Fachbeitrag

Thomas Bähr und Merle Friedrichsen

Konvertierung von PDF in XML für die Langzeitarchivierung und Weiterverarbeitung

DOI 10.1515/abitech-2017-0004

Zusammenfassung: In der Darstellung, Weitergabe und Aufbewahrung elektronischer Publikationen steht das Format PDF unangefochten an erster Stelle. Die Stärken des ISO-standardisierten Formats liegen in der Plattform- und Hardwareunabhängigkeit, in der seitengenauen Darstellung von Publikationen sowie in der einfachen Navigierbarkeit von komplexen Dokumenten. Dank der stetigen Weiterentwicklung des Formats existiert mittlerweile eine große Anzahl an PDF Profilen wie PDF/A, PDF/X, PDF/UA oder PDF/E.

Eine flexiblere Dokumentendarstellung ermöglicht hingegen die eXtensible Markup Language XML, welche nicht nur im Web, sondern auch vermehrt in der Druckvorstufe eingesetzt wird. Wie PDF ist auch XML medienneutral und plattformunabhängig. Im Gegensatz zu PDF-Dokumenten erlaubt XML hingegen mittels Erfassung der Inhalte in einer dokumentierten und transparenten Struktur eine Validierung der Inhalte wie auch eine gezielte Weiternutzung einzelner Teilinhalte.

Die Technische Informationsbibliothek (TIB) führte eine Analyse zur Machbarkeit einer PDF-nach-XML-Konvertierung durch. Ziel ist die Vorhaltung von XML-Dokumenten für zwei Prozesse: Erstens zur automatischen Katalogisierung von Kongressbänden auf Aufsatzebene, zweitens zur Aufbewahrung einer parallelen Repräsentation neben PDF-Dokumenten im Langzeitarchiv. Dieser Artikel stellt die Ergebnisse der Machbarkeitsstudie dar.

Schlüsselwörter: Strukturanalyse, Dateiformatkonvertierung, automatische Layouterkennung

Conversion of PDF to XML for preservation and usage

Abstract: PDF is without a doubt the most common file format choice when it comes to presenting, sharing and preserving electronic publications. The strengths of the ISO-standardized format lie in its independent platform and hardware, its page-exact rendering of publications as well as its smooth navigation of complex documents. Due to the ever-growing requirements of the community, a number of profiles for the file format exist today, such as: PDF/A, PDF/X, PDF/UA or PDF/E.

The eXtensible Markup Language XML, on the other hand, allows for more flexible handling of document display, leading to a high adoption of the format not only in the web but also in printing and publishing processes. Like PDF, XML is media-neutral and platform-independent. Contrary to PDF, XML makes use of a transparent and well-documented content structure, allowing for validation processes as well as for extraction processes targeting specific content parts.

TIB (the Technische Informationsbibliothek) conducted a proof-of-concept study on PDF to XML conversion. The study's background is the usage of XML as a second representation of the original PDF content in the digital archive. This article presents the outcome of the proof-of-concept.

Keywords: Strucutral Analysis, File Format Conversion, Automatic Layout recognition

1 State of the Art

„The loss of structural information in PDF files is so massive that its wide use for archival purposes is nothing less than troubling."[1] Vor diesem Hintergrund sollen in diesem Projekt die folgenden Fragen beantwortet werden: Welche Tools gibt es zur Strukturerkennung von wissenschaftlichen Publikationen, die PDF als Ausgangsformat verarbeiten können? Wie gut ist die Performanz der ausgewählten Tools? Ist das so erzeugte XML-Format eine sinnvolle Ergänzung zur Langzeitarchivierung von wissenschaftlichen Publikationen?

Um die Arbeitsweise der Tools zu verstehen, wird zunächst ein kurzer Überblick über den Aufbau von PDF-Dateien gegeben. Im darauffolgenden Teil wird die Literatur zur Strukturanalyse von wissenschaftlichen Dokumenten

1 Berg, Øyvind Raddum: *High precision text extraction from PDF documents*. Oslo, 2011. 81.

ausgewertet, anschließend werden einzelne Tools oder Studien vorgestellt.

