Transparency in Interactive Feature-based Machine Learning: Challenges and Solutions

Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften

vorgelegt von

Florian Stoffel

an der

Universität
Konstanz

Mathematisch-Naturwissenschaftliche Sektion

Informatik und Informationswissenschaft

Konstanz, 2018

Tag der mündlichen Prüfung: 5. Oktober 2018

1. Referent: Prof. Dr. Daniel A. Keim

2. Referent: Prof. Dr. Bela Gipp

Abstract

Machine learning is ubiquitous in everyday life; techniques from the area of automated data analysis are used in various application scenarios, ranging from recommendations for movies over routes to drive to automated analysis of data in critical domains. To make appropriate use of such techniques, a calibration between human trust and trustworthiness of the machine learning techniques is required. If the calibration does not take place, research shows that disuse and misuse of machine learning techniques may happen.

In this thesis, we elaborate on the problem of providing transparency in feature-based machine learning. In particular, we outline a number of challenges and present solutions for transparency. The solutions are based on interactive visual interfaces operating on feature-level. First, we elaborate on the connection between trust and transparency and outline the fundamental framework that builds the ground for this thesis and introduce different audiences of transparency. In the following, we present interactive, visualization and visual analytics-based solutions for specific aspects of transparency. First, the solution for the task of error analysis in supervised learning is presented. The proposed visual analytics system contains a number of coordinated views that facilitate sensemaking and reasoning of the influence of single features or groups of features in the machine learning process. The second solution is a visualization technique tailored to the interactive, visual exploration of ambiguous feature sets that arise in certain machine learning scenarios. Statistical and semantical information is combined to present a clear picture of the targeted type of ambiguities that can be interactively modified, eventually leading to a more specific feature set with fewer ambiguities. Afterward we illustrate how the concept of transparency and observable behavior can be of use in a real-world scenario. We contribute an interactive, visualization-driven system to explore a spatial clustering, giving the human control of the feature set, feature weights, and associated hyperparameters. To observe different behaviors of the spatial clustering, an interactive visualization is provided that allows the comparison of different feature combinations and hyperparameters. In the same application domain, we contribute a visual analytics system that enables analysts to interactively visualize the output of a machine learning system in context with additional data that have a common, spatial context. The system bridges the gap between the analysts utilizing a machine learning system and users of the results, which in the targeted scenario are two different user groups. Our solutions show that both groups profit from insights in the feature set of the machine learning. The thesis concludes with a reflection regarding further research directions and a summary of the results.

Zusammenfassung

Maschinelles lernen ist im Alltag allgegenwärtig. Techniken aus dem Bereich der automatischen Datenanalyse werden in verschiedenen Anwendungsbereichen genutzt, von der Empfehlung von Filmen, über zu fahrende Routen, bis hin zur automatischen Datenanalyse in kritischen Anwendungsbereichen. Um solche Analysetechniken sinnvoll einsetzen zu können, muss das entgegengebrachte Vertrauen des Nutzers der Vertrauenswürdigkeit der Methoden des maschinellen Lernens angeglichen werden. Forschung in diesem Bereich zeigt, dass falls dieser Prozess nicht stattfindet, Missbrauch und Nichtgebrauch der entsprechenden Systeme stattfinden kann.

Diese Arbeit befasst sich mit dem Problem, wie Transparenz bei merkmalsbasiertem maschinellem Lernen hergestellt werden kann. Insbesondere werden dazu Herausforderungen, als auch konkrete Lösungen, die auf interaktiver Visualisierung von Merkmalen basieren, vorgestellt. Zu Beginn wird die Verbindung von Vertrauen und Transparenz dargestellt, was Teil eines grundlegenden Konzeptes ist, dass die Grundlage für diese Arbeit stellt. Danach präsentieren wir eine Lösung für die Analyse von Fehlern in einem Problem des überwachten Lernens. Die Lösung besteht aus verschiedenen, miteinander verlinkten Visualisierungen, die es ermöglichen, den Einfluss einzelner Merkmale oder Merkmalsgruppen auf den Prozess des maschinellen Lernens zu erfassen und zu verstehen. Die zweite Lösung, eine Visualisierungstechnik, befasst sich mit dem Problem von mehrdeutig Merkmalen, wie sie in bestimmten Bereichen des maschinellen Lernens vorkommen können. Statistische und semantische Information wird kombiniert, um eine Abbildung der anvisierten, mehrdeutigen Merkmale zu ermöglichen. Es ist möglich, die Merkmale interaktiv zu modifizieren, was schlussendlich zu einer spezifischeren Menge von Merkmalen führt. Danach illustrieren wir den Nutzen des Transparenz-Konzeptes und der Möglichkeit, das Verhalten der automatischen Datenanalyse zu beobachten, anhand eines Echtweltprojektes. Dazu wird interaktive Visualisierung verwendet, um ein räumliches Clustering zu generieren und zu explorieren. Der Analyst hat dazu die Kontrolle über die verwendeten Merkmale, Merkmalsgewichtungen und die Hyper-Parameter der angewendeten Methodik. Um die verschiedenen Verhaltensweisen des Verfahrens beobachten zu können wurde eine interaktive, vergleichsbasierte Visualisierung entwickelt, die die Auswirkungen der verschiedenen Parameter sichtbar machen kann. Im selben Anwendungsfall stellen wir ein Visual Analytics System vor, dass die interaktive Visualisierung eines weiteren Verfahrens des maschinellen Lernens und weiterer Daten, die einen gemeinsamen geografischen Bezug haben, in einem gemeinsamen Kontext ermöglicht. Das System schlägt eine Brücke zwischen den Analysten, die Berechnungen mittels maschinellen Lernen anstellen und dessen Nutzern, die in dem anvisierten Szenario unterschiedliche Gruppen sind. Beide Gruppen profitieren von Einblicken in die verwendeten Merkmale des Verfahrens zum maschinellen Lernen. Die Arbeit endet mit einer Rekapitulation hinsichtlich weiterer Forschungsfragen und einer Zusammenfassung der Ergebnisse dieser Arbeit.

Acknowledgements

First and foremost, I want to thank my supervisor Daniel A. Keim. He gave me the freedom to conduct my own research and his continuous support — which finally lead to this thesis. Without him, this work would not have been possible. I would also like to thank my secondary advisor, Bela Gipp, who supported me during my work and was a great discussion partner for not only research and work-related matters.

I would like to thank my colleagues Dominik Jäckle, Juri Buchmüller, Wolfgang Jentner, Dominik Sacha and Halldór Janetzko for great discussions, joint work in the different research projects, collaborative paper writing, collective late night shifts, countless meetings in various coffee corners, and the joint #phdlife in general. Also, I would like to thank Peter Bak, Michael Behrisch, Fabian Fischer, Johannes Fuchs, Bum Chul Kwon, Florian Mansmann, Sebastian Mittelstädt and Christian Rohrdantz who supported me in various aspects during my time as a Ph.D. student and before. I also like to thank all my colleagues from the DBVIS group, the DBVIS support team, and all students that worked with me during my time at the University of Konstanz. It was a great and inspiring time working with all of you!

I had the chance to work in various research projects with different backgrounds and application fields, which was equally interesting and challenging. Besides the concrete project work, they provided me with valuable experiences besides pursuing my research and broadened the focus of my thinking. Among those projects, the collaboration with the State Office for Criminal Investigation of North Rhine-Westphalia in Germany stands out. There, I had the unique opportunity to solve real-world problems by pursuing and applying my research and was faced with new problems that we approached and solved together. I want to thank Daniela Pollich, Felix Bode, Marcus Stewen, Hanna Post and all the other people that I met for an exceptional collaboration, interesting and fruitful discussions, new insights, further inspiration to continue working in the field of data analysis and visualization, and for being there even after work.

Without steady support from my parents, Jožica and Peter, my sister Martina and my brother Andreas, all of this would not have been possible. Thank you.

Contents

1

Introduction

Techniques subsumed under the term *Machine Learning* are ubiquitous. They are part of everyday life and are essential parts of our surrounding, e.g., in the form of recommender systems or intelligent personalization on platforms such as Netflix [27], Amazon [11] or Google Maps [24]. It does not matter whether we are interested in watching movies, buying goods online or we plan a route from a place to another. Nowadays, machine learning is an essential part of the underlying data analysis, controls the adaption of a user interface, or computes the suggestions presented by the different platforms.

One of the earliest works that concretizes the idea of machine learning has been published by Arthur L. Samuel, one of the pioneers artificial intelligence and computer games. In a publication from 1959 about a program he *taught* to play checkers, he introduces the concept of teaching the game to a machine, i.e., the machine learns the game. His main point is, that machine learning can solve the complex problem of modeling every possible state of the game, and the corresponding potentially very large universe of moves. He states that having "*computers [able] to learn from experience should eventually eliminate the need for much of [. . .] detailed programming effort*" that is a "*time-consuming and costly procedure*" [193]. This statement and further ideas in the work of Samuel implies that a computer with the ability to learn should apply generalization techniques to learn the rules of the game of checkers, instead of having every rule preprogrammed, which would degrade the application problem — namely to play the game of checkers — to a search problem. On the one hand, this implies a loss of control when comparing the generalized rules (the model) to manual programming of rule by rule, as it was state of the art in 1959. On the other hand, it releases human computer scientist or programmers from the tedious work of programming every rule manually and copes efficiently with the universe of potential moves. In the early history of modern computer science that was limited by computing power, memory and storage space, expressing such a statement and the corresponding insight is fascinating and visionary at the same time.

Today, almost 60 years later, the idea of having machines that are able to learn has expanded beyond reducing programming complexity in favor of the spent time and money. Machine learning is primarily motivated by the idea of having not to program *everything*, but instead utilize potentially vast amounts of data — *big data* — to *learn* from the real-world. *Machine learning* describes techniques that are able to generalize effects, e.g., correlations, or more general, patterns in data, based on the automated analysis of massive datasets. The generalization can be understood as the method to *teach*, or in current terminology, to *train* models, and follows a specific way of generalization. The output of such training phases is typically a trained — initialized — model, that, based on a specific technique, describes a certain effect or relation of phenomena contained in the given training dataset. Those models can then be used to apply what has been *learned* on new, unseen data.

According to Hastie et al., machine learning can be divided into two different kinds of methods [99]. The so-called *supervised learning*, which comprises techniques such as classification or regression. Generalization and training happens on data instances with respect to a target variable, such as an instance label (categorical) or a specific attribute value (numerical, continuous). The goal of the training process is to model the associations in the dataset that lead to a specific target variable. Contrary, methods that implement *unsupervised learning* do not have any labeling or target variable induced. Instead, their goal is to model the inherent structure of the dataset, and in consequence, they are used for different applications than supervised learning. Popular methods utilizing unsupervised learning are clustering techniques or methods to compute association rules.

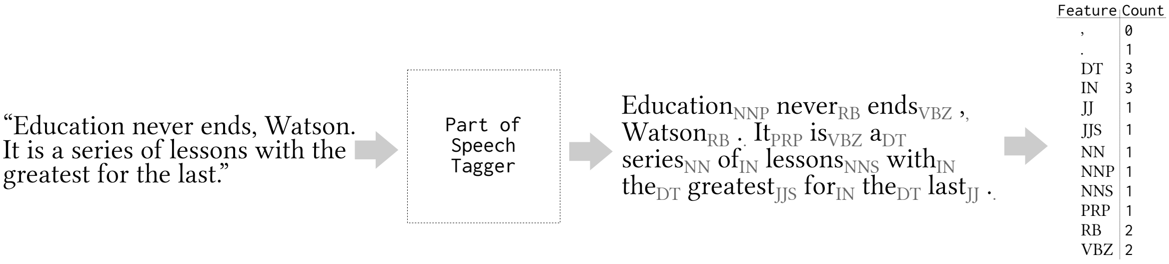

"Education never ends, Watson. It is a series of lessons with the greatest for the last."
Part of Speech Tagger
Education$_{\text{NNP}}$ never$_{\text{RB}}$ ends$_{\text{VBZ}}$,$_{,}$ Watson$_{\text{RB}}$.$_{.}$ It$_{\text{PRP}}$ is$_{\text{VBZ}}$ a$_{\text{DT}}$ series$_{\text{NN}}$ of$_{\text{IN}}$ lessons$_{\text{NNS}}$ with$_{\text{IN}}$ the$_{\text{DT}}$ greatest$_{\text{JJS}}$ for$_{\text{IN}}$ the$_{\text{DT}}$ last$_{\text{JJ}}$.$_{.}$
Feature Count
, 0
. 1
DT 3
IN 3
JJ 1
JJS 1
NN 1
NNP 1
NNS 1
PRP 1
RB 2
VBZ 2

Figure 1.1: *Illustration of a feature extraction process. In this example, the input text (left) is processed by a* part of speech tagger *(dashed box) that generates annotations with the part of speech per token (right of the dashed box). To be useful for specific tasks, e.g., similarity retrieval, the output is transformed into a feature vector (right) that contains the part of speech tags with their occurrences in the input text. Text from Arthur Conan Doyle's* The Adventure of the Red Circle.

Having a good, problem-specific representation of the data instances is crucial for machine learning techniques. Such representations are based on the instance attributes — features — of the dataset. In the simplest case, all data attributes are at the same time features, e.g., for a dataset containing only numerical attributes. For data that cannot be represented in such a forward fashion, for example, natural language text, features have to be computed (extracted) from the data instance attributes to make them fit to machine learning methods. Generally speaking, *feature extraction* [125] leads to a representation of the data instance in a potentially abstract feature space that is the input of machine learning methods. In Figure 1.1, a feature extraction of a text snippet is shown. The text is processed by a part of speech tagger (gray box) that assigns the corresponding tags, indicated by the gray subscript of each word on the right. The tags and their counts are then transformed into a feature vector (Figure 1.1 right) that contains the occurring part of speech tags and their number of occurrence in the processed text that can be used for further computations, for example, to assess the similarity of two documents based on the part of speech.

It is clear, that the quality of features has a considerable influence on the following application of machine learning techniques, as they form the data-wise foundation of the following automated generalization processes. Complementary, *feature selection* techniques [141] alter the feature space by selecting a subset of features based on some importance or quality criteria, e.g., to cope with the curse of dimensionality.

Feature-based machine learning processes follow a structure similar to the so-called KDD process shown in Figure 1.2.

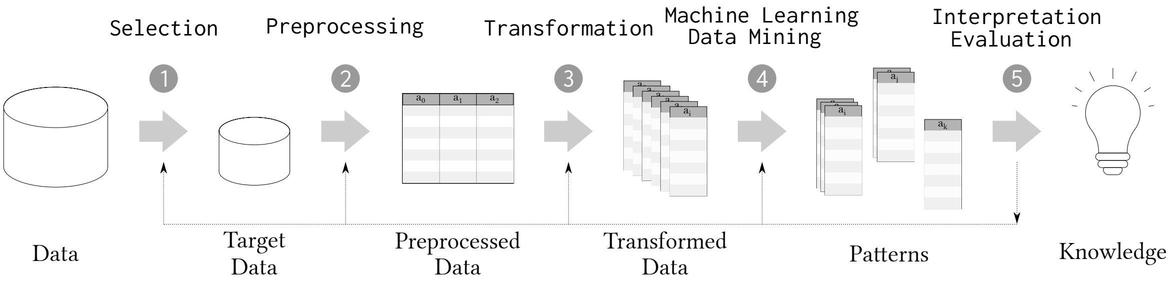


Figure 1.2: *The KDD Process proposed by Fayyad et al. [168]. It embeds data mining/machine learning techniques in a flow from data to knowledge and a feedback loop to all immediate phases of the process.*

In the beginning, a *selection* of data from a data pool, typically motivated by the application problem, is made to create the target data. Afterward, a single or potentially multiple *preprocessing* steps are preprocessing the target data. Then, a number of *transformation* steps is bringing the target data in a form suitable for the *data mining* task, that is typically described as a method to find *patterns* in the data, that are subject to later *interpretation* or *evaluation*. Eventually, this process generates knowledge about potentially unknown relations in the target data or helps to find and reveal previously unknown patterns.

Figure 1.2 also illustrates, that a number of techniques for data selection, preprocessing and transformation come together to be able to apply data mining or machine learning techniques to generate knowledge from a given dataset. Not only the methodological side but also the technical development goes in the direction of more and more complex solutions. This makes perfect sense, as for example, a non-linear regression is capable of being fit to nonlinear problems. But, having machine learning techniques that are getting overly complex and cannot be interpreted without additional tools, the question of how to recognize errors or non-optimal outcomes is getting more and more important. Having a system that is not comprehensible and has no facilities to be understood inevitably causes people not to trust the outcomes, in particular when the results are erroneous. This fact motivates our major research objective outlined in the following Section 1.1.

1.1 Objective

This thesis is based on the following very broad research question:

How can people trust machine learning methods?

We follow the idea of trust by Pedersen et al., as they elaborate on the term *trust* as follows: "*We exhibit an attitude of trust towards others as sources of information by acquiring beliefs through their testimony and by acting on them.*" [55]. Transferred to the context of this thesis, we understand *sources of information* as machine learning methods. Following the thoughts of Pedersen et al., the crucial aspect is that somehow it should be possible to *acquire beliefs* to trust anything a machine learning method is doing. Coming back to the introductory example of playing the game of checkers, human players should be able to assess the trustworthiness of the process powered by machine learning with their knowledge of the game. For example, as the rules of the game are well-defined, a human player should be able to assess whether a move makes sense or not, and adjust her level of trust accordingly.

The process of adjusting trust towards automated data analysis, and machine learning, in particular, serves as the main motivation for transparency. Lee and Moray introduced a framework that can be illustrated as shown in Figure 1.3.

Figure 1.3 describes the space between human trust on the y-axis and the trustworthiness, respectively the capabilities of an automated system, on the x-axis. When the user can assess the trustworthiness of the process correctly, i.e., has a *calibrated trust*, as indicated by the dashed bisecting line, the automated system is used in the most desirable way. In practice, this means that the tasks the automated system is entrusted with fit to the capabilities, and in consequence, the results can be trusted by the human. If this is not the case, e.g., the level of trust by humans is higher than the trustworthiness of the automated system, a situation located above the bisecting line, as exemplarily indicated by point A is present. Such a situation leads to misuse of the system. Contrary, a situation as indicated by B arises when the level of trust is lower than the trustworthiness of the system. The consequence is disuse of automated systems and its capabilities. The process of *trust calibration* denotes the shift of the trust level to the bisecting line, which indicates the right balance of trust and trustworthiness of the machine. Eventually, trust calibration leads to a proper mapping

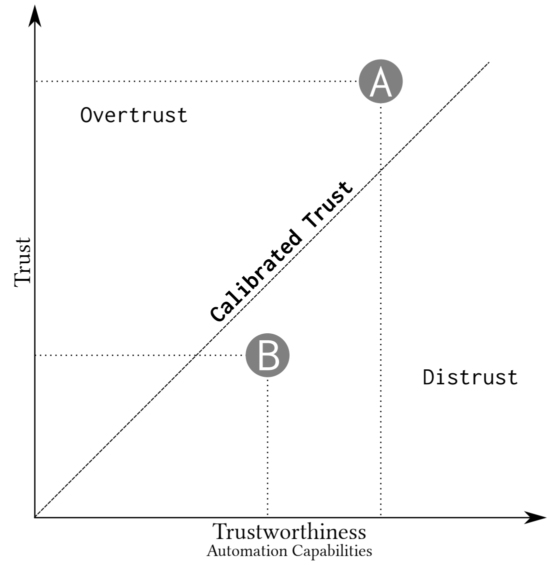


Figure 1.3: *Illustration of the relation between trustworthiness (automation capability) and trust according to Lee and Moray [172]. The points* A *and* B *illustrate situations of overtrust and distrust, respectively, that both lead to misuse or disuse of the automation capabilities. The area above the bisecting generally refers to situations of overtrust, below to distrust. A match between trust and trustworthiness is denoted as* calibrated trust.

of system capabilities and trust, which results in a responsible and appropriate usage of automation, or machine learning, as it is subject of this thesis. Transparency is a tool to enforce trust calibration, as it brings the inner workings to the human operator, improves the user's knowledge of the system and eventually triggers the trust calibration process.

Previous research showed that being able to adjust the individual level of trust is crucial [172, 59]. Advantages, such as the ability to find patterns in large datasets, or even the ability to cope with problems that require a large foundation of data, can get lost when the user/analyst does not trust an automated system or a machine learning method. When the subjective trust of the analyst and the objective trustworthiness of an automated system meet, the capabilities of automation and machine learning are utilized reasonably well [172].

Unfortunately, the world and the corresponding real-world problems are typically not as well defined as a board game, nor simple to understand, so an ad-hoc calibration of the user's trust is not possible, as there are no facilities to acquire belief about the employed automated analysis techniques. Even for the people that know about machine learning and have a reasonable mental model and corresponding knowledge of the inner workings of such methods, with increasing complexity of the techniques in question, it gets more difficult to *acquire belief* and ultimately adjust the level of trust in what such a method is doing, even in a research or development context. For example, there are linear classifiers such as Naïve Bayes [190] that can be subject for belief more easily compared to techniques, e.g., a non-linear Support Vector Machine [175]. Sticking with the problem of complexity, during the last few years, techniques utilizing artificial neural networks [161] have gained tremendous popularity [41], as latest incarnations show astonishing performances in various application domains [74, 66, 45, 28, 12]. It is obvious that with an increasing number of hidden layers, non-linear activation functions and the sheer number of neurons consuming input data and generating output data, getting insights in the machine learning processes and *acquire belief* is still an open research problem. A number of publications in the visualization domain are coping on this problem, e.g., by trying to enable analysts to *acquire belief* in the training process [7, 6, 5], or parts of the architecture of neural networks [18, 13]. Coming back to the initial example of the game of checkers by Samuel, Figure 1.4 that — in a more complex form — is part of the original publication illustrates the training process of the machine learning model and made the training *behavior* of the mathematical model that learned the rules *observable.* This proves that even in the early stages of machine learning, the *observability* of those processes was a concern.

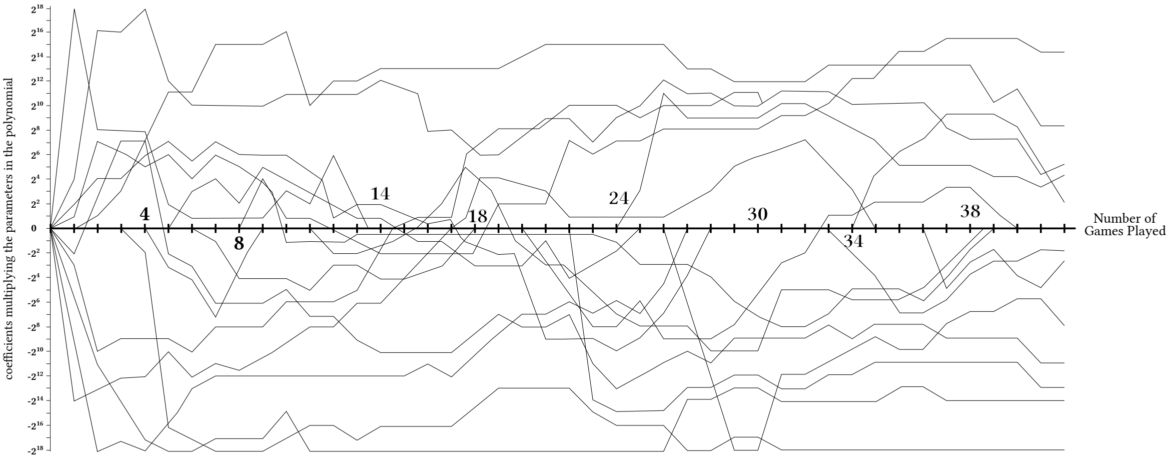


Figure 1.4: *Visualization of the learning phases, according to [193], each line corresponds to a coefficient of the mathematical model of the machine learning system. Of particular interests are the areas around move 14 where the initial signs of many terms change, as well as the area after move 30, where each term gains stability.*

Still, the question remains how people can trust machine learning methods, and how to support them in *acquiring belief* in those techniques. According to Fayyad et al., machine learning is a *tool to assist humans in extracting useful information (knowledge),* which brings humans and the interaction with a machine that does machine learning into the center of possible answers. Having a closer look at the ideas connected with human-machine interaction, and in particular the trust problem of such collaboration, there are various ways to try to reach the goal of calibrated trust in the sense of Lee and Moray.

In this thesis, we utilize what we call *transparency* to support the calibration of trust in different aspects. Mainly, this approach is motivated by Muir, as she states "*Behavior must be observable for trust to grow.*" [173]. Dzindolet et al. are elaborating in their article about reliance and automation with respect to trust, that "*users should be allowed to view how decisions are made*" [139]. Based on this ideas, we present methods to view how decisions are made or try to give analysts an idea of what the behavior of a machine learning method is, to *acquire belief* in what the employed technique is doing. In particular, we stick to the feature-level, as we follow the idea that without methodological insights, transparency can be effectively communicated via proxies, e.g., a methodological proxy that abstracts or aggregates what is happening in the machine learning process, or data level proxies, that are based on the set of features that a machine learning technique utilizes or produces. In the following section Section 1.1.1, we outline the concrete scope of this thesis in more detail.

1.1.1 Scope of the Thesis

The goal of exhibiting *observable behavior* of machine learning methods is very broad, as there are a number of different layers that can be exploited for that goal. i) pre-trained models that contain the generalized information from the training dataset, ii) the hyperparameters, which are essentially the parameters analysts can modify when applying machine learning, iii) internal parameters of the employed that are only exposed for internal purposes and are set automatically, and iv) the actual machine learning algorithm, for example, the method that performs the clustering or solves the classification problem.

Implementation details such as optimizations, fallback methods, or heuristics are a very technical layer of any implementation of an actual algorithm belong to the fourth layer.

All of these layers exhibit different, potentially algorithm and implementation specific details. When sticking to those details, this thesis would have been very narrow and specific to a certain machine learning technique and possibly its actual implementation. Instead, this thesis focuses on a different view, that at first glance seems much broader. We utilize the input and the output data of a machine learning method, e.g., the feature set and the computed class assignments, instead of the method itself, to *exhibit observable behavior.*

This approach has a number of advantages, compared to a purely technical and possibly algorithmically-based idea of behavior. First and foremost, all results from this thesis are agnostic to the actual machine learning method, which is a direct consequence of not utilizing technical details of the machine learning techniques to make *behavior observable.* Instead, we work on inputs and outputs of those algorithms and treat the part that does the machine learning as a black box. This approach makes sure that this thesis exhibits a level of generality that makes it possible to transfer the presented ideas to different application problems as well as their concrete algorithmic solution. Second, to base the idea of transparency purely on the input and output data of machine learning methods allows involving domain experts or domain-specific analysts in the process of designing and interpreting the results, as they are typically highly skilled in interpreting the data they are working with on a daily basis. At the same time, machine learning experts should be still able to observe specifics, expected or unexpected behavior of the utilized machine learning technique, as they know about the processing of data in detail. Lastly, this makes it possible that the solutions presented in this thesis can mostly be applied without the machine learning technique, given the results are available. For computationally expensive techniques this opens up new possibilities for interactions, as well as sensemaking with concerning the *exhibited behavior*, as a possibly time-consuming execution of the machine learning techniques is not required.

Having the data to exhibit behavior defined, the question remains how we achieve this level of exposed behavior, and make it observable for analysts. We utilize techniques from the field of visual data exploration, as "*the basic idea of visual data exploration is to present the data in some visual form, allowing the human to get insight into the data, draw conclusions, and directly interact with the data*" [148]. In addition, Keim states that "*visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters*" [148]. Therefore, visual interactive data exploration perfectly fits to the previously stated idea of involving users or analyst in the reasoning processes with respect to the inner workings of a machine learning system.

1.2 Thesis Structure and Scientific Contributions

Building upon the idea of visual analytics, the scientific contribution of this thesis is structured as follows.

In Chapter 2 introduces our general framework used to structure our research. In the beginning, we reason about the fundamental levels of machine learning that can be utilized to capture and communicate behavior from. The different levels to work with also refer to different groups of users, which in turn gives the motivation to identify different audiences for transparency. We identified three groups, namely *end users*, *application-level engineers* and *method-level engineers* that have different goals and tasks when using machine learning, which we outline in Section 2.1. In the following Section 2.2, we introduce different types of data that can be utilized to describe behavior in some form. It is important to note, that we do not understand the section as complete and finalized work. While machine learning is developed further, the application scenarios, as well as user task, are dynamic and virtually unlimited. In Chapter 2 we contribute starting points and challenges, as well as some elaborations on what is relevant for the research presented in this thesis.

Chapter 3 contributes an implementation of observability of a machine learning process, more precise a supervised classification problem. The application problem is to recognize features that are responsible for misclassification in a binary classification problem in the text domain. Large

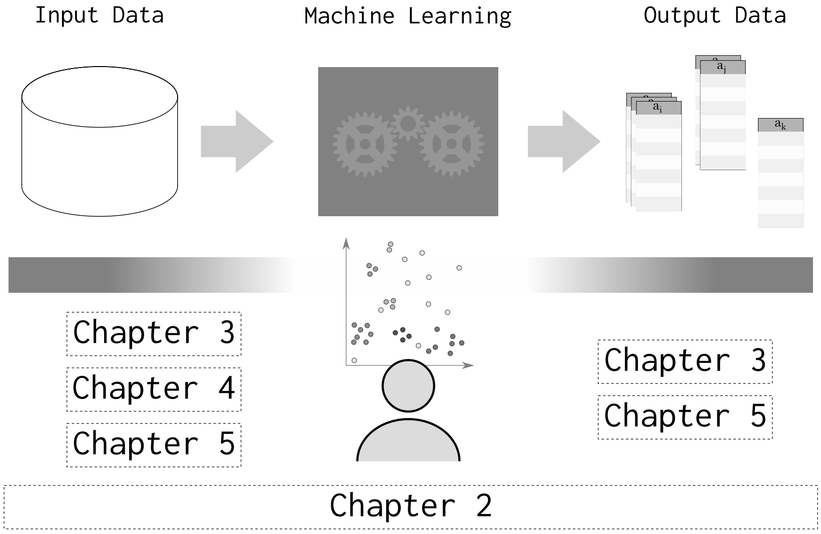


Figure 1.5: *Scope and structure of the thesis illustrated with a simplified machine learning workflow. Chapter 2 introduces the foundation and major challenges, while Chapter 3, Chapter 4, and Chapter 5 present solutions for aspects of transparent machine learning.*

feature sets (more than 1,000 features) on various levels of the text data are used to feed the machine learning process that classifies documents. Being able to have a closer look at the feature set in the light of errors enables domain analysts — *application-level engineers* — to reason about potential changes of the features, which is supported by an ad-hoc what-if analysis facility built into the visual, interactive exploration system called *Minerva*, that utilized multiple linked views to provide different perspective on the classification data. In this section, we contribute an interactive, visualization-based tool that enables application-level engineers to observe the behavior of the classification method concerning errors, which is supported by a novel what-if analysis that does not require the classification process to be run to see the impact of feature combinations on the result. We also contribute a structured way to approach this and similar problems that built upon the idea of close inspection of errors in machine learning processes. Finally, we demonstrate how our prototype leads to a better application performance using interactive, visual data exploration to observe the behavior of the classification model. Major parts of this research been published in the following publication:

> Florian Stoffel et al. "Feature-Based Visual Exploration of Text Classification". In: *Symposium on Visualization in Data Science (VDS) at IEEE VIS.* 2015

In the following Chapter 4, we elaborate on a solution that allows *application-level engineers* to disambiguate generated feature sets. At the example of natural language processing, or more precise at the example of named entity recognition, we illustrate how different properties of the data generated by a machine learning component can be utilized. We contribute a two-fold data approach that combines statistical and semantic information related to the generated, ambiguous features as the data to observe. Additionally, we contribute an extension of the well-known co-occurrence matrices for visualization that is enhanced with interaction to resolve ambiguities and provides a preview of what happens when the merge operation is translated to a machine learning model. At the example of the analysis of fictional literature, we showcase our approach. The generalizability is proven by the observation, that the proposed approach also provides insights in the authorship of documents and books, which is a hint that the contributed visualization technique can be utilized for other application scenarios that benefit from transparency, or in this case, observability with respect to the feature set. The research contributing to this chapter has been published in:

> Florian Stoffel et al. "Interactive Ambiguity Resolution of Named Entities in Fictional Literature". In: *Computer Graphics Forum* 36.3 (2017), pp. 189–200. ISSN: 1467-8659. DOI: 10.1111/cgf.13179

While and Chapter 2 and Chapter 3 showcase our scientific contributions with respect to the same audience, the *application-level engineer*, Chapter 5 illustrate transparency for feature-based machine learning in a real-world application scenario and a mix of users, *application-level engineers* as well as *end users*. We contribute techniques that approach a machine learning-based workflow for spatial predictions on different levels. The first technique is utilized by *application-level engineer* to produce a transparent, spatial clustering, and is part of Section 5.2. To achieve this, we contribute a workflow that involves interactive data exploration and interaction with the machine-learning output, while the most important determiners of the origin, the hyperparameters, are set by the analyst. Additionally, the visual analytics system implements a unique combination of clustering methods that have been developed in collaboration with the targeted user group. The implementation of the clustering method allows the users to run what-if analysis, e.g., by choosing different feature sets or hyperparameters such as the target density or expected number of data instances per cluster. A comparative visualization of the clusters enables the users to observe the effects of different hyperparameters or adjusted feature sets with respect to the employed machine learning approach. The second contribution is part of Section 5.3. Here, we showcase *polimaps*, a visual analytics system to explore the machine learning output, e.g., the spatial predictions, concerning user definable data that may or may not be part of the feature set used for the prediction task. The tool has been developed in close collaboration with *end users* and enables them to gain insights in the machine learning output using spatial data visualizations and data filtering facilities that give access to the spatial and temporal dimension of the datasets to explore. Although, the major task *polimaps* is designed for is the adaption and deployment of the machine learning output for the *end users*. At the end of Section 5.3, we illustrate that we contributed to the solution of the application problem with an informal user questionnaire Finally, Section 5.4 illustrates the problem of intransparent and in consequence inconclusive quality metrics to evaluate the performance of machine learning applications. We elaborate on state of the art in the application domain, argue about transparency issues of those metrics, and contribute a number of factors that need to be included to be able to evaluate predictions properly. The research of this chapter has been published in the following publications:

> Florian Stoffel et al. “Qualitätsmetriken im Bereich Predictive Policing: Die Variabilität und Validität von Trefferraten”. In: *Polizei & Wissenschaft* 4 (2017). Ed. by Clemens Lorei, pp. 2–15. ISSN: 1439-7404
>
> Florian Stoffel et al. “polimaps: Supporting Predictive Policing with Visual Analytics”. In: *EuroVis Workshop on Visual Analytics (EuroVA)*. ed. by Christian Tominski and Tatiana von Landesberger. The Eurographics Association, 2018. ISBN: 978-3-03868-064-2. DOI: `10.2312/eurova.20181111`

Significant parts dealing with police-relevant issues have been contributed by Felix Bode.

Chapter 6 concludes the thesis and reflects on various issues that we faced with during the research that contributed to this thesis. We outline further research directions and close with further remarks on open issues of the emerging field of transparency of machine learning.

1.2.1 Further Contributions

The advances made in the context of the research leading to this thesis have widespread into further publications that are not part of this thesis. In the following, we outline the publications and set them into context of the research agenda of this work.

> Florian Stoffel and Fabian Fischer. “Using a knowledge graph data structure to analyze text documents (VAST challenge 2014 MC1)”. In: *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014, Paris, France, October 25-31, 2014*. Ed. by Min Chen et al. IEEE Computer Society, 2014, pp. 331–332. ISBN: 978-1-4799-6227-3. DOI: `10.1109/VAST.2014.7042551`

In this publication, we showcase an early incarnation of the text analysis framework that provides transparency information and has been developed in conjunction with this thesis. The capabilities of transparency were used to link different entities together, and to provide a way to explore those connections on feature-level via graph-based visualizations.

Florian Stoffel et al. “VAPD - A Visionary System for Uncertainty Aware Decision Making in Crime Analysis”. In: *Symposium on Visualization for Decision Making Under Uncertainty at IEEE VIS 2015.* 2015

This publication outlines the concept of a system called VAPD that employs visualization and uncertainty information to augment decision making in crime analysis. The ideas with concerning the corresponding data Analysis and uncertainty were an outcome of the research contributing to this thesis.

Leishi Zhang et al. “Spherical Similarity Explorer for Comparative Case Analysis”. In: *Visualization and Data Analysis 2016, San Francisco, California, USA, February 14-18, 2016.* Ed. by David Kao et al. Ingenta, 2016, pp. 1–10

Wolfgang Jentner et al. “A Visual Analytics Approach for Crime Signature Generation and Exploration”. In: *The Event Event: Temporal & Sequential Event Analysis, IEEE VIS 2016 Workshop.* 2016

Dominik Sacha et al. “Visual Comparative Case Analytics”. In: *EuroVis Workshop on Visual Analytics (EuroVA).* ed. by Michael Sedlmair and Christian Tominski. The Eurographics Association, 2017. ISBN: 978-3-03868-042-0. DOI: `10.2312/eurova.20171119`

Wolfgang Jentner et al. “Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool”. In: *The Visual Computer* (Feb. 2018). ISSN: 1432-2315. DOI: `10.1007/s00371-018-1483-0`

Lucie Flekova et al. “Content-based Analysis and Visualization of Story Complexity”. In: *Visualisierung sprachlicher Daten.* Ed. by Noah Bubenhofer and Kupietz Marc. Heidelberg University Publishing, 2018. Chap. 7, pp. 185–223. ISBN: 978-3-946054-75-7. DOI: `10.17885/heiup.345.474`

All of these publications [31, 29, 19, 4, 3] are based on the outcomes of the machine learning framework developed in conjunction with this thesis. Without the transparency back-end that provides provenance information on feature-level, this would not have been possible.

2

Transparent Feature-based Machine Learning

Transparent machine learning, as understood in this thesis, is a label for techniques that allow insights or provide facilities to observe the behavior of the algorithms and methods deployed for training, validation, or the application of machine learning methods in general.

As introduced in Chapter 1, different layers are part of a typical machine learning approach, which are:

1. pre-trained models,
2. hyperparameters,
3. algorithms and corresponding parameters, and the
4. machine learning algorithms.

For application problems that are solved with machine learning, two further layers are part of the solution for the application problem, which are:

5. a machine learning workflow (pipeline), possibly containing out of many different data analysis components, and finally
6. the associated data, e.g., the input and output data of the whole machine learning process.

Those different layers can be exploited to provide insights into different aspects of the machine learning process. All of them require a different level of insight and understanding of the employed methods and are therefore suitable for different kinds of transparency, or a different audience correspondingly. For example, to make sense out of the hyperparameters that are determined by various utilized components, the analyst must know what they represent, which in turn requires some methodical knowledge. There is a considerable difference between selecting the number of clusters to identify, e.g., when utilizing a k-means clustering algorithm, and the setting of the $minPoints$ or ϵ parameters of a density-based clustering algorithm. Those parameters have different implications and requirements. For example, the absolute ϵ-distance can be very different in low and high-dimensional data spaces.

Therefore, we argue that to be able to expose observable behavior to an interested party for increased transparency, the actual behavior to expose should be made dependent on the envisioned user group, which is subject to the next section.

2.1 Audiences for Transparency

As argued in the introduction, having different layers of the machine learning method that exposes some different behavior also implicates different user groups or audiences. It is clear that end users from e-commerce platforms are not interested in errors or specific parameters of the machine learning workflow that aggregates a large number of product reviews. The level of abstraction from the user goal, which is to get an overview of product reviews, is entirely different from the selected models to analyze the part of speech of words, or the hyperparameters of the clustering method that assigns a group (cluster) to each review. Still, the latter can be useful when testing, training or validating a machine learning workflow — which is done either by a developer of the actual machine learning techniques or an engineer who trains models or builds a machine learning-based solution for an application problem. This simple example already motivates three different audiences for transparent machine learning:

- *End users*: includes potential users of machine learning methods.
- *Application-level engineers*: designs, configures and utilizes machine learning methods to solve an application problem.
- *Method-level engineers*: designs machine learning techniques.

Those groups are distinguishable by their different goals and in consequence also different requirements when it comes to transparent machine learning.

End users are interested in a pleasant user experience, which, in turn, depends on various, potential subjective factors. While there are some approaches to communicate the quality of machine learning methods, read: uncertainty, those are usually not based on the underlying data analysis methods in the end user context. For example, some big online-retailers use star-schemes or similar approaches generated by the end users to capture the helpfulness or quality of an analysis output, instead of trying to exploit metrics from the employed techniques. This impression is backed by the fact, that popular literature from connected fields such as opinion or sentiment mining does not cover ideas such as uncertainty that can contribute to exposing quality information, and therefore could be part of *observable behavior* [76, 36].

For application-level engineers, a more technical and possibly process-oriented view is appropriate, e.g., to be able to debug a machine learning process or to understand specific parts of what is happening during the machine learning process. A framework to provide insights into technical details and process-like operations is provided by Foster et al. [147]. They introduce a provenance data scheme as well as a query language, VDL, that is used to query and retrieve provenance data, that can be used to describe machine learning processes in varying level of details. Existing work in the data provenance field concentrates on process/workflow descriptions [147, 136, 144, 150, 122], data auditing [147], replication [147, 136] and propagation [122] aspects.

