Saarland University

Faculty of natural Sciences and Technology I

Department of computer Science

Semantic Model Extraction from Semi-Structured Textual Resources

Master`s Thesis in Computer Science

by

ÖzgünErensoy

supervised by

Prof. Dr. JörgSiekmann

advised by

Dr. Sergey Sosnovsky

reviewers

Prof. Dr. JörgSiekmann

Dr. Sergey Sosnovsky

October 2015

EidesstattlicheErklärung

IcherklärehiermitanEidesStatt, dassich die vorliegendeArbeitselbständigverfasst und keineanderenals die angegebenenQuellen und Hilfsmittelverwendethabe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis

Einverständniserklärung

Ich bin damiteinverstanden, dassmeine (bestandene) Arbeit in beidenVersionen in die Bibliothek der Informatikaufgenommen und damitveröffentlichtwird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,_____ _____
(Datum/Date) (Unterschrift/Signature)

Acknowledgements

I would like thank my advisor Dr. Sergey Sosnovsky for all his patience and support. Without his expertise and persistence it would have been a challenge to see theend of this study.Also I would like to extend my gratitude to my colleagues from CeLTech/DFKI, Saarland for making this last year a really fun ride. Additionally I thank the Max Planck Institute of the Computer Science family for all the funding and organizational support they provided during the last 3 years.

Last, but not least, I would like express my immense gratitude for my family for their never ending energy and motivation, and my girlfriend whom actually kept my sanity intact no matter how close I came to the edge.

Table of Contents

1 Introduction

The amount of digitally available documents is increasing rapidly. The number and variety of these documents make it challenging to process them manually. Hence, technologies for automated processing of digital documents, extraction of meaningful information from them and reasoning over them become more and more important.

Since a fair amount of the digital information is textual, automated processing of digital textual documents attracts a lot of attention. It is a large and active area of research with a variety of topics, including image processing [24], text extraction [23], document layout analysis [12], document structure extraction[12], knowledge and data extraction [25] etc. Often, to achieve the maximum accuracy, the technologies developed in these areas have to focus on one type of format (e.g. HTML, PDF, plain text, etc.) and/or one type of documents (e.g. news items, academic articles, textbooks, etc.).

This thesis focuses on textbooks as the target type of documents. Textbooks are very particular textual resources. They are written by experts in the respective domain, for novices in the domain, with the purpose to explain them the domain knowledge. A typical textbook author uses various formatting elements to facilitate understanding of its content by readers. Through the usage of formatting elements, authors structure textbooks in a way they themselves see the respective domain. Hence, when extracted, the structure of a textbook represents the knowledge map of the domain as perceived by its author.

Generally, the procedure required to extract the structure depends on the digital format of the textbook. Most of the formats used for textbook publishing are ignorant of the internal structure provided by the author. This is the reason textbooks are referred as "semi-structured documents" in this work, even though they do contain a hidden layer of structure manifested in their formatting. What makes the task even more challenging; the target format for the approach developed in this thesis is PDF. It is one of the most popular representation formats for electronic texts (incusing textbooks), Unfortunately, when it comes to explicit representation of text format or document structure, PDF provides publishers with a very limited toolset, and even this toolset remains largelyunderused.

1.1 Motivation

When a human reads a piece of a formatted text, a certain structure of the information conveyed in the text becomes immediately apparent, regardless of the domain, content, and, in some cases, even the language. We can utilize the format to start building an information/knowledge map and/or a plan in our mind, then fill the corresponding entries in that map as the text is read. As a matter of fact, according to a study conducted by Schmidet al., when humans read a piece of non-formatted, non-structured text, creating a mind map of the information obtained from that text takes much more effort than it would take with the formatted and structured version of it [3].

For example, a reader can determine that the text onthe Figure 1.1 is a table of contents, even if the reader does not speak French. It is nothing, but a simple task to identify "*1 Introduction*" as a chapter, "*1.1 Modélisation des phénomènesaléatoires*" as its sub-chapter and "*1.1.1 Univers*" as a sub-sub-chapter. The recognition of the roles of these phrases and theirordering into a hierarchy happen without the reader focusing on these tasks and putting any effort into them.

During recognition of the structure, a French speaking reader would understand the meaning behind the topics in parallel. For example seeing the "*1.1 Modélisation des phénomènesaléatoires*" and "*1.1.1 Univers*" lets the reader understand that the domain in question is statistics. This realization speeds up the hierarchical ordering of the topics, because the comprehension of the titles helps the reader to use his/her already existing knowledge about the titles and the domain.While a non-speaker can only observe a hierarchical order based on formatting, the French speaker utilizes the formatting to create an actual mind map of the domain.

Building a set of heuristics that would allow simulating the human reader ability to recognize the structure of a text and infer from it the structure of the domain is the main motivation for this work. Once the structure of a textbook is extracted and represented formally, it would enable semantic retrieval of information from the book, more detailed identification of roles various chunks of text play in the content of the book(e.g. identification of learning objects and their types), linking of textbook parts to other

information sources, reordering the textbook, hiding its parts and/or adding new ones. At the end various forms of personalized access and support for the to textbook material would become possible as well.

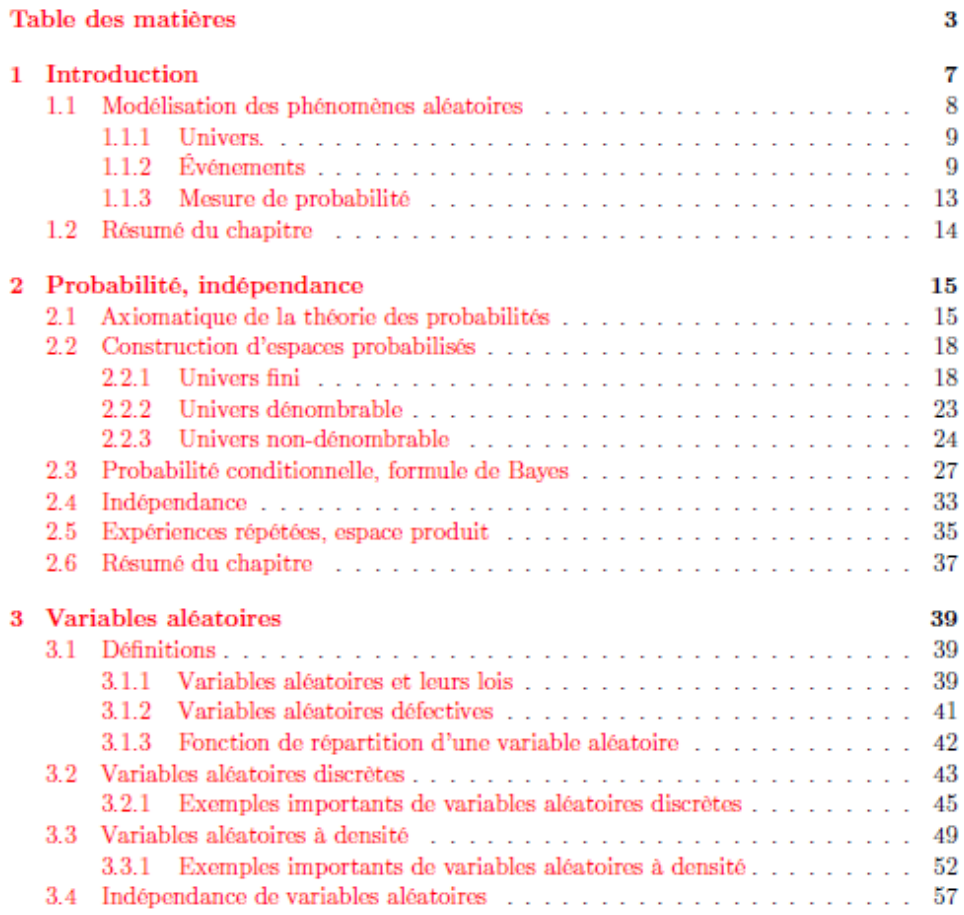


Figure 1.1 An Example of Table of Contents

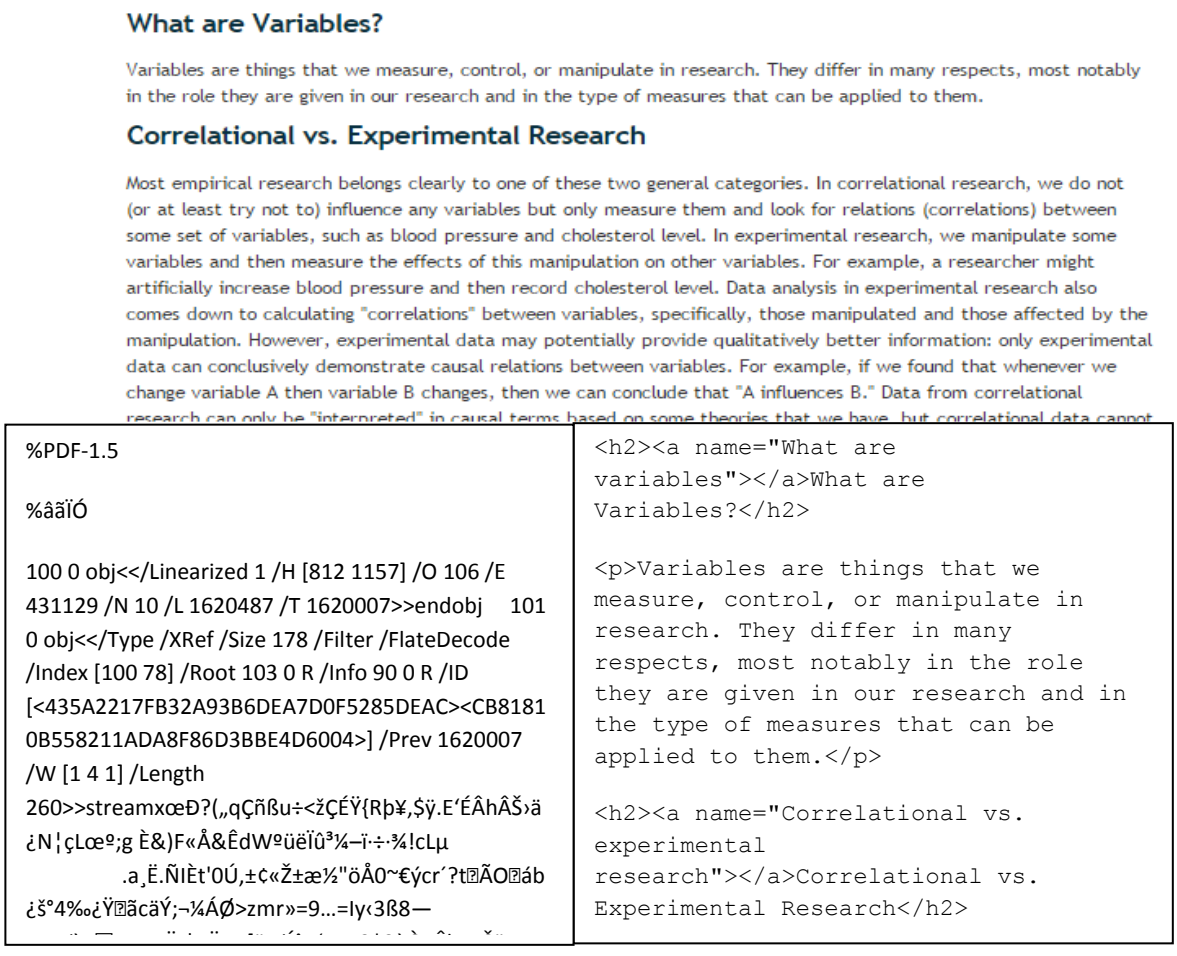
1.2 Problem Overview

PDF is one of the dominant publishing formatsfor digital textbooks, which is the main practical reason for this work to focus on PDF as the target document type. From the research perspective, PDF is a very challenging format to parse; therefore making sure that the conceptual approach can be implementedwith PDF would secure its applicability to less challenging (more structure-friendly) formats as well.

PDF is a format type that largely ignores its contents` internal structure. When a parser processes a PDF document, it would not get any structural tags like "<title>" (Figure 1.2), but just a byte stream of characters with their formatting information attached (e.g.

font size, type face etc.).Absence of structural representation of content makes it hard to identify the structural components (e.g. titles, sub-titles, headers, paragraphs, tables, lists etc.) and detect the relations between these components. One should also keep in mind that there are many different ways to structure, format and write a textbook. Such a variety makes it even more challenging to define a fine-grained set of rules to create relations between components of textbooks.

Even though PDF is an unstructured format by default, it provides a feature called 'tagging' to make up for it. Unfortunately, due to the amount of the required work to apply this feature, it is rarely used. Hence, it is not a reliable source of information to extract structure from PDF textbooks.



What are Variables?

Variables are things that we measure, control, or manipulate in research. They differ in many respects, most notably in the role they are given in our research and in the type of measures that can be applied to them.

Correlational vs. Experimental Research

Most empirical research belongs clearly to one of these two general categories. In correlational research, we do not (or at least try not to) influence any variables but only measure them and look for relations (correlations) between some set of variables, such as blood pressure and cholesterol level. In experimental research, we manipulate some variables and then measure the effects of this manipulation on other variables. For example, a researcher might artificially increase blood pressure and then record cholesterol level. Data analysis in experimental research also comes down to calculating "correlations" between variables, specifically, those manipulated and those affected by the manipulation. However, experimental data may potentially provide qualitatively better information: only experimental data can conclusively demonstrate causal relations between variables. For example, if we found that whenever we change variable A then variable B changes, then we can conclude that "A influences B." Data from correlational research can only be "interpreted" in causal terms based on some theories that we have, but correlational data cannot

%PDF-1.5

%âãÏÓ

100 0 obj<</Linearized 1 /H [812 1157] /O 106 /E 431129 /N 10 /L 1620487 /T 1620007>>endobj 101 0 obj<</Type /XRef /Size 178 /Filter /FlateDecode /Index [100 78] /Root 103 0 R /Info 90 0 R /ID [<435A2217FB32A93B6DEA7D0F5285DEAC><CB81810B558211ADA8F86D3BBE4D6004>] /Prev 1620007 /W [1 4 1] /Length 260>>streamxœÐ?(„qÇñßu÷<žÇÉŸ{Rþ¥,$ÿ.E'ÉÂhÂŠ›ä¿N¦çLœº;g È&)F«Å&ÊdWºüëÏû³¼–ï·÷·¾!cLµ .a¸Ë.ÑIÈt'0Ú,±¢«Ž±æ½"öÅ0~€ýcr´?t ÃO áb¿š°4‰¿Ÿ ãcäÝ;¬¼ÁØ>zmr»=9…=Iy‹3ß8—


```
<h2><a name="What are
variables"></a>What are
Variables?</h2>

<p>Variables are things that we
measure, control, or manipulate in
research. They differ in many
respects, most notably in the role
they are given in our research and in
the type of measures that can be
applied to them.</p>

<h2><a name="Correlational vs.
experimental
research"></a>Correlational vs.
Experimental Research</h2>
```

Figure 1.2 - PDF (left) vs. html (right) comparison

Another important challenge originates from the freedom that PDF provides during document creation. When creating a PDF document, it is possible to provide the minimum amount of required formatting information, instead of all the supported properties (font-

size, bleed box, crop box, expected space size etc.). Often, the publishers and authors follow the minimum requirements to create a PDF. This limits the amount of information that can be used to identify the structural components.

