery, size, user, picture quality, slr, feature, memory card, auto mode, zoom, flash, fuji, shutter speed, lcd, viewfinder, photo, kodak, camera, color, manual, shoot camera, mp, research, lcd screen, water, iso, model, amazon verified purchase, image quality, image stabilization, nikon, charger |
| ca. 500 | olympus, panasonic, button, battery life, video, focus, megapixel, sony, photographer, pic, photography, range, battery, size, user, picture quality, hd video, slr, memory card, auto mode, beach, software, zoom, flash, fuji, shutter speed, viewfinder, photo, auto, lens, color, manual, mp, research, lcd screen, iso, pros, model, amazon verified purchase, image quality, image stabilization, nikon |
| ca. 350 | olympus, panasonic, button, battery life, video, focus, megapixel, sony, photographer, pic, battery, size, user, picture quality, slr, memory card, auto mode, video quality, beach, zoom, flash, fuji, shutter speed, viewfinder, mode, photo, auto, lens, box, kodak, color, manual, shoot camera, lcd screen, iso, pros, model, amazon verified purchase, image quality, image stabilization, nikon, charger |
| ca. 250 | olympus, panasonic, button, pocket, battery life, video, focus, megapixel, sony, photographer, pic, battery, size, picture quality, performance, slr, option, brand, memory card, auto mode, video quality, beach, finger, zoom, ton, flash, fuji, shutter speed, viewfinder, mode, photo, auto, lens, box, color, manual, lcd screen, iso, model, amazon verified purchase, image quality, nikon |
| ca. 150 | image, panasonic, button, video, focus, purchase, sony, photographer, pic, battery, size, picture quality, slr, option, feature, video quality, beach, finger, zoom, flash, fuji, viewfinder, mode, photo, auto, lens, box, canon, color, manual, iso, model, amazon verified purchase, nikon |
| ca. 100 | olympus, image, issue, panasonic, button, video, purchase, sony, type, photographer, pic, tripod, battery, trip, size, picture quality, picutre, feature, video quality, finger, zoom, flash, fuji, mode, crystal, photo, auto, lens, box, canon, camera, color, pros, model, amazon verified purchase, image stabilization, nikon |
| ca. 50 | panasonic, button, plastic, canon camera, video, focus, type, battery, trip, size, picture quality, feature, quality, couple, video quality, beach, finger, zoom, flash, fuji, picture, mode, crystal, photo, auto, lens, box, canon, use, shutter, color, camera, manual, sound, pros, model, amazon verified purchase, nikon |

*Continued on next page*

| **Number of documents** | **Attributes** |
|---|---|
| ca. 25 | zoom brown, program set, picture, link, zoom, use, powershot sd1200is, grainy photograph, lens, optical image, photograph, color, amazon verified purchase, model, box, flash, canon powershot sd1100is 8mp digital camera, size, video, zoom brown link, plastic, powershot, auto, canon camera, video quality, feature, mode, toy, photo, reviewer, finger, nephew, course, camera, canon, quality, opinion, picture quality, color accent tool, focus, mp count, week |

# List of Figures

# List of Tables

# Bibliography

[1] A. Abbasi and H. Chen. Categorization and Analysis of Text in Computer Mediated Communication Archives using Visualization. In *Proceedings of the 2007 Conference on Digital Libraries (JCDL '07)*, pages 11–18. ACM, 2007.

[2] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the Influential Bloggers in a Community. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM '08)*, pages 207–218, 2008.

[3] D. Allen, 2009. Emails reach 210 billion per day, `http://www.techwatch.co.uk/2009/01/26/emails-reach-210-billion-per-day/`, accessed on 7/24/09.

[4] A. Andreevskaia and S. Bergler. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics: Human Language Technologies (ACL '08)*, pages 290–298, 2008.

[5] G. Antoniou and F. van Harmelen. *A Semantic Web Primer*. The MIT Press, 2008.

[6] ASV Toolbox (with Terminology Extraction Tool) of the University of Leipzig: `http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/index.htm`.

[7] M. Atkinson and E. Van der Goot. Near Real Time Information Mining in Mulitlingual News. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, pages 1153–1154. ACM, 2009.

[8] A. Aue and M. Gamon. Customizing Sentiment Classifiers to New Domains: A Case Study. unpublished, 2005. `http://research.microsoft.com/apps/pubs/default.aspx?id=65430`.

