Document Structure Analysis for Large Electronic Document Collections

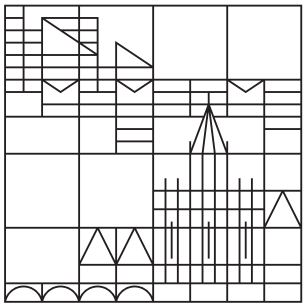
Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften

vorgelegt von

Andreas Stoffel

an der

Universität Konstanz



Mathematisch-Naturwissenschaftliche Sektion

Informatik und Informationswissenschaft

Tag der mündlichen Prüfung: 27. Juli 2013

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

2. Referent: Prof. Dr. Oliver Deussen

Acknowledgments

First I want to thank Prof. Dr. Daniel Keim for giving me the opportunity to work in his group on several interesting research topics. This thesis is also an outcome of his encouragement and support. I want to thank Prof. Dr. Oliver Deussen for his advice and discussions in the past years that helped me to find the right direction.

A big thank you for all the valuable discussions and feedback to the colleagues Peter Bak, Enrico Bertini, Miloš Krstajić, Daniela Oelke, Christian Rohrdantz, Hendrik Strobelt, and Franz Wanner, at University of Konstanz as well as Richard Suchenwirth-Bauersachs, Henrik Kinnemann, Bernd Radtke, and Roland Zimbel at Siemens AG in Konstanz. You had always an open door for my question and problems and you influenced my work in one way or another.

I also want to thank Fabian Fischer, Johannes Fuchs, Halldór Janetzko, Slava Kiselivic, Florian Mansmann, Sebastian Mittelstädt, Matthias Schäfer, Svenja Simon, David Spretke, Andrada Tatu, and Hartmut Ziegler as well as Karl-Heinz Bentele, Jakob Brendel, Sergey Grosman, Zhe Li, Udo Milezki, Uwe Nootbaar, Marc-Peter Schambach, and Matthias Schulte-Austum.

Abstract

Beside document collections containing a wide variety of different documents, such as the web, collections exist, which are collecting documents of a single or very few different types. For example, the EDGAR database of the SAC is collecting different documents of companies, which have to report regularly on different issues of their business. Similar databases exist in medicine or in industry. Due to the nature of this database, to contain very similar documents, it is often desirable to provide analysis functionalities that are able to automatically gain some insight into the filed information.

A major problem of the automatic processing is the lack of structure information in electronic documents. The majority of electronic document formats used for archiving, are based on visual representations for human readers. This makes automatic processing complex, because relevant and irrelevant content cannot be automatically distinguished easily.

This thesis addresses this issue and describes and evaluates techniques for logical and functional structure analysis. The presented techniques are based on machine learning. Whereas the analysis of logical structure uses mainly geometric and formatting information, inspects the analysis of functional structures the textual content.

The problem to identify and analyze errors in the structure analysis results is solved with visualization. The variable text scaling technique is designed to highlight interesting parts in logical and functional structures. It is also applicable to visualize keyword search results in document viewers.

Afterwards several examples using the presented techniques are discussed. The thesis concludes with a summary of the results and discusses open research questions.

Zusammenfassung

Neben Dokumentkollationen, die aus einer großen Bandbreite an unterschiedlichen Dokumenten bestehen, zum Beispiel das Internet, existieren auch Kollektionen, die einzelne oder sehr wenige unterschiedliche Dokumentarten sammeln. Zum Beispiel die EDGAR Datenbank des SEC sammelt unterschiedliche Dokumente von Unternehmen, welche regelmäßig über unterschiedliche Angelegenheiten ihres Geschäfts berichten müssen. Ähnliche Datenbanken gibt es in der Medizin oder in der Industrie. Auf Grund der Natur dieser Datenbanken sehr ähnliche Dokumente zu enthalten, ist es oft wünschenswert Analysefunktionalität bereitzustellen, die es ermöglicht einen Einblick in die abgelegten Informationen zu bekommen.

Ein großes Problem für die automatische Verarbeitung ist der Mangel an Strukturinformation in elektronischen Dokumenten. Die Mehrheit der elektronische Dokumentenformate, die zum Archivieren verwendet wird, basieren auf einer visuellen Repräsentation für menschliche Leser. Dies verkompliziert die automatische Verarbeitung, da relevanter und irrelevanter Inhalt nicht einfach automatisch unterschieden werden kann.

Diese Arbeit befasst sich mit diesem Problem und beschreibt und evaluiert Techniken für die logische und funktionale Strukturerkennung. Die präsentierten Techniken basieren auf maschinellem Lernen. Während die Analyse von logischen Strukturen vorwiegend Geometrie- und Formatierungsinformationen verwendet, untersucht die Analyse von funktionalen Strukturen den textuellen Inhalt.

Das Problem Fehler in den Ergebnissen der Strukturanalyse zu identifizieren und zu analysieren wird mit Visualisierung gelöst. Die Variable Text Scaling Technik ist entworfen worden um Interessante Abschnitte in logischen und funktionalen Strukturen hervorzuheben. Sie ist ebenso anwendbar um die Ergebnisse einer Schlüsselwortsuche in Dokumentbetrachtern darzustellen.

Anschließend werden mehrere Beispiele diskutiert, die die gezeigten Techniken anwenden, und schließt mit einer Zusammenfassung der Ergebnisse und einer Diskussion offener Forschungsfragen.

Contents

1 Introduction **1**

1.1 Document Structures 3

1.2 Overview and Contributions 5

2 Methods for Logical Structure Analysis **9**

2.1 Motivation 9

2.2 Related Work 11

2.3 Logical Structure Analysis Framework 13

2.3.1 Document Representation and Preprocessing 14

2.4 Features for Logical Structures 14

2.5 Methods for Logical Structure Analysis 16

2.5.1 Selection of Machine Learning Algorithm 18

2.5.2 Efficient Creation of Reference Data 20

2.6 Evaluation 23

2.6.1 Discussion 29

2.7 Summary 30

3 Methods for Functional Structure Analysis **33**

3.1 Motivation 33

3.2 Related Work 35

3.3 Features for Functional Structure Analysis 36

3.4 Learning Functional Structures 37

3.4.1 Selection of Machine Learning Algorithms 38

3.5 Evaluation 40

3.5.1 Discussion 48

3.6 Summary 51

4 Visualization for Document Structure Analysis 53

4.1 Motivation 53
4.2 Related Work 54
4.3 Variable Text Scaling 57
4.3.1 Creating Distorted Page Thumbnails 57
4.3.2 Overview of the Algorithm 58
4.3.3 Distortion of Text 59
4.3.4 Evaluation 63
4.4 Visualization for Document Structure Analysis 66
4.4.1 Visualization of Structure Analysis Results 66
4.4.2 Visualization of Features 68
4.5 Summary 71

5 Applications of Structure Analysis and Document Visualization 75

5.1 Logical Structure Analysis for Text Processing 75
5.2 Document Content Visualization with Distorted Thumbnails . . . 77
5.2.1 Keyword Search in Documents 79
5.2.2 Document Overview 80
5.3 Summary 84

6 Conclusion and Remarks 85

6.1 Logical Structure Analysis 85
6.2 Functional Structure Analysis 86
6.3 Variable Text Scaling and Structure Visualization 87

A Definition of Evaluation Measures 89

B Logical Structure Analysis 91

B.1 Features for Logical Structure Analysis 91
B.2 Results of the Classifier Selection 93
B.3 Iterative Learning of Structure Analysis Model 96
B.4 Evaluation Results 97

C Functional Structure Analysis **105**

C.1 Results of Classifier Selection 105

C.2 Evaluation Results 107

D Applications **111**

D.1 Document Overview 111

Chapter 1

Introduction

The term “document” has several different definitions in information science. For instance, any physical object depending on its purpose can be seen as a document [Buc97]. Within this thesis the term document is used only for textual documents such as books, articles, or reports. The purpose of such documents is transporting or archiving information in a standardized way. Standards can comprise the document contents, formats and even the way of exchange. They are defined by an authority or evolve as convention. For instance, the US law requires every company traded at an US stock market to create yearly a Form 10-K report and submit it to the U.S. Securities and Exchange Commission (SEC). The requirements for this report specify not only the required information but also how the reports have to be submitted and the structure of their content [SC12]. The standardization of the Form 10-K reports allows possible investors to find information more quickly and allows comparison of reports more easily. Papers published in journals or proceedings are a different example. The publishers or editors define style guides for the formatting of the papers. These guidelines make sure to have a uniform appearance of journal. The structure of the paper is not formally defined but a convention within the research community. It helps a reader in orienting and finding information in the document.

Often documents of the same kind containing similar information are collected for archiving purposes or to improve the accessibility of the informa-

tion. The EDGAR database of the SEC collects reports and other documents, including Form 10-K reports, and makes them public available. The database supports an information retrieval task and provides a search interface for users. The search interface allows complex queries for keywords and phrases. Queries are executed on a whole report. This makes it impossible to search for reports having a specific keyword in one of the parts. For instance, it could be interesting to retrieve only documents having a specific keyword in the description of risk factors.

There exist many other document collections with similar properties. For instance, in medicine doctors' letters are usually of similar structures and describe diagnoses and treatments. Service reports are a similar case in industry. Service technicians are writing reports about maintenance of productions, which describe failures, problems, and their solutions. All these document collections contain only a few different types of documents and the structure of the documents is very similar within one collection. An important property of these collections is that they grow over time. The EDGAR database grows with every additional filed report. It is the same for service report databases. If such a growing document collection exists over several years it is unavoidable that the structure of the collected documents is changing over time. The reasons for these changes are manifold. One reason for changes is an advance in technology. Databases collecting documents over several decades usually have a large number of documents written with typewriters, whereas nowadays documents are usually stored in an electronic format. Changes in regularities are another different reason for changes in document collections. For instance, if the rules for Form 10-K reports are changed by law, these changes are reflected within the documents of the EDGAR database.

In order to make use of these document collections, it is often desirable to be able to provide powerful analysis functionalities. For instance, service reports could be analyzed to figure out common sources of errors in products or processes. Or an automatically generated assessment of a portfolio based on Form 10-K reports could be an interesting analysis question. To solve such analysis questions automatic algorithms are needed that supports a user with

the analysis. For instance, a filtering of the documents in the collections based on the users need could be of great help. Or more complex automatic analysis could summarize multiple documents and create an overview about interesting properties of the documents.

A complex analysis of documents usually requires some information about the document structure. For instance, a simple information retrieval engine indexes every term in the document [MRS08] and treats them equally important. As a result the retrieval system can only find complete documents. Using document structure information could add an additional benefit for such a retrieval system. On the one side, different structures can be weighted differently allowing the system to assign lower weights to page headers than to headlines. On the other side, the system can provide a more powerful query language, which allows the specification of structure components. In case of the EDGAR database a user could search for a keyword only the description of risk factors and finding all companies reporting a risk related to this keyword.

A different problem exists for a linguistic analysis of the documents. In this case the document content must be prepared in a way that is possible to be processed with linguistic algorithms. For instance, the readability analysis requires complete sentences as input, because measures used for readability analysis are calculated on sentences. The document structure is used in this case to extract the running text from the document and remove content, such as headers or footers, which would mislead linguistic algorithms.

1.1 Document Structures

A document is a complex object composed of several different contents. The document structuring is grouping this content into meaningful elements that helps a reader in orientation within the document. Usually has several different document structures are distinguished.

The *physical structure* of a document describes the physical representation of a document, for example the placement and formatting of characters or images. The physical structure is usually created from the logical description of a

document. In the past, the physical structure of a document was created manually by compositors, who placed each single letter on a page. Nowadays, the physical structure is either created automatically via typesetting algorithms or manually with specific desktop publishing software.

The *logical structure* describes a document in logical terms, which are independent from the final medium. Logical structures, such as headlines, enumeration, and so forth, are usually mapped to different physical representations and are normally expressed with different visual styles. The logical structure is used as input for many contemporary text processing tools. For instance, Microsoft Word allows the user to express the logical structure with different formatting styles or LaTeX uses commands to define the logical structure of documents.

The *functional structure* of a document describes the functions of the different parts of a document. A typical functional structure is the introduction of an article. Its function is to motivate the topic of the article and to rouse interest. A functional element is usually represented with several logical structures, for instance an introduction can start with an according headline followed by paragraphs. The outline of a document usually describes its functional structure.

According to [WEK12] describes the *discourse structure* the patterns that a reader sees in multi-sentence texts. It can comprise topics, functions of sentences, and events or states in the text flow. The discourse structure mainly describes how an authors expresses different thoughts in a document. The discourse structure is independent from other structures described so far and is mainly a linguistic matter.

An electronic document filed in a database contains at least the physical structure. The physical structure can be stored as a rendered image of the document or as a proper description of the rendering process. The PDF format, which is widely used in archives for filing documents, supports both representations. Interestingly, the PDF format supports, since version 1.4, tagging of the logical structure within documents [Sta08]. Unfortunately, many PDF creators do not make use of this functionality. For instance, LaTeX, which is used to type-

set this thesis, does (in the moment) not tag the logical structure of a compiled PDF document. To be able to process the structures of these documents, an automatic process is advisable in order to process larger collections efficiently.

1.2 Overview and Contributions

This thesis is addressing the problem of logical and function structure recognition. The main focus are electronic document collections consisting of many documents of the same type, for example, the EDGAR database, medical reports, or paper collections. Contemporary systems for logical or functional structure analysis are complex solutions for specific document types and analysis tasks. For instance, the system described in [NNS04] focuses on mathematical article and extracts headlines and mathematical components. It uses this information to provide a special browser for mathematical articles. In contrast, the aim of this work is to evaluate different logical and functional structure analysis approaches and to create a general framework that is easily adaptable to different document types and analysis tasks.

The recognition of physical or discourse structures is not part of this thesis. The physical structure in document images can be recognized with existing OCR technology in good quality. In the majority of other electronic document formats used for archiving, such as PDF, the physical structure is used to describe the document content. The physical structure can therefore be extracted directly from these documents. The automatic recognition of the discourse structure is mainly a computer linguistic topic, which is out of the scope of this work.

Chapter 2 describes a machine learning approach for logical structure. This chapter discusses the different features for logical structures and evaluates different machine learning algorithms. Finally, the proposed system is evaluated and compared to two different approaches based on rules and grammars. It is shown that the presented approach outperforms existing techniques.

The analysis of functional structure is discussed in Chapter 3. Within the chapter different features for functional structure analysis are discussed and

several machine learning techniques are presented and evaluated. It is shown that functional structures are much more complicated to recognize than logical ones and reasons therefore are analyzed and discussed.

Visualizations of logical and functional structure analysis are presented in Chapter 4. A visualization technique is developed that allows highlighting of texts and structural elements based on interest functions. This technique is used to highlight uncertainty in logical and functional structures. In addition, the technique is used to visualize features of logical structures and allows a better understanding of these features and improves feature engineering process.

Chapter 5 discusses several application examples for document structure analysis and the developed visualization technique. The benefit of logical structures is shown with the examples of readability analysis and the Document Cards visualization. The visualization technique developed for structure analysis tasks is applied in a context of a document reader for keyword search and document overview.

Finally, the thesis concludes with a summary and a discussion of further directions and open questions in Chapter 6.

Parts of this thesis are published in:

Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel, Daniel A. Keim, and Oliver Deussen. “Document Cards: A Top Trumps Visualization for Documents”. In: *IEEE Trans. Vis. Comput. Graph.* 15.6 (2009), pp. 1145–1152.

Andreas Stoffel, David Spretke, Henrik Kinnemann, and Daniel A. Keim. “Enhancing Document Structure Analysis using Visual Analytics”. In: *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC2010)*. Ed. by Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung. ACM, 2010, pp. 8–12.

Henrik Kinnemann, Andreas Stoffel, Daniel Keim, and David Spretke. “Verfahren und Vorrichtung zum Erkennen und Klassifizieren von Dokumentteilen eines rechnerverfügbaren Dokuments durch schrittweises Lernen aus mehreren Trainingsmengen”. Patent DE102009050681. Dec. 5, 2011.

Daniela Oelke, David Spretke, Andreas Stoffel, and Daniel A. Keim. “Visual Readability Analysis: How to Make Your Writings Easier to Read”. In: *IEEE Trans. Vis. Comput. Graph.* 18.5 (2012), pp. 662–674.

Andreas Stoffel, Hendrik Strobelt, Oliver Deussen, and Daniel A. Keim. “Document Thumbnails with Variable Text Scaling”. In: *Comput. Graph. Forum* 31.3 (2012), pp. 1165–1173.

Chapter 2

Methods for Logical Structure Analysis

This chapter describes automatic logical structure analysis methods for document collections. The chapter starts with a motivation and continues with a discussion of related work for logical structure analysis. Afterwards the proposed framework, the features, and the used method for logical structure analysis are described and the approach is evaluated. Finally, this chapter is summarized. The framework, the features, and a previous approach based on decision trees are published in [Sto+10; Kin+11].

2.1 Motivation

The logical structure describes a document as a hierarchy of visually distinguishable components [Sum98]. These components, for instance headlines or paragraphs, are usually used to structure the document for a reader. Unfortunately, the explicit information about the logical structure of documents is in many cases lost when archiving or exchanging the document, even though it could improve the automatic processing of documents. The logical structure is lost when a document is printed or converted into a PDF, because these formats are mainly using visual information to represent documents.

Knowledge about the logical structure of documents is valuable, as it improves the document analysis tasks. For example, in web information retrieval the content is weighted differently depending on its logical structure. Weighting of structures can be done globally or by query. Global weighting uses predefined weights in order to improve the relevance of the retrieval result of a typical query. It increases the weight of relevant content and reduces the weight of irrelevant one [SB88]. For example, titles or headlines are typically relevant for queries and are weighted higher, whereas page header/footer or page numbers are weighted lower. Query based weighting is supported by some information retrieval systems. These systems allow users to specify the weights of terms in a document structure along with the query.

A different usage of logical structure is the cleaning and preparation of documents for further automatic processing, such as natural language processing (NLP) or information extraction. The majority of NLP algorithms are designed to work on running text or sentences. In order to apply them to arbitrary documents, the documents have to be cleaned to get the required running text input. Headlines, captions, and other text not belonging to the document body should be removed. In addition, the logical structure is used to correctly join text at column and page boundaries.

In addition to automatic processing of documents, the logical structure is important for displaying the document or parts of it. For instance, in information retrieval the relevant section of a larger book can be presented to a user instead of the whole book. Another application is the conversion of documents into another page format, for instance, for portable devices such as e-book readers, or the extraction of the table of contents for navigation purposes.

The logical structure information explicit available in an electronic document depends on the document format and how the document is created. If the electronic document is created from a paper document with an OCR application, the existing of logical structure depends on the OCR process. Contemporary OCR applications are partly analyzing the logical structure of an input image in order to improve the text recognition rate and restore the reading order of the document. In case the document is fully created in an electronic

way, the process creating the final document is determining the explicit available logical structure. With many word processing application, the author is specifying the logical structure of a document either with styles as in Microsoft Word or OpenOffice writer or with markups as in LaTeX or (X)HTML. Whether this logical structure is preserved in the final document or only the visual information is stored, depends on the application used to create the document. For instance, this thesis is written in LaTeX and contains logical annotations such as headlines or paragraphs. This information is lost by the LaTeX processor when converting the sources into PDF format, even though the PDF format supports tags for logical structure [Sta08].

Although the author is able to specify the logical structure of a document during the creation of an electronic document, this information is often not reliable. Many word processing applications rely on the assigned visual styles to detect the corresponding logical structure. In case a user changes the formatting of text directly without assigning the correct style this approach does not work. A similar problem exists with markup techniques. For instance, in HTML documents it is a common problem to misuse tables in order to arrange content on a web page in order to overcome restrictions of browsers.

2.2 Related Work

The analysis of document structure is mainly used for document image analysis and information extraction. Overviews of different structure analysis approaches for document images can be found in [Nag00; NJ07]. The most common techniques for structure analysis are rules and grammars systems. Rule-based approaches evaluate predefined rules to assign labels to the text regions [KLT01; NNS04]. Rule systems are specially designed for a specific document collection or analysis task and are using geometric, formatting, and content features. Simple rule systems use fixed coded rules, whereas more complex rule systems can adapt the rules to a concrete collection by adjusting weights in the rules. A more complex type of rule system is using emergent computing [Ish05]. Emergent computing is using multiple agents, which follow differ-

ent rules and are interacting with each other. Together the agents form a bigger system that can be adjusted to detect the structure of documents. Alternatively, various kinds of grammars have been proposed for structure analysis [Anj01; RNM07]. These systems model documents with different kinds of grammars and assigns labels to text regions by applying the predefined grammar rules to the documents. Grammar based approaches use the same feature classes than rule-based approaches. Their main advantage is the ability to express complex dependencies between logical components with a grammar. These approaches can be improved by using statistical rules or grammars, allowing the rule or grammar system to adapt itself to a training data set [BZI97; HNZ05; KDK00].

A different approach is the usage of sequential models for logical structure analysis [RB06]. The advantage of sequential models (e.g. hidden Markov models, or conditional random fields) over standard techniques is their ability to make use of context information. The feature types used for sequential modeling consisting of geometric, formatting, and content features are the same as used with rules and grammars.

The drawback of existing rule and grammar systems is their inflexibility. They are tailored to a specific type of document and analysis task. For instance, a system may only recognize the logical structure of articles in a specific journal. This problem is not solved with the ability to automatically adapt weights or probabilities of rules or grammars as supported by some systems. The automatic adaption of the weights helps in improving the quality of the system, because a larger number of documents can be considered in training the system than it is possible with a manual approach. The problem to define the rules or grammars manually, which makes these approaches inflexible to adapt to new tasks, is not solved by learning weights. In addition, rules and grammar systems are complicated to maintain, because the systems get more and more complex with the number of rules. Changing rules in this case can have implications that are complicated to overlook and the systems gets impossible to maintain.

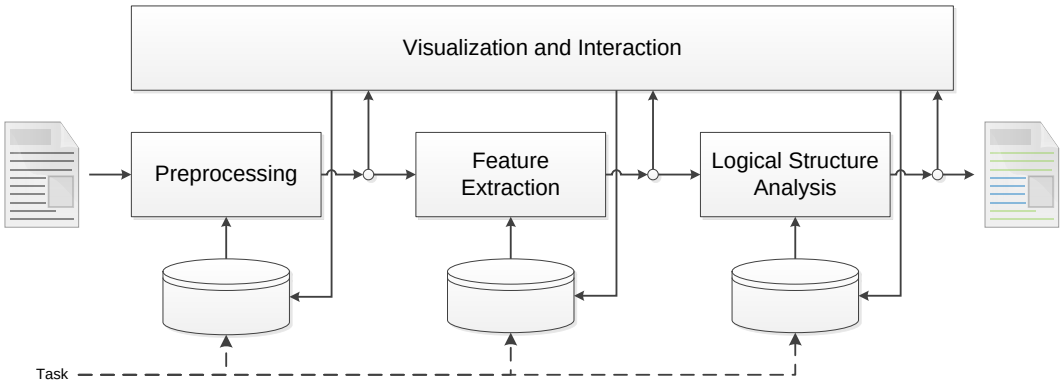


Figure 2.1: The logical structure analysis framework and its components.

2.3 Logical Structure Analysis Framework

The proposed logical structure analysis framework consists of four main components, as shown in Figure 2.1: Preprocessing, Feature Extraction, Logical Structure Analysis, and Visualization and Interaction. Each of these components is controlled by a task dependent model. The models allow the framework to process different types of documents and to perform different analysis tasks.

The Preprocessing component converts input documents into a hierarchical document format required for further processing. This document format describes the geometrical, formatting, and textual content of a document. The available information after preprocessing is depending on the input format and the model of the process.

The Feature Extraction component calculates the features required for logical structure analysis. The features can only depend on the physical structure and the content of a document, which allows only geometric, formatting, and content features to be extracted.

The final step is creating the annotated document, with the Logical Structure Analysis component. This component uses machine learning techniques to analyze the extracted features and to calculate the probabilities for the different logical structures.

The Visualization and Interaction component is able to visualize the results of the different steps. This component provides tools for inspecting and manually correcting the results of the different analysis steps. In addition, users are

able to create or adapt the different models. The visualization of logical structures is presented in Chapter 4.

2.3.1 Document Representation and Preprocessing

The structure analysis process requires an electronic representation of a document containing its physical structure. At least geometry, formatting, and textual content of text lines are required to be able to extract useful features. Several ways exists to convert an existing document into this format. Either the physical structure can be accessed directly from electronic documents, such as a PDF document, or the document has to be preprocessed with an OCR system to extract the according information.

This physical representation of a page is than converted into a hierarchy of rectangular nodes, which is used to express the physical structure of a page. For instance, a column node might contain nodes describing the different paragraphs, which itself contain nodes for text lines. Each of these nodes may contain additional information, for instance text nodes may describe the used font or line nodes may describe line spacing and indentations. Finally, the lines have to be ordered according to the reading order of the text.

OCR applications usually recognize columns and paragraph structures, and a reading order of text lines when analyzing documents, because this information helps improving the text recognition result. In case this information is not present, as in many PDF documents for example, OCR techniques (e.g. [Bre03]) can be used for recognizing the missing information. Especially the reconstruction of a reading order of text lines is important for the structure analysis.

2.4 Features for Logical Structures

After preprocessing a document, the content and the physical properties of the document are accessible. The physical properties can be divided into geometry and formatting properties. In addition to the content, all of these properties can

be used as features for logical structure analysis. A detailed description of the used features for logical structure analysis can be found in Appendix B.1.

The geometric features describe the position and size of a line on the page. These features can be used for identifying header and footer of pages. Lines that appear on top or bottom of a page are headers respectively footers if they have a small font size. In addition to the position of the line on a page, the position within the whole document is regarded as a feature. This is useful to identify structures appearing frequently at a specific region within the document, for instance titles at the beginning or references at the end.

The formatting features consider spacing, indention, and font properties. The spacing features describe the distances between a line and the previous one. With this type of features, structures with special spacing properties can be recognized. For instance, the distances between headlines and adjacent lines are usually larger than for normal text lines. Besides the spacing characteristics, the indentions of lines are represented as features. Depending on the type of justification of the text, these features can be used to recognize the beginning and the end of paragraphs. Formulas, captions or larger quotations have usually a different indention than normal text. In addition, font properties are used as features. The font style can be varying by use of different fonts, font size, weights or italic characters. Typically, headlines have a larger font weight and a larger font size than normal text, whereas headers and footers have usually a smaller font size.

In addition to the formatting and layout features, matches of regular expressions with the line content are used as features as well. These features are represented as binary values. They are set to 1 if the regular expressions match, otherwise to 0. Mainly two types of patterns are used. Patterns based on character classes (e.g. characters, digits, or letter case) are independent of a document type. Their main usage is the detection of enumerations or headlines. Patterns based on keywords are usually used to detect document type dependent structures. For instance, figures or tables in papers can be detected with appropriate keyword features. Extracting semantically meaningful features from the text content does not make sense for detecting logical structures, because log-

ical structure is used to organize the document for a reader and not to express semantics.

2.5 Methods for Logical Structure Analysis

During the recognition of the logical structure of a document, each text line is assigned a label of a logical structure. Text lines are used as the basic element for the analysis, because they are the basic output of the OCR process and any OCR software can be expected to be able to recognize text lines. In fact, many OCR applications are able to recognize paragraphs and columns as well. But their approach is very general and motivated by the improvement of the recognition of texts in document images. Structures special for a particular document type are usually not recognized with this approach.

The usage of a machine learning technique for analyzing the logical structure of documents has the advantage that the resulting system can be adapted easily to a specific type of document only by learning from provided examples. This solves the main problem of rule and grammar based solutions. In order to adapt the system to new document collections or different recognition tasks, only enough training examples need to be provided. The task to manually adapt or create rules or grammars is replaced by an automatic learning process. The examples are needed in any case, because the rules or grammars have to be evaluated in order to verify the resulting system.

In order to find a suitable machine learning algorithm, several opportunities are tested. For the test artificial neuronal networks (ANN), support vector machines (SVM), decision trees (DT), and conditional random fields (CRF) are selected. These algorithms are known to perform very well for different classification tasks [CN06; Kot07; HTF11]. It was shown for SVMs and ANNs that they are able to achieve high classification accuracies in different classification tasks. SVMs have in addition the property to create an optimal classifier in the respect that an SVM maximizes the classification bounds between classes. The main problem of SVMs and ANNs is their handling of nominal features. SVMs and ANNs assume rational input features, allowing calculations in the feature

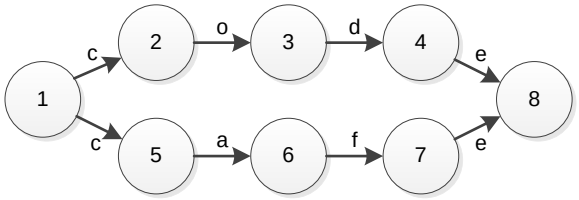


Figure 2.2: Illustration of the labeling bias problem of HMMs and MEMMs for the sequences “code” and “cafe”.

space. For nominal features is assumption does not hold. On the other hand DTs work very well with nominal features and have the additional advantage to create interpretable decisions.

So far the presented classifiers are classifying single instances and to not take neighborhood into account. The CRF is the only sequence classifier in the test, which optimizes the classification result over a sequence of instances. We decided not to include hidden Markov models (HMM) or maximum-entropy Markov models (MEMM), because the CRF is known to outperform both in many cases [LMP01]. The main advantage of CRFs is the ability to find an optimal state sequence considering the whole input, whereas the state sequence generated by HMMs or MEMMs depends on the local input of the particular state. This leads to the labeling bias problem of HMMs or MEMMs. The labeling bias problem describes a situation where the output of the model is mainly depending on the training data and not on the observed input. For instance, Figure 2.2 shows a simple state model for the two words “code” and “cafe”. The states 2, 3, and 4 respectively 5, 6, and 7 have one output and can only forward the incoming probabilities to this output. This leads to the effect that the probabilities arriving from state 4 and 7 at state 8 are depending on the probabilities distributed at state 1 to the states 2 and 5. For example, if the training data contains more “code” than “cafe”, state 1 will always assign more probabilities to state 2 then to state 5. Even if the input is “cafe”, a HMM or MEMM model will follow the state sequence for “code”. In contrast, CRFs are designed to avoid this labeling bias problem by optimizing the state sequence over the whole input sequence.

In contrast to CRFs are ANNs, SVMs, and DTs classifying single objects. When

using these classifiers for classifying logical structures, each line is classified by its own. Information about the reading order of lines is lost. To compensate this information loss two solutions are possible: either include context information in the feature vector or extend the algorithm to handle sequences of instances. In order to include context information in the feature vector, a feature vector for classifying a single line is extended with all features describing the adjacent lines. The resulting feature vector for line l is then $\vec{f}_l = \{F_{l-k}, \ldots, F_{l-1}, F_l, F_{l+1}, \ldots, F_{l+k}\}$ with F_i denoting all features calculated for line i. The second approach of extending an instance classifier is done by combining the instance classifier with a CRF. In this case the instance classifier uses only the features for a single line, but the classification result is used by a CRF to make the final decision.

2.5.1 Selection of Machine Learning Algorithm

All selected machine learning approaches are tested in order to select the best fitting algorithm. For ANN and DT the MultilayerPerceptron and REPTree implementations of WEKA [Hal+09] are used. LIBSVM [CL11] is used for the SVM algorithm. And for CRF the implementation of Mallet [McC02] is selected.
$ANN^{\pm 5}$, $SVM^{\pm 5}$, and $DT^{\pm 5}$ are denoting the versions using the extended feature vector with the features of the five lines before and after. In addition to the normal DT algorithm also a boosted version is used. Boosting is a common technique to increase the performance of classifiers. It combines multiple iterations of the same classifier trained on different weighted examples. The aim is to improve the overall performance. In this case, the DT is boosted with 10 iterations of AdaBoost [FS96]. CRFs are mainly used with binary features and not with rational ones. The CRF is therefore used without rational features. To test the influence of rational features on a CRF, an additional CRF, denoted with CRF^{all}, is trained with the rational features.

For the classifier selection a small data set consisting of 200 pages of computer science publications is used. An detailed description of the complete test set can be found in Section 2.6. Except for the DT algorithm, which is automatically selecting expressive features, a feature selection based on the predictive

Table 2.1: Comparison of the performance of different machine learning algorithms on the test data set.

Classifier	Accuracy (sd)	AUC (sd)
ANN	0.827 (0.019)	0.821 (0.019)
$ANN^{\pm 5}$	0.862 (0.021)	0.821 (0.019)
SVM	0.870 (0.006)	0.961 (0.006)
$SVM^{\pm 5}$	0.909 (0.006)	0.969 (0.006)
DT	0.838 (0.025)	0.926 (0.010)
$DT^{\pm 5}$	0.811 (0.020)	0.903 (0.014)
boosted DT	0.880 (0.017)	0.942 (0.018)
boosted $DT^{\pm 5}$	0.872 (0.013)	0.926 (0.016)
CRF^{all}	0.623 (0.051)	0.903 (0.014)
CRF	0.898 (0.015)	0.903 (0.014)
ANN+CRF	0.840 (0.028)	0.888 (0.015)
SVM+CRF	0.897 (0.022)	0.959 (0.022)
DT+CRF	0.925 (0.015)	0.980 (0.010)
boosted DT+CRF	0.920 (0.016)	0.963 (0.015)

ability and redundancy of the features [Hal98] is performed in order to reduce the feature set. After feature selection, the parameters for each algorithm are optimized with a random search [BB12] of approximately 200 trials. For each trial a 10-fold cross-validation is performed to estimate the quality of the classifier. After the parameters for each algorithms where found, the data set is used to compare the performance of the different algorithms.

In addition to accuracy, the ROC analysis is used for evaluation of the algorithms. The "Area Under a ROC curve" (AUC) measure is not depending on the class distribution as it is the case for accuracy [LHZ03]. It suites therefore better for comparing classifiers on data sets with skewed class distributions than accuracy. The AUC can only be calculated for each structure type on its own. The multi-class extension described in [HT01] is used to calculate an overall AUC measure for each classifier. This extension is still independent from the

class distribution and is derived from the fact that the AUC is equivalent to the probability that a classifier will rank a positive instance higher than a negative one.