Finally, the method-level engineer is part of the technical audience. At this stage, questions about the correct behavior of a machine learning technique, as well as algorithmic correctness arise and need to be answered. The primary distinguishing aspect of this group compared to the application-level engineer is the fact, that the method developer is interested in solving an abstract problem that is not necessarily part of an application problem. For example, the clustering of a number of data records is an abstract problem and already motivates many different approaches to solve that problem on an abstract level. Although, having a clustering algorithm that in principle can partition a dataset into different groups of records, does not — and does not have to — solve a concrete application problem per se. Instead, the methodical or technical challenge motivates their activities, which requires a strong technical background that implicates prior knowledge of the behavior of a machine learning process in detail. Also, being able to work with the machine learning methods on such a fundamental level, there are a different of tools such as a debugger or debug-level logfiles that give an insight on the actual, technical behavior of the data analysis processes. Therefore, we implicate that the group of method-level engineers is not the main minor interested audience for transparent machine learning in general.

Having distinguishable audiences brings up the question if there are users that could be associated with more than one of the described groups. A possible characterization of such a group of users could be, that users are mostly interested in the results from a machine learning process, but also care about the methodological details to some extent — or vice versa. Motivated by changing or non-clearly separated goals when applying machine learning techniques, the intention of the audience is the driving factor for difference or commonalities of their requirements when it comes to observable behavior or transparency. This characterization fits the group of people most of the work presented in this thesis, namely analysts that are working in a specific application domain that utilize machine learning methods to solve their application problem. From working and discussing with those users of machine learning systems we know, that all hypotheses, findings, and decisions must be grounded in some data and must be explainable to other analysts, colleagues or supervisors that have not followed the sensemaking processes from the analysts. Having mostly a black box machine learning-powered system may solve their problems, but is not able to provide the required justification, which is a criterion to stick to methods and techniques that do not use machine learning — and a strong motivation for providing transparency concerning the machine learning processes.

2.2 Describing Behavior

In the previous section, we argued that the audience of observable behavior can be different, and separates into at least three high-level groups. It is also clear that they have different requirements for a transparent data analysis process, which in particular is true concerning the data that is used to describe the behavior.

Coming back to the introduction of this chapter, we argued that for an application problem solved with machine learning, six layers of behavior can be differentiated from each other, which are:

1. pre-trained models,
2. hyperparameters,
3. algorithms and corresponding parameters,
4. machine learning algorithms,
5. a machine learning workflow (pipeline), possibly containing out of many different data analysis components, and finally
6. the associated data, e.g., the input and output data of the whole machine learning process.

Pre-trained models are subject to a train and test methodology, meaning that they are trained with some partition(s) from the dataset, and tested with other parts from the data where both partitions are mutually exclusive. The resulting performance scores, for example, an *accuracy* metric, gives some insights into the performance that can be expected from a machine learning algorithm utilizing the corresponding model. Hyperparameters can already expose some ideas about the behavior of the method that is steered by the parameters. Although, as this requires expert knowledge on the methodological level, we stick to the interpretation of hyperparameters as determining factors for the behavior of an automated data analysis process. They enrich the behavior information as they are one of the major driving forces, but they do not describe behavior in detail. The same is true for the general algorithms contained in a machine learning workflow and their corresponding parameters.

Still, it is clear that this information needs to be part of the behavior description, e.g., as metadata, as the employed algorithms and the (hyper)parameters are essential determiners for the overall behavior and functionality of a machine learning workflow. The most determining force of behavior in a machine learning solution is the actual machine learning algorithm. While it is an integral part of the solution to select an appropriate algorithm, there are of course fundamental differences in behavior that can be observed of algorithms that solve the same problems. Having a look at clustering, this is clear while looking at a decision tree-based technique, or a density-based clustering technique. They differ in capabilities and their overall methodology as well as hyperparameters, but solve the same application problem — and expose different kinds of data. Undoubtedly, a decision tree is best described by its natural form of a tree, where each node represents an attribute or attribute combination, the corresponding split criterion and the children below the node. For a density-based clustering algorithm, the data that can be generated to describe behavior depends on hyperparameters that initialize reachability distances, data records, and their cluster memberships, and data records classified as noise, if applicable.

The machine learning workflow is itself a natural way to describe the behavior of a machine learning solutions, namely the workflow itself. There are different approaches to describe workflow provenance, on data level, e.g., as provenance graphs [106, 113, 91], or in explicitly visualization driven ways [112, 90, 60]. All of these works understand the application-dependent machine learning workflow as a blueprint of processes that interact with each other by input and output data. Each of the connected components is part of the provenance data that documents the corresponding, specific parts of the analysis workflow they are responsible for. This kind of data facilitates a number of different findings to be made from the collected data, for example, the high-level inspection of the involved components and their overall appropriateness. Additionally, a detailed analysis of single components can be conducted, e.g., when the workflow per component is enriched with the input and output data. This is in particular interesting for components that do a preprocessing of the data, e.g., to fill in missing data to judge the strategy of the now filled-in data values, as well as to observe when changes of the data occur, for example in preprocessing stages of natural language processing.

Finally, the behavior of a machine learning component can be made visible based on the data that is subject to the component. Fundamentally, there are two different kinds of data, the input and output data. As argued before, the input data may contain the unmodified data records from the original dataset, as well as a corresponding number of feature vectors that encode the data records in a suitable form for the machine learning algorithms, or a mixture of both. For example, the values of a data attribute can be subject to a discretization or normalization that, in turn, can be regarded as a feature that is derived from the original value. After applying the machine learning algorithm, typically some output is produced. For example, for a classification problem, this is typically a new attribute that assigns a cluster to a data record, which can be used to reconstruct the different clusters and the contained data records. Due to the nature of the input and output data, it is clear that a domain expert that handles similar data all day as work routine has some intuitions about the result of a machine learning process. For example, having a clustering application, some ideas about the data separation or cluster memberships of the data records could be present. Additionally, it can also be the case that because of extensive experience with the data over time, a prediction result could be anticipated by the data expert. For this, no methodical or technical knowledge is required. Instead, the anticipation and expectations of an analysis output in the light of the input can be based on expert knowledge based on the data itself. This fact makes this kind of data quite interesting to describe behavior, as, for example, the quality of a classification output can be measured by comparing the output data to a gold standard, e.g., by *precision*, *recall*, or an *F-measure* [109]. Similarly, the quality of a clustering can be expressed by the cluster silhouettes [179], validity measures [151] that typically evaluate clusters separately from each other in the light of a specific validity measure, or global separation measures [188]. These metrics, solely based on the input/output data, do not rely on the inner workings of machine learning solution, but still, give an impression of the quality, and therefore one aspect of the behavior of the machine learning technique.

Machine Learning Level	**Exposed Behavior**
pre-trained model	validation/test quality
hyperparameters	parameter value
algorithms & parameters	parameter value
machine learning algorithms	methodology
machine learning workflow	provenance data
input & output data	quality metrics

Table 2.1: *Overview of levels of machine learning and corresponding exposed behavior.*

Table 2.1 lists the discussed separate machine learning layers and the corresponding behavior data that can be observed. While they are all separated, for applications that rely on observable behavior, a mixture of them could be interesting. For example, it is clear that when a machine learning method is replaced with a different one, better generalization capabilities of the learned models could lead to different performance characteristics of the input and output data. The machine learning workflow, combined with the different contained algorithms can be utilized to compute various metrics of the problem solution, e.g., the class label of a data record. It can be interesting to compute the uncertainty or certainty scores of different, consecutive machine learning components when they refer to the same piece of information using many different methods that range from univariate statistics to complex simulations [108]. Therefore, depending on the application problem and the required transparent, observable behavior, combinations of different levels are possible and can be utilized accordingly.

2.3 Making Behavior Observable

Having some different data sources of behavior introduced in Section 1.1.1, the observability of this information to achieve a degree of transparency is crucial.

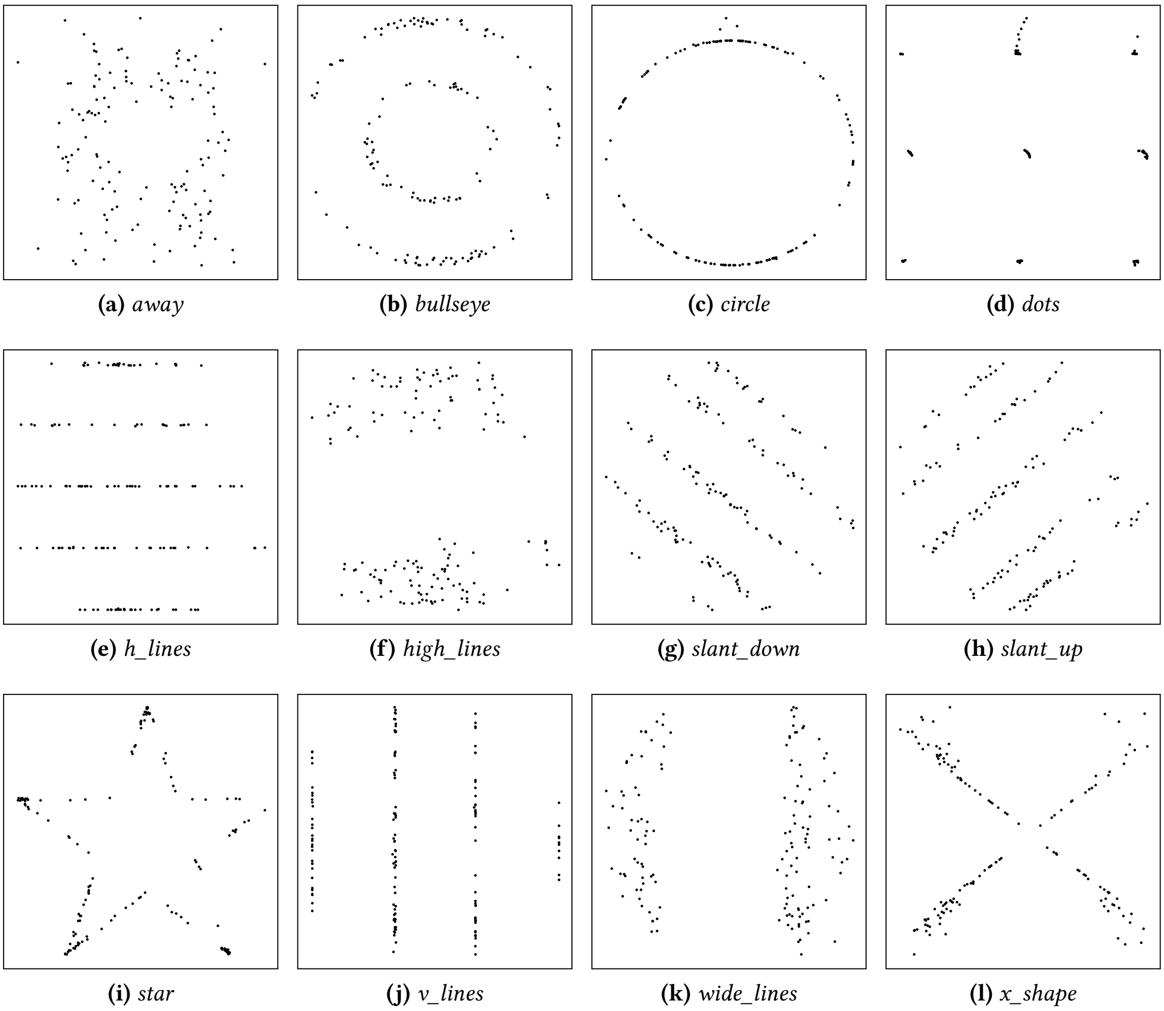
Appropriate techniques to make the behavior data observable is the raw presentation of the data itself. When having a look at real-world application solutions based on machine learning, they typically are applied to large datasets, e.g., for training and in productive use. Having to deal with large amounts of data instantly leads to doubts concerning the scalability of data presentations, which in the end makes the behavior observable.

Therefore, we chose interactive visualization as the toolbox for making behavior data observable. More precisely, we opt to follow the lines of research in the field of *Visual Analytics*, that is to involve humans, the human perception, potential expert or prior knowledge, and interaction to explore potentially large datasets [56]. The idea of visual analytics is that interactive visual interfaces allow human analysts to effectively explore datasets and come up with findings that are expected or unexpected. With the help of interaction, the findings evolve to a hypothesis that can be verified or falsified and finally contributes to human knowledge. As a discipline, visual analytics brings together data visualization, interaction and human cognition to foster reasoning processes with the goal of generating new knowledge.

As argued, the primary reason why we rely on visual analytics to make the behavior of a machine learning process observable lies in the scalability problems of standard tabular interfaces. Ellis and Dix show that typically, data visualization techniques have no upper limit concerning the number of displayed data records — and if they do, techniques to reduce clutter and overplotting exist to cope with such problems. That property is a significant advantage of visual interfaces compared with tabular like displays, that may provide a space-efficient representation of a large dataset, but the interaction to navigate in the data — to scroll on the horizontal or vertical axis — makes them tedious to use. While in tabular interfaces, human interactions are required to scroll through the data, interaction in data visualization is primarily used for data exploration, which is part of the data exploration activities [186] that visual analytics is building upon.

Besides the technical advantages of visualization, when it comes to interpreting a dataset, non-visual methods typically rely on aggregation and provide a variety of summary statistics, e.g., statistical moments, diversity or in general measures that give the degree of statistical dispersion [99]. Why this falls short and to rely purely on summary statistics is not an appropriate tool to explore data is illustrated in Figure 2.1.

Anscombe already illustrated in 1973 why a purely statistical analysis of datasets is not enough for exploration tasks [189]. More recently, Matejka and Fitzmaurice extended the work of Anscombe



(a) *away* **(b)** *bullseye* **(c)** *circle* **(d)** *dots*

(e) *h_lines* **(f)** *high_lines* **(g)** *slant_down* **(h)** *slant_up*

(i) *star* **(j)** *v_lines* **(k)** *wide_lines* **(l)** *x_shape*

Figure 2.1: *Illustration of the* Datasaurus Dozen *[14]. Each of the given images has very similar summary statistics, as shown in Table 2.2.*

and developed a technique that produces some graphics that look very different, see Figure 2.1, but share the same statistical properties, as listed in Table 2.2.

Property	Mean	Standard Deviation
Mean X	54.26570	0.00332
Mean Y	47.83510	0.00353
Standard Deviation X	16.76762	0.00283
Standard Deviation Y	26.93550	0.00402
Correlation	-0.06602	0.00305

Table 2.2: *Arithmetic mean and the standard deviation of the datasets shown in Figure 2.1. Although their appearance in the plots differs drastically, the summary statistics are very similar.*

Those examples make clear, that when exploring data, visualization must be an essential component of the workflow to be employed.

Besides the pitfalls of non-visualization driven data exploration, visualization has some advantages that are worth leveraging for knowledge generation tasks. First, data visualization is suited to deal with outliers visually and is therefore capable of pointing the user to data records that have some very different attribute values. Second, visual interfaces following the idea of visual analytics [56] effectively offload all methodological and algorithmic work to the model building process. In consequence, it enables users without methodical knowledge to still utilize corresponding visual methods. Third, visualization and in particular visual displays are the channel with the highest bandwidth from computer to humans and are therefore suitable for depicting large amounts of information [72]. Finally, data visualization relies on human perception and offloads cognitive load in the form of visual representations [53], when the visual mapping is appropriate. One huge advantage that can be exploited with a suitable visual mapping is the so-called *pre-attentive* processing capabilities of the human visual system. That capability leads to effects, e.g., *"reaction times that remain flat or even decrease slightly as the number of items increases"* [163], or as described by Julesz as immediate and effortless perception [187]. Even when the circumstances for pre-attentive perception are not fulfilled completely, visualizing data has many advantages. Card et al. argue, that visualization amplifies human cognition, and list arguments for that statement [159]. Among others, there are:

- Visualization reduces the search for information.
- Visualization enhances the recognition of patterns.

Those two arguments strengthen the decision to utilize information visualization to make the behavior of a machine learning observable and make it to a subject of exploration, hypothesis generation, and finally knowledge generation. It reduces the amount of time to search for a piece of information and fosters the recognition of patterns, which are the starting points for findings made in information visualization [56]. In a similar direction, Munzner argues that visualization is suitable when human capabilities must be augmented, and not replaced with decisions made by computational processes [53]. This is a powerful, strong motivation, as machine learning is of course by definition an automated process, to exploit the possibilities of interactive information visualization to make the behavior of a fully automated system observable. To judge the system and to generate knowledge out of the observable behavior is still — and will be in the foreseeable future — a task requiring humans.

3

Feature-level Error Analysis for Supervised Learning

In this chapter, we demonstrate our contribution in the area of feature-level error analysis. The work has been conducted with *application-level engineers*, namely computational linguists that utilize machine learning techniques to solve their application problem, which is, in this case, a binary classification of documents. The problem we contribute a solution for was to identify features that are likely to cause errors, i.e., that have specific ranges of their domains that are often present when the classification is wrong. Our contribution in this chapter is an interactive visual analytics tool, *Minerva*, that allows analysts to inspect the behavior of a document classification system, in the light of misclassifications. Together with the domain experts, we identified a workflow for error analysis in text classification, and provide the implementation integrated into *Minerva*.

3.1 Challenges and Tasks

With the increased availability and rapid growth of textual data, analyzing text data has gained tremendous popularity. The vast variety of applications includes, but is not limited to, sentiment analysis, essay grading, user profiling, automated feedback processing, or the partitioning of a given document collection into various topics. The foundation of these applications is machine learning that is based on feature vectors extracted from the text data. These vectors are composed out of a number different features, for example, statistics that measure text properties like average length of a sentence, or lexicon features which determine the occurrence or share of lexicon words in a given text document. Most applications utilizing textual data can be implemented using freely available libraries like Stanford CoreNLP [51]. Those libraries do not require a high level of expertise in natural language processing, work reasonably well for most applications, and do not impose detailed knowledge of the actual feature set. However, having an application that analyses text data, the analysis on feature-level can be very informative to understand common errors or flaws in the outcome of the machine learning methods, because besides word occurrences and statistical properties semantics play a role too. For example, “*enjoy*” is usually of very positive polarity, although a negation (*didn’t*) can turn it to be negative: “*Alice* **`didn’t`** `enjoy` *riding Bobs new bike*”.

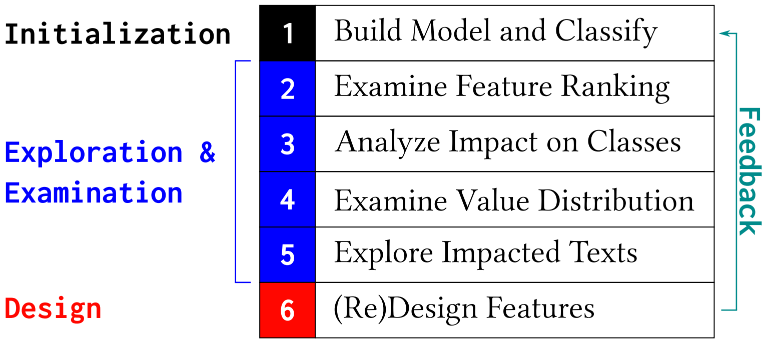

Initialization
1 Build Model and Classify
Exploration &
Examination
2 Examine Feature Ranking
3 Analyze Impact on Classes
4 Examine Value Distribution
5 Explore Impacted Texts
Design
6 (Re)Design Features
Feedback

Figure 3.1: *The Text Classification Analysis Process TeCAP. It is comprised of three phases, the* Initialization Phase *(black), the* Exploration and Examination Phase *(blue), and the* Design Phase *(red). To account for insights gained during the process, a feedback loop originates from Step 6 to Step 1 (indicated on the right) allows to repeat the process, e.g., with a modified feature set.*

Adding heuristics to recognize this or similar cases is useful only to a limited extent because heuristics cannot include all possible variations of negations, as they are a linguistic phenomenon which is volatile for various reasons. This issue also holds for a variety of other problems in natural language processing, for example, the detection and proper handling of irony or sarcasm. The dynamics and semantics of natural language are one of the primary reasons why working with text data is challenging. To cope with these different challenges, we propose that analysts, or more specifically *application-level engineers*, visually inspect the feature set to get an idea of the cause of errors or unexpected outcomes that is visible on the feature-level, given that the technology used is working as expected. The formalization of this process that has been developed in collaboration with experts from the domain of computational linguistics is a six-stage procedure which we call *Text Classification Analysis Process* (TeCAP), see Figure 3.1, which has been developed in close collaboration with practitioners in the field of machine learning and natural language processing.

TeCAP contains three phases, consisting of six stages:

i) **Initialization Phase** (black): the machine learning task is executed, and the results are modeled (stage one).

ii) **Exploration and Examination Phase** (blue): the exploration of machine learning results on the feature-level, observation and validation of findings (stages two to five).

iii) **Design Phase** (red): insights from the previous phase can be used to change the feature set (stage six).

Note, that the stages in the Exploration and Examination and Design Phase do not need to be executed subsequently. After the Initialization Phase, analysts are free to choose which visualization they use, although the level of detail on each stage varies from a very high level (importance of features) down to the actual feature-level (occurrences in the text). To account for insights into the application problem, each of the stages can be skipped to reach the feedback loop from stage six to stage one. We implemented this strategy and support for two of the three stages in a prototype called *Minerva*, which employs visualization and visual analytics techniques to support the exploration and examination stages in particular. As postulated in Section 1.1.1, this work is agnostic of the actual machine learning back-end used to classify text documents. Therefore, support for the third phase, namely the redesign of the features, is only possible to a limited extent: Minerva supports a simplified what-if analysis based on linear combinations of features that allow an interactive what-if analysis for cases where it is enough to modify the feature set with similar combinations of already existing features.

In this chapter, we claim the following contributions: i) The structuring of a feature-based machine learning exploration technique *TeCAP*. ii) The prototypical implementation of TeCAP in a standalone application *Minerva*, that allows the exploration of text classification results on feature-level using visual analytics techniques. iii) We demonstrate the applicability and usefulness of Minerva on a real-world problem in an application example.

3.2 Related Work

Although our work is not focusing on feature selection and visual applications of feature selection, in particular, this work has foundations in that discipline. Guyon and Eliseff [141] introduce different ranking and selection techniques, which are considered as state of the art. An early work bringing together visualization and feature selection in an interactive manner has been published by Guo [140]. Based on the selection of subspaces in a high dimensional data space, interactive visualizations are provided to enable analysts to explore the data space. Noteworthy is the integration of steerable techniques to support the data exploration like orderings, groupings, and the control of aggregation methods. May et al. present a visualization technique designed for feature subset selection called *SmartStripes* [87]. The authors tightly integrate feature selection algorithms and visualization that allows the user to refine and steer the automatic feature selection. Krause et al. [48] present a system based on similar principles, but in contrast to *SmartStripes* it is designed to support predictive modeling in a specific use case. To do so, a specific glyph design and ranked layouts of them are applied.

Application wise, Mayfield and Penstein-Rosé are closely related to our work [95]. They report on an interactive application designed to support error analysis in text classification tasks based on a matrix display of the confusion matrix. Heimerl et al. introduce a system which combines instance level visualization of the classification and a cluster view [73]. A cluster exploration system for linguistically motivated data is introduced by Lamprecht et al. in [64]. Seifert et al. propose a user-driven classification process by visualizing the classifier confidence and input documents [96]. Ankerst et al. visualize features but in contrast to this work concentrate solely on decision trees [157]. Van den Elzen and van Wijk present a similar application [83]. Seo and Shneiderman present a system implementing a rank-by-feature framework [129]. They use multiple visualizations such as matrices, histograms, and scatter plots to visualize the features and various statistics.

There already has been research in the field of reasoning concerning feature combinations and selections in machine learning tasks [165, 65]. This aspect of machine learning in the text application domain is the primary motivation for us to add the design phase to TeCAP.

The related work shows that there have been only a few works that provide the ability to analyze applied methods on the feature-level, which is in our understanding required to understand the outcome of text mining, because of the earlier mentioned inherent semantic dimension and dynamics of natural language text data. This is the research gap that we bridge with TeCAP and Minerva.

3.3 Interactive Error Visualization

The prototypical implementation of TeCAP is called Minerva. It supports the *Exploration and Examination Phase* with visualizations and includes facilities supporting the *Design Phase.* In the following, we outline the system and present our visualization designs for each of the stages in the *Exploration and Examination Phase.*

3.3.1 System Design

Minerva has five main components, which are: 1. Input (load feature vectors); 2. Classification Model Creation (input of classes and confusion matrix); 3. Data Processing (filter, order, combine, remove); 4. Visualization; and 5. Data Export (export feature vectors). Each component operates on separate input data, which allows the examination of different data sources at the same time, for example, to compare the outcome of two different feature sets extracted from the same dataset.

The system design abstracts from specific machine learning libraries or applications to allow the examination of different machine learning techniques or feature sets based on the same data. The system *reads the feature vectors* from ARFF files, as produced by WEKA [98] and similar libraries, CSV, and raw text files. The *classification model* allows Minerva to examine the machine learning algorithm outcome in detail. This includes the ability to judge whether a data instance has been misclassified and if a confusion matrix has been provided, whether a misclassified data instance belongs to false negatives or true negatives. The *processing* component provides utilities for the visualization (filtering, ordering), as well as standalone functionality to combine or remove features (design stage of TeCAP). The result is seamlessly integrated into the data model that any connected visualization or export component uses the ordered, filtered, or created features together with the imported ones.

Changes in the feature set can be stored in ARFF files that can be used as input for the popular machine learning library WEKA. The *export facility* allows filtered or combined features in the output, making it a suitable mechanism to re-run the machine learning task to inspect any differences afterward.

3.3.2 Visualization Designs

Stage 1: Build Model and Classify. As argued in the introduction of this chapter, Minerva does not integrate any modeling or classification facilities. Instead, it treats the machine learning back-end essentially as a black box and utilizes the corresponding outputs.

Stage 2: Explore and Examine Feature Ranking. To give a quick impression of the importance of a feature in a potentially huge feature set — in our experiments we used feature sets with more than 5,000 features) — we provide a word cloud of the top *n* most important features. Limiting the numbers and also the possibility of restricting or excluding feature labels ensures that the analyst has access to tools that reduce the number of displayed features to specific features at hand, while at the same time provides capabilities of a quick overview of the whole feature set. Feature importance is double encoded in the label size and color. The double encoding makes sure that even if the label of an important feature is shorter than others the analyst is still able to perceive the feature as important, and the label is not getting lost in the word cloud. The importance of features is computed according to state of the art measures such as information gain, symmetrical uncertainty, or the chi-square test statistic.

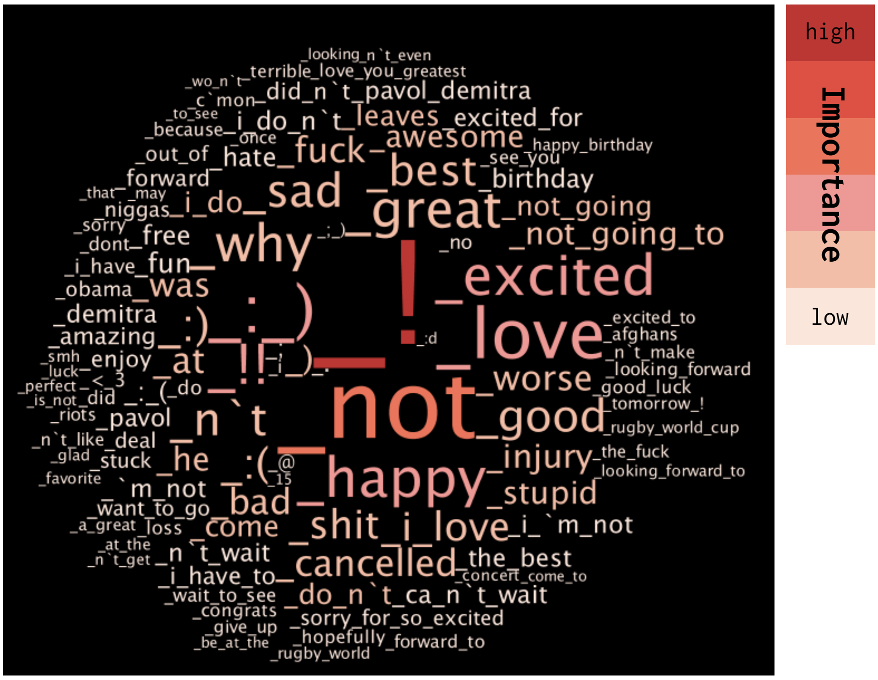


Figure 3.2: *Word cloud displaying the top 100 features according to their importance computed with the chi-square test statistic. The importance of a feature is double encoded in its color and the size of the feature label.*

Analysts can adjust the number of visible features, as well as filter features or feature families for exclusion or inclusion in the visualization. The word cloud layout is based on Rolled-out Wordles from Strobelt et al. [82], as it can be seen in Figure 3.2, which is known to generate compact layouts suitable for interactive systems.

Stage 3: Analyze Impact on Classes. At this stage, analysts can examine which features have predictive power for which class. For each feature, a glyph is displayed, which is built out of four equal sized segments of a circle (see Figure 3.3). The design is inspired by Guyon and Eliseff [141], as they show that features with distinct properties, like a distinct value distribution per instance set, can be used to form more predictive ones.

For each data instance, we compute the average value of true positives and true negatives to false negatives and false positives respectively. Each of the two segments showing the difference to the average false negatives/false positives is mapped with the same color.

The visual design exhibits four distinct error patterns which give a good idea of why a classification error has occurred (see Figure 3.4). The patterns are: 1. Features where the average value of false negative instances is closer to true negatives than true positives. They are likely to cause false negatives. 2. Features where the average value of false positive instances is closer to true positives than to true negatives. They are likely to cause false positives. 3. Features where both, 1 and 2 are

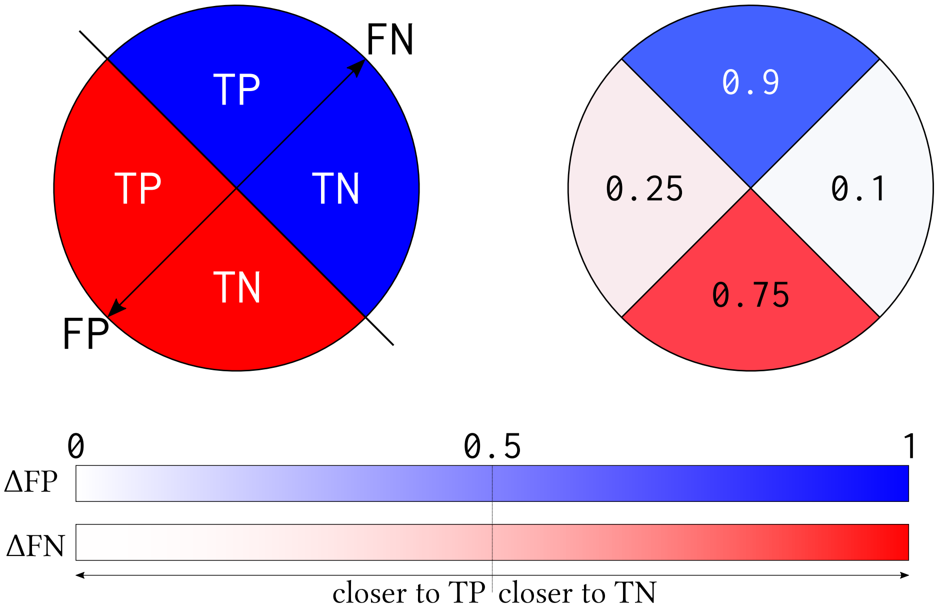


Figure 3.3: *Class impact visualization. Top left shows the segmentation of the glyph and the referenced values. Read:* TP *and* FN *as* $\Delta(\overline{TP} - \overline{FN})$ *(top segment). On the top right, an example of the given values is shown. The utilized colormaps are shown at the bottom.*

the case. Those features are a potential cause of both, false negatives and false positives. 4. Features where both, 1 and 2 are not the case. This indicates a good predictive power of the feature on misclassified instances.

Besides the visible ordering based on information gain, we also implemented a glyph ranking based on its visual properties, for example, error pattern affinity to make it easy to spot groups of similarly behaving features. A visualization of a whole feature family (part of speech tags and part of speech tag patterns) can be seen in Figure 3.5. To enable the comparison of different features, normalization relative to the minimal and maximal average true positive and average true negative value is applied for each glyph separately. The resulting values of misclassified instances can be inspected relative to correct classified instances.

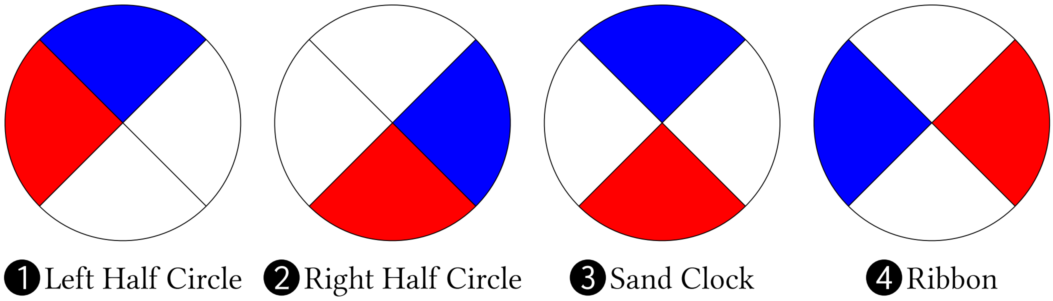


Figure 3.4: *The four error patterns. Red sectors indicate the difference to the false negatives, cyan sectors the difference to false positives. White colored segments do not contribute to the described patterns and are therefore left blank.*

The visualization makes two simplifying assumptions to make sure the visual design reflects the desired properties of a feature. i) The distribution of correct and incorrect classified instances has roughly the same shape. ii) The peak of the distribution of misclassified instances lies between the distribution peaks of correctly classified instances. The validity of these two assumptions can be verified with more detailed visualizations provided by Minerva.

Stage 4: Examine Value Distribution. At this stage, analysts can examine the distribution of feature values and also get information about the size of the overlap of values in different classes. To show the value distribution, we combine two classes from the classification in a histogram display, as it is illustrated in Figure 3.6, which enables comparison of the value distribution of two classes, for example, false negatives and false positives. The height and background color of a histogram bar reflect the class with the most instances in the corresponding bin, the class with the smaller number of instances is indicated by the color and height of the inner T.

If necessary, the analyst can enable an additional coloring of the remaining background space of a histogram bin, which indicates the total number of instances in a bin with a colormap from dark gray (fewest) to white (most). This indicator explicitly states the number of instances in the bin and allows the bin-wise comparison of multiple instances not only for a single feature but also the complete feature set. See Figure 3.10 for an illustration of that feature.

Stage 5: Explore Impacted Texts. The different visualizations presented in this section enable the analyst to develop new hypotheses about errors and their feature-wise origin and to select interesting documents for error analysis. Together with the feature set, the classified text documents are the ultimate tool to confirm or falsify the hypothesis of an error source, because nothing can illustrate the outcome of a feature extractor better than the actual data source.

To allow analysts to verify or falsify their hypothesis with concerning features and the corresponding texts, Minerva implements a document view which is augmented by the extracted features. The view allows the visualization of feature families like n-grams, negations, modal verbs, or word endings, based on the imported feature set. Furthermore, custom lexicons can be added if required by the analyst. Selected feature families are highlighted directly in the text by coloring the text span corresponding to the features (see Figure 3.7).

3.3.3 Interactive Visualizations

Minerva provides a generic framework for interactive visualizations, which is utilized by each of the different views. It is based on an infinite canvas and provides zooming and panning capabilities

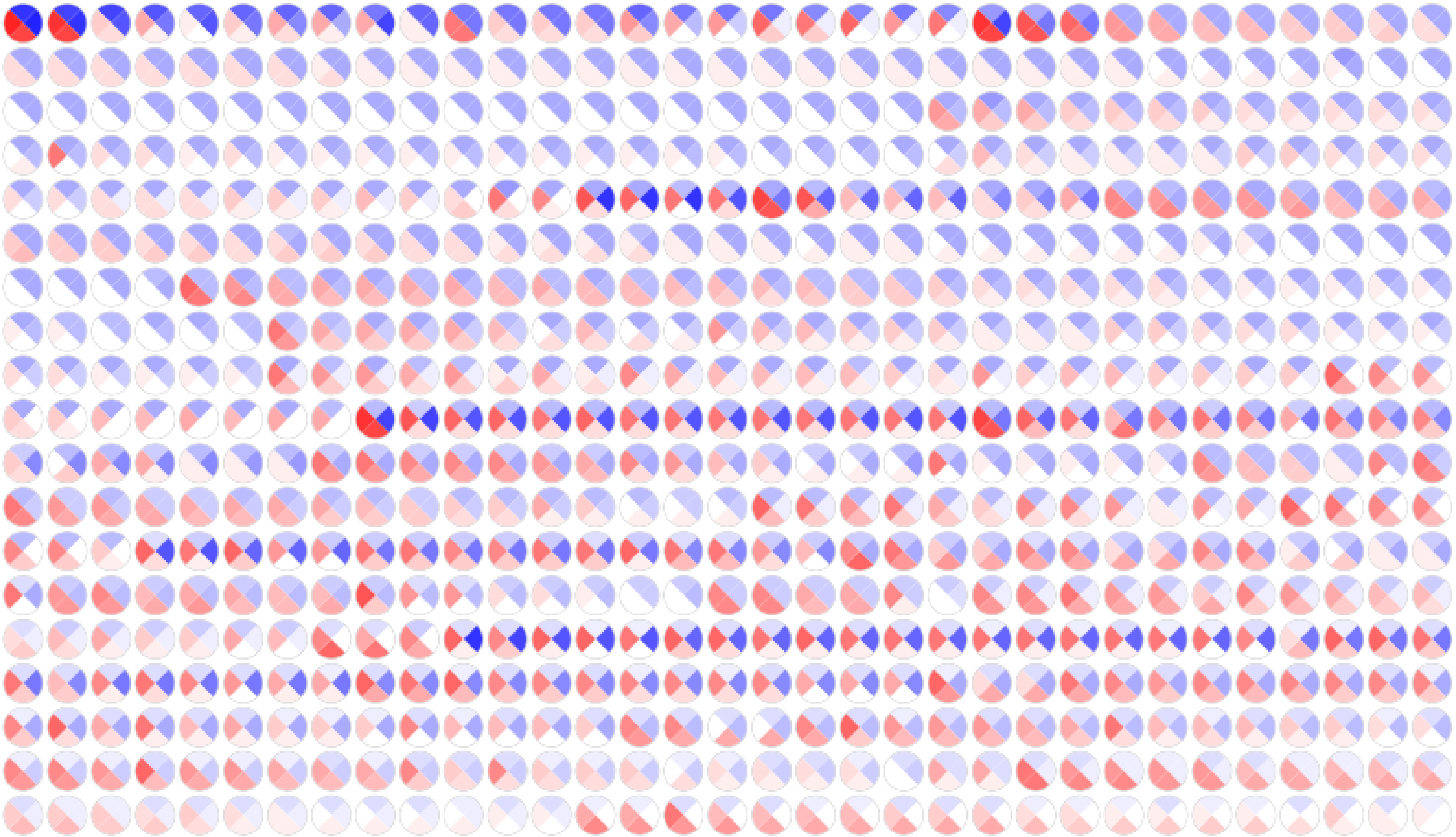


Figure 3.5: *Visualization of relative feature value differences between the four instance classes. "Half-circle" glyphs indicate features which are likely to cause false positives or false negatives. The order of glyphs is determined by the displayed visual patterns and reflects their importance according to the information gain measure.*

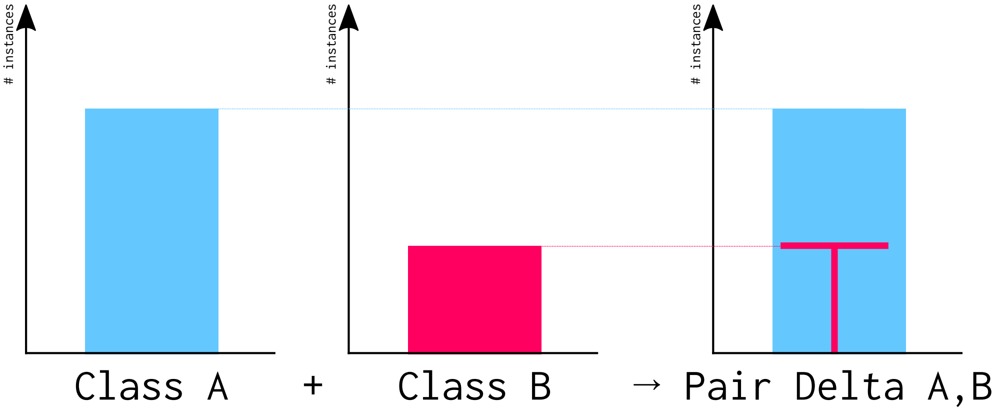
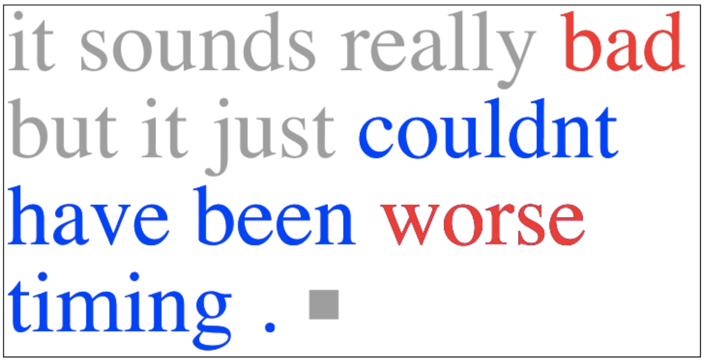


Figure 3.6: *Pair Delta Visualization Construction. The histogram on the right is created by overlaying the two histograms on the left. The class with the most instances in a bin is represented by the bar color on the right; the smaller class is indicated by the color of the inner* T.



it sounds really bad but it just couldnt have been worse timing . ■

Figure 3.7: *Document Viewer. A tweet with highlighted polar words (red: negative polarity) and negation spans (blue). In this example, the polarity score was computed to be* 0*, although the message is of negative nature.*

to facilitate the visual information-seeking mantra by Shneiderman: *Overview first, zoom and filter, then details-on-demand* [169].