For high coverage and accuracy, the approach developed in this thesishad to recognize structural components based on their visual representation with minimum formatting requirements. The set of rules to link those components should be generic enough to be applicable to a large variety of textbooks in a broad range of domains.

1.3 Approach Summary

The first phase of the developed procedure is to identify the elements of the overall textbook layout,such as titles, sub-titles, paragraphs, table of contents etc., by applying heuristics over every page of a textbook. After identification of these elements, system creates relations between them according to the rules derived during the first phase, and their respective order of occurrence throughout the textbook. Lastly, the semantic model of the resource is created, and enriched with the information present in the textbook. The semantic model is connected to the central repository, so that it can be linked with the models of similar textbooks existing in the repository. The intelligent educational system INTERLINGUA (Chapter 3) uses the repository as its knowledge storage.

To perform the enrichment, the index of a textbook is required to be processed. While extraction of table of contents and citations from different document types has attracted an amount of researches, index extraction is neglected as logical element.Since both the table of contents and the index has their own specific set of rules to exists in a textbook, a variation of generic table of contents extraction method was used to obtain the index.

An important contribution of this work is the proposed unified approach that implements all stages of the textbook model extraction and uses all possible formatting elements introduced by the textbook authors. As can be seen in Chapter 2, related works usually focus on the detection and/or extraction of specific elements or the layouts. The system proposed in this thesis identifies the structure of textbooks, and converts it into a machine-readable format, recognizes the physical elements and determines their logical

meaning. Additionally, unlike any other, it enriches the structure by employing the domain related information residing within the logical elements. It provides means of connecting the processed textbooks to external resources by making them aware of their content`s structure and the domain organization.

1.4 The Structure of this Thesis

The rest of this thesis is organized as the following. Chapter 2 covers the background knowledge and prior work related to document layout mining and structure extraction with a particular focus on PDF as the source format. The third chapter explains the larger research context of this work –the INTERLINGUA project.The fourth chapter discusses the theoretical premise of this thesis and presents the conceptual details of the developed approach. The fifth chapter describes the implementation of the approach in details. Finally, the sixth chapter discusses the results, outlines some directions for future work and concludes the thesis.

2 Related Work

This section provides an overview of seminal research onthe layout analysis and structure extraction from textual resources with a special focus on PDF-formatted documents.

Much research in this area focuses on digital formats with easily interpretable structure, such as HTML. Arasu et al. [15] studied a way to extract the structured data from web pages without any training data or human input by defining a formal template. They tested the developed approach with the pages from amazon.com and achieved good results:they make it possible toss sophisticated queries over the contents of such websites. Crescenzi et al. [16] proposed an approach to extract data from HTML pages with wrappers. Unlike Arasu et al., they do not focus on only structured websites, but any kind of available website. To make their wrappers maintainable, they automate the wrapper generation in a way that it will not rely on the target page and the content.

There is another group of research focusing on the structure extraction from unstructured digital formats such as images or PDF. Unlike the XML-based formats, which provide some means of machine readable structural representation, these formats are oblivious to their contents` structure. The absence of structural representation, the freedom of layout and the variety provided by these formats make the structure extraction more challenging. The layout, the purpose and the domain of the documents become important criteria to extract the structure. However, even when two documents have the same purpose – job announcements, for instance, their layouts can be very different from each other. This caused researches projects in the field to narrow down their scopes and address simplified tasks.

PDF is another popular format for publishing and sharing textual resources, thus, it attracts a fair amount of research interest as well. However, even before the PDF became popular, there has been a considerable amount of work regarding layout and structure analysis of the unstructured documents. They employed image processing (geometrical) approaches as the basis. Geometrical approaches extract information based on the physical properties of the document. The details of physical elements will be covered later on in

Section 2.1. A well-known early work based on geometrical approaches is WISDOM++ [20]. This is a language-independent text block identification and classification program. It accepts the documents in image format as input. Its process is divided into 4 main phases; document analysis, document classification, document understanding, and text recognition. The underlying idea of the approach is to use a rule base, which is obtained through trained data, to perform the tasks through four phases. For further reading on early progress of the information extraction techniques, a good set of references is available in [21].

Geometrical approaches were mostly limited to extracting the vague layout and structure of a document. While they are a good way to obtain the basic structure, they usually fall short to extract more precise information. To achieve more detailed structural representations of the documents, the inherent structure of the content become an important element. Hence, interest started to shift towards extraction of logical elements (logical approach) (e.g. table of contents, footer/header, sub-titles/titles etc.) from documents. Anjewierden [13], proposed an approach which incrementally extracts logical structures named AIDAS. Another work was proposed by D´ejean et al. [14] to convert PDF documents to XML format. First, they extract the heterogeneous streams in PDF and convert them to XML. Their process mostly focuses on traditional image processing methods. However, some of the steps they used were specific to PDF properties. Lin et al. [15] proposed a method, where they apply headline matching and layout modeling over the table of contents of the PDF book. He et al. [22] extracted the hierarchical logical structure of PDF book documents, by combining the spatial and semantic information obtained from the table of contents of the book.

Other than the above mentioned papers, it is possible to find a more detailed survey of general digital document structure analysis in [12]. Additionally, in [11] the methods for extracting low-level structural data from PDF documents was surveyed.

Due to their significance for this thesis, the following three papers [1], [4], [10] are going to be covered in depth.

2.1 Layout Analysis

Klink et al. [1] proposed a way to analyze the structure of a document by combining the geometrical and logical approaches. Currently this work is considered as one of the standard header/footer detection methods. The approach divides the document into two parts; physical and logical.

Physical part consists of elements that physically exist on a document (letters, pages, lines etc.). They define words as the lowest level physical element. Words form lines, lines form blocks and blocks form pages. It lacks detail, but its simplicity makes it possible to apply layout to almost all kinds of textual documents.

The logical part is concerned with what those aggregated words, lines and blocks stand for. One or more blocks can create a logical unit. Logical units are the components such as titles, headers, footers, tables, paragraphs, lists, abstracts, authors, headings etc. The types and the amount of the logical units can change depending on the document. The general layout that they depict can be seen in Figure 2.1.

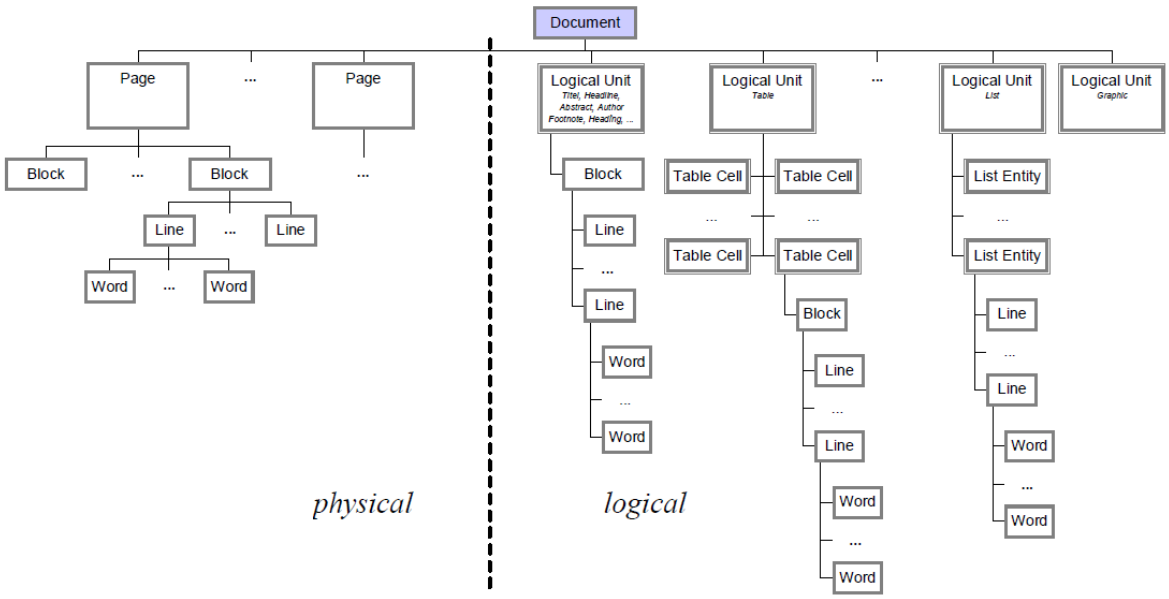


Figure 2.1 Document Layout

They identify the generic logical elements like header and footer in a crude yet simple manner, and this is independent from the rest of the domain/document specific element identification. They search for a horizontal line with a small top margin for header, and a small bottom margin for the footer.

As for the domain/document dependent logical elements, they introduce a flexible rule based testing. For every possible label, there exists only one rule defined. The system goes through the physical blocks and tests them individually for all labels. Based on the rules that fulfill, labels are assigned to the blocks. A block can have more than one label assigned to it, with different ratios. Ratios are determined with the amount of the rule units fulfilled for a label and the importance (weight) of every fulfilled rule unit. A rule unit is a logical expression consisting of attributes. It can be said that an attribute is a simple question. For example, "are all the words bold?" or "Is the block inside a special region on a page?". The attributes are also divided into two; self-related and cross-related. The self-related attributes are concerned solely with individual block`s properties. These properties can be the positioning of the block in the page, font-face/font-family/font-size of the block, number and alignment of the lines in the block. On the other hand, as the name implies, cross-related properties consists of the relations with other blocks. These inter-block relations are :

- **Geometrical**: This inspects the positioning of a block with respect to another block with specific labels or attributes. It follows the simple mind set of: "If there is a block A, then there should be a block B underneath.". Since "underneath" is a vague term in terms of distance and can apply to any block positioned below the currently observed object, they extend it by adding a distance constraint. A simple example for geometrical relation is; if there is a block labeled as table, the block underneath with distance of more than 1, and less than 4 times the line height of the caption, should be a caption.
- **Textual**: The absolute or relative number of common words between two blocks or a given predefined list of words that needs to occur in the related block are checked. The example for such rule is defined by them as: "There must be a block which has at least one word in common with the block for which the rule might be applied and furthermore it needs to contain the words 'Dear' and 'Mr.' ". This rule can express the relation between the recipient field of a letter and the salutation field.
- **Label**: Here, the existence of a logical object triggers labeling of another object or objects. This relation type is extremely useful to increase the precision of logical objects which are usually found together with other logical objects. Author or abstract in a scientific book is one of those logical objects. When the title is found

and labeled, that means the author or the abstract block should be close by. If there is no title found, then probably there will not be an author block either.

A block does not need to satisfy a rule with 100% precision to get the label assigned to it. It is possible for a block to satisfy more than one rule with partial results. The matching function of an attribute returns 1, when a block satisfies the requirements of an attribute within the expected value range. If a block satisfies an attribute only partially, then the percentage is returned as result. The percentage is calculated by how much of the requirements were met.

The ratio of a label is determined by combining the matching values of self-related and cross-related attributes of the rule with a weighted function. Weight of every attribute ranges between 0 to 1. Weight of every attribute can be defined individually within the rule.

More than anything, this work combined many different approaches to achieve a better solution for the structure/layout analysis. It can be said that the document model they defined in Figure 2.1 is now accepted and heavily relied on by other works, which includes this thesis. In this thesis the document model they introduced is employed during PDF parsing, and blocks are formed accordingly on the physical layer.

2.2 Structure Extraction

Gao et al. [4], introduced a method to extract the structure from PDF-based books. In this method, they used combination of both geometrical and logical approaches.

The core of this approach is built upon the idea pointed out by Bart et al [8] namely, "the repetition of physical structure is prevalent in documents. Such repetition conveys the underlying logical structure of the data in document design.".

The common type-setting practices are the manner that this kind of repetition manifests itself within books. The common type-setting is achieved by using the same formatting properties for the page elements belonging to the same functional component. This means, the titles of the same level would share a regular formatting throughout the book. This format consistency among the components is named as: "Style Consistency of page components" or "SCC" in short by them.

The general architecture of the approach can be seen in Figure 2.2 below. The process is separated into two phases; physical structure extraction and logical structure extraction.

The physical structure extraction is rather simple when compared to the logical structure extraction. The procedure follows a bottom-up approach during the block creation process. They form words from characters, lines from words etc. during page layout analysis. The global typography detection is the point where the properties consistent throughout the pages of a textbook (e.g. formatting properties for titles of different level, the line spacing of body text, the text body area of pages, the header/footer) are handled.

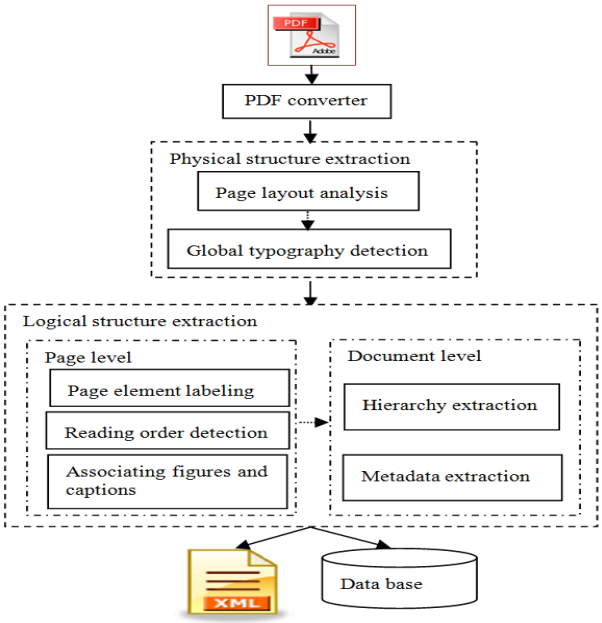


Figure 2.2 Architecture Layout

The logical structure extraction is divided into two main sections. The page level extraction focuses on labeling the blocks which are formed during the physical extraction phase. The document level extraction focuses on forming the hierarchical structure according to the results of the page level extraction.