The final results of the comparison are shown in Table 2.1. Figure 2.3 shows the ROC curves for the four best performing classifiers on the four most frequent structure types. The ROC curves for all structure types can be found in Appendix B.2. Comparing the basic classification algorithms without the feature vector extension and classifier combinations, the boosted DT has the best performance with an accuracy of 0.880. Taking the feature vector extension into account, the SVM outperforms the other algorithms with an accuracy of 0.909. It is interesting that the feature vector extension improves the accuracy of the ANN and SVM algorithms but not their AUC measure. The DT algorithm does not profit from the extension. The CRF algorithm has problems with rational features. Hence, the accuracy of the CRF improves when the rational features are removed. Overall, the best performance is achieved with the combination of a DT and a CRF. This combination yields on the test set an accuracy of 0.925 and an AUC of 0.980. Combining the CRF with a DT improves the classification quality, because the rational features are taken into account by the DT classifier. The combinations ANN+CRF and SVM+CRF perform worse, because the probability output of the ANN or SVM does not fit to the CRF algorithm.

The ROC curves in Figure 2.3 partly explain the advantage of the DT+CRF approach. The graphs for “Text” and “Enumeration” clearly show better ROC curves for the DT+CRF approach than for the other classifiers. The main reason is the ability of a CRF to describe the context of a line with hidden states. This allows the CRF to identify an item of an enumeration with many lines correctly. Even then, when the local context of such a line does not contain a bullet and the line seems to be a text line with a larger indentation.

2.5.2 Efficient Creation of Reference Data

Any structure analysis approaches requires training or validation examples for learning or verifying the created model. An efficient way of generating example

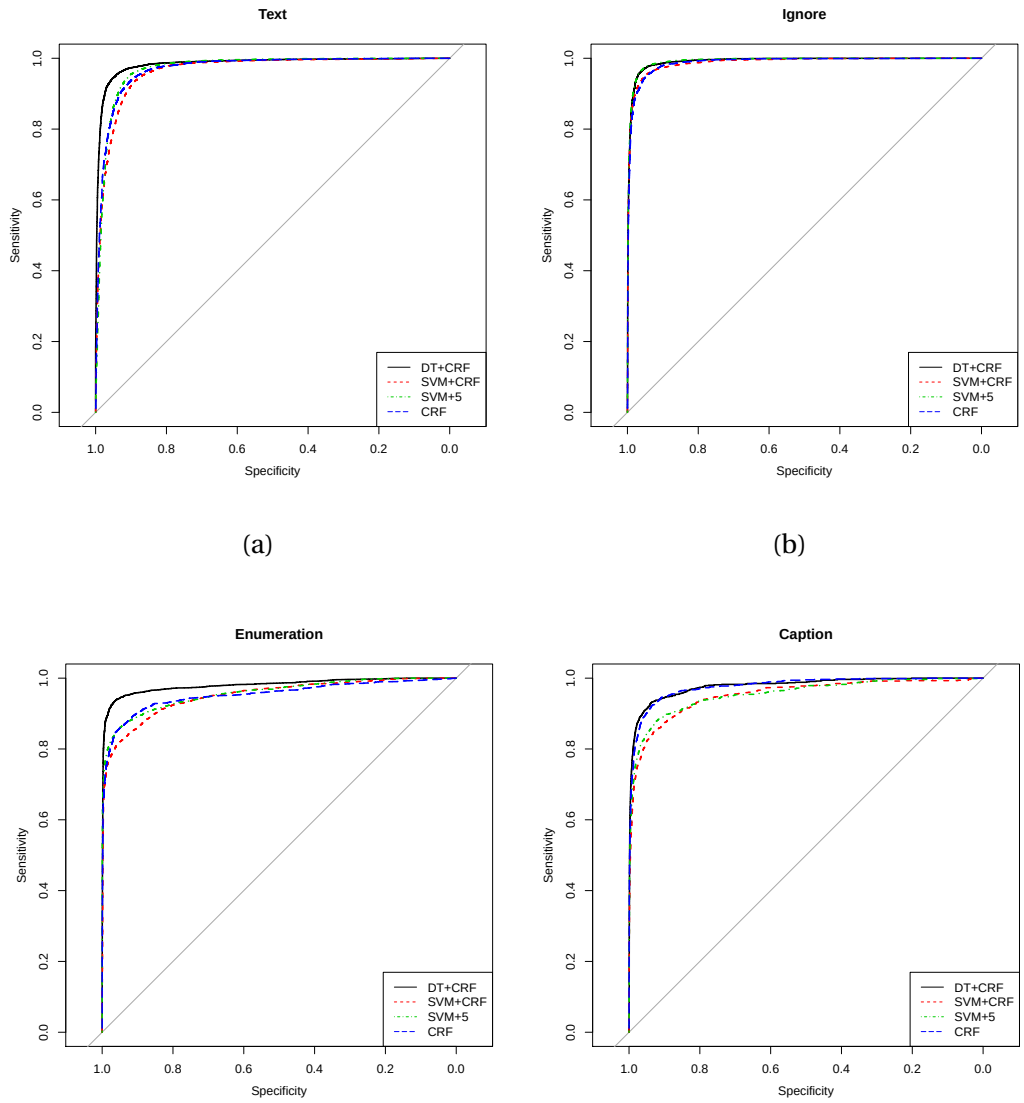


Figure 2.3: ROC-curves of different classification approaches on the test data set. Only the ROC-curves for the four most frequent structure types and the four best performing classifiers are shown here.

data is the combination of automatic structure analysis with learning from user interactions. A user is correcting or annotation structures in documents and the structure analysis process learns from the interactions by adding the corrected documents to the example data. In case the document collection is changing over time, for instance in archives or mailrooms, users responsible for processing new documents can verify and, if necessary, correct the recognized structures. An additional benefit of this process is that it automatically generates the data required to adapt the analysis model to the changes in the documents.

When starting without any example data, the required example data can be generated in an iterative process. Each iteration starts with the analysis of a new set of pages with the analysis model generated in the previous step. Afterwards the user is correcting the automatic result and the corrected pages are added to the example data. The last step of an iteration is the update of the analysis model with the created example data. This process reduces the manual efforts in creating the example data, because only the wrong recognized structures must be corrected. In addition, the iterative approach allows direct feedback of the model quality. In each step the user is able to judge the quality of the analysis models by looking at the number and types of corrections. This allows the user to stop creating example data, when the model quality fits to the users need. An open question in this scenario is the number of examples the user has to annotate during an iteration to get an optimal progress in model quality.

To evaluate this approach the quality of the DT+CRF approach is evaluated over different training sets. For the training set a annotated collection of paper is used and subsets of different size between eight and 256 pages are randomly created for the evaluation. The calculation of the classification error and the F-measure of the different labels a 10-fold cross-validation is used. The results are shown in Figure 2.4 and Figure 2.5. More details can be found in Figure B.2 in the Appendix B.3. Figure 2.4 shows the average relative error, the percentage of miss-classified lines, over the size of the training set in pages. The relative errors range from 24 % of miss-classified lines for eight training pages to 7 % when using 512 pages. Interesting is the approximately linear trend of the relative

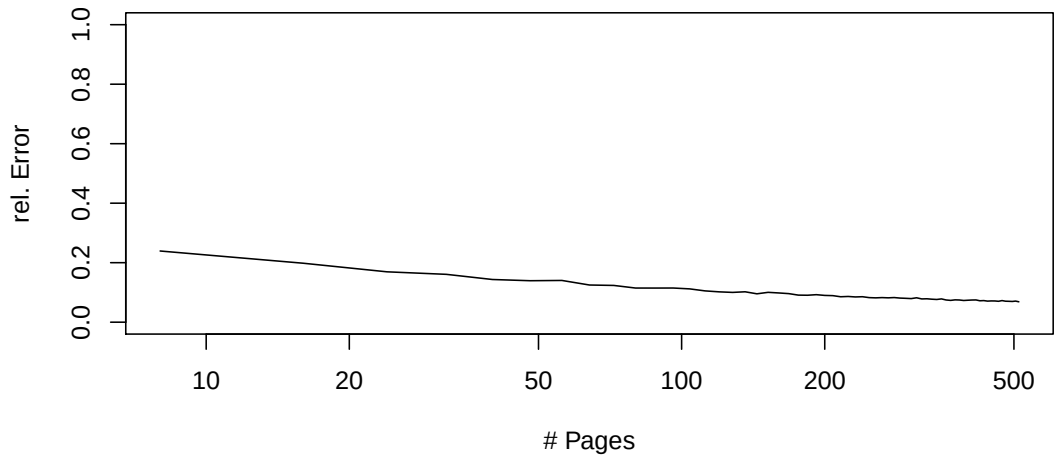


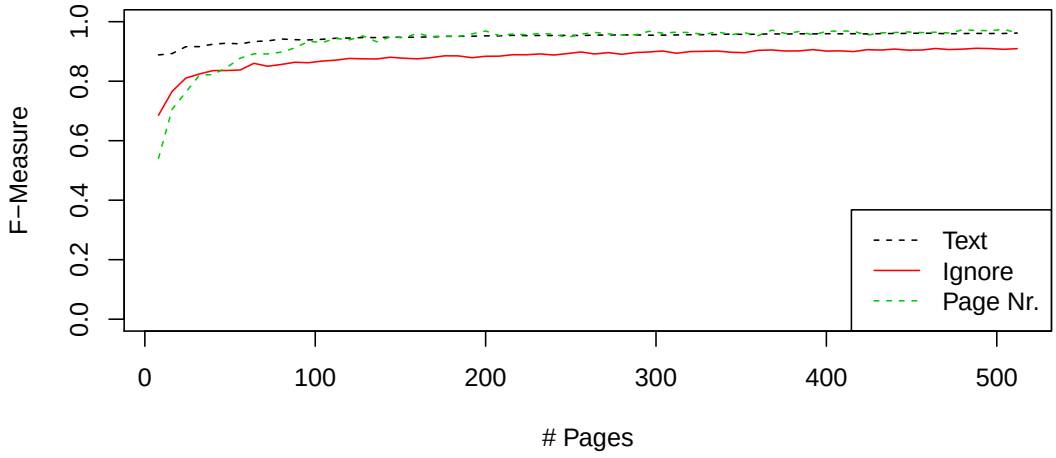
Figure 2.4: Relative error of DT+CRF model at different sizes of the training set.

error over the logarithmic x-axis. This means that the relative error is decreasing exponentially with the number of pages used for training. From this follows that the size of the training set has to increase exponentially in order to keep the number of corrections between two iterations constant. For instance, the size of the training set could be doubled from iteration to iteration.

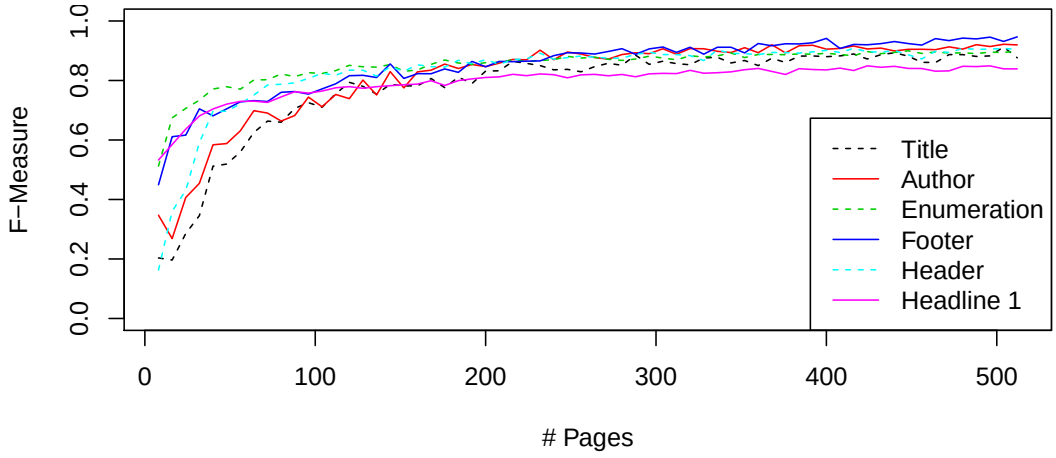
The results in Figure 2.5 show three different groups of structure types. The first group shown in Figure 2.5a can be learned by the classifier with a small number of examples. A classifier trained with approximately 64 pages can achieve F-measures above 0.8 for these types. Increasing the training set further yields only very small improvements on these types. The second group of structures shown in Figure 2.5b is not as easy to learn as the first one. The classifier needs more than 150 example pages to achieve a F-measure above 0.80 for these structure types. Figure 2.5c shows the third group of logical structures, which are hard to learn. The F-measure increases with larger training sets, but compared to the other two groups the F-measure increases slowly on a lower level.

2.6 Evaluation

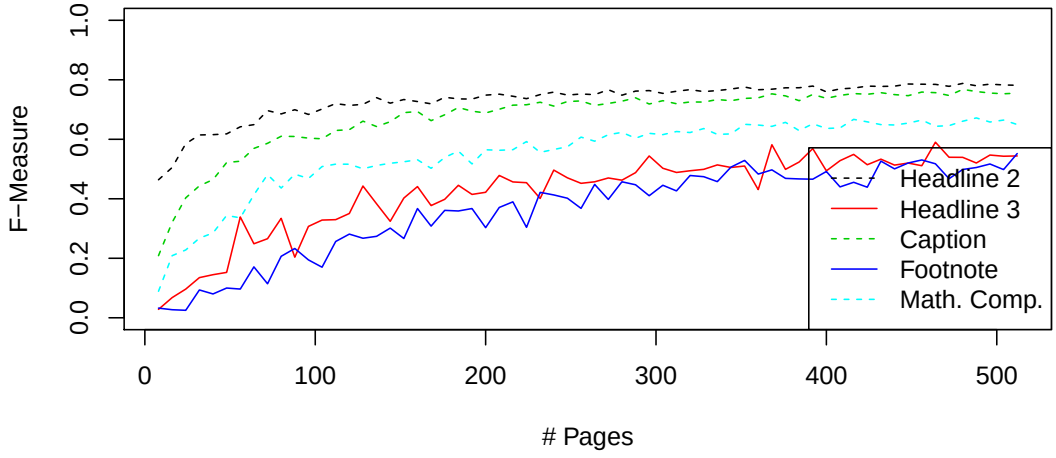
The logical structure analysis approach is evaluated on two different collections of documents. The first collection consists of 250 papers with 1995 pages from proceedings of the computer science conferences IEEE InfoVis 1995-2005, IEEE



(a)



(b)



(c)

Figure 2.5: F-measure of the different structure types of number of pages using a DT+CRF model.

Vis 1990-2005, SIGMOD 1997-2007, ACM SAC 2005-2008, VLDB 2000-2008 and of articles from INTEGERS Electronic Journal of Combinatorial Number Theory vol. 0-9. The structural elements of the papers are tagged with one of the following types, the number in parenthesis shows the fraction of lines for the particular type in percent: Title (25.4), Author (1.1), Headline 1 (1.4), Headline 2 (0.8), Headline 3 (0.1), Page Header (0.6), Page Footer (0.3), Page Number (0.4), Running Text (60.7), Enumeration (11.8), Mathematical Component (1.1), Caption (2.3), Footnote (0.3), and Ignore (17.7). The Ignore type is used for text that does not fit into one of the other categories, for instance, text appearing in figures or in tables.

The second collection consists of 50 product manuals of different products from various manufactures that are accessible on the web. The manuals are collected via a standard search engine using the keyword "manual" and narrow down the results to PDF documents from home pages of consumer electronics manufactures. An important difference between product manuals and papers is the lack of formal formatting guidelines, which exists for each journal or conference publication. The following structure types are annotated in the manuals, the number of parenthesis is the fraction in percent on all lines: Title (0.1), Headline 1 (0.1), Headline 2 (0.1), Headline 3 (4.4), Page Header (1.0), Page Footer (9.9), Page Number (0.7), Running Text (36.2), Enumeration (28.1), Caption (0.2), Footnote (0.0), and Ignore (26.5).

With the labeled training collection a new structure analysis is trained and compared to the methods of Nakagawa et al. [NNS04] and Ratté et al. [RNM07]. The method of Nakagawa et al. describes an rule-based algorithm for extracting structure information and mathematical components from papers. The method of Ratté et al. uses syntactic information and a grammar to identify titles, headlines and enumerations in documents. These two different approaches are compared on the scientific data set to the DT+CRF approach. The DT+CRF approach is evaluated with a 10-fold cross-validation. The methods of Nakagawa et al. [NNS04] and Ratté et al. [RNM07] do not use training and therefore cross-validation is useless for them. For all methods, the precision, recall, and F-measure are calculated for each label on text lines. For the DT+CRF the AUC

Table 2.2: Performance of different logical structure analysis systems on the example paper collection.

	Nakagawa et al.		Ratté et al.		DT+CRF		
	Pr/Re	F_1	Pr/Re	F_1	Pr/Re	F_1 (sd)	AUC (sd)
Title	0.78/0.97	0.86	0.49/0.62	0.55	0.94/0.93	0.93 (0.04)	1.00 (0.00)
Author	0.74/0.86	0.79			0.92/0.95	0.93 (0.06)	1.00 (0.00)
Headline*	0.49/0.90	0.63	0.69/0.54	0.60	0.98/0.96	0.97 (0.01)	1.00 (0.00)
Headline 1	0.64/0.53	0.58			0.92/0.88	0.90 (0.02)	1.00 (0.00)
Headline 2	0.31/0.32	0.32			0.82/0.88	0.84 (0.03)	1.00 (0.00)
Headline 3	0.02/0.42	0.05			0.84/0.52	0.62 (0.11)	1.00 (0.00)
Header	0.08/0.81	0.14			0.96/0.92	0.94 (0.02)	1.00 (0.00)
Footer	0.16/0.96	0.27			0.99/0.98	0.98 (0.01)	1.00 (0.00)
Page Nr.	0.97/0.97	0.97			0.98/0.98	0.98 (0.01)	1.00 (0.00)
Running Text					0.97/0.97	0.97 (0.00)	0.99 (0.00)
Enumeration			0.55/0.35	0.43	0.93/0.92	0.92 (0.02)	0.99 (0.01)
Math. Comp.	0.99/0.56	0.72			0.81/0.70	0.75 (0.05)	0.99 (0.02)
Caption					0.85/0.79	0.82 (0.04)	0.99 (0.01)
Footnote					0.80/0.61	0.67 (0.09)	0.99 (0.01)
Ignore					0.92/0.93	0.92 (0.02)	0.99 (0.00)

measure is calculated additionally. Calculating the AUC measure for the other method is not possible, because calculating the AUC requires a classifier to provide probabilities or scores for the different labels, which the method of Nakagawa et al. and Ratté et al. are not able to calculate. The results on the paper collection are shown in Table 2.2. Table 2.4 contains the results on the product manuals for the DT+CRF classifier trained on the manuals. Finally, the Table 2.3 show the results of the different approaches trained for the papers on the product manuals collection.

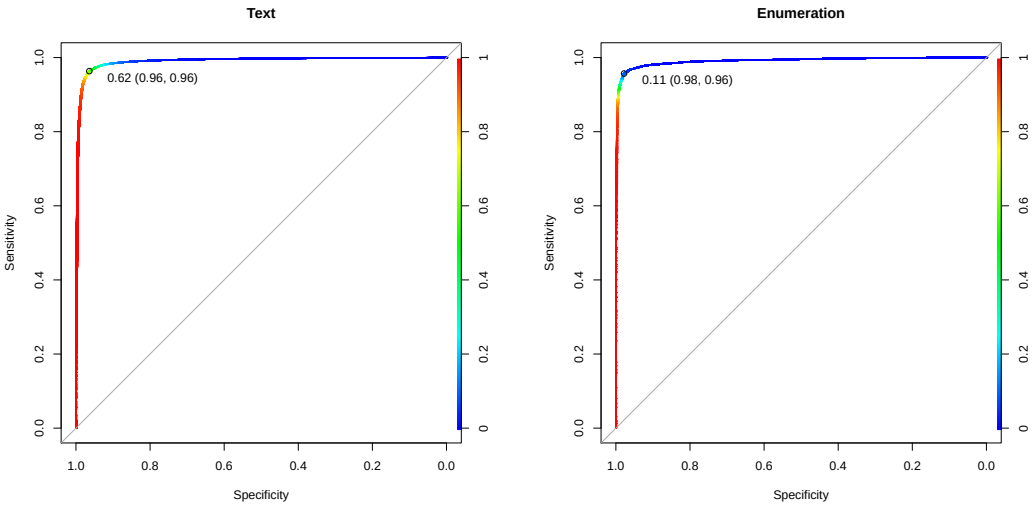
On the paper collection the method of Nakagawa et al. achieves an accuracy of 0.70, Ratté et al. an accuracy of 0.85, and the DT+CRF an accuracy of 0.95 (sd=0.01). The AUC of the DT+CRF approach is 0.99 (sd=0.00). A comparison of the F-measures in Table 2.2 shows that the method of Ratté et al. has the worst performance on all structure types. The results of Nakagawa et al. show an optimization of the rule set to achieve high precision values for mathemati-

cal components but the focus of the other types is on high recall. The DT+CRF approach is able to identify headlines with a good performance, but has problems to differentiation the levels of headlines from each other. The DT+CRF performs very well on the majority of structure types, but has problems with mathematical components, captions, and footnotes. The DT+CRF approach performs in general much better than the approach of Nakagawa et al. Only for authors, page numbers, and mathematical components is the performance comparable between these two approaches.

Interesting is the difference between the AUC values and the F-measure of the DT+CRF classifier. The main reason lies in the property of the AUC measure to take into account all possible cut-off values for the decision, whereas a concrete classification has to use a single cut-off value. With different optimal cut-off values for the different structure types the performance of the different types differs. For example, the ROC curves and the optimal cut-off value for “Title” and “Enumeration” corresponding to the AUC measure are shown in Figure 2.6. The ROC curves for the remaining structure types can be found in Figure B.3 in Appendix B.4.

Table 2.3 and Table 2.4 show the performance of the structure analysis algorithms on the product manual collection. Table 2.3 shows the results of the analysis model created applied to the product manual collection. In this case the approach of Nakagawa et al. achieves an accuracy of 0.61, Ratté et al. an accuracy of 0.67, and the DT+CRF an accuracy of 0.65 (AUC=0.81). The ROC curves for the DT+CRF classifier are shown in Figure B.4 in Appendix B.4. In general, the analysis model for papers does not work for product manuals. Interesting is that observation that the structure elements page number, running text, enumeration, and ignore achieve a much higher F-measure than the other structure elements, which is not the case for headlines. This results from to the fact that headlines in product manuals are typeset differently from headlines in papers.

In Table 2.4 the results of the DT+CRF trained on the product manuals evaluated with a 10-fold cross-validation are shown. The algorithm achieves an accuracy of 0.86 (sd=0.01) and a AUC of 0.87 (sd=0.11). The corresponding ROC



(a) (b)

Figure 2.6: ROC curves of the DT+CRF classifier on the paper collection. The color of the curve shows the cut-off value for classification. The marked point on the curves correspond to the optimal cut-off value.

Table 2.3: Performance of the model for scientific papers on product manuals.

	Nakagawa et al.		Ratté et al.		DT+CRF		
	Pr/Re	F_1	Pr/Re	F_1	Pr/Re	F_1	AUC
Title	0.90/0.31	0.46	0.73/0.38	0.50	0.06/0.05	0.06	0.88
Headline*	0.05/0.06	0.06	0.05/0.04	0.04	0.49/0.31	0.38	0.93
Headline 1	0.01/0.01	0.01			0.04/0.26	0.07	0.92
Headline 2	0.16/0.00	0.00			0.13/0.06	0.08	0.93
Headline 3	0.03/0.00	0.00			0.35/0.01	0.03	0.88
Header	0.03/0.38	0.05			0.27/0.31	0.28	0.96
Footer	0.03/0.21	0.05			0.17/0.01	0.01	0.96
Page Nr.	0.49/0.79	0.61			0.59/0.66	0.62	0.99
Running Text					0.70/0.72	0.71	0.85
Enumeration			0.40/0.04	0.04	0.81/0.55	0.65	0.87
Ignore					0.62/0.84	0.71	0.91

Table 2.4: Performance of the DT+CRF on product manuals.

	DT+CRF		
	Pr/Re	F_1 (sd)	AUC (sd)
Title	0.28/0.12	0.16 (0.28)	0.76 (0.41)
Headline*	0.82/0.80	0.81 (0.03)	0.99 (0.01)
Headline 1	0.65/0.49	0.55 (0.13)	0.99 (0.01)
Headline 2	0.71/0.55	0.68 (0.07)	0.99 (0.00)
Headline 3	0.71/0.73	0.72 (0.04)	0.98 (0.01)
Header	0.88/0.83	0.86 (0.05)	1.00 (0.00)
Footer	0.90/0.87	0.89 (0.06)	0.98 (0.03)
Page Nr.	0.92/0.90	0.91 (0.06)	1.00 (0.00)
Running Text	0.86/0.88	0.87 (0.01)	0.96 (0.02)
Enumeration	0.89/0.86	0.87 (0.04)	0.97 (0.02)
Ignore	0.87/0.89	0.88 (0.02)	0.98 (0.01)

curves can be found in Figure B.5 in Appendix B.4. The model trained on the product manuals perform much better than the models created for papers as shown in Table 2.3. There are still problems in the recognition of titles and headlines. The distinction of headlines in the different levels is even a bigger problem than for papers. Overall the classifier does not achieve the high quality for product manuals than for papers.

2.6.1 Discussion

The comparison of the different approaches on the paper collection clearly shows the power of the machine learning approach. It achieves on papers with 0.95 a much better accuracy than the two other approaches with 0.70 or 0.85. In addition, the DT+CRF approach is easily adaptable to different documents, as shown with the product manuals. All the structure analysis models are adapted to their document collection and uses information about the general formatting within the collection for the analysis. A different document type is likely to fol-

low a different formatting convention, which misleads the analysis models. This can be observed in Table 2.3 where the models created for papers are applied to product manuals. Nevertheless do papers and manuals have some similarities, which enables to DT+CRF model to recognize some structure elements with an F-measure of at least 0.60.

An advantage of the machine learning approach is the ability to automatically create an analysis model from example documents. Creating a specific analysis model improves the accuracy on the product manual collection from 0.65 with the paper model to 0.86. Using grammars or rules would have required a user to adapt the model manually. Event with the adapted model does the DT+CRF not achieve similar accuracy values than for the paper collection. This is not a surprise, because the formatting of papers is strongly regulated, whereas such a regulation does not exist for product manuals.

2.7 Summary

Logical structure analysis uses geometry, formatting, and simple content features to recognize logical structures. Existing approaches uses several types of rules or grammars, which are complex to maintain. Adapting such a system to a new document type means creating a new rule or grammar system, because the formatting conventions differs for different document types. A solution to this problem is the usage of machine learning algorithms that can adapt itself automatically to a new set of example documents.

Due to the sequential nature of documents, the default machine learning algorithms, such as ANNs, SVMs, or DTs, have problems with the recognition. The mixture of nominal and rational features is an additional problem for many machine learning algorithms. It turned out that the best solution is a combination of a DT algorithm, capable of dealing with nominal and rational features, with a CRF, which models sequences of lines. This combination also outperforms the tested rule and grammar based approaches on the example data sets.

An additional advantage of a machine learning approach is the ability to create a model with a very few examples. As expected have such models a lower

quality than a model created with many examples, but they are appropriate for an iterative process, which successively creates additional examples. This process reduces the manual efforts in creating large example collection, which is needed for any kind of analysis model, because event rules or grammars are requiring examples for verification.

A different result of the comparison of the different structure analysis approaches is the observation that the analysis models only works for document they were created for. Applying such models to a different document type must fails, because the formatting conventions between the different document types differ. The quality of logical structure analysis depends for the same reason on the variance of the formatting within a document collection. On collections with very strict formatting guidelines, such as papers, the analysis models are likely to achieve a better quality than for document collections without these guidelines, such as product manuals.

Chapter 3

Methods for Functional Structure Analysis

This chapter describes the analysis of functional structures in documents. The presented technique extends the framework for logical structure analysis, described in the previous chapter, is extended to extract the functional structure of documents. The chapter starts with a motivation for the recognition of functional structures and a discussion of related work. Then features and methods for the recognition of functional structures are presented. The suggested approach is evaluated and finally the chapter is summarized.

3.1 Motivation

The physical and logical structures describe the physical layout and the logical elements of a document, but both types of structure are mainly independent of the document content. This is, for example, reflected in the features used for logical structure analysis. The textual features used are focusing on common patterns without deeper meaning, for instance the type of characters at the beginning of a line. In contrast, the functional structure is related to the content and expresses the organization of documents. Consequently, the functional structure is better suited to represent documents for humans than the

logical structure. For instance, logical structure can recognize running text, but cannot tell whether this text belongs to an introduction or a conclusion.

One application of the functional structure is to learn the typical organization of papers in a conference or journal, which could help a new author in writing a paper for a community. At least, an outline of a typical paper can be extracted. A different motivation is the ability to improve document retrieval in collections of similar documents. Users could be enabled to search in specific parts of a document. This would be a more user centric search interface as it is possible with the logical structure alone.

A requirement of the functional structure analysis is a similarity between documents with respect to their functional elements. In contrast to the physical and logical structure is the functional structure not expressed with visual properties, such as geometry or formatting, but with content. A function, such as a problem description, is expressed with a specific vocabulary and word usage. In order to be able to automatically analyze the functional structures in a document collection, the collection must contain documents of similar functional structures. Document collections fulfilling these requirements contain usually documents of the same type. For instance, papers usually contain introduction and conclusion. Other common functional elements of papers are related work, methodology, or evaluation.

The functional structure analysis technique combines part of speech (POS) tagging and machine learning in order to create an analysis model. The POS tagging detects the word types of single tokens and allows, for instance, the distinction of verbs from adjectives. The word type is important, because words can point to different functions depending on their type. For instance, verbs are describing actions whereas nouns identify subjects and objects. With the combination of word and POS as features, machine learning techniques are used to create analysis models for functional structure analysis.

3.2 Related Work

The analysis and recognition of functional structures is depending on the application. In some cases is a functional structure recognizable with the logical structure analysis methods. For instance, the abstract of a paper starts with a special headline followed by some paragraphs. Such functional structure are often recognized together with the logical structure of a document. In case of papers this are often the abstract and biography information [KDK00; KLT01; RB06]. These approaches are limited to functional structures with a special visual representation. The recognition of structures that are not distinguishable by visual or simple textual properties, such as keywords or regular expressions, is not possible with techniques used for logical structure analysis.

A related approach to functional structure analysis is the recognition of the table of contents (TOC). Techniques for automatic recognition of TOCs are used for automatic index generation and conversion of document formats. For example the TOC can be used to create an index of the content when converting PDFs into the EPUB format, which is often used by portable display devices. In contrast to the analysis of functional structure, is the aim of the TOC recognition to automatic detection of the outline of a document. TOC recognition techniques usually use a two step approach. In the first step the entries of the TOC of a document are recognized and in the second step these entries are verified with the headlines and titles found in the content. The recognition of entries in a TOC is often based on rules and heuristics [DM09; MMS10; Tsu+01], or data mining techniques [BES01; Gao+09; STK95]. Even specific part-of-speech taggers are created for parsing TOC entries [BPV00]. For the second step several different heuristics are used to select most likely TOC entries. Typical heuristics require the TOC entries to be ordered with increasing page numbers, or that the text of the headline must appear on the corresponding page. All these methods are requiring a TOC in a document. Unfortunately, papers or many kind of reports are short documents and therefore do not contain a TOC. This makes these approaches of TOC recognition useless for the target documents of this thesis.

As discussed in the motivation of this chapter is the functional type of an element depending on its content. An obvious approach is therefore the usage of information retrieval techniques in combination with a k-nearest neighbor classifier. A set of training texts is indexed in the information retrieval system and the functional class is assigned to an unknown text by a majority vote over the k most similar examples. Different techniques for indexing and retrieving documents exists [MRS08] but they have all in common to retrieve documents with similar content. In the case of functional structure analysis, the aim is not to find similarities in terms of content but in terms of functions.

The approaches based on formatting, keywords or regular expressions are not able to identify arbitrary functional structures. Structures that do not start with a common keyword or do not have a special formatting are not recognizable with these approaches. In order to overcome this problem, the textual content must be taken into account. The TOC recognition approaches are able to recognize the outline of a document but papers or reports usually do not have a TOC, which makes these approaches useless for these documents. The information retrieval techniques, which are analyzing the content, are designed to find similar documents in terms of content and not in terms of functional structure. Instead of using the information retrieval similarity measures, machine learning is used for functional structure analysis, which is able to identify features separating the different functional elements from each other.

3.3 Features for Functional Structure Analysis

The function of a text within a document is determined by its content. The physical structure is not relevant for the functional structure analysis whereas the logical structure is of interest. The different functional parts of a document are usually expressed in different sections or chapters. The section bounds detected with logical structure can therefore be helpful in recognizing functional element. Using the logical structure allows to model for functional analysis to incorporate changes of chapters and sections. In addition, the text processing for extracting textual features can make use of the logical structure and apply

different algorithms for different logical elements. For instance, headlines usually do not contain full sentences and may be skipped by natural language processing algorithms, whereas on running text natural language processing is applied.

The simplest content feature is to use the text itself similar to the default information retrieval pipeline. In order to get meaningful features from text content, the text is transformed into features by splitting the text into tokens. Stemming is applied to words in order to map the different inflected or derived forms to the same feature. Finally, the features are then created according to the bag of word model by counting the different tokens and normalizing the resulting feature vector to the length of 1. The normalization is needed to account for the length of the texts.

This approach uses stemming to merge the different inflection and derivations of a word into one feature. Information about the word type is thereby lost. To integrate this information in the features, an additional POS tagging step is necessary that detects the part of speech of each word. Together with the stemming the part of speech the different derivations of a stem are distinguishable but its inflections are put together into one feature.