To foster the combination of different views for an effective exploration and examination of the data, Minerva provides a linking and brushing functionality [177]. Each visualization implements brushing mechanism suitable to the visual mapping of the features and propagates selection and de-selection of features to an abstract, data-based selection subsystem, which in turn notifies the remainder of the opened views of changes about the selected features.

Besides the linking and brushing functionality, Minerva provides a view synchronization facility. View synchronization is realized by describing the current viewport of a visualization regarding the displayed features. The abstraction from the graphical contents of a view makes it possible to synchronize the viewports of different kind of views. View synchronization can be enabled and disabled by the analyst, which makes Minerva suitable for explorative analysis as well as hypothesis building and verification tasks.

The combination of these three functionalities — linking, brushing, and view synchronization — allow analysts to switch visualizations and walk through TeCAP while maintaining focus at the currently selected feature set.

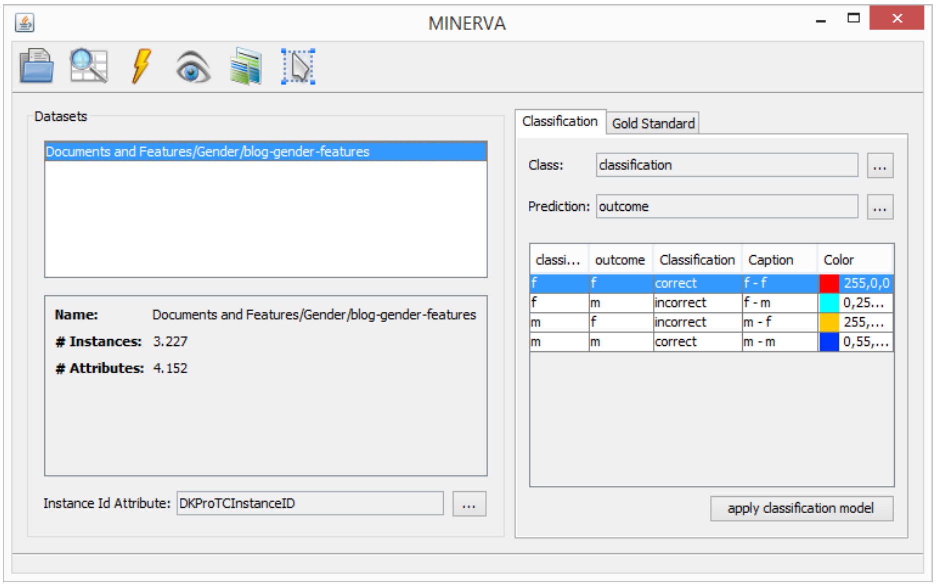
3.4 Evaluation

In this section, we show how Minerva can be used to gain knowledge about machine learning tasks working with text data. We demonstrate a popular sentiment polarity detection task, using a publicly available dataset with Twitter data from the 8th International Workshop on Semantic Evaluations (SemEval 2014 Task 9, Subtask B). Our goal is to demonstrate that achieving better performance is possible also through better understanding — enabled by TeCAP — of the textual features rather than standard machine learning customization.

Stage 1: Initialization: Build Model and Classify

Our goal is to determine whether a given tweet is of positive, neutral, or negative sentiment. We use WEKAs SVM-SMO classifier with the information feature selection filter. Feature extraction is done by the UIMA-based [131] open source DKPro framework [118]. The feature set is based on successful applications from the literature. It contains a number of word- and character-level n-grams [171], text surface properties such as interpunction [149] or smileys [93], sentiment lexicons [76, 67, 154, 81], and syntactic measures of individual part of speech tag ratios and groups (n-grams on part of speech level).

Minerva abstracts from the actual machine learning task to not depend on a single machine learning library and keep the applicability of our methods as general as possible. As a consequence, it is



Documents and Features/Gender/blog-gender-features

Name: Documents and Features/Gender/blog-gender-features
# Instances: 3.227
# Attributes: 4.152

Instance Id Attribute: DKProTCInstanceID

Class: classification

Prediction: outcome

classi...	outcome	Classification	Caption	Color
f	f	correct	f - f	🟥 255,0,0
f	m	incorrect	f - m	🟦 0,25...
m	f	incorrect	m - f	🟧 255,...
m	m	correct	m - m	🟦 0,55,...

Figure 3.8: *Minerva's main window. On the left, the current dataset some meta information is presented. On the right, the setup of the in-place classification model is shown.*

required that after loading the data an in-place classification model are configured by the analyst, as it can be seen in Figure 3.8 on the right. This makes sure that the classification outcome and details about the classes are available in the application, even though the actual classification process runs outside Minerva.

Stage 2: Exploration and Examination: Examine Feature Ranking In practice, one of the most interesting questions concerning a feature-based machine learning task and the feature set is: *What are the most useful features?* In Figure 3.2, the top 100 n-grams from the positive and negative sentiment classification output can be seen. The importance of smileys, swear words, and interpunction is clearly visible, which is an indicator for designing and adding new features in that areas of the feature set.

Furthermore, not only n-grams but any other feature subgroups can be examined with this visualization. Our feature set contains LIWC lexicons [154], which are helpful to separate neutral tweets from emotional ones (LIWC is an analytical framework frequently used by psychologists). Figure 3.9 illustrates the importance of LIWC features in the classification task. Besides the expected influence of positive and negative emotion words, `Affect` and `Anger`, the frequency of personal pronouns `Ppron` (I, them, her) and verbs `Verbs` (walk, go, see) play an important role. It is also interesting to see, that the frequencies of assents (*Agree*, *OK*, *yes*, ...) and negations (*no*, *not*, *never*) are also important.

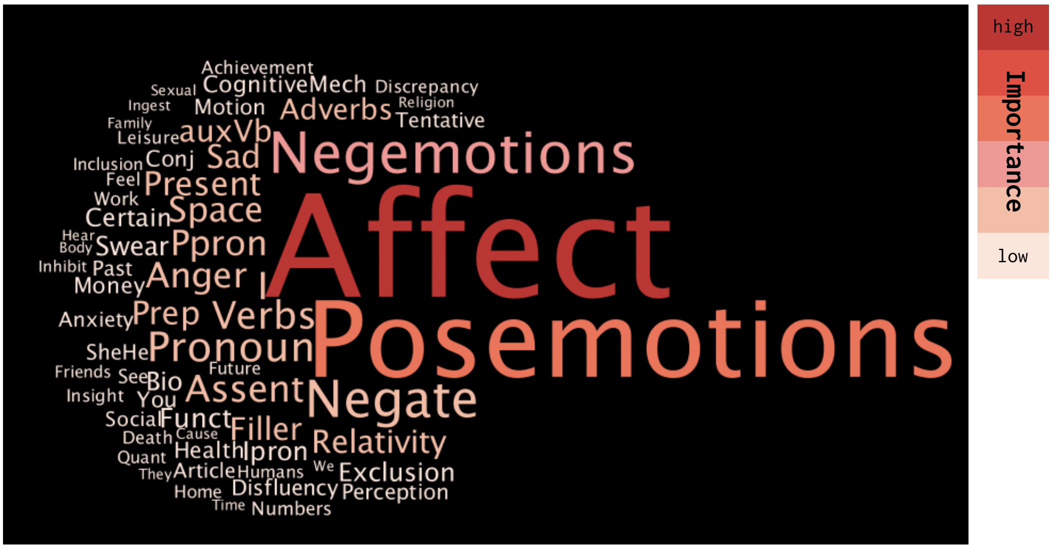


Figure 3.9: Word cloud of LIWC features and their importance in the sentiment distinction task. As expected, positive and negative emotion words have a significant influence on the result and are therefore important. However, it is also clearly visible, that the frequency of personal pronouns `Ppron` *and verbs plays an important role.*

Stage 3: Exploration and Examination: Analyze Impact on Classes   The previous stage gives an initial understanding of which features matter in the sentiment polarity problem. Having now identified the important features, it is of interest to see which features have predictive power for which class.

Figure 3.5 shows features from Steinberger's polarity lexicon [81]. Each glyph represents a word from the polarity lexicon. Features corresponding to the left-half circle pattern, as introduced in Figure 3.4, are part of correct classification outcomes, if the represented word is present in a tweet, without causing false positives. If not, they lead to opposite conclusions in some cases, which results in false negatives. An opposite situation appears for the 5th feature in the top line, the *Ratio of verbs in tweet*. It suggests that in our test set a certain, i.e., low, rate of verbs predicts well a neutral tweet, while the other (high) verb rate cannot, on average, distinguish a polar tweet from a neutral one. Similarly, the positive-negative tweet problem can be analyzed. We observe that the right-half circle features are represented by n-grams such as *shit* or word groups such as *Disgust*, while left-half circles are lexicon words such as *Joy* or *n-grams* such as *looking forward*. Combining the left-half circle and right-half circle features (e.g., Joy-Disgust) in the preprocessing can lead to improved results, and at the same time eliminates the need for the demonstrated in-domain knowledge.

As just shown, the visualization is a useful instrument to refactor existing lexicons or create new ones, especially in tasks where the relation between the class and the words from the lexicon are not as clear as they are in the presented application.

Stage 4: Exploration and Examination: Examine Value Distribution With the help of the previously shown visualization, we were able to observe that numerous sentiment lexicons suffered from the same issue of predicting a polar tweet to be neutral when no lexicon word was found. What is missing are insights into the actual distribution of feature values and also information about the size of the overlap in different classes. In the sentiment classification task, the *Pair Delta Visualization* can be used to examine the problem that sentiment lexicon features perform poorly when predicting a polar tweet to be neutral when the lexicon word is not part of the tweet to classify.

By the left and middle column of Figure 3.10, it becomes apparent, that even the combination of lexical features cannot lead to an improved classification performance of tweets with lexicon polarity value around zero. The highlighting in the background indicates that close to zero values of the polarity are coming from other lexicon features as well. A possible explanation would be, that people indicate emotions without using sentiment words. However, for syntactic features (right column in Figure 3.10) the feature values are well distributed across value intervals, which makes a separation into two classes possible. Hence, combining syntactic features with the ones based on lexicon words could lead to a classification improvement.

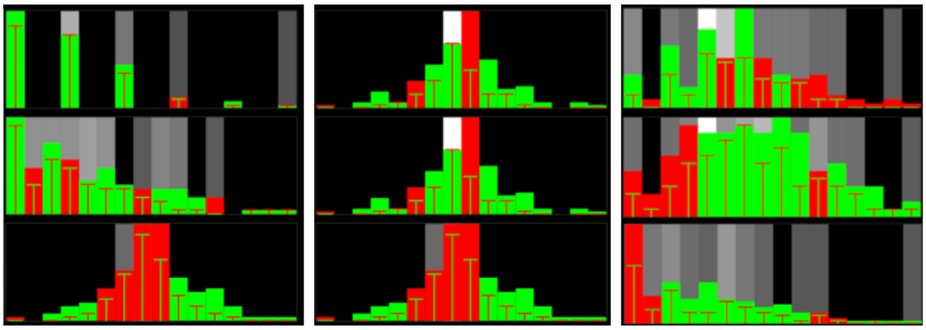


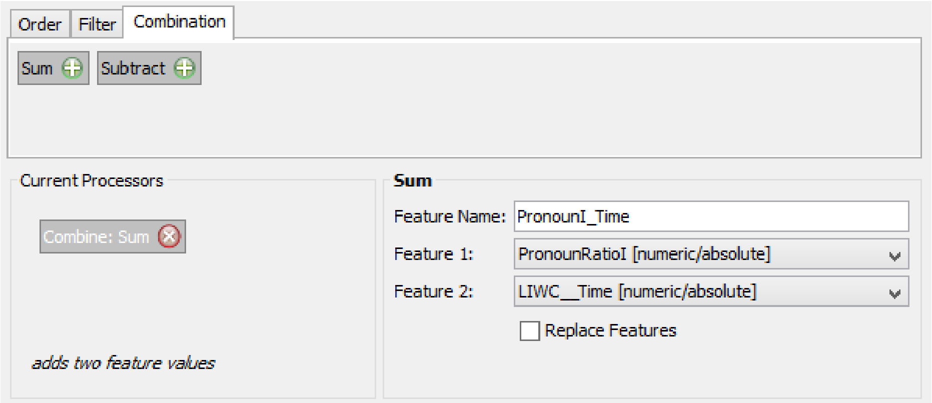
Figure 3.10: *Value distribution of six sentiment lexicon features (left and middle column). While these lexical features share similar error overlap, impacted instances are distributed more evenly over syntactic feature values, as they can be seen in the right column.*

Stage 5: Exploration and Examination: Explore Impacted Texts The exploration of impacted text can be used to see if *the feature computations have been well defined.* In particular, during this stage documents can be explored to find common errors in misclassified documents, which could indicate that the feature measures a different phenomenon than intended.

In Figure 3.7, the resulting highlighting of polar word negation spans is illustrated. Using this visualization, we saw that while for certain words inverting the polarity score in negation was sensible (*"It doesn't sound bad"*, *"I wouldn't say it's great"*), for many cases it was counterproductive. The tweet shown in Figure 3.7 has a neutral polarity, because *bad* counts as `-1`, and *couldn't* + *worse* as `-(-1)`, which results in an overall polarity of `0`. Using this view, we also found that skip-n-grams were not suitable features as they were ignoring random occurring negation words in between. Other errors come from ambivalent words such as loose (control vs. weight), or from ironic or sarcastic messages: *“now that I can finally sleep... can’t wait to work for another 8 hours or so tomorrow... yay...”*.

Stage 6: Design/(Re)Design Features The last stage of TeCAP can be understood as the implementation from insights gained in the exploration and examination phase. In the spirit of Guyon [141], we allow feature combinations to be created directly in Minerva by providing an interface to create linear combinations of existing features.

Using our built-in feature design facilities shown in Figure 3.11, we first combined positive and negative n-grams into features which behave as a sentiment lexicon. Additionally, we created combinations of all lexicon-based features with syntactic features, especially verbs, pronouns, and adjective indicators.



Sum

Feature Name: PronounI_Time

Feature 1: PronounRatioI [numeric/absolute]

Feature 2: LIWC__Time [numeric/absolute]

□ Replace Features

Figure 3.11: *The user interface to create feature combinations. In this example, the combination of a semantic (time indicating words, LIWC) and a syntactic (pronoun ratio) feature is shown.*

Lessons Learned and Results Based on the insights gained from the shown exploration and examination part of the Feature Engineering and Error Analysis Cycle, we adjusted the classification process as follows: i) We added an additional sentiment lexicon-based on positive and negative n-grams to enhance the existing polarity lexicons. ii) We combined lexicon-based semantic and syntactic features, especially for verbs, pronouns, and adjective indicators. iii) The ArkTweet POS Tagger [84] has been complemented with the finer grained Stanford POS Tagger [145] to enhance the overall POS tagging accuracy. iv) The negation scoring was modified so that *“Can’t be better”* is treated as positive and *“Can’t be good”* as negative.

Besides the described run of the Feature Engineering and Analysis Cycle, further applications of the cycle and implementation of the suggested changes lead to an improvement of the macro-average

F-Score from 56.2 to 64.1, which is a difference of 7.9. This result would lead to a final ranking in Semeval 2014 in the Top 20 of 50 participants, compared to the 38th rank we would reach with the initial setup without applying our analysis method (see `http://alt.qcri.org/semeval2014/task9/`).

3.5 Summary

Semantic properties are a common cause for problems which can lead to a degraded performance when applying machine learning techniques in natural language-based application. Starting at this general problem, we developed the strategy *TeCAP* that conveys these problems to the human, who has a much broader knowledge of semantics and understands the analyzed text data. The strategy, developed in close collaboration with practitioners from the field of machine learning and natural language processing, gives the process of knowledge generation in text mining a structured way that can be followed easily and answers the most pressing questions when it comes to feature-based analysis of machine learning outcomes.

For example, using the presented Feature Cloud users can quickly get an impression of the feature ranking. This information can be used to match users expectations of the actual machine learning tasks by confirming or falsifying previous knowledge of the data analyst.

The visualization for the impact analysis of features on the classes (Figure 3.5) is designed for specific error patterns. We abstract more from the actual feature vector data, but we still allow single feature analysis. The presented visual design distributes the feature glyphs in a grid, with a customizable number of rows and columns. To extend the visual design to allow also the perception of clusters of features, an improved version of the visualization layout based on the perception of the glyphs is an open research challenge. The current matrix layout, based on a lexicographic order of the feature labels, gives insight in the error-wise behavior of the different feature families contained in the feature set, as they typically share portions of their labels. Currently, this order makes it possible to spot part of speech tags and n-gram features that are behaving differently, as they are shown next to each other. When users want to focus more on the different groups — clusters — behaving similarly regarding caused errors, a force directed layout will make more sense than the current matrix view. To configure the forces, the four error patterns, see Figure 3.4, should be fixed in the edges of a square or rectangle, and place the single glyphs according to the attraction to these patterns. The resulting view should be able to effectively communicate with the different groups in the feature set concerning the error patterns. Having such a layout in place, new challenges arise, for example how to reduce the inevitable glyph overplotting. Since the feature ranking is crucial for domain experts, it will be of interest to develop or apply existing techniques to include the ranking in the resulting visualization.

We also see potential for improvement in the text visualization, as it is shown in Figure 3.7. Currently, we use colored text spans, and lines above and under text spans to indicate where a feature is located in the text. This is complemented with a tooltip containing the names of features if they overlap at parts of text spans. Next steps will be the examination of text highlighting methods, for example, background shadows or different font styles to be able to display more than three features at once and also visualize the spans where they overlap.

Another aspect that is interesting to continue working on is the integration of the different loops of visual analytics applications as proposed by the model of Sacha et al. [56]. There is enormous potential for a *good* visual analytics-based application concerning the model, because the overall goal of our technique is gaining knowledge using insights from the feature-level. We already have strong support for the exploration loop, and verification is possible because we provide different views on the same data. Including knowledge generation support is a huge challenge, but will be a significant achievement, not only for data analysts in the field of natural language processing. In the current version of Minerva and the application domain of natural language processing, we rely heavily on world knowledge, which is hard to externalize and even harder to grasp via automatic processes.

Technically, the implementation of Minerva and also most of the visualization designs are designed for binary classification problems. This leads to clarity of the analysis questions in each stage of TeCAP, but at the same time makes Minerva tedious to use for multiclass classification problems. Future steps to overcome this limitation include an extension of the underlying data model, as well as bigger design space to demonstrate the generalization of TeCAP and Minerva.

4

Interactive Ambiguity Resolution of Ambiguous Feature Sets

In this section, we present an interactive visualization technique that combines different kinds of information to simplify a feature set generated by machine learning, more precisely in a supervised learning scenario to identify named entities. A number of software packages exist that can be utilized in real-world applications that achieve acceptable performance. Although, as it is the case when working with unstructured natural language data, achieving a good performance while solving an application problem still poses challenges. In the case of named entity detection, it is the fact that ambiguous references to the same entities are part of the analyzed documents, e.g., caused by grammar or the writing style of the author. In this chapter, we contribute an interactive visualization that facilitates reasoning about ambiguous features in the feature set. We enable *application-level engineers* to explore the results of state of the art named entity recognition packages and modify the feature set when they observe any ambiguities. The visual interface utilizes matrix visualizations that are known to scale for a large number of rows and columns, which are based on two different kinds of information. First, co-occurrence, which is the pairwise count of entities that occur together, e.g., in a sentence, which is statistical information. Second, this information is complemented with what we call *character embedding*, a variant of the widely used word embeddings, that is tailored to characters, as we are dealing with fictional literature. We showcase the generalizability of this approach as we demonstrate and argue about different machine learning back-ends that can be used to detect named entities, as outlined in Section 1.1.1.

4.1 Challenges and Tasks

Named entity recognition (NER) techniques are used to identify passages of text that are likely to refer to entities and label them with categories such as *person*, *place*, *company* and other categories [121].

The current state of the art techniques are based on manually annotated text corpora, which are automatically analyzed and transferred in a corresponding model of language use, grammar, and other properties of the annotated text. These models can be used to find and classify entities in previously unknown documents. State of the art NER techniques are provided with off the shelf models created from vast amounts of annotated documents, such as news corpora or similar document collections. For many use-cases, the standard models perform well and are well-suited for the integration in different applications, such as the analysis of customer reviews [77], or the automated extraction of locations or characters in literature [33]. Whenever these techniques and models are applied to documents that contain unusual or underrepresented linguistic characteristics compared to the training data, the performance of NER systems degrades, which in consequences makes errors more likely to happen. This is specifically the case, whenever the language use, grammar, or vocabulary, depending on the properties have been learned and stored in the model, i.e., when analyzing documents from different authors. To cope with that problem, state of the art NER packages provide facilities to create custom models from a collection of documents. Given that for the application at hand enough data is available, text annotation requires domain experts to annotate the data in sufficient amount and quality, which is a time-consuming and challenging process [79].

Charlie Weasley	George Weasley	
Fred Weasley	Percy Weasley	Ron Weasley
Ronald Weasley	Ginny Weasley	Weasleys
Harry Potter	H. Potter	Potter
The Potters	James Potter	

Table 4.1: *A selection of entities classified as* person *that share at least one token. Extracted from* Harry Potter and the Sorcerer's Stone *by J. K. Rowling with state of the art NER software.*

In this chapter, we introduce an interactive, visualization-based approach to improve the performance of NER software concerning ambiguities, e.g., the multiple occurrences of the same entity caused by different attributions and references, as illustrated in Table 4.1. With the help of our technique, *AmbiguityMatrix*, such ambiguities can be identified through model visualization and interactions so that further annotation of the data and the corresponding time-consuming training phase is not required. The visualization makes use of two interlinked triangular matrix plots representing the results of a specific NER model/technique as the row/column appearance, as they can be seen in Figure 4.1. The color-coded cells in the upper triangular display the pairwise co-occurrence of entities, while the cells in the lower triangular matrix show the semantic similarity of the entity surrounding based on word embeddings [138, 103]. Both views guide the analyst to possible entity ambiguities, indicated by co-occurrences (upper half) or semantic similarity of the character surroundings (lower half). Entities with high co-occurrence and/or high semantic similarity — that can be assumed to be related to ambiguities of the NER — stand out by a dark color in both matrix triangles. Less obvious, potential ambiguities follow the visual pattern of a single, outstanding cell in one of the two views. By merging two rows or two columns, the user can derive disambiguation rules, e.g., the merger of two entities, which can be used to improve NER models in a post-processing stage, or by feeding back the interaction and corresponding data to a model-refinement or recreation phase. We focus on referential ambiguities as their detection and resolution requires a high level of text understanding, which is currently not feasible for natural language processing techniques. In contrast, our approach relies on text understanding and world-knowledge of the user and can be applied without requiring machine-readable, externalized knowledge, for example in the form of knowledge bases. Existing automated approaches utilize external knowledge bases, but naturally, they are limited in the amount of contained information, as no data source can provide a reasonably complete depiction on world-knowledge, which is required to solve this class of ambiguities. Also, the creation and integration of corresponding ontologies or knowledge-bases impose huge effort that not every project relying on NER software can afford. For illustration purposes we focus on fictional literature to account for a great variety of writing styles, language use, and grammar that is typically not contained in training data utilized to create off the shelf NER models. We concentrate on characters — named entities of type *person* — in fictional literature, which is subject to real-world applications as well as current research [49, 44] in the NER area.

In this chapter, we claim the following contributions: i) A visualization-based approach for linking of named entities based on referential ambiguity. ii) A novel combination of state of the art in visualization and natural language processing that assists human analysts in the entity linking process. iii) An interactive interface that combines content and context information to support interactive rule building from ambiguous data sources.

4.2 Related Work

This work resides in three different areas: named entity detection, entity or record linking, and entity relationship visualization. In the following, we give an overview of related work in those areas.

4.2.1 Named Entity Detection

Named entity detection is a part of the techniques collectively called NERC (Named Entity Recognition and Classification) [100], which summarizes techniques that identify entities and classify them into categories, e.g. *person*, *organization*, or *location*, in an unstructured source of text. Techniques for entity detection are manifold and utilize various methods. In the following, we outline the most important techniques and give some insights into their construction.

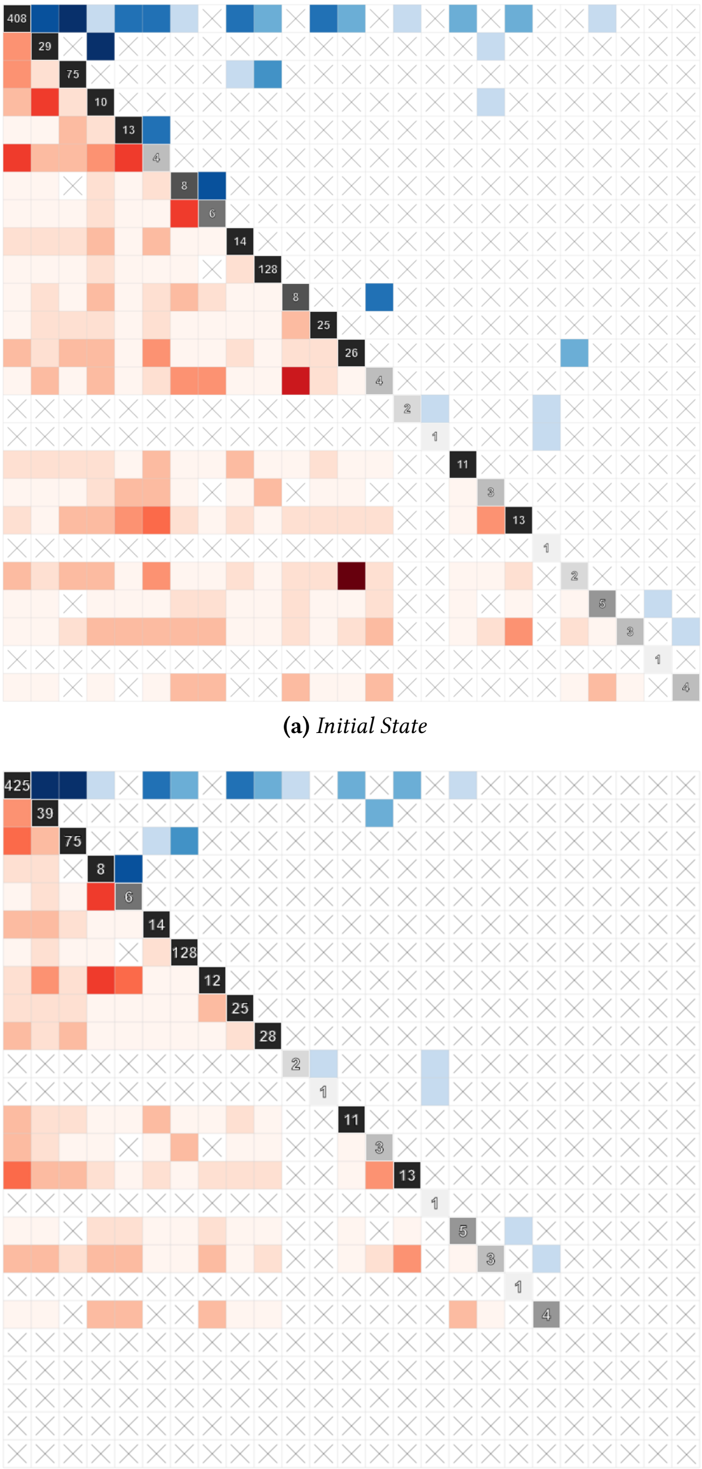
Lexicon- or dictionary-based systems come pre-equipped with lists of frequent names or locations, which are used for keyword searches, as well as further adaptions based on statistical techniques [160, 137, 124]. *Statistical* approaches utilize non-specific sources of text or word information, such as large document collections, to compute relationship statistics applied in the actual entity detection process [134, 135]. These techniques do not require a labeled training set and are independent of the document language, as they purely rely on statistical properties of the documents. Recently, *machine-learning* techniques have gained popularity in the field [162, 156, 105]. They perform well, can be utilized even without a learning process by distributing the appropriate models, and are highly adaptability to the processed data. Naturally, a drawback of these methods is the dependence on the model that has been trained from labeled documents. A third family of techniques is based on *handcrafted rules or heuristics* that can incorporate and employ different properties of the analyzed text documents [178, 158, 104]. These systems have advantages if there is a consistent structure among the documents to analyze, that can, for example, be induced by the domain of the document contents or specific writing styles, as these can be defined precisely and modeled in the rule set. Therefore, rule-based NER software is expected to reach high precision, while the recall depends on the variety and flexibility of the rule-set. Last, a number of techniques combine different aspects from the aforementioned named entity detection techniques to benefit from strengths of the different techniques and methodologies [155, 54]. A comprehensive overview of different methodologies and approaches can be found in [121, 111].

4.2.2 Entity/Record Linking

The task of merging different entities into a single one is also known as entity linking, which is nowadays part of many natural language processing and NER systems.

Bunescu and Pasca presented a technique that uses support vector machines to learn rankings from categories and names using data from Wikipedia [124]. Similarly, Mihalcea and Csomai exploit the Wikipedia hyperlink structure to associate documents with Wikipedia articles, which can also be used to process and link named entities [120]. Other data sources, namely DBpedia and YAGO, are used by Hoffart et al. to compute the disambiguation of entities, which could also be used to link entities [85]. Large-scale knowledge-bases are exploited by Lin et al., where the authors propose corpus level features, e.g., the similarity of word contexts, as well as textual matches in the knowledge base to merge entities together [75]. Trani et al. present an approach that is based on manual, collaborative entity linking, and is therefore strongly related to our work [58]. Moro et al. provide a method that is based on graphs, computes entity linking, and go into detail about the possible technical solution to find *similar* entities [52].

All these methods induce considerable extra effort, either when it comes to creating a knowledge-base from Wikipedia, DBpedia, or other sources, as well as the computational effort which is



(a) *Initial State*

(b) *Final State*

Figure 4.1: *AmbiguityMatrix at the example of George Orwell's 1984, generated by OpenNLP Name Finder. The initial sparse matrix got even sparser; most changes are visible in the lower left half depicting the semantic similarity measure. For visual clarity, we removed the entity labels.*

problematic for interactive approaches. While our proposed technique could utilize information from knowledge-bases, it does not require any computationally expensive processing or data mining steps, as we rely on human knowledge and understanding of the ambiguity resolution and linking task, respectively.

4.2.3 Entity Relationship Visualization

A number of related work is addressing the problem of visualizing relational data, such as in named entities relationships. In [133], Ghoniem et al. compare the two main approaches, node-link diagrams and (adjacency) matrix visualizations, for their readability. The empirical study found that matrix visualizations are particularly suitable in cases where the associated graph is dense. However, like the problem of finding a useful 2D layout for nodes in a node-link diagram, matrix-based representations inevitably require an appropriate row-/column ordering for representing the dataset structure effectively. Recently, Behrisch et al. presented in [25] a survey with guidelines on selecting the most appropriate matrix reordering algorithm. However, the survey states that reordering methods for asymmetric matrices are rare and often lead to unsatisfactory results. In cases, such as the ones presented in this paper, human-assisted reordering should be applied to improve the quality of matrix visualizations [25, p. 712].

Similar to our proposed visualization, matrix layouts have been applied in various context for showing entity relationships. For example, in MatrixExplorer, Henry and Fekete show a synchronized node-link and matrix exploration interface for depicting entity-relationship structure in social networks [127]. A hybrid node-link/matrix visualization, called NodeTrix, was proposed by Henry et al. for representing the co-author relationships in large research communities[119].

Whenever the relationship characteristics are not univariate, but potentially even conflicting relationship weights exist, such as in our guiding use case of entity disambiguation and model comparison, node-link diagrams will have the drawback to invest multiple edges for each relationship. On the other hand, matrix-based visualization with a complex glyph cell representations have been introduced by Im et al. in [63] and shown, e.g., by Beck and Diehl in [92] for the comparative analysis of different dependency relationships in software systems or by Behrisch et al. in [61] for the comparison of rankings and orderings. Recently, an empirical evaluation of the effectiveness of juxtapositions for triangular matrices has shown that distributions and patterns in large feature spaces can be explored with juxtapositions of asymmetric triangular matrices, showing a different feature/metric on the upper and lower triangular matrix part [37].

In the domain of text analysis, entity relationship found specifically appeal. Shen et al. give a dashboard-like view in their tool NameClarifier that allows the simultaneous exploration of different aspects of authors that are subject of disambiguation [21]. They combine single entity sequences, group-based, and list views to support the disambiguation task. In the tool Jigsaw, Stasko et al. present multiple linked views that deal with entities extracted from documents [114]. Connections between entities are symbolized with lines between multiple lists. To visualize the temporal evolution of entities, Mazeika et al. utilize stacked areas [88]. Oelke et al. present a technique that, based on so-called *fingerprint matrices*, depicts the co-occurrences of entities over the course of literature [69]. Gold et al. utilize circular visualizations where sectors of the circles represent specific topics or speakers [34]. Their visualization technique supports animation over the temporal progression of the visualized data per entity and shows the corresponding relations popping up over time. Similar, the work of El-Assady et al. utilizes animated display of topics or speakers over time to illustrate their evolution in conversations [26].

For most of the show related work, we see shortcomings regarding the scalability of the proposed visualization technique. Typical solutions are animations over time, which keeps the number of entities to display simultaneously low. Additionally, techniques that present entities side by side suffer from scalability issues, as the length of the documents, in connection with the chosen aggregation level, dictates the final size of the visualization that could exceed the screen space. Therefore, neither animation nor fingerprint-like visualizations will serve the purpose of display entity information of a whole piece of literature as it is required for ambiguity resolution of them.

4.3 Interactive Ambiguity Visualization

To support the ambiguity resolution of named entities, it is required to give the analyst a way of assessing the relatedness of two given named entities. In our approach, we express relatedness by two measures derived from the analyzed literature: character-level statistics and content-based information. Each of those is represented by a proxy metric, as explained in the next paragraphs.

Character Statistics. We compute the character statistics via so-called pairwise co-occurrences of entities [126]. The reason to compute this particular measure is grounded by the language use in fictional literature. Entities, in particular characters, are commonly referred differently in consecutive sentences, as it is considered to be a good writing style. See for example the excerpt from a Harry Potter novel given in Figure 4.2. There, one entity, Harry Potter, is referenced three times, and each of the references uses different wording. Co-occurrence metrics are the method to capture those variations in writing style. By utilizing this concept, we can express relationships between entities based on their co-occurrence in the form of *Entity A* is co-occurring *n-times* with *Entity B*, and capture variations in the language with respect to entities.

Content Information. So far, the occurrences of characters with each other can be quantified and covers, therefore the statistical view on entities. To be able to communicate the content-based context of entities, we compute the word embedding for each entity, that contains its linguistic context [138, 103], and therefore approximates the near content-wise context of an entity. To relate the word vectors to entity similarity, the distance of these vectors can be computed by utilizing the Szymkiewicz-Simpson coefficient [194]. Using the resulting similarity score, we can compare entities not only based on their co-occurrence, but also on the similarity, or dissimilarity of their context in terms of actual words.

The computation of both metrics is part of Phase Two of the interactive ambiguity resolution process described in the following section.

4.3.1 Interactive Ambiguity Resolution Process

We structured the interactive ambiguity resolution in a process that consists of four phases. It covers the data generation, model creation, interactive visualization, as well as possible feedback to other processes that make the interactive ambiguity resolution process suitable to embed in more complex text processing workflows or act as a stand-alone tool for exploring NER data.

Phase One: Named Entity Recognition. In this phase, named entity recognition software is applied to the documents to process. The result of this phase are the entities and their classes so that the next phases can utilize this information. This phase is executed automatically and does not require any interaction or assistance by users, besides the initial NER technique and the corresponding model selection. We utilize two state of the art and widely used representatives of today's machine learning-based systems; namely, the Stanford Named Entity Recognizer [128] and OpenNLP Name Finder[1]. Due to the lack of rule-based NERs, and to be able to cover all NER families given in the related work (Section 4.2), we implemented our own rule-based NER, as described in Section 4.3.2. At this stage, it is important to note that no assumptions concerning the technical foundations of named entity recognition as well as the class of entities that are subject to interactive ambiguity resolution are made, according to Section 1.1.1. Although, as argued before, in this work we concentrate on entities of type person, as we analyze fictional literature that typically relates heavily to characters.

Phase Two: Linguistic Data Preparation. This phase has two purposes: first, it computes the pairwise co-occurrence of entities [126]. We utilize the concept of co-occurrence, as it can be observed that in literature references to the same character in consecutive sentences are expressed

[1] Website: `https://opennlp.apache.org/`

differently (see the two examples at the beginning of Section 4.3.4). Second, the word-embedding [138, 103] per entity is created. Word-embeddings are commonly understood as collections of words that share the same context. We adapted the concept and computed the embedding per entity, which yields a vector of words containing the context of the given entity. Afterward, the pairwise distance per word-vector is computed, which can be interpreted as semantic similarity, because word-embeddings are known to preserve the semantic relationships [66]. We assume, that a high semantic similarity is a hint for similar entities, and therefore points to potential ambiguities. Similar to Phase Two, this phase is fully automated and does not require interaction. The linguistic data creation is also designed to be as generic as possible, as co-occurrence computation makes no assumptions of the processed text data. Similarly, the creation of the word-embeddings and the resulting semantic similarity measure is possible for all kinds of words, as it poses no requirements for the context it should be computed for.

Phase Three: Interactive Visualization. The third phase contains the depiction of the previous analysis results While the two preceding phases are automatized completely, this phase is where humans and automated data analysis come together, and interactions with the created models from the data analysis are possible. This is demonstrated in the following parts of this chapter, where we show examples of our approach executed with three different named entity recognition software packages. To visualize the data, we use the metaphor of an entity-to-entity matrix, where each cell represents a pair of entities. The metaphor directly maps to the nature of the data extracted in the two preceding phases, as they are referring to a single entity (Phase One), as well as entity to entity (Phase Two). Details of the visualization and supported interaction are given in Section 4.3.4 and Section 4.3.5, respectively.

Phase Four: Feedback. In this work, the primary goal is to improve the NER results, which is prepared in this stage. To do so, interactions such as the merger of two entities are captured (Phase Three) and translated (this phase) into a format that is understood by the NER employed in Phase One. As this depends on the actual NER component in place, the outcome of this phase can be of different nature. For example, off the shelf modules such as Stanford NER will be accompanied most likely with post-processing based on the user interactions, which is configured and transferred to the corresponding component during this phase. This requires a corresponding post-processing component, as well as a specified way of adding new rules to it. For rule-based NERs, such as our prototypical implementation, it is possible to formulate new rules and inject them right into the rule set.

For real-world applications, support for the described process can be implemented in various ways. Although, it is clear that the implementation of the visual and interactive processes are not suitable for batch processing of documents, as human interaction is required. Instead, we expect that the outcome of the interactive ambiguity resolution process, which are either postprocessing rules or new additions to rule-based NERs, will be integrated into the data processing pipelines. We also see this process as a tool for interactive model comparison of either different NER methods, or models that differ in parameters necessary for their training or document selection. The actual configuration depends heavily on the employed NER technology and the use case at hand. Therefore, we see the process elaborated in this section as ideal, and abstracted from use-cases or technical parameters.

4.3.2 Rule-based Named Entity Detection

In most application areas, rule-based named entity recognition has been superseded by machine-learning based approaches [121]. Although, there is a significant advantage of rule-based approaches: because of the specificity of the typically manually created rule-set, the expected precision is higher than the precision of competing techniques.

Rule	**Example**

Action Words	`said Harry`
	`address Hermione`

Possessives	`Harry's world`
	`Tom's mind`

Salutations	`Mr. Longbottom`
	`Mr. and Ms. Longbottom`

Titles	`Professor Dumbledore`
	`Lord Voldemort`

Verbs, 3rd person, singular	`Dudley walks towards Harry`

Table 4.2: The rule-set of our rule-based NER. **Bold** denotes the main anchors of the rules, red text the extracted entity. The rules are completed with dictionaries that provide action words, salutations, and titles, as well as a set of character-based post-processing rules.

A disadvantage of the necessity of a fixed rule-set is the fact that similar to pre-trained machine-learning models; the input has to conform to the data that has been used to formulate the rules. If this is not the case, rule-sets are expected to fail similarly than model-based techniques, but these errors are easier to counter with extensions of the rule-set. Besides, maintaining a general set of rules is tedious and time-consuming. Therefore, it makes sense to target a specific type of text, for example, fictional literature, with rule-based NERs. A significant advantage of rule-based named entity recognition is the computation time. It is noticeably lower than the processing time of state of the art, model-based NER software. For interactive applications, such as ours, this is a huge plus. Because of these two advantages, expected high precision as well as the suitability for interactive systems, we implemented a custom, rule-based NER software tailored explicitly to fictional literature.

Table 4.2 lists the core of the rule-based NER. The rule-set leverages surface properties of the processed text data, which eliminates the need for time and resource intensive parsing of the documents. Additional information, such as dictionaries with titles, salutations, or action words are generated from DBpedia [116]. Further post-processing is applied to the results so that overlapping matches are included only once, or matches of more than one word are matched as a single entity.