Page level extraction uses a rule based approach to label the blocks. After the blocks of a book are formed, they are grouped together according to their SCC via clustering. In the resulting set of clusters, each one stands for a different logical component type.

Then, the system extracts 23 different features from every block depending on the rules defined for each feature. The features are categorized as block level (local attributes of

a block); number of lines in the block, font size of the block etc., page level (relation between blocks); distance between blocks in a page, adjacency of a block to an image, and document level (local attributes of a cluster); number of blocks in a cluster. After the features are determined for the blocks, a Support Vector Machine is employed to label the blocks conforming to their features.

The most prominent point of this work is, the capability to handle the reading order of the blocks. Some books follow a heterogeneous layout. In some pages the number of the columns may vary, or in some pages blocks may form a L-shape (Figure 2.3). The reading order of Figure 2.3 should be A-B-C-D-E-F. To achieve the correct order they combine different methods to create a weighted bipartite graph of blocks. Each edge in between blocks represents the possibility that the latter node is the following node in the reading order. After the bipartite graph is created, the optimal matching algorithm is applied to resolve the graph into the most possible reading order.

The document level extraction creates a hierarchical structure from the table of contents of the book. The table of contents extraction performed according to the work proposed in [14]. After determining the table of content entries, the chapter/section titles corresponding to those entries are assigned with the same hierarchical order.

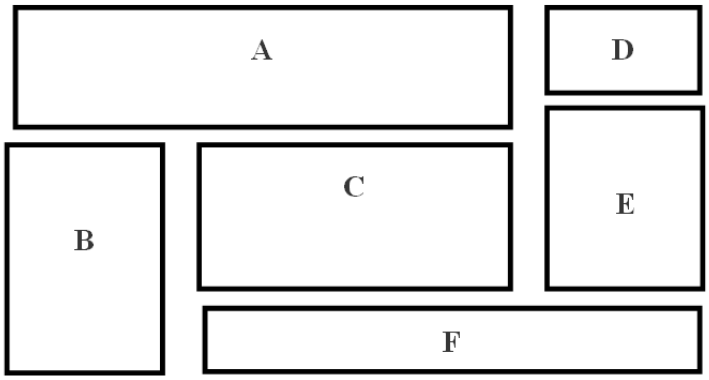


Figure 2.3 L-shaped Page Layout

2.3 Table of Contents Extraction

Wu et al. [10] introduced a generic approach to identify and extract the table of contents (TOC) from PDF book documents. The styles of TOC tend to be different from one resource to another. Hence, it is challenging to come up with a method to fit for all possible

layouts. However, after extensive amount of observations, they claim that any TOC can be categorized as one of the following; "flat", "ordered" or "divided". They observed that, whatever the actual layout may be, a table of contents falls under one of these generic categories.

Their approach states that the effective automation of TOC extraction has three sub-tasks to be addressed: detection, parsing and linking of table of contents. The general architecture of this approach can be seen in Figure 2.4 below.

The TOC detection identifies the start and the end point of the table of contents. Finding the beginning point is an easier task than finding the end. By experience, they determined a TOC shows up in the first 20 pages and the start of the TOC is always distinctively indicated with a title containing words such as "Contents" or "Table of Contents" etc. The end point proves a little bit harder to detect. They check if the page numbers at the end of each TOC entry are legal. If a line with a page number 'p' does not have any legitimate page numbers within the 5 following lines, then that line with page number 'p' is accepted as the last entry of TOC.

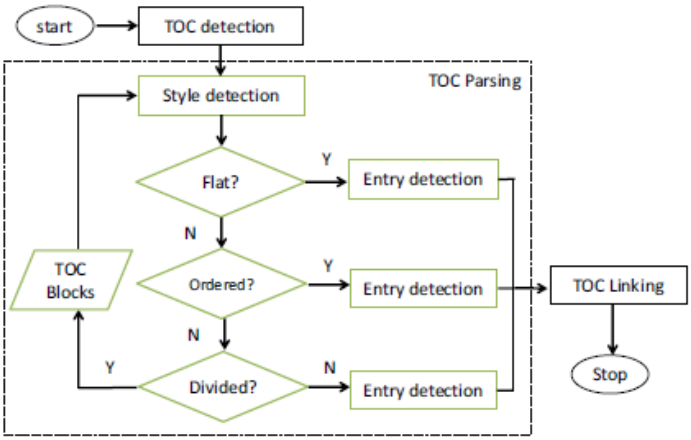


Figure 2.4 TOC Detection Approach Architecture

The TOC parsing, basically parses the detected table of contents and determines which of the above mentioned categories would be a proper fit for it. A TOC is flat when, all entries of TOC is the same. They share the same visual formatting, the same start X coordinates and the same line spacing in between the entries. It is ordered when, the entries

start with ordered section numbering. It is divided when, the entries can be grouped based on the differences between line spacing of entries.

Classifying the TOC as one of these generic categories makes it easier to extract the structure of the table of contents. This resolves the problem of having a specific extraction algorithm for every possible TOC layout, and sizes it down to 3, one for each category.

- When a table of contents is classified as flat, that means every entry is on the same level. Hence, the structure will have a flat hierarchy.
- When it is classified as ordered, following the order of the section numbers is sufficient to extract the structure and the hierarchical relation between entries.
- When it is classified as divided, each group corresponds to the same level in the hierarchy, and the entries in a group are hierarchically reside under the groups` first entry accordingly.

The TOC linking is the process of finding the corresponding titles in their respective pages and creating the link between the table of contents entries.

This thesis is not an extension of one approach, but a combination of different approaches with different purposes to increase the quality and usability of the extracted structure.

3 Context Of This Work :INTERLINGUA

INTERLINGUA[*] is a multi-lingual intelligent educational system developed in the frameworkof a project "*Interlingua: supporting students of Greater Region with interlingual educational resources*"funded by INTERREG-IVA-GR programme[†]. Its main purpose is to createsupport students who have to study in a foreign languages by providing them with links between the target study material and relevant reading resources in their mother tongue.To clarify the purpose, and to make it more tangible, the following use case can be considered: A French student decides to study at a university in Germany. He/she may know German enough to take care of most of the day-to-day operations, but lacks the required specific vocabulary to comprehend related topics in the domain of a course. Considering that the study materials(textbooks, slides etc.) are in German language, it would prove hard for this student to find the proper translations and definitions of the German terms in French. If nothing else, it would be a tedious and time-consuming process. This is the point where INTERLINGUAtries to help. Through INTERLINGUA, a student reading a chapter from a German textbookcan ask for a related part of a French textbook. The supported languages are English, German, and French. At the moment, it supports only one domain: "Probability Theory and Mathematical Statistics".

The basic architecture of INTERLINGUA consists of the four main components: the textbook model extractioncomponent, the reference ontology, the linking component and the central repository. This thesis describes the functionality of the textbook model extractioncomponentof INTERLINGUA.To have a better understanding of theentire environment and the role the textbook model extractioncomponent plays in it, INTERLINGUA the rest of this chapter provides a brief description ofINTERLINGUA.

The reference ontology is the central modelof the domain of statistics. This ontology is required to link all the key terms between the textbooks during textbook model extraction.In addition to that, it provides the capabilities to translate a term between languages, and provide a definition for a term from DBpedia. Every domain includedin the

[*] www.**INTERLINGUA**-**project**.eu
[†] www.**interreg**-**4agr**.eu

system needs to have onemodel.The current system has only reference-ontology for the statistics domain and it contains has more than 3500 concepts. It is obtained from ISI glossary, and the entries are enriched with DBpedia links.

The textbook model extractionconsists of the system proposed in this thesis.Identification/extraction of the table of contents and the index, as well as the linking between the index terms and the reference-ontology mentioned above take place here.As an additional feature, it uses thelinks created between the index terms and the reference-ontology to enrich the PDF textbooks by highlighting them on the pages. These highlights contain a pop-up menu. It provides the following functionalities: requesting the translation of the term, requesting the DBpedia definitions of the term, and requesting the assessment questions related to the term`s chapter.When it is processes a resource, it transfers the extracted models to the central repository to store them.

The linking componentis executed by thecentral repository to link a newly added textbook with the already existing textbooks. This component processes the textual contents to find the similarities betweenthetextbooks. When two chapters of the two different textbooks show similarities, they are linked to each other. To achieve accurate results it uses training data, namely expert mapping. Expert mapping consists of a manually created link tables between the chapter and section titles of textbooks. This task is performed by multi-lingual experts of the respective domain.

The central repository is the back-bone platform that keeps all the components connected. It handles all the storing and delivery actions. Splitting the textbooks into smaller segments based on the textbook model, filling the related tables for the textbooks, keeping the communication in between other components, accepting queries from the client programs and replying to them are some of its major functionalities. Additionally, it handles the popup-menu functionalities mentioned in the textbook model extraction component. As long as the central repository`s pre-defined queries are used, any kind of client can connect to the repository and make use of the provided the functionalities of the system.

4 Foundation Concepts

4.1 Textbooks

Every textbook reflects its author`s perception of therespective domain`s model.To transfer their understanding of the domain, the authors often use differentiating line spacing sizes, font-family, font-sizes, indentation etc., andcreateahierarchical order among the blocks of information. They lead the readers with this inherent structure to facilitate the comprehension of the conveyed piece of information.Moreover, the readers expect to get an insight of the domain and learn the relevant material. This is why , it is possible to obtain the general structure of a domainby extracting the structure of a textbook.

The depth, detail and the style of a textbook`s structure can change, based on the author`s style and the expertise of the target audience. For example, a biology textbook for grade school students would contain more graphical depictions, pictures with simple explanatory information, which would lead to a coarse grained domain representation.On the other hand, a biology textbook for university students would contain less graphical depictions, but more text with explanations. The structure of such textbook would represent the domain in much more detail.

It is also possible that different authors structure the same domain differently, since there is not only one right way to model a domain. This explains the existence of textbooks with different layouts even on the same topic, let alone the same domain. However, as there are fundamental guidelines, rules and notions to teach, there are some common notions, guides and rules to write a textbook. These are simple notions such as titles should have bigger font sizes than the paragraphs, page numbers should be located either on the top or the bottom of a page, elements of the same logical purpose and level should have the same style. A more detailed explanation about design issues of a textbook can be found in [2].

The main assumption of this work is thatevery textbook has an inherent structure provided by its author. Even though differenttextbooks have different typographic preferences to reflect its structure, the typography of a textbook will be consistent through

itself. This means that a logical element on a page will share the same typographic characteristics with the other logical elements who share the same classification on other pages. In other words, all titles of the same hierarchical-level, or all the table/figure captions will have the same visual attributes. The logical element extraction (section 5.3) and the text-block forming (section 5.2)are built upon this idea.

4.2 Human reader perception

According to Schmid et al.[3], the inherent structure emerging from formatting was mostly neglected in most of the researches focusing on theelaboration of cognitive representation of the text content in late 90`s. However, later on it was realized that it is not possible to disregard the formatting of a linguistic content during an automated reading process. Hence, they experimented over the effects of the formatting to human cognition when reading a text. The experiments stated that formatting indeed does affect the human cognition. Unlike a plain text, without any kind of formatting and grouping, the formatted text is read slower by the reader. On the other hand, the time spent at the end of the reading action to create a logical map of the informationis shorter.Moreover the reader can remember more of the formatted version.

As can be seen on Figure 4.1, for a human reader it is nothing but a trivial process to identify the title, sub-title and body of the text regardless of its language. How does the human brain achieves such feat in such short amount of time is beside the extent of this work. However, the steps of this procedure is essential to simulate similar results on a computer.

When a human reader looks at a page of a textbook, he/she can perceive the different groupings within the page even without actually reading the page. This means it should be possible to reason over a page without actually reasoning over the meaning of the each and every individual word or sentence.

Igarashi et al. [28], experimented over this notion.They aim to recognize hidden or implicit structures in human-organized layouts by creating an iterative human-perception-like parser. Their main concern and focus is to find the grouping relations between visual objects in 3-dimensions. Every object consists of a block of text or image.Then these

contents are used to define a context among the groupings to increase accuracy and human-likeness.

In this thesis we are simulating a human like textbook parserbased on similar notionswith Igarashi et al. However, we are concerned with 2-dimensions, and the objects are required to be detected by the system automatically before they can be classified and ordered into structure.

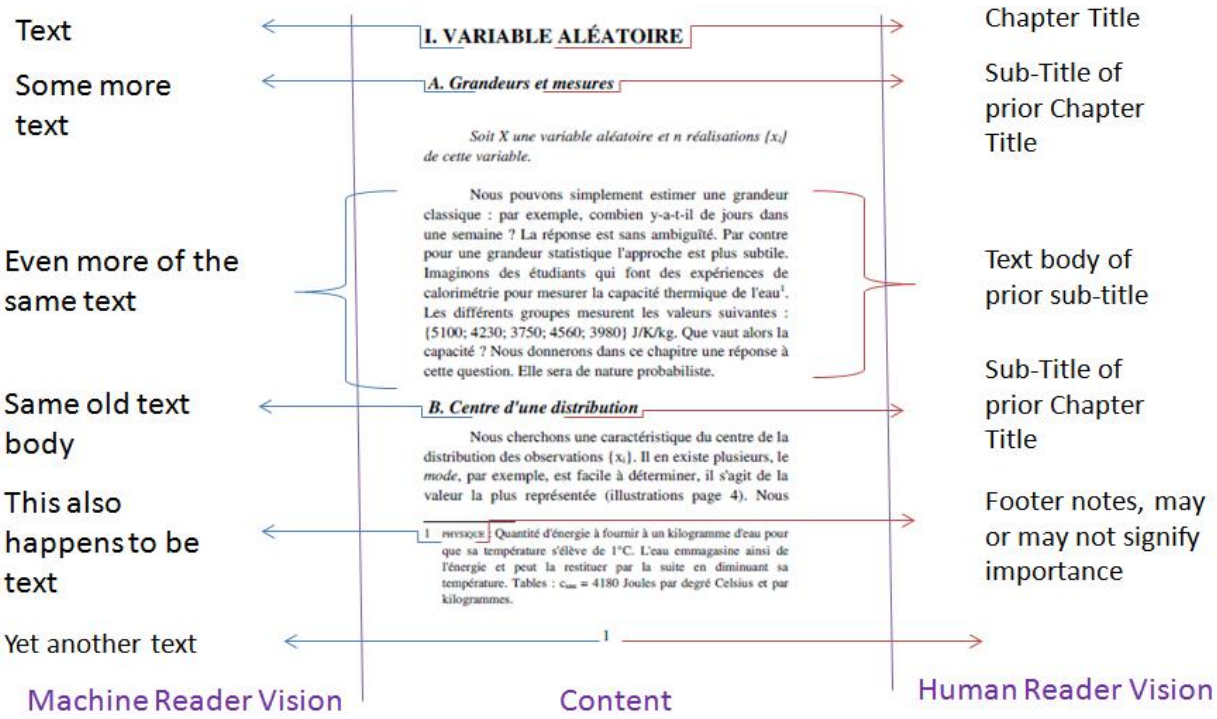


Figure 4.1 Different perspectives over the same content

As briefly explained in [3], and covered in depth in [28], the human-reader perceives the objects (in this case text blocks) as collections by clustering them through their characteristics such as proximity and similarity.When two objects have smaller proximity in between than the rest of the objects, usually they are grouped together. On the other hand, even when the proximity between two objects is small, if there is a big visual difference between the objects, then the human perception will group them separately. This can be observed in Figure 4.2. The gray box can be grouped with the vertical boxes above it, or with the group on the left side of it. However, the gray block is perceived by itself as a group due to its visual difference.While the three boxes above, and the five boxes on the left side perceived as two different clusters.