[9] A. P. Azcarraga, T. N. Yap, J. Tan, and T. S. Chua. Evaluating Keyword Selection Methods for WEBSOM Text Archives. *IEEE Transactions on Knowledge and Data Engineering*, 16(3):380–383, 2004.

[10] P. Bak, F. Mansmann, H. Janetzko, and D. A. Keim. Spatiotemporal Analysis of Sensor Logs using Growth Ring Maps. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[11] T. Ball and S. G. Eick. Software Visualization in the Large. *IEEE Computer*, 29(4):33–43, 1996.

[12] R. Barzilay and M. Lapata. Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*, pages 141–148. ACL, 2005.

[13] How much information? 2003, `http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/`, last accessed on 7/24/2009.

[14] M. Billig. The language of critical discourse analysis: the case of nominalization. *Discourse & Society*, 19(6):783–800, 2008.

[15] J. N. G. Binongo. Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution. *Chance*, 16(2):9–17, 2003.

[16] D. Bourigault, I. Gonzalez-Mullier, and C. Gros. LEXTER, a Natural Language Processing Tool for Terminology Extraction. In *Proceedings of the Seventh EURALEX International Congress on Lexicography (EURALEX '96)*, pages 771–779, 1996.

[17] M. Brunzel and M. Spiliopoulou. Domain Relevance on Term Weighting. In *Proceedings of the 12th International Conference on Applications of Natural Language to Information Systems (NLDB '07)*, pages 427–432, 2007.

[18] C. S. Butler. *Computers and Written Texts.* Basil Blackwell, 1992.

[19] G. Carenini and L. Rizoli. A Multimedia Interface for Facilitating Comparisons of Opinions. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '09)*, pages 325–334. ACM, 2009.

[20] P. Carvalho, L. Sarmento, M. J. Silva, and E. de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion (TSA '09)*, pages 53–56. ACM, 2009.

[21] J. Chae and A. Nenkova. Predicting the fluency of text with shallow structural features: case studies of machine translation and human-written text. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*, pages 139–147. ACL, 2009.

[22] C. Chen. Searching for intellectual turning points: Progressive Knowledge Domain Visualization. In *Proceedings of the National Academy of Sciences of the United States of America (PNAS '04)*, 2004.

[23] M. Coleman and T. Liau. A computer readabilty formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284.

[24] C. Collins, F. B. Viégas, and M. Wattenberg. Parallel Tag Clouds to Explore and Analyze Faceted Text Corpora. In *IEEE Symposium on Visual Analytics and Technology (VAST '09)*, pages 91–98, 2009.

[25] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC '99)*, 1999.

[26] K. Collins-Thompson and J. Callan. A Language Modeling Approach to Predicting Reading Difficulty. In *Proceedings of the Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting 2004*, 2004.

[27] K. Dave, S. Lawrence, and D. M. Pennock. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pages 519–528. ACM, 2003.

[28] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM '08)*, pages 231–240. ACM, 2008.

[29] X. Ding, B. Liu, and L. Zhang. Entity discovery and assignment for opinion mining applications. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pages 1125–1134, 2009.

[30] A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement, B. Shneiderman, and C. Plaisant. Discovering interesting usage patterns in text collections: Integrating text mining with visualization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM '07)*, pages 213–222. ACM, 2007.

[31] P. Drouin. Detection of Domain Specific Terminology Using Corpora Comparison. In *Proceedings of the International Language Resources Conference*, pages 79–82, 2004.

[32] Europe Media Monitor, `http://emm.newsbrief.eu/overview.html`.

[33] Encyclopedia Britannica, `http://www.britannica.com/EBchecked/topic/482436/public-opinion`.

[34] A. Farghaly, editor. *Handbook for Language Engineers.* CSLI Publications, 2003.

[35] J.-D. Fekete and N. Dufournaud. Compus: Visualization and Analysis of Structured Documents for Understanding Social Life in the 16th Century. In *Proceedings of the fifth ACM Conference on Digital Libraries (DL '00)*, pages 47–55. ACM, 2000.

[36] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir. Text mining at the term level. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD '98)*, pages 65–73. Springer-Verlag, 1998.

[37] R. Feldman and J. Sanger. *The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007.

[38] D. Fisher, A. Hoff, G. Robertson, and M. Hurst. Narratives: A Visualization to Track Narrative Events as they Develop. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST '08)*, pages 115–122, 2008.