4.3.3 Automated Analysis Performance

To get an impression of the performance of the three named entity recognition packages (Stanford Named Entity Recognizer, OpenNLP Name Finder, custom rule-based NER), we present the typically used performance metrics recall and precision and a false positive rate in Table 4.5. The exact versions and utilized models are given in Table 4.3. We selected a number of books from our literature collection, so that fictional literature of different length, different authors, and different genres is represented. For practicality, we limited the selection to titles where the creation of a gold standard for the automatic evaluation is possible, for example by leveraging various online fan-wikis or Wikipedia.

Software	Version	Notes
Stanford NER	3.6.0	
Stanford NER Models	3.6.0	7 Class, Distsim
OpenNLP Name Finder	1.5.3	
OpenNLP Models	1.5	en-ner-person
Rule-based NER	1.0	
Rule-set	1.0	see Table 4.2

Table 4.3: *The utilized named entity recognition software and the corresponding models.*

Table 4.4 lists the selected books that we show in our examples. To be able to judge the quality of the out of the box performances, we apply common metrics to evaluate named entity recognition software, which are precision and recall. The required gold standard for automated evaluation has been created manually from data sources such as fan-Wikis or Wikipedia. As this is no accepted or widely used data, precision and recall in our studies should not be interpreted absolutely, as the gold standard data contains some side characters, animals, or other entities that play only minor roles. For example, the gold standard for the Harry Potter novel contains *Snowy* and *Tibbles*, which both are cats. These errors were not corrected because we aimed at a broad data foundation of the

ID	Author and Title	Genre	Year
AC	*Agatha Christie*: Death Comes as the End	Mystery	1944
GO	*George Orwell*: 1984	Dystopian	1949
JR	*J. K. Rowling*: Harry Potter and the Sorcerer's Stone	Fantasy	1997
SK	*Stephen King*: Doctor Sleep	Horror	2013

Table 4.4: *The collection of literature used for our tests.*

gold standard. Therefore, the numbers should only serve for relative comparison as they are used in this work, not as absolute performance metrics.

Table 4.5 contains precision, recall and the false positive rate for the chosen literature and the three considered named entity recognition packages. The book from Agatha Christie (AC) seems to be well reflected in the Stanford model, as well as the rule-set of our custom NER. Both technologies reach a recall of 1.00, and compared to the results of analyzing the other books a quite high precision, although the rule-based NER outperforms the other candidates with a precision of 0.87. OpenNLP excels in the analysis of George Orwell's *1984* (GO), the other two NERs have huge problems with this book, in particular, the Stanford NER, that reaches precision and recall values very low compared with the competitors. The results of the analysis of the Harry Potter novel (JR) is similar and almost comparable to all three techniques, while the Stanford NER shows a very good recall compared to OpenNLP and the rule-based NER. For the book of Stephen King (SK), we see that all participants have problems, and perform comparably concerning the precision, although the recall of OpenNLP and the Stanford NER is better than the recall of the rule-based NER approach. Generally, we see that the false-positive rate of all candidates is quite high, in three of the four displayed cases even over 75 percent, while all techniques have strengths and weaknesses for each of the candidates.

4.3.4 Entity and Data Visualization

Consider the following sentence: "*Mr. Dursley wondered whether he dared tell her he'd heard the name "Potter"*" [164, Chapter One]. While it is a natural assumption, that references to the name *Potter* in a book about *Harry Potter* refer to *Harry Potter*, there is currently no way to actually resolve the reference given in the example sentence, which in the example given above is actually referring to the parents of *Harry Potter*.

In the example shown in Figure 4.2, the same entity, *Harry Potter*, is referenced three times in three sentences, which all use different wording. With the proposed visual design, we are targeting this exact type of referential ambiguity, which regarding data analytics, can be expressed as the co-occurrence of entities. Admittedly, this technique cannot capture all varieties of possible ambiguities

Book	NER-Type	Precision	Recall	FP
AC	Stanford	0.41	1.00	0.58
	OpenNLP	0.34	0.57	0.65
	Rules	0.87	1.00	0.12
GO	Stanford	0.01	0.04	0.98
	OpenNLP	0.70	0.80	0.78
	Rules	0.24	0.25	0.76
JR	Stanford	0.23	0.58	0.76
	OpenNLP	0.24	0.27	0.79
	Rules	0.22	0.26	0.77
SK	Stanford	0.09	0.75	0.90
	OpenNLP	0.11	0.51	0.88
	Rules	0.11	0.30	0.88

Table 4.5: *Precision, recall and the false positive rate of the three NERs. Details about the books can be found in Table 4.4.*

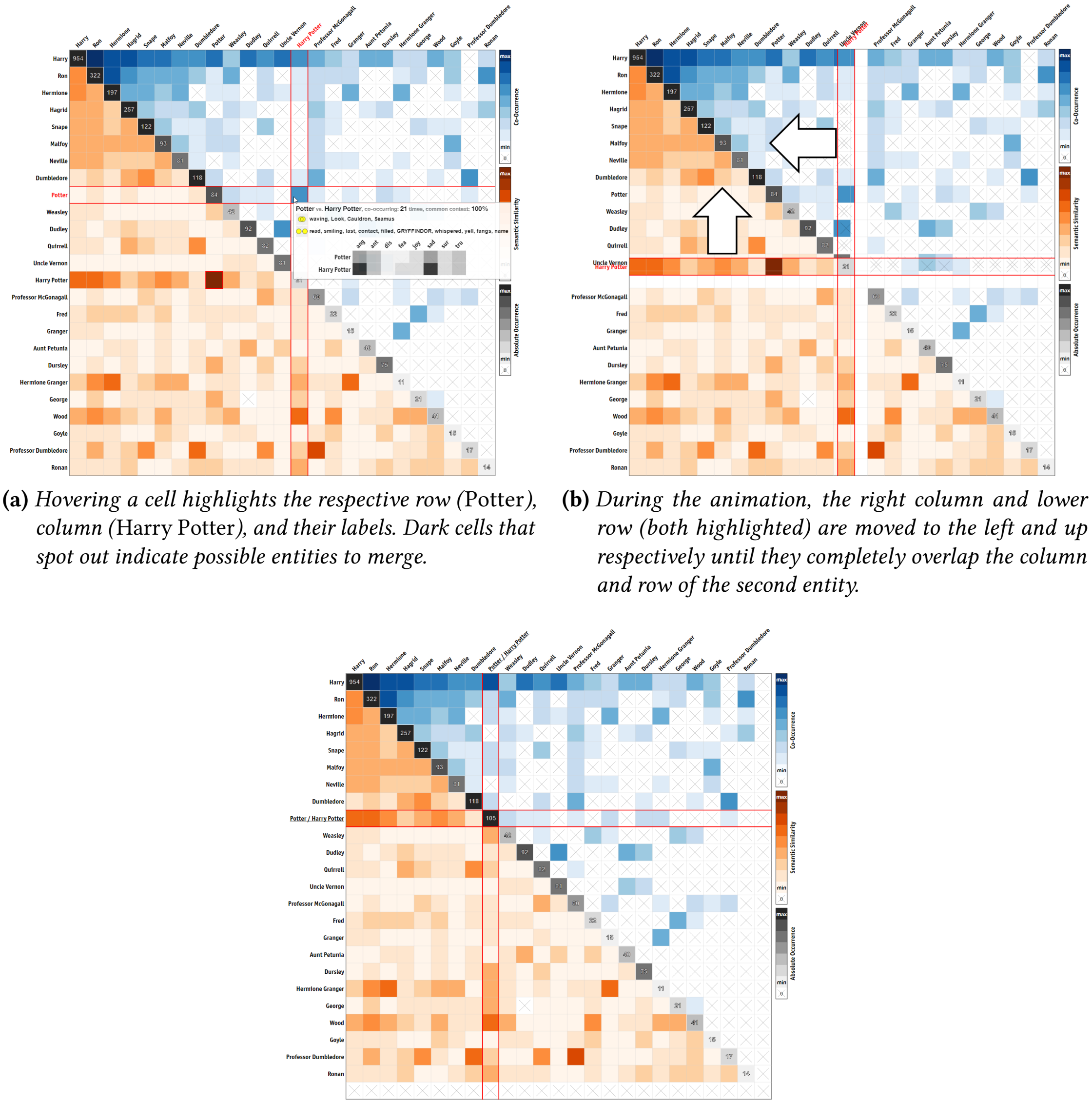
> “Bless my soul [...] **Harry Potter** ... what an honor.” He hurried out from behind the bar, rushed toward **Harry** and seized his hand, tears in his eyes. “Welcome back, **Mr. Potter**, welcome back.”

Figure 4.2: *Harry Potter is referenced in three different ways in three consecutive sentences. Excerpt from Harry Potter and the Sorcerer’s Stone by J. K. Rowling [164, Chapter Five].*

of entity references, as it is based purely on statistics and does not take the contents of the text into account.

To counter this issue, we contrast the co-occurrence information with an additional level of semantic information, the semantic similarity as described in Section 4.3, description of *Phase Two: Linguistic Data Preparation.* The combination of the semantic similarity measure and co-occurrence information adds the ability to cross-validate findings made either by the inspection of co-occurrences or the semantic similarity so that decisions made on statistics can be backed with semantics and vice versa. This allows findings such as *two entities co-occur together, but do have a different semantic context*, which is a strong pointer that both entities do not refer to the same person, and are not possible by providing only co-occurrence or similarity information.

It is clear that we need to visualize both types of information in a single view. For each pair of entities, we have two metrics available, namely the co-occurrence and the similarity of the word-embeddings. On data level, this corresponds to an undirected graph (as the scores are symmetric) of entities with relationships (nodes) that are quantified (weighted) with two numeric values. Visualizing this in a classic node-link diagram is prone to visual clutter, as we have to represent each data value with a separate edge. Instead, we visualize the node-link structure with adjacency matrices, which are known to represent graph structures in a space-efficient and compact manner. Each cell represents a pair of characters that are indicated by the corresponding rows and columns. In



(a) *Hovering a cell highlights the respective row (*Potter*), column (*Harry Potter*), and their labels. Dark cells that spot out indicate possible entities to merge.*

(b) *During the animation, the right column and lower row (both highlighted) are moved to the left and up respectively until they completely overlap the column and row of the second entity.*

(c) *The matrix after the merge process:* Potter *and* Harry Potter *are now merged and their co-occurrences and similarities are updated accordingly, the corresponding row and column are highlighted.*

Figure 4.3: *The process of merging entities in AmbiguityMatrix from left to right. On the left, the initial state of the visualization is shown. Users can explore the visualization, as well as the displayed data by tooltips. The middle image shows the process of merging, the prior selected row and column are moved on to the top left, as the arrows indicate. The right image depicts the state after the merge process. The visualizations are created with data from* Harry Potter and the Sorcerer's Stone *by J. K. Rowling.*

graphs, the neighborships between nodes are symmetric, which in turn leads to symmetric adjacency matrices, which means that in the visualization, two matrix cells are representing the same pair of entities. This property fits well to the data that we want to visualize, as we also have two metrics per entity pair. Recent studies in the area of matrix visualization showed that asymmetric matrices can be read as effectively as symmetric ones [37], so we do not lose interpretability. The following subsections elaborate on the visual design, as well as interaction possibilities of the matrices.

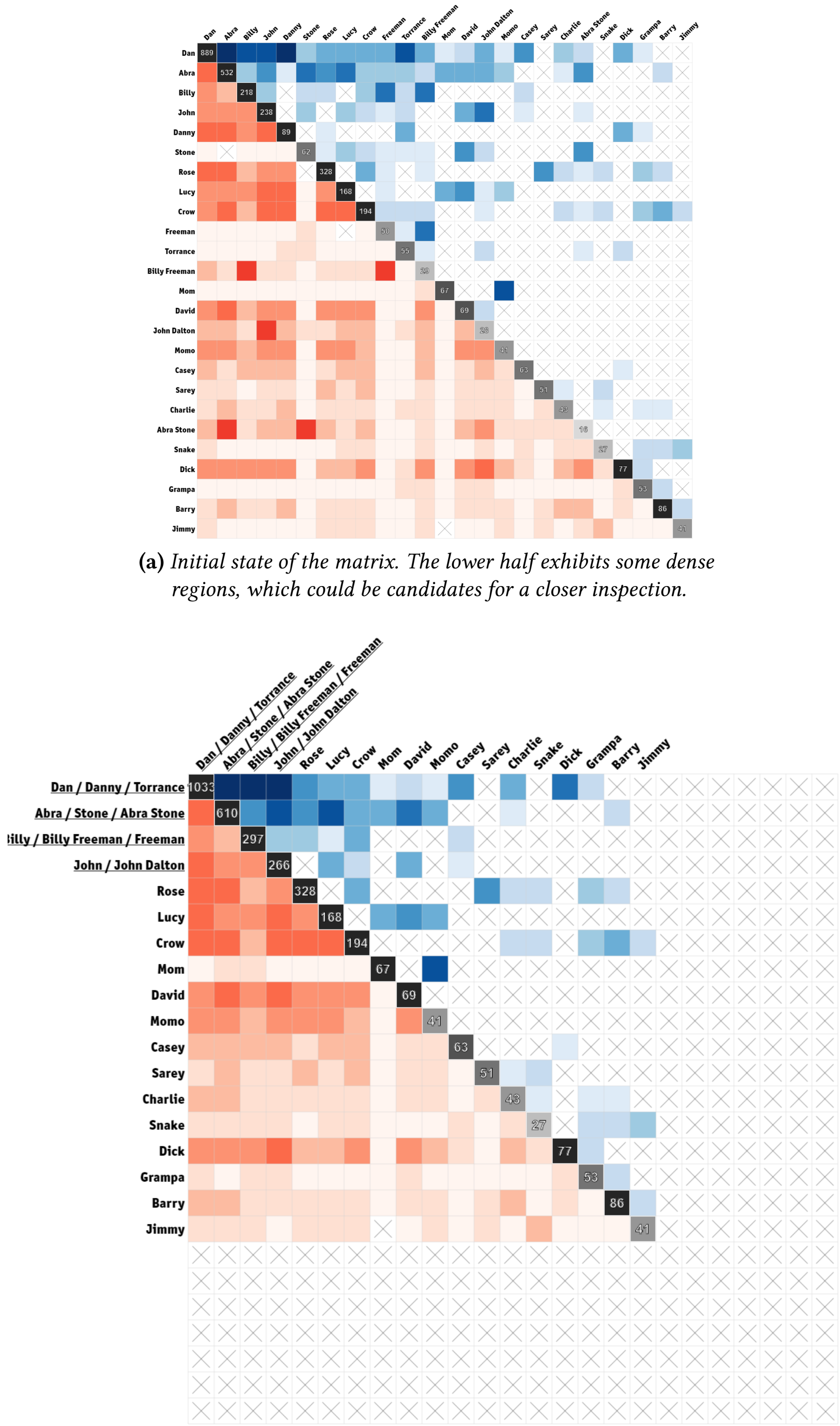
Ambiguity Matrix

Frequently co-occurring entities either suggest that individual entities have a high rate of interaction, or that the same entity is addressed in different ways. Visualizing co-occurrences in a matrix helps the user to conceive and to assess entity relationships based on their co-occurrence. The co-occurrences are mapped onto a color map in which dark and saturated blueish colors represent a high co-occurrence. In the case of no co-occurrence, the cell is white and marked with a gray cross.

To contrast the single viewpoint of entity co-occurrences, we split the matrix into the upper and lower triangular submatrix. The lower-left part of the matrix is used to depict a semantic similarity score based on the entity related word-embedding information, as elaborated in the paragraph *Phase 2 – Linguistic Data Preparation* of Section 4.3. A different colormap is used in the same fashion as for the co-occurrences to emphasize the different attributes, but with reddish colors.

On the diagonal, the overall occurrences of an entity are shown, see Figure 4.3. This meta-information is represented with a black-to-white colormap. For further clarity, the number of occurrences is additionally printed to the cell center.

Having a matrix-based visualization, the ordering of rows and columns plays an important role. In our work, we use an ordering according to the average co-occurrence of each entity. This order typically places the main entities of a book in the upper-left corner of the matrix as they interact with most of the other entity. Therefore, entities that are occurring less often move more to the lower-right part of the matrix. As the survey in [25] describes, matrix reordering approaches are mostly designed for symmetric matrix data. However, our visual design implies an asymmetric table ordering. Although correspondent analysis (CA) techniques could be used in this case, we decided to emphasize the semantic and domain-specific row/column ordering, that based on the absolute character occurrences reflects the *importance* of entities in the literature.



(a) *Initial state of the matrix. The lower half exhibits some dense regions, which could be candidates for a closer inspection.*

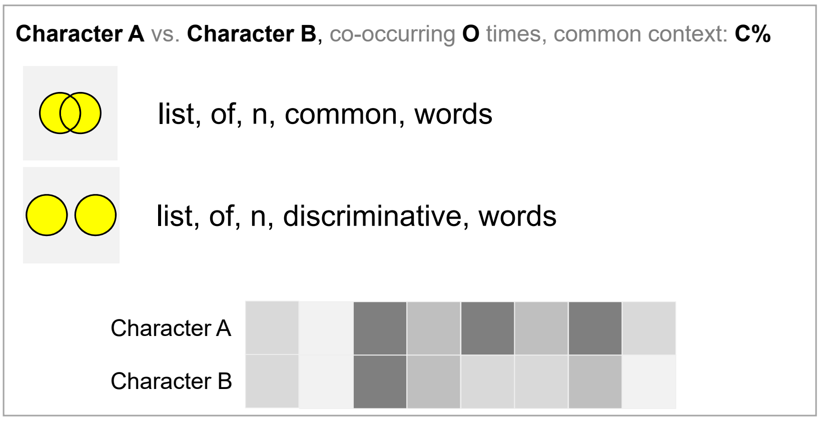
(b) *Final state after merging the entities together. The captions of the rows and columns indicate merged features with a slash /.*

Figure 4.4: *Initial and final state of AmbiguityMatrix at the example of Doctor Sleep by Stephen King, generated with the OpenNLP Name Finder.*

4.3.5 Interacting with Co-occurrences and Semantic Similarity

The user's task is to detect and assess the darker spots in the matrix yielding to a high co-occurrence or a high similarity. The user can now decide to merge the two entities based on the provided information and her background/domain knowledge. Clicking on a cell triggers the merge process for two entities, which is displayed with animation to help the user to understand the effects in the matrix. Firstly, the row and column belonging to the entity further down and to the right in the matrix are moved towards the entity which is placed more on top and left until they completely overlap. The moving row and column are highlighted during this animation to support the user to track them (see Figure 4.3b). Afterward, the entity labels at the edges of the matrix are merged, e.g., *Potter / Harry Potter.* Eventually, the merge affects the co-occurrences as well as the similarities to all the other entities. The co-occurrences of the two merge candidates to every other entity are summed up to provide the new co-occurrence value. A recomputation of the similarity score taking into account the merger of two entities is too slow to be performed in an appropriate time for the interactive system. Therefore, the maximum similarity score of the merged entities to every other entity adopted, see Figure 4.3c.

After the entities have been sorted, we do not apply the matrix ordering, as we would undoubtedly damage the navigational context created during the inspection of the matrix by changing the order and therefore also the visual appearance. Instead, we move the merged rows and columns to the top and left, which is a heuristic to keep the sorting mantra intact as all co-occurrences and similarities of the merged character must be equal or higher than the values of the two individual entities. To help the user in navigating in the matrix, the current row, column, and their labels are highlighted in the upper and lower part of the matrix whenever the mouse is moved over them, as depicted in Figure 4.3a. The corresponding cell on the opposite side of the diagonal is


Character A vs. **Character B**, co-occurring **O** times, common context: **C%**

list, of, n, common, words

list, of, n, discriminative, words

Character A

Character B

Figure 4.5: *Illustration of the character cell tooltip. On top, basic information such as the co-occurrence and the overall similarity of contexts is shown. Below, two lists of the ten most important common (top) and distinct (bottom) words are shown. At the bottom, the rectangles indicate Plutchik's basic emotions [185] of each character to intensity, where low means light, and dark high intensity.*

highlighted correspondingly, providing an immediate way to compare co-occurrence and similarity values.

To communicate additional context information concerning the characters, a tooltip showing additional information about the entities of a hovered cell is displayed, as shown in Figure 4.5. The tooltip is divided into three parts. On top, the names of the two characters the hovered cell refers to are shown, together with the number of co-occurrences, as well as the similarity score of the character-wise word-embeddings. The second area contains further information about the actual context words, namely the words that are similar and dissimilar from both characters, ranked according to their word-embedding score. This provides good insight in the typical "word-wise" surrounding of a character and allows direct insight into the similarities and differences for the current entity pair. Last, the area at the bottom of Figure 4.5 illustrates the Plutchik's basic emotions [185] per character, extracted with the emotion lexicons by Mohammad and Turney [67]. We map the pairwise emotions, which are anger, anticipation, disgust, fear, joy, sadness, surprise and trust, to colored squares. The color is assigned according to the relative intensity, from low (light gray) to high (dark gray). This display communicates the emotional profile of a character, and allows a direct comparison between both, helping the user to assess the similarity of their emotional traits.

4.4 Evaluation

In this section, we showcase the initial state of AmbiguityMatrix, and the final state after the recognized ambiguities have been resolved. The examples have been conducted by ourselves to keep the results consistent, as we argue that the whole process depends on world-knowledge which differs from user to user, which would make the results almost impossible to discuss.

Agatha Christie: Death Comes as the End As it is visible in Table 4.5, two of the three NER packages provide already good results concerning precision, recall, and false positives. It is in line with this insight that we did not find more than two ambiguous references to entities using the Stanford NER data, as well as the data from the rule-based named entity recognition technique.

George Orwell: 1984 In Figure 4.1, the initial (left) and final (right) state of the visualization based on data generated with the OpenNLP Name Finder is shown.

In general, and backed by the data given in Table 4.5, this book seems to be hard to analyze for the Stanford NER, as well as the rule-based NER approach. After loading the data from these two techniques, we saw that no ambiguities were visible for the most frequently occurring 25

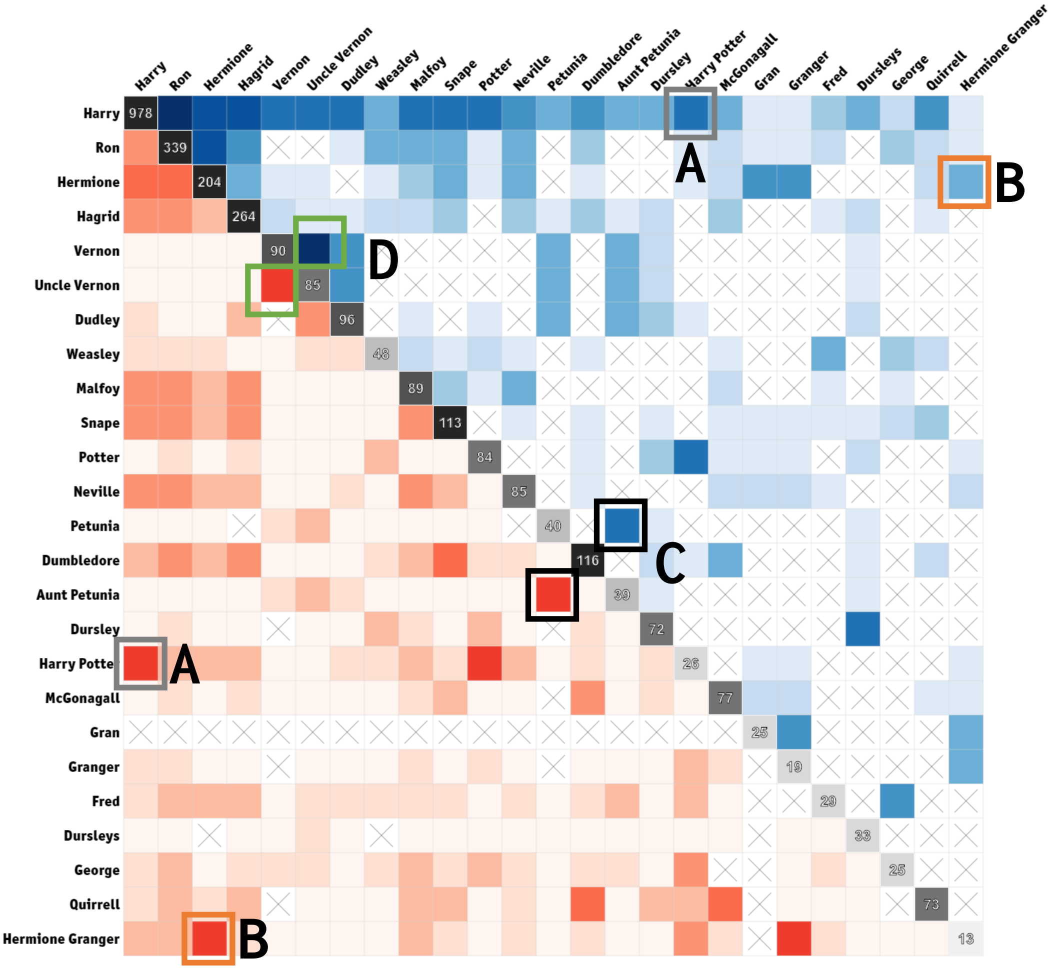


Figure 4.6: *Initial view of data from the Stanford NER output.* A, B, C, *and* D *are visually outstanding ambiguities that are candidates to be merged.*

characters. From that, we conclude, that most of the errors are caused by rarely occurring characters, which is plausible due to the high false positive rate of all NER techniques. Data created by the OpenNLP Name Finder contained the most resolvable ambiguities in the inspected data (*Winston* and *Winston Smith*, *Goldstein* and *Emmanuel Goldstein*). This is unexpected because of high precision and recall, although we observe a high false positive rate of 78%, which is likely the cause of these ambiguities.

J. K. Rowling: Harry Potter and the Sorcerer's Stone In Figure 4.6, the initial view of AmbiguityMatrix with data created by the Stanford NER is shown. Some local patterns stand out in both parts of the matrix, as well as some cells that are immediately recognizable to have either a high co-occurrence or similarity value, indicated by dark blue or red color of the cells. In the figure, we highlighted four of the immediately recognizable ambiguities with colored rectangles as follows: 1. Gray indicates the merger of *Harry* and *Harry Potter*, the co-occurrence as well as the similarity cell indicate high overlap of these two entities (A); 2. The ambiguity of *Hermione* and *Hermione Granger* stands out in the similarity part of the matrix, where it is indicated with an orange rectangle (B); 3. The consolidation of *Petunia* and *Aunt Petunia* is indicated with black (C); 4. Another case where the salutation is part of the consolidation and therefore resolves the ambiguity between *Vernon* and *Uncle Vernon* in the analysis results is highlighted with green rectangles (D).

Because of the similar performance of the other named entity recognition tools, their findings are also similar. Although, some details differ, for example, the OpenNLP-based results do not suggest to merge *Harry* and *Potter*. The rule-based system seems to be stronger with entities containing salutations. In our experiment, the visualization based on rule-based NER data was the only one that prompted to consider *Dumbledore* and *Professor Dumbledore* as a single entity.

Stephen King: Doctor Sleep In the last example, we applied AmbiguityMatrix on the horror novel *Doctor Sleep* by Stephen King.

The initial and final state, created with data generated by the rule-based NER, is depicted in Figure 4.4. We found a number of clear ambiguities, such as or *Abra* and *Abra Stone*, that are shown in Figure 4.4b. Noteworthy is that in contrast to the other candidates we found some examples where the similarity, as well as the co-occurrence, do not correspond towards merging characters, such as it is the case for *Dan* and *Danny*. While the co-occurrence is pointing in the direction of a candidate of ambiguities to merge, the similarity does not stand out. This is one of the cases, where human knowledge is required to make sure that the two candidates are merged for a good reason.

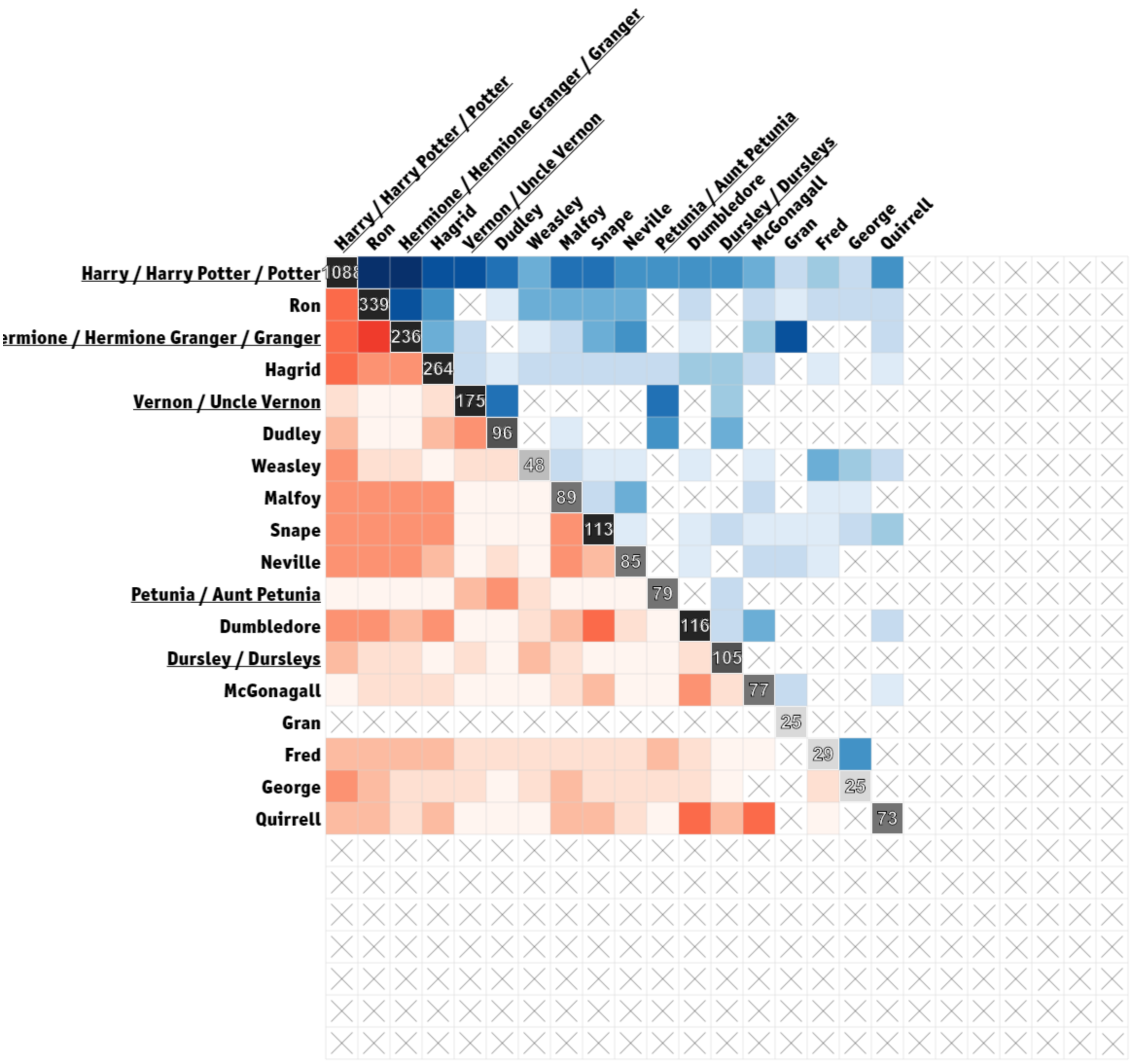


Figure 4.7: *Final view of data from the Stanford NER that started with 25 entities in Figure 4.6. This view has 18 entities left.*

An interesting observation made during the ambiguity resolution process is that with the merger and immediate display of the results transitive ambiguities, and their possible consolidation get visible. In the example of the Harry Potter novel, after merging *Harry* and *Harry Potter*, the connection of the newly merged entity and *Potter* got visible. This lead to an entity composed out of *Harry*, *Harry Potter*, and *Potter*. A similar observation can be made with *Hermione*, *Hermione Granger*, and *Granger*.

4.5 Summary

In addition to what has been showed in this chapter, iterations with adapted NER models or rule-sets are also possible, for example by displaying more than one visualization in juxtaposed views. This comparison also has the potential to allow the visual examination of the model evolution. Matrix-based visualizations are specifically powerful for depicting visual patterns. In a small-multiple setting, the presence and absence of visual patterns caused by the corresponding interactions of the user can be spotted easily. We are currently experimenting with these user-interface extensions, which are additionally providing a graphical depiction of analytic provenance, similar to the illustration in Figure 4.3.

Coming back to the presented use-case of interactive named entity disambiguation, we introduced our work as a tool to create (post-processing) rules. It is natural to extend our approach feeding the user interactions back to the named entity recognition process to adopt the corresponding data analysis techniques. For a rule-based NER, an inference of new rules based on the manually selected entities is possible. For machine learning-based techniques, a large number of interactions and examples will be needed to be able to re-train the model or the employed method to incorporate human feedback. The available models are trained on massive datasets, and a small number of new examples will have no noticeable impact on the performance. To cope with that problem, techniques to over-represent the manually selected examples could be employed. For interactive systems, this can lead to significant processing and waiting times for machine-learning based NER software, though this problem will not occur for rule-based NER.

While we focus on *semantic ambiguities* that are grounded in typical errors such as missing salutations or family names, in some cases we found errors and duplicates caused by punctuation and other characters, that have not been stripped from the extracted entity. This allows the distinction of four types of meaningful rules that are possible to generate with AmbiguityMatrix. First, rules based on ambiguities of entities indicated by co-occurrence in the upper right part of AmbiguityMatrix. Second, the semantic similarity, as depicted in the lower left part of the matrix visualization. Third, there are rules which are triggered by both, the co-occurrence and semantic similarity. Fourth, rules that correct spelling based post-processing. This information could be a good start for future extensions of the presented analysis and visualization technique.

The presented approach was designed for the interactive improvement of named entity recognition but can also serve other purposes, such as interactive model comparisons, which can be done by showing multiple of the proposed visualization side by side. Enriching the views with linking and brushing is an effective tool for comparisons, in particular of the data represented per column and row is kept the same for all views. Besides different models and software packages, also different parameter sets can be compared interactively.

In this work, we concentrated on the entities of type *person*, while NER software typically assigns more classes, such as *location*, *company* or *time*. It is of interest to examine whether the proposed methodology is also applicable to these entity classes, although, we suspect this is not the case. The character disambiguation of characters (persons) as presented in this work relies heavily on the type of referral to the characters, because of the language used by the author, see Figure 4.2 for example. From our experience with the data, this cannot be assumed for other entity classes, such as locations or company names. Location or companies are commonly addressed by their name. In consequence, we do not expect to find enough variance in the data, so that the idea of exploiting co-occurrences to point to possible candidates for a merger will work. The semantic similarity based on word-embeddings should still be useful, as the embedding does not depend on the word it is created for.

During our experiments, we also found misleading cues. For example, in the Harry Potter novel that is subject to the evaluation, 4.4, the visual hint was pointing strongly to merge *Professor McGonagall* and *Professor Dumbledore*. The semantic similarity between these two characters is 0.83, in a range from 0 to 1, while the visualization of co-occurrence was not strongly indicating a possible merging candidate. While we only rarely observed these problematic cases, they are a hint to add even more context as presented in Section 4.3.5 to the visualization, for example by presenting relevant text snippets of entity (co-) occurrences.

In this work, all matrices presented contain at max 25 characters. We did this on purpose, as experiments with the data showed that books in our collection did not contain more than 25 main characters, determined by frequent interaction. For this work, there are two different types of scalability to take into account: data-, and visual scalability. Data-wise, increasing the number of characters is not an issue, as the employed NER toolkits already generate more than 25 characters. In a collection of around 1,600 e-books, we found that the majority of books contains between one and 200 characters (entities of type person), while the number of character co-occurring with others is much lower, around 120. A study by Ghoniem et al. [133] suggests that even for 100 characters, the matrix visualization stays interpretable, and will perform better than competing techniques such as node-link diagrams. Therefore, we argue that the proposed visualization technique meets the scalability requirements for the presented task of resolving ambiguities in feature sets for a machine learning application.

5

Towards Transparent Predictive Policing

In this chapter, we illustrate how the transparency concept described in Chapter 2 can be applied in real-world scenarios. More specifically, we showcase two points where we provided actionable, interactive visualization-based visual analytics tools that control or illustrate feature sets and their context for the application problem of visual analytics. Together with domain experts, we developed the so-called *predictive policing process* that structures the application problems in five consecutive steps — plus an evaluation phase — that serve as the application-level foundation of this chapter. Again, the actual machine learning implementation is not part of this work. Instead, we utilize its results, similar to the previous chapters, to control a spatial clustering method and provide insights concerning a changing feature set and hyperparameters. Second, a visual analytics tool called *polimaps* has been developed specifically for the visualization of the machine learning output enriched with a user-definable set of features. *polimaps* is the communication facility between the machine learning system that computes the predictions, analysts that can choose from the output what fits best to their current strategy or prior knowledge that has not been included in the prediction model, and the police forces that utilize the prediction in their daily work routine. In this chapter, we contribute two concrete and productively used visual analytics systems that provide transparency on two levels. First, the behavior of a spatial clustering is made observable by involving the analyst into the hyperparameter selection, configuration of the feature set and the adaption of the feature weights Using comparative visualization, the effects — so to say, the different behavior — of the clustering method can be inspected visually. Second, we contribute a visualization system that enables analysts to enrich machine learning outputs with feature sets they can choose on their own, implementing the concept of visualization and reasoning with the interactive visual interface.

Note: *All pictures shown in this chapter serve illustrational purposes only. The displayed spatial data has been generated based on population density, if not stated otherwise. All other depicted information serves illustrative purposes only. Utilized map tiles © OpenStreetMap contributors, satellite images © Esri.*

5.1 Challenges and Tasks

Recently, predictive policing has gained much attention. The benefits of methods labeled as *predictive policing* make their deployment and application in real-world scenarios very attractive for law enforcement agencies (LEAs). According to Perry et al., there are four different categories of methods in the area of predictive policing [70], which are:

- Methods to **predict crimes**. This is the most obvious category of methods and a very fundamental category. Typically, they concentrate on predicting the spatial location and time of offenses.
- Methods to **predict offenders**. Techniques of this category try to identify individuals with a risk of committing an offense in the future.
- Methods to **predict the identities of perpetrators**. This category conveys methods to identify characteristics to match offenders that committed crimes in the past.
- Methods to **predict victims** of crimes. Includes techniques to identify individuals or groups of individuals that are likely to become victims of crimes.

On the organizational side, there are a number of potential benefits, for example, a better allocation of existing personnel which can lead to increased cost-effectiveness of the police force.

In the following, all methods and techniques refer to the first category and contribute to forecasts of spatial locations and the time of offenses[1].

Police forces all over the world are using databases to store information about offenses, e.g., the date and time of the incident, the location, and various other metadata that typically depends on the type of crime. These vast amounts of data are used primarily for spatiotemporal analysis, for example, to identify trends or hotspots [94, 38].

Besides standard data analysis tasks that refer to the past, predictive modeling and in consequence predictive policing is gaining more and more attention [86, 50], as it promises to anticipate offenses or (new) crime hotspots appearing in the future [70].

[1] No individual data is involved.

Consequently, strategic decisions based on predictive policing methods are designated to lead to better resource allocation, and ultimately to a reduction of crime. Even though that there are commercial applications available [2, 9], positive effects of their application have not been proven yet [46, 39, 30]. A major problem of evaluating predictive policing is that in general there are spatial and temporal correlations of the criminal incidents and the deployment of predictive policing methods. These correlations are caused by effects such as sparse or incomplete data, global influences such as weather, huge events or other external influences that are hard to include in the pool of data utilized for a quantitative evaluation. There is no objective indicator of the effectiveness of predictive policing, as a clear and definitive causal relationship has to be identified first, which has not been done convincingly yet. Although, the expected benefits make the development and implementation of such systems attractive to police forces.

What we present in this chapter is part of a predictive policing pilot study of a German law enforcement agency on state level. Analytically, the goal of the study was to develop and deploy a method to predict the risk of a specific offense (domestic burglary) for a set of predefined geographical regions, the so-called *prediction areas*. The prediction models are created with state of the art data mining methods by the LEA's analysts, as another goal was to understand the technical and analytical background of the procedures that constitute predictive policing implementations. This approach in line with the solutions presented in the previous chapters, as they also include a black box-like machine learning back-end, as motivated in Section 1.1.1.

The process that builds a predictive policing solution is typically depicted as the "*Prediction-Led Policing Business Process*" by Perry et al. [70]. The authors build and document a predictive policing cycle as well as the interplay of elements that are part of an actual implementation. Unfortunately, details about a concrete implementation are missing, which is usually the case in the literature that documents similar approaches, and builds the foundation for a *good* quality measure. Therefore, we propose the predictive policing process, as shown in Figure 5.1, to bridge this methodological gap. Divided into six different steps, the proposed schema is motivated from a practical view on the problem of predictive policing, although deviations, and possibly more detailed variations, are conceivable. Still, the overall structure of the process should be preserved and enriched with techniques from the visualization domain. Also, in the light of real-world applications and the deployment tools such as *polimaps*, that will be introduced later in this chapter, a dedicated feedback cycle for continuous evaluation and feedback by different user groups is included.

In the following, we introduce the predictive policing process, which provides a generic platform for similar experiments or applications, not only concerning visualization. The process has been developed in close collaboration with the domain experts from the LEA and is a simplified, task-oriented view of the application problem. Each of the proposed steps is a challenge for transparency in itself, a combination of them is even more challenging, as also the target user groups change.