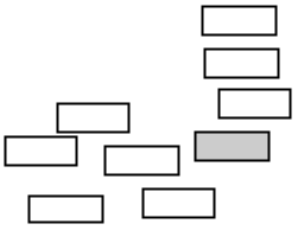
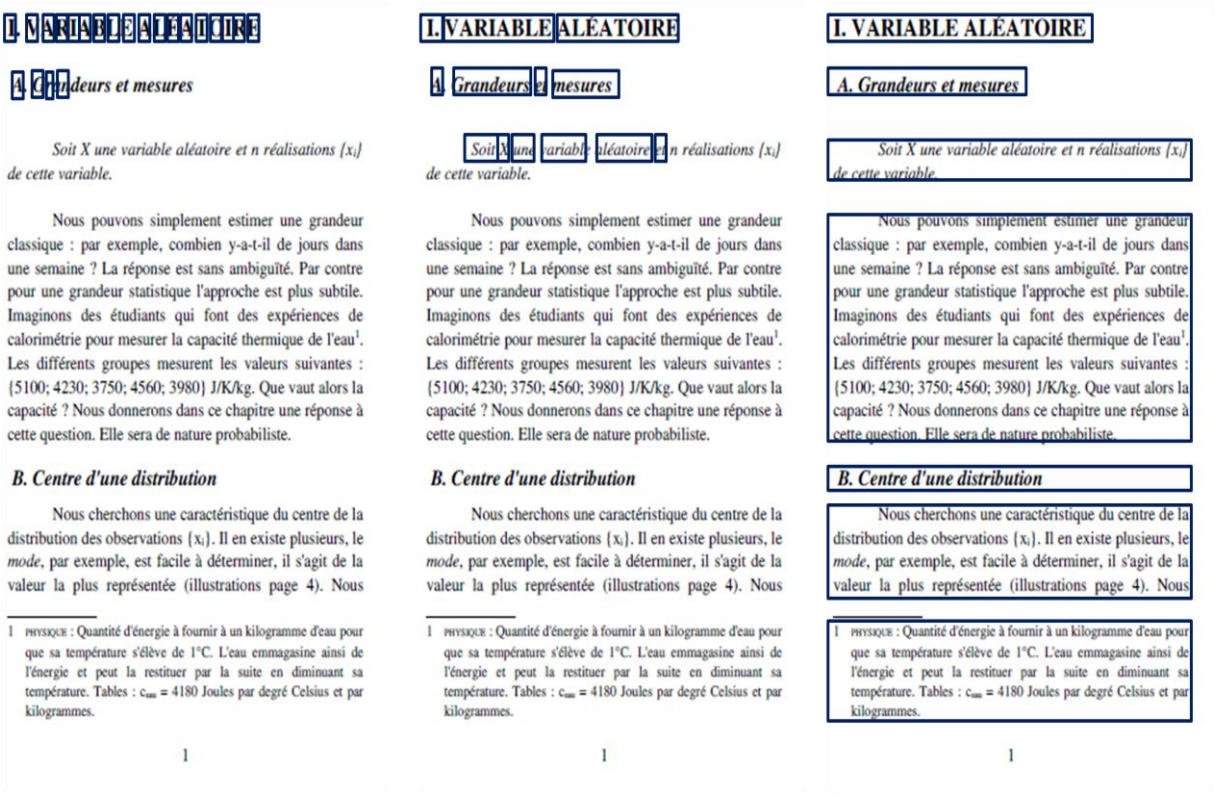


Figure 4.2 Human Perception Clustering Example

Another important aspect is that the nature of human perception. It functions in both bottom-up and top-down manner. The proposed approach focuses on the bottom-up functionality, because the system first requires to obtain the structural components from the textbook. In other words, systemfirst establishes objects, and then creates structure.

The process of a bottom-up human-reader perception proceeds as the following (Figure 4.3); the reader perceives the proximity of each character to its neighbor to identify the words. Then the proximity of each word to its neighbor is perceived to identify lines. After that, the proximity between lines is used to identify the line clusters. In this step, the similarities among line blocks are also important to define line clusters. Here, the similarity refers to a visual similarity than a similarity in context and content. The steps concerning the forming of all elements will be covered in the Chapter 5.2 in more detail.



I. VARIABLE ALÉATOIRE

A. Grandeurs et mesures

Soit X une variable aléatoire et n réalisations $\{x_i\}$ de cette variable.

Nous pouvons simplement estimer une grandeur classique : par exemple, combien y-a-t-il de jours dans une semaine ? La réponse est sans ambiguïté. Par contre pour une grandeur statistique l'approche est plus subtile. Imaginons des étudiants qui font des expériences de calorimétrie pour mesurer la capacité thermique de l'eau[1]. Les différents groupes mesurent les valeurs suivantes : {5100; 4230; 3750; 4560; 3980} J/K/kg. Que vaut alors la capacité ? Nous donnerons dans ce chapitre une réponse à cette question. Elle sera de nature probabiliste.

B. Centre d'une distribution

Nous cherchons une caractéristique du centre de la distribution des observations $\{x_i\}$. Il en existe plusieurs, le *mode*, par exemple, est facile à déterminer, il s'agit de la valeur la plus représentée (illustrations page 4). Nous

1 PHYSIQUE : Quantité d'énergie à fournir à un kilogramme d'eau pour que sa température s'élève de 1°C. L'eau emmagasine ainsi de l'énergie et peut la restituer par la suite en diminuant sa température. Tables : c_{eau} = 4180 Joules par degré Celsius et par kilogrammes.

1

Figure 4.3 Human Reader Text Recognition

5Approach

5.1 Approach Overview

First, the text is parsed to obtain its formatting information. Then the physical elements such as lines (4.1b) and the logical elements such as titles(4.1c) are created based on the formatting information. Finallythe structureisconverted(4.1d) into a machine readable format, so it can be integrated and reused later on.

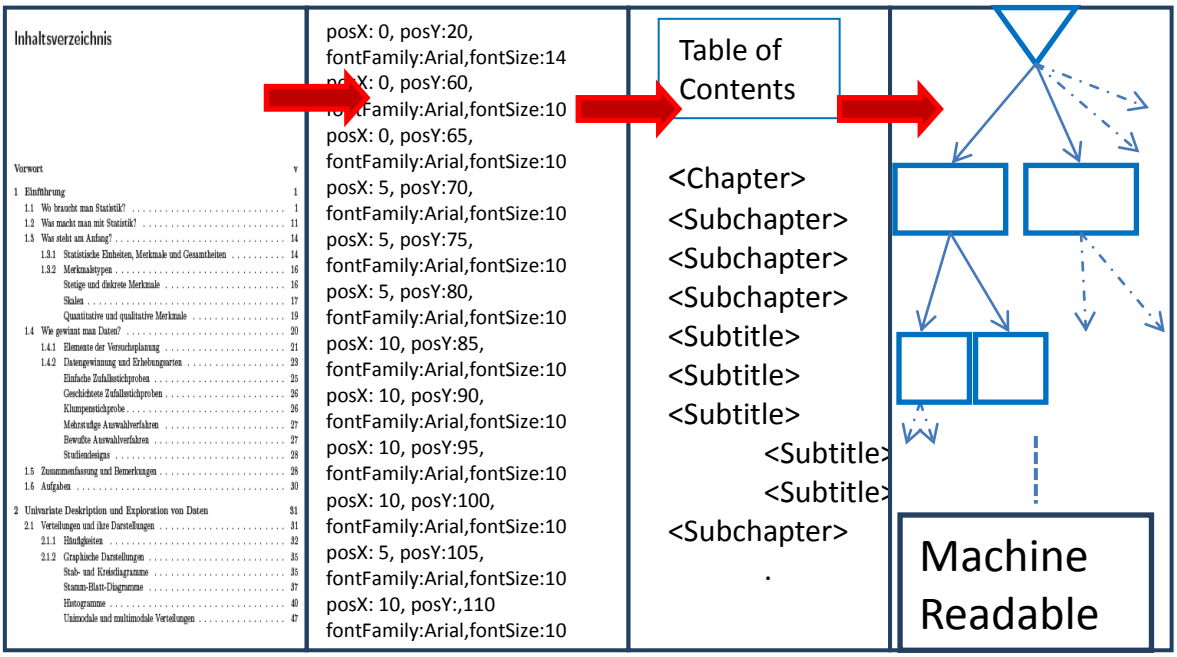


Figure 5.1 (left to right a ,b, c ,d) The Approach Steps

5.2Parsing

The depth and the detail of the provided information by PDFtends to vary from document to document. However, a minimal amount of required visual properties for every printed character are always providedsuch as the font family, the font size and the coordinates.To extract them a simple PDF parser library (PDFBox[‡]) is sufficient.

[‡]https://pdfbox.apache.org/

When the PDF document is put through a parser, the parser reads the document page by page. Every page is processed as a content stream. The content streamsare processed character by character.

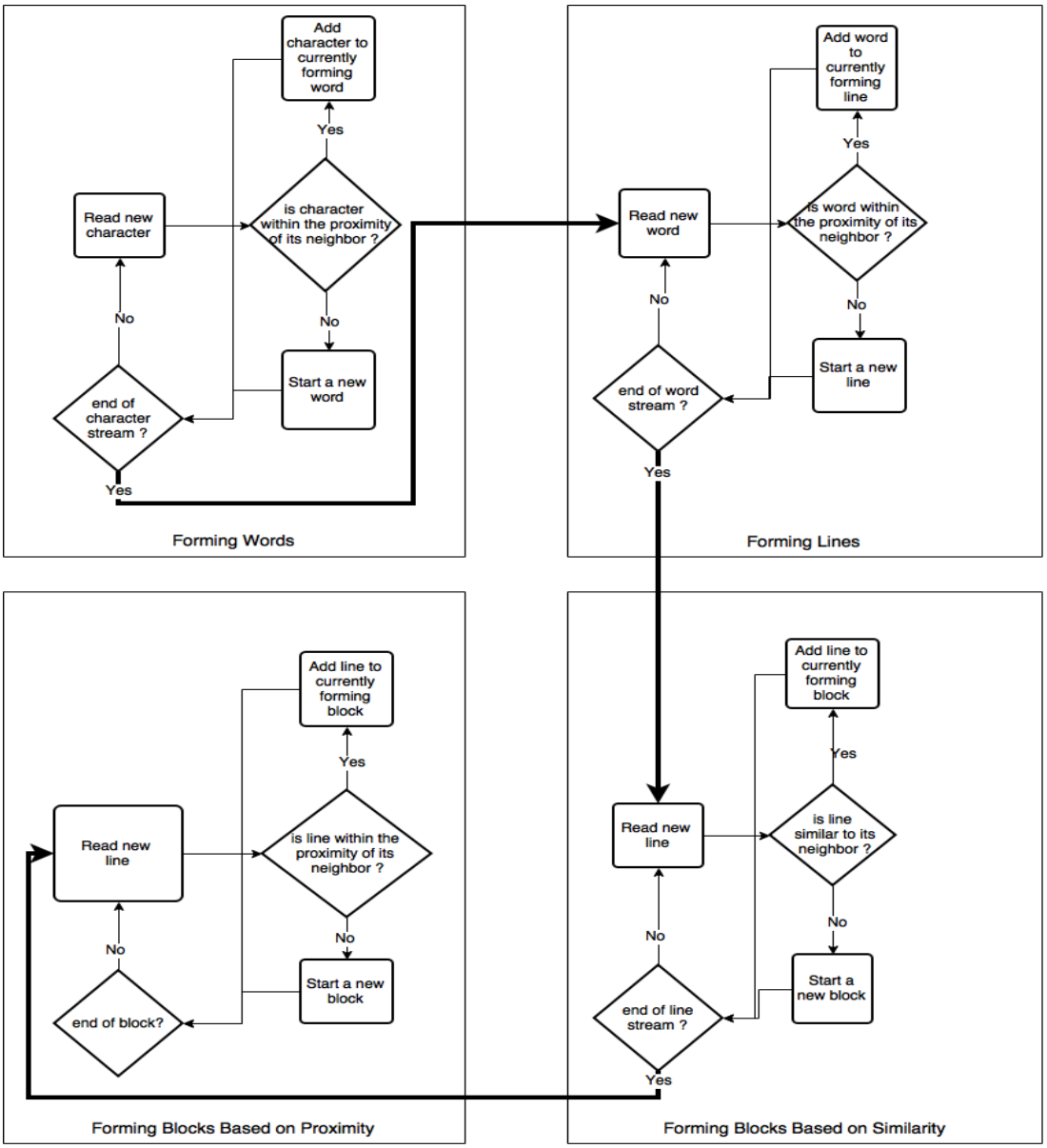


Figure 5.2 Parsing Architecture

As mentioned in the section 4.2(Figure 4.3), the perception starts with grouping characters into words, and ends with identifying the respective structure among text blocks. Since this work aims to create an approach close to human reading, it follows the human-perception example and performs the parsing in a bottom-up manner. The general parsing architecture is depicted in the Figure 5.2.