[39] R. F. Flesch. A New Readability Yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

[40] B. Fortuna, D. Mladenic, and M. Grobelnik. Visualization of Text Document Corpus. *Informatica Journal*, 29(4):497–502, 2005.

[41] FrameNet, `http://framenet.icsi.berkeley.edu/`.

[42] M. Friendly. Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association*, 89:190–200, 1994.

[43] M. Gamon and A. Aue. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing (FeatureEng '05)*, pages 57–64, 2005.

[44] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining Customer Opinions from Free Text. *Advances in Intelligent Data Analysis VI*, pages 121–132, 2005.

[45] M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst, and A. C. König. BLEWS: Using Blogs to Provide Context for News Articles. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM '08)*, 2008.

[46] General inquirer project: `http://www.webuse.umd.edu:9090/`.

[47] Mark Greenwood: Noun Phrase Chunker Version 1.1, `http://www.dcs.shef.ac.uk/~mark/phd/software/chunker.html`.

[48] M. L. Gregory, N. Chinchor, P. Whitney, R. Carter, E. Hetzler, and A. Turner. User-directed Sentiment Analysis: Visualizing the Affective Content of Documents. In *Workshop on Sentiment and Subjectivity in Text*, pages 23–30, 2006.

[49] C. Grün, J. Gerken, H.-C. Jetter, W. König, and H. Reiterer. MedioVis - a User-Centred Library Metadata Browser. In *Research and Advanced Technology for Digital Libraries, Proceedings of the 9th European Conference on Digital Libraries (ECDL '05)*. Springer Verlag, 2005.

[50] R. Gunning. *The technique of clear writing*. McGraw-Hill, forth printing edition, 1952.

[51] Project Gutenberg, `http://www.gutenberg.org`.

[52] Dictionary of the most frequent words in the Project Gutenberg, `http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/PG/2006/04/1-10000`, last accessed on 01/10/2010.

[53] Dictionary of the most frequent words in the Project Wortschatz Universität Leipzig, `http://wortschatz.uni-leipzig.de/html/wliste.html`, last accessed on 01/10/2010.

[54] J. Hartigan and B. Kleiner. A mosaic of television ratings. *The American Statistician*, 38:32–35, 1984.

[55] V. Hatzivassiloglou and J. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 299–305. ACL, 2000.

[56] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[57] M. A. Hearst. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'95)*, 1995.

[58] M. Heilman, K. Collins-Thompson, J. Callan, and M. Eskenazi. Combining Lexical and Grammatical Features to Improve Readability Measures for First and Second Language Texts. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT '07)*, pages 460–467, 2007.

[59] G. Heyer, F. Holz, and S. Teresniak. Change of Topics over Time and Tracking Topics by Their Change of Meaning. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR '09)*, 2009.

[60] G. Heyer, U. Quasthoff, and T. Wittig. *Text Mining: Wissensrohstoff Text – Konzepte, Algorithmen, Ergebnisse.* W3L-Verlag, 2003.

[61] D. Hilbert. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen*, 38:459–460, 1891.

[62] S. Hoenisch, Identifying and Resolving Ambiguity, 2004, `http://www.criticism.com/linguistics/types-of-ambiguity.php`, accessed on September 8th, 2009.

[63] D. I. Holmes. Authorship Attribution. *Computers and the Humanities*, 28:87–106, 1994.

[64] D. Holten and J. J. van Wijk. A User Study on Visualizing Directed Edges in Graphs. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI '09)*, pages 2299–2308. ACM, 2009.

[65] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. WEBSOM - Selforganizing maps of document collections. *Neurocomputing*, 21:101–117, 1998.

[66] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 168–177. ACM, 2004.

[67] M. Hu and B. Liu. Mining Opinion Features in Customer Reviews. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI '04)*, pages 755–760, 2004.

[68] D. Jurafsky and J. H. Martin. *Speech and Language Processing.* Pearson Higher Education, 2000.

[69] K. Kageura and B. Umino. Methods of Automatic Term Recognition: A Review. *Terminology*, 3(2):259ff, 1996.

[70] D. Keim, H. Ming, U. Dayal, H. Jantzko, and P. Bak. Generalized Scatter Plots. *Information Visualization Journal*, 2010.

[71] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. *Visual Analytics: Scope and Challenges.* Lecture Notes in Computer Science (LNCS). Springer, 2008.