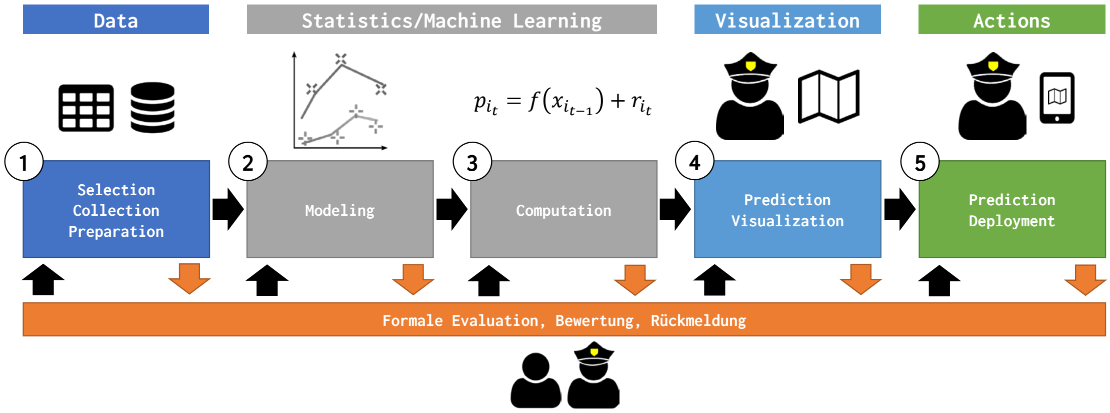


Figure 5.1: Predictive Policing Process. *Steps illustrated in color, (1), (4) and (5) profit from visualization in various ways. Steps (2) and (3) are the points to develop and apply methods from visual analytics or aspects of human-centered machine learning. The steps (4) and (5) are executed by local police forces and therefore include the targeted end users. (6) sketches an ideal, ongoing parallel evaluation and feedback process that refers to all steps of the process.*

The primary goal of the predictive policing process is to structure the contained tasks and methods of data wrangling, machine learning/statistics, and visualization, for a clear understanding of the parts of a predictive policing implementation. In the following, we outline the different tasks and give connections to our work.

Step 1: Data selection, Collection, and Preparation. The subject of this step is the selection of the utilized datasets, identify and exploit reliable sources for these datasets, and develop a data preparation process, if needed. Almost any geospatial visualization can help to visually identify correlations, e.g., by overlaying data from police databases that indicate offenses on the map with other datasets from different sources. Naturally, it is also possible to inspect the effects of the data preparation with a visualization of the datasets before and after the data is processed, for example, geographic coding of addresses or point of interests that come without spatial coordinates. Of particular interest is the spatial and temporal relationship in the data as the spatial and temporal prediction of the goal of the application problem. At this stage, police data, e.g., from the police databases, as well as non-police data, such as weather, properties of residential areas, or the distance to the nearest motorway, can be combined to form a single dataset. Therefore, it is crucial that all of the potential datasets that are combined exhibit a spatial reference. A dataset that is suitable for automated methods such as machine learning will be the output of this stage. There is a variety of different concepts and techniques to choose from that have different requirements concerning the data. For example, there are possibilities to include data that describes what is commonly called *Near Repeat* [101], or more scientifically grounded ideas. Although the concepts and theories are different in their fundamental assumptions, they exhibit some commonalities, e.g., they do not rely on data of individuals, most of them focus on domestic burglaries, and data-wise, historical data from the offense type to predict is utilized for modeling.

Step 2 and Step 3: Predictive Modeling and Computation. This step can be subsumed under the terms *statistics*, or *machine learning*, and is the core of any predictive policing implementation. In this two steps, the actual model is created (Step 2) and applied to compute predictions in space and time (Step 3). As prediction models are the methodological core of any predictive policing implementation, they can benefit most from augmenting methods, such as human-centered machine learning [20]. Reason for that is that there exists no model of criminal incidents that is exhaustive and sufficiently accurate to be useful to a practical extent. Therefore, the incorporation of expert knowledge to select appropriate models or model parameters is essential in the modeling phase. Using the historical data from the previous step, a model is trained. The goal is to be able to model the spatial and temporal occurrences of offenses with the highest precision, which in turn makes a validation phase an integral part of the modeling phase. Potentially, machine learning techniques based on regression [32], decision/classification trees (e.g., CHAID [184]) or artificial neural networks [130] could be used to model spatial and temporal occurrences of criminal offenses. During computation of the forecast, the model created in the previous step is applied to compute the probability that an offense happens in a corresponding geographic area (prediction area). The result from the prediction computation are risk-scores assigned to the prediction areas, that indicate a higher risk of the modeled offense compared to other areas.

Step 4: Prediction Visualization. Supporting this step is the core of any visualization system that should be used in the context of predictive policing, although the visualization techniques depend on the actual model and prediction subject. Prediction results, in our case the predicted risk scores, could be visualized together with the corresponding prediction area. As it is subject to our work, adding visualizations with data from the past helps to create a visual context of a prediction for plausibility checks, or further model adjustments. The prediction visualization comprises an adequate visualization of the output of the previous step to be able to deploy the prediction to operative police forces.

Step 5: Prediction Deployment. Visualization systems that support predictive policing should be able to export views, for example to an image, for further use. The exported views could be used as a handout for personnel working on the streets, but also to visually document the prediction of a specific model. Additionally, if possible, a tool supporting this step could be the method for prediction deployment itself, for example on personal devices of police officers. A question coming up at this stage is whether the prediction visualization should be distributed in a digital or non-digital form. The first possibility can be utilized for the interactive visualization, e.g., with common GIS tools, while the latter refers to a suitable distribution on paper.

Step 6: Evaluation. In an ideal world, the evaluation phase spans all of the different steps of the predictive policing process. For each of the steps, different evaluation methods and quality criteria can be formulated. For example, during the first step *Data Preparation*, possible quality measures could include a quantification of the completeness of the datasets as a general idea of quality, or some precision indicator for the included spatial coordinates. Second, from the domain of machine learning, a number of quality measures exist to specify the quality of such models, e.g., a training and test/validation methodology that produces accuracy or precision measures. The same applies to the *Prediction Computation* phase. In the last two steps, the *Prediction Visualization* and *Prediction Deployment/Action* phases, the quality is much harder to assess, as we will outline in Section 5.4.3.

5.1.1 User and Task-based Transparency

Our work targets two different users groups. First, the analysts from the LEA on state-level, and second the field personnel from local police departments. Both of these groups differ fundamentally in their tasks.

The LEA analysts are responsible for the methodological and technical parts of the predictive policing process. They are in charge of the selection of potential datasets, as well as the creation of a unified data foundation for the machine learning process (Figure 5.1 1). Additionally, the predictive modeling, as well as the forecast computation, is part of their involvement, which comprises Steps 2 and Step 3 from Figure 5.1, and results in a number of prediction areas with associated risk scores.

The personnel from local police departments act as the executive force and bring the predictions into action. Fundamental for their work is the involvement in the methodological forecast generation, as they can adjust the prediction area selection, e.g., based on local strategies or knowledge from general police work — to accomplish this task, a visual interface is required (Figure 5.1 4). Finally, the deployment of the prediction to the police offers, as well as any coordinated actions based on the forecast, are also part of the local police force personnel.

Based on this user groups and task allocation, the following high-level task assignment, based on the predictive policing process, to the user groups have been created:

Based on this task assignment, the following potential methods to provide transparency, or observable behavior are possible.

1.1–1.3: Transparent Data Selection, Data Collection, and Data Preparation. Given the fact that in machine learning workflows, data is an essential input for model training and therefore the fundamental generalization process, the data input itself can be provided directly for immediate transparency. Additionally, metadata such as the date of collection, any preprocessing steps as well as their concrete parameters could be interesting on a methodological level. Although, having a number of different data sources, follow up issues for transparency arise, e.g., the reasons for their selection.

	Task	User Group
1.1	Data Selection	LEA Analysts
1.2	Data Collection	LEA Analysts
1.1	Data Preparation	LEA Analysts
2	Predictive Modeling	LEA Analysts
3	Prediction Computation	LEA Analysts
4.1	Prediction Visualization	LEA Analysts, Local Police Forces
4.2	Prediction Area Selection	Local Police Forces
5	Prediction Deployment	Local Police Forces
6	Evaluation	LEA Analysts, Local Police Forces

Table 5.1: *High-level task and user groups assignments, based on the proposed predictive policing process.*

In this project, the motivation for selecting the datasets was coming mostly from a theoretical background that has been created to fit the offense type to forecast [17], the provenance documentation of the data utilized is done by a data management back-end.

2: Transparent Predictive Modeling. At this stage, the most crucial source of transparency is the predictive model itself as well as its hyperparameters (see Section 2.2). Fundamental differences in the actual suitability of models with regards to the idea of observable behavior exist. For example, it is clear that a naïve Bayes classifier is inherently better suited to communicate its decisions and outcomes, in contrast to a neural network-based technique with a number of hidden layers and potentially complex activation functions. In this project, primarily for reasons of transparency, a decision tree is used to do the predictive modeling. Such techniques are predestinated for transparent machine learning, as similar to the naïve Bayes classifier, they are building up the final model based on data features and decisions that can be followed easily. Besides the technique to compute predictions and the data input, a spatial reference, previously introduced as *prediction areas* are required. For a transparent process, it is crucial to be able to explain the prediction areas in some way. While an initial set of prediction areas by a vendor of the data was available, we designed a visual analytics tool that supports the LEA analysts in creating the prediction areas based on a configurable feature set. Therefore, the prediction area origin is documented by the hyperparameters of the corresponding method, see Section 5.2.

3: Transparent Prediction Computation. The actual process of prediction computation and its exhibited degree of transparency is depending almost entirely on the employed machine learning methodology. As argued before, the transparency of deep learning-based methods is still an open research topic, techniques such as the employed decision trees provide an unmatched degree of transparency during prediction computation. For each risk score assignment, the decision tree can be followed from the root node to the final leaf-node, while each of the nodes in-between is a combination of a feature and therefore represents an observable and interpretable decision criterion. Finally, decision trees are transparent by definition and can be visualized and inspected with the corresponding modeling tools.

4,5: Prediction Visualization and Prediction Area Selection. Here, we leave the transparency idea as coined in Chapter 2, as the prediction visualization is not including any information from the machine learning technique as it is used in the previous task. In contrast, we try to make the behavior of the prediction creation observable by taking most of the features that contribute to the prediction model and visualize it together with the prediction areas using their spatial reference. This approach may not be contributing to methodical transparency, but it allows the analyst on state-level, as the local police forces to set the prediction outcome in the context of the utilized datasets, which facilitates an understanding of the prediction. On a fundamental level, the idea is to enable analysts to interactively explore the correlation of the features that build up the prediction model, and its outcome using an explorative interface. Also, the visual context of a prediction can be created by entirely different datasets, e.g., data that depicts other offense types, which contributes to the second task: the prediction area selection.

6: Transparent Evaluation As there is until today no convincing evaluation of a predictive policing system, doing a transparent — or observable — evaluation poses a challenge. The reasons for that are manifold. Most prominently, the details of the commercial predictive policing systems are a company secret and are kept confidential. Exposing the behavior of the predictive policing system in a thorough evaluation would expose those secrets, as an evaluation of the methodological level is an essential building block. Similar, such an open evaluation that is comprehensible would open strategic information from the police forces, e.g., details and information from the police databases that are not shared with the public not to give interested parties insights in concrete police work. On the academic level, most of the studies about predictive policing are flawed, either by design, by the data involved or do not care enough about the dynamic nature of offenses in space, time and the social and economic influences.

In the following, we elaborate on work in the area of prediction visualization and prediction area selection — predictive policing process, step 4 and step 5 — as well as a mix of Step 1 and Step 2 that constitutes the transparent computation of prediction areas.

5.2 Transparent Prediction Areas

In commercial predictive policing systems, mostly from the US, predictive policing systems typically utilize regular *prediction boxes* [8, 35], e.g., with a cell size of 150m×150m. While this seems appropriate for the US, where a block is an essential element in urban planning, there are good reasons to compute potentially non-regular prediction areas. For example, in Europe urban areas are mostly formed by some growth processes over decades and have not been planned block by block. Also, the custom computation of prediction areas allows for further constraints, e.g., a homogeneous distribution of households, a restriction of the minimal and maximal number of buildings in a prediction area, the exclusion of purely industrial areas, or similar that could be dependent on the offense type.

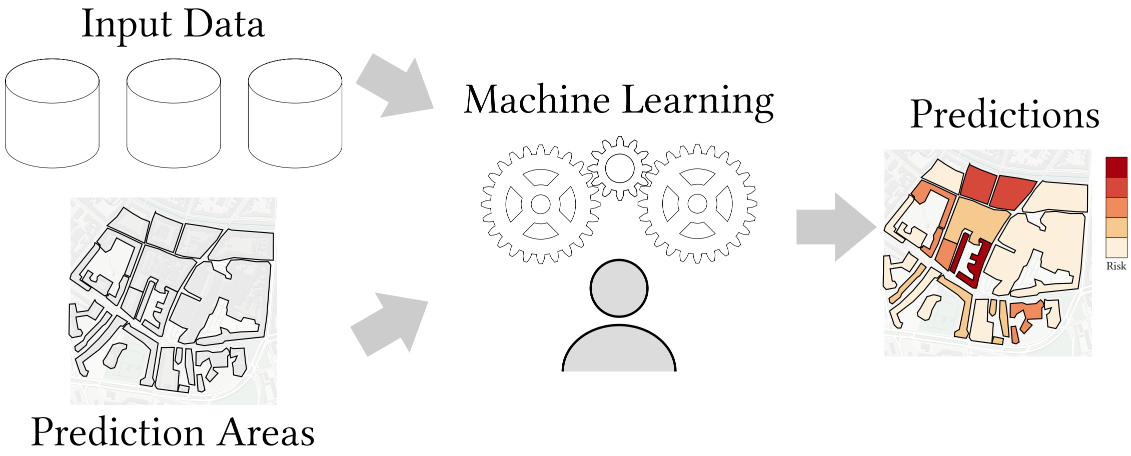


Figure 5.2: *Schematic of the predictive policing application problem, illustrating the input and outputs of the process.*

As argued, the prediction areas are a crucial part of the machine learning technique as they represent the spatial reference of the data, and are a main element of the machine learning problem, as illustrated in Figure 5.2.

As a number of clustering methods are available that produce geographic partitions that can be used as prediction areas, e.g., k-means [182], CLARANS/SDCLARANS [174], DENCLUE [143], and other clustering algorithms such as DBSCAN [167] that can be utilized with spatial distance metrics, and derivative versions of those techniques. To compute the prediction areas, a commercial, dataset with socioeconomic data from the areas where the predictions should be computed has been available that is outlined in Table 5.2. The dataset exposes four different hierarchy levels, from single houses to municipalities, and provides different attributes per hierarchy level.

	Name	**Reference**	**Features**	**Examples**
Level 0	Municipality	area	104	population
Level 1	Residential Quarter	area	361	purchasing power, number of households
Level 2	Road Section	point	121	distance to train station, primary use of buildings
Level 3	Building	point	112	number of households, year of construction

Table 5.2: *An overview of the hierarchical socioeconomic dataset that is used to compute prediction areas.*

While there is, at Level 1, already an area-based spatial reference available, its origin is unclear, and must not coincide with any social or demographic attributes that fit the offense type. This motivates a visual analytics approach that allows the interactive manipulation of the feature set used to cluster. The tool is tailored to the analysts on state-level and involves them directly in the computation of the spatial references for the prediction algorithm. Therefore, we argue the prediction areas have been created transparently, as the feature set, feature weights, and hyperparameters of the spatial clustering can be interactively modified, re-run, and compared in a simple geographic view. Additionally, the tool supports an ad-hoc analysis of cluster borders to check if the hyperparameters and the selected feature set can discriminate the sociodemographic data as required by the state-level analysts.

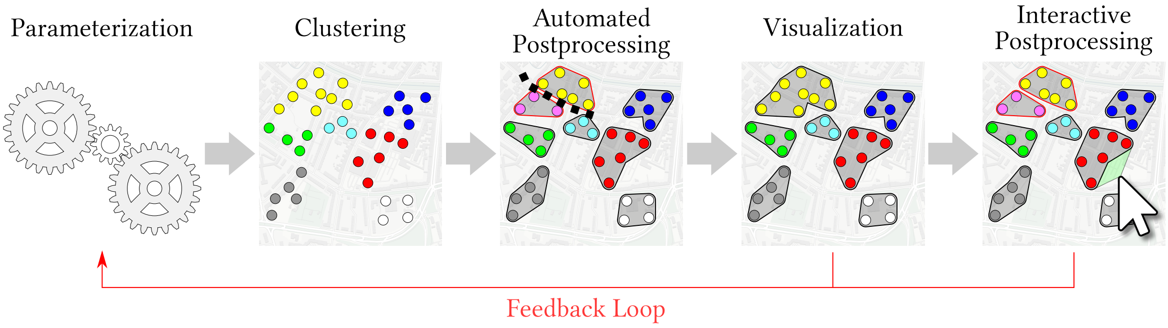


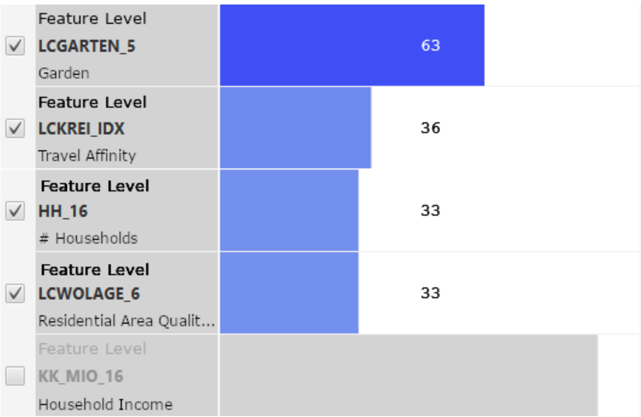
Figure 5.3: *Clustering Workflow, from the parameterization of the clustering algorithm over interactive visualization to the generation of the prediction areas and the corresponding features. To adapt the clustering, a feedback loop from the visualization part exists that feeds back to the parameterization.*

The solution for the generation of transparent prediction areas has six different parts, as shown in Figure 5.3, which are part of an interactive, visualization-based prototype.

5.2.1 Spatial Clustering

A number of possible techniques to partition data points that refer to a building exist. Therefore, before implementing the final clustering solution, we executed an iterative and user-driven feedback process to explore faults and artifacts generated by common clustering techniques such as DBSCAN [167]. For the feedback process, many different clusterings have been produced, and the state-level LEA experts conducted an ad-hoc evaluation of the results. The feature sets, as well as the parameter settings for the clusterings that were subject for evaluation, have been created in conjunction with the LEA experts. Methodically, the clustering is divided into three phases. In the first phase, the data is flattened into a tabular representation on Level 3 (buildings, see Table 5.2). All features from the parent levels that are part of the feature set are copied to the corresponding building so that all data from the road section of the building, the residential quarter and the municipality can be utilized without the need for hierarchical processing of the data. After the data has been flattened into a non-hierarchical structure, the data matrix is extended by two columns:

a latitude and the longitude of the corresponding building, so that also the spatial reference is included in the data matrix. Afterward, the matrix is indexed with a metric tree [123] that supports efficient geospatial neighbor queries on the building coordinates, as well as a classic R* index [71] to support efficient point to area queries. After the data structure has been initialized, a two-phase clustering approach is executed. First, a k-means clustering extended with least squares quantization by Lloyd [182] is performed, to create an initial, complete partition of the spatial area the data points span. k is initialized with the desired numbers of prediction areas, which correspond to the found clusters. In this initial clustering, attributes except for the location of the data points are not part of the utilized distance metric, which uses a Haversine formula-based approach to compute fast, but approximative spatial distances. Afterward, an OPTICS Xi-based density clustering [157] is executed to further separate dense regions and refine the initial cluster result. It operates on all features from the feature set that are not referring to the location of the data points, the utilized feature set and the corresponding weights have to be set by the analyst, see Figure 5.4 for the user interface. The parameters $minPoints$ can be set in the clustering combination as the minimal size of clusters, ϵ is a generic configuration parameter referring to the density of the produced clusters, which also serves as the primary tool when creating different clusterings of the same geographic region. The Xi is set via a simple, inverse heuristics of the lowest density of the smallest five percent of the generated cluster. This heuristic has been approved in a series of experiments that focused on clusters in areas with homogeneous building structures, as well as border regions that are, compared to inner areas of cities, sparse regarding the number of contained buildings.



☑ Feature Level
LCGARTEN_5
Garden
63

☑ Feature Level
LCKREI_IDX
Travel Affinity
36

☑ Feature Level
HH_16
Households
33

☑ Feature Level
LCWOLAGE_6
Residential Area Qualit...
33

□ Feature Level
KK_MIO_16
Household Income

Figure 5.4: *Feature controls for the clustering configuration. Each feature is represented by separate controls that allow to quickly include or exclude them (checkbox on the left), shows some information about the feature and includes the colored bar that indicates the feature weight on the right. The number is referring to a relative weight between 1 and 100 and is shown for reference of the analyst only; the background colormaps the same value on a range from light blue to dark blue.*

Compared to an ordinary DBSCAN clustering, this approach has two significant advantages: first, the k-means clustering produces a complete partitioning of the data space, so no noise is created by the first cluster method. Second, Optics Xi generates clusters with a varying degree of data point density, in particular, it produces better cluster in areas that are of a varying density, e.g., border regions of cities, or mixed regions with residential and industrial developments. Finally, the two cluster results are merged, so that the k-means clustering is refined in dense areas, as a configuration parameter allows to set a maximal size of buildings in a cluster. Wherever possible, clusters that have fewer than the required number of buildings are merged with neighboring areas.

To cluster a metropolitan area containing around 35,500 buildings, a standard workstation[2] requires between two and six minutes. Although, the actual runtime depends heavily on the actual configuration. Still, the performance proved to be suitable for the envisioned approach of computing different results to determine suitable parameters and finally produce a usable set of prediction areas.

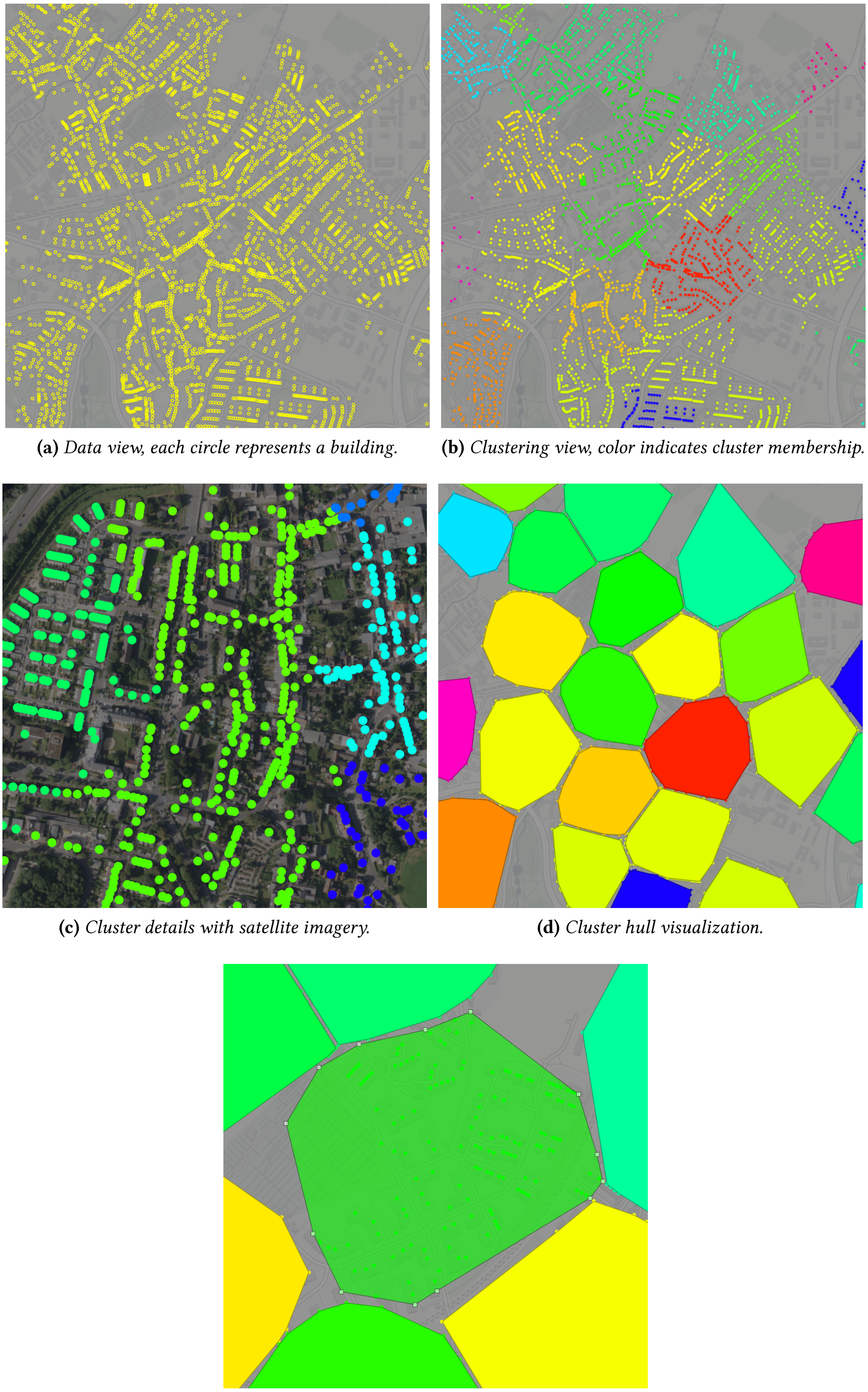
5.2.2 Cluster Visualization

To provide insights into the behavior of the clustering algorithm, we visualize its outputs. As each of the computed clusters is composed out of a number of points, each of the points is visualized on the corresponding spatial location of the building, as each point represents a building from the clustered dataset, see Figure 5.5a. During cluster computation, each record is assigned a cluster id, which can be used to identify the members of a cluster, as illustrated.

When a specific clustering is selected, the points are filled with a color according to the cluster id, see Figure 5.5b and Figure 5.5c. The colors are assigned in a two-step process. First, the cluster color is chosen by utilizing the color cone of the HSB color space. Given n clusters to visualize, each cluster is assigned an angle α on the hue dimension of the cone as $\alpha = 360/n$, which generates distinctive colors for all clusters. To make sure that neighboring clusters have a distinctive color mapping, the clusters are ordered to maximize their pairwise distances before the colors are assigned. For the evaluation of the clusters, this simple visual mapping is already suitable and has approved to be useful to the state-level analysts. For example, Figure 5.5c illustrates the setting when the cluster borders are inspected visually. It can be seen that a residential area has been separated into different parts, although, from the satellite images, no difference is apparent. Based on the assumption that differences in the building style and their surrounding also indicate different socio-economic population which, in turn, indicates that the corresponding prediction area should not be separated, this could be the reason to modify the clustering parameters, e.g., by reducing the number of desired clusters, and re-run the spatial clustering.

Figure 5.5d illustrates how the convex cluster hulls are visualized. The analyst can choose between concave [78] and convex [166] cluster hulls, depending on the cluster shape. Initially, the system

[2] Intel Core i7 3770 3.4 GHz, 16 GB main memory, SSD drive



(a) *Data view, each circle represents a building.*

(b) *Clustering view, color indicates cluster membership.*

(c) *Cluster details with satellite imagery.*

(d) *Cluster hull visualization.*

(e) *Cluster hull in edit-mode.*

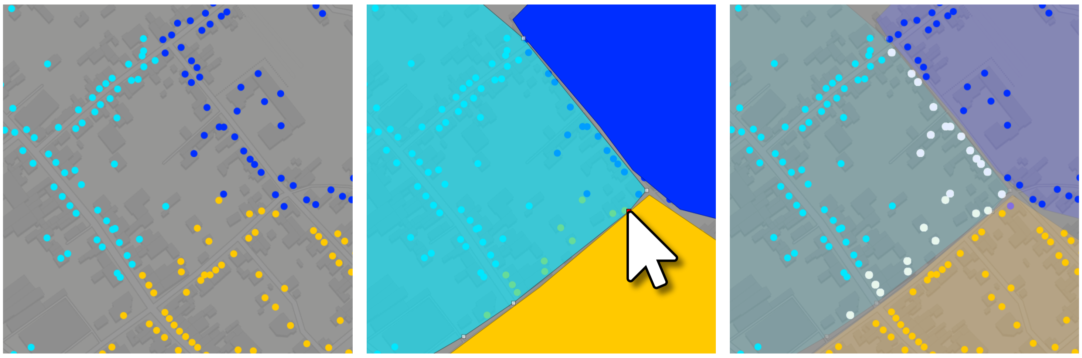
Figure 5.5: *Visualizations of the data and clusters.*

determines a suitable hull type for a random five percent sample of clusters, that can be changed manually later. Also, the hull computations can be subject to a separate simplification or buffering stage, so that the resulting hulls have a more appealing shape, as they could be part of printouts and visualizations.

5.2.3 Cluster Postprocessing

Because of parameter settings, the clustering result may produce clusters that span a river, a large street, or parts of two distinctive residential areas. As the clustering algorithm includes no restrictions and purely relies on the given configuration, such constraints regarding spatial features that are not allowed to be part of a prediction area, or act as a cutting element are part of a separate, automated postprocessing stage, see Figure 5.3 in the middle. The postprocessing supports two kinds of operations. First: splits. Splits are executed for large spatial features, e.g., rivers or big streets, that should not be spanned by a prediction area. Second: wherever a prediction area overlaps lakes, rivers, or similar spatial regions, the overlap is removed from the prediction area. The overlapping areas are artifacts from the hull computation; therefore this step is necessary to produce a clean set of prediction areas. For both of the automated cluster postprocessing steps, data from the OpenStreetMap is used, as it is available via the OverPass API[3].

While the described postprocessing happens directly after the clusters have been computed, interactive postprocessing is available that tightly integrates with the cluster visualization as illustrated in Figure 5.6.



(a) *Unclear separation between blue, yellow and the cyan cluster.* **(b)** *Interactive editing of the transparent cluster hull.* **(c)** *After editing, the white points will be part of the cyan cluster.*

Figure 5.6: *Interactive cluster editing.*

When the analyst finds areas where the cluster separation is not sufficient, e.g., a residential area is separated into three different parts, as shown in Figure 5.6a, interactive postprocessing can be applied

[3] OverPass API: `https://wiki.openstreetmap.org/wiki/Overpass_API`

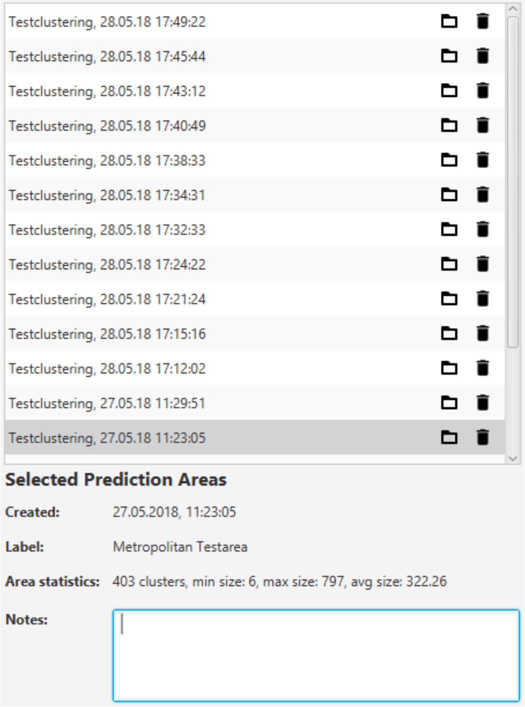
to resolve such issues. Using the cluster hull which can be modified, the cluster separation can be resolved by clicking on the cluster hulls. Each point contributing to the hull can be interactively modified, and the hull is updated and correspondingly visualized together with the interaction. At the same time, the opacity of the hull is reduced so that underlying structures and data points are visible while the changes are made, as it is illustrated in Figure 5.6b. Finally, when the editing is committed by de-selecting the changed cluster hull, the cluster memberships of the points are recomputed and visualized accordingly. In Figure 5.6c, for illustration of the changed memberships the points with changed cluster membership are shown in white. In the application, they are assigned the same color as the cluster, cyan in the illustrational case in Figure 5.6.

5.2.4 Prediction Area Generation

After the clustering workflow as depicted in Figure 5.3 has been executed, the corresponding prediction areas can be exported. The export includes the cluster shapes, as well as the cluster memberships for each data record that can be identified with a unique id. Generating only the actual spatial prediction areas is not enough. The following machine learning application relies on a dataset, as shown in Table 5.2, which exhibits a hierarchical nature. While the building (level 3) and road section (level 2) data can be just copied accordingly, the residential quarters (level 3) have changed most definitely, as they are the target of the prediction area computation. In consequence, all data in the dataset on level 3 corresponds to the original residential areas, and not the newly computed ones. Therefore, we tested a number of interpolation methods that map the data to the new prediction areas that resemble the previous residential areas in their structure. In a number of experiments we found, that standard spatial interpolation methods that refer to point data are hard to translate, and also hard to apply to the area-based numeric values that have to be interpolated. In particular, the data interpolation problem is not comparable to the standard, spatial interpolation of point values, as in this case it is about distributing existing data that refers to an area to new spatial distributions. We ended up using a mean interpolation method, where each numeric value from a previous residential area a is weighted with $\phi = |a|/|a'|$ where a' denotes the new predictive area and a and a' have a non-empty overlap. Finally, the interpolated data and the new prediction areas can be saved on disk, so that the machine learning process that assigns risk scores for the prediction areas can access both, the newly computed spatial partition as well as the corresponding data values.

5.2.5 Feedback Loop and Result Comparison

The clustering tool supports a feedback loop that can be used to adapt the clustering configuration, e.g., the desired number of clusters, the minimal and maximal number of buildings in a cluster, as well as some further hyperparameters that are soft optimization targets. After adaption, the analyst has to start the cluster computation manually, as the computations are memory and computation time expensive. There are two potential starting points for the feedback loop, and in consequence possibly changed clusterings: first, the cluster visualization, and second the interactive postprocessing. There are not objectified, static criteria that can be formulated and put into the clustering process as side conditions, which motivates why there is automation of the previously mentioned criteria. Therefore, both possible entries to the feedback loop rely on expert knowledge of the analyst, and can be utilized for different reasons, e.g., a large number of homogeneous residential quarters are divided into smaller areas which should be avoided, or the overall number of prediction areas is too large or too small.



Testclustering, 28.05.18 17:49:22
Testclustering, 28.05.18 17:45:44
Testclustering, 28.05.18 17:43:12
Testclustering, 28.05.18 17:40:49
Testclustering, 28.05.18 17:38:33
Testclustering, 28.05.18 17:34:31
Testclustering, 28.05.18 17:32:33
Testclustering, 28.05.18 17:24:22
Testclustering, 28.05.18 17:21:24
Testclustering, 28.05.18 17:15:16
Testclustering, 28.05.18 17:12:02
Testclustering, 27.05.18 11:29:51
Testclustering, 27.05.18 11:23:05

Selected Prediction Areas

Created: 27.05.2018, 11:23:05
Label: Metropolitan Testarea
Area statistics: 403 clusters, min size: 6, max size: 797, avg size: 322.26
Notes:

Figure 5.7: Clustering result switcher controls, organized by date and time of the clustering run. In the list on top, the different cluster runs are shown of which one can be selected. On the bottom, a summary of the selected prediction areas, i.e., clustering, is shown.

To enable reasoning about different spatial clusterings and their parameters, the tool provides the ability to switch the visualization between different clusterings rapidly. After a clustering run has been completed, the results are stored to the hard disk and can be loaded with a single click via the list on top of the UI shown in Figure 5.7. The visual mapping, as described before, allows the quick visual inspection of changes on an overview level, as well as detailed inspection on the building level, for example, to clarify changes in the cluster borders. While switching between different clusterings, the views are staying in their current settings to enable analysts to make quick visual comparisons or continue an exploration process at the same position with different clustering results.

5.2.6 Summary

The clustering tool supports the transparent generation of prediction areas for the predictive policing application. Transparency is achieved by involving the analyst in the parameter selection for the spatial clustering, and extensive visual inspection and adjustment functionality with interactive visualizations. It is also possible to compare different clusterings (sets of prediction areas) with each other visually, and therefore try to reach a configuration of the spatial clustering that produces a suitable and appropriate spatial partition. During the predictive policing study, it has been shown that the resulting prediction areas have a more homogeneous structure than the residential quarters that are part of the original dataset, which is regarded as useful for the application problem, and potentially beneficial for the performance of the machine learning method that assigns risk scores to the prediction areas.

5.3 Visual Analytics for Feature-based Transparency

Part of our efforts to achieve transparency in the pilot study relies on the visualization of the prediction that is realized as a risk score per prediction area, in context with other data. The method of choice was an interactive, visualization-based tool that is called *polimaps*. *polimaps* follows what Keim et al. designated as the basic idea of visual analytics, namely it "*visually represent[s] the information, allowing the human to directly interact with the information, to gain insight, to draw conclusions, and to ultimately make better decisions*" [107]. The insight and conclusions are in the context of the predictive policing application tailored to the interactive selection of prediction areas by local police forces, which has two benefits. First, it forces the local analyst to inspect the predictions and to work with the machine learning output, instead of just accepting something the state-level authorities create. Second, it allows the incorporation of local knowledge in the prediction provisioning that naturally does not exist on the state level, e.g., about usual hotspots or areas that are part of a local, offense-specific crime prevention strategy.

In subsequent discussions with analysts from the LEA, we identified two fundamental tasks that *polimaps* is required to support.

Prediction Visualization & Prediction Area Selection. The primary goal of *polimaps* is to allow analysts to inspect the prediction outcomes visually. To do so, we map the risk scores onto the fill of the related prediction area with a set of sequential colormaps [142] the analyst can choose from. The result is a classic choropleth map indicating the predicted risk over the set of all prediction areas. The lowest layer that is the foundation for all visualizations in *polimaps* is fixed to a map visualization that, by default, displays OpenStreetMap tiles [16], and is capable of offline tile rendering. Users are free to choose a virtually unlimited number of layers on top of the map and to combine them arbitrarily. This task refers to Step 4 of the predictive policing process (Figure 5.1). The prediction visualization is interactive. Each polygon in the choropleth map (prediction area) can be removed or restored from the visualization with a single click. This task is part of supporting Step 4 and Step 5 of the predictive policing process (Figure 5.1).

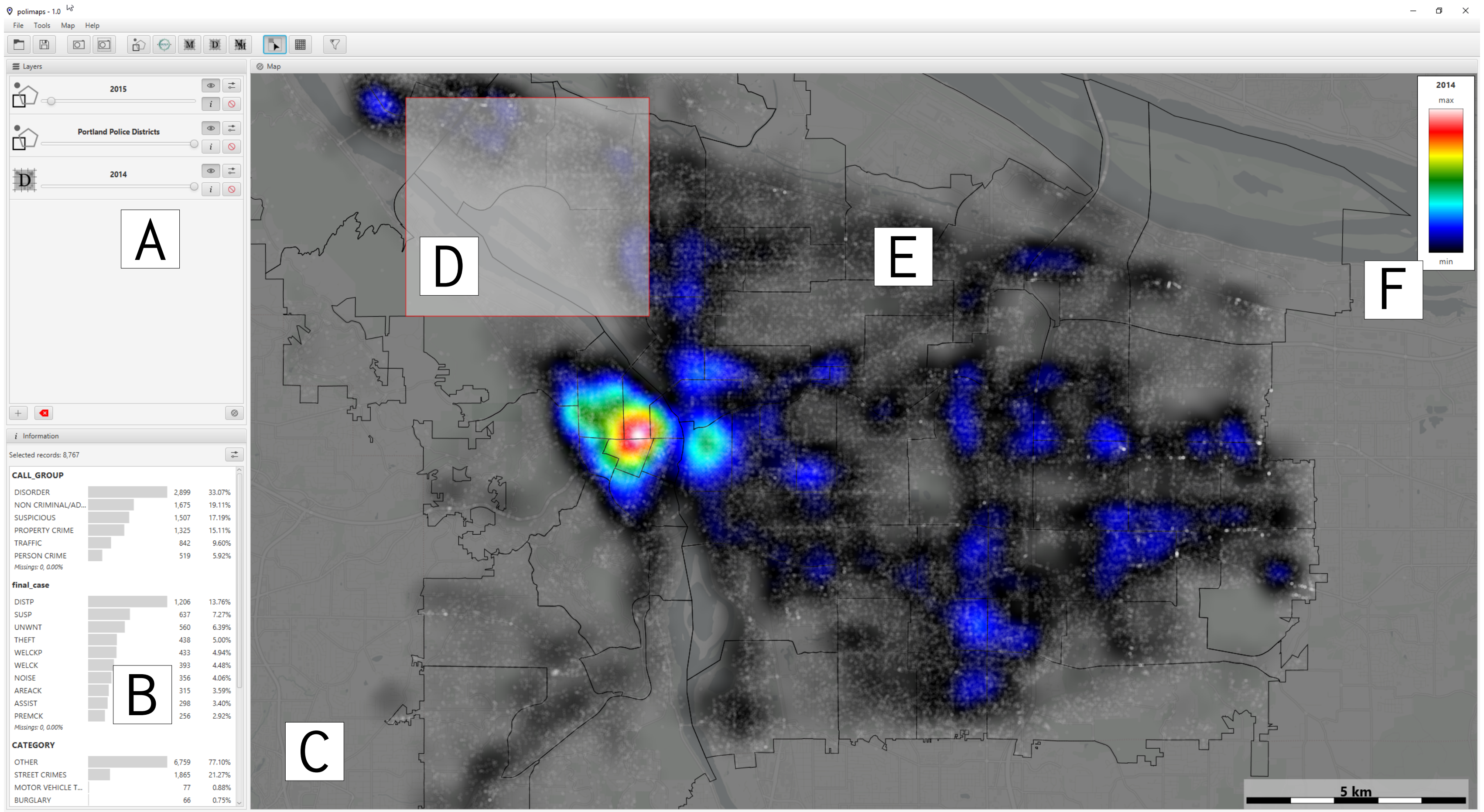
Visualization of Prediction Context. The creation of the context for a prediction is supported by two different visualization techniques. First, the visualization of spatial primitives, and second heat maps. The visualization of spatial primitives, such as points, rectangles or lines, supports custom fills, strokes, and some other parameters, e.g., custom labels of the primitives, contributing to their appearance (see Figure 5.8 E). The heat map, based on kernel density estimation [181], is capable of showing point densities or the aggregated value of an attribute and can be normalized with any data value that is available in the dataset. Therefore, *polimaps* is capable of contrasting a prediction visualization with a classic hotspot map [102] (see Figure 5.8 E). Besides hotspot mapping, the heat maps can also be used to visualize information such as the population density or average household income as available from commercial data providers. This task refers to Step 4 of the predictive policing process (Figure 5.1), but also fits the deployment scenario in Step 5.