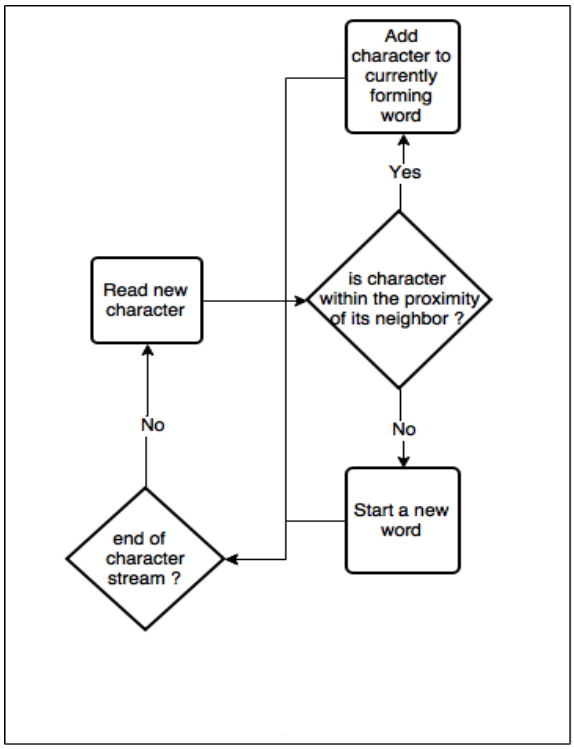


Figure 5.3 Forming Words

The system starts by reading a character (Figure 5.3). Each time a character is read, its formatting (font-size, font-type) and geometrical (coordinates) information are extracted from the character`s stream. Then the proximity of current character`s coordinates are compared with the coordinates` of its preceding neighbor to form the words. Both the X and Y - coordinates are used to perform this proximity checks. If the adjacencybetween two characters is within the expected range, then they form a word. If the preceding neighbor already belongs to a currently forming word, then the current character is appended to it. However, if the adjacency between them is not within the expected range, then the current character marks the beginning of a new word, and the previously forming word is concluded. The expected thresholds for proximity are determined by the font-size of the characters. Every font-size has its own expected whitespace distance. This information is embedded with the font-size information in the PDF document, and obtained by the PDF parser automatically.

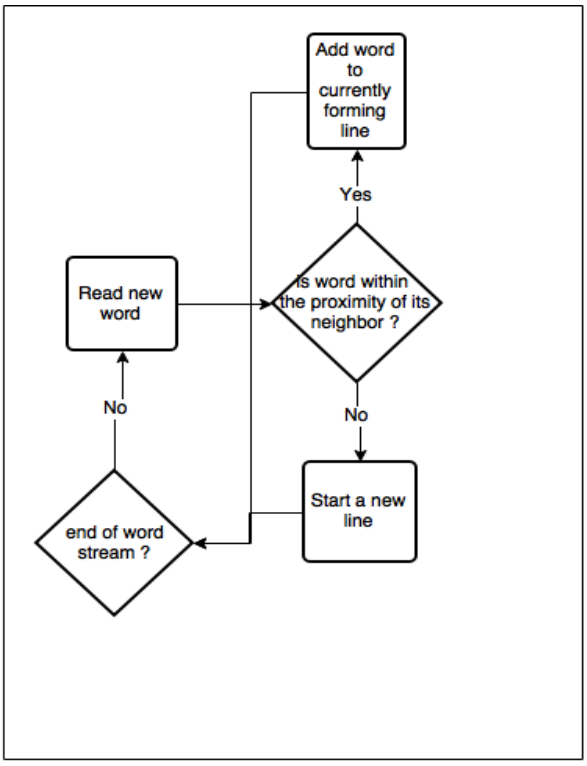


Figure 5.4 Forming Lines

As soon as all the characters in a page are formed into words, the procedure progresses onto forming lines from words (Figure 5.4).During this step, words are used instead of characters and the proximity check is performed by comparing only the Y-coordinates. This phase starts with reading a formed word. The Y-coordinate ofthisword is compared with its preceding neighbor`s. If the proximity of two words are within the expected the range, thenthey are formed into a line. If the preceding neighbor already belongs to a currently forming line, then the current word is appended to it. However, if the proximity between the two is not within the expected range, then the current word marks the beginning of a new line, and the previously forming line is concluded. The expected thresholds for the proximity are determined by the font-size of the words. Every font-size has its own expected maximum character height and line-spacing between two lines. The generic nature of parsers has a tendency to causeerrors when there are corner cases.A superscript, e.g.2nd, over a character is one of those corner cases. The positioning of it may cause false lines and words. That is why, for higherprecision thedefault thresholds provided by PDF are enlarged to avoid such problems.

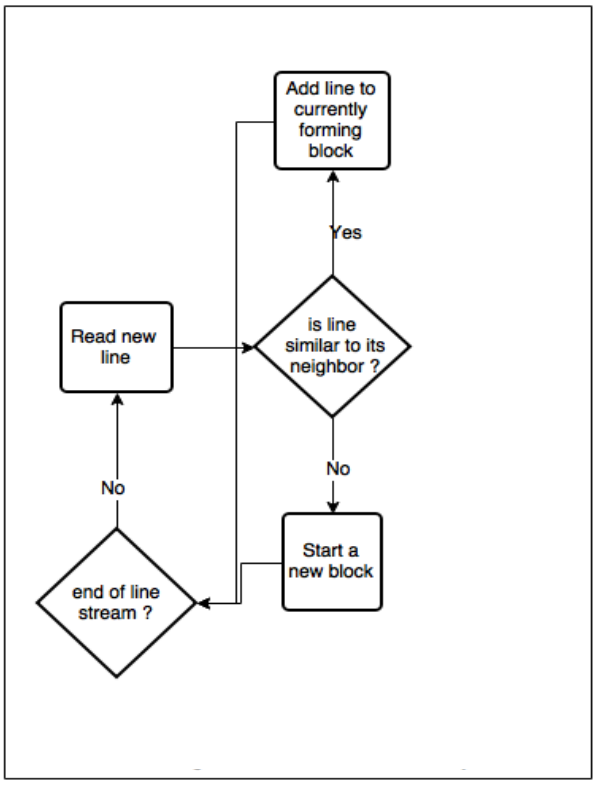


Figure 5.5 Forming Blocks Based on Similarity

In the following step text-blocks are formed from lines (Figure 5.5).Unlike word and line creation, applying only the proximity check during the text-block creation may introduce errors. When there is a list in the page,falsely formed text-blocksbecome a problem. To avoid this, thesimilarity of the lines is used before the proximity of the lines to form the text-blocks.The font-size is the main criteria for the similarity check. As explained in section 4.1, the same logical elements in a textbook share the same formatting information, including font-size. In light of this, the first set of text-blocks are formedby reading a line and comparing its font-size with its preceding neighbor. If the font-sizes match, then they are formed into a text-block. If the preceding neighbor already belongs to a currently forming text-block, then the current line is appended to it. However, if the font-sizes do not match, then the current line marks the beginning of a new text-block, and the previously forming text-block is concluded. The font size of a line is defined based on the most frequent font-size of its member words. In most of the instances, every line has one font-size occurring more than the others, and it is referred as dominant font-size of the line in this work. This way, the lines belonging to the logical elements of same levelare grouped together. Since the proximity of the lines were not taken into consideration during this step, it is possible to have some lines with big gaps in between within a text-block. An instance of such would be

having bullets of a list and the following paragraph in the same group, because they have the same font-size(Figure 5.6a). At this point, the proximity check between lines is applied within every text-block to fix it.

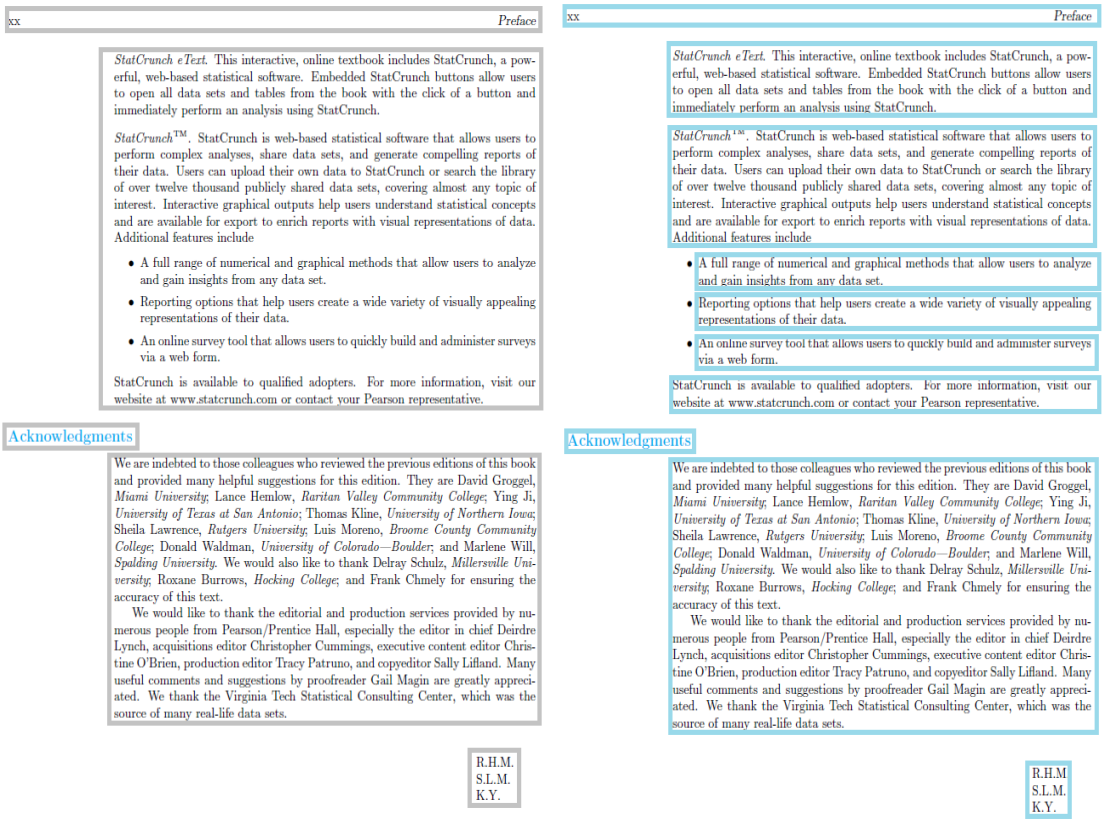


Figure 5.6 Text-Block Creation With Similarity (left to right) a, b

The proximity check (Figure 5.7) is the last step of the parsing. With this, previously formed text-blocks are divided further into smaller text-blocks according to the line spacing between lines. At the end, it would look as it would be perceived by a human reader. The threshold to split text-blocks is its most recurring line spacing value. If the line spacing between two lines equalsto thethreshold value, then they are formed into a text-block. If the preceding neighbor already belongs to a currently forming text-block, then the current line is appended to it. However, if the line spacing is bigger than thethreshold, then the current line marks the beginning of a new text-block, and the previously forming text-block is concluded. This way the cluster in the Figure 5.6a is transformed into the cluster in the Figure 5.6b.

The entireparsingactions correspond to theFigure 5.1b. The next step is the identification of the logical elements in the pages.

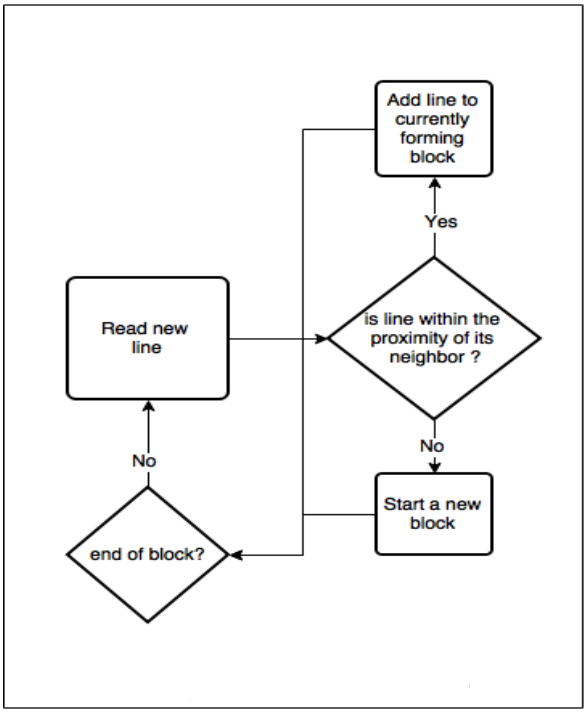


Figure 5.7 Text-Block Creation with Proximity

5.3Logical Element Extraction

This section corresponds to the Figure 5.1c, and it aims to define the logical meaning behind the extracted text-blocks.

5.3.1 Logical Element Identification

A human reader achieves the recognition of logical elements throughcomparison of the perceptually formed text-blocks with his/her accumulated knowledge and experience. These experience and knowledge mostly consist of the basic identification rules and layouts for logical element classification.

It should be possible for acomputer to simulate a similar functionality with a rule based comparison algorithm. With this in mind, the system attempts to identify the logical labels of the text-blocks by employing the generic rules for writing a textbook.

In order to speed up the comparison process, and to make it feasible to apply to a whole textbook, the text-blocks with same formatting information are grouped under the same formatting labels (Figure 5.8).A catalogue is created from the group labels, namely formatting dictionary. Thanks to this, the system avoids extensive amounts of comparison

actions between all text-blocks. It only needs to compare the formatting information betweentheselabels to identify their logical order. The format dictionary of Figure 5.8 can be seen in Figure 5.9.