[72] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in Visual Data Analysis. In *Proceedings of the Conference on Information Visualization (IV '06)*, pages 9–16, 2006.

[73] D. A. Keim and D. Oelke. Literature Fingerprinting: A New Method for Visual Literary Analysis. In *IEEE Symposium on Visual Analytics and Technology (VAST '07)*, pages 115–122, 2007.

[74] S.-M. Kim and E. Hovy. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*, pages 1–8. ACL, 2006.

[75] J. P. Kincaid, R. P. Fishburn, R. L. Rogers, and B. S. Chissom. Derivation of New Readability Formulas for Navy Enlisted Personnel. Research branch report 8-75, Naval Air Station Memphis, Millington, Tennessee, February 1975.

[76] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL '03)*, pages 423–430, 2003.

[77] D. Klein and C. D. Manning. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 3–10. MIT Press, 2003.

[78] N. Kobayashi, K. Inui, Y. Matsumoto, K. Tateishi, and T. Fukushima. Collecting Evaluative Expressions for Opinion Extraction. In *Natural Language Processing IJCNLP 2004*, pages 596–605, 2004.

[79] S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative Integration of Visual Insights during Patent Search and Analysis. In *Proceedings of the IEEE Symposium on Visual Analytics and Technology (VAST '09)*, pages 203–210, 2009.

[80] M. Krstajić, F. Mansmann, A. Stoffel, M. Atkinson, and D. A. Keim. Processing Online News Streams for Large-Scale Semantic Analysis. In *1st International Workshop on Data Engineering meets the Semantic Web*, 2010.

[81] R. Kuhlen. *Experimentelle Morphologie in der Informationswissenschaft.* Verlag Dokumentation, 1977.

[82] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen. Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration. In E. Simoudis, J. Han, and U. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pages 238–243. AAAI Press, 1996.

[83] K. Lagus and S. Kaski. Keyword selection method for characterizing text document maps. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN '99)*, pages 371–376, 1999.

[84] A. Lauriston. Automatic recognition of complex terms: problems and the TERMINO solution. *Terminology*, 1(1):147–170, 1994.

[85] K. Lerman and R. McDonald. Contrastive Summarization: An Experiment with Consumer Reviews. In *Proceedings of the North American Association for Computational Linguistics (NAACL '09)*, 2009.

[86] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the Dynamics of the News Cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pages 497–506. ACM, 2009.

[87] B. Liu, M. Hu, and J. Cheng. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*, pages 342–351. ACM, 2005.

[88] M. Mandic and A. Kerne. Using intimacy, chronology and zooming to visualize rhythms in email experience. In *Conference on Human factors in Computing Systems, Poster paper (CHI '05)*, pages 1617–1620. ACM, 2005.

[89] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda. Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats. *Proceedings of the IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1105–1112, 2007.

[90] Y. Matsuo and M. Ishizuka. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. In *Proceedings of the 16th Florida AI Research Society (FLAIRS '03)*, pages 392–396, 2003.

[91] H. G. McLaughlin. SMOG grading - a new readability formula. *Journal of Reading*, pages 639–646, May 1969.

[92] A. Mehler and R. Köhler, editors. *Aspects of Automatic Text Analysis*. Studies in Fuzziness and Soft Computing. Springer, 2007.

[93] Q. Mei and C. Zhai. A mixture model for contextual text mining. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pages 649–655, 2006.

[94] R. Mitkov, editor. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.

[95] F. Moretti. *Graphs, maps, trees - abstract models for a literary history*. Verso, 2005.

[96] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining Product Reputations on the Web. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pages 341–349. ACM, 2002.

[97] G. Morton. A Computer Oriented Geodetic Data Base; and a New Technique in File Sequencing. In *Technical Report, Ottawa, Canada: IBM Ltd.*, 1966.

[98] S. T. Moturu, J. Yang, and H. Liu. Quantifying Utility and Trustworthiness for Advice Shared on Online Social Media. In *Proceedings of the 2009 International Conference on Computational Science and Engineering (CSE 09)*, pages 489–494, 2009.

[99] V. Ng, S. Dasgupta, and S. M. N. Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618, 2006.

[100] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim. Visual Readability Analysis: How to make your writings easier to read. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST '10)*, 2010. (to appear).

[101] W. B. Paley. TextArc: Showing Word Frequency and Distribution in Text. In *IEEE Symposium on Information Visualization, Boston, MA (Poster paper)*, 2002.