Concerning usability, an important goal was to keep *polimaps* task specific and not become too broad regarding the program features and the user interface. This goal made sure that both user groups, the analysts from the LEA on state-level, as well as local police forces, can work with *polimaps*. The user-centered design approach of *polimaps* was ensured by involving police officers and data analysts with different backgrounds in the collaboration, and followed the process proposed by Sedlmair et al. [80]. The precondition and core stages were driven by mutual on-site visits and subsequent discussions, sketching ideas and fast feedback rounds with the domain experts. To define the capabilities of *polimaps*, we started to identify the desired visualization techniques and must-have features, such as prediction area visualization or hotspot maps for different data sources, in a half day workshop. Technical details and the actual implementation were worked on off-site, typically in two to three-month cycles. The first, usable version of *polimaps* was available after three months and was subsequently extended and tested in the target environment by the analysts from the LEA. To support the feedback process, we established a build environment that produced over ten months 46 stand-alone, runnable distributions of *polimaps* that were subsequently shared with our collaboration partners.

polimaps contains features from classical geographic information systems, in particular, the visualization of geospatial primitives such as points, lines or multipolygons. Therefore, the main window of the application, as shown in Figure 5.8, resembles such tools, such as QGIS[4] or uDig[5]. The main window is divided into two parts. A sidebar on the left, Figure 5.8 A and B, and a visualization canvas that displays data and contains a number of overlays, Figure 5.8 C D E F.

[4] previously Quantum GIS, `https://www.qgis.org/`

[5] uDig: User-friendly Desktop Internet GIS, `http://udig.refractions.net/`



Selected records: 8,767

CALL_GROUP

DISORDER	2,899	33.07%
NON CRIMINAL/AD...	1,675	19.11%
SUSPICIOUS	1,507	17.19%
PROPERTY CRIME	1,325	15.11%
TRAFFIC	842	9.60%
PERSON CRIME	519	5.92%

Missings: 0, 0.00%

final_case

DISTP	1,206	13.76%
SUSP	637	7.27%
UNWNT	560	6.39%
THEFT	438	5.00%
WELCKP	433	4.94%
WELCK	393	4.48%
NOISE	356	4.06%
AREACK	315	3.59%
ASSIST	298	3.40%
PREMCK	256	2.92%

Missings: 0, 0.00%

CATEGORY

OTHER	6,759	77.10%
STREET CRIMES	1,865	21.27%
MOTOR VEHICLE T...	77	0.88%
BURGLARY	66	0.75%

Figure 5.8: *Main user interface of* polimaps *showing example data ([15]). A and B present the current layers as well as contextual information. C denotes the main visualization canvas, D the current spatial selection for the aggregated information shown in B, E an exemplary heat map with point overlay. F indicates the colors used to indicate point density in E.*

5.3.1 Interactive Visualization

In discussions with the end users, it was always clear that there are two primary data dimensions that the analysts are interested in: space and time. In consequence, the visualization capabilities are tailored to the spatial dimension of the data; the filter facilities as described in Section 5.3.2 give practical support to work with the time dimension as well. Based on discussions with the state-level LEA analysts, *polimaps* supports two fundamentally different kinds of visualizations: spatial interpolation of data, and the visualization of spatial primitives.

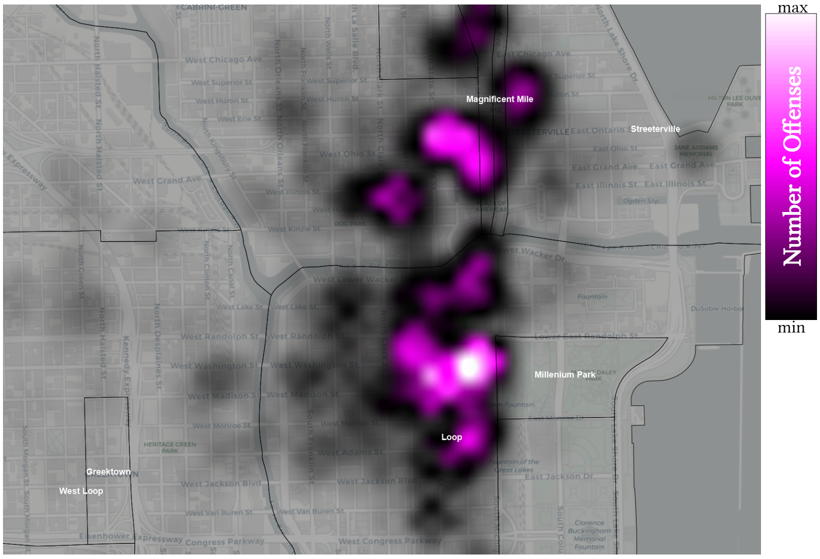


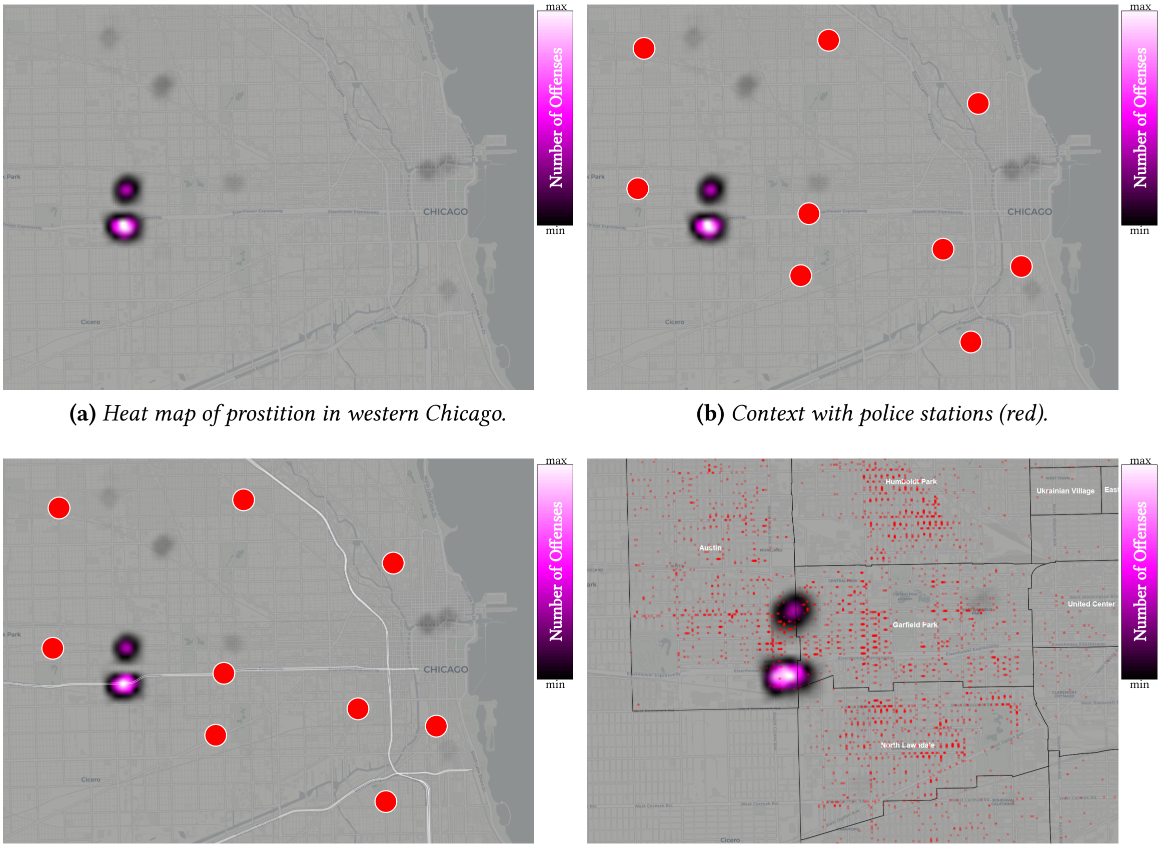
Figure 5.9: *Detail of offenses categorized as thefts from buildings in Chicago, 2017.*[6]

The spatial interpolation is based on an optimized implementation of what is known as *inverse distance weighting* (IDW), as it can be seen in Figure 5.9. More concrete, we utilize a modified form of the Shepard interpolation technique [191], which computes the neighboring cells values in a configurable radius around the current cell. By utilizing an additional spatial index structure, the computational effort is logarithmic, and therefore suitable for interactive data exploration. The data space computation is enhanced with an OpenCL [97] back-end, which is dynamically selected based on the problem size and the available graphics hardware and its capabilities, respectively. To support reasoning about derived measures, the density interpolation can be normalized, e.g., it is possible to visualize the average number of residents per building, while the sociodemographic data on Level 3, see Table 5.2, only provides the number of buildings or the number of residents. The normalization can be configured via a simple graphical user interface. *polimaps* provides many different colormaps to foster the visual interpretation of the data. From a task perspective, the tool is mostly driven by the localization task [115], that is the analysts are searching for specific areas that are of interest along the spatial distribution of data points that are mapped to the colors in

[6] Data from `https://data.cityofchicago.org/`.

the colormap. A classic rainbow colormap with seven unique colors is available, in addition to two sequential colormaps that range from black over cyan to white, while one of those emphasizes the lower half of the data distribution to get a better impression of areas with a comparatively low spatial density.

The visualization of spatial primitives is used to visualize data from geographic data sources directly on the map. As illustrated in Figure 5.10, that capability can be of use to create a rich context for other visualized data, e.g., a density map of offenses as shown in Figure 5.10.



(a) *Heat map of prostition in western Chicago.*

(b) *Context with police stations (red).*

(c) *Context with police stations (red) and main streets (white).*

(d) *Detail of the two hotspots, overlayed with narcotics-related offenses (red).*

Figure 5.10: *Visual Context of offenses categorized as prostitution in Chicago, 2017*[7]*.*

On the western part of Figure 5.10, two distinct hotspots can be observed. Datasets that contain point data can be of use as illustrated in Figure 5.10b, where each red point indicates a police station in Chicago. The visualization indicates that the two prostitution hotspots appear in no direct neighborhood of a police station, which can lead to the impression that a certain distance to police stations is kept on purpose. In Figure 5.10c, the main streets of Chicago are added to the visualization. It gets apparent that the two hotspots appear along the Dwight D. Eisenhower

[7] Data from `https://data.cityofchicago.org/`.

Expressway, which spikes the hypothesis that the high number of prostitution-related offenses (southern hotspot) might be connected with heavy traffic going by. Finally, to follow a known correlation of prostitution with narcotics, the spatial primitive visualization capabilities can be used to add narcotics-related offenses to the visualization of prostitution hotspots, as shown in Figure 5.10d. Red points indicate narcotics-related offenses; their density is mapped to the opacity of the point, which leads to a higher opacity and a deeper red color in areas where more offenses are registered. In the resulting visualization, the correlation between those two types of offenses can be acknowledged, as, in the surroundings of the two hotspots, the red points appear more frequently, and have a high opacity. This resulting visualization has been proven to be useful to build a visual context and the corresponding reasoning task.

A variant from the visualization of spatial primitives is the dedicated prediction visualization facility that requires two different data inputs. First, the prediction areas as they have been introduced in Section 5.2, they serve as a spatial reference which is naturally also the primitive that is visualized. The input comprises of the actual prediction, as it is generated by the machine learning process that is part of step three of the predictive policing process illustrated in Figure 5.1.

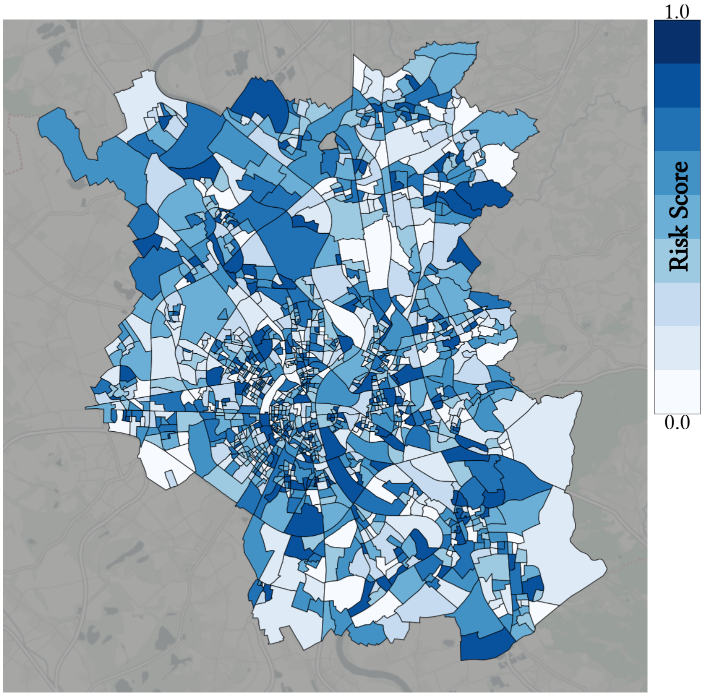
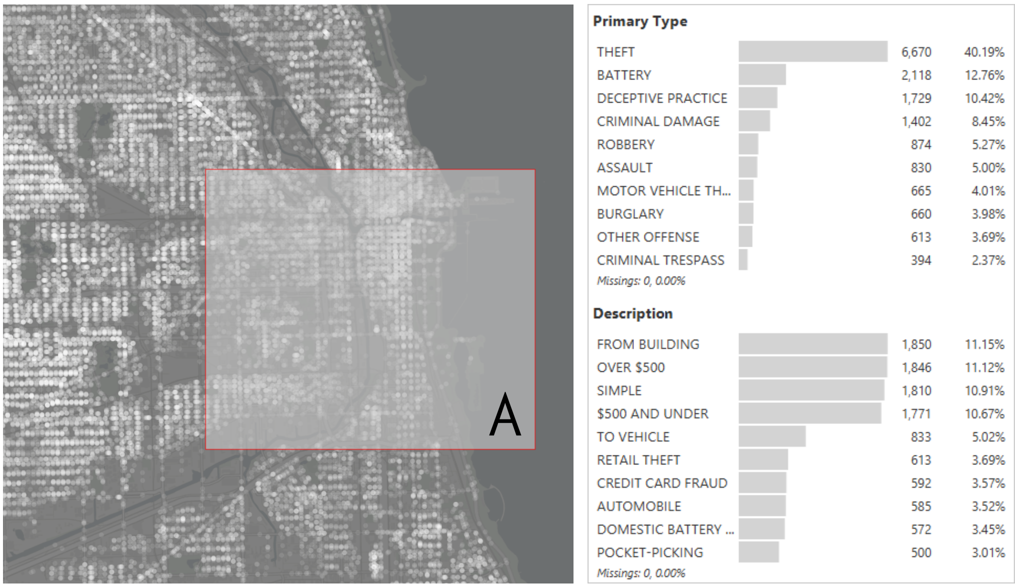


Figure 5.11: *An example of a choropleth map indicating — for this example randomly — computed risk scores.*

The final dataset for the prediction visualization is generated by executing a spatial join of both input datasets, resulting in a complete dataset where each of the prediction areas has another feature, namely the computed risk score. By default, the score is normalized between 0 and 1, where 0 indicates no risk, and 1 the highest risk score for an offense to happen in the prediction period in the corresponding prediction area. The scores are visualized with a choropleth as shown in Figure 5.11, the risk scores are mapped to a sequential, 9 class colormap [142], e.g., ranging from white (0.0) to deep blue (1.0). The analyst can choose from a number of other sequential colormaps, most prominently shades of red from light to dark red. Interaction-wise, the prediction visualization allows the removal of prediction areas from the visualization, as finally the decision about the prediction itself, as well as the deployment is the responsibility of the local police analyst.

All visualizations support zoom and pan, implemented in the fashion as proposed by Wijk and Nuij [146] to foster the exploration of the potentially large, spatial information spaces. Besides zooming and panning, a linked view provides aggregated information of a specific selected area, see Figure 5.12 for example.



Primary Type		
THEFT	6,670	40.19%
BATTERY	2,118	12.76%
DECEPTIVE PRACTICE	1,729	10.42%
CRIMINAL DAMAGE	1,402	8.45%
ROBBERY	874	5.27%
ASSAULT	830	5.00%
MOTOR VEHICLE TH...	665	4.01%
BURGLARY	660	3.98%
OTHER OFFENSE	613	3.69%
CRIMINAL TRESPASS	394	2.37%

Missings: 0, 0.00%

Description		
FROM BUILDING	1,850	11.15%
OVER $500	1,846	11.12%
SIMPLE	1,810	10.91%
$500 AND UNDER	1,771	10.67%
TO VEHICLE	833	5.02%
RETAIL THEFT	613	3.69%
CREDIT CARD FRAUD	592	3.57%
AUTOMOBILE	585	3.52%
DOMESTIC BATTERY ...	572	3.45%
POCKET-PICKING	500	3.01%

Missings: 0, 0.00%

(a) *Chicago city center with offense data from 2017 (white). The gray area* A *indicates the extent of the current aggregation.*

(b) *Aggregated information from the selected area on the map.*

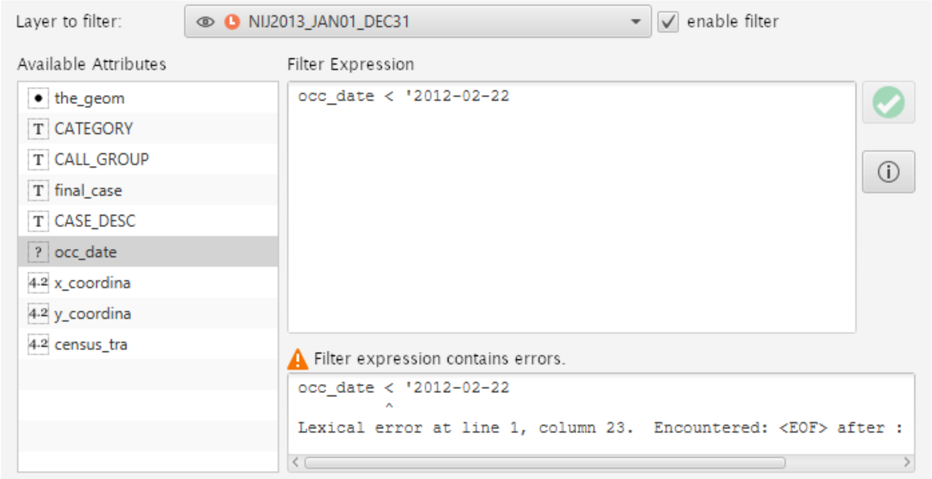
Figure 5.12: *Illustration of the linked aggregation view that provides details of a selected region from the map-based visualization*[8]*.*

Using this on-the-fly aggregation facility, insights about the distribution of feature values, as well as information about missing values, can be retrieved with a simple click and drag interaction. In feedback from the domain experts, this was regarded as a highly useful feature. Additionally, a drill down to the data records is possible, which are presented in a customizable tabular view. The aggregation view is available on all data visualizations.

[8] Data from `https://data.cityofchicago.org/`.

5.3.2 Context Filtering

As the visualization capabilities are tailored to the visualization of spatial feature properties, a temporal dimension had to be integrated into *polimaps*. While this could have been done visually, for example by using a space-time cube [132], we went for a non-integrated approach to not endanger the goal of good usability by increasing visual and cognitive complexity. Instead, the temporal dimension is made explicit by adding a number of corresponding filter clauses that give access to different levels of the temporal hierarchy.


Layer to filter: NIJ2013_JAN01_DEC31 ☑ enable filter

```
occ_date < '2012-02-22
```

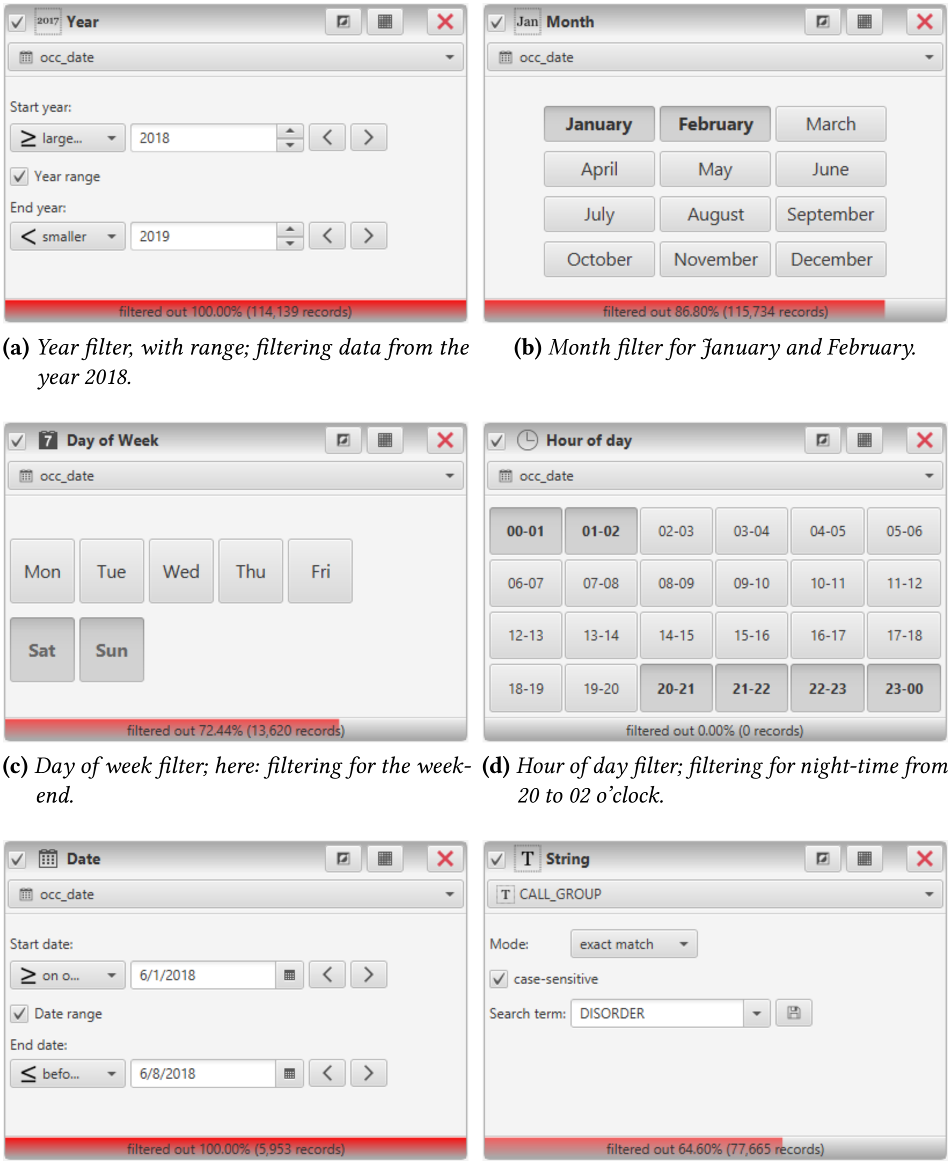
Filter expression contains errors.

```
occ_date < '2012-02-22
          ^
Lexical error at line 1, column 23.  Encountered: <EOF> after :
```

Figure 5.13: *The initial version of the filter interface. On the left, the available features from the selected dataset are shown, on the right filter expressions in CQL (contextual query language) can be composed.*

The integrated filter back-end can be used with queries in Contextual Query Language (CQL) [68], which is known to produce human-readable queries and was, therefore, the natural choice for the query language in general. While we distributed the first build of *polimaps* containing the filter back-end with text-based CQL queries, we found that the target groups had difficulties in creating queries in such a textual query language. Therefore, we started to iterate on the query interface with the analysts from the state level LEA and found, that the textual query interface had to be replaced with control elements that are tailored to the specific filter clauses they represent. Additionally, after discussing the initial filter facility with the domain experts, we introduced two limitations. First, the filter clauses are always concatenated via AND, the OR conjunction is not available anymore. That reduces the flexibility of filters to a great extent, but at the same time lowers the complexity and prevents pitfalls when creating filter expressions considerably. In the improved filter interface, each filter clause is displayed on its own, as a graphical user interface, and as they are all connected via AND conjunctions, all clauses that are visible are applied together.

Second, we limited the filter clauses to a set of seven filters relevant to the work of data analysts when exploring offense data. The filters



☑ 2017 Year

occ_date

Start year: ≥ large... 2018

☑ Year range

End year: < smaller 2019

filtered out 100.00% (114,139 records)

☑ Jan Month

occ_date

filtered out 86.80% (115,734 records)

(a) *Year filter, with range; filtering data from the year 2018.*

(b) *Month filter for January and February.*

☑ Day of Week

occ_date

filtered out 72.44% (13,620 records)

☑ Hour of day

occ_date

filtered out 0.00% (0 records)

(c) *Day of week filter; here: filtering for the weekend.*

(d) *Hour of day filter; filtering for night-time from 20 to 02 o'clock.*

☑ Date

occ_date

Start date: ≥ on o... 6/1/2018

☑ Date range

End date: ≤ befo... 6/8/2018

filtered out 100.00% (5,953 records)

☑ T String

T CALL_GROUP

Mode: exact match

☑ case-sensitive

Search term: DISORDER

filtered out 64.60% (77,665 records)

(e) *Date filter, with range from June 1 to June 8 2018.*

(f) *String filter, filtering for case sensitive match of* DISORDER.

Figure 5.14: *The graphical user interfaces to generate a CQL-based filter.*

Date and Time	
Date	specific date or date ranges
Year	single year and range of years
Month	month of year, set of months
Day of Week	day of week, set of days
Hour of Day	hour of day, set of hours
Feature Value	
Number	specific value or value range, value comparisons
String	specific string, wildcard, like, starts/ends with

Table 5.3: *Overview of the filter categories and corresponding filter instances.*

Each of the filter clauses has a dedicated graphical user interface that is tailored to the filter clause predicate and the corresponding parameters. In Figure 5.15, an overview of the filter user interface is shown. Each filter has a dedicated on/off switch that allows inspecting the impact of a filter clause with a single click. Additionally, each filter can be inverted, which inverts the filter clause and matches all the other data instances when enabled. To communicate that a filter clause is inverted, the invert toggle is not only shown as selected but also highlighted with a red background. Finally, on each of the potentially consecutive filter clauses, a data inspection right after the filter clause has been applied is possible, as each filter control provides a dedicated button to open a tabular data view that shows data instances directly after the filter clause has been applied.

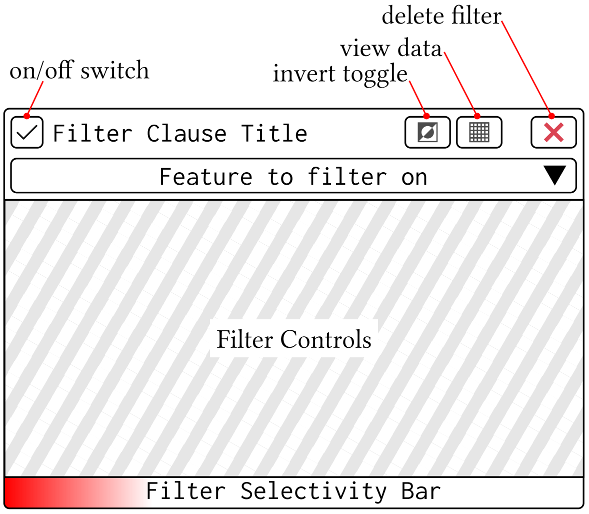


Figure 5.15: *Illustration of the graphical user interface of filters. The* filter controls *area in the middle contains the clause-dependent controls, while the decorations and buttons are the same for all filters.*

This enables the inspection of the impact of single filter clauses when a cascade of filter clauses is configured and enabled. Finally, at the bottom of each graphical filter interface, a selectivity bar indicates the percentages of data records that have been filtered out, and in consequence, are effectively removed from the visualization. The number is mapped to a reddish color that is dark red when all data instances are filtered and is only slightly red when the filter affects any records at all, values in between are interpolated linearly. This makes sure that the analyst is continuously aware of the active filters, as well as their corresponding selectivity.

5.3.3 Process Integration

An important during development of *polimaps* was the integration in the predictive policing process, as introduced in Section 5.1. While it is evident, that the prediction visualization, as well as the visual enrichment of the prediction using further data sources is an integral part, some further aspects make the visual analytics solutions a suitable and versatile part of a predictive policing process. Typically, a number of different tools, and correspondingly also different file formats, are involved in predictive policing process, e.g., a statistical environment such as R[9] for the modeling or more general applications such as Microsoft Excel or Word for various different purposes.

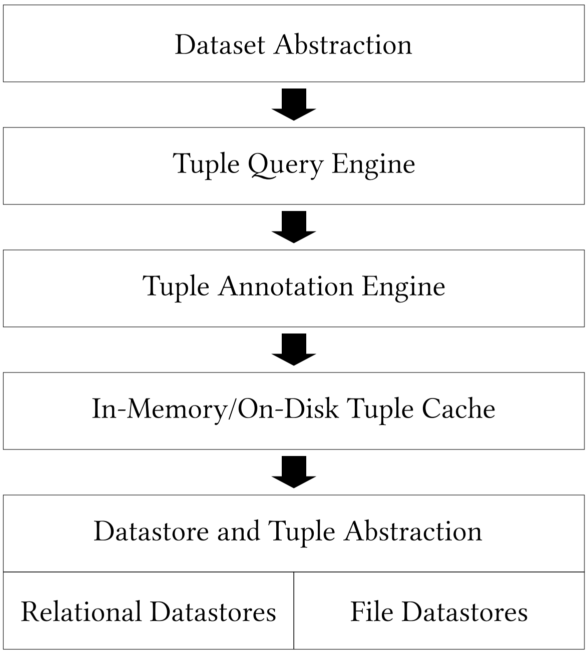


Figure 5.16: *Illustration of the dataset abstraction of* polimaps.

For example, predictions that are computed in Step 2 and Step 3 of the predictive policing process could be exported and stored in proprietary formats, e.g., serialized data frames from R, or exported Excel sheets. To be able to support a variety of different data sources, *polimaps* includes a dataset abstraction, as illustrated in Figure 5.16. The abstraction layers for datastores, which hides the on-disk or relational origin data, e.g., a shapefile or a PostgreSQL server, allows the transparent

[9] R: The R Project for Statistical Computing, `https://www.r-project.org/`

access to all of those different datastores. Additionally, all datastores are supported with transparent caching, tuple annotations, as well as a tuple query facility (the CQL back-end as mentioned in Section 5.3.2). That allows the visualization of prediction areas to retrieve the computed risk scores from a relational database, Microsoft Excel files, or a serialized R data frame. Additionally, there is support for standard text-based data files in read and write fashion, which makes it possible to integrate *polimaps* in existing processes and workflows.

5.3.4 Prediction Deployment

To deploy a prediction, *polimaps* integrates an export functionality. The current view can be exported by selecting a region — or the complete visualization canvas, as indicated in Figure 5.8 C, to be exported. The export is done either as an image or in the form of a styled pdf, as shown in Figure 5.17. The pdf output also provides space for a title and notes, which allows the output to be annotated to indicate interesting findings or other noteworthy elements in the visualization.

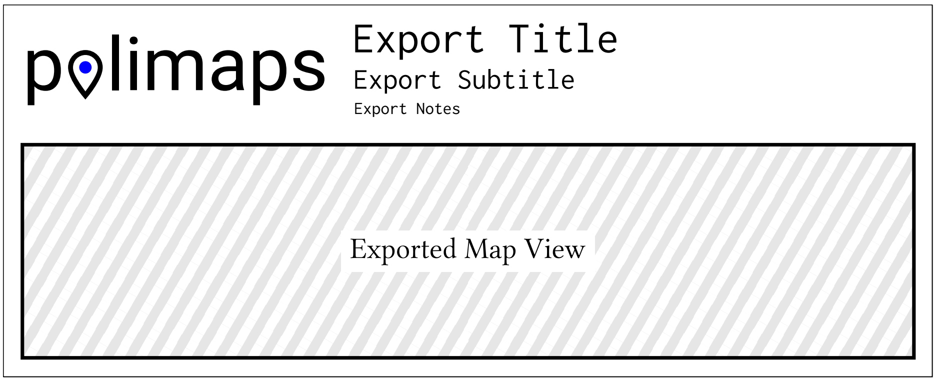
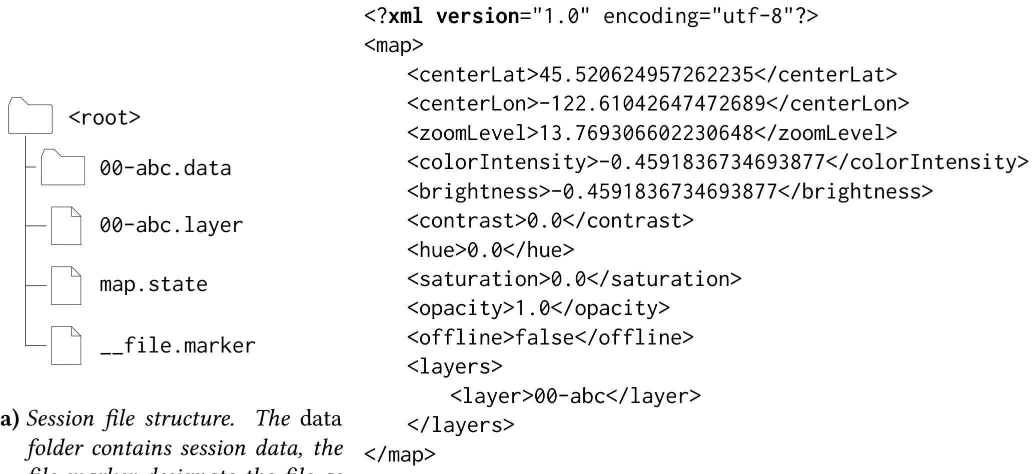


Figure 5.17: *Template for the pdf export. On top, there are meta information, such as a title, subtitle, and further free text notes. On the buttom*

Next to the possibility of exporting the current view, *polimaps* is also capable of exporting the complete state of the application, the current session, to a file.

Additionally, the state of a session can be saved and exported. The save file of a session resembles a classic save feature as it is present in many other applications. The session export stores a session to a single file, including the visualized data next to the state of *polimaps*, see Figure 5.18 for an overview of an exemplary exported session.

Therefore, a complete *polimaps* session, containing the current visualization configuration, filters, and layer settings, can be distributed to other analysts that do not have access to the visualized data. To ensure the integrity of the session data, including the program state, an SHA-256 checksum of the contained data is stored along with the session information. By comparing the checksum of



```
<?xml version="1.0" encoding="utf-8"?>
<map>
    <centerLat>45.520624957262235</centerLat>
    <centerLon>-122.61042647472689</centerLon>
    <zoomLevel>13.769306602230648</zoomLevel>
    <colorIntensity>-0.4591836734693877</colorIntensity>
    <brightness>-0.4591836734693877</brightness>
    <contrast>0.0</contrast>
    <hue>0.0</hue>
    <saturation>0.0</saturation>
    <opacity>1.0</opacity>
    <offline>false</offline>
    <layers>
        <layer>00-abc</layer>
    </layers>
</map>
```

(a) *Session file structure. The* data *folder contains session data, the file marker designate the file as* polimaps *session.*

(b) *Map view state in XML notation, containing the view data, the list of existing layers, as well as color adjustments of the map canvas.*

Figure 5.18: *The session output for deployment.*

the restored data and the checksum created at the time of export, the integrity of the data can be ensured. In case the checksums do not match, an error message pointing to a data integrity issue is shown.

5.3.5 User Feedback

Two months after the final version of *polimaps* has been delivered and used in practice, we handed out a questionnaire with four areas of questions, see the results in Table 5.4. The first area of questions deals with the learnability of *polimaps*, as this was one of the major issues during the user-driven development and iterations (Figure 5.19a). The second group of questions was questioning for usability, e.g., ease of use, the support of tasks and the user interface in general, Figure 5.19a. Both, the first and second group of questions were based on the work from Lewis [170] and Lund [152]. The third area contained questions about the visualization capabilities, parameters, and the linked aggregation view (Figure 5.19c). As the density-based heat map view was the most requested visualization technique by the end users, the last block of questions gathered feedback concerning parameter and color choices, performance and the perceived understanding of the visualization (Figure 5.19d). Details and the composition of questions can be found in Table 5.4. Six users have answered the questionnaire, two data scientists/analysts and four trained police officers.

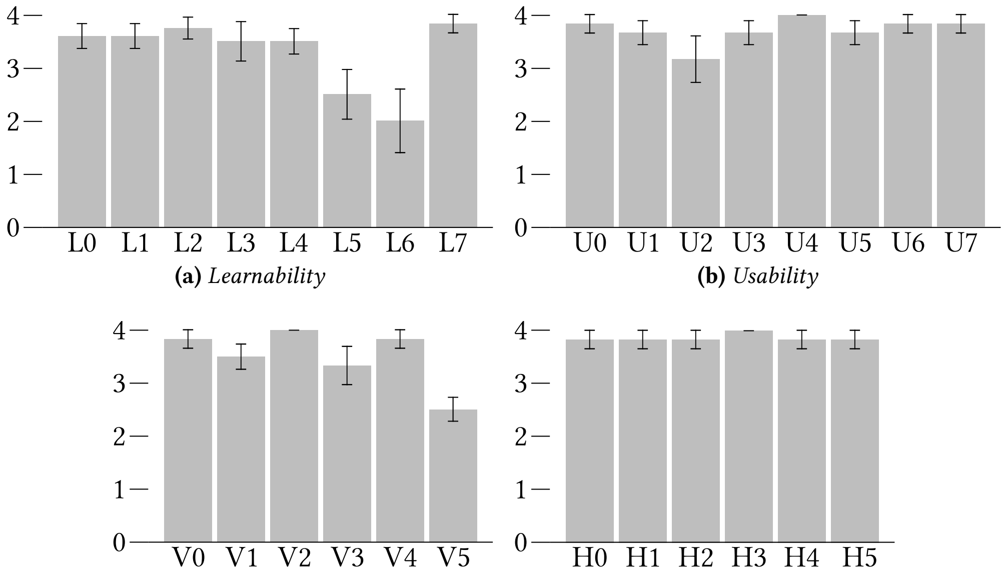
Starting with the obvious in Figure 5.19d, the heat map/density map view is judged as useful, the users have the feeling that they understood the parameters, colormaps, and *polimaps* provides a satisfactory level of performance. Concerning the general visualization techniques (Figure 5.19c),

Learnability			
L0, L1:	Terminology	L4, L5:	Memorability
L2, L3:	Testing progress	L6, L7:	Learning curve
Usability			
U0, U1:	Ease of use	U4, U5:	Data presentation
U2, U3:	Task support	U6, U7:	UI and appearance
General Visualization			
V0, V1:	Adequacy	V4:	Info-Mode
V2, V3:	Parameters	V5:	Completeness
Heat Map Visualization			
H0, H1:	Understanding	H4:	Parameters
H2, H3:	Colormaps	H5:	Performance

Table 5.4: *Overview of the questionnaire consisting of four categories.*

we see potential for further improvement of the linked aggregation view (Info-Mode) (V3), as well as the completeness of the provided techniques (V5). In particular with the completeness, there seems to be the most potential for improvement, as the score of V5 is significantly lower than the others. Similarly, the answers to the usability questions (Figure 5.19b) indicate weaknesses concerning the supported tasks (U2), although the overall judgment in this aspect shows a high level of satisfaction. The most potential for improvement seems to be in the aspect of learnability (Figure 5.19a). There, we see some participants not feeling confident enough to answer the questions (P4, P0, and P1), and despite overall good feedback, some aspects such as the memorability (L5) and the learning curve (L6) need to be improved.

Together with the questionnaire, we asked for textual feedback that each participant filled out. Negative aspects of *polimaps* were: i) no possibility to export shapefiles or any other spatial file format, ii) no editing or annotation feature for the visualized vector data, iii) some error descriptions are not informative enough, and iv) the heatmap re-computations, based on the current zoom level, require a technical understanding that cannot be assumed to be present. Some of the positive aspects mentioned were: i) usability and user-friendliness, ii) speed and reliability of the tool, iii) no expert knowledge is required to visualize geodata, iv) the visualization facility is flexible enough to be useful beyond the prediction and context visualization. From the positive feedback, we can conclude that the major goals have been reached. The tool is perceived as easy to use, requires in some areas no expert knowledge, and is flexible enough to be used beyond the tasks that it has been primarily developed for. On the negative side, the most critical issue seems to be to include an explanation or help concerning the adaptable heat map implementation. Also, the export facility that currently is tailored to produce Microsoft Excel files should be extended, so that *polimaps* integrates even better into existing processes.