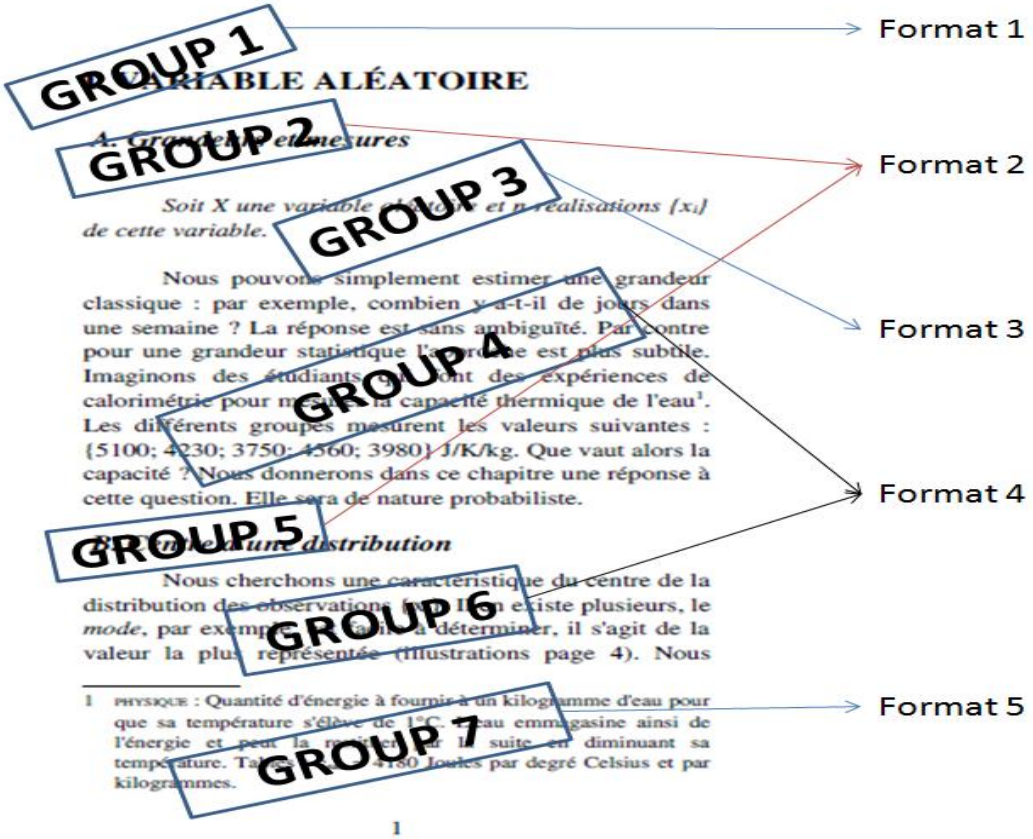
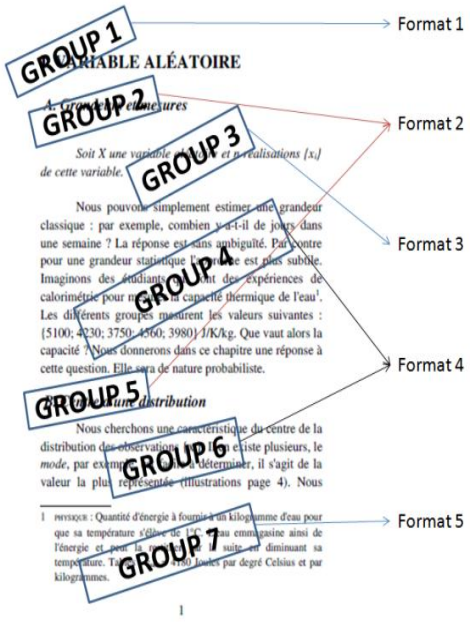


Figure 5.8 Format Assignment of Text-Blocks

The rules to order the labels are obtained from the generic rules and guidelines to write a textbook. Some of them were mentioned during the Section 4.1 and the Section 4.2.The order of importance is decided as the following: If font-size is bigger, then it is placed higher. If font sizes are the same, then the one with a font-face (bold, italic) is placed higher. If both font-sizes and font-faces are the same, then the one closer to the left side of the page is placed higher.

While thelabel ordering provides a general layout for the hierarchy of the text-blocks, it lacks to provide definitive information as for their logical meaning. This is because,the ordering does not contain any kind of base value to compare the labels.However, it is enough to identify only one label`s logical meaning to be able to infer

the rest from this order. Hence, the system only requires one base value to resolve the whole dictionary.



Format	FontFamily	FontSize	FontFace
Format 1	Arial	16	Bold
Format 2	Arial	14	Bold
Format 3	Times New Roman	11	Italic
Format 4	Times New Roman	11	_
Format 5	Calibri	9	_

Figure 5.9 FormattingDictionary

To keep the approach as document independent as possible, the system relies on the information that can be obtained from the content. The idea is that the dictionary entry label with the highest number of occurrence throughout the textbook should be the one for the regular text and paragraph. The respective numbersfor each labelare calculated when they are created. Any label placed below this label is considered as insignificant and disregarded, since they do not directly contributeto the structure. Anything above this label isidentified as title/sub-title.

For example in Figure 5.9, the order would be, in a descending order, Format 1 > Format 2 > Format 3 > Format 4 > Format 5. According to the highest number of occurrence, Format 4 would be the regular text label. Format 5 would be discarded, because it is placed under content format in the order. Format 1, Format 2 and Format 3 would be labeled as title/sub-title.

However, theexample above has a false-positive. Format 3 is not a title, even though it was labeled as such. That is because it has aitalic font-face. To fix such errors, a cross check needs to be applied. During this process, the TOC entries are compared with the

identified titles/sub-titles to filter out the false positives.This creates an opportunity to identifythe logical elements between the titles and the regular content in the formatting dictionary. However, this deeper identification is out of our scope for now.

When the results of the example above is cross-checked with the TOC entries, Format 1 will keep itstitle label, Format 2`s label will stay as sub-title, and Format 3 will lose its sub-title label, since it does not appear in the table of contents.For the simplicity of requirementsof this work,such logical elements are labeled as content. Hence, Format 3 will be labeled as content. However, theidentification of such in-between elements would be a good direction asa future work.

Even though it is possible to obtain the structure and titles only from the TOC, the method proposed uses it as a fail-safe. The reason is that the structure extraction from TOC fails when there are resources with a badly formatted TOC, or without one atall. Our approach makes sure that the system always provides logical elements and a structure for the textbook, even if it has to sacrifice accuracy to some extent.

5.3.2Special Case Logical Entities

During the extraction, it is important to make use of the easy-to-detect logical entities whenever possible, namely the index and theTOC.They are easy to detect because of their well definedlayouts and unique placement in the textbooks (by experience table of contents mostly among the first 20 pages, while index is among the last 20 pages). Both theTOCand the index have the distinct layout of having a page number at the end of most of its lines throughout a page. However, the index usually contains more lines, and columns than the TOC. The combination of these distinct layouts and the pre-determined locations in the textbookmakes it possible to detect both without any of the above mentionedlabel comparisons.

As an addition to the above mentioned logical entities, we have one extra unique case due to the nature of the INTERLINGUA. As mentioned in the Chapter 3, INTERLINGUAaims to create meaningful links in between the chapters of different textbooksof different languages. One way for the system to achieve this is to extract the index terms fromthe textbooks, and to link themto the glossary terms from the reference-

ontology. By doing so, the systemenriches the structures and creates a port from the structures to external resources.The reason why glossary terms are counted as a logical entity is that they can appear in a textbook through their links to the index terms. However, it is not rare to come across to the glossary terms in the content even though they are not included in the index.Regardless, those terms still carry the same importance with the indexed terms for the system , and that makes them a unique case of logical entity.

5.3.2.1 Table of Contents Extraction

The practicality of the TOC is agreed by all the people working on the field of layout analysis and extraction. When it is formatted properly, it can be said that extractingit is the fastest way to get the structure and the hierarchy of a document. Hence, it is a common practice to check its existence during the structure extraction before anything else.

The detection of TOC is one of the two points in this thesis where the systemis language dependent. As explained at the beginning of this chapter, theTOC always has an indicating title, regardless of the resource`s language.The most common way to recognize a TOC isto searchfor this title within the first twenty page. Another approach isto look for its unique layout; every TOC entry has a page number at the end. The only issue with this second approach is that the other logical entities like list of figures have the same layout. Hence, it is highly possibleto mistake one for another. To avoid this, the first approach is employed to detect the beginning of the TOC, and the second one to detect the end. When the system finds the title, it is followed by line ending check.If the majority of the lines are ending with a page number, than that page is added to the TOC. The system keeps checking the following pages with the second approach until it comes across to a line with the font-size of the TOC title, or a page which does not fit into the layout.

After the TOC is identified, the following step is to process it to extract the hierarchy and the structure. There have been different approaches to achieve this. Most of them have a specific scope of TOC style. An example work is done by Luo et al. [6] to detect the TOC in Japanese documents by detecting the connecting dots between page numbers and the titles.While it is an effective way, it would fail to work in the absence of the connecting dots. Another examplewas proposed by Mandal et al. to detect the TOC from

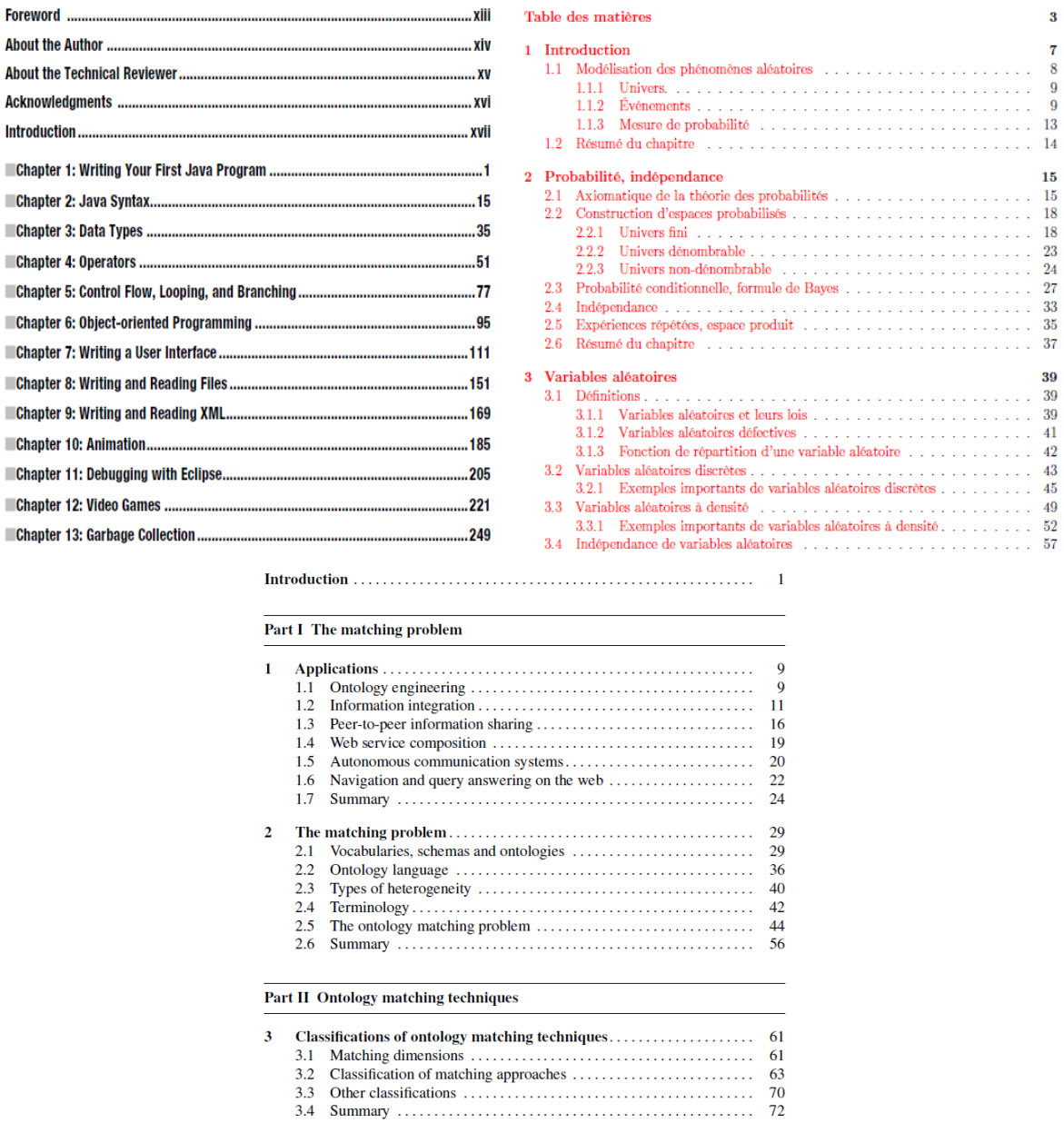
document images by using page number-related heuristics [31].Tsuruoka et al. [7]detects the chapters and sections from table of contents based on the indentation and font-size variation[7]. This method fails when it isapplied to a TOC without any indentations.

This thesis follows a relaxed version of theapproach proposed byWu etal. [10]to determine the structure from the TOC(Chapter 2.3). It can be summarized as the following;

ATOC is flat (Figure 5.10a) when all of its entries share the same visual formatting, the same start X coordinates and the same line spacing in between the entries. It is ordered (Figure 5.10b) when the entries start with ordered chapter/section numbers. It is divided(Figure 5.10c) when the entries can be grouped based on the differences between the line spacing of the entries.

Each layout has its own rules to extract the structure;

- When a TOC is classified as flat,it means that every entry is on the same level. Hence, the structure will have a flat hierarchy.
- When it is classified as ordered; following the order of the section numbers is sufficient to extract the structure and the hierarchical relation between entries.
- When it is classified as divided; each group corresponds to the same level in the hierarchy and the entries in a group are hierarchically placed under the group`s first entry.



Foreword xiii
About the Author xiv
About the Technical Reviewer xv
Acknowledgments xvi
Introduction xvii
Chapter 1: Writing Your First Java Program 1
Chapter 2: Java Syntax 15
Chapter 3: Data Types 35
Chapter 4: Operators 51
Chapter 5: Control Flow, Looping, and Branching 77
Chapter 6: Object-oriented Programming 95
Chapter 7: Writing a User Interface 111
Chapter 8: Writing and Reading Files 151
Chapter 9: Writing and Reading XML 169
Chapter 10: Animation 185
Chapter 11: Debugging with Eclipse 205
Chapter 12: Video Games 221
Chapter 13: Garbage Collection 249

Table des matières 3

1 Introduction 7
1.1 Modélisation des phénomènes aléatoires 8
1.1.1 Univers. 9
1.1.2 Événements 9
1.1.3 Mesure de probabilité 13
1.2 Résumé du chapitre 14

2 Probabilité, indépendance 15
2.1 Axiomatique de la théorie des probabilités 15
2.2 Construction d'espaces probabilisés 18
2.2.1 Univers fini 18
2.2.2 Univers dénombrable 23
2.2.3 Univers non-dénombrable 24
2.3 Probabilité conditionnelle, formule de Bayes 27
2.4 Indépendance 33
2.5 Expériences répétées, espace produit 35
2.6 Résumé du chapitre 37

3 Variables aléatoires 39
3.1 Définitions 39
3.1.1 Variables aléatoires et leurs lois 39
3.1.2 Variables aléatoires défectives 41
3.1.3 Fonction de répartition d'une variable aléatoire 42
3.2 Variables aléatoires discrètes 43
3.2.1 Exemples importants de variables aléatoires discrètes 45
3.3 Variables aléatoires à densité 49
3.3.1 Exemples importants de variables aléatoires à densité 52
3.4 Indépendance de variables aléatoires 57

Introduction 1

Part I The matching problem

1 Applications 9
1.1 Ontology engineering 9
1.2 Information integration 11
1.3 Peer-to-peer information sharing 16
1.4 Web service composition 19
1.5 Autonomous communication systems 20
1.6 Navigation and query answering on the web 22
1.7 Summary 24

2 The matching problem 29
2.1 Vocabularies, schemas and ontologies 29
2.2 Ontology language 36
2.3 Types of heterogeneity 40
2.4 Terminology 42
2.5 The ontology matching problem 44
2.6 Summary 56

Part II Ontology matching techniques

3 Classifications of ontology matching techniques 61
3.1 Matching dimensions 61
3.2 Classification of matching approaches 63
3.3 Other classifications 70
3.4 Summary 72

Figure 5.10 a, b, c (left, right, bottom) TOC Types

5.3.2.2 Index Extraction

Unlike the TOC, the index extraction did not get much of an attention as a standalone area of interest. The content related to it does not go beyond small sections and remarks in some articles in the structure extraction field.