[102] B. Pang and L. Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL '04)*, pages 271–278. ACL, 2004.

[103] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[104] B. Pang, L. Lee, and 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)*, pages 79–86. ACL, 2002.

[105] Pew Internet & American Life Project, `http://www.pewinternet.org/Trend-Data/Online-Activites-Total.aspx`, last accessed on 4/19/2010.

[106] E. Pitler and A. Nenkova. Revisiting Readability: A Unified Framework for Predicting Text Quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 186–195. ACL, 2008.

[107] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05)*, pages 339–346. ACL, 2005.

[108] L. A. Ramshaw and M. P. Marcus. Text Chunking using Transformation-Based Learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora (WVLC-3)*, pages 82–94, 1995.

[109] RocketReader, `http://www.rocketreader.com/works/readability.html`, last accessed on 4/26/10.

[110] Readability Studio, `http://www.oleandersolutions.com/readabilitystudio.html`, last accessed on 4/26/10.

[111] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI '99)*, pages 474–479, 1999.

[112] E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP '03)*, pages 105–112. ACL, 2003.

[113] S. Rose, S. Butner, W. Cowley, M. Gregory, and J. Walker. Describing Story Evolution from Dynamic Information Streams. In *Proceedings of the IEEE Symposium on Visual Analytics and Technology (VAST '09)*, pages 99–106, 2009.

[114] V. Rubin and E. Liddy. Assessing Credibility of Weblogs. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs (CAAW '06)*, 2006.

[115] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[116] J. Schneidewind, M. Sips, and D. A. Keim. An automated approach for the optimization of pixel-based visualizations. *Information Visualization*, 6(1):75–88, 2007.

[117] T. Schreck, J. Schneidewind, and D. Keim. An Image-Based Approach to Visual Feature Space Analysis. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '08)*, 2008.

[118] S. E. Schwarm and M. Ostendorf. Reading Level Assessment Using Support Vector Machines and Statistical Language Models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*, pages 523–530. ACL, 2005.

[119] C. Seifert, B. Kump, W. Kienreich, G. Granitzer, and M. Granitzer. On the Beauty and Usability of Tag Clouds. In *Proceedings of the 2008 12th International Conference Information Visualisation (IV '08)*, pages 17–25. IEEE Computer Society, 2008.

[120] Y. Seki, K. Eguchi, and N. Kando. *Multi-Document Viewpoint Summarization Focused on Facts, Opinion and Knowledge*, pages 317–336. Theory and Applications (The Information Retrieval Series). Springer, 2005.

[121] L. Si and J. Callan. A Statistical Model for Scientific Readability. In *Proceedings of the tenth International Conference on Information and Knowledge Management (CIKM '01)*, pages 574–576. ACM, 2001.

[122] H. Sichel. On a Distribution Representing Sentence-Length in Written Prose. *Journal of the Royal Statistical Society (A)*, 137:25–34, 1974.

[123] S. Siersdorfer, A. Kaster, and G. Weikum. Combining Text and Linguistic Document Representations for Authorship Attribution. In *SIGIR Workshop: Stylistic Analysis of Text for Information Access (STYLE)*, 2005.

[124] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[125] A. Spoerri. InfoCrystal: a visual tool for information retrieval & management. In *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM '93)*, pages 11–20. ACM, 1993.

[126] Stanford Log-linear Part-Of-Speech Tagger, `http://nlp.stanford.edu/software/tagger.shtml`.

[127] J. Stasko, C. Görg, and Z. Liu. Jigsaw: Supporting Investigative Analysis through Interactive Visualization. *Information Visualization*, 7(2):118–132, 2008.

[128] J. Stasko and E. Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization 2000 (InfoVis '00)*, 2000.

[129] A. Stoffel, D. Spretke, H. Kinnemann, and D. A. Keim. Enhancing document structure analysis using visual analytics. In *Proceedings ACM Symposium on Applied Computing 2010 (SAC '10)*, 2010.

[130] H. Strobelt, D. Oelke, C. Rohrdantz, A. Stoffel, D. A. Keim, and O. Deussen. Document Cards: A Top Trumps Visualization for Documents. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1145–1152, 2009.

[131] R. T. Tally. Review of Franco Moretti's Graphs, Maps, Trees: Abstract Models for a Literary History. *Modern Language Quarterly (MLQ)*, 68(1):132–135, 2007.