(a) *Learnability*

(b) *Usability*

(c) *Visualization*

(d) *Heat Map Visualization*

Figure 5.19: *User feedback. Each plot shows the mean (bar) and the standard deviation (line) for each question.*

5.4 Towards Transparent Quality Metrics

Discussions on the predictive analysis of crimes, also known as predictive policing, are currently dominating criminological and police science literature. Vendors of corresponding software solutions are advertising promising results, and the media covers the topic with varying degrees of depths, diversity, and criticism. Therefore, there are more and more political decisions are made to implement systems for predictive policing. The solutions range from pragmatic, isolated approaches to techniques that have a strong theoretical and scientific foundation. All of the different approaches have in common that they express the quality of their predictions using some form of *hit rates*. The basic idea of a hit rate is to measure, how often an actual offense happened in an area that has been predicted by the predictive policing implementation. Still, the question remains what precisely a hit rate measures, how it should be interpreted, and most importantly whether it is possible to compare different systems that predict crimes based on some hit rates.

This general problem also transfers to the application of *polimaps*. As we have shown in the preceding part of this chapter, the functionality of depicting predictions from a predictive policing system is a welcome addition and received very well. Although, having a look at predictions and their context in space and time, e.g., points of interest or offenses during the last few weeks, information coming from the employed predictive policing methodology could be a noteworthy addition for decision-makers when it comes to judge a prediction or implement it during the next period of the prediction. Therefore, in the following, we will have a closer look at two core properties of quality metrics, such as a hit rate: their variability and validity. To do so, we come back to the predictive policing process introduced in Section 5.1, and discuss its implications on a methodological level. Following, we outline the most prominent techniques to calculate hit rates and give some visual clues about some of their properties. Finally, we discuss implications of the hit rate calculations concerning variability and validity of those quality metrics and implications on tools such as *polimaps*.

In general, the subject of predictive policing systems can be manifold [1, 30]. As our partners from the application domain are predicting the risk of specific areas, introduced previously as *prediction areas*, we concentrate on the same problem in the following: given a spatial partition of a prediction zone, such as a city or a specific city district, in prediction areas, the predictive policing approach computes the likelihood of those areas that offenses of a specific type occur during the prediction period.

In the following, we elaborate on a number of aspects that play an important role when measuring the performance of predictive policing applications.

5.4.1 The Problem of Data Quality

As typical machine learning problems, the application of predictive policing follows a pipeline-like approach where multiple components are working together and built upon each other, which is indicated by the arrows in Figure 5.1.

If the data processed in step 1 is already erroneous, the whole machine learning process already is error prone. In this case, it is also clear that reliability of the process as a whole is in question, which of course also concerns the results, e.g., the output of the prediction computation phase, as well as all the consecutive phases. Therefore, it is also clear that the corresponding quality measures and available metrics are inherently unreliable. Sources for data errors are manifold, e.g., utilizing datasets from data sources that are inadequate as they do not contribute to the problem the machine learning method is applied to. In the predictive policing use case, an inadequate dataset is characterized by a missing a causal relationship to the predicted type of offenses. On the technical side, there are potential sources of data errors, too. For example, having an unspecified, or an incorrectly specified data format may cause incomplete data import or join operations while preparing the dataset for the machine learning application. When dealing with spatial data, incorrect documentation of the spatial reference system of the geographic coordinate could cause errors that are hard to spot in the following phases.

Besides those concrete problems that could happen in the first step of the predictive policing process, the notion of data quality also includes what is subsumed under the term *data uncertainty* [176, 47]. Here, uncertainty plays a huge role when it comes to measuring effects and real-world properties to include in the target dataset that is the output of the first step of the predictive policing process.

For the application of predictive policing, common problems that contribute to uncertainty are for example the time when an offense happened. Having a closer look at domestic burglaries, this exact time of the offense is not known, and in consequence, the time is included as the potential start and end-time of the offense in the databases. Additionally, it is possible that the precision of the satellite-based geocoordinates fluctuates, as it depends on local factors, for example, weather conditions. In particular, for police data, it can be observed that sometimes offense data is unclear regarding its type or the report happens too late so that any specific time reference is inaccurate and contributes to data uncertainty. All of those problems propagate through the model building, the machine learning techniques, and any further visualization or application of the machine learning output.

After the data collection, such problems are nearly impossible to reconstruct or to correct — when they are not an integral part of the data source, for example when the satellite-based geocoordinates are marked with a certainty or uncertainty score that is measured when the coordinates are acquired. In consequence, every quality measure that is based on the output of machine learning processes, even when specified in an objective and transparent way, contains an unclear amount of variance caused by data quality issues.

5.4.2 Correlation vs. Causality

Typically, machine learning methods are independent of the application problem, besides the methodological implications of the problem itself. Machine learning relies on the assumption that the data that is subject to the model building process sufficiently describes the phenomenon that is subject to the model building and training. If this assumption does not hold the resulting model is skewed to the partition of the phenomenon in the training dataset, which must not be revealed by any following model validation or model test phases, as they will be skewed as well. Technically, all of the learning methods model the target variable using statistical inference and include some residual, which is minimized during training, when the variety of the training data covers the variety of the problem sufficiently. Therefore, having large datasets will most likely lead to a good performance of machine learning techniques, in particular when their modeling is not bound to linear approaches, e.g., in the case of neural networks or support vector machines. For criminal offenses, a number of problems can be observed in this regard. First, the amount of data for a specific offense might be limited, e.g., for minor offense types. Additionally, criminal incidents are complex processes that have many influences. Most dominantly, there is space and time, which can be measured for most of the cases. Although, there are effects such as opportunity [183] that are hard to measure at all. Also, coincidences and further random influences may play a role when a crime is committed, which is naturally not measurable. Therefore, for criminal incidents, the available datasets will most likely be incomplete, leading to gaps in the inference, and most likely to degraded model quality. The assumption that in the predictive policing use case causality is most likely not contained in the data that is used to model and predict crimes, further motivates transparent machine learning, eventually leading to sensemaking processes that are aware of the shortcomings described previously. Also, having such a system implemented in productive use, the question about the quality of the machine learning models and the corresponding interventions gains more importance.

5.4.3 Measuring Prediction Quality

When implementing a predictive policing system, and in particular while creating the model that predicts risk scores or the likelihood of an offense at a given point and time, the question of how to judge the model quality arises, not only for the *method-level engineer* but also for the *end user*. As argued before, finding suitable quality metrics is a challenging task as uncertainty is inherent in such models, mainly caused by the data and data related processes, as well as the concrete prediction methodology. Additionally, predicting a rare phenomenon is posing a much bigger challenge, as the generalization that is done in the model that builds the foundation of the machine learning process is potentially incomplete. In consequence, the probability of having an instance of the modeled phenomenon in the future is much higher than of an unknown instance of a frequently occurring problem. Also, one of the objectives of predictive policing is the ability to prevent crimes. That objective contradicts the ability to measure the prediction quality in real-world applications, as the performance of the model is getting distorted by the active prevention of crimes.

In the following, we elaborate on a number of quality metrics and methods to measure the quality of predictive policing implementations.

Hit Rate (HR). The absolute hit rate (accuracy) is the most common quality metric of predictive policing implementations [110, 102, 46, 39]. Hit rates always refer to a timespan t, for example, a day, a week or a month.

$$\mathrm{HR} = \frac{n}{N} \cdot 100 \tag{5.1}$$

In Equation (5.1), n denotes the number of offenses in the prediction areas, and N the total number of registered offenses. The computation of the metric is straightforward, and is easy to comprehend; we argue that the hit rate computation is suitable for a transparent quality assessment in the sense of observability. As the concept is so simple, variants of the original hit rate HR are used, for example:

$$\mathrm{HR} = \frac{a}{A} \cdot 100 \tag{5.2}$$

Equation (5.2) shows the corresponding adoption to prediction areas. a denotes the number of all prediction areas where offenses have been predicted successfully (areas that *have been hit*), A refers to the total number of prediction areas. The hit rate is reducing the phenomena of offenses to a single number, and leaves out fundamental spatial aspects, as they are induced by the prediction areas, for example. Therefore, comparisons of hit rates have to be seen and interpreted with caution.

Predictive Accuracy Index (PAI). To solve the problem, that the hit rate is only referring to the number of prediction areas and does not consider their size, Chainey et al. propose the Predictive Accuracy Index that is calculated as follows:

$$\mathrm{PAI} = \frac{\frac{n}{N} \cdot 100}{\frac{a}{A} \cdot 100} \tag{5.3}$$

In Equation (5.3), n refers to the number of offenses in the prediction areas, N to the total number of prediction areas, a to the area of hit prediction areas, and A to the total area of the prediction areas. The PAI measures the prediction accuracy concerning the prediction area and gives a sense of the spatial extent of the correct predictions. While the PAI can be used to make a comparison of different predictive policing methods, a comparison is only valid if they have the very same spatial partitioning. Otherwise, the foundation of the computations are different and make a comparison of two different methods invalid right from the start.

Standarized Accuracy Efficiency Index (SAEI). The Standardized Accuracy Efficiency Index, see Equation (5.4), is introducing a quality index to measure the achieved efficiency in contrast to what is theoretically possible [40].

$$\begin{aligned} \mathrm{SAEI} &= \frac{\mathrm{AE} - (\mathrm{OE} \cdot \mathrm{A})}{\mathrm{AC}} & & \\ \mathrm{AE} &= \frac{N}{A} & \text{Achieveable Efficiency} & \\ \mathrm{OE} &= \frac{n}{A} & \text{Observed Efficiency} & \\ \mathrm{AC} &= \frac{n}{N} & \text{Accuracy, Hit Rate} & \end{aligned} \tag{5.4}$$

The SAEI is a measure in between of the classic hit rate and the efficiency by means of the prediction area coverage. The variability of the results is highly dependent on the single terms and is therefore hard to compare different time spans or different methodological implementations of predictive policing.

Differences in Case Number. While the previous metrics are computing performance indices, arguing about the quality of a predictive policing implementation can — obviously — also be done by comparing the number of cases before and after a corresponding deployment. While the general idea seems useful and is apparently at heart of what predictive policing is intended to do, it leaves out essential aspects, for example, the number of prediction areas or their spatial extent. As argued before, correlation is not causality, which means that any difference in the number of cases can have a different cause than the application of predictive policing. It is even possible to think of random influences that make it look like an effect from whatever (preventive) methods are applied. Additionally, criminal offenses are typically caused by a number of elements and a potentially complex interplay of them, which is in no way reflected in a simple comparison of two numbers.

Real-world Experiments. As documented in the literature, some experiments are trying to measure the effect of predictive policing based on statistical methods [46, 89]. The area in question is separated into two distinct areas, the control region and the experimental area by random choice. In the experimental area, predictive policing is utilized, while the control region is not subject to any corresponding actions. After a predefined amount of time, indicators of the control and experimental area are compared, for example, the differences in case number, as indicators for the effectiveness of the employed methods. While this looks valid and straightforward in the beginning, there are fundamental problems when dealing with the occurrence of crimes in the way of a controlled lab experiment. It is immediately evident that the interactions between individuals in a city cannot be as controlled as it is possible in an actual lab experiment, which makes the transferability of this technique even more questionable. For example, Boggs showed that the neighborhood has a significant influence on the crimes committed in the surroundings [192]. Murray et al. identified some significant factors contributing to the occurrence of property crimes, e.g., the density of public bus stops, distance to police stations [153]. These factors already differ, as having a city that can be divided into different regions that are entirely similar to the factors mentioned before is impossible.

5.4.4 Discussion

Overall, it is hard to give a sense of how good or bad a predictive policing implementation is performing, in particular as the current toolbox of quality measures has apparent flaws. We showed that common metrics have built-in flaws, e.g., leave out an essential dimension of crime (space), or are by definition not comparable to different methodologies and different runs. Even the practice to conduct controlled lab experiments that is accepted in other critical domains is not suitable due to the dynamics and complexity of the subject of study. Additionally, having a *good* prediction and the ability to enforce the right actions for preventing a crime to happen, the prediction will look bad a posteriori, as there is no way to measure how many offenses have been prevented. This fundamental problem is inherent in all implementations and is getting more severe when the model quality is increasing. In particular, the inability to clearly distinguish between correlation and causality worsens the aspect of quality metrics in this application domain even more. Of course, there is the possibility to apply state of the art to judge the model quality after training, e.g., cross-validation methods, but those concentrate on historical data. While this seems suitable for model building, the quality of a predictive policing system cannot be assessed with historical data, as the influences contributing to criminal incidents happening are diverse enough to keep the residual next to the generalized effects in the machine model potentially large.

To form a truthful and transparent quality metric, we identified three factors that must be included, as they are illustrated in Figure 5.20.

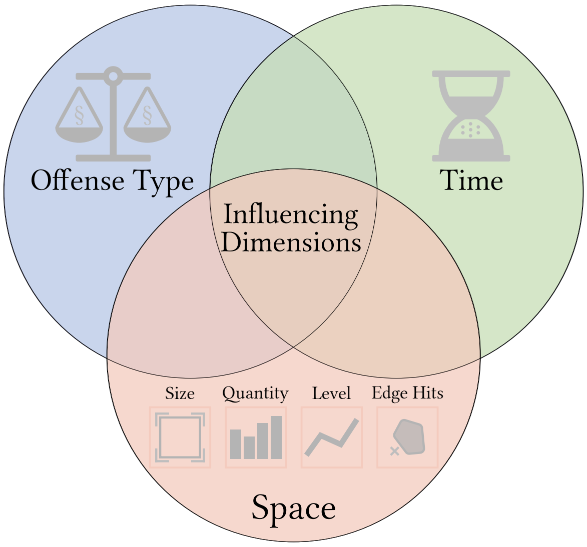


Figure 5.20: *The three influencing factors of quality metric for predictive policing: the offence type, time, and space.*

First, the offense type. From a methodological standpoint, it is clear that a predictive policing implementation should be geared towards a specific offense type. Additionally, if the offense type is static, i.e., varies only very little over space and time, classic hotspot maps would be more appropriate than a potential dynamic machine learning model. The same is true for offenses that are part of what is understood as a victimless offense, which is not registered because of reports from victims but is discovered by actions from the law enforcement agencies, e.g., crimes that involve narcotics. It is clear that offenses that happened in a prediction area but are not part of a prediction model should not be included in a corresponding quality metric. Second, the time factor of a prediction, the prediction period or prediction duration, is an essential factor to incorporate. This because of an empirical correlation of the prediction duration with the number of observed offenses, which means that the longer the time span is, that is part of the quality metric, the *better* the metric gets. In consequence, to be able to compare computed quality indices, they are required to refer to the same prediction duration. The last factor is space and has four contributing elements. i) The prediction area size. It is clear, that the larger the prediction area gets, the higher the probability that an offense happens in that area, and in consequence the higher the hit rate and related measures. The same is true for small prediction areas, which lead to low hit rates. ii) The number of prediction areas. Having a larger number of prediction areas will increase the total area of a prediction — given a fixed area that is considered — and in turn, also leads to higher hit rates. The inverse is true for a small number of prediction areas. iii) The level of crime. When a prediction area describes a partition in space where the level of crimes is generally higher compared to other areas, the likelihood that the prediction area *is hit* with an offense is higher. In consequence, hit rates and related measures are also likely to be high. iv) Edge hits. Having a spatial partition that constitutes the prediction areas, the question arises how to deal with offenses that are right next to a prediction area, e.g., in the order of meters. Besides ignoring them, which is the most rational choice, it is possible to include offenses in a predefined buffer around a prediction area, and weigh them not as high as a *hit* of a prediction area. Still, it is unclear where the buffer size and the offense weight stems from. To create a transparent quality metric for predictive policing poses a huge challenge, which has not been solved so far. All of the utilized experimental methods and existing quality metrics have flaws in specific areas, which are typically ignored, in particular by vendors of commercial predictive policing systems. Therefore, up to now, there is no way to transparently evaluate, judge, and compare different implementations that claim predict criminal offenses. Although, in our point of view, a transparent quality metric is an essential step for the correct assessment of this technology, as well as its correct implementation.

5.5 Summary

In this chapter, we showed different elements contributing to a transparent, i.e., observable, implementation of predictive policing. In Section 5.1, we introduced challenges and tasks and argued about their connections to a user- and task-based transparency as introduced in Chapter 2. More precisely, we developed methods along the predictive policing process (see Figure 5.1), to make some parts of the tasks and the associated processes observable. This includes the generation of prediction areas, which is a crucial part of a predictive policing implementation not relying on regular "prediction boxes", as the prediction areas are the foundation for the data collection, data preprocessing and modeling tasks. In consequence, the prediction areas are also the spatial reference of each prediction, in the described case the risk score, which makes them ubiquitous in all methodical aspects of predictive policing. Our research made it possible to explain the origin of the prediction areas on feature-level, which follows our understanding of transparency.

Additionally, we provided an interactive visualization system that augments the process connected with the utilization of a prediction. Based on density-based visualization, as well as the visualization of spatial primitives, analysts can amplify the utilization of the machine learning output, the predicted risk scores, based on their expert knowledge. This achieves transparency on the data level, meaning that the input data can be reconstructed to build a mental model of the prediction and various data related influences. This enables analysts to develop an understanding, as well as to identify parts of the predictions that are counterintuitive or go beyond what can be considered as reasonable from an expert perspective, which cannot be externalized to the degree that it can be integrated into the computation model. The visual analytics system, *polimaps*, has been a huge success not only for the prediction visualization and deployment but also for geospatial data visualization in general. This illustrates a tremendous potential for interactive visualization in application domains where machine learning plays an important role.

Finally, we illustrated state of the art in the evaluation of the machine learning methods for predictive policing. We argued why the current hit-rate based quality metrics fall short, and which dimensions should be included for a transparent, observable quality assessment of the corresponding methods.

In the foundation of the predictive policing framework introduced at the beginning of this chapter, we achieved some degree of transparency. Most of the motivation to be able to observe the behavior of different methods and tasks was coming out of the initial project setup, which was all about understanding and clarifying what is happening on a methodological level.

	Task	**User Group**	**Transparent**
1.1	Data Selection	LEA Analysts	✓
1.2	Data Collection	LEA Analysts	✓
1.1	Data Preparation	LEA Analysts	✓
2	Predictive Modeling	LEA Analysts	✓
2.1	Prediction Area Computation	LEA Analysts	✓
3	Prediction Computation	LEA Analysts	✓
3.1	Hyperparameters	LEA Analysts	✓
4.1	Prediction Visualization	LEA Analysts, Local Police Forces	✓
4.2	Prediction Area Selection	Local Police Forces	✓
5	Prediction Deployment	Local Police Forces	—
6	Evaluation	LEA Analysts, Local Police Forces	?

Table 5.5: *Tasks and user groups assignments, based on the proposed predictive policing process, as well as the degree of transparency per task. A ✓ indicates achieved transparency, ✓ indicates some degree of transparency, — indicates a varying level that cannot be generalized.*

In Table 5.5, an overview of the involved tasks and the corresponding degree of transparency is shown, based on our experiences and informal assessments of the domain experts. We distinguish between three different levels. Achieved transparency, some transparency and an unclear degree of transparency, which is indicated by ✓, ✓, and — respectively. The crucial components of the prediction process, data, machine learning technique, and hyperparameters were already set up in a transparent way. The analysts of the LEA can observe behavior, and more importantly, determine the actual behavior by their own decisions. Each of the involved methods, e.g., the merging of different data sources or the parameters of the employed decision tree-based prediction model, is visible to the analysts, as well as capable of direct manipulation. Therefore, we argue that the goal of exhibiting transparency — for the analysts working with the corresponding methods — have been achieved, and are indicated with a ✓in Table 5.5. Also, as the creation of the prediction areas happens in a transparent way using the visual analytics approach described in Section 5.2, we claim that the machine learning model is constructed transparently. The degree of transparency is lowered when it comes to the prediction visualization. In Section 5.3 we argued, that domain experts can utilize the data that has been used to train the forecasting methods to create a visual context of the prediction using interactive visual interfaces. A big issue in that respect is the selection of specific datasets that are visualized, which in turn heavily relies on the expert knowledge of the analyst that is utilizing the visual interface — which is without proper externalization of the decision and the way the decision had been made, non-transparent. Our visualization system does not force the users to give a reason when a dataset is loaded, so at this stage, the creation of the visual context of the prediction cannot be described as transparent. Still, as the views can be exported and distributed as part of the prediction deployment, a consensus must exist, as otherwise, the visualization does not exhibit any utility to the police forces. The demand for the visualization solution documented that the opposite is the case. The same argumentation holds for the task of selection prediction areas, which could rely heavily on the expert knowledge of the analysts who are selecting the prediction areas for the local department. From experiences with intelligence data analysts we know, that decisions that influence actual police work must be documented and justifiable by the person that made the corresponding decisions. Therefore, some degree of transparency must exist for the prediction area selection, but not in our sense of observable behavior, which is the same for the prediction deployment task. Finally, transparent evaluation is an important goal, in particular for predictive policing. In this thesis, we aim at methodic and data-based transparency. However, in this application domain, the evaluation should be much more detailed and go beyond methodical and data-related issues. Unfortunately, we showed that even the quality measures based on data are flawed and cannot be regarded as valid (see Section 5.4). We proposed some ideas on how to construct valid quality measures, but leave the concrete realization for further research.

6

Reflections and Concluding Remarks

As stated in the introduction, involving domain experts and their domain knowledge when it comes to designing or interpreting the results was one of the core ideas of this thesis. During this work, a number of different end users, data analysts and domain experts have been involved in different parts of the conducted research. This included in particular personnel from security authorities or law enforcement agencies, which are usually people that are used to explain their decisions and how they arrived at the points that contributed to their actions. While this thesis is based on the assumption that transparency of machine learning processes is a desirable property as it undeniably contributes to comprehensibility, we wanted to ground this assumption with people that are working in critical domains. For example, our collaboration partners in different research projects such as VALCRI[1]. There, we had access to three groups of analysts from different LEAs that were part of the project. We conducted a series of unstructured, informal interviews, each about 45 minutes long, where we wanted to explore the attitude of analysts in the intelligence domain with respect to quality and errors when they are presented with results from machine learning methods, and whether they ask for *some kind of transparency* or, as phrased in this thesis, *observable behavior*. We structured the interview in two parts. First, we wanted to know if the analysts are aware of any quality issues or more generally, any issues in the field of automated support in their daily work. The most significant outcome of this phase of the interview was similar for all three groups. When they encounter an obvious error, e.g., something does not conform to mental models or their previous work experience, the task leading to the error or unexpected outcome is repeatedly executed. First with the same parameters, then possibly with a changed set of hyperparameters. If the result does not change or is not getting better by the experts' subjective assessment, they fall back to methods without automation support, e.g., coding data in Microsoft Excel sheets and using the annotation features of Excel for further analysis. From this observation we conclude, that the

[1] VALCRI — *Visual Analytics for Sensemaking in Criminal Intelligence Analysis* is an EU funded FP7 research project, contract number: FP7-IP-608142.

ultimate solution for something that is incomprehensible is falling back to the expert knowledge and a tedious, manual task of working with data — which is by execution also transparent, as it is observable to the degree the analyst requires.

The second part was a concrete what-if scenario, adjusted to their daily work routine that we asked for in beforehand. There, we wanted to see if any countermeasures or ideas to circumvent reliability and comprehensibility problems exists, and how transparency of the methods could influence their sensemaking or decision-making processes. All experts agreed that something like an observable process is desired, and would augment their decision making, as well as hypothesis testing in the way that they cannot foresee, as besides the research prototype from the VALCRI project no automated data analysis is used, or none of the utilized tools provides similar facilities. Although, we experienced that the technical details to judge what kind of problems can arise, e.g., while analyzing natural language text, were not known. In consequence, it is challenging to ask for such transparent or observable data analysis methods, e.g., some comprehensible documentation of the preprocessing where text data could be changed or removed from the data that is about to be processed.

This insight us the following further directions of research.

6.1 Further Research Directions

During the work that contributed to this thesis and beyond, we identified a number of different research directions that contribute to different aspects of what we understand as transparent machine learning approaches.

Raising Awareness. Besides the technical topics discussed in this section and during the thesis, we tried to push corresponding specialized solutions also in some research projects. While what we illustrated in Chapter 5 worked out quite well — mostly because of the initial project setup was to be able to understand and explain the techniques of predictive policing — we saw in other contexts skepticism from the domain experts. Admittedly, trying to validate what a machine learning system produces regarding it outputs is quite hard, as today even augmenting results with quality information, that we regard as a primitive, highly aggregated form of transparency, is not common. From our experiences we conclude, that there seems to be only little understanding of the fact that machine learning, or how they are sometimes called systems based on *artificial intelligence*, do fail. Such methods are error-prone for several good reasons, most prominently when they are not trained *well enough*. That includes different aspects, such as a sufficient amount of training data, the training data should resemble the real-world problems in all facets, as well as the input data of the machine learning process should fit into the data space of the training data, as this partition of the data has been used to train the machine learning model. These facts are typically not communicated by systems that implement data analysis based on such methods. Therefore, we conclude that awareness for such problems needs to be built to enable users to effectively calibrate their trust on their utilities for automated data analysis. Statements such as "*all quantitative models of language are wrong—but some are useful*" [62] or "*there is no globally best method for automated text analysis*" [62] are very rare, even in scientific literature. In the source of the quotes, the authors Grimmer and Stewart argue about the promises and pitfalls of automated content analysis for text data and motivate an application where the automated analysis heavily relies on machine learning and is crucial for the application related problems. Still, those opinions are rare in literature and almost non-existent in real-world which is why we conclude, that raising awareness is key, in particular when techniques from the machine learning domain are getting more widely used in a wide variety of application domains. Automated data analysis will get even more important than it is today for a number of reasons, e.g., the increasing amount of data that has to be analyzed. In this aspects, humans are limited and need assistance from intelligent data analysis techniques based on machine learning. Still, we agree that there must be a good motivation to spend time with methods that provide insights into the automated data analysis methods, e.g., by the idea of exhibiting *observable behavior*, even when they are communicated by means of visualization and are augmented by the powerful human perception and cognition. In our point of view, understandably communicating the described issues is critical to pave the way for any time that is spent in validating what an automated, black box, machine learning-based system produces — in particular in critical domains where decisions have a wide-ranging consequence on society or humans.

Record Behavior to Observe. In one of the research project we worked on, we implemented a data provenance facility for natural language processing. The provenance component acts as a proxy in between method calls of any component that processes input data and emits outputs, which are reused by other components in the processing pipeline. The concrete provenance information was consisting of two major parts. First, the configuration and hyperparameters of the employed data processing and machine learning components. This includes detailed information about pre-trained models as well as the concrete parameter name and value combinations. Additionally, the input and output data of each component is compared using the Myers diff algorithm [180]. Using this technique makes it possible to follow all changes of the dataset, as it happens during preprocessing or cleanup stages in natural language processing, along with the information of what component with which set of parameters was responsible for the corresponding changes. The result was what we called the *modification trail*, which proved to be hard to maintain in near-real-world applications for reasons of computation time and memory consumption. The technical framework spiked some interest from law end ethics people, although it was not enough to get a movement going and continue to work in the provenance for transparency field. Unfortunately, in the research project where these developments happened, there was no effort planned to join forces from technical and law/ethics/privacy people to follow that route. Still, we are convinced that data provenance can be a useful vehicle to transport transparency and make the behavior of automated data analysis

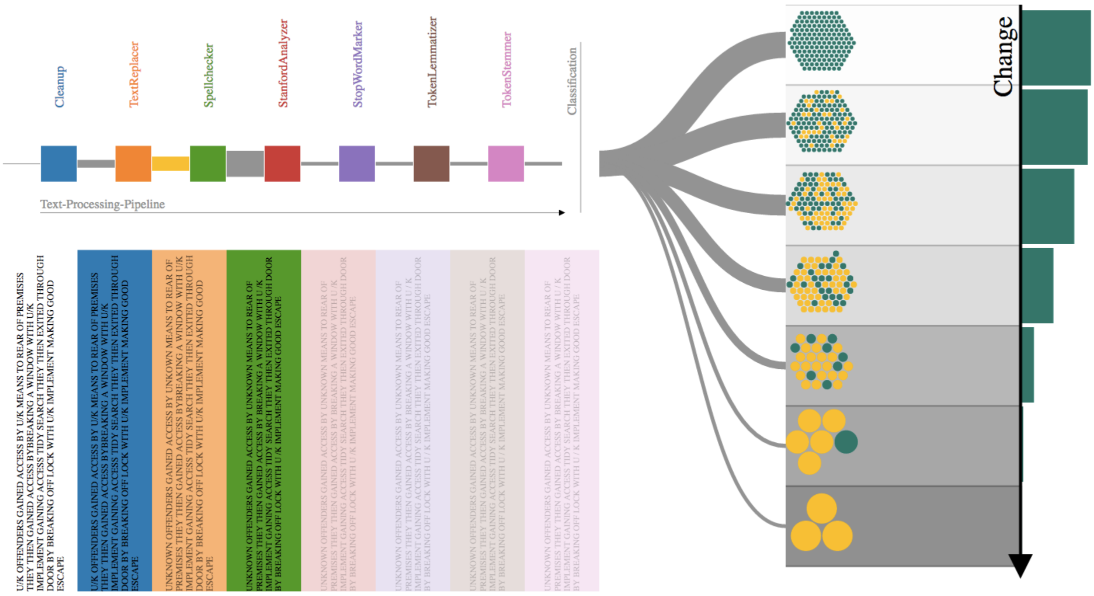

Cleanup
TextReplacer
Spellchecker
StanfordAnalyzer
StopWordMarker
TokenLemmatizer
TokenStemmer
Classification
Text-Processing-Pipeline
U/K OFFENDERS GAINED ACCESS BY U/K MEANS TO REAR OF PREMISES THEY THEN GAINED ACCESS BYBREAKING A WINDOW WITH U/K IMPLEMENT GAINING ACCESS TIDY SEARCH THEY THEN EXITED THROUGH DOOR BY BREAKING OFF LOCK WITH U/K IMPLEMENT MAKING GOOD ESCAPE
U/K OFFENDERS GAINED ACCESS BY U/K MEANS TO REAR OF PREMISES THEY THEN GAINED ACCESS BYBREAKING A WINDOW WITH U/K IMPLEMENT GAINING ACCESS TIDY SEARCH THEY THEN EXITED THROUGH DOOR BY BREAKING OFF LOCK WITH U/K IMPLEMENT MAKING GOOD ESCAPE
UNKNOWN OFFENDERS GAINED ACCESS BY UNKOWN MEANS TO REAR OF PREMISES THEY THEN GAINED ACCESS BYBREAKING A WINDOW WITH U/K IMPLEMENT GAINING ACCESS TIDY SEARCH THEY THEN EXITED THROUGH DOOR BY BREAKING OFF LOCK WITH U/K IMPLEMENT MAKING GOOD ESCAPE
UNKNOWN OFFENDERS GAINED ACCESS BY UNKNOWN MEANS TO REAR OF PREMISES THEY THEN GAINED ACCESS BY BREAKING A WINDOW WITH U /K IMPLEMENT GAINING ACCESS TIDY SEARCH THEY THEN EXITED THROUGH DOOR BY BREAKING OFF LOCK WITH U /K IMPLEMENT MAKING GOOD ESCAPE
Change

Figure 6.1: Sketch of an interactive text processing pipeline observer.

transparent. In particular in the natural language processing domain, where questions about the data preprocessing and corresponding modifications will arise — once it is known what happens to the data before it is analyzed, which refers to the awareness problem that we identified. Using data provenance, this question can be answered with the processed data, without requiring any technical or methodological understanding from the audience.

Making behavior observable. Having recorded information that describes the behavior of a machine learning method is the first step to make it observable. In Section 2.3, we described the initial idea of utilizing interactive data visualization to create abstract views on the data that exposes behavior.

Figure 6.1 depicts a prototype of a visual, interactive interface that shows recorded behavior of a natural language text analysis pipeline. On the top left, the singe components of the analysis pipeline are shown, the thickness of each connection between the components indicates how many data instances are getting modified from the component from left to right. The documents are shown on the bottom left area. The background of each document indicates where in the pipeline (above) the current document is located and therefore gives insights into the progress of the current processing. On the right, a binned view shows seven different classes of modifications. From top to bottom, the amount of modifications, e.g., by a preprocessing step or a dictionary-based expansion component, is illustrated. To convey a sense of how much documents each bin represents, each document is represented by a single circle in the bin, and a vertically aligned bar chart is shown on the right of the bins. This simple interface illustrates effectively how many documents have been modified to what extent, and therefore makes it explicit what the data preprocessing facilities of the natural language processing components do.

Still, having so many details about the actual machine learning processes and documents in the visualization makes this interface not suitable for domain experts. Therefore, we think it is crucial to find a level of abstraction where the recorded behavior is conveyed as effective as possible and is suitable to be integrated into the application tasks. Starting with the data shown in Figure 6.1, a complete view on the problem with as many details as the example exhibits is apparently targeted at an analyst who is either working with the text processing pipeline, e.g., by modifying or adding/removing components, or a very interested end-user audience which is aware of some of the technical background. For an end user who is just interested in how many data instances have been modified, maybe a simple number, or the histogram on the right of Figure 6.1 can be already enough to convey a sense of caution when working with a system and methods that are based on data that is in, say, 50% of all instances heavily modified.

In that respect, the task also plays an important role. While it could be possible that an analyst is interested in the inner workings of a machine learning process at the detail level as it is illustrated in Figure 6.1, this kind of visualization is perfectly fine. On the other hand, this amount, and more importantly, this kind of information is not required if the application problem requires the analyst to get an idea if the machine learning system should be used at all, based on the amount of modified data instances. In that case, a simple number that gives the percentage of documents that have been changed by the data analytics process is enough to satisfy the corresponding information need.

Another important issue when it comes to conveying the observed behavior is the time an analyst is able and willing to spend on the inspection of a transparent machine learning system. This constraint also heavily depends on the task and the application problem, but also the level of experience and prior knowledge an analyst has. Also, there could be hard limits regarding the available time for such a meta-analysis besides solving the application problem, which can be higher or lower for different varying reasons.

Therefore, we see the problem of making behavior as an open research problem, that is not only complicated by the machine learning methodology, but also by other issues that cannot be foreseen during the design time of an interactive, visual interface that is perfectly capable of conveying the required information.

6.2 Concluding Remarks

This thesis shed light on the emerging topic of transparency of machine learning processes. A high-level overview of the concept was introduced in Chapter 1, motivating the understanding of transparency as *behavior* of processes that need to be *observable*. In the following chapters, we outlined challenges of providing observable behavior in different aspects and presented concrete solutions based on systematic analysis of errors, the interactive visual exploration of ambiguities of feature sets, as well as purely visualization-driven visual context of machine learning results. Due to the complexity of the topic, identifying the corresponding challenges, propose solutions and bring them into practice is — still and despite the omnipresent machine learning applications — a huge challenge. We concentrated explicitly on feature-based machine learning techniques, as this gave the flexibility to conduct what has been presented in this work without referring to the machine learning back-end, which keeps this work general. It is clear, that for even more complex methodologies, e.g., neural networks with autoencoders the proposed methodology needs to be adapted and revised to a certain extent. Still, the major contributions of this thesis are the identification and concrete solutions towards understandable, observable behavior of machine learning processes based on their feature set, and a motivation and a high-level framework for research in the corresponding problem domain.

Chapter 3 introduces *error analysis for supervised learning* as a proxy for observable behavior in supervised classification tasks. With our novel approach we illustrated, that visualization, and in particular interactive what-if analysis coupled with data visualizations, opens new ways to understand the behavior of an underlying machine learning techniques. Ultimately, we were able to improve the performance of the classifier while the approach is completely agnostic in terms of the underlying data analysis and machine learning techniques. The following chapter introduces a technique that tackles the problem of ambiguous feature sets, as they can occur in the domain of natural language processing. We coupled statistics and semantics in an interactive, visual interface that serves two purposes. First, it visualizes the feature set and conveys an overview of the features very effectively. Findings made using that statistics view can be verified with semantic information, that utilizes the lower half of the symmetric matrix that serves as the main visualization interface, and vice versa. Second, it enables the generation of rules, postprocessing instructions, and gives a perspective for a direct feedback loop to the machine learning backed, to resolve ambiguous feature sets.

While the previous approaches are targeted at users that have a technical background and interests in the inner workings of a machine learning process, the work presented in Chapter 5 switches perspective. It is part of a real-world research project and promotes the idea of observable behavior to a large user group. We demonstrated how parts of the machine learning data can be generated in an observable manner, and how interactive data visualization can be utilized for sense-making and reasoning with respect to machine learning outputs. Additionally, the question about evaluation and quality metrics of machine learning, and their connections to a transparent machine learning process were motivated with data and modeling-related challenges that we encountered in the application area of predictive policing. At the time of writing, we see that even in application domains where transparency is a major issue, raising awareness for future problems arising with non-transparent machine learning techniques is a challenging problem. Additionally, we outline concrete problems to work on that contribute to a transparent, observable machine learning process.

In the beginning, we motivated why transparency in machine learning is a desirable goal and elaborated on different components that can be researched to target concrete solutions. In this thesis, we presented different approaches with respect to observable behavior. What we presented is connected to the idea that data, and in particular as we deal with machine learning, features that contribute to the modeling, or are created by a machine learning process can concur to an observable behavior of an automated data analysis process. Although, the number of open challenges and the number of dependencies, e.g., on the implementation of machine learning techniques, learning techniques, and future developments in the area of machine learning leaves many problems for further research.

List of Figures

List of Tables

Bibliography

[1] Berk Richard, Sherman Lawrence, Barnes Geoffrey, Kurtz Ellen, and Ahlman Lindsay. "Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 172.1 (), pp. 191–211. DOI: 10.1111/j.1467-985X.2008.00556.x.

[2] Azavea. *HunchLab - Next Generation Predictive Policing Software*. https://www.hunchlab.com/. accessed January 21, 2018. 2018.

[3] Lucie Flekova, Florian Stoffel, Iryna Gurevych, and Daniel Keim. "Content-based Analysis and Visualization of Story Complexity". In: *Visualisierung sprachlicher Daten*. Ed. by Noah Bubenhofer and Kupietz Marc. Heidelberg University Publishing, 2018. Chap. 7, pp. 185–223. ISBN: 978-3-946054-75-7. DOI: 10.17885/heiup.345.474.

[4] Wolfgang Jentner, Dominik Sacha, Florian Stoffel, Geoffrey Ellis, Leishi Zhang, and Daniel A. Keim. "Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool". In: *The Visual Computer* (Feb. 2018). ISSN: 1432-2315. DOI: 10.1007/s00371-018-1483-0.

[5] Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng (Polo) Chau. "ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models". In: *IEEE Trans. Vis. Comput. Graph.* 24.1 (2018), pp. 88–97. DOI: 10.1109/TVCG.2017.2744718.

[6] Mengchen Liu, Jiaxin Shi, Kelei Cao, Jun Zhu, and Shixia Liu. "Analyzing the Training Processes of Deep Generative Models". In: *IEEE Trans. Vis. Comput. Graph.* 24.1 (2018), pp. 77–87. DOI: 10.1109/TVCG.2017.2744938.

[7] Nicola Pezzotti, Thomas Höllt, Jan Van Gemert, Boudewijn P. F. Lelieveldt, Elmar Eisemann, and Anna Vilanova. "DeepEyes: Progressive Visual Analytics for Designing Deep Neural Networks". In: *IEEE Trans. Vis. Comput. Graph.* 24.1 (2018), pp. 98–108. DOI: 10.1109/TVCG.2017.2744358.

[8] PREDPOL. *From Theory to Practical Deployment — The Science Behind PredPol*. Tech. rep. accessed May 21, 2018. PREDPOL, 2018.

[9] Predpol. *Predictive Policing Software | PredPol*. https://www.hunchlab.com/. accessed January 21, 2018. 2018.

[10] Florian Stoffel, Hanna Post, Marcus Stewen, and Daniel A. Keim. “polimaps: Supporting Predictive Policing with Visual Analytics”. In: *EuroVis Workshop on Visual Analytics (EuroVA).* Ed. by Christian Tominski and Tatiana von Landesberger. The Eurographics Association, 2018. ISBN: 978-3-03868-064-2. DOI: `10.2312/eurova.20181111`.

[11] Ralf Herbrich. “Machine Learning at Amazon”. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017.* Ed. by Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang. ACM, 2017, p. 535. ISBN: 978-1-4503-4675-7. DOI: `10.1145/3018661`.

[12] Andrej Karpathy and Li Fei-Fei. “Deep Visual-Semantic Alignments for Generating Image Descriptions”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.4 (2017), pp. 664–676. DOI: `10.1109/TPAMI.2016.2598339`.

[13] Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. “Towards Better Analysis of Deep Convolutional Neural Networks”. In: *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), pp. 91–100. DOI: `10.1109/TVCG.2016.2598831`.

[14] Justin Matejka and George W. Fitzmaurice. “Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017.* Ed. by Gloria Mark, Susan R. Fussell, Cliff Lampe, m. c. schraefel, Juan Pablo Hourcade, Caroline Appert, and Daniel Wigdor. ACM, 2017, pp. 1290–1294. ISBN: 978-1-4503-4655-9. DOI: `10.1145/3025453.3025912`.