3.4 Learning Functional Structures

A special problem of functional analysis is the large number and the sparseness of features created during the feature extraction process. The number of features is depending on the different word and part of speech combinations in the document collection. As the frequency of words is following a Zipf distribution [MRS08] and a few words are very frequent but the majority of words are infrequent, which results in a spares distribution of the corresponding features.

In classical information retrieval systems, the most relevant documents for a query are selected with the cosine similarity measure. The cosine similarity has many advantages for information retrieval, for instance is it independent from the length of documents and queries but measures the angle between documents and queries in the term space. In addition, with the cosine similarity

to most similar documents to a query can be calculated efficiently with posting lists [MRS08], because only the terms appearing in a query are relevant for the cosine similarity. This property fits best with a k-nearest neighbor (k-NN) classifier. With a k-NN classifier the cosine similarity measure can be used for functional structure analysis, by classifying an element with a majority vote over the k most similar examples in the training set.

With the cosine similarity the feature selection process is important, because the similarity is not able to learn important combinations of features but every feature gets the same weight. Many machine learning algorithms are able to learn useful combinations of features, but they have problems with high dimensional and sparse feature sets [Kot07]. To address this problem, the number of features and the sparseness must be reduced. The typical ways to reduce the dimensionality is by filtering the statistically not relevant features or using projection techniques, such as the principal component analysis (PCA). The PCA maximizes the variance in the lower dimensions, which results in a low variance in the upper dimensions. The dimensionality can then be reduced by removing features with a low variance. A feature with a low variance can be discarded, because its value is more or less constant and hence does contain less information. In addition, removing the low variance features also reduces the noise in the data.

3.4.1 Selection of Machine Learning Algorithms

For the selection of the machine learning algorithm for functional structure analysis several algorithms might be possible. In order to find good solution several algorithms are tested. The test is mainly using the same algorithms than tested for logical structure analysis. The naive Bayes classifier is included in the test, because it is able to deal with large and spare feature sets [Kot07]. To compare the results of the machine learning algorithms with information retrieval techniques, a k-NN classifier with a cosine distance is included in the test. The DT and ANN are using the REPTree and MultilayerPerceptron implementation of WEKA [Hal+09]. WEKA is also used for the naive Bayes classifier. Finally, the SVM classifier uses the implementation of LIBSVM [CL11].

Table 3.1: Performance of different machine learning techniques on the example data set.

	Filtering		PCA	
Classifier	Accuracy (sd)	AUC (sd)	Accuracy (sd)	AUC (sd)
k-NN	0.34 (0.03)		0.60 (0.03)	
k-NN^{POS}	0.34 (0.03)		0.56 (0.03)	
Naive Bayes	0.54 (0.04)	0.67 (0.06)	0.49 (0.03)	0.66 (0.04)
Naive Bayes^{POS}	0.63 (0.04)	0.63 (0.04)	0.33 (0.04)	0.58 (0.04)
boosted DT	0.49 (0.03)	0.58 (0.06)	0.49 (0.03)	0.62 (0.04)
boosted DT^{POS}	0.50 (0.02)	0.59 (0.05)	0.48 (0.04)	0.60 (0.06)
ANN	0.65 (0.02)	0.68 (0.06)	0.67 (0.04)	0.67 (0.05)
ANN^{POS}	0.64 (0.04)	0.68 (0.06)	0.65 (0.03)	0.68 (0.06)
SVM	0.70 (0.03)	0.88 (0.05)	0.71 (0.04)	0.87 (0.07)
SVM^{POS}	0.69 (0.02)	0.85 (0.06)	0.72 (0.02)	0.87 (0.06)

For each classifier the two feature sets, with and without POS tagging, are tested. In addition, for each classifier the two dimension reduction techniques, filtering irrelevant features and PCA projection, are tested in order to find the best solution for the analysis problem. The parameters for each classifier and the number of selected features are optimized for each feature set separately using a random parameter search [BB12]. The test is using the computer science paper collection used in Section 3.5 with a minimal text length of 300 words.

Table 3.1 shows the performance of the selected algorithms on a test data set. Regarding the dimension reduction technique, the algorithms, except for the k-NN and the naive Bayes, perform equally with both techniques. The k-NN has a clear advantage with the PCA transformed features, whereas the naive Bayes is significantly better with the filtered feature set. The additional information extracted with the POS tagging leads to an improvement of 9 % in accuracy of the naive Bayes classifier, but in general are the other algorithms able to achieve equal accuracy with or without POS tagging. The best performing algorithm is the SVM with an accuracy around 70 %. The second best algorithm is the ANN

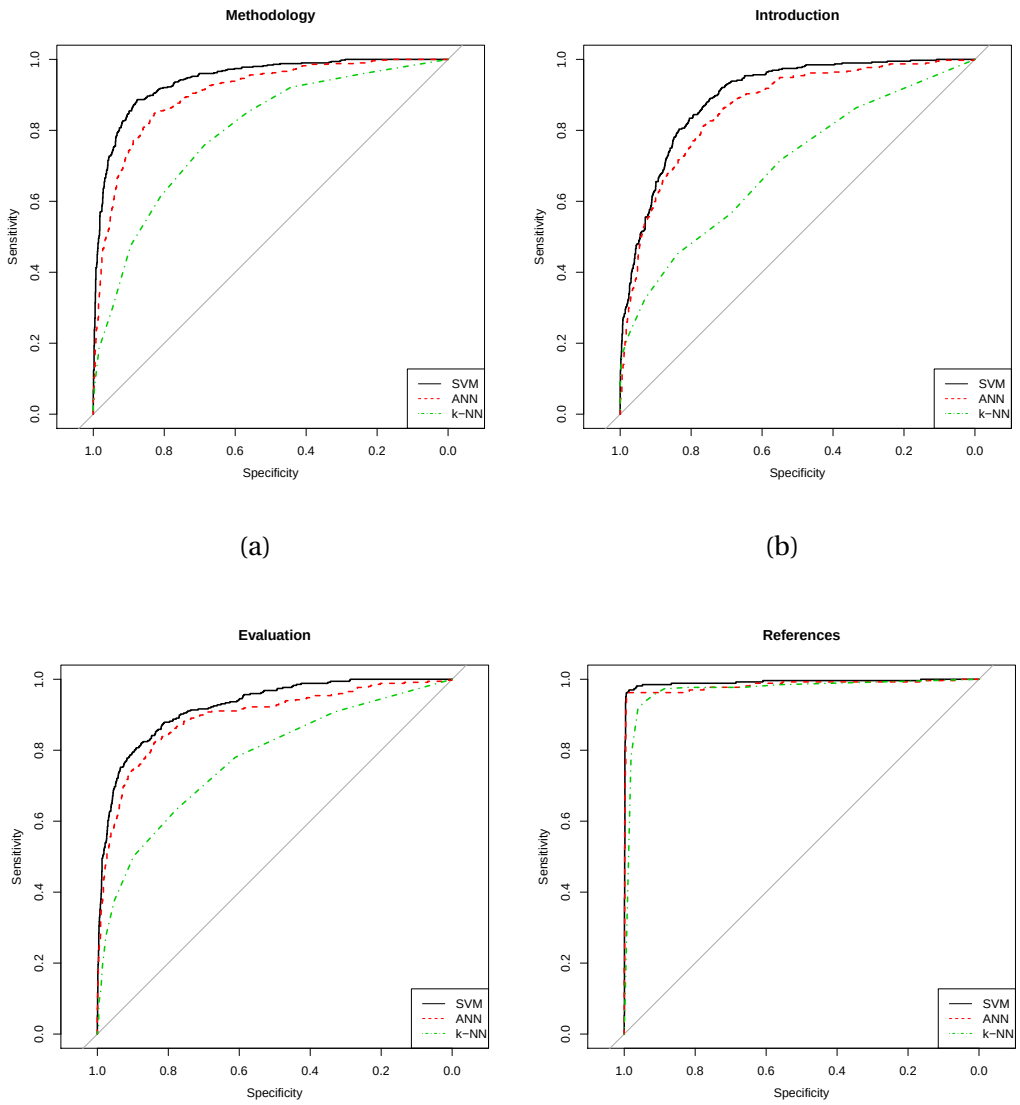
with an accuracy around 65 % and the other algorithms achieve an accuracy of approximately 60 %. The ROC curves of the three most accurate classifiers and the four most frequent functional structures are shown in Figure 3.1. All ROC curves can be found in Figure C.1 in Appendix C.1. Overall, the results show a clear advantage of the SVM over the other classifiers.

3.5 Evaluation

The functional structure analysis approach is evaluated on the functional structures of papers. Two different paper collection are used. The first paper collections comprise 709 computer science papers containing the 250 papers used for the evaluation of the logical structure approaches plus the papers from the VisWeek conferences Vis/SciVis, InfoVis, and VAST of the years 2009-2012. The second paper collection consists of 661.504 biomedical research articles from the PMC Open Access Subset [Med13] (PubMed) database. The performance of different machine learning approaches is compared in the previous section. This evaluation uses 10-fold cross-validation to asses the performance of the algorithm with the best accuracy, which is the SVM classifier and the feature set using POS tagging and PCA projection.

The ground truth for the evaluation is created by assigning the same functional label to complete sections of the papers. Authors are creating sections to organize the different functional parts of a paper and allow a reader to find interesting parts easily. It is therefore valid to regard a section as a functional part. Each section is assigned to one function. A section is discarded, if its function ambiguous.

Besides the features and the learning algorithms, is the amount of available text an important factor for the recognition of functional units. In order to extract meaningful features the text must have a minimal length. If the text is too short, the influence of a single word is very high and can distract the trained model. On the other side, if the amount of text required for good recognition results is too large, small functional units cannot be detected reliably. In order to test the influence of the minimal text length on the recognition results, the



(a) (b)

(c) (d)

Figure 3.1: ROC curves of the three most accurate classifiers and four the most frequent logical structures. All classifiers are using the PCA project feature set and only the SVM classifier is using the POS tagged feature set.

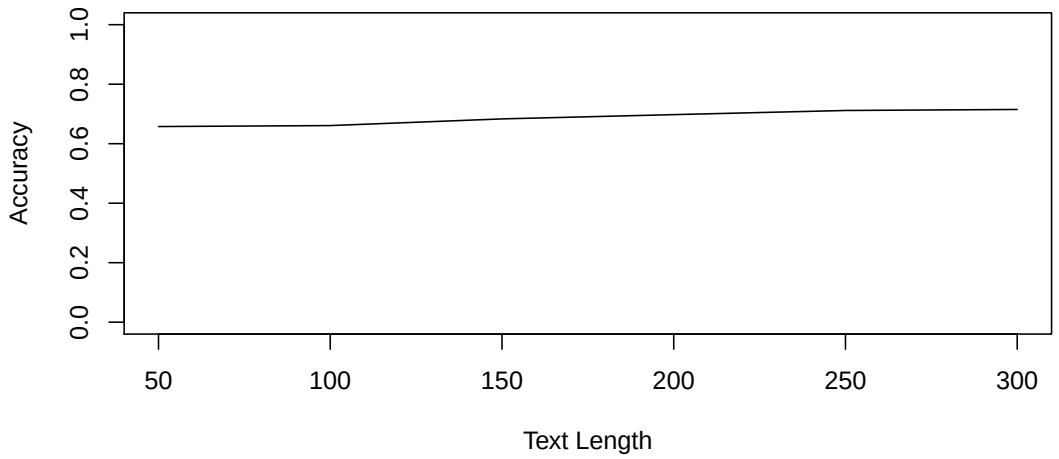
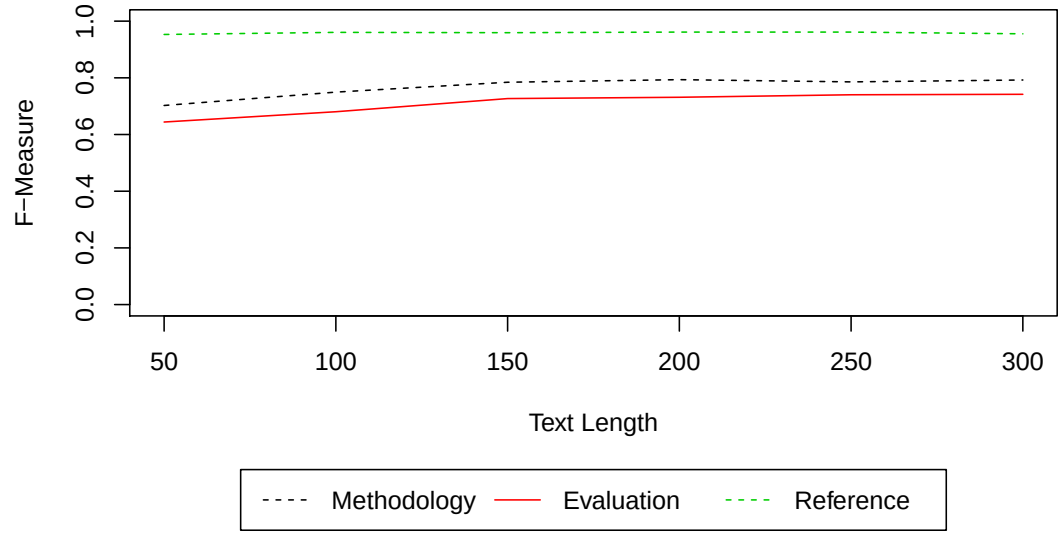


Figure 3.2: Accuracy of the SVM over different text lengths on the computer science data set. The SVM is using the feature set with POS tags and PCA projection.

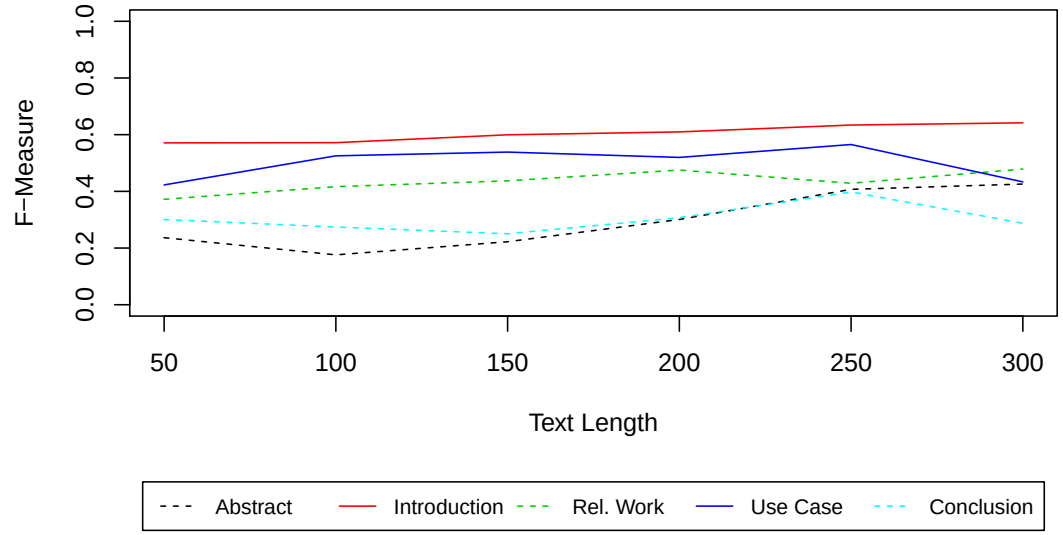
labeled sections are split with different minimal number of words at sentence borders. The classification performance is than computed on the different text lengths.

On the computer science data set the following functional parts are identified with the described method, the proportion of each structure is shown in parenthesis in percent: Abstract (2.0), Introduction (21.0), Related Work (8.7), Methodology (27.0), Evaluation (18.6), Use Case (2.5), Conclusion (5.5), Acknowledgment (0.1), Reference (14.3), and Biography (0.1).

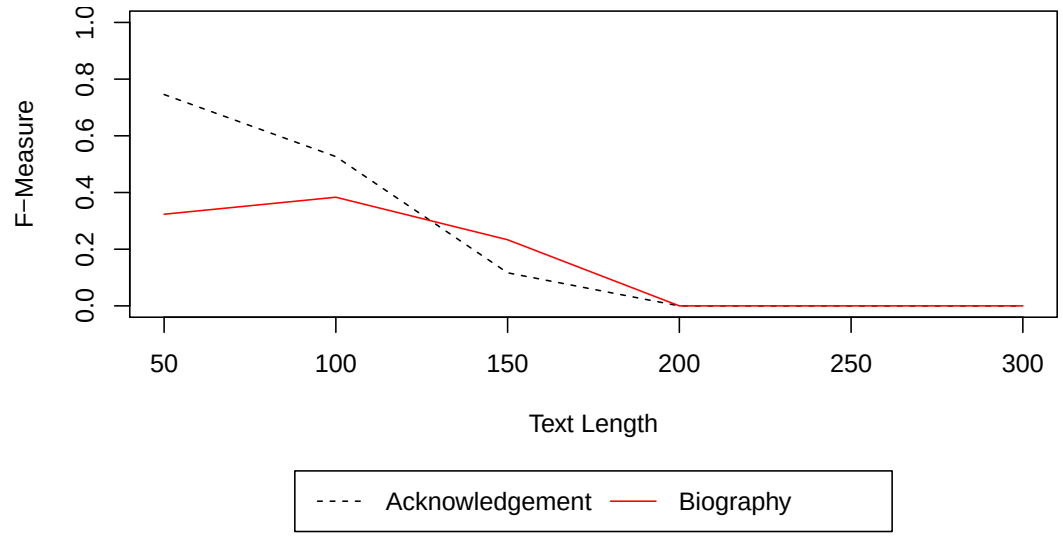
Figure 3.2 shows the overall accuracy of the selected classifier over different text length. The accuracy increases slightly from 0.66 at a text length of 50 words to 0.72 at a text length of 300. The influence of the text length on the F-measure of the different functional types is shown in Figure 3.3. Three groups of features can be recognized. The first group shown in Figure 3.3a achieves a high F-measure but at a text length over 150 words the improvement in F-measure are very small. Figure 3.3b shows the second group of functional types. The recognition of these functional types is more complicated than for the first one. But the F-measure increases slightly with longer texts. The F-measures of the third group are shown in Figure 3.3. For this group of functions the F-measure decreases with longer texts. This can be explained with the length of



(a)



(b)



(c)

Figure 3.3: F-Measure of the different functional types over different text lengths on the computer science data set.

Table 3.2: Performance of the SVM classifier over the different functional structures with a text length of 300 words on the computer science data set.

Classifier	Pr/Re	F_1 (sd)	AUC (sd)
Abstract	0.56/0.38	0.43 (0.33)	0.82 (0.31)
Introduction	0.59/0.70	0.64 (0.06)	0.90 (0.03)
Related Work	0.57/0.42	0.48 (0.12)	0.91 (0.04)
Methodology	0.77/0.82	0.79 (0.03)	0.94 (0.01)
Evaluation	0.75/0.73	0.74 (0.05)	0.92 (0.03)
Use Case	0.48/0.44	0.43 (0.34)	0.92 (0.09)
Conclusion	0.36/0.25	0.29 (0.17)	0.85 (0.06)
References	0.95/0.96	0.96 (0.05)	0.99 (0.02)
Acknowledgment	0.00/0.00	0.00 (0.00)	0.15 (0.32)
Biography	0.00/0.00	0.00 (0.00)	0.05 (0.16)

these sections. Acknowledgments and biographies are usually short compared to the other sections and have rarely more than 200 words.

To evaluate the performance of the approach on the different functional types, the measures precision/recall, F-measure, and AUC are calculated for each type. A text length of 300 is chosen for this evaluation, because this text length yields the best accuracy. The results are shown in Table 3.2. The results show a big difference in the F-measure and AUC for the different types. The fist group of functional types consists of methodology, evaluation and reference. Elements of this group are recognizable. The SVM achieves a F-measure of 0.79 for methodologies, 0.74 evaluations, and 0.96 for references. The second group comprises the types abstract, introduction, related work, use case, and conclusion. The F-measures of this group range from 0.29 to 0.64, which shows that these types are hard to recognize. The third group of functional types consists of acknowledgments and biographies, which are not recognizable with an F-measure of 0.00.

The functional structure of articles in the PubMed data set is less diverse than in the computer science data set. On this data set the following six functional

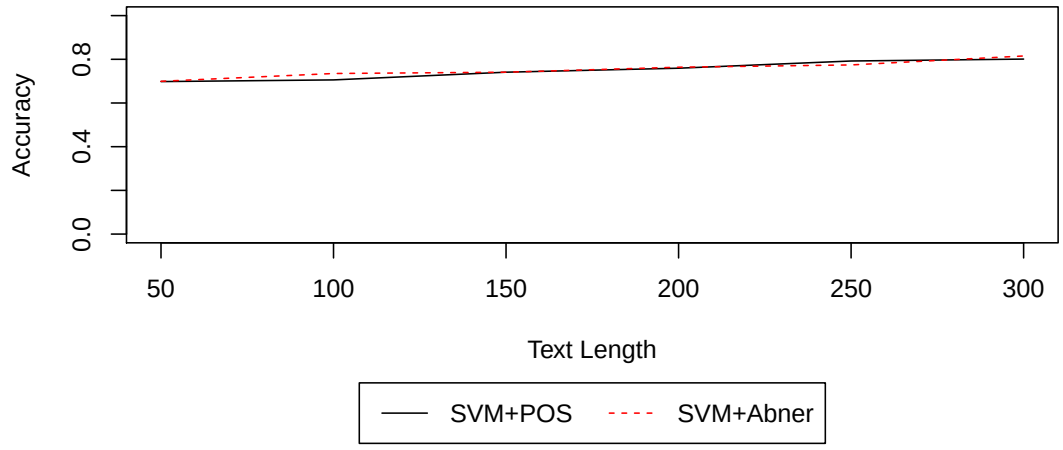
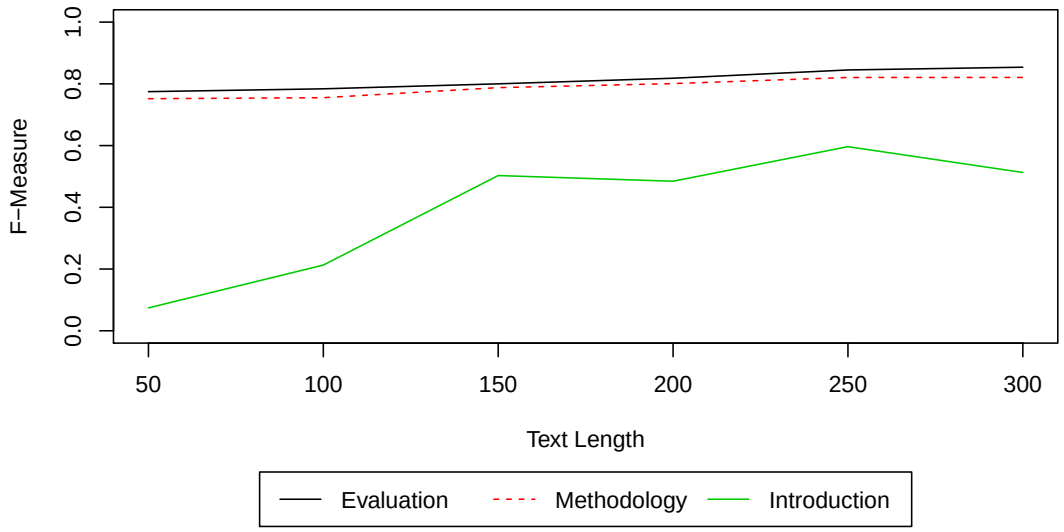


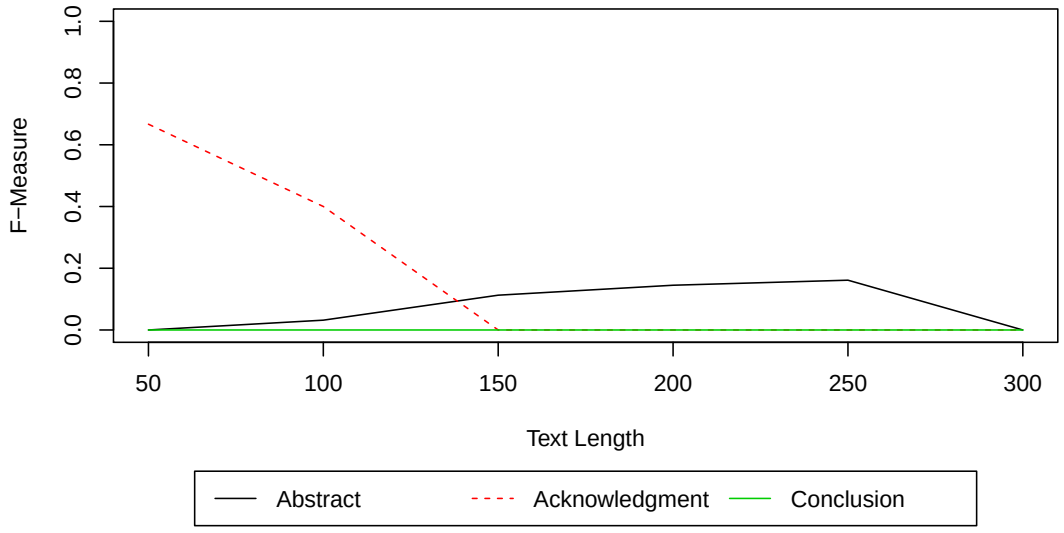
Figure 3.4: Accuracy of the SVM over different text lengths on the PubMed data set. The chart shows the accuracy of both POS and ABNER tags using PCA projection.

parts are identified, the proportion of each structure is shown in parenthesis in percent: Abstract(3.1), Introduction (9.9), Methodology (27.8), Evaluation (58,4), Conclusion(0.7), Acknowledgment (0.1). With the ABNER tagger [Set05] a special named entity recognized for biomedical text exists. In addition to the POS tags, the evaluation tests the results with tags created by the ABNER tagger, because the tags created by the ABNER tagger are more specific for biomedical texts than the general POS tags.

Figure 3.4 shows the accuracy of the SVM with the different taggers over different text length. The accuracy increases slightly with the amount of text for both feature sets. With the POS features the accuracy increased from 0.70 with 50 words to 0.80 with 300 words per chunk and with the ABNER feature set increased the accuracy from 0.70 with 50 word to 0.82 with 300 words per chunk. The influence of the text length on the F-Measure of the different functional types is shown in Figure 3.5 and Figure 3.6. The performance of the SVM is similar with both the POS and ABNER feature sets. The classifier needs only 50 words to be able to identify Evaluation and Methodology. Longer text samples only slightly improve the performance on these types. The Introduction profits most from longer texts. At least 150 to 200 words are required to gain an F-Measure of about 0.5. In total Introductions are complicated to identify

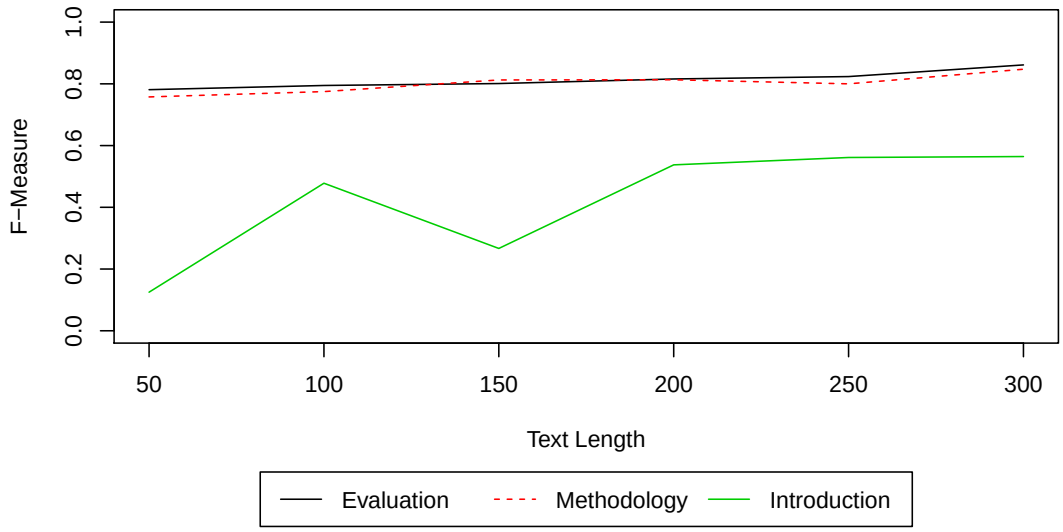


(a)

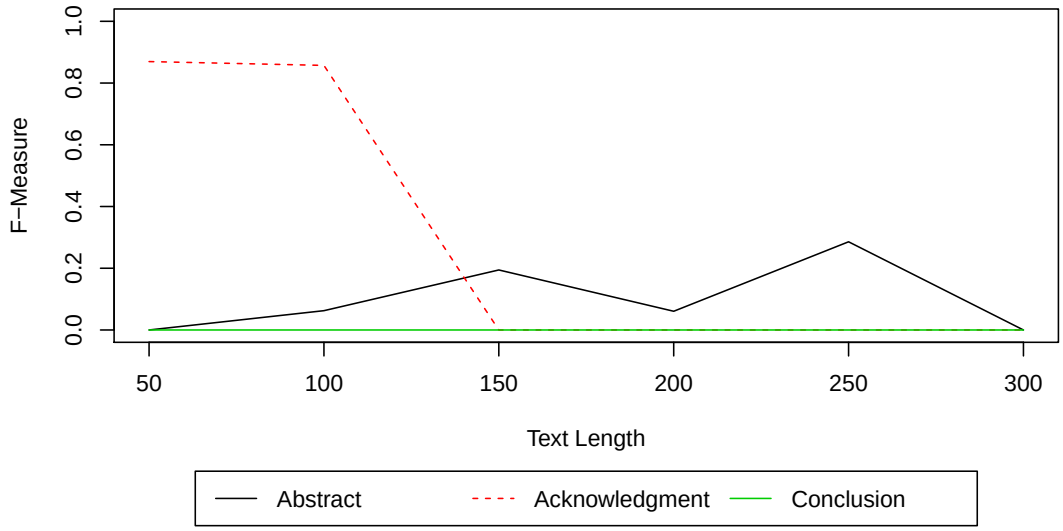


(b)

Figure 3.5: F-Measure of the different functional types in the PubMed data set over different text length using the POS tags.



(a)



(b)

Figure 3.6: F-Measure of the different functional types in the PubMed data set over different text length using the ABNER tags.

correctly. Interesting are the results on the Acknowledgment type. With both feature sets, the SVM is able to identify an Acknowledgment when the minimal text length is 50 words. The ABNER tags achieves here with an F-Measure of 0.87 a significantly better results than the POS tags with 0.67. With a minimal text length of 100 words, the performance of the classifier with the POS features drops to a F-Measure of 0.40, whereas the F-Measure with the ABNER features is more or less stable at 0.86. As almost all Acknowledgment sections have less than 150 words the F-Measure drops in both cases to 0.00 when the minimal text length 150. No or very few examples is also the reason why the classifier is not able to gain noteworthy performance for the functional types Abstract and Conclusion.

Table 3.3 and Table 3.4 show details of the performance of the labels. The corresponding ROC curves can be found in Figure C.3 and Figure C.4 in Appendix C.1. The minimal text length is set to 300, because the classifier performed best at this text length. The results of the two feature sets are similar to each other. The functional types Evaluation and Methodology perform best and gain an F-Measure between 0.82 and 0.86. The AUC (0.89 or 0.92) for Introduction indicates that the classifier is in principle able to classify Introduction with the same performance than Evaluation or Methodology. The discrepancy to the F-Measure (0.48 or 0.55) shows that the classifier cannot weight the different functional types correctly. The remaining three functional types Abstract, Conclusion, and Acknowledgment have usually less than 300 words, which results in no or very few instances for the classification and the F-Measure consequently drops to 0.00.

3.5.1 Discussion

The evaluation of both data sets shows that with longer texts the overall accuracy of the SVM improves. A problem with respect to the text length is length of the functional structures. There is no global optimal choice of text length. If the length is short even short structures can be recognized, but the overall accuracy

Table 3.3: Performance of the SVM classifier on the different functional structures on the PubMed data set with a text length of 300 words and POS tags.

Classifier	Pr/Re	F_1 (sd)	AUC (sd)
Evaluation	0.81/0.91	0.85 (0.05)	0.89 (0.04)
Methodology	0.83/0.82	0.82 (0.04)	0.94 (0.03)
Introduction	0.69/0.38	0.48 (0.22)	0.89 (0.14)
Abstract	0.00/0.00	0.00 (0.00)	0.76 (0.12)
Conclusion	0.00/0.00	0.00 (0.00)	0.29 (0.41)
Acknowledgment	0.00/0.00	0.00 (0.00)	0.00 (0.00)

Table 3.4: Performance of the SVM classifier on the different functional structures on the PubMed data set with a text length of 300 words and ABNER tags.

Classifier	Pr/Re	F_1 (sd)	AUC (sd)
Evaluation	0.82/0.91	0.86 (0.02)	0.89 (0.03)
Methodology	0.85/0.85	0.85 (0.06)	0.96 (0.01)
Introduction	0.72/0.47	0.55 (0.15)	0.92 (0.03)
Abstract	0.00/0.00	0.00 (0.00)	0.85 (0.11)
Conclusion	0.00/0.00	0.00 (0.00)	0.10 (0.25)
Acknowledgment	0.00/0.00	0.00 (0.00)	0.00 (0.00)

of the classifier is reduced. Working with longer texts improves the classification accuracy, but short structure cannot be identified.

For both data sets the classifier achieves very good results with Evaluation and Methodology. One reason is that these structures are comparably longer than others. The other reason is that the texts of Evaluation and Methodology contain unique signal words that are very uncommon in other structures. For structures, such as Abstract, Introduction, or Conclusion, no such signaling words exists. They are often referring to other section in a paper, which make it hard to distinguish these structures. This is also the reason, why these structures benefit from longer text samples, because with longer texts the function of these structures gets clearer.