An index keeps track of the introduction point of every important key-word or term within a textbook.The terms are usually the same in every textbook for the same language in the same domain. This means that when an author is talking about "geometric mean" in a

statistics book, the term will appear as "geometric mean" in the other textbooks ofthe same domain.

The consistency of the terms within a language creates the connection between different textbooks in a language. The knowledge of the terms that appear in a section or chapter would create the chance to relate the chapters, the sections or even the paragraphs to external resources where all or some of the same terms appear.

With this in mind, we decided to focus on the index by taking into account its own special circumstances. It usually appears within the last 20 pages of a book, and usually consists of two columns. However depending on the size, the coverage, and the page limitation of a textbook, it can consist of one or three columns. Every index term entry ends with a page number. Some index examples can be seen in the Figure 5.11.

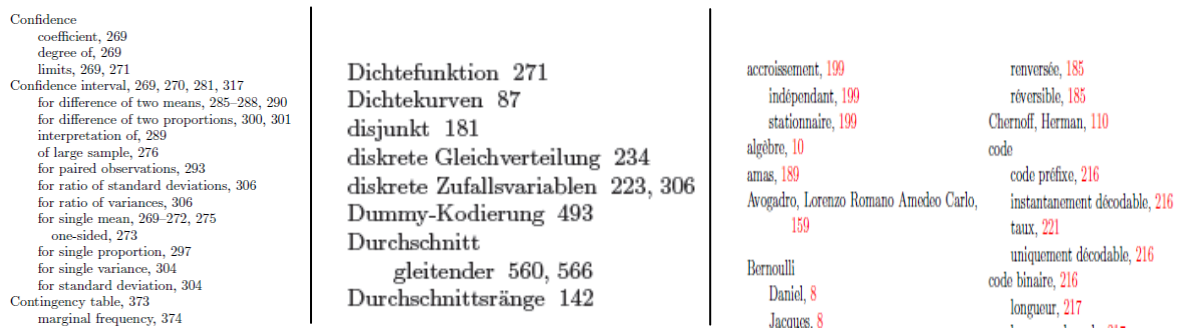


Figure 5.11 Example Index (left to right) a, b, c

The detection of the index is the other point, where the systemis language dependent. Same as the TOC, the index has indicative titles at the beginning. However unlike the TOC, the index exists at the end of the textbook.It is a common practice to include the index in the TOC. Since at this point the TOC is already extracted, it is possible to search for the index in its entries. If we find it, we extract the page number from the corresponding TOC entry and mark it as the start of the index. In every now and then , it is possible that the index was excluded from the TOC even though it exists. To cover for those corner cases, the system proceeds to perform the indicative index title same search within the last twenty pages. After the beginning of the index is detected, every following page is checked to find out if they are fitting into the rules defined for the index layout. This is achieved by deconstructing the columns of pages into one column, and then checking if most of the lines are ending with page numbers.

When the index is identified, the next step is to extract the index terms. The extraction of index terms has two corner cases. The first one is the multiple lined index terms (Figure 5.11a), andthe second one is the grouped index terms (Figure 5.11b).

The multiple lined index terms are rather simple to handle. Instead of having a page number at the end of the first line, they have the page number at the end of the line where the term ends. During the term extraction, the lines are appended to the same index term until there is a line ending with a page number.

The grouped index term refers to the index terms with a nested hierarchy.Sometimes to collect some relevant properties of a broad index term, the authors put that index term with or without a page number and then start to add the following related terms under it with an indentation. A group index term can be seen in the Figure 5.12. These groups should be resolved before the construction of index terms.

Figure 5.12 Multi Layered Grouped Index

The extraction of the grouped index terms is harder due to two reasons: first, the requirement of separation between the grouped index terms and the multi lined index terms, second, the grouped index terms may contain another grouped index terms.

A nest is detected by checking the following index entries` start positions. The grouping continues until the system comes across to an index term with the same starting X coordinates as the first entry of the group. The detected index terms up till this point are added to this group.When the nest is defined, an identification process starts from the lowest level to combine the entries with their nest parent. This is done in a recursive function, so that thenest can be resolved without extra concern, even if it contains other

nests in it. Lastly, the system searches for the formed term in the respective page to confirm the accuracy of the terms word order.

For example, in the Figure 5.12, the term 'Index' is the first entry of the nest, and itslast entry is 'von Paasche'. The terms 'Aktien-', 'Mengen-' and 'Preis-' are the sub groups of the nest. The entries, 'von Laspeyres' and 'von Paasche' are the lowest level of the nest. The resolving process starts from the lowest level and combines them with their parents. In this example this creates 'Mengen- von Paasche' etc. After this, the process climbs up to the upper level nest and repeats the same action. An instance of this would be ' Index Mengen- von Paasche 552'. Then, the 552 is examined to check the term. After this examination, the term is reformed into 'Mengen- Index von Paasche'.

As explained in the section 5.3.2, the glossary terms usually exist in a textbook as index terms. To be able to create a structure as rich as possible, we are linking the core ontology glossary terms to the corresponding sections through the index terms of the textbook.

The procedure to achieve this is simple. After the extraction of the index terms are done, we are performing string similarity (Cosine Similarity) between each index term and the glossary term labels. The index terms and the glossary terms with the highest similarity are linked to each other. To keep a high accuracy for the linked parties, there is a minimum threshold of 0.7 for the similarity result. This value is decided after many experiments to avoid the false positive matches due to the similarity between different words (e.g. bed and bad), but to detect the conjugated form of the words in a sentence.

The linked terms are stored in a database. These information are later on used to enrich the structure to create a richer semantic model for the textbook. The linked glossary terms are dropped from the glossary term list to avoid possible overlaps during the glossary term detection step.

5.3.2.3 Glossary Term Detection

Due to the flexible nature of textbook authoring, it is possible for an author to use a term related to the domain, but refrain from adding it into the index of the textbook.Those

skipped, excluded or forgotten terms may be trivial from the perspective of the author for the current section or chapter. However, this does not change the fact that those terms are relevant and important for the section.

This means that it is possible to find glossary terms from the reference-ontology within the textbook without any links to the index terms. Their existence may be insignificant within the content, but the relation originating from the glossary term`s existenceisbeneficial for the structure enrichment.

The detection of these terms are achieved by string similarity (Cosine Similarity). Unlike the index detection, the glossary term detection does not haveany predefined page number to inform the system about the whereabouts of the term. Hence every glossary term is searched in every page of the textbook.

For the search, an open source information retrieval library (Lucene) is used. Even if the Lucene is not 100% accurate, its speed makes it a viable choice. It returns the pages for the terms with the highest possibility without much of strain to the platform. The system applies the string similarity function over the five pages with the highest match values returned by the Lucene. When the string similarity is above 0.7 in one of the pages, the glossary term is appended to the index list of the textbook. If there is a similarity tie in between pages, the one with smaller page number is assigned as the term`s introduction page. This approach is saving a considerable amount of computational time and resources, while losing an acceptable fraction of accuracy.

5.4 Semantic Conversion

The semantic conversion is the process of storing all the data extracted up until this point in a SKOS model.

SKOS is a W3C standard, based on other Semantic Web standards (RDF and OWL).Itprovides a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies and other similar types of controlled vocabulary [32].

The main hierarchical structure is obtained from the extracted TOC. Every detected chapter, sub-chapter, section, sub-section are put in the order with broader and narrower relations of SKOS. For example; if a chapter title A has a section titled B, the concept of A is going to have a 'broaderThan' relation with concept of B, while concept of B has 'narrowerThan' relation with the concept of A (Figure 5.13).

This way all the extracted title/sub-title elements will be converted into their respective SKOS concepts with their respective hierarchical relationships. Again, the hierarchical relation between these elements is determined based on their corresponding formatting dictionary entry. After this step is done, the end result will represent the hierarchical structure of the whole textbook (Figure 5.14).

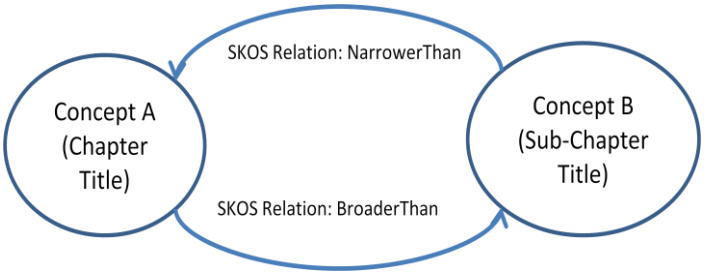


Figure 5.13 An Example SKOS Relation Between Titles

In the following step, the systemconverts theextracted index (Section 5.3.2.2) and the glossary (Section 5.3.2.3) terms into SKOS concepts.These concepts have their page numbers linked to them (Figure 5.15).Next, the relations between the terms and the corresponding reference-glossary conceptsare established.

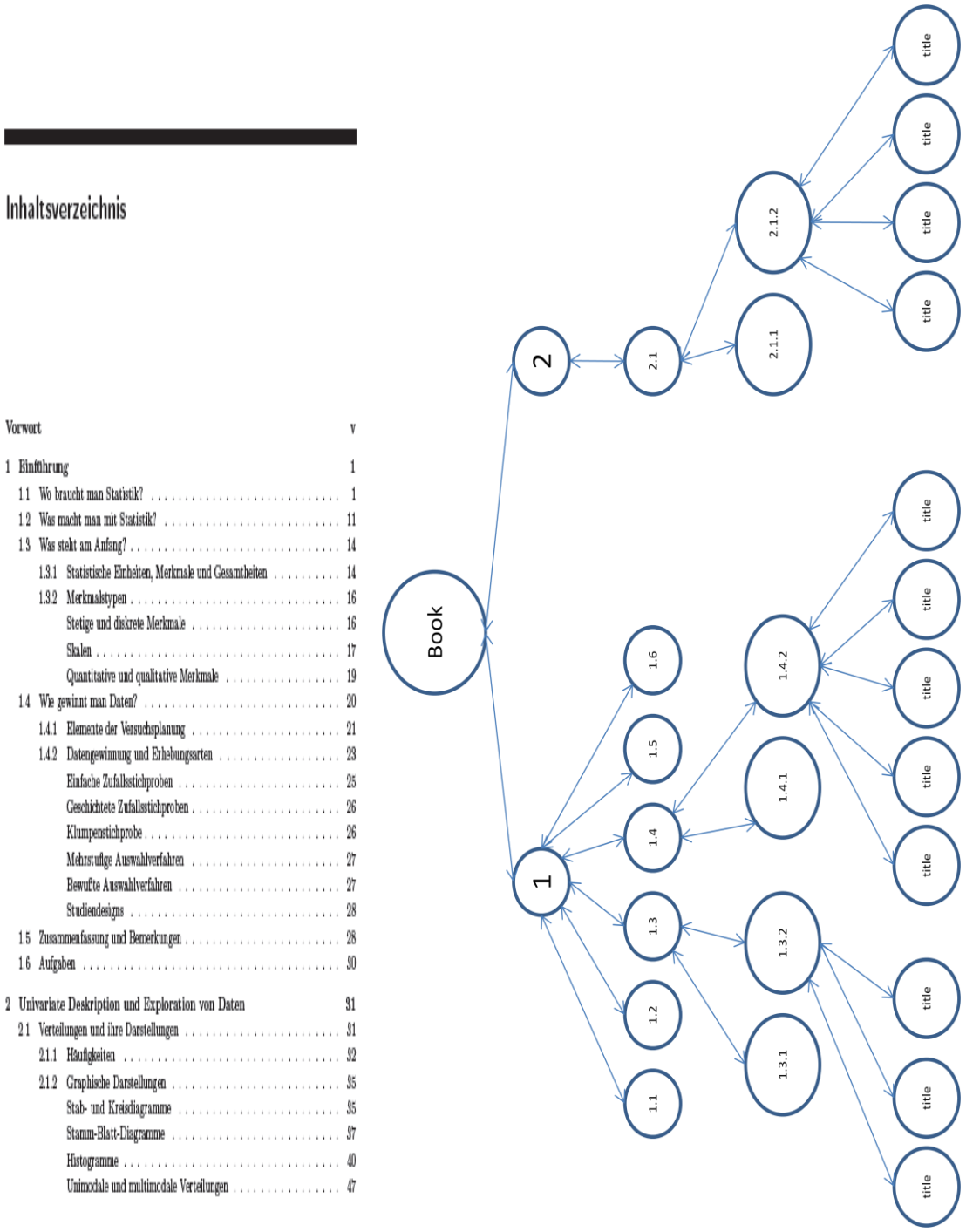


Figure 5.14 TOC snippet and its Structure

Lastly the system adds the links between the pages and the concepts ofthe SKOS model. Moreover, the paragraphsdetected during the logical element extraction are linked to their corresponding conceptstoo. A snippet from a resulting semantic model is depicted in the Figure 5.16, and an example of a connection in between two chapters is shown in the Figure 5.17.The Chapter 3.2 of the book A is connected to the Chapter 1.1 of the book B through the glossary term 'aleatory variable'.