[15] National Institute of Justice. *Crime Forecasting Challenge.* `https://www.nij.gov/funding/Pages/fy16-crime-forecasting-challenge-document.aspx`. accessed January 21, 2018. 2017.

[16] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org.* `https://www.openstreetmap.org`. 2017.

[17] Daniela Pollich and Felix Bode. “Predictive Policing — Zur Bedeutung eines (sozial-) wissenschaftlich geleiteten Vorgehens”. In: *Polizei und Wissenschaft* 3 (Sept. 2017), pp. 2–12.

[18] Paulo E. Rauber, Samuel G. Fadel, Alexandre X. Falcão, and Alexandru C. Telea. “Visualizing the Hidden Activity of Artificial Neural Networks”. In: *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), pp. 101–110. DOI: `10.1109/TVCG.2016.2598838`.

[19] Dominik Sacha, Wolfgang Jentner, Leishi Zhang, Florian Stoffel, and Geoffrey Ellis. “Visual Comparative Case Analytics”. In: *EuroVis Workshop on Visual Analytics (EuroVA).* Ed. by Michael Sedlmair and Christian Tominski. The Eurographics Association, 2017. ISBN: 978-3-03868-042-0. DOI: `10.2312/eurova.20171119`.

[20] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John Aldo Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. “What you see is what you can change: Human-centered machine learning by interactive visualization”. In: *Neurocomputing* 268 (2017), pp. 164–175. DOI: `10.1016/j.neucom.2017.01.105`.

[21] Qiaomu Shen, Tongshuang Wu, Haiyan Yang, Yanhong Wu, Huamin Qu, and Weiwei Cui. “NameClarifier: A Visual Analytics System for Author Name Disambiguation”. In: *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), pp. 141–150. DOI: `10.1109/TVCG.2016.2598465`.

[22] Florian Stoffel, Felix Bode, and Daniel A. Keim. “Qualitätsmetriken im Bereich Predictive Policing: Die Variabilität und Validität von Trefferraten”. In: *Polizei & Wissenschaft* 4 (2017). Ed. by Clemens Lorei, pp. 2–15. ISSN: 1439-7404.

[23] Florian Stoffel, Wolfgang Jentner, Michael Behrisch, Johannes Fuchs, and Daniel A. Keim. “Interactive Ambiguity Resolution of Named Entities in Fictional Literature”. In: *Computer Graphics Forum* 36.3 (2017), pp. 189–200. ISSN: 1467-8659. DOI: `10.1111/cgf.13179`.

[24] Zbigniew Wojna, Alexander N. Gorban, Dar-Shyang Lee, Kevin Murphy, Qian Yu, Yeqing Li, and Julian Ibarz. “Attention-Based Extraction of Structured Information from Street View Imagery”. In: *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017.* IEEE, 2017, pp. 844–850. ISBN: 978-1-5386-3586-5. DOI: `10.1109/ICDAR.2017.143`.

[25] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. “Matrix Reordering Methods for Table and Network Visualization”. In: *Comput. Graph. Forum* 35.3 (2016), pp. 693–716. DOI: `10.1111/cgf.12935`.

[26] Mennatallah El-Assady, Valentin Gold, Carmela Acevedo, Christopher Collins, and Daniel A. Keim. “ConToVi: Multi-Party Conversation Exploration using Topic-Space Views”. In: *Comput. Graph. Forum* 35.3 (2016), pp. 431–440. DOI: `10.1111/cgf.12919`.

[27] Carlos A. Gomez-Uribe and Neil Hunt. “The Netflix Recommender System: Algorithms, Business Value, and Innovation”. In: *ACM Trans. Management Inf. Syst.* 6.4 (2016), 13:1–13:19. DOI: `10.1145/2843948`.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016.* 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[29] Wolfgang Jentner, Geoffrey Ellis, Florian Stoffel, Dominik Sacha, and Daniel A. Keim. “A Visual Analytics Approach for Crime Signature Generation and Exploration”. In: *The Event Event: Temporal & Sequential Event Analysis, IEEE VIS 2016 Workshop.* 2016.

[30] Jessica Saunders, Priscillia Hunt, and John S. Hollywood. “Predictions put into practice: a quasi-experimental evaluation of Chicago’s predictive policing pilot”. In: *Journal of Experimental Criminology* 12.3 (Sept. 2016), pp. 347–371. ISSN: 1572-8315. DOI: 10.1007/s11292-016-9272-0.

[31] Leishi Zhang, Chris Rooney, Lev Nachmanson, B. L. William Wong, Bum Chul Kwon, Florian Stoffel, Michael Hund, Nadeem Qazi, Uchit Singh, and Daniel A. Keim. “Spherical Similarity Explorer for Comparative Case Analysis”. In: *Visualization and Data Analysis 2016, San Francisco, California, USA, February 14-18, 2016.* Ed. by David Kao, Thomas Wischgoll, and Song Zhang. Ingenta, 2016, pp. 1–10.

[32] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control (Wiley Series in Probability and Statistics).* Wiley, 2015. ISBN: 1118675029.

[33] Lucie Flekova and Iryna Gurevych. “Personality Profiling of Fictional Characters using Sense-Level Links between Lexical Resources”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015.* 2015, pp. 1805–1816.

[34] Valentin Gold, Christian Rohrdantz, and Mennatallah El-Assady. “Exploratory Text Analysis using Lexical Episode Plots”. In: *Eurographics Conference on Visualization (EuroVis) - Short Papers.* Ed. by E. Bertini, J. Kennedy, and E. Puppo. The Eurographics Association, 2015. DOI: 10.2312/eurovisshort.20151130.

[35] HunchLab. *HunchLab: Under the Hood.* Tech. rep. accessed May 21, 2018. HunchLab, 2015.

[36] Bing Liu. *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions.* Cambridge University Press, 2015. ISBN: 978-1-10-701789-4.

[37] Xiaotong Liu and Han-Wei Shen. “The Effects of Representation and Juxtaposition on Graphical Perception of Matrix Visualization”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015.* 2015, pp. 269–278. DOI: 10.1145/2702123.2702217.

[38] Jonas Lukasczyk, Ross Maciejewski, Christoph Garth, and Hans Hagen. “Understanding hotspots: a topological visual analytics approach”. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, November 3-6, 2015.* 2015, 36:1–36:10. DOI: 10.1145/2820783.2820817.

[39] G. O. Mohler, M. B. Short, Sean Malinowski, Mark Johnson, G. E. Tita, Andrea L. Bertozzi, and P. J. Brantingham. “Randomized Controlled Field Trials of Predictive Policing”. In: *Journal of the American Statistical Association* 110.512 (2015), pp. 1399–1411. DOI: 10.1080/01621459.2015.1077710.

[40] Motorola. *Predictive Analytics vs. Hotspotting.* Tech. rep. Motorola Solutions, 2015.

[41] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003.

[42] Florian Stoffel, Lucie Flekova, Daniela Oelke, Iryna Gurevych, and Daniel A. Keim. “Feature-Based Visual Exploration of Text Classification”. In: *Symposium on Visualization in Data Science (VDS) at IEEE VIS*. 2015.

[43] Florian Stoffel, Dominik Sacha, Geoffrey Ellis, and Daniel A. Keim. “VAPD - A Visionary System for Uncertainty Aware Decision Making in Crime Analysis”. In: *Symposium on Visualization for Decision Making Under Uncertainty at IEEE VIS 2015*. 2015.

[44] Hardik Vala, David Jurgens, Andrew Piper, and Derek Ruths. “Mr. Bennet, his coachman, and the Archbishop walk into a bar but only one of them gets recognized: On The Difficulty of Detecting Characters in Literary Texts”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 2015, pp. 769–774.

[45] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *CoRR* abs/1409.0473 (2014). arXiv: 1409.0473.

[46] Priscillia Hunt, Jessica Saunders, and John S. Hollywood. *Evaluation of the Shreveport Predictive Policing Experiment*. RAND Corporation, 2014. ISBN: 9780833086914.

[47] Christoph Kinkeldey, Alan M. MacEachren, and Jochen Schiewe. “How to Assess Visual Communication of Uncertainty? A Systematic Review of Geospatial Uncertainty Visualisation User Studies”. In: *The Cartographic Journal* 51.4 (2014), pp. 372–386. DOI: 10.1179/1743277414Y.0000000099.

[48] Josua Krause, Adam Perer, and Enrico Bertini. “INFUSE: Interactive Feature Selection for Predictive Modeling of High Dimensional Data”. In: *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), pp. 1614–1623. DOI: 10.1109/TVCG.2014.2346482.

[49] Aibek Makazhanov, Denilson Barbosa, and Grzegorz Kondrak. “Extracting Family Relationship Networks from Novels”. In: *CoRR* abs/1405.0603 (2014).

[50] Abish Malik, Ross Maciejewski, Sherry Towers, Sean McCullough, and David S. Ebert. “Proactive Spatiotemporal Resource Allocation and Predictive Visual Analytics for Community Policing and Law Enforcement”. In: *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), pp. 1863–1872. DOI: 10.1109/TVCG.2014.2346926.

[51] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. “The Stanford CoreNLP Natural Language Processing Toolkit”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, System Demonstrations*. 2014, pp. 55–60.

[52] Andrea Moro, Alessandro Raganato, and Roberto Navigli. “Entity Linking meets Word Sense Disambiguation: a Unified Approach”. In: *TACL* 2 (2014), pp. 231–244.

[53] Tamara Munzner. *Visualization Analysis and Design.* A.K. Peters visualization series. A K Peters, 2014. ISBN: 978-1-466-50891-0.

[54] Arvind Neelakantan and Michael Collins. "Learning Dictionaries for Named Entity Recognition using Minimal Supervision". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden.* 2014, pp. 452–461.

[55] Nikolaj Jang Lee Linding Pedersen, Kristoffer Ahlström-Vij, and Klemens Kappel. "Rational trust". In: *Synthese* 191.9 (2014), pp. 1953–1955. DOI: 10.1007/s11229-014-0451-0.

[56] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey P. Ellis, and Daniel A. Keim. "Knowledge Generation Model for Visual Analytics". In: *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), pp. 1604–1613. DOI: 10.1109/TVCG.2014.2346481.

[57] Florian Stoffel and Fabian Fischer. "Using a knowledge graph data structure to analyze text documents (VAST challenge 2014 MC1)". In: *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014, Paris, France, October 25-31, 2014.* Ed. by Min Chen, David S. Ebert, and Chris North. IEEE Computer Society, 2014, pp. 331–332. ISBN: 978-1-4799-6227-3. DOI: 10.1109/VAST.2014.7042551.

[58] Salvatore Trani, Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. "Manual Annotation of Semi-Structured Documents for Entity-Linking". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014.* 2014, pp. 2075–2077. DOI: 10.1145/2661829.2661854.

[59] Ewart J. de Visser, Marvin S. Cohen, Amos Freedy, and Raja Parasuraman. "A Design Methodology for Trust Cue Calibration in Cognitive Agents". In: *Virtual, Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments - 6th International Conference, VAMR 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014, Proceedings, Part I.* Ed. by Randall Shumaker and Stephanie J. Lackey. Vol. 8525. Lecture Notes in Computer Science. Springer, 2014, pp. 251–262. ISBN: 978-3-319-07457-3. DOI: 10.1007/978-3-319-07458-0_24.

[60] Hazeline U. Asuncion. "Automated data provenance capture in spreadsheets, with case studies". In: *Future Generation Comp. Syst.* 29.8 (2013), pp. 2169–2181. DOI: 10.1016/j.future.2013.04.009.

[61] Michael Behrisch, James Davey, Svenja Simon, Tobias Schreck, Daniel A. Keim, and Jörn Kohlhammer. "Visual Comparison of Orderings and Rankings". In: *EuroVis Workshop on Visual Analytics.* Ed. by M. Pohl and H. Schumann. The Eurographics Association, 2013, pp. 1–7. DOI: 10.2312/PE.EuroVAST.EuroVA13.007-011.

[62] Justin Grimmer and Brandon M. Stewart. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts". In: *Political Analysis* 21.3 (2013), pp. 267–297. DOI: 10.1093/pan/mps028.

[63] Jean-Francois Im, Michael J. McGuffin, and Rock Leung. "GPLOM: The Generalized Plot Matrix for Visualizing Multidimensional Multivariate Data". In: *IEEE Trans. Vis. Comput. Graph.* 19.12 (2013), pp. 2606–2614. DOI: 10.1109/TVCG.2013.160.

[64] Andreas Lamprecht, Annette Hautli, Christian Rohrdantz, and Tina Bögel. "A Visual Analytics System for Cluster Exploration". In: *51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Proceedings of the Conference System Demonstrations.* 2013, pp. 109–114.

[65] Zitao Liu. "A Comparative Study on Linguistic Feature Selection in Sentiment Polarity Classification". In: *CoRR* abs/1311.0833 (2013).

[66] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* 2013, pp. 3111–3119.

[67] Saif Mohammad and Peter D. Turney. "Crowdsourcing a Word-Emotion Association Lexicon". In: *Computational Intelligence* 29.3 (2013), pp. 436–465. DOI: 10.1111/j.1467-8640.2012.00460.x.

[68] OASIS Standard. *searchRetrieve: Part 5. CQL: The Contextual Query Language Version 1.0.* http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/part5-cql/searchRetrieve-v1.0-os-part5-cql.html. 2013.

[69] Daniela Oelke, Dimitrios Kokkinakis, and Daniel A. Keim. "Fingerprint Matrices: Uncovering the dynamics of social networks in prose literature". In: *Computer Graphics Forum* 32.3 (2013), pp. 371–380. DOI: 10.1111/cgf.12124.

[70] Walter L. Perry, Brian McInnis, Carter C. Price, Susan C. Smith, and John S. Hollywood. *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations.* RAND Corporation, 2013. ISBN: 9780833081483.

[71] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. "Geodetic Distance Queries on R-Trees for Indexing Geographic Data". In: *Advances in Spatial and Temporal Databases - 13th International Symposium, SSTD 2013, Munich, Germany, August 21-23, 2013. Proceedings.* Ed. by Mario A. Nascimento, Timos K. Sellis, Reynold Cheng, Jörg Sander, Yu Zheng, Hans-Peter Kriegel, Matthias Renz, and Christian Sengstock. Vol. 8098. Lecture Notes in Computer Science. Springer, 2013, pp. 146–164. ISBN: 978-3-642-40234-0. DOI: 10.1007/978-3-642-40235-7_9.

[72] Colin Ware. "Chapter One - Foundations for an Applied Science of Data Visualization". In: *Information Visualization (Third Edition)*. Ed. by Colin Ware. Third Edition. Interactive Technologies. Boston: Morgan Kaufmann, 2013, pp. 1–30. ISBN: 978-0-12-381464-7. DOI: https://doi.org/10.1016/B978-0-12-381464-7.00001-6.

[73] Florian Heimerl, Charles Jochim, Steffen Koch, and Thomas Ertl. "FeatureForge: A Novel Tool for Visually Supported Feature Engineering and Corpus Revision". In: *COLING, (Posters)*. 2012, pp. 461–470.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed. by Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger. 2012, pp. 1106–1114.

[75] Thomas Lin, Mausam, and Oren Etzioni. "Entity Linking at Web Scale". In: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. AKBC-WEKEX '12. Montreal, Canada: Association for Computational Linguistics, 2012, pp. 84–88.

[76] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012. DOI: 10.2200/S00416ED1V01Y201204HLT016.

[77] Bing Liu and Lei Zhang. "A Survey of Opinion Mining and Sentiment Analysis". In: *Mining Text Data*. Ed. by Charu C. Aggarwal and ChengXiang Zhai. Boston, MA: Springer US, 2012, pp. 415–463. ISBN: 978-1-4614-3223-4. DOI: 10.1007/978-1-4614-3223-4_13.

[78] Jin-Seo Park and Se-Jong Oh. "A New Concave Hull Algorithm and Concaveness Measure for n-dimensional Datasets". In: *J. Inf. Sci. Eng.* 28.3 (2012), pp. 587–600.

[79] Jonathon Read and John A. Carroll. "Annotating expressions of Appraisal in English". In: *Language Resources and Evaluation* 46.3 (2012), pp. 421–447. DOI: 10.1007/s10579-010-9135-7.

[80] Michael Sedlmair, Miriah D. Meyer, and Tamara Munzner. "Design Study Methodology: Reflections from the Trenches and the Stacks". In: *IEEE Trans. Vis. Comput. Graph.* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/TVCG.2012.213.

[81] Josef Steinberger, Mohamed Ebrahim, Maud Ehrmann, Ali Hurriyetoglu, Mijail Alexandrov Kabadjov, Polina Lenkova, Ralf Steinberger, Hristo Tanev, Silvia Vázquez, and Vanni Zavarella. "Creating sentiment dictionaries via triangulation". In: *Decision Support Systems* 53.4 (2012), pp. 689–694. DOI: 10.1016/j.dss.2012.05.029.

[82] Hendrik Strobelt, Marc Spicker, Andreas Stoffel, Daniel A. Keim, and Oliver Deussen. "Rolled-out Wordles: A Heuristic Method for Overlap Removal of 2D Data Representatives". In: *Comput. Graph. Forum* 31.3 (2012), pp. 1135–1144. DOI: 10.1111/j.1467-8659.2012.03106.x.

[83] Stef van den Elzen and Jarke J. van Wijk. "BaobabView: Interactive Construction and Analysis of Decision Trees". In: *2011 IEEE Conference on Visual Analytics Science and Technology, VAST 2011.* 2011, pp. 151–160. DOI: 10.1109/VAST.2011.6102453.

[84] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. "Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments". In: *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Short Papers).* 2011, pp. 42–47.

[85] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. "Robust Disambiguation of Named Entities in Text". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL.* 2011, pp. 782–792.

[86] Ross Maciejewski, Ryan Hafen, Stephen Rudolph, Stephen G. Larew, Michael A. Mitchell, William S. Cleveland, and David S. Ebert. "Forecasting Hotspots - A Predictive Analytics Approach". In: *IEEE Trans. Vis. Comput. Graph.* 17.4 (2011), pp. 440–453. DOI: 10.1109/TVCG.2010.82.

[87] Thorsten May, Andreas Bannach, James Davey, Tobias Ruppert, and Jörn Kohlhammer. "Guiding feature subset selection with an interactive visualization". In: *2011 IEEE Conference on Visual Analytics Science and Technology, VAST 2011.* 2011, pp. 111–120. DOI: 10.1109/VAST.2011.6102448.

[88] Arturas Mazeika, Tomasz Tylenda, and Gerhard Weikum. "Entity timelines: visual analytics and named entity evolution". In: *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011.* 2011, pp. 2585–2588. DOI: 10.1145/2063576.2064026.

[89] Bruce Taylor, Christopher S. Koper, and Daniel J. Woods. "A randomized controlled trial of different policing strategies at hot spots of violent crime". In: *Journal of Experimental Criminology* 7.2 (June 2011), pp. 149–181. ISSN: 1572-8315. DOI: 10.1007/s11292-010-9120-6.

[90] Manish Kumar Anand, Shawn Bowers, and Bertram Ludäscher. "Provenance browser: Displaying and querying scientific workflow provenance graphs". In: *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA.* Ed. by Feifei Li, Mirella M. Moro, Shahram Ghandeharizadeh, Jayant R. Haritsa,

Gerhard Weikum, Michael J. Carey, Fabio Casati, Edward Y. Chang, Ioana Manolescu, Sharad Mehrotra, Umeshwar Dayal, and Vassilis J. Tsotras. IEEE Computer Society, 2010, pp. 1201–1204. ISBN: 978-1-4244-5444-0. DOI: 10.1109/ICDE.2010.5447741.

[91] Manish Kumar Anand, Shawn Bowers, and Bertram Ludäscher. "Techniques for efficiently querying scientific workflow provenance graphs". In: *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings.* Ed. by Ioana Manolescu, Stefano Spaccapietra, Jens Teubner, Masaru Kitsuregawa, Alain Léger, Felix Naumann, Anastasia Ailamaki, and Fatma Özcan. Vol. 426. ACM International Conference Proceeding Series. ACM, 2010, pp. 287–298. ISBN: 978-1-60558-945-9. DOI: 10.1145/1739041.1739078.

[92] Fabian Beck and Stephan Diehl. "Visual comparison of software architectures". In: *Proceedings of the 5th international symposium on Software visualization - SOFTVIS '10* (2010), p. 183. DOI: 10.1145/1879211.1879238.

[93] Gustavo Laboreiro, Luìs Sarmento, Jorge Teixeira, and Eugenio Oliveira. "Tokenizing microblogging messages using a text classification approach". In: *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data, AND 2010).* 2010, pp. 81–88. DOI: 10.1145/1871840.1871853.

[94] Ross Maciejewski, Stephen Rudolph, Ryan Hafen, Ahmad M. Abusalah, Mohamed Yakout, Mourad Ouzzani, William S. Cleveland, Shaun J. Grannis, and David S. Ebert. "A Visual Analytics Approach to Understanding Spatiotemporal Hotspots". In: *IEEE Trans. Vis. Comput. Graph.* 16.2 (2010), pp. 205–220. DOI: 10.1109/TVCG.2009.100.

[95] Elijah Mayfield and Carolyn Penstein Rosé. "An Interactive Tool for Supporting Error Analysis for Text Mining". In: *Proceedings of the NAACL HLT 2010 Demonstration Session.* The Association for Computational Linguistics, 2010, pp. 25–28.

[96] Christin Seifert, Vedran Sabol, and Michael Granitzer. "Classifier Hypothesis Generation Using Visual Analysis Methods". In: *Networked Digital Technologies - Second International Conference, NDT 2010, Part I.* 2010, pp. 98–111. DOI: 10.1007/978-3-642-14292-5_11.

[97] John E. Stone, David Gohara, and Guochun Shi. "OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems". In: *Computing in Science and Engineering* 12.3 (2010), pp. 66–73. DOI: 10.1109/MCSE.2010.69.

[98] Mark A. Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update". In: *SIGKDD Explorations* 11.1 (2009), pp. 10–18. DOI: 10.1145/1656274.1656278.

[99] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition.* Springer series in statistics. Springer, 2009. ISBN: 9780387848570.

[100] Satoshi Sekine and Elisabete Ranchhod, eds. *Named Entities: Recognition, classification and use.* John Benjamins Publishing Company, July 2009. DOI: 10.1075/bct.19.

[101] Wim Bernasco. "Them Again?: Same-Offender Involvement in Repeat and Near Repeat Burglaries". In: *European Journal of Criminology* 5.4 (2008), pp. 411–431. DOI: 10.1177/1477370808095124.

[102] Spencer Chainey, Lisa Tompson, and Sebastian Uhlig. "The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime". In: *Security Journal* 21.1 (Feb. 2008), pp. 4–28. ISSN: 1743-4645. DOI: 10.1057/palgrave.sj.8350066.

[103] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: deep neural networks with multitask learning". In: *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008.* 2008, pp. 160–167. DOI: 10.1145/1390156.1390177.

[104] Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. "A Proper Approach to Japanese Morphological Analysis: Dictionary, Model, and Evaluation". In: *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco.* 2008.

[105] Jana Diesner and Kathleen M. Carley. "Conditional random fields for entity extraction and ontological text coding". In: *Computational & Mathematical Organization Theory* 14.3 (2008), pp. 248–262. DOI: 10.1007/s10588-008-9029-z.

[106] James Frew, Dominic Metzger, and Peter Slaughter. "Automatic capture and reconstruction of computational provenance". In: *Concurrency and Computation: Practice and Experience* 20.5 (2008), pp. 485–496. DOI: 10.1002/cpe.1247.

[107] Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. "Visual Analytics: Scope and Challenges". In: *Visual Data Mining - Theory, Techniques and Tools for Visual Analytics.* Ed. by Simeon J. Simoff, Michael H. Böhlen, and Arturas Mazeika. Vol. 4404. Lecture Notes in Computer Science. Springer, 2008, pp. 76–90. ISBN: 978-3-540-71079-0. DOI: 10.1007/978-3-540-71080-6_6.

[108] S. H. Lee and W. Chen. "A comparative study of uncertainty propagation methods for black-box-type problems". In: *Structural and Multidisciplinary Optimization* 37.3 (May 2008), p. 239. ISSN: 1615-1488. DOI: 10.1007/s00158-008-0234-7.

[109] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval.* Cambridge University Press, 2008. ISBN: 978-0-521-86571-5.

[110] Berk Richard, Sherman Lawrence, Barnes Geoffrey, Kurtz Ellen, and Ahlman Lindsay. "Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 172.1 (2008), pp. 191–211. DOI: 10.1111/j.1467-985X.2008.00556.x.

[111] Sunita Sarawagi. “Information Extraction”. In: *Found. Trends databases* 1.3 (Mar. 2008), pp. 261–377. ISSN: 1931-7883. DOI: 10.1561/1900000003.

[112] Carlos Eduardo Scheidegger, David Koop, Emanuele Santos, Huy T. Vo, Steven P. Callahan, Juliana Freire, and Cláudio T. Silva. “Tackling the Provenance Challenge one layer at a time”. In: *Concurrency and Computation: Practice and Experience* 20.5 (2008), pp. 473–483. DOI: 10.1002/cpe.1237.

[113] Bowers Shawn, McPhillips Timothy M., and Ludäscher Bertram. “Provenance in collection-oriented scientific workflows”. In: *Concurrency and Computation: Practice and Experience* 20.5 (2008), pp. 519–529. DOI: 10.1002/cpe.1226. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.1226.

[114] John T. Stasko, Carsten Görg, and Zhicheng Liu. “Jigsaw: supporting investigative analysis through interactive visualization”. In: *Information Visualization* 7.2 (2008), pp. 118–132. DOI: 10.1057/palgrave.ivs.9500180.

[115] Christian Tominski, Georg Fuchs, and Heidrun Schumann. “Task-Driven Color Coding”. In: *12th International Conference on Information Visualisation, IV 2008, 8-11 July 2008, London, UK.* IEEE Computer Society, 2008, pp. 373–380. ISBN: 978-0-7695-3268-4. DOI: 10.1109/IV.2008.24.

[116] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. “DBpedia: A Nucleus for a Web of Open Data”. In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.* 2007, pp. 722–735. DOI: 10.1007/978-3-540-76298-0_52.

[117] Geoffrey P. Ellis and Alan J. Dix. “A Taxonomy of Clutter Reduction for Information Visualisation”. In: *IEEE Trans. Vis. Comput. Graph.* 13.6 (2007), pp. 1216–1223. DOI: 10.1109/TVCG.2007.70535.

[118] Iryna Gurevych, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch. “Darmstadt Knowledge Processing Repository Based on UIMA”. In: *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the GSCL.* 2007.

[119] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. “NodeTrix: a Hybrid Visualization of Social Networks”. In: *IEEE Trans. Vis. Comput. Graph.* 13.6 (2007), pp. 1302–1309. DOI: 10.1109/TVCG.2007.70582.

[120] Rada Mihalcea and Andras Csomai. “Wikify!: linking documents to encyclopedic knowledge”. In: *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007.* 2007, pp. 233–242. DOI: 10.1145/1321440.1321475.

[121] David Nadeau and Satoshi Sekine. "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26. DOI: 10.1075/bct.19.03nad.

[122] Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Chris Hayworth, Shubha Nabar, Tomoe Sugihara, and Jennifer Widom. "Trio: A System for Data, Uncertainty, and Lineage". In: *Proceedings of the 32Nd International Conference on Very Large Data Bases*. VLDB '06. Seoul, Korea: VLDB Endowment, 2006, pp. 1151–1154.

[123] Alina Beygelzimer, Sham Kakade, and John Langford. "Cover trees for nearest neighbor". In: *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*. Ed. by William W. Cohen and Andrew Moore. Vol. 148. ACM International Conference Proceeding Series. ACM, 2006, pp. 97–104. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143857.

[124] Razvan C. Bunescu and Marius Pasca. "Using Encyclopedic Knowledge for Named entity Disambiguation". In: *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. 2006.

[125] Isabelle Guyon, Masoud Nikravesh, Steve Gunn, and Lotfi A. Zadeh, eds. *Feature Extraction*. Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-35487-1. DOI: 10.1007/978-3-540-35488-8.

[126] Joseph Hassell, Boanerges Aleman-Meza, and Ismailcem Budak Arpinar. "Ontology-Driven Automatic Entity Disambiguation in Unstructured Text". In: *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*. 2006, pp. 44–57. DOI: 10.1007/11926078_4.

[127] Nathalie Henry and Jean-Daniel Fekete. "MatrixExplorer: a Dual-Representation System to Explore Social Networks". In: *IEEE Trans. Vis. Comput. Graph.* 12.5 (2006), pp. 677–684. DOI: 10.1109/TVCG.2006.160.

[128] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". In: *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. 2005.

[129] Jinwook Seo and Ben Shneiderman. "A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data". In: *Information Visualization* 4.2 (2005), pp. 96–113. DOI: 10.1057/palgrave.ivs.9500091.

[130] G. Peter Zhang and Min Qi. "Neural network forecasting for seasonal and trend time series". In: *European Journal of Operational Research* 160.2 (2005), pp. 501–514. DOI: 10.1016/j.ejor.2003.08.037.

[131] David A. Ferrucci and Adam Lally. "UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment". In: *Journal of Natural Language Engineering* 10.3-4 (2004), pp. 327–348. DOI: 10.1017/S1351324904003523.

[132] Peter Gatalsky, Natalia V. Andrienko, and Gennady L. Andrienko. "Interactive Analysis of Event Data Using Space-Time Cube". In: *8th International Conference on Information Visualisation, IV 2004, 14-16 July 2004, London, UK*. IEEE Computer Society, 2004, pp. 145–152. ISBN: 0-7695-2177-0. DOI: 10.1109/IV.2004.1320137.

[133] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. "A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations". In: *10th IEEE Symposium on Information Visualization (InfoVis 2004), 10-12 October 2004, Austin, TX, USA*. 2004, pp. 17–24. DOI: 10.1109/INFVIS.2004.1.

[134] Yusuke Shinyama and Satoshi Sekine. "Named Entity Discovery Using Comparable News Articles". In: *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*. 2004.

[135] Joaquim Ferreira da Silva, Zornitsa Kozareva, and José Gabriel Pereira Lopes. "Cluster Analysis and Classification of Named Entities". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. 2004.

[136] Jun Zhao, Carole A. Goble, Robert Stevens, and Sean Bechhofer. "Semantically Linking and Browsing Provenance Logs for E-science". In: *Semantics for Grid Databases, First International IFIP Conference on Semantics of a Networked World: ICSNW 2004, Paris, France, June 17-19, 2004. Revised Selected Papers*. 2004, pp. 158–176. DOI: 10.1007/978-3-540-30145-5_10.

[137] Oliver Bender, Franz Josef Och, and Hermann Ney. "Maximum Entropy Models for Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*. 2003, pp. 148–151.

[138] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3 (2003), pp. 1137–1155.

[139] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. "The role of trust in automation reliance". In: *Int. J. Hum.-Comput. Stud.* 58.6 (2003), pp. 697–718. DOI: 10.1016/S1071-5819(03)00038-7.

[140] Diansheng Guo. "Coordinating computational and visual approaches for interactive feature selection and multivariate clustering". In: *Information Visualization* 2.4 (2003), pp. 232–246. DOI: 10.1057/palgrave.ivs.9500053.

[141] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[142] Mark Harrower and Cynthia A. Brewer. "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps". In: *The Cartographic Journal* 40.1 (2003), pp. 27–37. DOI: 10.1179/000870403235002042.

[143] Alexander Hinneburg and Daniel A. Keim. "A General Approach to Clustering in Large Databases with Noise". In: *Knowl. Inf. Syst.* 5.4 (2003), pp. 387–415. DOI: 10.1007/10.1007/s10115-003-0086-9.

[144] Carmen Pancerella, John Hewson, Wendy Koegler, David Leahy, Michael Lee, Larry Rahn, Christine Yang, James Myers, Brett Didier, Renata McCoy, Karen Schuchardt, Eric Stephan, Theresa Windus, Kaizar Amin, Sandra Bittner, Carina Lansing, Michael Minkoff, Sandeep Nijsure, Gregor von Laszewski, Reinhardt Pinzon, Branko Ruscic, Al Wagner, Baoshan Wang, William Pitz, Yen-Ling Ho, David Montoya, Lili Xu, Thomas Allison, Jr. William Green, and Michael Frenklach. "Metadata in the Collaboratory for Multi-Scale Chemical Science". In: *International Conference on Dublin Core and Metadata Applications* 0.0 (2003). ISSN: 1939-1366.

[145] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network". In: *HLT-NAACL*. 2003.

[146] Jarke J. van Wijk and Wim A. A. Nuij. "Smooth and efficient zooming and panning". In: *9th IEEE Symposium on Information Visualization (InfoVis 2003), 20-21 October 2003, Seattle, WA, USA*. IEEE Computer Society, 2003, pp. 15–23. ISBN: 0-7695-2055-3. DOI: 10.1109/INFVIS.2003.1249004.

[147] I. Foster, J. Vockler, M. Wilde, and Yong Zhao. "Chimera: a virtual data system for representing, querying, and automating data derivation". In: *Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on*. 2002, pp. 37–46. DOI: 10.1109/SSDM.2002.1029704.

[148] Daniel A. Keim. "Information Visualization and Visual Data Mining". In: *IEEE Trans. Vis. Comput. Graph.* 8.1 (2002), pp. 1–8. DOI: 10.1109/2945.981847.

[149] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification using Machine Learning Techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics. 2002, pp. 79–86.

[150] J. Frew and R. Bose. "Earth System Science Workbench: a data management infrastructure for earth science products". In: *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on*. 2001, pp. 180–189. DOI: 10.1109/SSDM.2001.938550.

[151] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. "On Clustering Validation Techniques". In: *J. Intell. Inf. Syst.* 17.2-3 (2001), pp. 107–145. DOI: 10.1023/A:1012801612483.

[152] Arnold Lund. "Measuring Usability with the USE Questionnaire". In: *Usability and User Experience Newsletter of the STC Usability SIG*. Vol. 8. Jan. 2001.

[153] Alan T. Murray, Ingrid McGuffog, John S. Western, and Patrick Mullins. "Exploratory Spatial Data Analysis Techniques for Examining Urban CrimeImplications for Evaluating Treatment". In: *The British Journal of Criminology* 41.2 (2001), pp. 309–329. DOI: `10.1093/bjc/41.2.309`. eprint: `/oup/backfile/content_public/journal/bjc/41/2/10.1093/bjc/41.2.309/2/410309.pdf`.

[154] James W Pennebaker, Martha E Francis, and Roger J Booth. "Linguistic inquiry and word count: LIWC 2001". In: *Mahway: Lawrence Erlbaum Associates* 71 (2001), p. 2001.

[155] Choong-Nyoung Seon, Youngjoong Ko, Jeong-Seok Kim, and Jungyun Seo. "Named Entity Recognition using Machine Learning Methods and Pattern-Selection Rules". In: *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium, November 27-30, 2001, Hitotsubashi Memorial Hall, National Center of Sciences, Tokyo, Japan*. 2001, pp. 229–236.

[156] Shumeet Baluja, Vibhu O. Mittal, and Rahul Sukthankar. "Applying Machine Learning for High-Performance Named-Entity Extraction". In: *Computational Intelligence* 16.4 (2000), pp. 586–596. DOI: `10.1111/0824-7935.00129`.

[157] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. "OPTICS: Ordering Points To Identify the Clustering Structure". In: *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*. Ed. by Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh. ACM Press, 1999, pp. 49–60. ISBN: 1-58113-084-8. DOI: `10.1145/304182.304187`.

[158] Mary Elaine Califf and Raymond J. Mooney. "Relational Learning of Pattern-Match Rules for Information Extraction". In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA*. 1999, pp. 328–334.

[159] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in information visualization - using vision to think*. Academic Press, 1999. ISBN: 978-1-55860-533-6.

[160] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. "NYU: Description of the MENE Named Entity System as Used in MUC-7". In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. 1998.

[161] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN: 0132733501.

[162] Daniel M. Bikel, Scott Miller, Richard M. Schwartz, and Ralph M. Weischedel. "Nymble: a High-Performance Learning Name-finder". In: *ANLP*. 1997, pp. 194–201.

[163] Julian S. Joseph, Marvin M. Chun, and Ken Nakayama. “Attentional requirements in a ‘preattentive’ feature search task”. In: *Nature* 387.6635 (June 1997), pp. 805–807. DOI: 10.1038/42940.

[164] J. K. Rowling. *Harry Potter and the sorcerer’s stone*. New York: Scholastic, 1997. ISBN: 978-0439708180.

[165] Yiming Yang and Jan O. Pedersen. “A Comparative Study on Feature Selection in Text Categorization”. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. ICML ’97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 412–420. ISBN: 1-55860-486-3.

[166] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. “The Quickhull Algorithm for Convex Hulls”. In: *ACM Trans. Math. Softw.* 22.4 (1996), pp. 469–483. DOI: 10.1145/235815.235821.

[167] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*. Ed. by Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad. AAAI Press, 1996, pp. 226–231. ISBN: 1-57735-004-9.

[168] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. “From Data Mining to Knowledge Discovery in Databases”. In: *AI Magazine* 17.3 (1996), pp. 37–54.

[169] Ben Shneiderman. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”. In: *VL*. 1996, pp. 336–343.

[170] James R. Lewis. “IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use”. In: *Int. J. Hum. Comput. Interaction* 7.1 (1995), pp. 57–78. DOI: 10.1080/10447319509526110.

[171] William B. Cavnar and John M. Trenkle. “N-Gram-Based Text Categorization”. In: *Proceedings of SDAIR-94, Third Annual Symposium on Document Analysis and Information Retrieval*. 1994, pp. 161–175.

[172] John D. Lee and Neville Moray. “Trust, self-confidence, and operators’ adaptation to automation”. In: *Int. J. Hum.-Comput. Stud.* 40.1 (1994), pp. 153–184. DOI: 10.1006/ijhc.1994.1007.

[173] Bonnie M. Muir. “Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems”. In: *Ergonomics* 37.11 (1994), pp. 1905–1922. DOI: 10.1080/00140139408964957. eprint: https://doi.org/10.1080/00140139408964957.

[174] Raymond T. Ng and Jiawei Han. “Efficient and Effective Clustering Methods for Spatial Data Mining”. In: *VLDB’94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*. Ed. by Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo. Morgan Kaufmann, 1994, pp. 144–155. ISBN: 1-55860-153-8.

[175] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992.* 1992, pp. 144–152. DOI: `10.1145/130385.130401`.

[176] M. Granger Morgan, Max Henrion, and Mitchell Small. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis.* Cambridge University Press, 1992. ISBN: 0521427444.

[177] Andreas Buja, John Alan McDonald, J. Michalak, and Werner Stuetzle. "Interactive Data Visualization Using Focusing and Linking". In: *IEEE Visualization.* 1991, pp. 156–163. DOI: `10.1109/VISUAL.1991.175794`.

[178] L. F. Rau. "Extracting company names from text". In: *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application.* Vol. i. Feb. 1991, pp. 29–32. DOI: `10.1109/CAIA.1991.120841`.

[179] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: `https://doi.org/10.1016/0377-0427(87)90125-7`.

[180] Eugene W. Myers. "An O(ND) Difference Algorithm and Its Variations". In: *Algorithmica* 1.2 (1986), pp. 251–266. DOI: `10.1007/BF01840446`.

[181] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, 1986. ISBN: 0412246201.

[182] Stuart P. Lloyd. "Least squares quantization in PCM". In: *IEEE Trans. Information Theory* 28.2 (1982), pp. 129–136. DOI: `10.1109/TIT.1982.1056489`.

[183] Lawrence E. Cohen, Marcus Felson, and Kenneth C. Land. "Property Crime Rates in the United States: A Macrodynamic Analysis, 1947-1977; With Ex Ante Forecasts for the Mid-1980s". In: *American Journal of Sociology* 86.1 (1980), pp. 90–118. DOI: `10.1086/227204`.

[184] G. V. Kass. "An Exploratory Technique for Investigating Large Quantities of Categorical Data". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 29.2 (1980), pp. 119–127. ISSN: 00359254, 14679876.

[185] Robert Plutchik. "A general psychoevolutionary theory of emotion". In: *Theories of emotion* 1.3-31 (1980), p. 4.

[186] John W Tukey. *Explorative data analysis.* 1977.

[187] Bela Julesz. "Experiments in the Visual Perception of Texture". In: *Scientific American* 232.4 (1975), pp. 34–43. ISSN: 00368733, 19467087.

[188] J. C. Dunn. "Well-Separated Clusters and Optimal Fuzzy Partitions". In: *Journal of Cybernetics* 4.1 (1974), pp. 95–104. DOI: `10.1080/01969727408546059`.

[189] F. J. Anscombe. “Graphs in Statistical Analysis”. In: *The American Statistician* 27.1 (1973), pp. 17–21. ISSN: 00031305.

[190] Richard O. Duda and Peter E. Hart. *Pattern classification and scene analysis*. A Wiley-Interscience publication. Wiley, 1973. ISBN: 0471223611.

[191] Donald Shepard. “A Two-dimensional Interpolation Function for Irregularly-spaced Data”. In: *Proceedings of the 1968 23rd ACM National Conference*. ACM ’68. New York, NY, USA: ACM, 1968, pp. 517–524. DOI: 10.1145/800186.810616.

[192] Sarah L. Boggs. “Urban Crime Patterns”. In: *American Sociological Review* 30.6 (1965), pp. 899–908. ISSN: 00031224.

[193] Arthur L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[194] Dezydery Szymkiewicz. *Une contribution statistique a la géographie floristique*. Polskie Towarzystwo Botaniczne, 1934.