Within both data sets three types easy to learn structure exists. For these structures the performance of the classifier is acceptable even with short texts. The second group of structures require much more text information, but with enough text the SVM is able to identify these structures. These structures are also the reason why the accuracy of the classifier improves with longer texts. The third group contains all structure that are usually shorter than the optimal text length. This group can therefore not be classified with the SVM approach.

Using a domain dependent tagger, such as the ABNER tagger for biomedical texts, can improve the quality of the structure recognition. In the PubMed data set a Conclusion is much better recognized with ABNER tags than with the POS tags. Overall the performance of the classifier with POS or ABNER tags does not differ and other properties of functional structures have a bigger impact on the performance, for instance the length and the domain.

The last important factor of functional structure analysis is the domain of the documents. The accuracy on the PubMed data set is 10 % higher than on the computer science articles. The main reason for this difference is clearer structure of the article in the PubMed data set. The variation of different topics and article types is higher within the computer science data set, which results in a less accurate classification.

3.6 Summary

The functional parts of documents that can be recognized are depending on the concrete document type. The documents must consist of a similar functional structure in order to be able to extract meaningful features and create automatic analysis models. Papers fulfill this requirements, their structure is often similar, which improves the orientation within a paper and enables functional structure analysis. In general, functional structure requires analysis of text content. The usage of keywords or formatting properties may allow the recognition of some functional types, for instance abstracts with an italic font, but does not address the recognition of other functional structures in general.

The quality of the recognition results are depending on the functional types and their relationships. A functional type that has a unique property, such as the lack of sentence structures in references, is easy recognizable. Complicated to recognize are functional structures that are referring to different parts. For instance, abstracts or conclusions are summarizing other sections in a paper and contain therefore content that is similar to the summarized sections. In this case the distinction between the abstract and the summarized section is difficult. A more severe problem exists with functional structures that are usually very short, for instance acknowledgments or biographies. Whereas larger texts improves the recognition results of other functional parts, these parts are too short to be recognized when using a too large minimal text length.

Chapter 4

Visualization for Document Structure Analysis

This chapter describes methods for visualizing document structures and how these techniques can be used to analyze structure analysis results or improve the feature sets. After the motivation and related work, two different approaches for visualizing document structures are presented. Finally, the chapter is summarized. The variable text scaling technique described in this chapter is published in [Sto+12] and the usage of thumbnails for structure visualization is published in [Sto+10].

4.1 Motivation

The visualization component of the presented structure analysis framework is used to evaluate the results of the different steps of the analysis process and to create the required training data for the machine learning algorithms.

The evaluation of the structure analysis process is not only used to assess its quality but also to figure out the problems of the algorithms. The quality of the structure analysis can simply be calculated with measures such as accuracy, precision and recall, or AUC. These measures allow a judgment of the overall quality, but they do not allow to identify problems in the process. In case the system is not able to achieve the requested quality, the problem is to identify the

actions to take in order to improve the situation. Visualization can help in this respect, as it allows an expert user to identify possible problems by inspecting the outputs of the different analysis steps.

For creating or correcting training data, the system must provide visualizations and interactions for the user. The problem of the iterative training approach is the identification of miss-classification. As the quality of the process improves with additional training iterations, it gets more and more complicated to identify the classification errors. Including the probabilities of the machine learning process in the visualization can help the user to identify the problematic spots and improve the quality of the corrections.

Two different techniques are used to solve the visualization task. Thumbnails are used to create an overview over multiple documents, because with their small size it is possible to show many pages on a screen at once. In addition, distortion of the page content is used to highlight interesting parts of the documents. Using distortion for highlighting has the advantage that color is still available for mapping of other properties.

4.2 Related Work

Visualization is commonly used for labeling structures in documents. Colored highlights are the usual technique to visualize the different structures. Either colored overlays [RHI03; YSS05] are drawn over the particular content or the pixels on the document image are colored [SLS09]. To change the label of a region, a user selects the region and chooses the correct label. Thumbnails are used in a navigation area to create an overview over a complete document by show multiple pages [YSS05] and allowing a user to select interesting pages. These approaches focus only on the labeling of structures in documents. They do not support a user in finding unlikely labels automatically or allow further analysis of the structure analysis process.

In general, three different techniques are used for document overview and navigation: abstraction from the document with pixel based representations, thumbnails with different highlighting techniques, and semantic zooming.

A common pixel based technique is TileBars [Hea95], which visualizes the length of documents and the distribution of search terms within these documents with a rectangular pixel-based visualization. Instead of using a different representation of a page, the search terms can also be highlighted in the scrollbar of the detail view with pixel visualization [Byr99]. This allows a user to scroll directly to the occurrence of the terms. Both techniques are abstracting from the original page layout and are focusing on the textual content. This makes them less useful for visualizing the logical structure of documents, because the logical structure is mainly visible in the layout of the documents and not in the textual content.

Thumbnails, small version of the document or page, are commonly used for overview and navigation. The space-filling thumbnail approach [CGA06] avoids scrolling in the overview of a document, by positioning the thumbnails of all pages on a grid on the screen and resizing the thumbnails to fit the window size. Thumbnails can be combined with popouts [Suh+02]. The popouts highlight search terms by rendering them in a readable size with a semi-transparently colored background above the original thumbnail. The enhanced thumbnail technique for web pages [Woo+02] modifies, in addition to popouts for keywords, the original HTML document to enlarge the size of headlines. The main problem of popouts is overplotting in areas with many highlighted terms. In this case the different popouts are overplotting each other and bottom most popouts are not clearly visible.

Thumbnails are often combined with semantic zooming. For instance, fisheye lenses are combined with page thumbnails [Woo+02]. The thumbnails are arranged on a grid and a user can magnify a single page using a fisheye lens for reading and checking context. In order to highlight the interesting lines (e.g. headlines) in thumbnails, the uninteresting lines can be even shrunk further [HF01; BO08]. Additionally, interesting terms can be highlighted with a colored background, which avoids the overplotting problem of popouts. Showing the whole text on a thumbnail is often not required. Unimportant words can be removed and the remaining text can be cropped in order to fit it into thumbnails on a readable size [LB05]. The presented semantic zooming techniques

use automatic algorithms to decide what information to show or hide. A different approach uses interaction and let the user define, which parts of a document should be show in more detail and which should be hidden [Bau+04]. Unfortunately, is the semantic zooming technique destroying the original layout of a page. This is acceptable when reading the document, because the important information is highlighted. But for structure analysis the layout information is important and the general layout of the document has to be preserved to allow efficient analysis.

Algorithms developed for calculating graph layouts are the most similar to the suggested distortion technique. The size of nodes in a graph visualization are scaled according to their degree of interest and their positions are recalculated [SM95]. Depending on the user needs, the interesting nodes increase in size while the other nodes are shrunk. The final update of the node position preserves the orthogonal ordering of the nodes in the visualization. This idea can be used for interactive zooming in hierarchical networks as well [Bar+95]. After a drill down or a roll up, the weights of the visible nodes are adjusted and their size change accordingly. The scaling and layout techniques are focusing on graphs and are using a linear scaling of degree of interest to size. The properties of textual documents limit the scaling possibilities for algorithms. Especially the fact that longer words require more space in the horizontal direction and common page formats have more space in the vertical direction is a limiting factor. An example of applying the zooming technique to text documents can be seen in Figure 5.4b.

Using thumbnails for navigation and a distortion on the thumbnails as well as on the detail view allows visualization of the document structures. The distortion has the advantage that interesting parts can be highlighted with larger size and color is still available for mapping other values. In addition to the existing structure visualization approaches, the technique can additionally be used to visualize features and thereby allows a better insight into the structure analysis process. The used distortion is an adapted version of the zooming techniques presented in [Bar+95] but is carefully designed to work best with textual data and preserve the page layout. The distortion avoids the overplotting prob-

lem of popouts and preserves the global layout, which is not guaranteed with the semantic zooming techniques.

4.3 Variable Text Scaling

The zooming algorithm described in [Bar+95] resizes the nodes of a hierarchical graph according to a degree of interest (DOI) function. According to the DOI of the visible nodes, the graph visualization is resized and the final size and position of the nodes is calculated. The algorithm is designed for graph visualization and does not respect the special properties of textual documents. In order to adapt the algorithm for textual document the linear scaling of the algorithm is replaced with a distortion that respects the DOI of words but does not scale linear. The distortion algorithm is designed to fulfill the following requirements:

D1 Overplotting and occlusion should be avoided and the overall page layout after the distortion should match the undistorted view, to aid navigation.

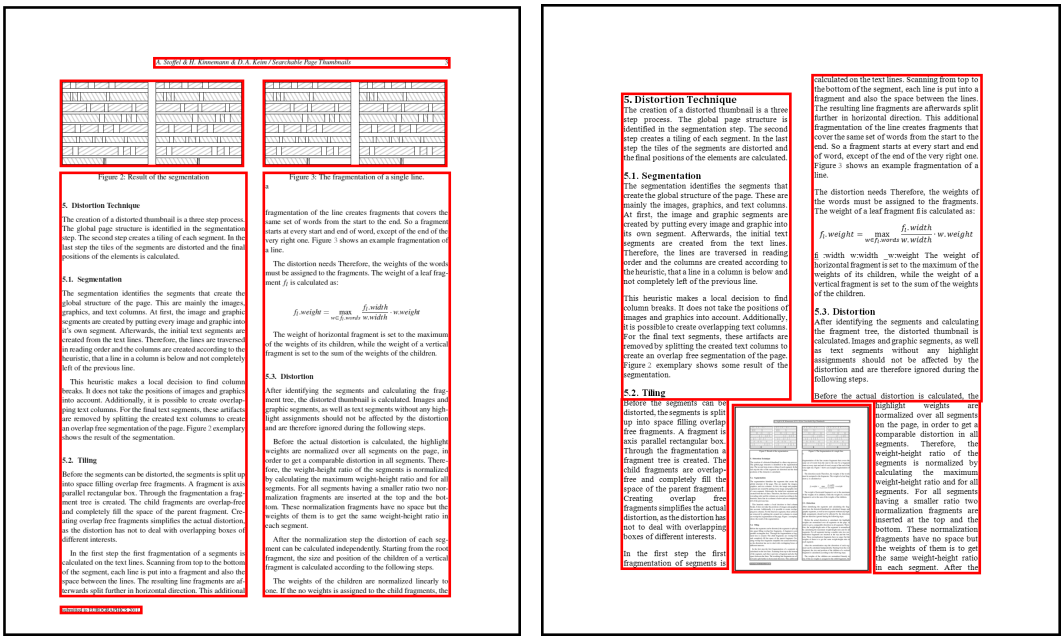
D2 The context of interesting text should be shown to give the user an additional hint of whether a passage in the document is of interest or not. A context in this case is typically a window of n words before and after an interesting one.

D3 The most interesting text must be readable even on thumbnails. In extreme conditions least interesting text may not be shown at all.

D4 The user should be able to control the degree of distortion and the size of the interesting text.

4.3.1 Creating Distorted Page Thumbnails

The distortion of a page is created in a three step process. First, the degree of interest (DOI) for each word is calculated. The second step identifies the text columns forming the global page layout. In the last step the text content of the columns are distorted and the final position and size of the words are calculated.



(a) (b)

Figure 4.1: Results of the segmentation of different pages. The created segments are marked with a red border.

4.3.2 Overview of the Algorithm

The input of the distortion algorithm is a document and an interest function that assigns a DOI to each word in the document. A simple interest function can be created from a string search by setting the DOI of a word to 1 if the word matches the search string, otherwise to 0. After applying the interest function to the document, the DOI along the text flow is smoothed by applying a Gaussian kernel. This smoothing distributes the degree of interest to the context of the interesting words and ensures the visibility of context in the final thumbnail. The user is able to control the size of interesting context by changing the size of the Gaussian kernel. The smoothing step is necessary to fulfill the requirements from D2, but might be omitted depending on the characteristics of the interest function.

In the second step, the segments forming the global layout of the page are identified. These segments are the text columns, the images and the graphics on a page. For the distortion process, only the text columns are of interest, be-

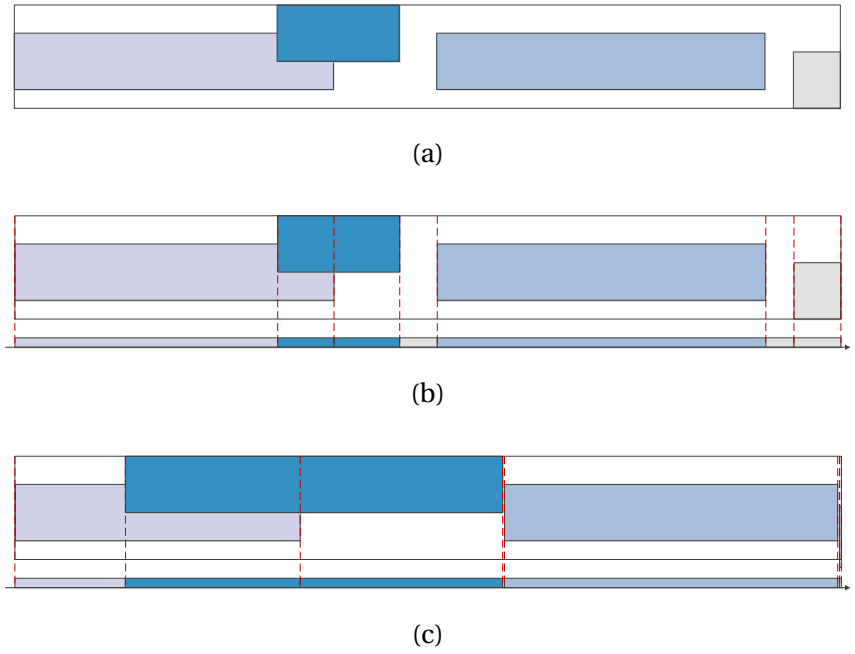


Figure 4.2: The horizontal intervals of a simple line. a) shows the bounding boxes of the words, b) the created intervals, and c) the final distortion. The blue boxes are the interesting ones and the saturation corresponds to the degree of interest.

cause images and graphics are not affected by the distortion. Figure 4.1 shows the result of the segmentation process for two different pages. The global page layout is preserved (D1) by distorting only the content of text columns and not their bounding boxes. The last step is to calculate the size of words in a text column in the horizontal and vertical direction according to the assigned DOI.

4.3.3 Distortion of Text

The distortion algorithm is a modified version of the zooming algorithm for node networks described in [Bar+95]. The original zooming algorithm works independently in horizontal and vertical directions by projecting the node boundaries on vertical and horizontal axes. The projected bounds are then moved along the axes such that the sizes of the intervals between them match the weight assigned to the corresponding nodes. An example for the horizontal intervals of a single line created by the projection is shown in Figure 4.2.

As text data is different to node networks, applying the zooming algorithm to the text column does not give an optimal result. For instance, the zooming algorithm preserves the horizontal order of nodes, which on text data is only required within a line and not within a complete text column. In order to improve the scaling of text data, the zooming algorithm is modified in the following way:

1. The vertical distortion is calculated on complete lines within a column instead of the single words. The horizontal scaling is then applied to the words on each line separately.

2. The user controllable parameter ϑ with $\vartheta \geq 1$ is introduced that limits the maximal width and height of a word to $w' \leq \vartheta \cdot w$, respectively $h' \leq \vartheta \cdot h$. Setting ϑ to 1 disables any distortion and the normal page is generated. When setting the parameter to infinity the words are scaled linearly with the DOI. Limiting the maximal size of a word has two effects. First, it introduces a variable that controls the distortion, which fulfills requirement D4. Second, the most interesting words consume less space, which can be distributed among the remaining words, especially for the context words—the requirement D2.

3. The size of the most interesting words is increased at the expense of decreasing the size of the least interesting ones. The size of the words is calculated in order from the highest to the lowest interest. This allows us to assign the optimal size to the high interest words. If too little space is available to show all words on a line, all the space is consumed by the high interest words and the low interest words are not shown at all. Figure 4.3 shows the results of the vertical, horizontal, and the combined distortion.

For the vertical distortion a DOI for each line must be calculated, because words are not directly used for adjusting the vertical size. In this case the DOI of a line l_i is defined as the maximal DOI of its words t:

$$l_i.\text{doi} = \max_{t_j \in l_i} t_j.\text{doi}$$

The DOI of a spacing between two lines is set to the average DOI of its adjacent lines.

Next, the DOI in the whole document has to be normalized in order to get comparable line heights. The normalization is achieved by forcing the DOI to height ratio of all text columns in the document to the same value. The normalization does not change the DOI of lines, but adjusts the DOI to spacing intervals. After the normalization of the document D, all text columns $C = \{c | c \in D\}$ have the same DOI to height ratio n_v:

$$n_v = \max_{c_k \in D} \frac{c_k.\text{doi}}{c_k.\text{height}}$$

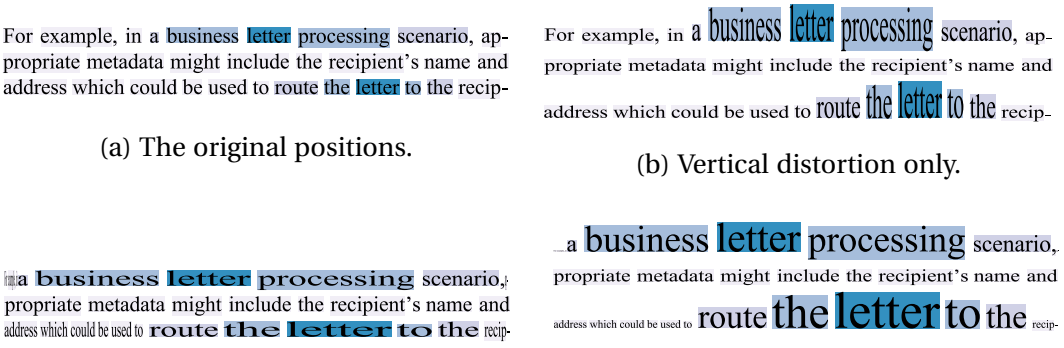
After normalization the DOI in a document, the final size and positions of lines and words is calculated. Algorithm 1 shows the algorithms for calculating the height of lines in detail. Starting from the line with the maximal DOI, the height is adjusted linearly according to the total DOI and the available space. If the new height h' of an interval would exceed the limit $\vartheta \cdot h$ with the linear scaling, the height is fixed to this limit and the DOIs of the remaining intervals are adjusted in order to preserve the normalized DOI to height ratio n_v of the text column. Figure 4.3b shows an example of the distortion of lines distortion.

The horizontal distortion of words within a line is analogous to the vertical distortion of lines within a text column. Instead of projecting to the vertical axis, the word bounds are projected on the horizontal axis. For normalization, the DOIs of the space intervals are changed in order to get the same DOI to width ratio for all lines in the document. The final size of the intervals is then calculated with an adapted version of Algorithm 1. An example of the horizontal distortion is shown in Figure 4.3c.

Finally, the distorted page thumbnails are created from a combination of the vertical and horizontal distortions by scaling each word separately in a way that the word fits into the distorted vertical and horizontal borders and is aligned to the baseline of the text line. Figure 4.3d shows the result after combining the vertical and horizontal distortions.

Algorithm 1 Distortion of Lines in a Column.

$\vartheta \leftarrow$ the maximal up scale
$n_v \leftarrow$ the vertical normalization factor
$L \leftarrow$ lines of the text column
$H \leftarrow$ total column height
$I \leftarrow \sum_{l_i \in L} l_i.\text{doi}$
while $L \neq \varnothing$ **do**
  $l \leftarrow \arg\max_{l_i \in L} l_i.\text{doi}$
  $L \leftarrow L / \{l\}$
  $h \leftarrow n_v \cdot l.\text{doi}$
  if $h \leq \vartheta \cdot l.\text{height}$ **then**
    $l.\text{height}' \leftarrow h$
  else
    $l.\text{height}' \leftarrow \vartheta \cdot l.\text{height}$
    distribute the degree of interest $n_v \cdot (h - l.\text{height}')$ to the elements in L proportional to their height
  end if
  $H \leftarrow H - l.\text{height}'$
  $I \leftarrow I - l.\text{doi}$
end while



(a) The original positions.

(b) Vertical distortion only.

(c) Horizontal distortion only.

(d) Combined horizontal and vertical distortion.

Figure 4.3: Example of the distortion of three lines. The saturation of the blue color shows the degree of interest of the different boxes.

4.3.4 Evaluation

The distortion technique fulfills the requirements D1 and D4. D1 requests a overplotting and occlusion free algorithm that preserves the global page layout. This requirement is fulfilled, because the algorithm does not introduce any overplottings or occlusions and preserves the positions of text columns and images. D4 requires the distortion to be controllable by the user, which is the case through the user controllable parameter ϑ in the distortion. Whether the distortion fulfills requirements D2 and D3 is still open questions. Another question to investigate is difference between the presented distortion algorithm and the original zooming algorithm presented in [Bar+95].

The requirements D2 and D3 request the interesting terms and their context to be readable. The readability of a words is directly related its size. If the algorithm is able to scale up the size of a word by factor α, the page can be shrunk by the same factor and the size of the word is the same size as before. In order to evaluate the requirements D2 and D3 the size of words before and after the distortion is compared. For the comparison the area factor $A = \frac{w' \cdot h'}{w \cdot h}$ is used, which expresses how much a word was scaled trough the distortion. As a final measure the median of the area factor of interesting words and their context is used. The median is chosen rather than the arithmetic mean, because it guarantees that half of the words have an area factor greater or equal to this value.

For a fair comparison of the distortion algorithm with the original zooming algorithm, the zooming algorithm is not applied to a whole page, but to each text column separately. Additionally, the size of a text column is kept fixed and not changed by the algorithm. These two modifications ensure that the zooming algorithm does not change the layout or dimension of the page, as the distortion algorithm does.

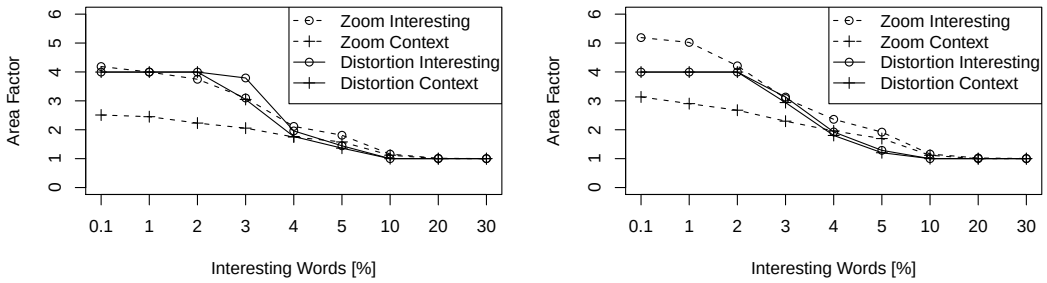
The amount of distortion depends mainly on the available space and the distribution of interesting terms. The available space is determined by the page format and layout of a page. For instance, the possible amount of horizontal distortion is larger with a single column layout than on a page with a two columns. In order to test the influence of the page layout on the distortion two

document collections are used: the MICAI 2011 proceedings with a single column and the EuroVis 2011 proceedings with a two column layout. The EuroVis 2011 proceedings contain 538 pages with 48047 lines (average 10.57 words per line). The MICAI 2011 proceedings contain 591 pages with 23300 lines (average 13.12 words per line). In order to emulate different distribution of interesting terms, randomly generated interest functions with different distribution of interesting terms are used for the evaluation. After filtering stop words, the DOI is distributed over the remaining words with different densities and varying the kernel size.

Results and Discussion

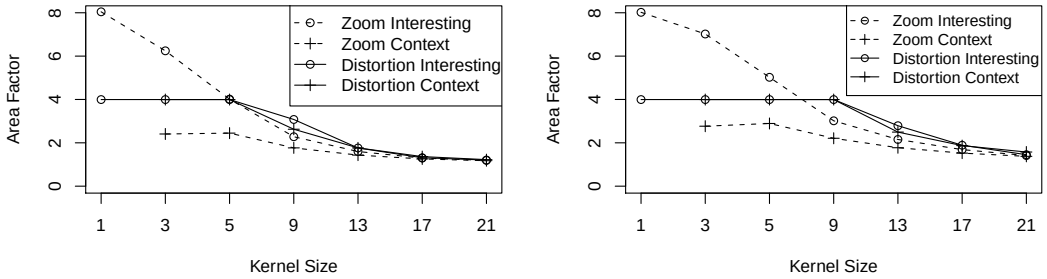
Figure 4.4 shows the area factor on different term distributions at a kernel size of 5. The parameter ϑ of the distortion is set to $\vartheta = 2$. For the zooming technique of Bartram et al. the weight of an interesting word is set to 3.75 times the weight of a normal word. These parameters are adjusted in a way that the median area factor is closest to 4.0 when 0.1 % of terms are of interest. The results in Figure 4.5 show the area factor over different kernel sizes with 2 % of interesting terms. The parameters of the distortion and of the zooming algorithm are the same as before.

Figure 4.4 shows the difference between the zooming and the distortion algorithm according to the density of interesting words. In the case of Figure 4.4a, the area factors of interesting words behave similar for both techniques: with a few interesting words the area factor is high and degrades as the number of interesting words increases. The area factor becomes 1 when either the height or the width of the words cannot be increased, because both algorithms preserve the aspect ratio. The difference between the two algorithms is the size of the context words when the number of interesting words is low. In this case the distortion is able to assign more space to the context words than the zooming algorithm. In the case of 0.1 % interesting words, the size of the interesting words has reached the maximal allowed value and thus the context words can use the additional space. This effect becomes clearer in the results of the single column layout in Figure 4.4b. The distortion is able to increase the size of the context



(a) EuroVis 2011 (b) MICAI 2011

Figure 4.4: Area factor over different distribution of interesting terms.



(a) EuroVis 2011 (b) MICAI 2011

Figure 4.5: Area factor for different kernel sizes.

words with the same factor as the interesting words for up to 2 % of interesting words. The zooming algorithm on the other hand does not have an upper bound for the size of words. This leads to the situation that the most interesting words consume the majority of available space and less space is available for the context words.

The results in Figure 4.5 compare the two techniques for different kernel sizes. For large kernel sizes, the two algorithms produce similar results on both collections. But for small kernel sizes, the results of the algorithms are different. As before, the distortion assigns more space to the context words, because of the upper bound of word size.

As expected, the area factor for the single column layout of the MICAI proceedings is slightly larger than for the two column layout of the EuroVis proceedings, because the width of a line limits the amount of horizontal distortion. As a single column layout allows wider lines, more distortion can be applied to the text of the MICAI proceedings.

The comparison of our distortion algorithm with the original zooming algorithm shows that for high densities of interesting terms both algorithms behave similar. In this case too little words are of low interest and neither the zooming nor the distortion algorithm is able to increase the size of interesting words. In case, the interesting terms are sparsely distributed, both algorithms increase the size of the words differently. The zooming algorithm linearly scales the words according to the assigned interest, which results in larger interesting words and smaller context words. In contrast, the distortion algorithm limits the maximal size of the interesting words and is able to increase the size of the context words with the same factor. As a result, the context words are larger and therefore better readable with the distortion algorithm then with the zooming algorithm.

4.4 Visualization for Document Structure Analysis

For annotating and correcting the structure of documents a visual interface is used. The interface contains two main components: the thumbnail view, allowing a user to navigate within the document, and a detail view, showing the content of a single page in detail (see Figure 4.6).

4.4.1 Visualization of Structure Analysis Results

The visualization of the structure analysis results uses color to annotate the structure types. The thumbnail view is used to spot problems and errors in the annotation and to navigate within the documents. It shows only the structure annotation. The original content of the page, such as text or images, is discarded in order to not distract a user. The detail view shows the original page with its complete content and adds colored overlays and a textual abbreviation

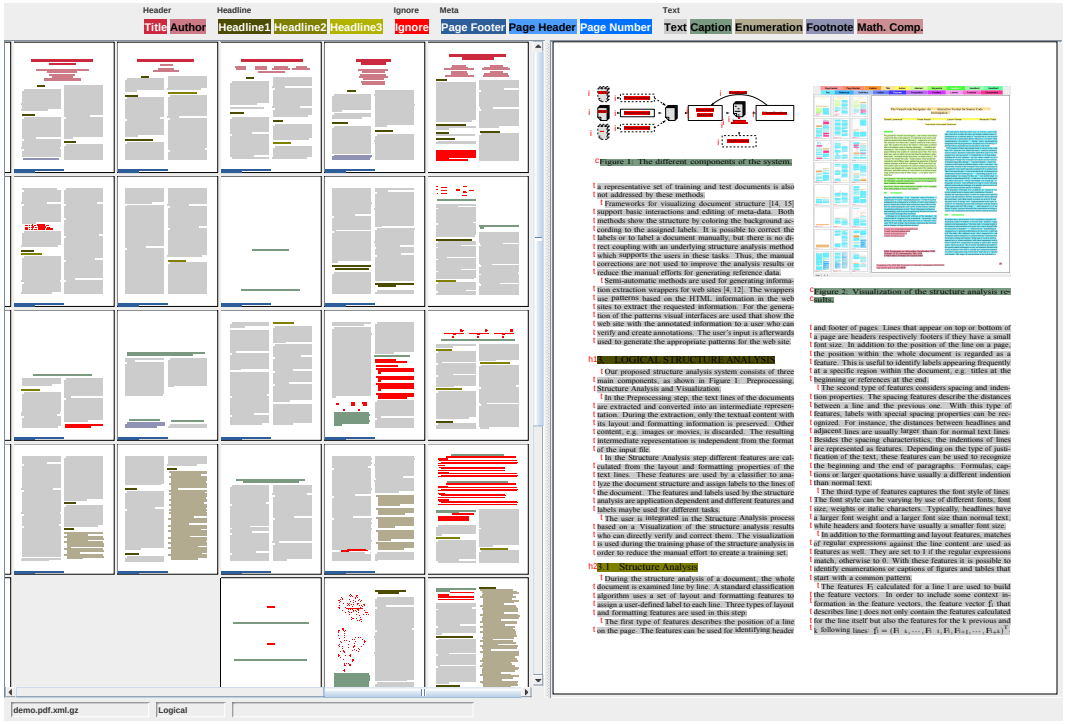


Figure 4.6: Visual interface for annotating and correction document structures.

for the structure types. The detail view is used to correct the analysis results or to analyze the problems of the model. This double coding improves the visibility of the different structures.

Figure 4.7 shows structure analysis results for two models of different quality. Errors of the low quality model in Figure 4.7a are easy recognizable in the thumbnails. For instance, the red title and author in the third document or the dark blue annotation on the fourth page of the last document are clearly visible. A low quality model makes the same mistake over and over again, because of the systematic flaw within the model. The recognition errors of low quality models can therefore be founfoundd by looking for irregularities and inhomogeneities in the thumbnails.

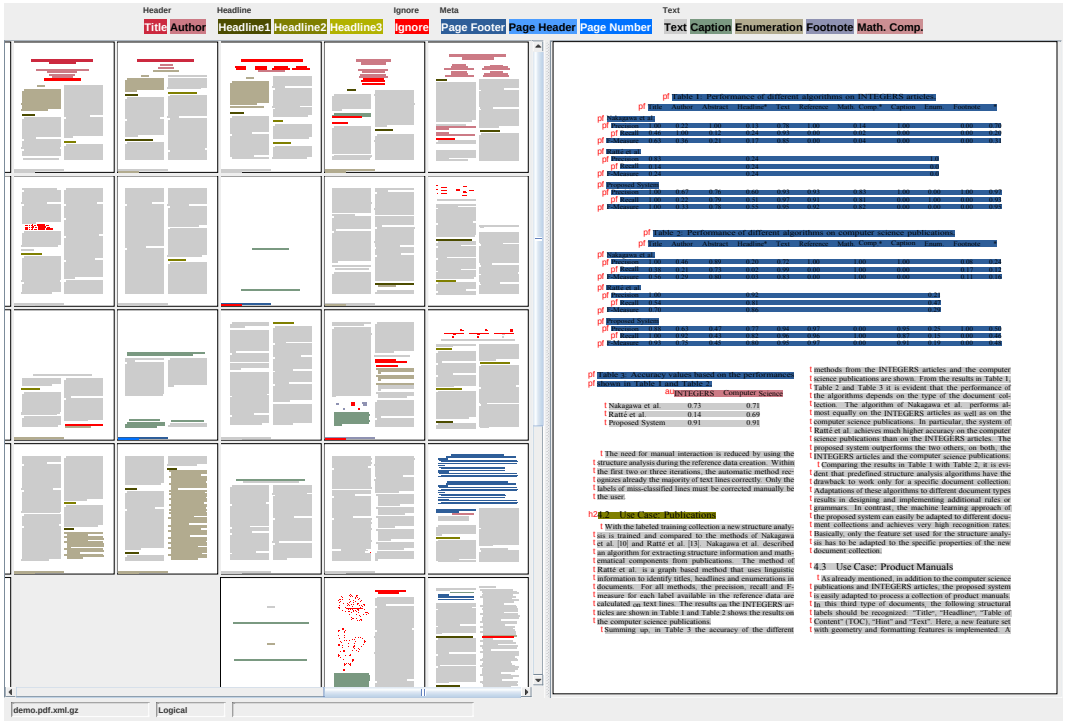
For medium and high quality models analysis errors are much harder to recognize with the original thumbnails. For example, in Figure 4.7b the detail view shows the first page of the last document. By looking at the detail view it is clear, that the model has problem to recognize headlines correctly or to distinguish the title from the author. Both problems are barely visible in the thumbnail

view. The model has achieved a quality that allows it to classify the majority of lines correctly. The systematic problems of the low quality model, allowing the thumbnails to reveal errors, are mainly solved. The main problems of medium and high quality models is the separability of classes in border cases. This leads to wrong analysis results for single lines, which are less visible in the thumbnails than errors in larger blocks.