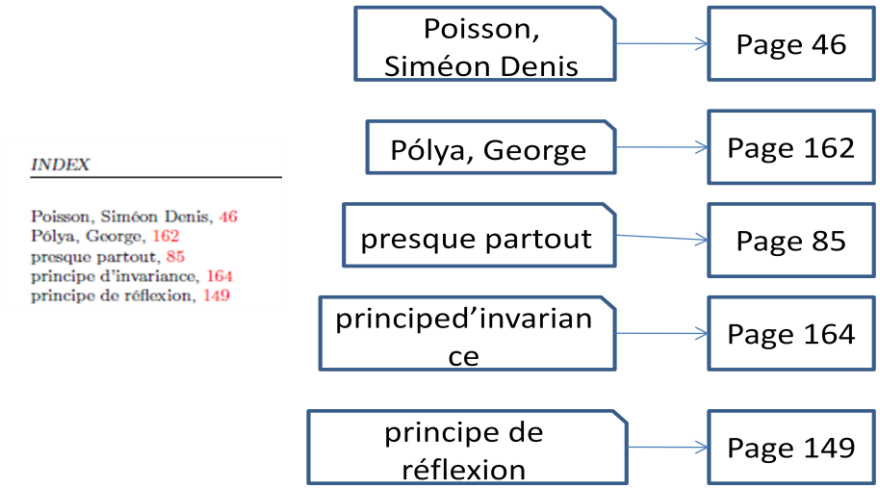


Figure 5.15 Index Snippet

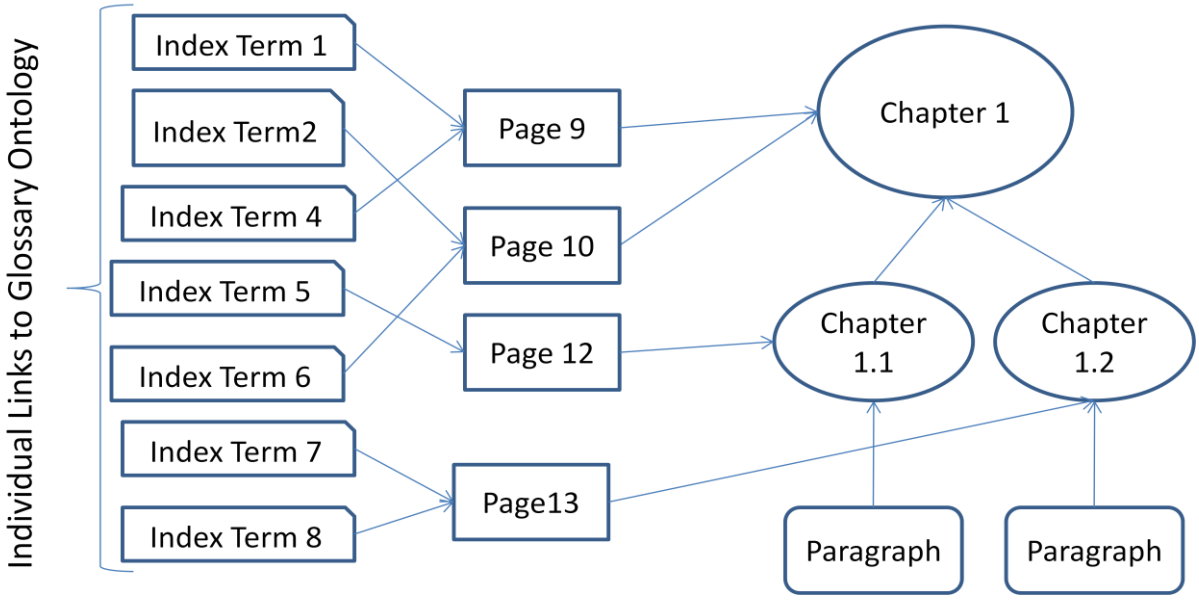


Figure 5.16 SKOS model snippet

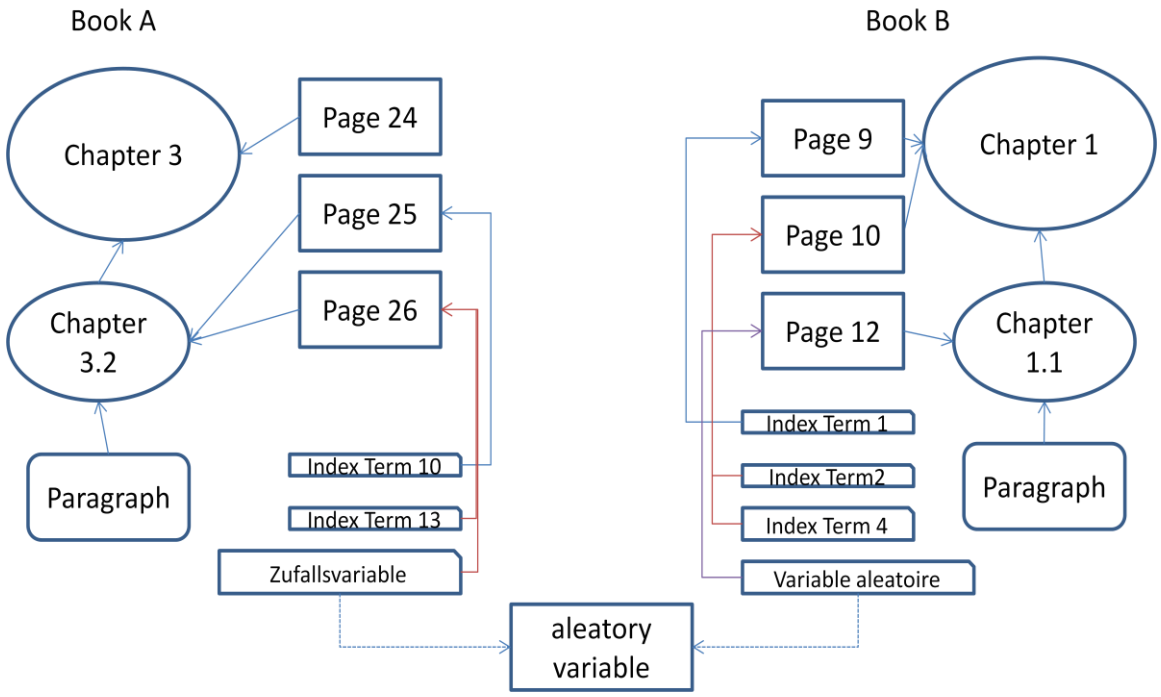


Figure 5.17 Connection Between Chapters of Two Different Textbooks

6 Results and Conclusion

This thesis has described an approach to identifying and extracting hierarchical structures from PDF-based textbooks, and further enriching the extracted structures with the semantic information obtained from the textbooks themselves.

The implemented software component extracts the major physical and logical elements from the pages of a textbook. These elements include components such as titles, footers, headers, pages numbers, paragraphs, the table of contents (TOC), the index etc. The system tries to simulate human reading behavior when recognizing structure from textual resources to extract the physical elements from the pages. It parses the textbook and constructs those elements in a similar fashion to the way a human mind does. Afterwards, to identify the logical meaning of the physical elements a simple rule set based on common guidelines to write a book is applied. Then, the hierarchical structure of the textbook is obtained from the logical element TOC. Next, the system extracts the terms used in the textbook from the logical element Index. The terms are linked one by one to an external glossary of the domain of the textbook. The index terms with external links are attached to the titles based on the page numbers of the terms to facilitate the connection of those titles to the external sources.Lastly, the enriched hierarchical structure is converted into an RDF ontology model. The created model can be used to link the titles of different textbooks to each other or connect a textbook model to another system to manipulate the textbook based on its structure.

Since this system is deployed under the INTERLINGUA project, the domain of processed textbooks is statistics. Three statistics textbooks were used to verify the approach: one per supported language. The French textbook [29] has 238 pages with an ordered TOC of size 118 titles, and it contains 298 index terms. The English textbook [5] has 613 pages with an ordered TOC of size 253 titles, and it contains 606 index terms. The German textbook [30] has 812 pages with an ordered TOC of size 251 titles, and it contains 625 index terms. Both quality and the quantity of the results are always relative to the size and the detail level of the resource.

During the detection and extraction of physical and logical elements no problems were detected, as long as the characters were represented and encoded properly in the content stream of the PDF. All the existing page numbers, headers, footer, titles and paragraphs were correctly identified.

Both the TOC and the index of all three textbooks were detected and extracted without any issues. Their hierarchical structures were created according to their TOCs. In the French textbook,13out of 118 titles were labeled as chapters. This number is 18 out of 253 for English, and 22 out of 251 for German.As for the index terms, all of the index terms were accounted for. Among those index terms, In the French textbook, 120 of them were linked to the reference-ontology; the corresponding numbers for the English and the German texts are 300 and 223.

This approach makes it possible to extract the structure of a textbook almost independent of a language, since it does not require understanding of the actual content presented in the textbooks. There are only two points where the system requires maintaining awareness of the text and the language: the detection of the TOC and the index.

Resolving the language dependency for TOC and index detection to create a hundred percent language independent system would be the next in line to improve this approach. The current system concerns itself with only high-level logical elements such as TOC, index, titles etc. to create the structure of the textbook. It is possible to provide coverage for lower level elements like definitions, examples, tables and lists to provide an even better enrichment for the extracted structure. At last, extending the scope of this system from textbooks to other well-authored educational resources like lecture slides would be another important direction to further development.

Citations

[1] Klink, S., Dengel, A. and Kieninger, T. (2000) Document Structure Analysis Based on Layout and Textual Features. In Proc. of the 4th International Workshop on Document Analysis Systems (pp. 99-111), IAPR .

[2] Thomas M. Duffy, and Robert Waller (1985) Designing Usable Texts (Chapter 8), Academic Press Inc.

[3] Sabine Schmid and Thierry Baccino (2003,) Perspective Shift and Text Format: An Eye-Tracking Study. Current Psychology Letters 9(3), (pp. 73-79).

[4] LiangcaiGao ,Zhi Tang , Xiaofan Lin , Ying Liu , RuihengQiu , and Yongtao Wang (2011), Structure extraction from PDF-based book documents, Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries (pp. 11-20).

[5] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye (2012), Probability & Statistics for Engineers & Scientists 9th Edition, Prentice Hall.

[6] Q. Luo, T. Watanabel and T. Nakayama (1996), Identifying Contents Page of Documents. In ICPR96 (pp. 696-700).

[7] S. Tsuruoka, C. Hirano, T. Yoshikawa and T. Shinogi (2001) Image-based Structure Analysis for a Table of Contents and Conversion to XML Documents. In DLIA.

[8] Bart, E., and Sarkar P. (2010), Information Extraction by Finding Repeated Structure. In Proc. of the 9th International Workshop on Document Analysis Systems (pp. 175–182).

[9] Gao, L.C., Tang, Z., Lin, X. F., Tao, X. and Chu, Y.M. (2009), Analysis of Book Documents' Table of Content Based on Clustering. In *Proc. of the 10th International Conference on Document Analysis and Recognition* (pp. 911–914).

[10] Z. Wu, P. Mitra, and C. L. Giles (2013), Table of contents recognition and extraction for heterogeneous book documents. in Proceedings of *International Conference on Document Analysis and Recognition* (pp. 1205–1209).

[11] Hassan, T. (2009), User-Guided Wrapping of PDF Documents Using Graph Matching Techniques. In Proc. of the 10th International Conference on Document Analysis and Recognition (pp. 631–635)

[12] Song Mao, Azriel Rosenfeld and Tapas Kanungo (2003), Document structure analysis algorithms: a literature survey, in *Proc. SPIE* 5010, Document Recognition and Retrieval X, 197.

[13]Anjewierden, A. (2001), AIDAS: Incremental Logical Structure Discovery in PDF Documents. In Proc. of the 6th International Conference on Document Analysis and Recognition (pp. 374–378).

[14] D´ejean, H. and Meunier, J. L. (2006), A System for Converting PDF Documents into Structured XML Format. In Proc. of the 7th International Workshop on Document Analysis Systems (pp. 129–140).

[15] Arasu, A. and Garcia-Molina H. (2003), Extracting structured data from web pages. SIGMOD (pp. 337-348)

[16] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo (2001), Roadrunner: Towards automatic data extraction from large web sites. In Proc. of the 2001 Intl. Conf. on Very Large Data Bases (pp. 109-118).

[17] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo (1997), Extracting semi-structured information from the Web. In Proc. of the Workshop on the Management of Semi-structured Data (pp. 18-25)

[18] Andrew Carlson and Charles Schafer (2008), Bootstrapping Information Extraction from Semi-structured Web Pages, In Proc. of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I (pp. 15-19).

[19] Hanno Walischewski (1997), Automatic Knowledge Acquisition for Spatial Document Interpretation, in Proc. of the 4th Intern. Conference on Document Analysis and Recognition (pp. 243-247).

[20] O. Altamura, F. Esposito, and D. Malerba (1999), WISDOM++: An Interactive and Adaptive Document Analysis System, in Proceedings of the 5th . Conference on Document Analysis and Recognition (pp. 366 - 369).

[21] Yuan Y. Tang, Chang D. Yan, and Ching Y. Suen (1994), Document Processing for Automatic Knowledge Acquisition, in Proc. IEEE Trans. on Knowledge and Data Engineering, Vol. 6,No.1 (pp. 3-21).

[22] He, F., Ding, X., and Peng, L. (2004), Hierarchical Logical Structure Extraction of Book Documents by Analyzing Tables of Contents. In Proc. of the International Conference on Document Recognition and Retrieval XI (pp. 6–13).

[23] Charu C. Aggarwal, and Cheng Xiang Zhai (2012), Mining Text Data, Springer Publishing Company Incorporated.

[24] Rafael C. Gonzalez, and Richard E. Woods (2006), Digital Image Processing (3rd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ.

[25] Nhung Do, J. Wenny Rahayu, and Torab Torabi (2012), Developments in Data Extraction, Management, and Analysis (1st Edition), IGI Publishing Hershey, PA, USA.

[26] Fabian M. Suchanek ,Gjergji Kasneci, and Gerhard Weikum (2007), Yago: a core of semantic knowledge, Proceedings of the 16th international conference on World Wide Web (pp. 697-706).

[27] Baccino, T. and Pynte, J. (1998). Spatial encoding and referential processing during reading. European Psychologist, 3/1 (pp. 51-61).

[28] T. Igarashi, S. Matsuoka, T. Masui, and V. Haarslev (1995), Adaptive recognition of implicit structures in human-organized layouts, 11th IEEE Int. Symp. Visual Languages (pp. 258 -266).

[29] Y. Velenik (2012), Probabilités et Statistique.

[30] Ludwig Fahrmeir, Rita Künstler, Iris Pigeot, and Gerhard Tutz (2007), Statistik, Berlin/Heidelberg, Germany: Springer.

[31] S. Mandal, S.P. Chowdhury, A.K. Das and B. Chanda (2003), Automated Detection and Segmentation of Table of Contents Page from Document Images. In Proc. 7th International Conference on Document Analysis and Recognition (pp. 398-402).

[32] Antoine Isaac, and Ed Summers (2009), SKOS Simple Knowledge Organization System Reference, W3C Recommendation.