[132] K. Thiel, F. Dill, T. Kotter, and M. R. Berthold. Towards Visual Exploration of Topic Shifts. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, (ISIC '07)*, pages 522–527, 2007.

[133] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* National Visualization and Analytics Center, 2005.

[134] I. Titov and R. McDonald. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT '08)*, pages 308–316. ACL, 2008.

[135] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03)*, pages 173–180. ACL, 2003.

[136] K. Toutanova and C. D. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC '00)*, pages 63–70, 2000.

[137] 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)*, pages 417–424. ACL, 2002.

[138] M. Twain. *Following the Equator.* National Geographic, 2005.

[139] H. van Halteren, F. Tweedie, and H. Baayen. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.

[140] F. van Ham, M. Wattenberg, and F. B. Viégas. Mapping Text with Phrase Nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176, 2009.

[141] P. Velardi, M. Missikoff, and R. Basili. Identification of relevant terms to support the construction of domain ontologies. In *Proceedings of the Workshop on Human Language Technology and Knowledge Management*, pages 1–8, 2001.

[142] F. Viégas, M. Wattenberg, and K. Dave. Studying Cooperation and Conflict between Authors with history flow Visualizations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*, 2004.

[143] F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory Visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, 2009.

[144] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar. What's Being Said Near "Martha"? Exploring Name Entities in Literary Text Collections. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*, pages 107–114, 2009.

[145] F. Wanner, C. Rohrdantz, F. Mansmann, D. Oelke, and D. A. Keim. Visual Sentiment Analysis of RSS News Feeds Featuring the US Presidential Election in 2008. In *Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW '09)*, 2009.

[146] F. Wanner, C. Rohrdantz, F. Mansmann, A. Stoffel, D. Oelke, M. Krstajić, D. A. Keim, D. Luo, J. Yang, and M. Atkinson. Large-scale Comparative Sentiment Analysis of News Articles. In *IEEE Symposium on Information Visualization (InfoVis 2009), Posterpaper.*

[147] C. Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2004.

[148] C. Ware. *Visual Thinking for Design.* Morgan Kaufmann, 2008.

[149] M. Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '02)*, pages 110–116. IEEE Computer Society, 2002.

[150] M. Wattenberg and F. B. Viégas. The Word Tree, an Interactive Visual Concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008.

[151] W. Weerkamp and M. de Rijke. Credibility improves topical blog post retrieval. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics: Human Language Technologies (ACL 08)*, 2008.

[152] A. M. Wensel and S. O. Sood. VIBES: Visualizing Changing Emotional States in Personal Stories. In *Proceeding of the 2nd ACM International Workshop on Story Representation, Mechanism and Context (SRMC '08)*, pages 49–56. ACM, 2008.

[153] J. Wise. The ecological approach to text visualization. *Journal of the American Society for Information Science*, pages 1224–1233, 1999.

[154] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings of the 1995 IEEE Symposium on Information Visualization (InfoVis '95)*, pages 51–58, 1995.

[155] H. F. Witschel. *Terminologie-Extraktion: Möglichkeiten der Kombination statistischer und musterbasierter Verfahren.* Content and Communication: Terminology, Language Resources and Semantic Interoperability. Ergon Verlag, Würzburg, 2004.

[156] WordNet, `http://wordnet.princeton.edu/`.

[157] H. Yang, J. Callan, and L. Si. Knowledge Transfer and Opinion Detection in the TREC 2006 Blog Track. In *TREC*, 2006.

[158] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM '03)*, pages 427–434, 2003.

[159] V. H. Yngve. A Model and an Hypothesis for Language Structure. In *Proceedings of the American Philosophical Society*, volume 104, 1960.

[160] H. Yu and 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)*, pages 129–136. ACL, 2003.

[161] C. Zhai, A. Velivelli, and B. Yu. A Cross-Collection Mixture Model for Comparative Text Mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 743–748, 2004.

[162] J. Zhang, C. Chen, and J. Li. Visualizing the Intellectual Structure with Paper-Reference Matrices. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1153–1160, 2009.

[163] J. Zhang, Y. Kawai, T. Kumamoto, and K. Tanaka. A Novel Visualization Method for Distinction of Web News Sentiment. In *10th International Conference of Web Information Systems Engineering (WISE '09)*, pages 181–194, 2009.

[164] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie Review Mining and Summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM '06)*, pages 43–50, 2006.

# Index