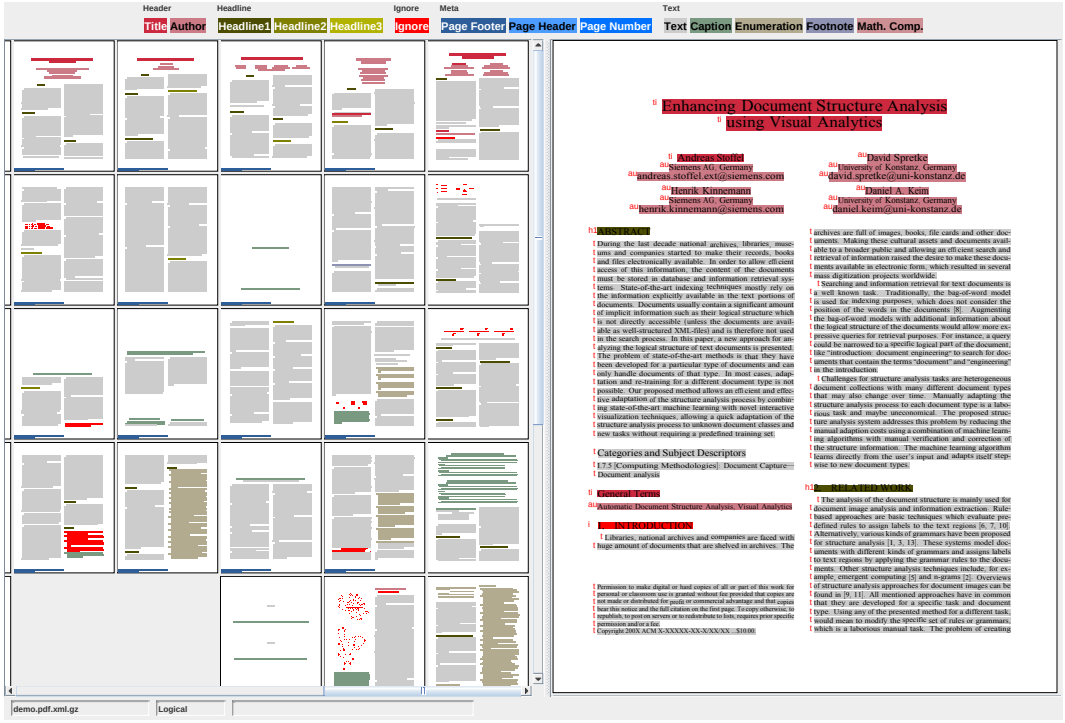
A solution to the problem of the thumbnails with medium and high quality models is the usage of the model's confidence value. In Figure 4.8 two different configurations of the visualization of the medium quality model are shown, which take the model confidence into account. In Figure 4.8a only the thumbnail view take the model confidence into account. The thumbnails are generated with the variable text scaling technique and the uncertainty of the model as interest function. In addition, the color is faded into the white for text lines with high confidence. In the resulting thumbnail, the lines with a high confidence of the model disappear and the lines with high uncertainty gain more space and better visible color. For example, the detail view shows the first page of the right most document shown in the thumbnail view. The uncertainty of the model points a user to verify the title and author information as well as the check the headlines. Additionally, the scaling techniques can be applied to the detail view, as shown in Figure 4.8b. This improves the visual perception of the text lines with high uncertainty.

4.4.2 Visualization of Features

Improving or adapting the structure analysis process for a new document collection or to a new analysis tasks, requires the analysis of the existing structure analysis features. Instead of using the structure type for coloring the document elements, the color can be derived from a feature value. Figure 4.9 shows example visualizations of the rational Font Size Change and the nominal Paper Keywords features. This visualization helps to verify the implementation of the feature and to understand the problems of the structure analysis models.

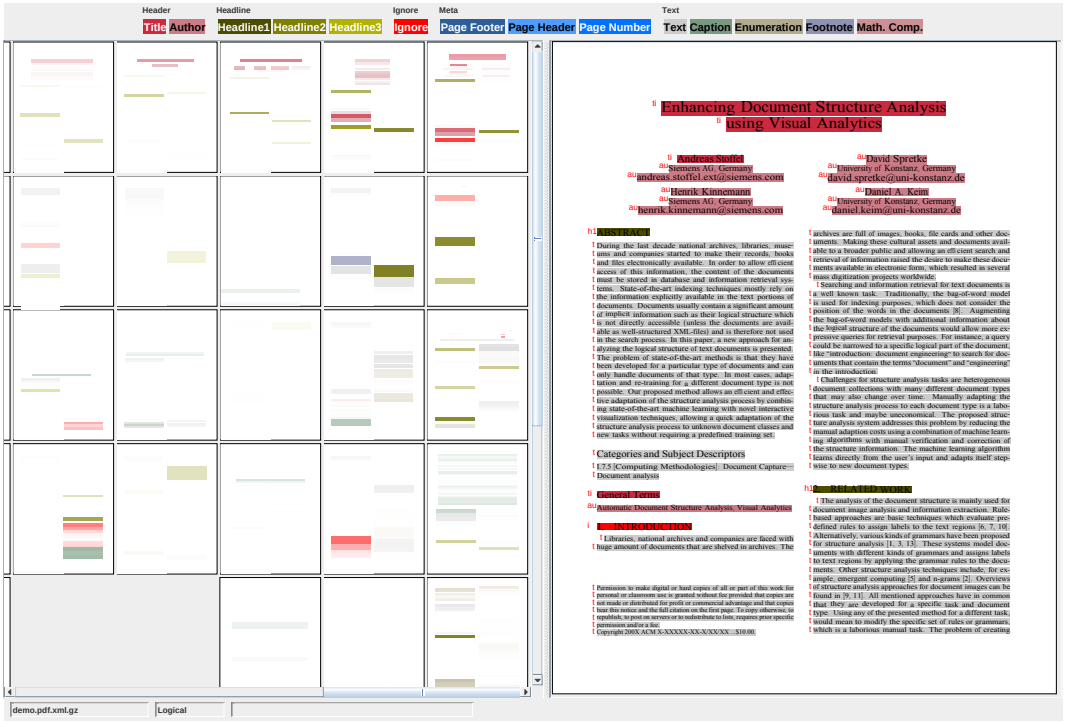


(a) Low quality model trained with four example documents.

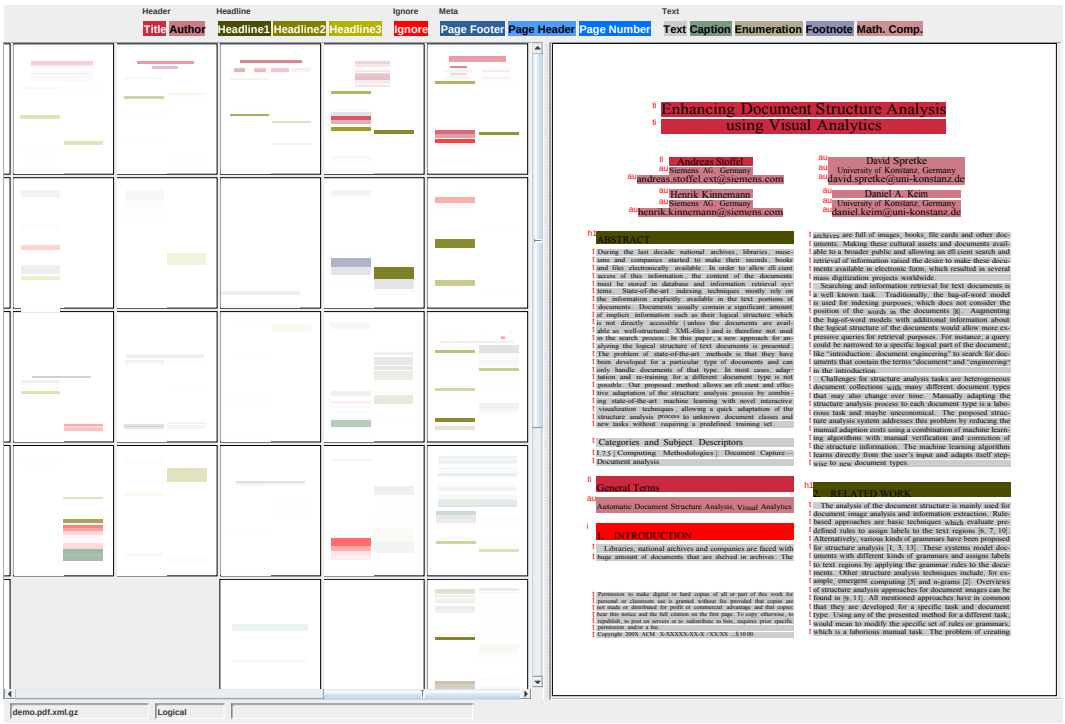


(b) Medium quality model trained with 16 example documents.

Figure 4.7: Logical structure analysis results with two models of different quality.



(a) Highlighting of uncertain results only in the thumbnail view.



(b) Highlighting of uncertain results in the thumbnail and detail views.

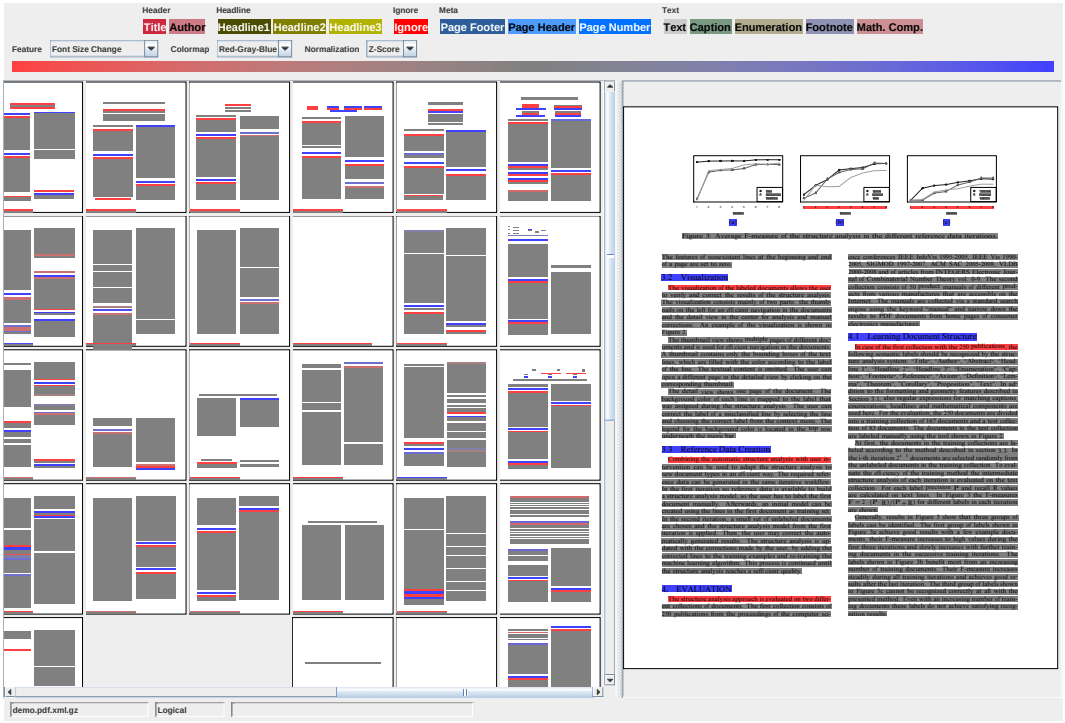
Figure 4.8: Logical structure visualization based on model confidence.

The Font Size Change feature shown in Figure 4.9a is suited for recognizing headlines in general. A headline usually starts with a large increase in font size follows by a large decrease. This feature allows to detect headlines but not to distinguish the different levels of headlines from each other. Accidentally, this pattern might be observed in vector graphics as well. This feature leads to high recall for headlines, but to get high precision additional features must be considered. The case of the Paper Keywords feature, shown in Figure 4.9b, is similar. Detecting keywords, such as “Figure” or “Table”, at the beginning of lines, results in a high recall for captions, but also in false positives in the running text. A reliable recognition of captions with a high precision needs additional features, for instance measuring the line width and justification.

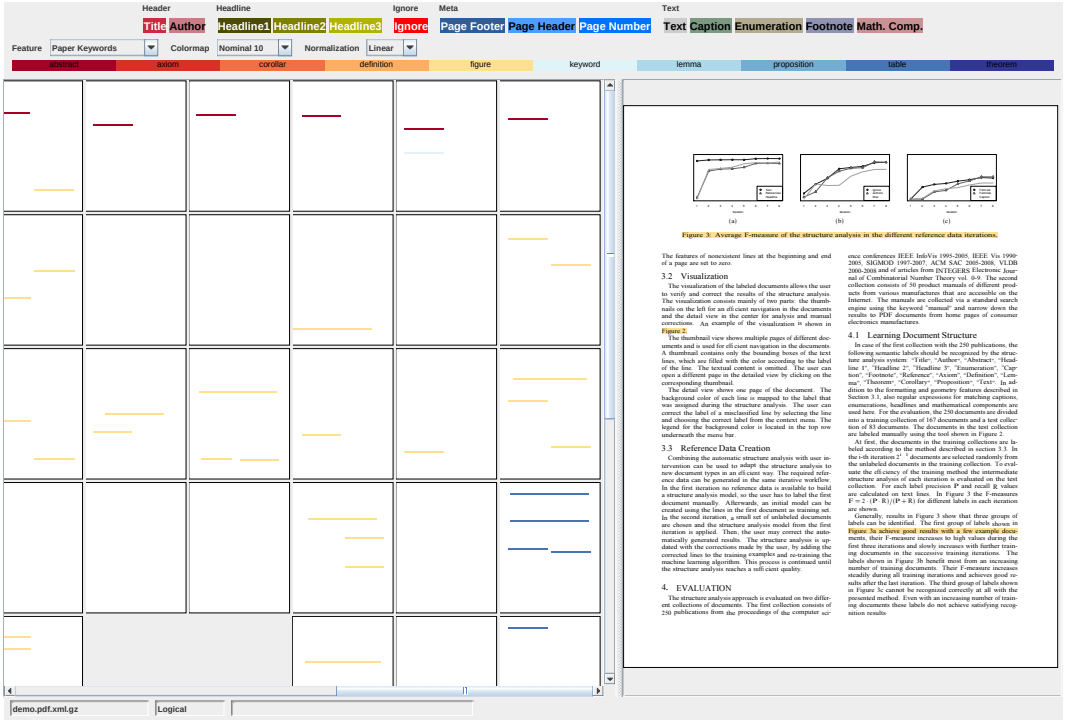
4.5 Summary

Document structures and structure analysis features can be visualized through coloring the background of a document. Thumbnails help in navigation between documents and pages. They are especially helpful in finding systematic errors of low quality models. A single miss-classified line is barely visible with the default thumbnails. To improve its visibility, the confidence of the classification is taken into account. Mainly two techniques are used, either the color of the confident elements is faded out or the size of the uncertain ones is increased with variable text scaling. A combination of both techniques is also possible. Thumbnails created with these techniques allow a user to identify pages with a high likelihood of classification errors and allow a user to focus on these ones for improving the analysis model. In addition, features for logical structure analysis can be visualized with the same technique. This visualization allows a verification of a feature implementation as well as the analysis of a feature itself. For instance, it shows the benefits of the features but the problems are visible as well.

The variable text scaling technique is designed for thumbnails but can be used in the detail view as well. It allows highlighting of single lines or words by increasing their size. The scaling has the advantage to avoid overplotting



(a) The rational Font Size Change feature normalized with the Z-score normalization.



(b) The nominal Paper Keywords feature.

Figure 4.9: Visualization of features for logical structure analysis.

and occlusions, as they would occur with popout or fisheye techniques. This is achieved with an interest function controlled resizing of the page content. In contrast to the zooming algorithm of Bartram at al. is the variable text scaling specially designed for textual content with its constraints on aspect ratio and size.

Chapter 5

Applications of Structure Analysis and Document Visualization

In this chapter applications of the logical structure analysis technique and the variable text scaling technique are presented. The chapter starts with a discussion how the logical structure analysis is used in the Document Cards [Str+09] and the VisRa [Oel+12] applications. Then the application of the variable text scaling technique for keyword search and document overview is shown. The chapter ends with a summarizations.

5.1 Logical Structure Analysis for Text Processing

Electronic documents contain much textual information that is not part of the running text. For instance, captions, headlines, or page headers are interrupting the text flow of the running text. Especially for automatic natural language processing (NLP) the interleaving of running text with other textual components is a problem. The majority of NLP algorithms, such as POS taggers, are expecting to get valid sentences as input. Insertions of text not belonging to the sentence are misleading the algorithms. For example, Figure 5.1 shows the POS tags of two different sentences calculated with the Stanford Parser [KM12]. The only difference between the sentences is the insertion of the page number in the second sentence, resulting from a page break between the tokens “likes” and

Tokens	My	dog	also	likes		eating	saussage	.
POS-Tags	PRP$	NN	RB	VBZ		VBG	NN	.
Tokens	My	dog	also	likes	2	eating	saussage	.
POS-Tags	PRP$	NN	RB	VBZ	CD	JJ	NN	.

Figure 5.1: Result of POS tagging two sentences with the Stanford Parser [KM12]. In the second sentence is created from the first one by inserting a page number.

“eating”. With the effect that the token “eating” changed its POS tag from a verb in gerund form (VBG) to an adjective (JJ).

Document Cards [Str+09] are an automatic summary technique for documents based on text and images. An example is shown in Figure 5.2. The extraction of keywords is based on the term distribution of nouns over the section of a document. The detection of a document section is based on the automatically recognized logical structure information. A section starts with a top level headline and ends at the next top level headline or at the end of the document. Finally, the noun distribution of a section is calculated by applying a POS tagger to the running text of that section and counting the recognized nouns. To extract this information, the logical structure model created in Chapter 2 analyzes papers in too many details. For instance, page headers or footers are recognized by the model but do not play a role for calculating the noun distortion over sections, unless the page headers or footers are not recognized as running text. Hence, the number of different logical structure types recognized by the model can be reduced, which improves the quality of the model.

Another application example of the structure analysis is the readability analysis of papers. Figure 5.3 shows three different views of the VisRA [Oel+12] application for readability analysis. The readability analysis of VisRA is based on different language features calculated for each sentence. A similar logical structure analysis is used as for the Document Cards. Mainly the sections and the running text of a paper are identified. The readability values are calculated on the running text and the section information is used for visualizing the results. The application offers several visualizations of the readability, among others,

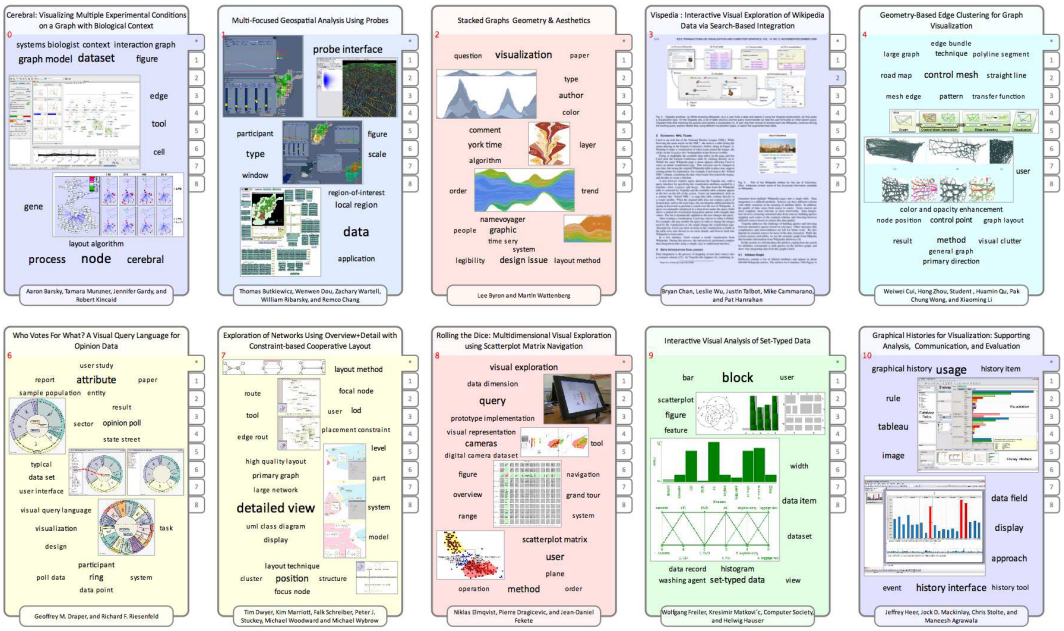


Figure 5.2: Examples of the Document Cards [Str+09] visualization. The keyword extraction is based on logical structure information.

the thumbnail technique described in Chapter 4. The thumbnails show the readability scores in the original layout of the paper. Authors familiar with the layout can judge the readability scores and identify critical parts of the paper. In addition, structure information is used for aggregating the readability results on a section level, which allows the creation of a details-on-demand interface.

5.2 Document Content Visualization with Distorted Thumbnails

The variable text scaling technique presented in Section 4.3 is used to improve the visible of uncertain analysis results. The technique itself is a general technique for highlighting interesting texts and can be used for different tasks than visualization of structure analysis results. For instance, interesting text could be highlighted in document viewers.

The user interface of a typical document viewer application, such as Adobe Reader, consists of a detail view and one or more views for navigation within

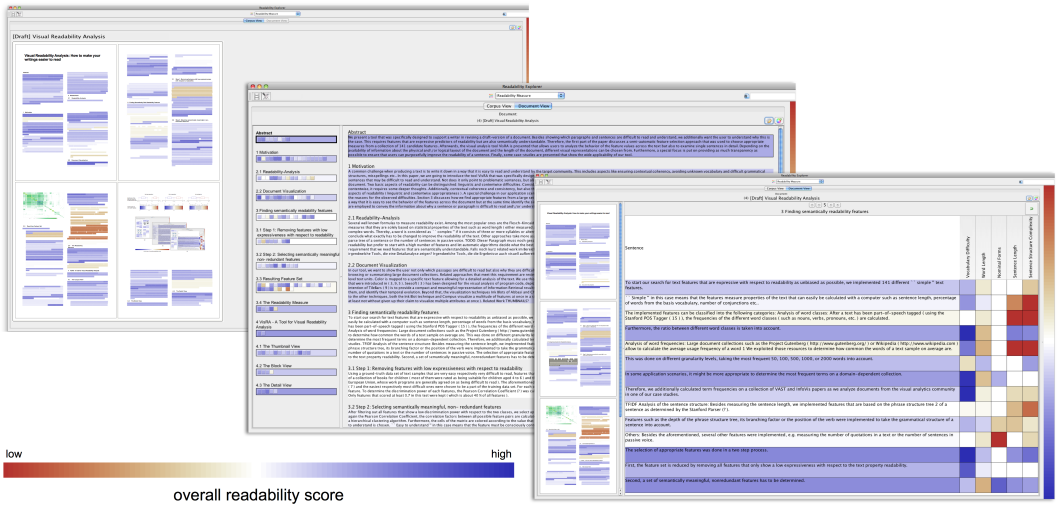
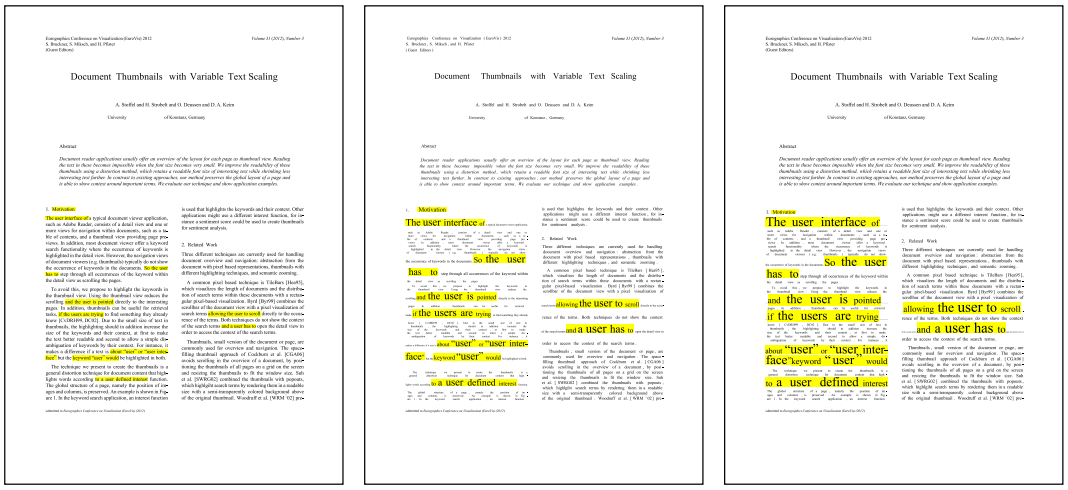


Figure 5.3: Examples of the VisRA user interface [Oel+12] for readability analysis. The readability analysis and the visualizations are based on logical structure information.

documents, such as a table of contents, and a thumbnail view providing page previews. In addition, most document viewer offer keyword search functionality where the occurrence of keywords is highlighted in the detail view. However, the navigation views of document viewers (e.g. thumbnails) typically do not show the occurrence of keywords in the documents. So the user has to step through all occurrences of the keyword within the detail view by scrolling the pages.

To avoid this, keywords could be highlighted in the thumbnail view. This reduces the scrolling and a user is pointed directly to the interesting pages. In addition, thumbnails can be useful for retrieval tasks, if users are trying to find something they already know [Cze+99; DC02]. Due to the small size of text in thumbnails, the highlighting should increase the size of the keywords and their context, at first to make the text better readable and second to allow a simple disambiguation of keywords by their context. For instance, it makes a difference if a text is about “user” or “user interface” but the keyword “user” would be highlighted in both.



(a) (b) (c)

Figure 5.4: Example of a distorted thumbnails. a) The original thumbnail with highlights. b) The thumbnail generated with the zooming technique described in [Bar+95] c) The thumbnail generated with the variable text scaling technique.

5.2.1 Keyword Search in Documents

Searching for keywords in documents is a common task for uses of document reader applications. With the distorted document thumbnails this task is supported by highlighting the search terms and their context in the thumbnails. The required interest function simply maps the search terms to 1 and all the other terms to 0. The context of the search terms is also interesting, so this interest function is smoothed with a Gaussian kernel that also highlights the context words of each occurrence of the search terms in the document.

An example of a keyword search result is shown in Figure 5.4. In this case the keyword “user” is searched on the first page of [Sto+12]. The thumbnails show the result for different techniques. In Figure 5.4a the thumbnail is created by scaling the page to the desired size and highlighting the keywords with a yellow background. Through this highlighting the user can identify the positions on the page where the keywords occur. But the small size of the thumbnails makes the text barely readable.

Using the zooming algorithm of [Bar+95], the readability improves (see Figure 5.4b) with an increase in the size of the keywords and their context and a

corresponding decreasing the size of the other text. Consequently, it is easier to spot the keywords and even the context is readable due to the effect of smoothing the DOI with the Gaussian kernel. The keyword “user” gets the maximal size and the size of the context words decrease with additional distance from the keyword.

Figure 5.4c shows thumbnail created with the variable text scaling technique. Whereas the distortion maintains a normalization of the keyword size, the size relation between keywords and context words does not reflect the difference in interest. This is the advantage of the distortion, because it is able to assign similar size to both the context and the keywords.

The advantage of showing the context is obvious when comparing the different thumbnails. Through the context shown in Figure 5.4b and Figure 5.4c the usage of the word “user” becomes clear. The first occurrence refers to a user interface and not to the user itself. Other occurrences show what the user can do with page thumbnails. For instance, the user can jump, find and navigate within a document.

5.2.2 Document Overview

Using the keyword search functionality, an automated overview of a document can be generated by highlighting the terms appearing in the title. In Figure 5.5 an overview of the first three pages a EuroVis 2011 paper is created in this way. The top row shows the original thumbnails highlighting the title words. The bottom row shows the same thumbnails, but in addition the variable text scaling technique is applied to improve the visibility of the search terms and their context. The same procedure is applied to the page thumbnails in Figure 5.6 in an MICAI 2011 paper. The thumbnails of the full papers can be found in Appendix D.1.

The thumbnails in the bottom row of Figure 5.5 give a user, in addition to the page layout, a hint about the content of the different pages. For instance, the introduction on the first page starts with talking about molecular visualizations. The related work section on the second page discusses visualization techniques

and then structural abstractions, and a reader interested in continuity would want to read page four. This is a clear benefit of the variable text scaling technique, because on a normal page thumbnail the text is not readable at all.

Figure 5.6 shows the result of the same procedure applied to a single column MICAI 2011 paper. The results are similar to the two column EuroVis paper. The original thumbnail in the top row only show the distribution of title nouns. The noun and the context are not visible in these thumbnails. The results after applying variable text scaling are shown in the bottom row. Here, the keywords and their content can be read. For instance, the second page is mainly about "opponent models" and on the third page "strategy" comes in.

Figure 5.5 and Figure 5.6 also show the limitations of the distorted thumbnails. The maximal possible size of a word is restricted by the size of the text column. The scaled text in single column layouts can get larger, because a column in a single column layer is wider than a column in a two column layout. The second parameter that influences the amount of distortion is the distribution of interesting words on a page. In the best case, interesting words are equally distributed with many uninteresting words in-between, which can be shrunk. In the worst case, every word is of interest and hence no distortion is allowed at all, which results in a normally scaled thumbnail with yellow highlights.

In these examples both problematic cases exists. Firstly, the text columns with limited space are the page headers, which only contain a single line and the text cannot increase in size vertically. Secondly, areas with many interesting words can be found, e.g. the left column on the third or seventh page. However, in this example the thumbnail size is less than a sixteenth part of the original page size and a lot of interesting text is still readable. This is achieved by relaxing the normalization of the word size. Instead of normalizing the size in the whole document, we normalized the average size of interesting words and allowed local variation when not enough space is available.

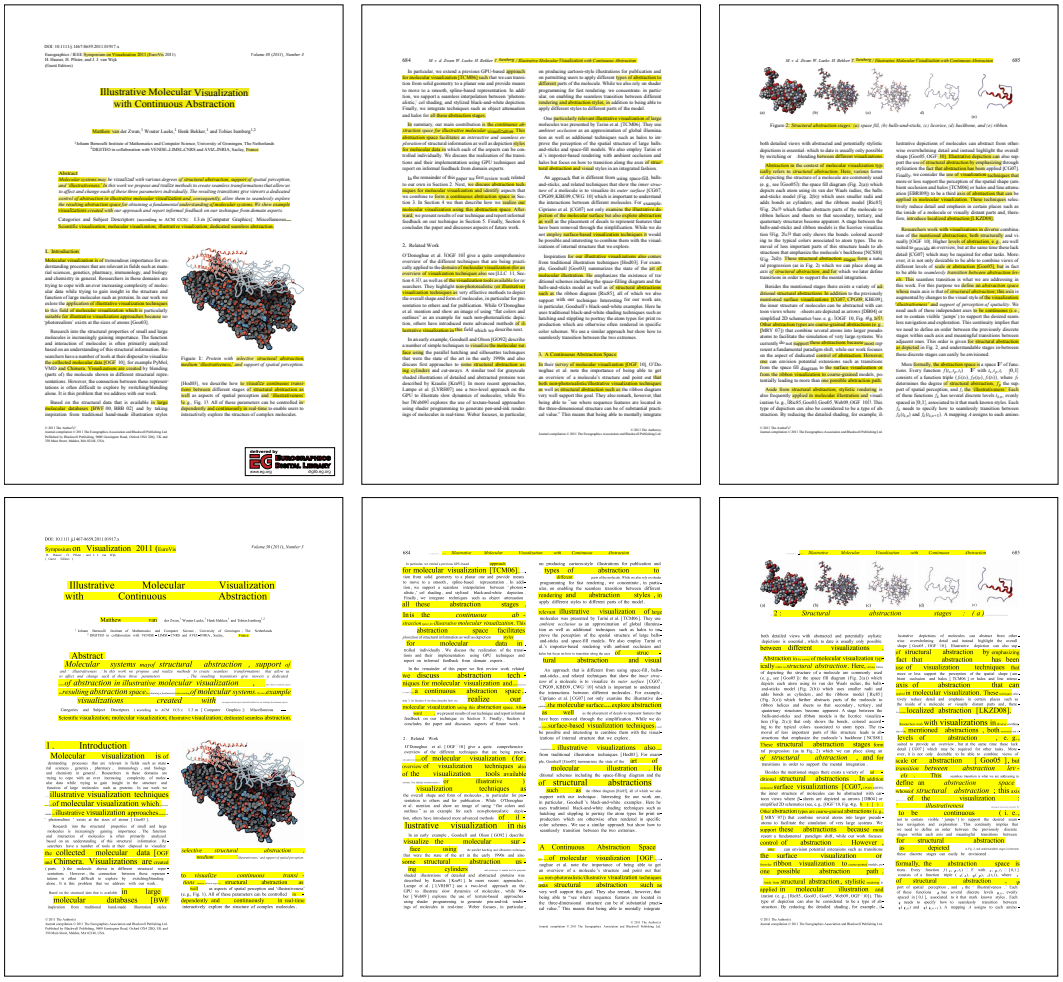


Figure 5.5: Page thumbnails created for one of the EuroVis2011 paper [Zwa+11]. All nouns appearing in the title are highlighted. The top row shows the original thumbnails and the bottom row the results after applying variable text scaling.

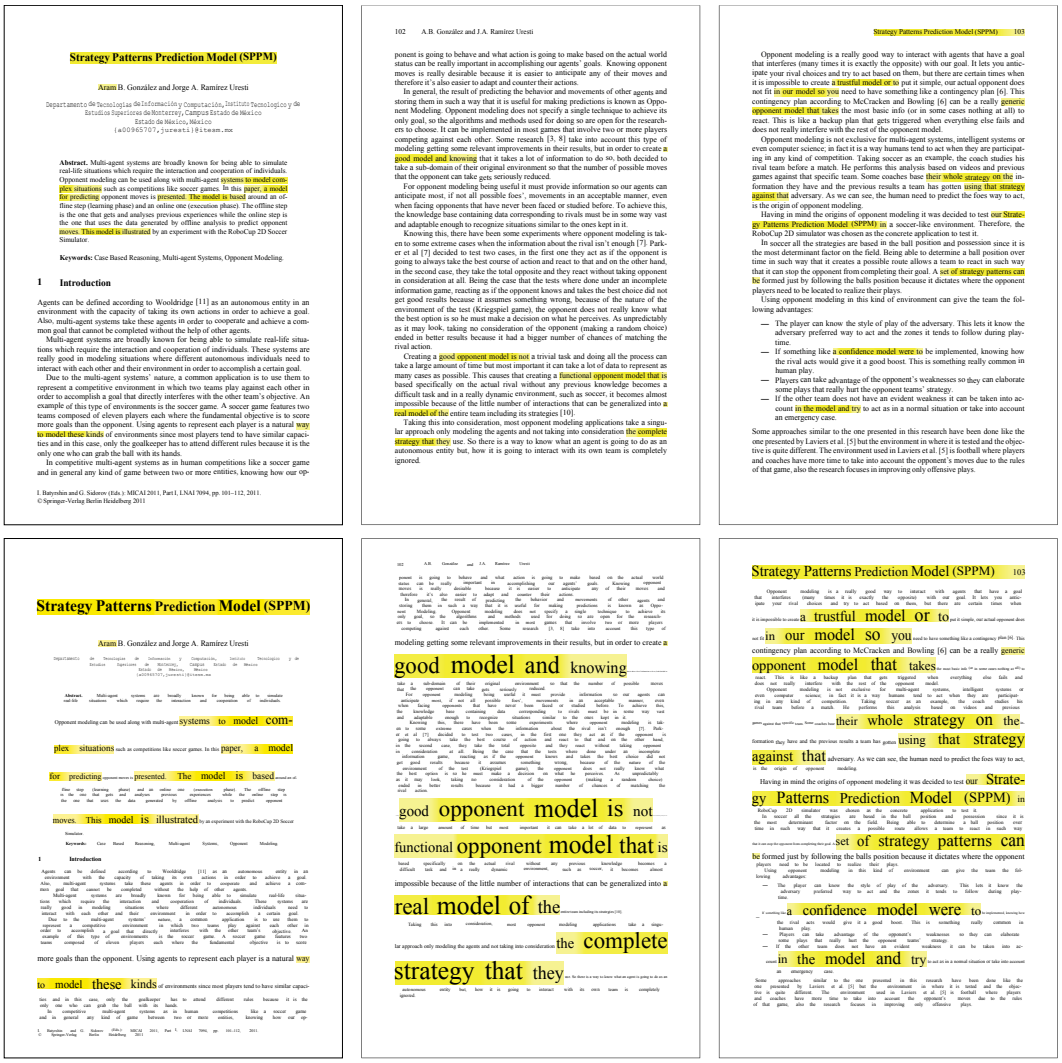


Figure 5.6: Page thumbnails created for the MICA 2011 paper [GU11]. Nouns appearing in the title are highlighted. The top row shows the original thumbnails and the bottom row the results after applying variable text scaling.

5.3 Summary

Logical structure analysis is a useful preprocessing technique for extracting section and running text from documents. Especially for NLP algorithms a correct recognition of the running text is important, because the majority of NLP algorithms assume a valid sentence as input. Information about the sections of a document can be used for calculating statistics within the document, as used by the Document Cards example, or can be used for aggregation and detail on demand interfaces, for example in VisRA. Both application examples do not use all possible logical structures. Using machine learning for logical structure analysis allows the adaption of the recognition model to the application needs, which will improve the quality of the structure recognition results.

The thumbnail visualization and the variable text scaling have additional applications apart from visualizing document structures or features. In VisRA the thumbnails are used to visualize the readability values in the original layout of the paper. The variable text scaling allows page thumbnails to visualize keyword search results by improving the readability of the keywords and their context.

Chapter 6

Conclusion and Remarks

This thesis presents different techniques for automatic analysis of logical and functional document structures. The presented approaches are addressing the structure analysis in electronic document collections containing only a few similar documents, for example a paper collection with papers of a research field. Such document collections are also common outside of research, for example the EDGAR database of the SEC, which contains obligatory reports of companies, or reports in medicine and industry.

6.1 Logical Structure Analysis

The analysis of logical structure is mainly used to improve the quality of OCR techniques or to extract interesting logical parts. The approach presented in this thesis is focusing on the latter case and assumes to get an electronic representation of the document as input. The evaluation of the approach shows a clear advantage of the presented machine learning approach compared to rule- or grammar based approaches. The main advantage is the ability to learn analysis models from example documents, which makes it possible to generate reference data efficiently and to adapt the model to a large set of example documents.

The evaluation of the logical structure analysis only worked on a single type of document, either publications or product manuals. The question how to in-

tegrate several document types into one model is of interest for different other applications, e.g. automatic processing of mail in mail rooms. Another open question is how to deal with continuous updates of the reference data. For instance, a user dealing with the processing of reports could correct the automatically recognized structures result and thereby create additional reference data. To include these data in the analysis model a completely new mode has to be created, which is a time consuming task for a large number of examples. A way to efficiently include these corrections in the model is open for further research. An additional important point is the comparison of logical structure analysis approaches. A complete test set of documents does not exist and the approaches are usually designed to work on different document types. This makes a fair comparison of approaches complicated. A solution to this problem could be the creation of a larger benchmark data set that allows the direct comparisons of the algorithms.

6.2 Functional Structure Analysis

The analysis of functional structures extends the structure analysis approach to textual content and detects functional parts of a document. Functional parts are better suited for users to work with than logical structures. Using the textual content allows the recognition of functional parts that are not distinguishable by formatting or keywords. An accurate recognition of a functional part is only possible, if its content is different from the other parts. This condition holds for methodology and evaluation in papers. But the evaluation also shows that other functional types, which for instance contain summarizations of other parts, are difficult to recognize with the presented approach.

Further investigations are needed to improve the quality of the functional structure analysis. Especially the analysis of structures, which refer to other parts, has to be improved. It is also an open question for future research, whether the presented approach is applicable to other document types and how to deal with multiple languages.

6.3 Variable Text Scaling and Structure Visualization

An important role for the development of a logical or functional structure analysis is the analysis of features and the visualization of the analysis results. The ability to highlight interesting parts of a document and preserving the global layout is important for the analysis of geometric and position based features. It can also be used to visualize uncertainty of the automatic analysis algorithms and improve the visibility of the uncertain results.

The application of the variable text scaling technique is not limited to the structure analysis domain, but can be used to highlight keywords for instance. For this application the technique could be improved in several ways. For instance, it is questionable whether stop words could be removed to gain additional space for the interesting ones. Another opportunity for improvement is the question whether every occurrence of a keyword is of interest. It is conceivable not to highlight a keyword when the same keyword is already highlighted in the direct neighborhood.

Appendix A

Definition of Evaluation Measures

The evaluation of logical and functional structure analysis uses the following measurements to asses the quality of the algorithms:

		Prediction	
		positive	negative
Truth	positive	True Positive (TP)	False Negative (FN)
	negative	False Positive (FP)	True Negative (TN)

Accuracy

$$\begin{aligned} \text{accuracy} &= \frac{TP+TN}{TP+FP+TN+FN} \\ &= \frac{\text{correct classified}}{\text{correct classified + incorrect classified}} \end{aligned}$$

Precision and Recall

$$\text{precision} = \frac{TP}{TP+FP}$$

$$\text{recall} = \frac{TP}{TP+FN}$$

F-Measure

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$
$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \text{harmonic mean}$$

Sensitivity and Specificity

$$\text{sensitivity} = \frac{TP}{TP+FN} = \text{recall}$$
$$\text{specificity} = \frac{TN}{TN+FP}$$

AUC The area under the ROC curve. The advantage of the AUC measure is its property to be independent from the class distribution. ROC curves are depicting the sensitivity and specificity of a classifier over all possible cut-off values. Because the sensitivity is only depending on positive and the specificity only on negative examples, the distribution of positive and negative examples is not effecting the ROC curve or the AUC.

Appendix B

Logical Structure Analysis

B.1 Features for Logical Structure Analysis

Table B.1: Geometric Features

Feature	Description
X Position	The x-coordinate of a line's baseline.
Y Position	The y-coordinate of a line's baseline.
Indentation Left	The size of the gap left of a line and it's column.
Indentation Left Change	The change of Indent Left between a line and it's predecessor.
Indentation Left Frequency	The relative frequency of Indention Left.
Indentation Right	The size of the gap right of a line and it's column.
Indentation Right Change	The change of Indent Right between a line and it's predecessor.
Indentation Right Frequency	The relative frequency of Indention Right.
Line Space	The size of the gap between a line and it's predecessor.

cont. on next page

Table B.1: Geometric Features (cont.)

Feature	Description
Line Space Change	The change of Line Space between a line and it's predecessor.
Line Space Frequency	The relative frequency of a line's Line Space.

Table B.2: Formatting Features

Feature	Description
Font Style	The font style (regular, bold, italics) of a line.
Font Size	The font size of a line.
Font Size Change	The change of the font size between a line and it's predecessor.
Font Size Frequency	The relative frequency of a line's Font Size.

Table B.3: Content Features

Feature	Description
Text Type of Word n	The text type (alpha, numeric, punct, alnum, ...) of the n^{th} token of a line.
Capitalization of Word n	The capitalization of the first character of the n^{th} token of a line.
Page Number	The page number of a page.
Reverse Page Number	The page number counting from the last page.
Paper Keywords	Whether a lines contains one of the typical keywords in papers (e.g. abstract, introduction, keywords, ...).

Table B.4: Computer Science Publication Features

B.2 Results of the Classifier Selection

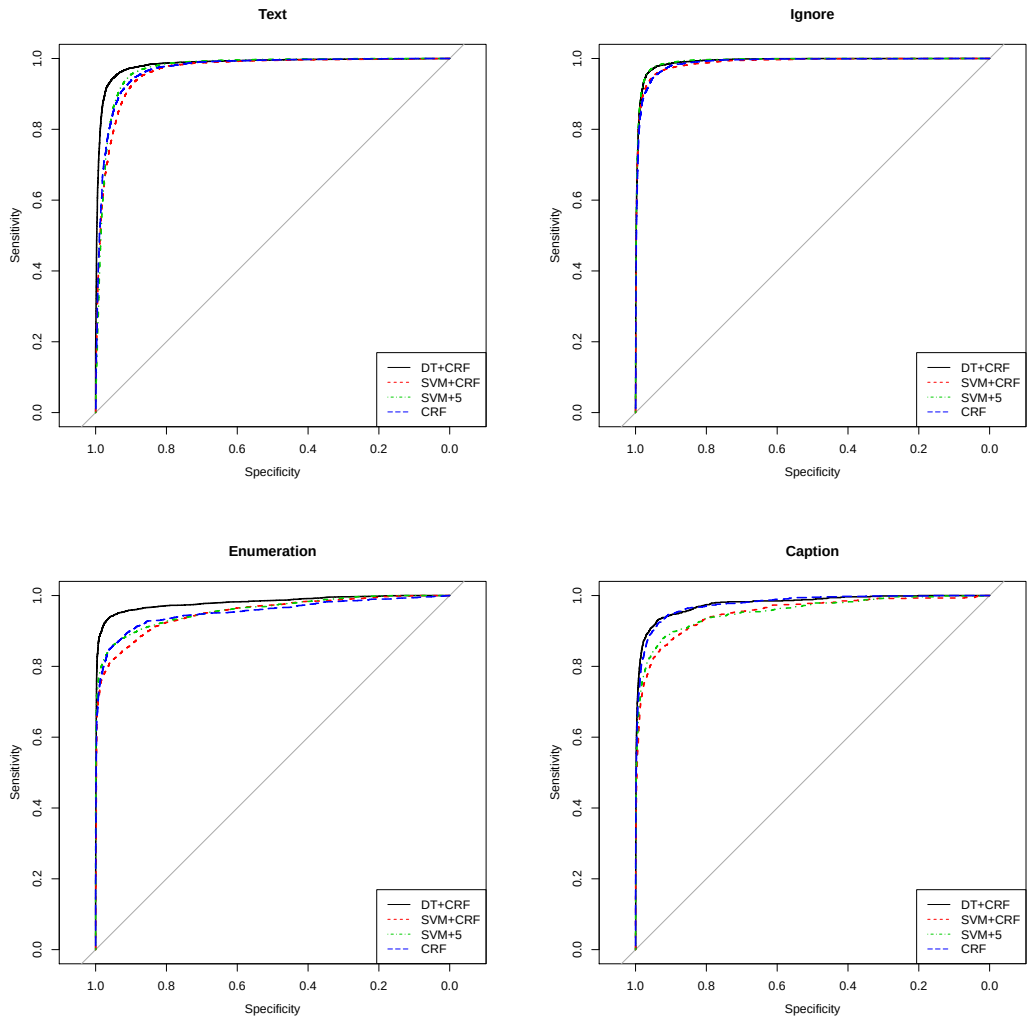


Figure B.1: ROC curves for the four best classifiers on the different labels. ↵

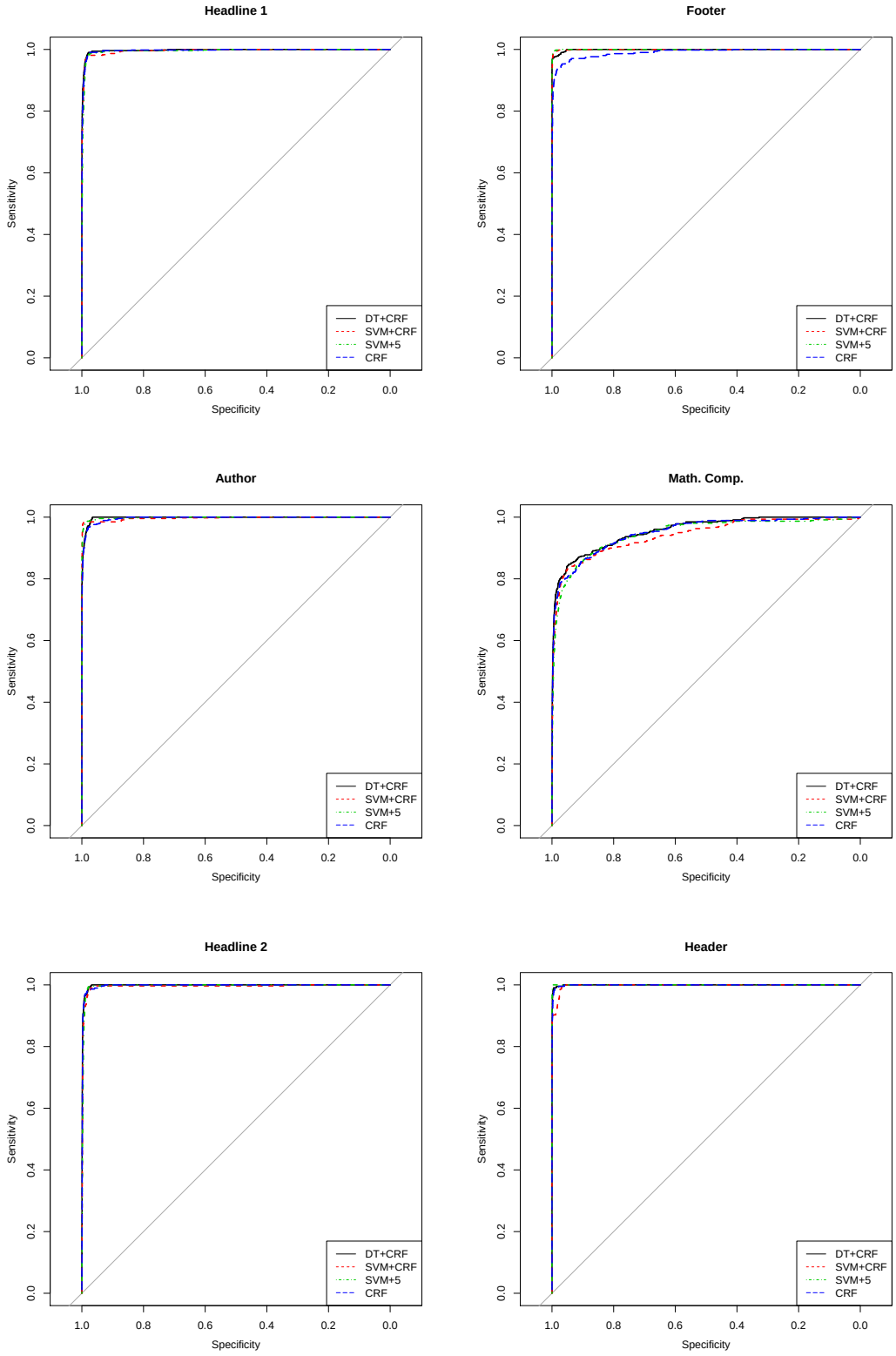


Figure B.1: ROC curves for the four best classifiers on the different labels. (cont.)

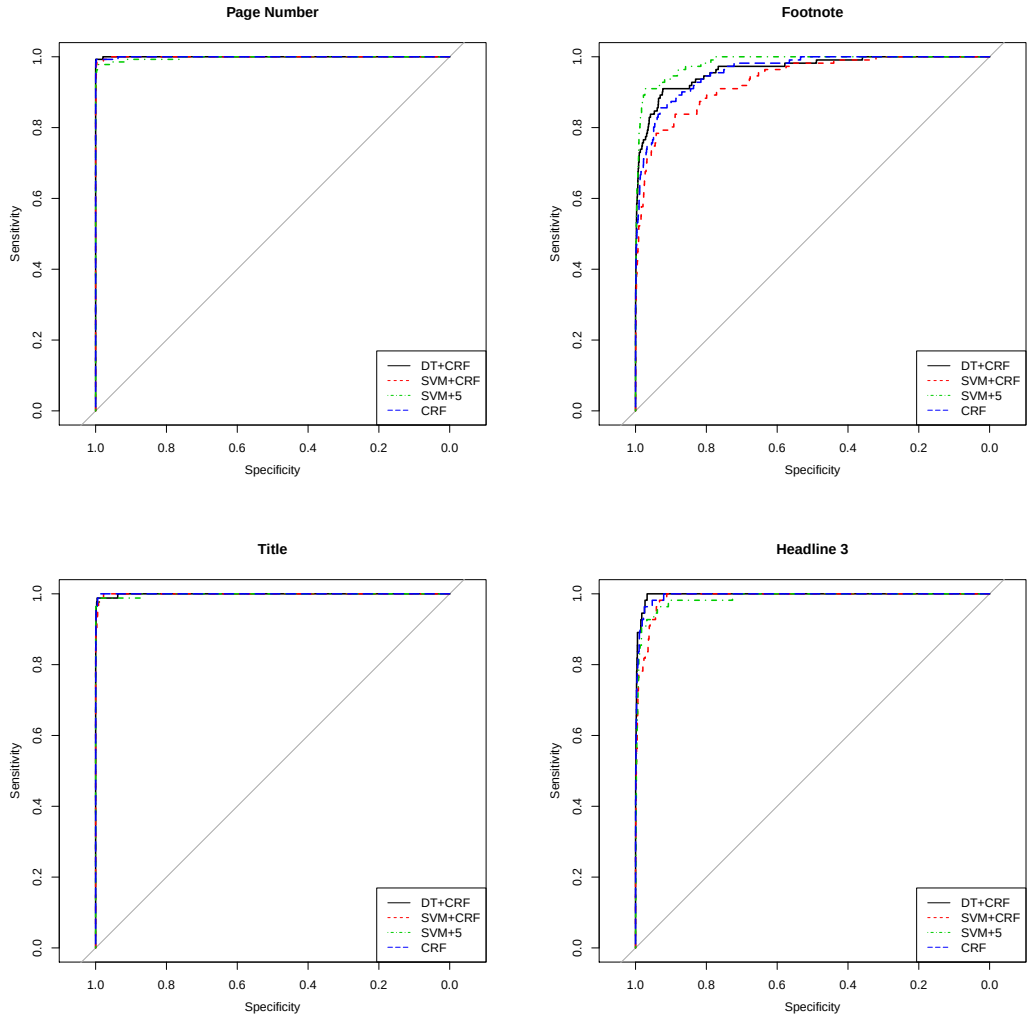


Figure B.1: ROC curves for the four best classifiers on the different labels. (cont.)

B.3 Iterative Learning of Structure Analysis Model

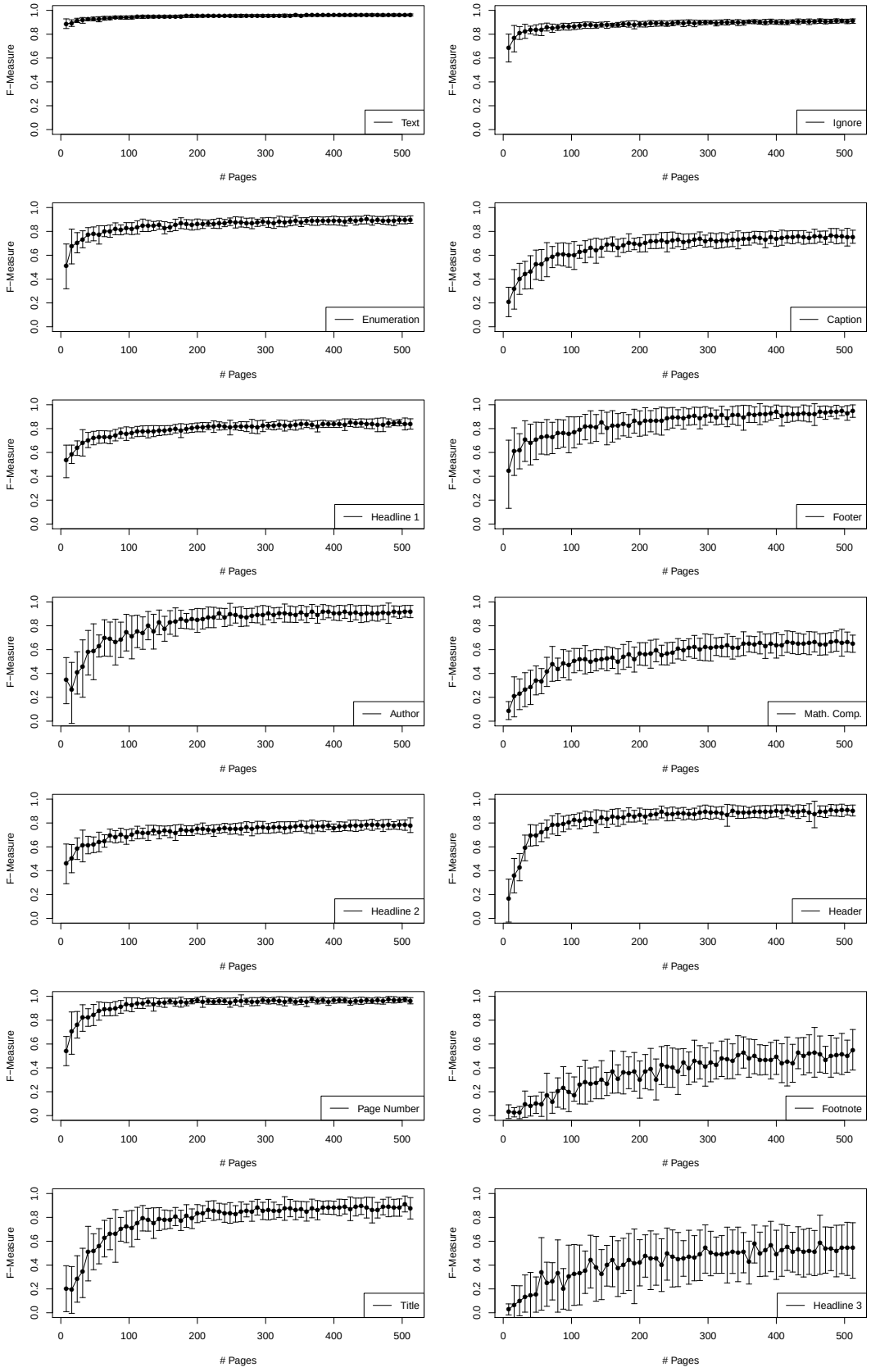


Figure B.2: F-measure of logical structures over different number of pages in the training set. The error bars show the standard deviation of the F-measure.

B.4 Evaluation Results

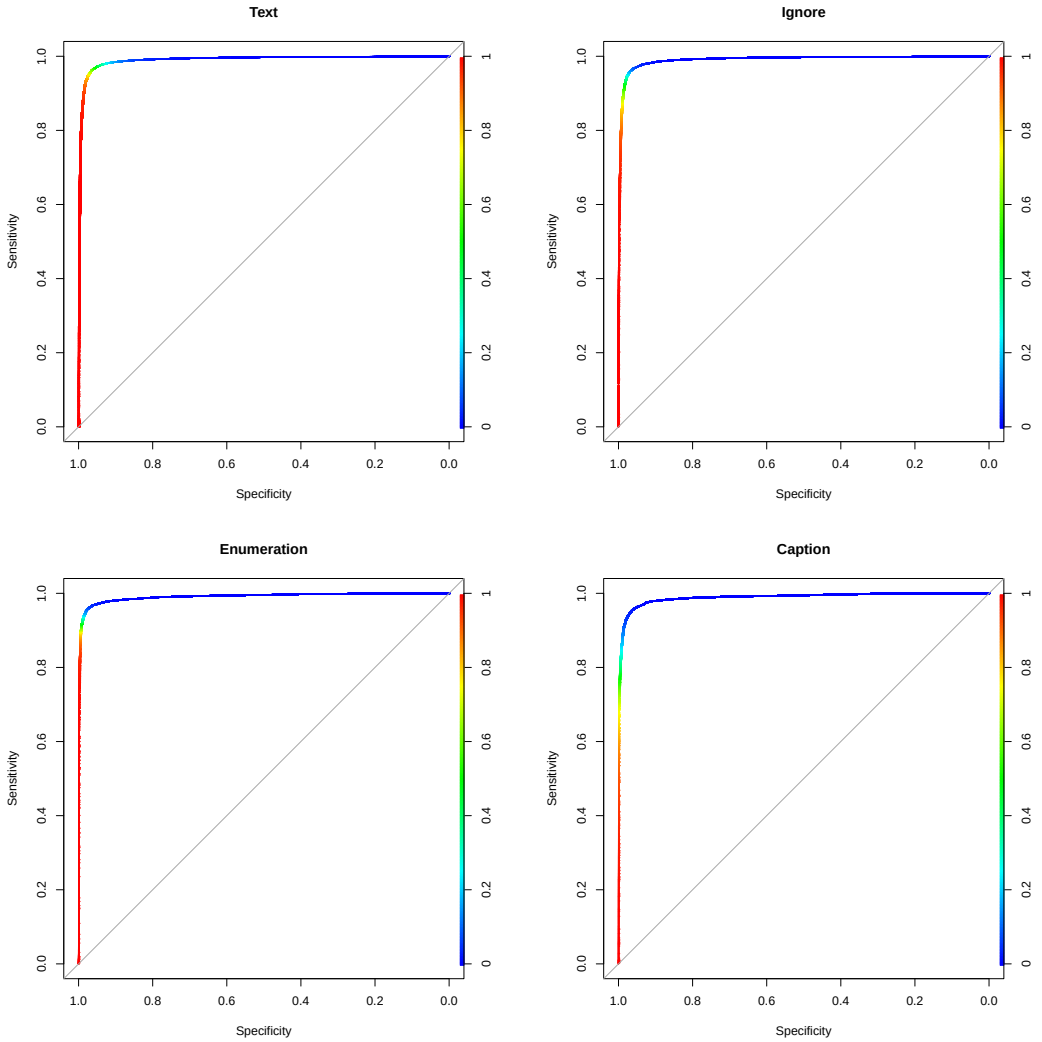


Figure B.3: ROC curves for the DT+CRF on the paper collection. The color of the ROC curves show the cut-off value for classification. ↵

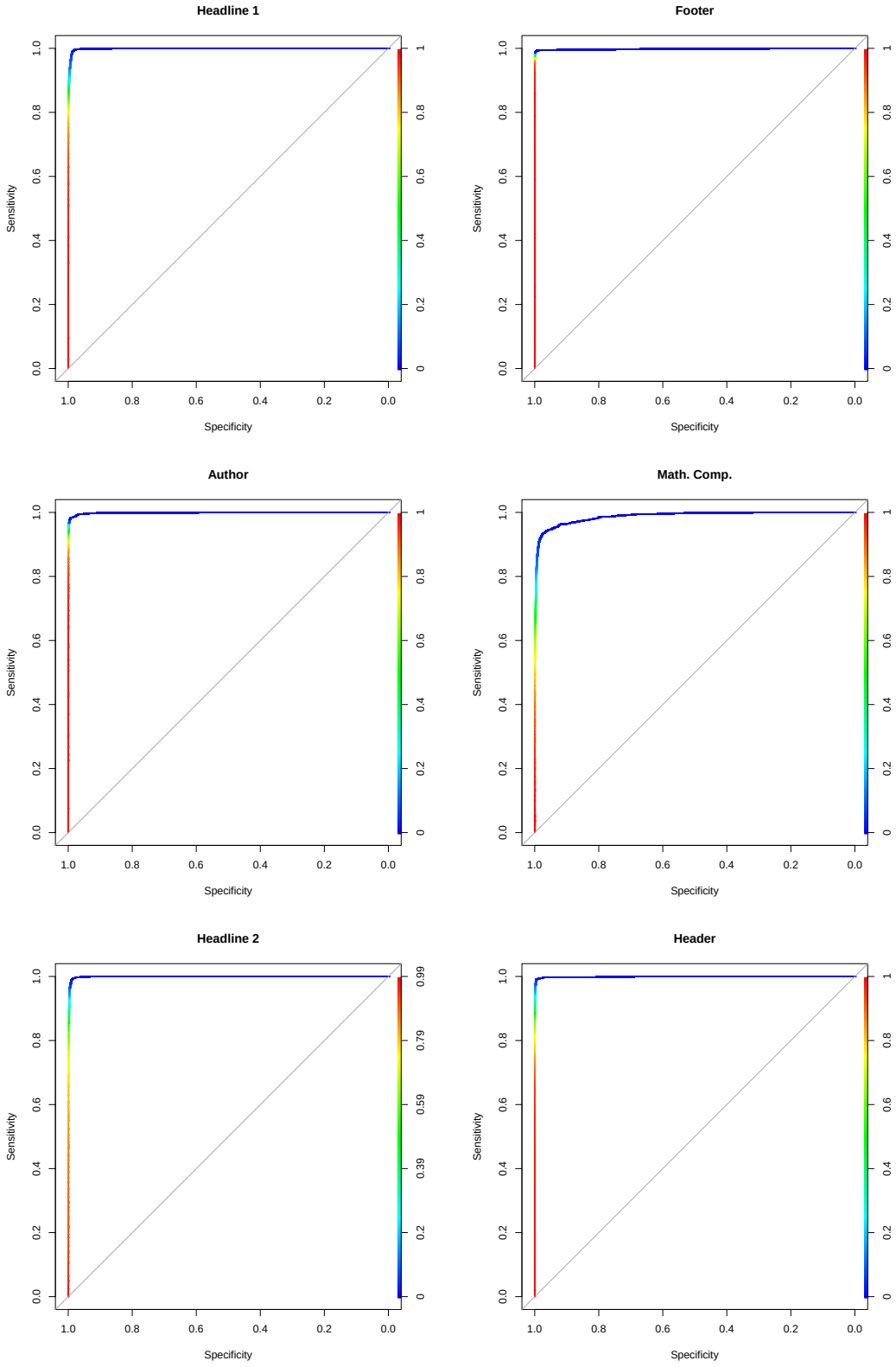


Figure B.3: ROC curves for the DT+CRF on the paper collection. The color of the ROC curves show the cut-off value of the classifier. (cont.)

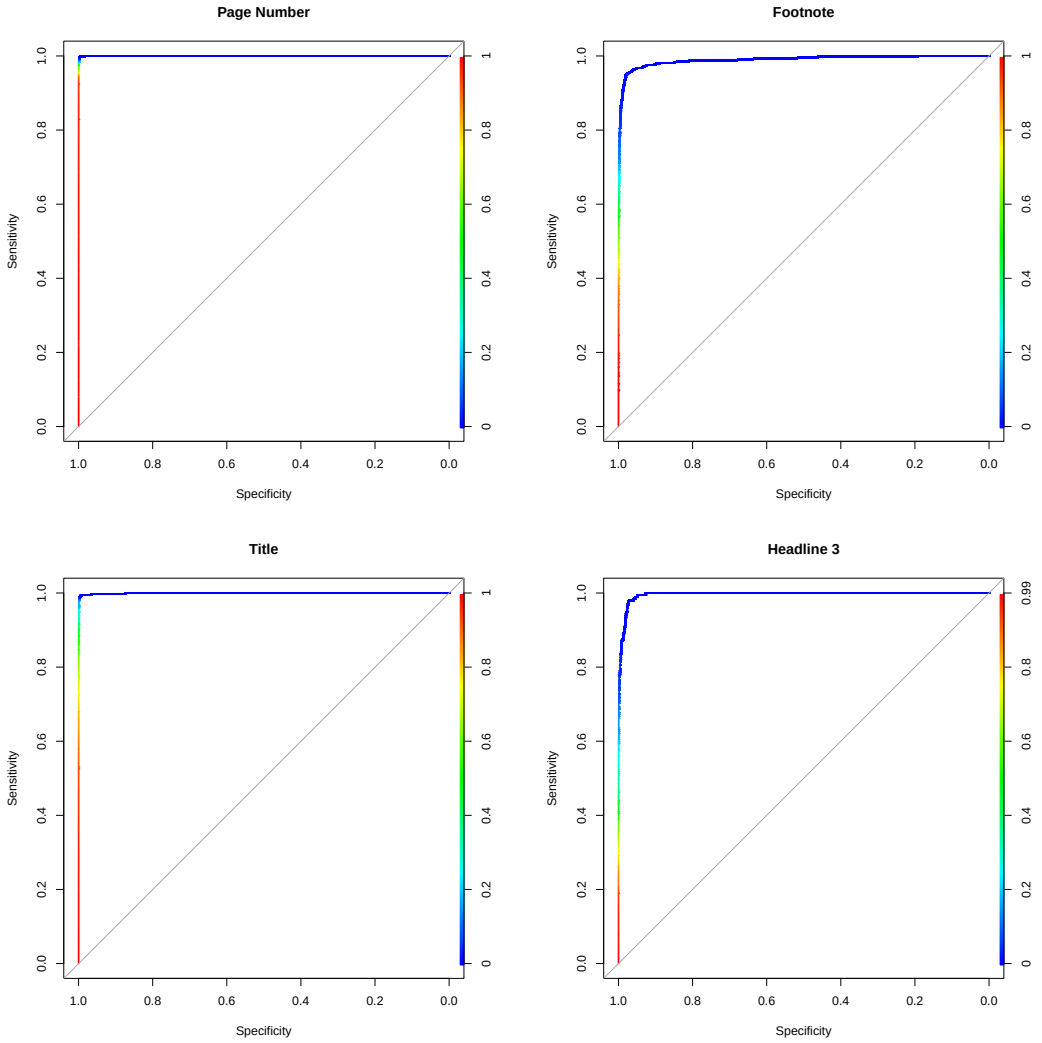


Figure B.3: ROC curves for the DT+CRF on the paper collection. The color of the ROC curves show the cut-off value for classification. (cont.)

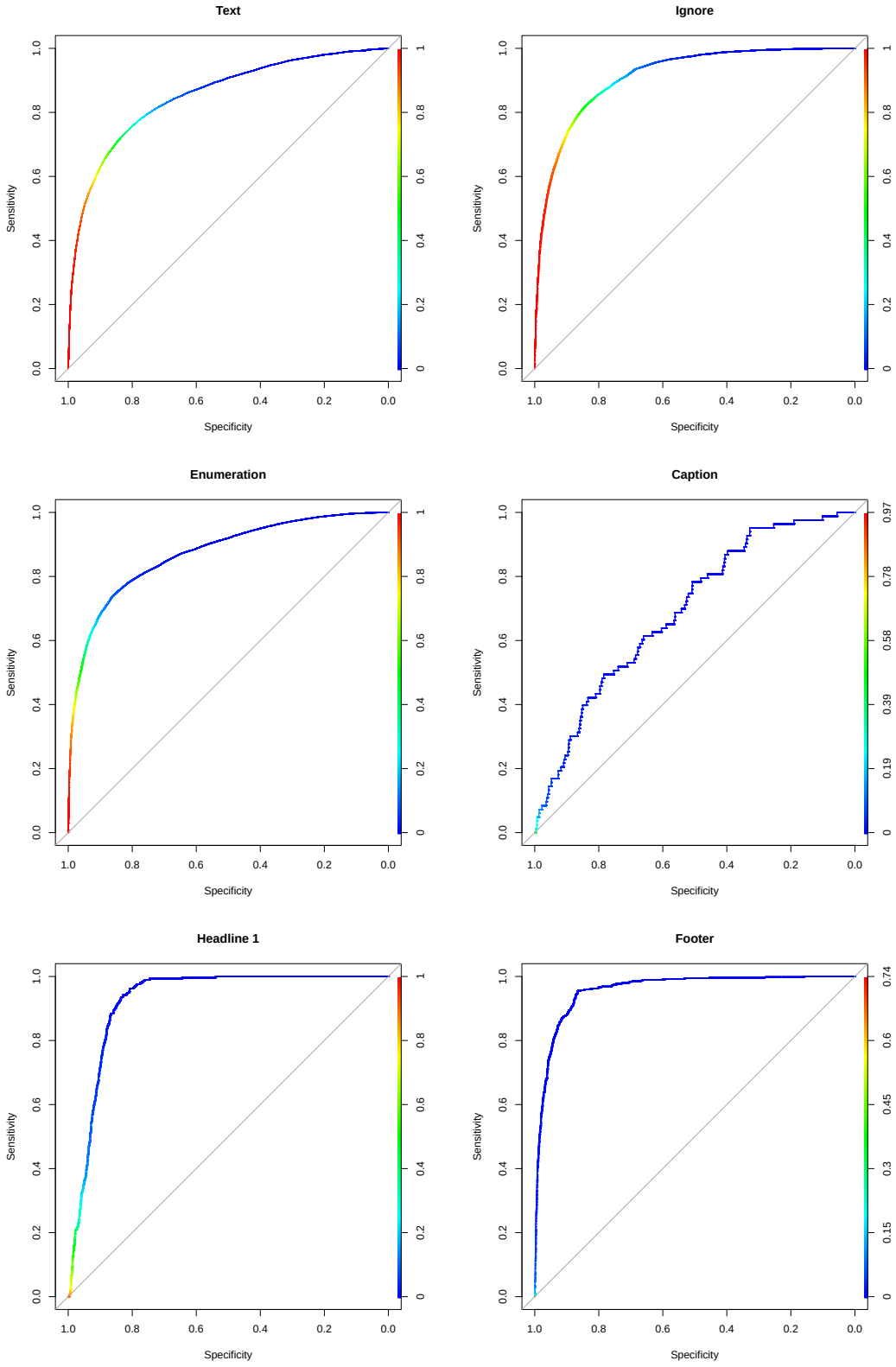


Figure B.4: ROC curves for the DT+CRF trained on the paper collection but applied to the product manuals. The color of the ROC curves show the cut-off value for classification.

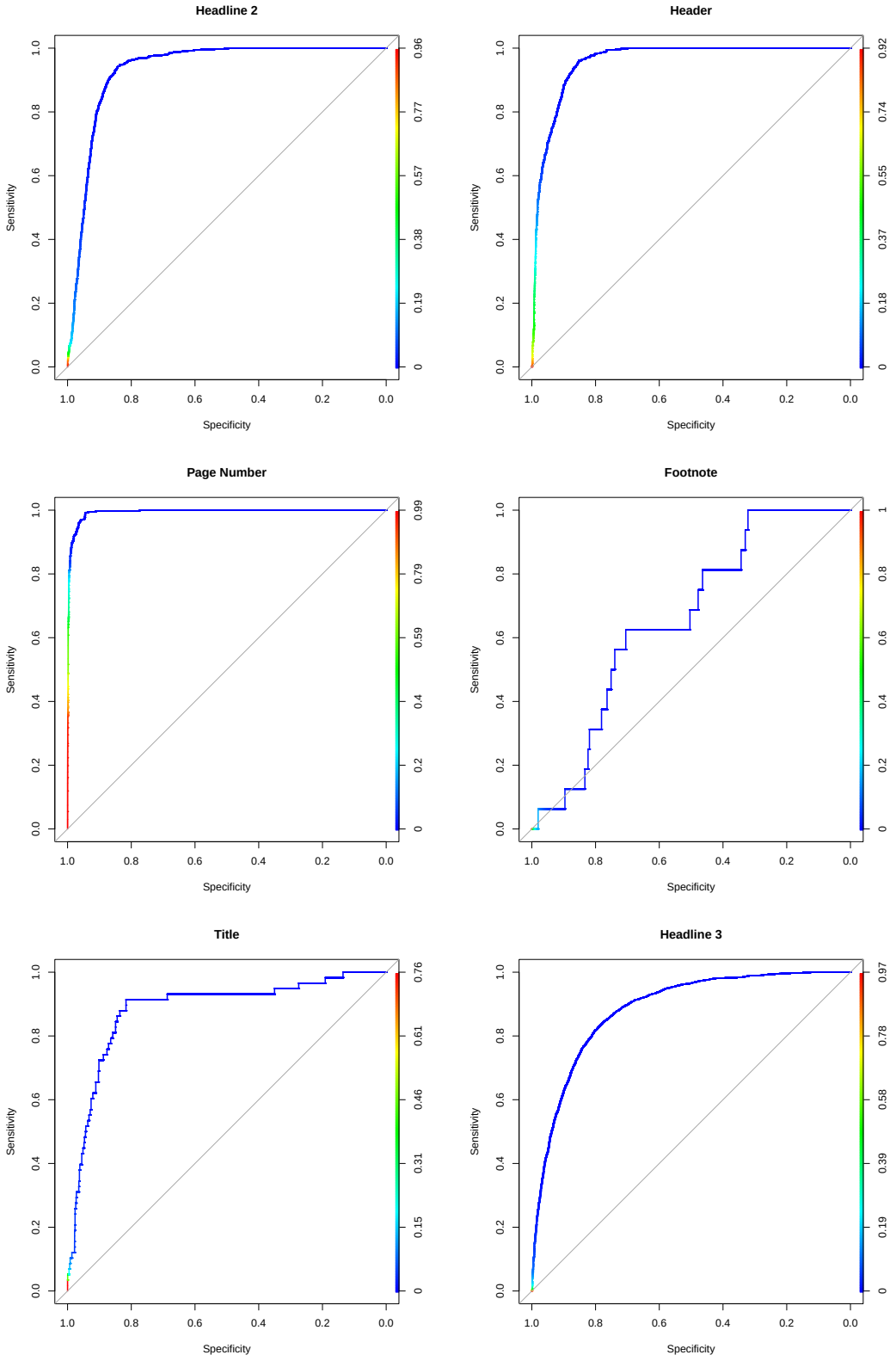


Figure B.4: ROC curves for the DT+CRF trained on the paper collection but applied to the product manuals. The color of the ROC curves show the cut-off value for classification. (cont.)

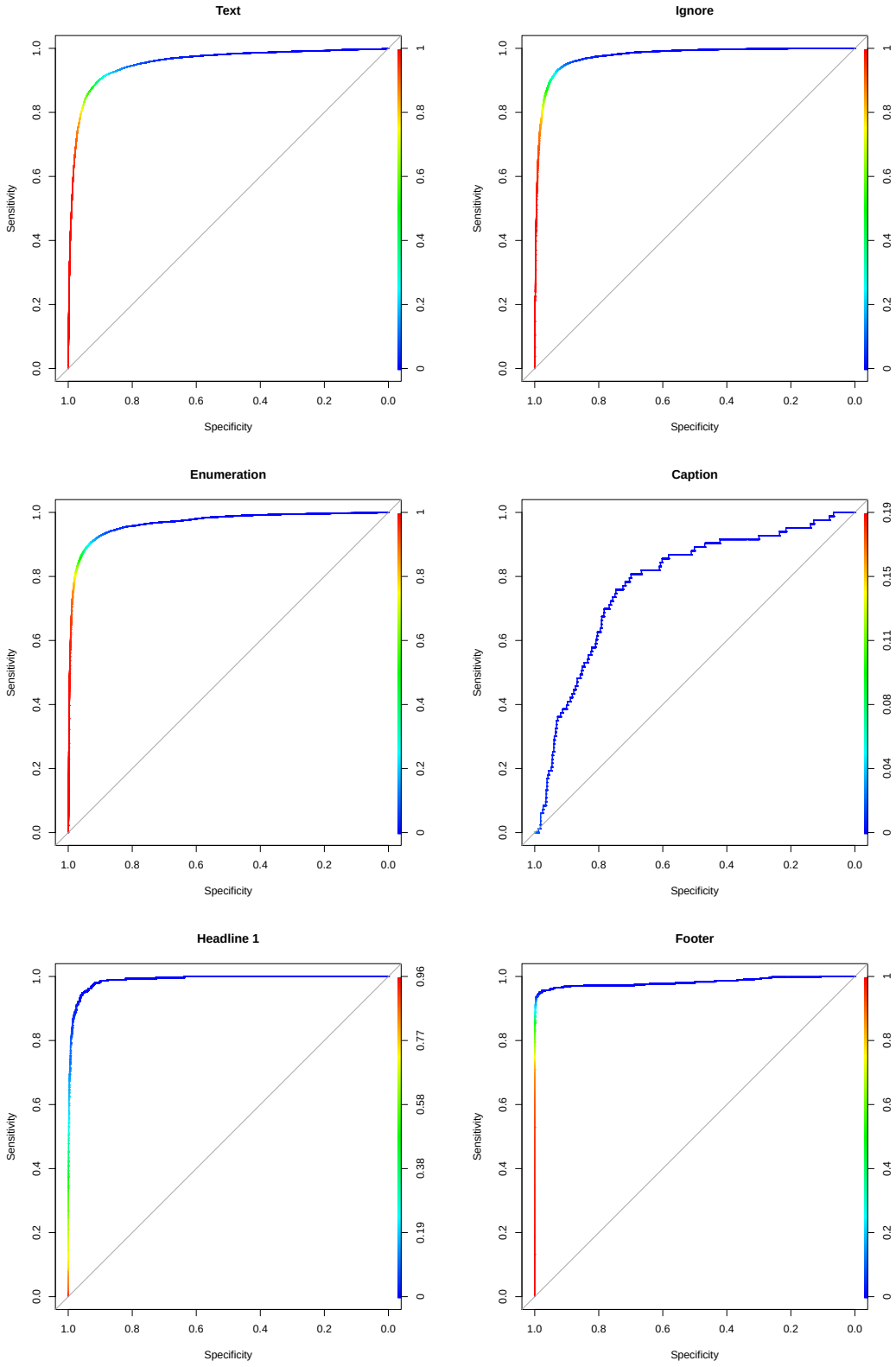


Figure B.5: ROC curves for the DT+CRF for the product manuals. The color of the ROC curves show the cut-off value for classification.

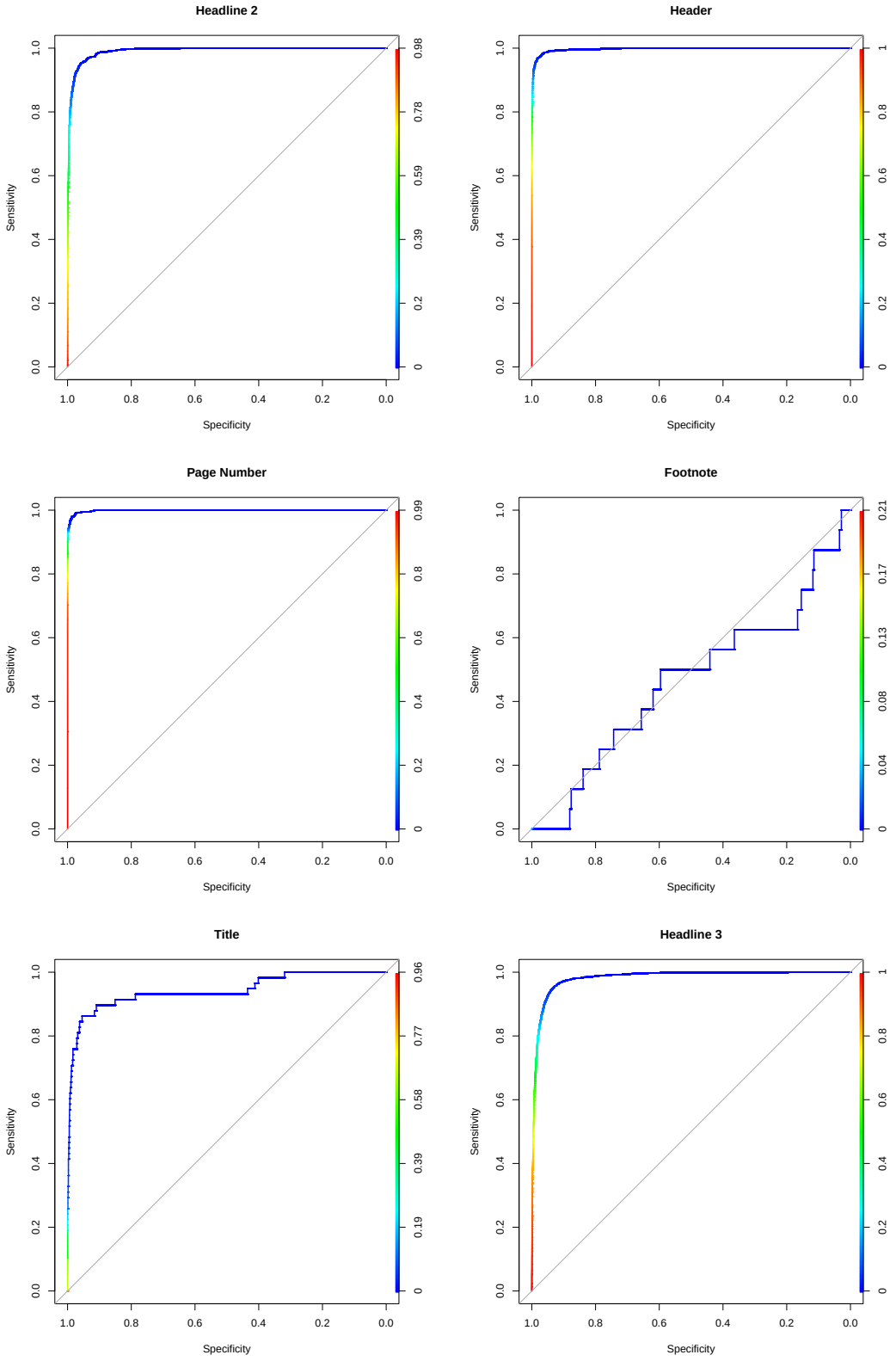


Figure B.5: ROC curves for the DT+CRF for the product manuals. The color of the ROC curves show the cut-off value for classification. (cont.)

Appendix C

Functional Structure Analysis

C.1 Results of Classifier Selection

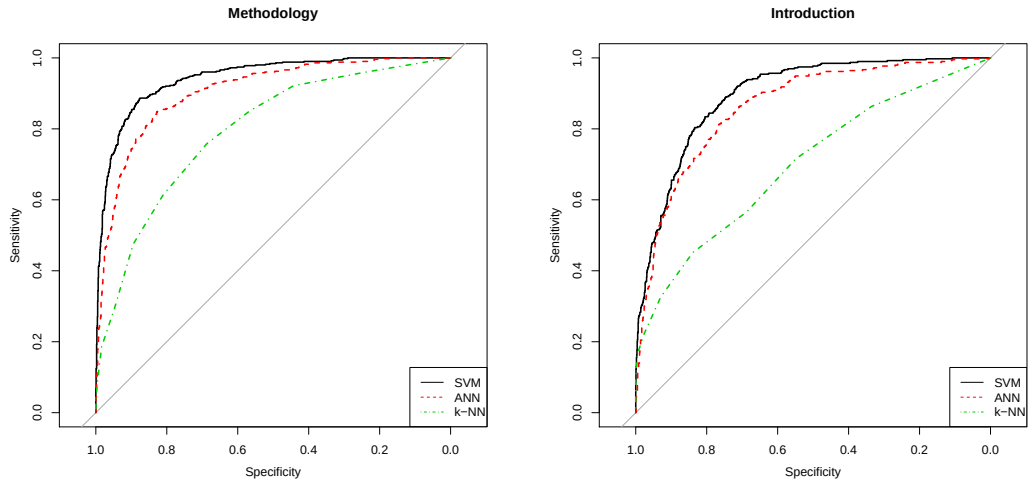


Figure C.1: ROC curves for the three best classifiers on the different functional structures at a text length of 300 words. ↵

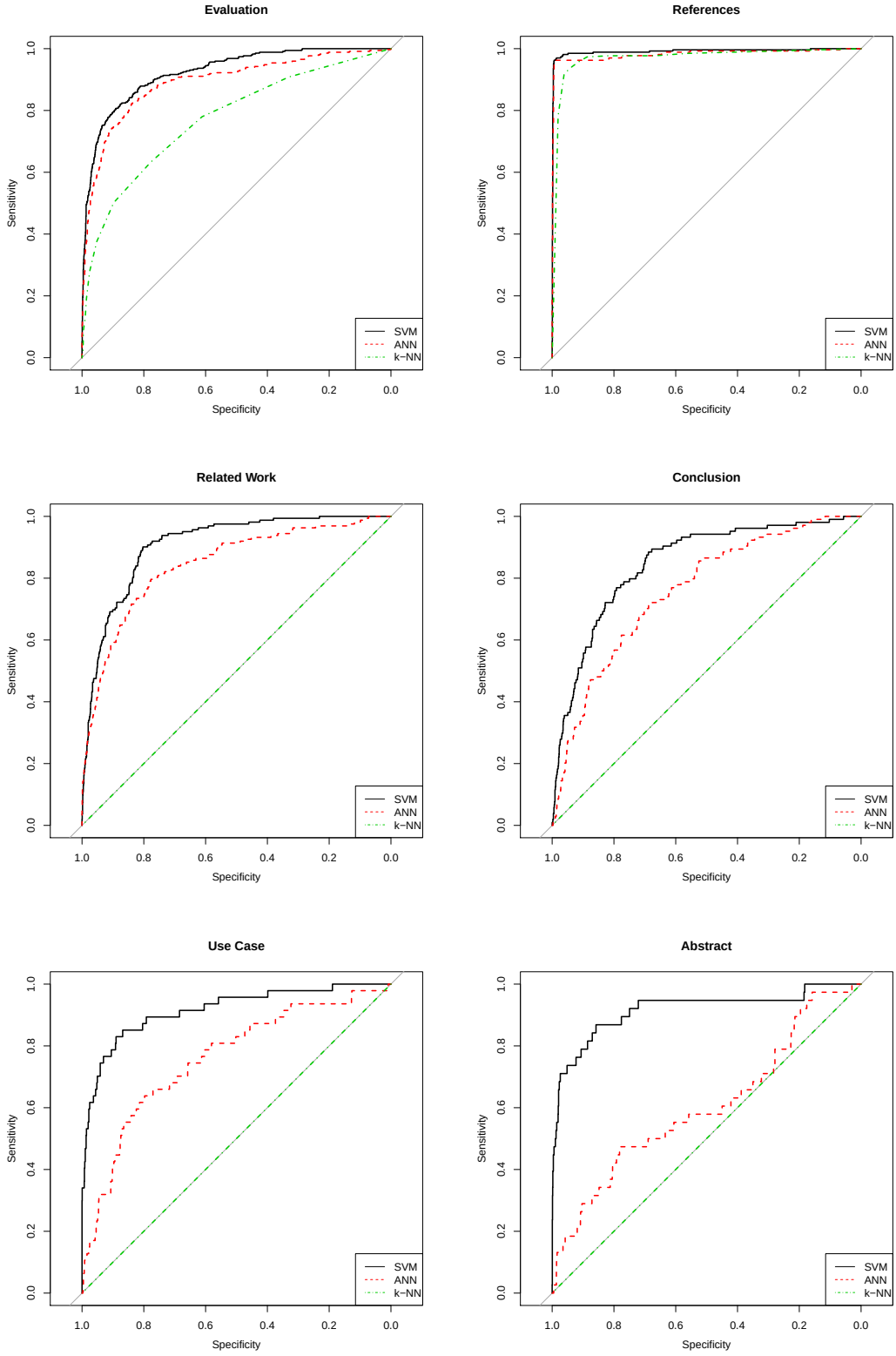


Figure C.1: ROC curves for the three best classifiers on the different functional structures at a text length of 300 words. (cont.)

C.2 Evaluation Results

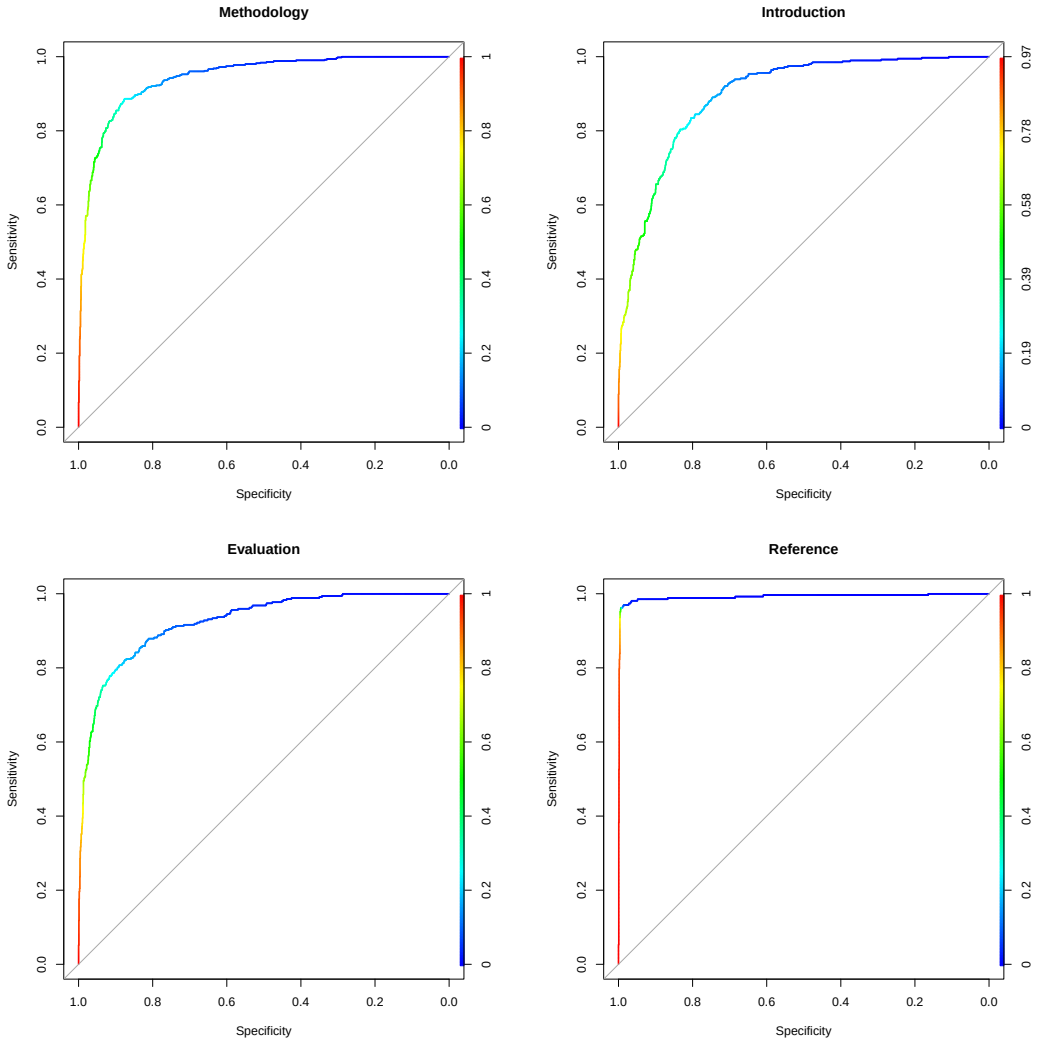


Figure C.2: ROC curves of the SVM classifier and the POS tagged features on the computer science data set. The color is mapped to the cut-off values. ↩

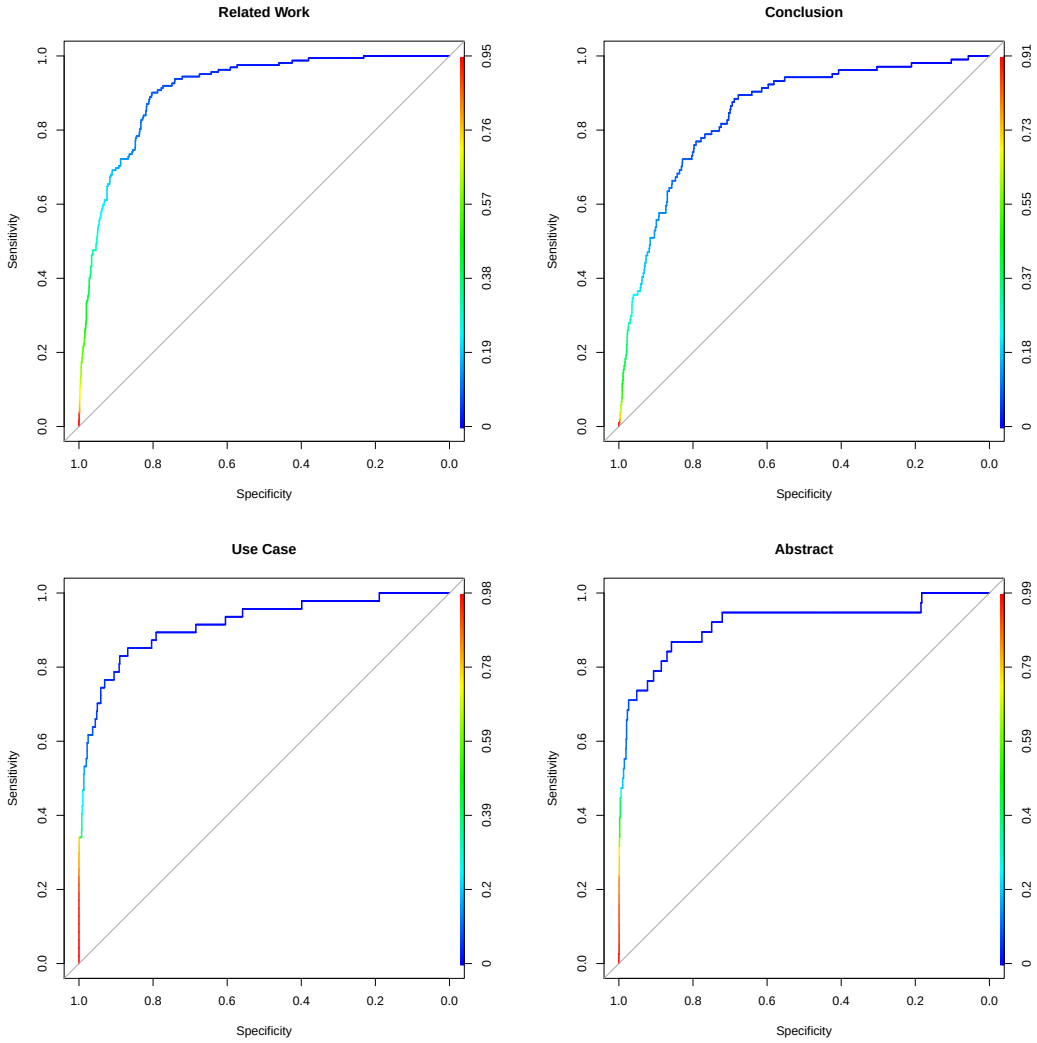


Figure C.2: ROC curves of the SVM classifier and the POS tagged features on the computer science data set. The color is mapped to the cut-off values. (cont.)

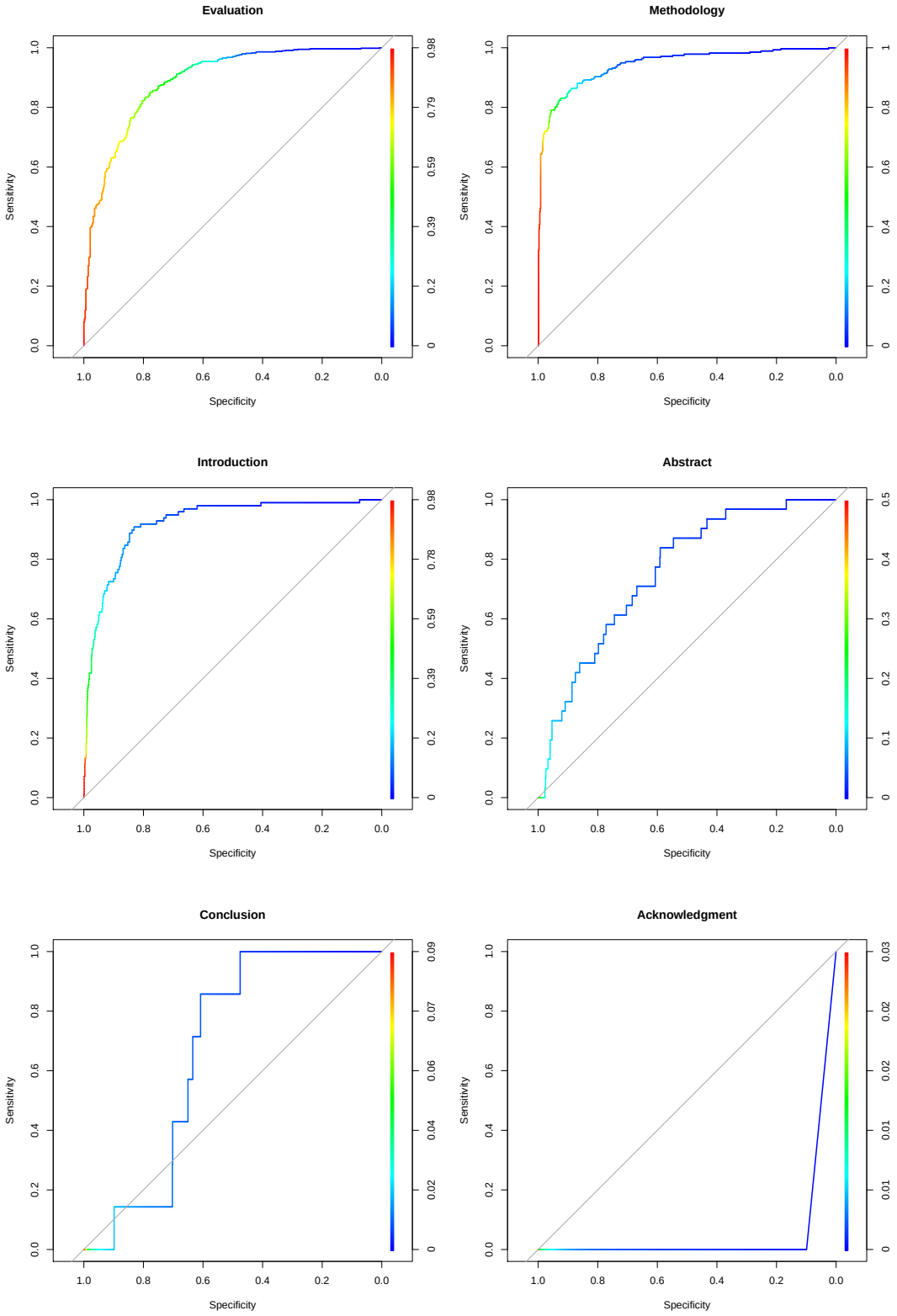


Figure C.3: ROC curves of the SVM classifier and the POS tagged features on the PubMed data set. The color is mapped to the cut-off values.

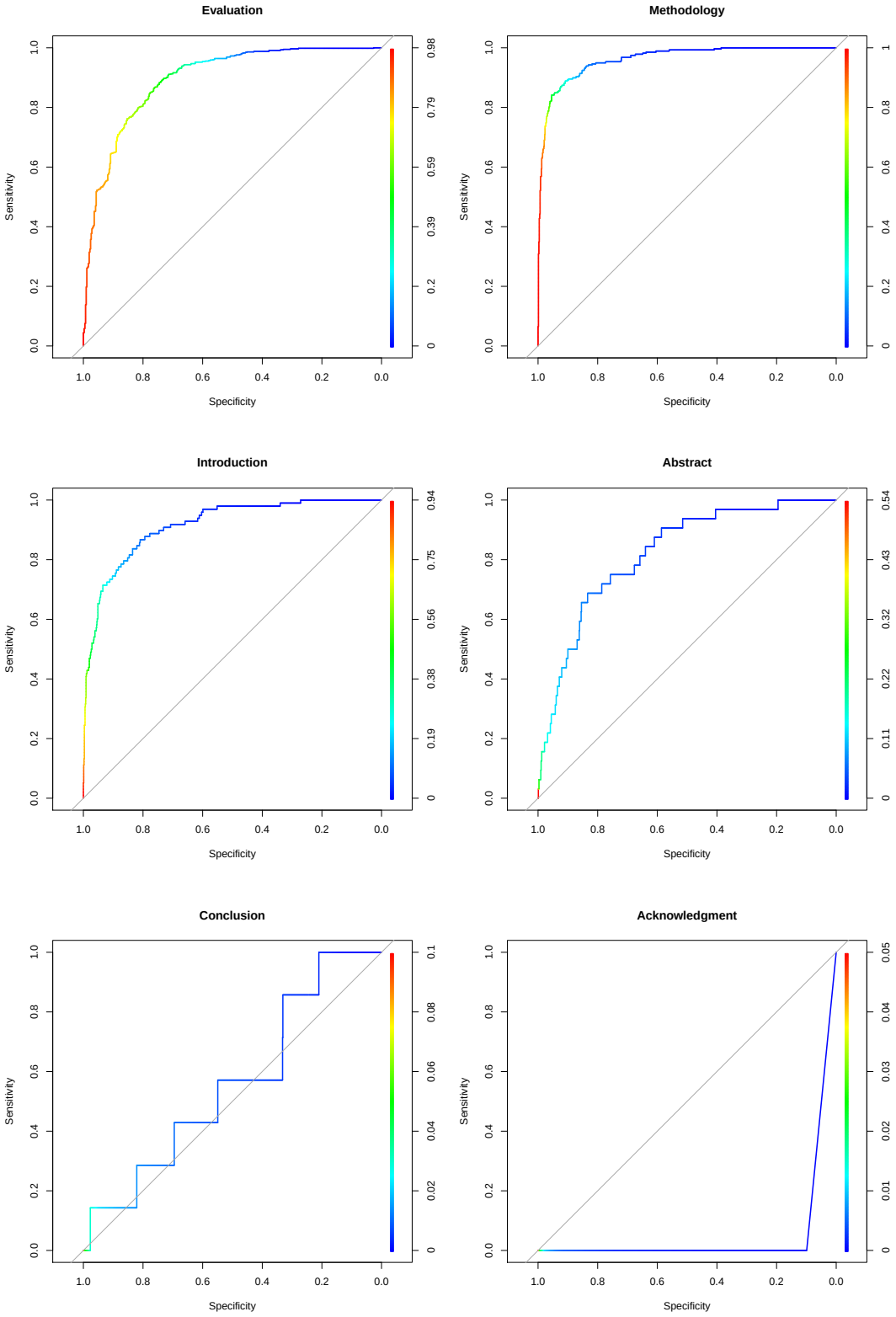


Figure C.4: ROC curves of the SVM classifier and the Abner tagged features on the PubMed data set. The color is mapped to the cut-off values.

Appendix D

Applications

D.1 Document Overview

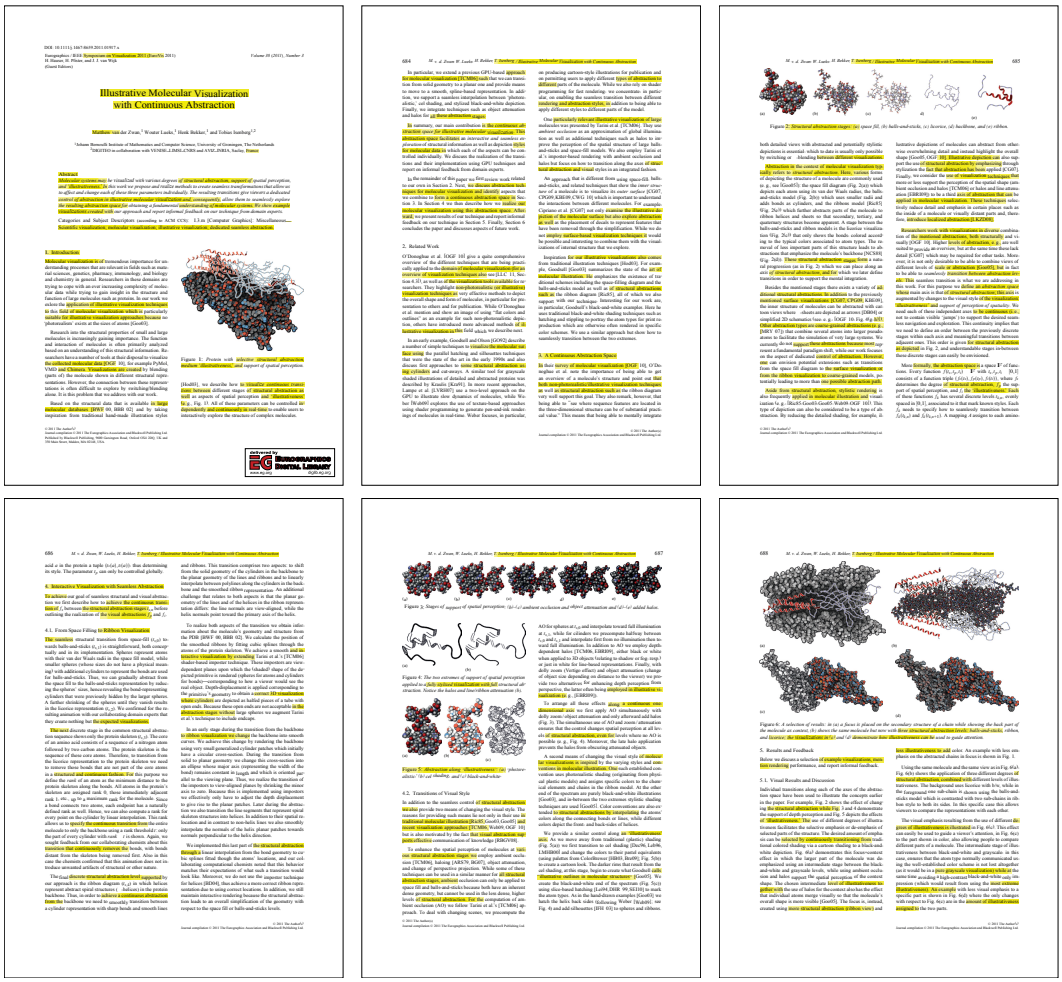


Figure D.1: Page thumbnails created for one of the EuroVis2011 paper [Zwa+11]. The nouns appearing in the title are highlighted in the whole document.

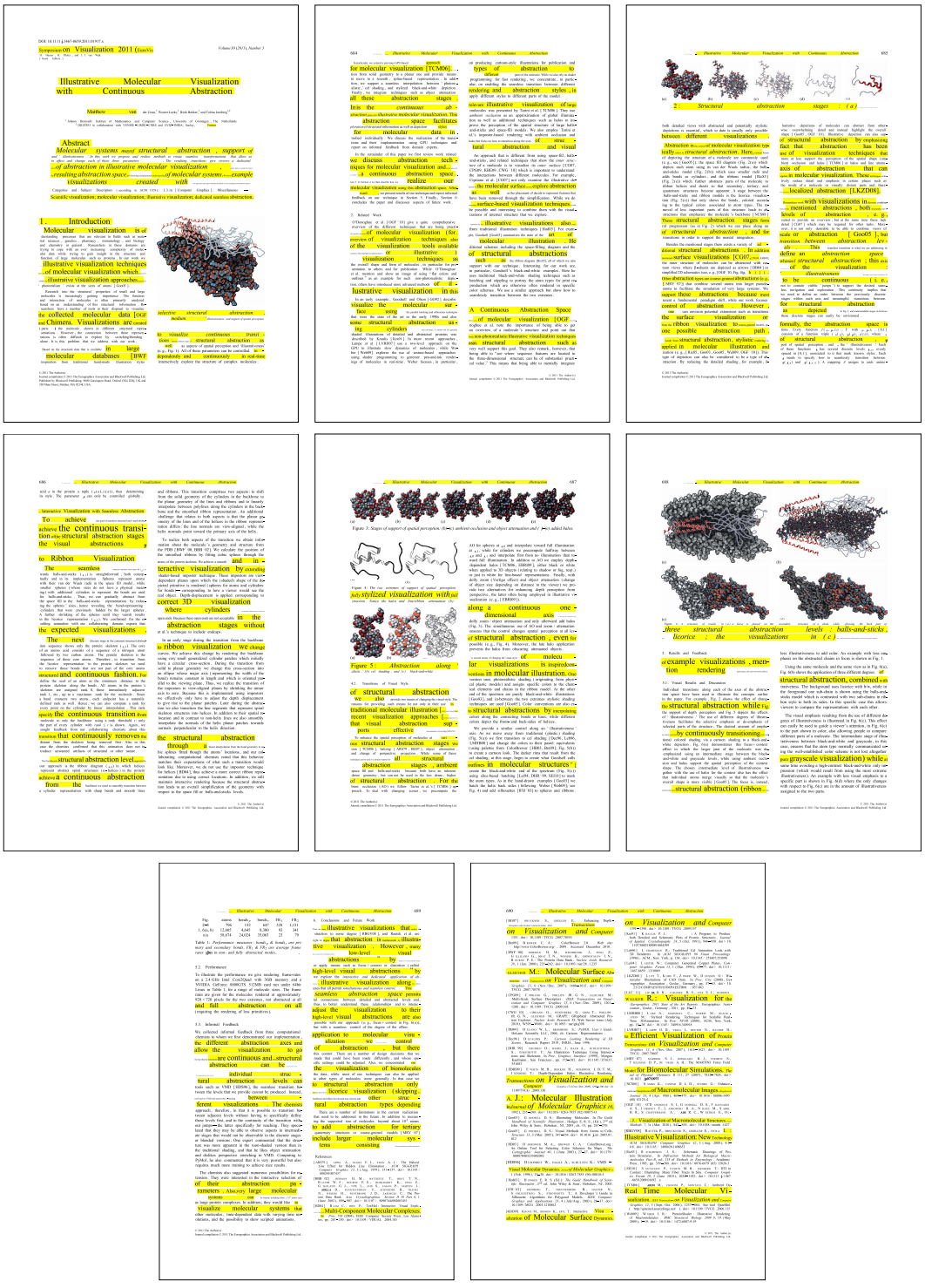


Figure D.2: Thumbnails of the same document shown in Figure D.1. In addition to highlighting, the size of interesting nouns is increased with variable text scaling.

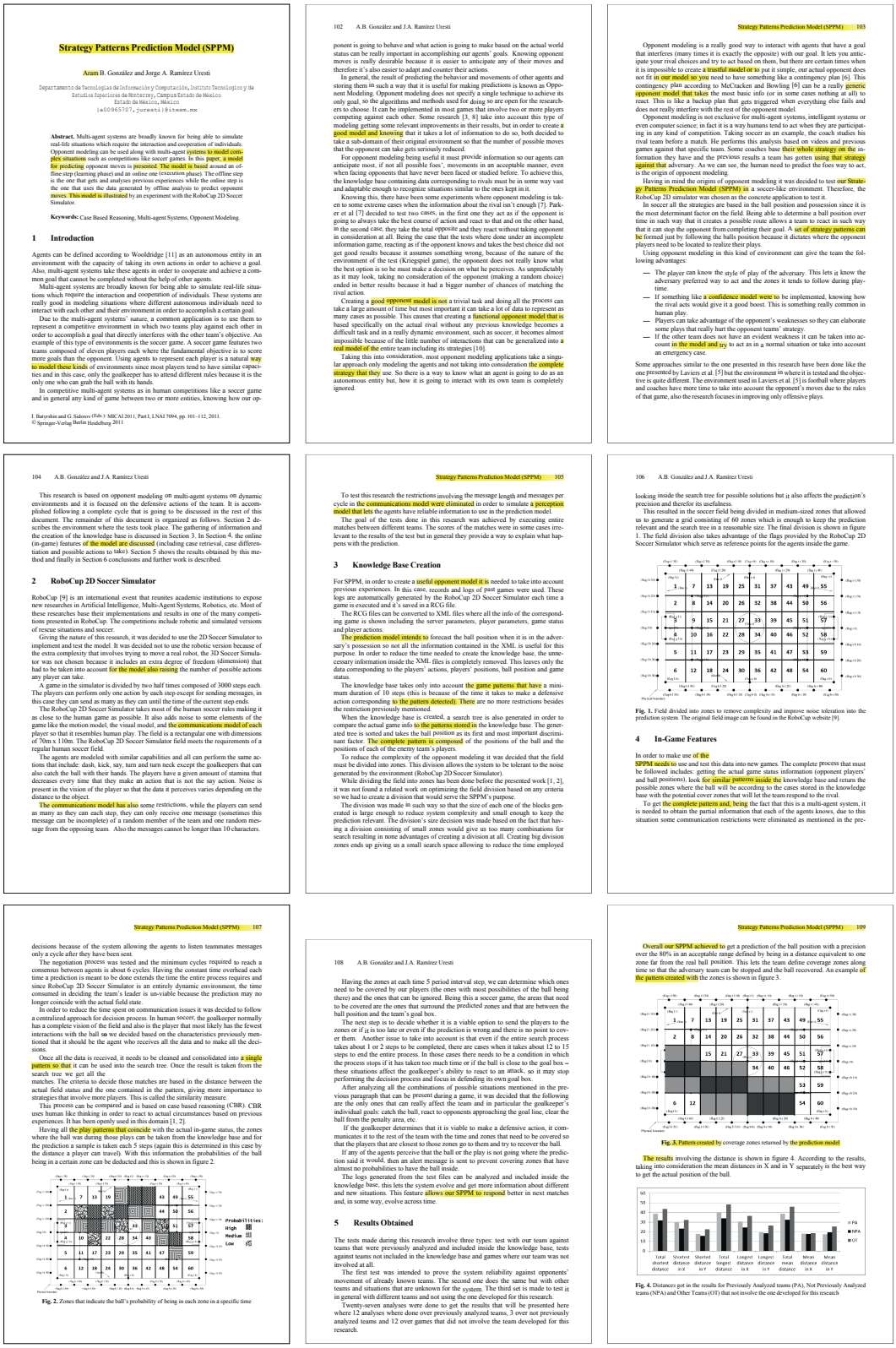


Figure D.3: Page thumbnails created for one of the EuroVis2011 paper [Zwa+11]. The nouns appearing in the title are highlighted in the whole document.

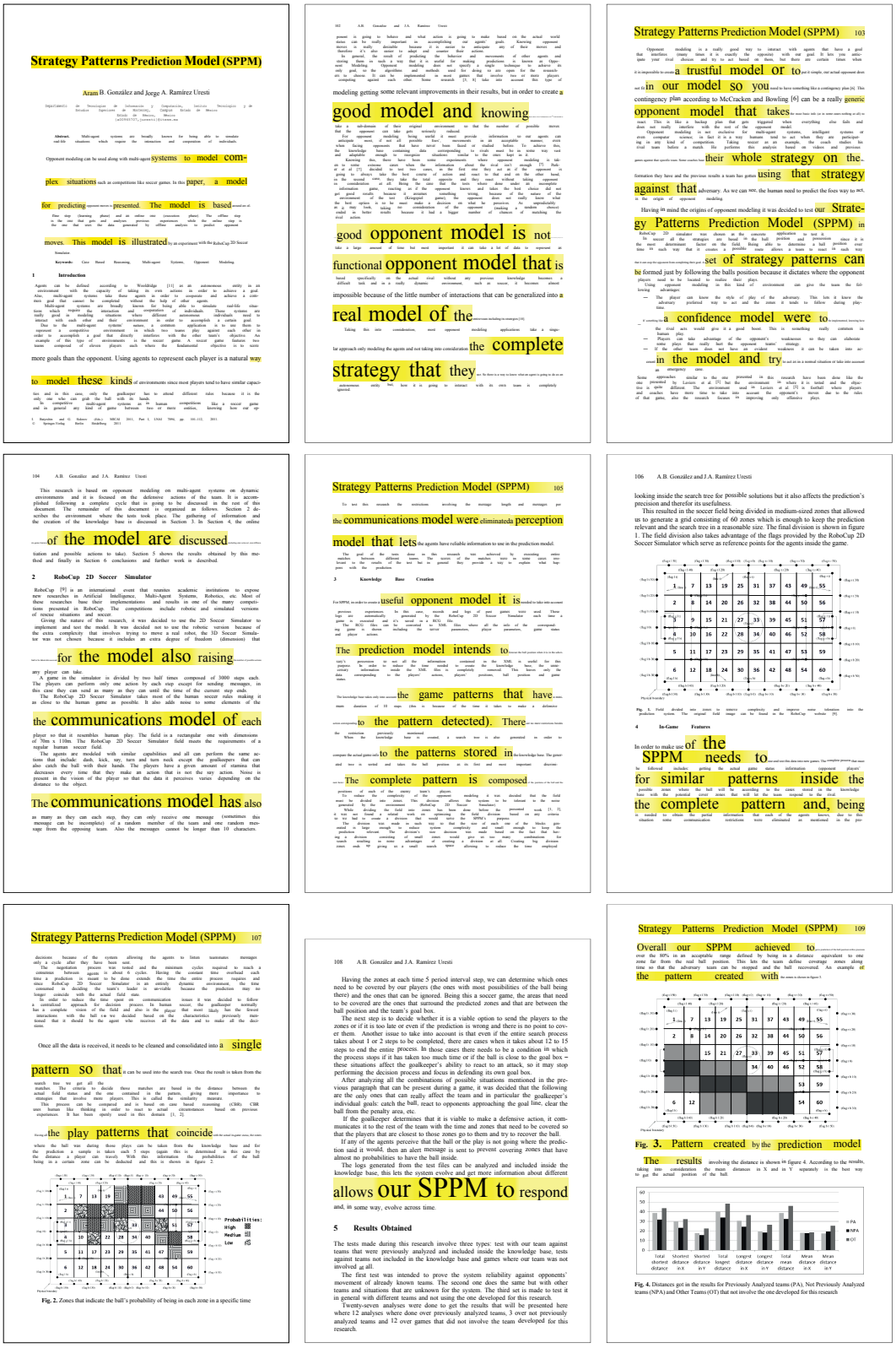


Figure D.4: Thumbnails of the same document shown in Figure D.3. In addition to highlighting, the size of interesting nouns is increased with variable text scaling.

Bibliography

[Anj01] Anjo Anjewierden. “AIDAS: Incremental Logical Structure Discovery in PDF Documents”. In: *6th International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2001, pp. 374–378.

[Bar+95] Lyn Bartram, Albert Ho, John Dill, and Frank Henigman. “The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces”. In: *Proceedings of the 8th annual ACM symposium on User interface and software technology*. ACM, 1995, pp. 207–215.

[Bau+04] Patrick Baudisch, Xing Xie, Chong Wang, and Wei-Ying Ma. “Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content”. In: *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. Ed. by Steven Feiner and James A. Landay. ACM, 2004, pp. 91–94.

[BB12] James Bergstra and Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.

[BES01] Frank Le Bourgeois, Hubert Emptoz, and Souad Souafi-Bensafi. “Document Understanding Using Probabilistic Relaxation: Application on Tables of Contents of Periodicals”. In: *6th International Conference on Document Analysis and Recognition (ICDAR 2001)*. IEEE Computer Society, 2001, pp. 508–512.

[BO08] George Buchanan and Tom Owen. “Improving Skim Reading for Document Triage”. In: *Proceedings of the 2nd International Conference on Information Interaction in Context*. Ed. by Mounia Lalmas, Anastasios Tombros, Pia Borlund, Jesper W. Schneider, Diane Kelly, John Feather, and Arjen P. de Vries. Vol. 348. ACM International Conference Proceeding Series. ACM, 2008, pp. 83–88.

[BPV00] Abdel Belaïd, L. Pierron, and N. Valverde. “Part-of-Speech Tagging for Table of Contents Recognition”. In: *ICPR*. 2000, pp. 4451–4454.

[Bre03] Thomas M. Breuel. “High Performance Document Layout Analysis”. In: *Symposium on Document Image Understanding Technology*. 2003, pp. 209–218.

[Buc97] Michael K. Buckland. “What Is a “Document”?” In: *Journal of the American Society for Information Science (JASIS)* 48.9 (1997), pp. 804–809.

[Byr99] Donald Byrd. “A Scrollbar-Based Visualization for Document Navigation”. In: *Proceedings of the Fourth ACM conference on Digital Libraries*. ACM, 1999, pp. 122–129.

[BZI97] Rolf Brugger, Abdel Wahab Zramdini, and Rolf Ingold. “Modeling Documents for Structure Recognition Using Generalized N-Grams”. In: *4th International Conference Document Analysis and Recognition*. IEEE Computer Society, 1997.

[CGA06] Andy Cockburn, Carl Gutwin, and Jason Alexander. “Faster Document Navigation with Space-Filling Thumbnails”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Ed. by Rebecca E. Grinter, Tom Rodden, Paul M. Aoki, Edward Cutrell, Robin Jeffries, and Gary M. Olson. ACM, 2006, pp. 1–10.

[CL11] Chih-Chung Chang and Chih-Jen Lin. “LIBSVM: A Library for Support Vector Machines”. In: *ACM TIST* 2.3 (2011), p. 27.

[CN06] Rich Caruana and Alexandru Niculescu-Mizil. “An Empirical Comparison of Supervised Learning Algorithms”. In: *Proceedings of the 23rd international conference on Machine learning*. ICML ’06. Pittsburgh, Pennsylvania: ACM, 2006, pp. 161–168.

[Cze+99] Mary P. Czerwinski, Maarten van Dantzich, George Robertson, and Hunter Hoffman. “The Contribution of Thumbnail Image, Mouse-over Text and Spatial Location Memory to Web Page Retrieval in 3D”. In: *Human-Computer Interaction: INTERACT ’99*. Ed. by Martina Angela Sasse and Chris Johnson. IOS Press, 1999, pp. 163–170.

[DC02] Susan Dziadosz and Raman Chandrasekar. “Do Thumbnail Previews Help Users Make Better Relevance Decisions About Web Search Results?” In: *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2002, pp. 365–366.

[DM09] Hervé Déjean and Jean-Luc Meunier. “On tables of contents and how to recognize them”. In: *IJDAR* 12.1 (2009), pp. 1–20.

[FS96] Yoav Freund and Robert E. Schapire. “Experiments with a New Boosting Algorithm”. In: *Machine Learning, Proceedings of the Thirteenth International Conference (ICML ’96)*. Ed. by Lorenza Saitta. Morgan Kaufmann, 1996, pp. 148–156.

[Gao+09] Liangcai Gao, Zhi Tang, Xiaofan Lin, Xin Tao, and Yimin Chu. “Analysis of Book Documents’ Table of Content Based on Clustering”. In: *10th International Conference on Document Analysis and Recognition (ICDAR 2009)*. IEEE Computer Society, 2009, pp. 911–915.

[GU11] Aram B. González and Jorge A. Ramírez Uresti. “Strategy Patterns Prediction Model (SPPM)”. In: *Advances in Artificial Intelligence - 10th Mexican International Conference on Artificial Intelligence, MICAI 2011, Proceedings, Part I*. Ed. by Ildar Z. Batyrshin and Grigori Sidorov. Vol. 7094. Lecture Notes in Computer Science. Springer, 2011, pp. 101–112.

[Hal+09] Mark 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.

[Hal98] M. A. Hall. “Correlation-based Feature Subset Selection for Machine Learning”. PhD thesis. Hamilton, New Zealand: University of Waikato, 1998.

[Hea95] Marti A. Hearst. “TileBars: Visualization of Term Distribution Information in Full Text Information Access”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. Ed. by Irvin R. Katz, Robert L. Mack, Linn Marks, Mary Beth Rosson, and Jakob Nielsen. ACM/Addison-Wesley, 1995, pp. 59–66.

[HF01] Kasper Hornbæk and Erik Frøkjær. “Reading of Electronic Documents: The Usability of Linear, Fisheye, and Overview+Detail Interfaces”. In: *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems*. Ed. by Julie A. Jacko and Andrew Sears. ACM, 2001, pp. 293–300.

[HNZ05] John C. Handley, Anoop M. Namboodiri, and Richard Zanibbi. “Document Understanding System Using Stochastic Context-Free Grammars”. In: *Eighth International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2005, pp. 511–515.

[HT01] David J. Hand and Robert J. Till. “A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems”. In: *Machine Learning* 45.2 (2001), pp. 171–186.

[HTF11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2011.

[Ish05] Yasuto Ishitani. “Logical Structure Analysis of Document Images Based on Emergent Computation”. In: *IEICE Transactions* 88-D.8 (2005), pp. 1831–1842.

[KDK00] Stefan Klink, Andreas Dengel, and Thomas Kieninger. “Document Structure Analysis Based on Layout and Textual Features”. In: *4th IAPR International Workshop on Document Analysis Systems*. Ed. by Nabeel Murshed and Adnan Amin. Dec. 2000, pp. 99–111.

[Kin+11] Henrik Kinnemann, Andreas Stoffel, Daniel Keim, and David Spretke. “Verfahren und Vorrichtung zum Erkennen und Klassifizieren von Dokumentteilen eines rechnerverfügbaren Dokuments durch schrittweises Lernen aus mehreren Trainingsmengen”. Patent DE102009050681. Dec. 5, 2011.

[KLT01] Jongwoo Kim, Daniel X. Le, and George R. Thoma. “Automated labeling in document images”. In: *Document Recognition and Retrieval VIII*. Ed. by Paul B. Kantor, Daniel P. Lopresti, and Jiangying Zhou. Vol. 4307. Spie Proceedings Series. SPIE, 2001, pp. 111–122.

[KM12] Dan Klein and Christopher D. Manning. *The Stanford Parser*. 2012. URL: `http://nlp.stanford.edu/software/lex-parser.shtml`.

[Kot07] Sotiris B. Kotsiantis. “Supervised Machine Learning: A Review of Classification Techniques”. In: *Informatica (Slovenia)* 31.3 (2007), pp. 249–268.

[LB05] Heidi Lam and Patrick Baudisch. “Summary Thumbnails: Readable Overviews for Small Screen Web Browsers”. In: *Proceedings of the 2005 Conference on Human Factors in Computing Systems*. Ed. by Gerrit C. van der Veer and Carolyn Gale. ACM, 2005, pp. 681–690.

[LHZ03] Charles X. Ling, Jin Huang, and Harry Zhang. “AUC: A Better Measure than Accuracy in Comparing Learning Algorithms”. In: *Advances in Artificial Intelligence, 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI 2003)*. Ed. by Yang Xiang and Brahim Chaib-draa. Vol. 2671. Lecture Notes in Computer Science. Springer, 2003, pp. 329–341.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting

and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*. Ed. by Carla E. Brodley and Andrea Pohoreckyj Danyluk. Morgan Kaufmann, 2001, pp. 282–289.

[McC02] Andrew Kachites McCallum. "MALLET: A Machine Learning for Language Toolkit". http://mallet.cs.umass.edu. 2002.

[Med13] Bethesda (MD): National Library of Medicine (US). *PMC Open Access Subset*. 2013. URL: `http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/` (visited on 11/19/2013).

[MMS10] Simone Marinai, Emanuele Marino, and Giovanni Soda. "Table of Contents Recognition for Converting PDF Documents in E-book Formats". In: *ACM Symposium on Document Engineering*. Ed. by Apostolos Antonacopoulos, Michael J. Gormish, and Rolf Ingold. ACM, 2010, pp. 73–76.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008, pp. I–XXI, 1–482.

[Nag00] George Nagy. "Twenty Years of Document Image Analysis in PAMI". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000), pp. 38–62.

[NJ07] Anoop Namboodiri and Anil Jain. "Document Structure and Layout Analysis". In: *Digital Document Processing*. Ed. by Bidyut B. Chaudhuri. Advances in Pattern Recognition. Springer London, 2007, pages.

[NNS04] Koji Nakagawa, Akihiro Nomura, and Masakazu Suzuki. "Extraction of Logical Structure from Articles in Mathematics". In: *Mathematical Knowledge Management, Third International Conference*. Ed. by Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec. Vol. 3119. Lecture Notes in Computer Science. Springer, 2004, pp. 276–289.

[Oel+12] Daniela Oelke, David Spretke, Andreas Stoffel, and Daniel A. Keim. “Visual Readability Analysis: How to Make Your Writings Easier to Read”. In: *IEEE Trans. Vis. Comput. Graph.* 18.5 (2012), pp. 662–674.

[RB06] Yves Rangoni and Abdel Belaïd. “Document Logical Structure Analysis Based on Perceptive Cycles”. In: *Document Analysis Systems VII, 7th International Workshop, DAS 2006*. Ed. by Horst Bunke and A. Lawrence Spitz. 2006, pp. 117–128.

[RHI03] Maurizio Rigamonti, Oliver Hitz, and Rolf Ingold. “A Framework for Cooperative and Interactive Analysis of Technical Documents”. In: *GREC03: Fifth IAPR International Workshop on Graphics Recognition*. 2003, pp. 407–417.

[RNM07] Sylvie Ratté, Wilfried Njomgue, and Pierre-André Ménard. “Highlighting Document’s Structure”. In: *International Journal of Electrical and Electronics Engineering* 1.6 (2007), pp. 367–371.

[SB88] Gerard Salton and Chris Buckley. “Term-Weighting Approaches in Automatic Text Retrieval”. In: *Information Processing & Management* 24.5 (1988), pp. 513–523.

[SC12] United States Securities and Exchange Commission. *FORM 10-K: Annual Report Pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934*. 2012.

[Set05] Burr Settles. “ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text”. In: *Bioinformatics* 21.14 (2005), pp. 3191–3192.

[SLS09] Eric Saund, Jing Lin, and Prateek Sarkar. “PixLabeler: User Interface for Pixel-Level Labeling of Elements in Document Images”. In: *10th International Conference on Document Analysis and Recognition (ICDAR 2009)*. IEEE Computer Society, 2009, pp. 646–650.

[SM95] Margaret-Anne D. Storey and Hausi A. Müller. “Graph Layout Adjustment Strategies”. In: *Graph Drawing*. Ed. by Franz-Josef Bran-

denburg. Vol. 1027. Lecture Notes in Computer Science. Springer, 1995, pp. 487–499.

[Sta08] International Organization for Standardization. *Document Management — Portable Document Format — Part 1: PDF 1.7*. ISO 32000-1:2008. July 2008.

[STK95] Shin'ichi Satoh, Atsuhiro Takasu, and Eishi Katsura. "An Automated Generation of an Electronic Library based on Document Image Understanding". In: *Third International Conference on Document Analysis and Recognition (ICDAR 1995)*. IEEE Computer Society, 1995, pages.

[Sto+10] Andreas Stoffel, David Spretke, Henrik Kinnemann, and Daniel A. Keim. "Enhancing Document Structure Analysis using Visual Analytics". In: *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC2010)*. Ed. by Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung. ACM, 2010, pp. 8–12.

[Sto+12] Andreas Stoffel, Hendrik Strobelt, Oliver Deussen, and Daniel A. Keim. "Document Thumbnails with Variable Text Scaling". In: *Comput. Graph. Forum* 31.3 (2012), pp. 1165–1173.

[Str+09] Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel, Daniel A. Keim, and Oliver Deussen. "Document Cards: A Top Trumps Visualization for Documents". In: *IEEE Trans. Vis. Comput. Graph.* 15.6 (2009), pp. 1145–1152.

[Suh+02] Bongwon Suh, Allison Woodruff, Ruth Rosenholtz, and Alyssa Glass. "Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces". In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. ACM, 2002, pp. 251–258.

[Sum98] Kristen Maria Summers. "Automatic Discovery of Logical Document Structure". PhD thesis. Cornell University, 1998.

[Tsu+01] Shinji Tsuruoka, Chihiro Hirano, Tomohiro Yoshikawa, and Tsuyoshi Shinogi. “Image-based Structure Analysis for a Table of Contents and Conversion to XML Documents”. In: *Proceedings of Document Layout Interpretation and its Application, DLIA01*. 2001, pp. 59–62.

[WEK12] B. Webber, M. Egg, and Valia Kordoni. “Discourse Structure and Language Technology”. In: *Natural Language Engineering* 18.4 (2012), pp. 437–490.

[Woo+02] Allison Woodruff, Ruth Rosenholtz, Julie Bauer Morrison, Andrew Faulring, and Peter Pirolli. “A Comparison of the Use of Text Summaries, Plain Thumbnails, and Enhanced Thumbnails for Web Search Tasks”. In: *Journal of the American Society for Information Science and Technology* 53.2 (2002), pp. 172–185.

[YSS05] Sherif M. Yacoub, Vinay Saxena, and Sayeed Nusrulla Sami. “PerfectDoc: A Ground Truthing Environment for Complex Documents”. In: *Eighth International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2005, pp. 452–457.

[Zwa+11] Matthew van der Zwan, Wouter Lueks, Henk Bekker, and Tobias Isenberg. “Illustrative Molecular Visualization with Continuous Abstraction”. In: *Computer Graphics Forum* 30.3 (2011), pp. 683–690.