1.1 Aufbau von PDF-Dateien

Der Aufbau einer PDF-Datei soll hier nur überblicksartig beschrieben werden. Neben dem File-Header, der beschreibt, welche PDF-Spezifikation angewandt wurde, und Elementen, die den Zugriff auf die Datei erleichtern (Trailer, Cross-Reference-Table), stehen die wichtigsten Informationen im File-Body, auch Object-Stream[2] genannt. In Abbildung 1 ist der Object-Stream dargestellt.

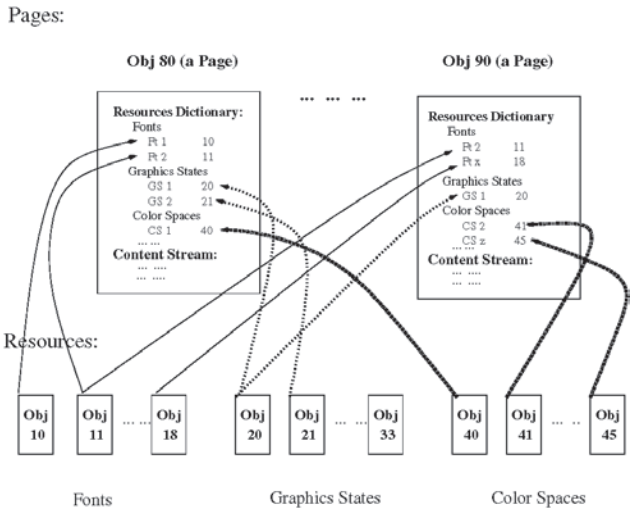


Abb. 1: Aufbau einer PDF-Datei (Shao, Mingyan; Futrelle, Robert P.: „Recognition and Classification of Figures in PDF Documents." In: *Graphics Recognition Ten Years Review and Future Perspectives*. Hrsg. von David Hutchison, Takeo Kanade. 235. Berlin, Heidelberg: Springer, 2006.)

Jedes Objekt (Seite, Schriftart, Schriftgröße etc.) erhält eine Objektnummer[3]. Anhand der Nummer kann ein Objekt referenziert und mehrfach verwendet werden[4].

PDFs können in Abhängigkeit davon, mit welchen Tools und in welcher Art und Weise sie erstellt wurden, sehr unterschiedlich aufgebaut sein. Ein Dokument, welches mit Word erzeugt und in PDF umgewandelt wurde und z. B. Informationen zur verwendeten Schriftart enthält, unterscheidet sich wesentlich von dem PDF, welches entsteht, wenn ebenjenes Dokument ausgedruckt, eingescannt, in PDF umgewandelt und mit Texterkennung versehen wurde. Informationen wie die ursprüngliche Schriftart (und viele weitere) sind dann nicht mehr rekonstruierbar.

1.2 Strukturanalyse bei PDFs in der Literatur

Es gibt mehrere Studien, die sich mit der Strukturanalyse von wissenschaftlichen Dokumenten beschäftigen. Diese beschreiben jeweils (1) die Zielsetzung, (2) das technische Vorgehen der eingesetzten Tools bei der Analyse, (3) die Performanz der Tools und (4) einen Ausblick auf die weitere Entwicklung oder Einbindung in bestehende Abläufe oder Software.

Viele Tools erkennen das physische Layout (geometrische Struktur/Layout) wie Seiten, Zeilen, Worte; andere Tools unterteilen auch in das logische Layout (logische Struktur) wie Autor, Abstract, Quellenangaben[5]. Auf diesen Tools liegt der Fokus dieses Projektes.

Die technischen Ansätze der Tools bei der Strukturanalyse sind verschieden und reichen von regelbasierten Verfahren[6] bis hin zu unüberwachten Machine-Learning-Techniken.[7] Für die Strukturerkennung wird auch das Ausgangsmaterial verschieden betrachtet. Bei einigen Tools wird das physische Layout mittels Optical Character Recognition (OCR) analysiert.[8] Bei anderen Tools werden die in der PDF-Datei hinterlegten Objekte genutzt,[9] so dass

2 Schneeberger, Hans Peter: *PDF in der Druckvorstufe: PDF-Dateien erstellen, prüfen, korrigieren, automatisieren und ausgeben*. Bonn, 2014. 179.

3 Shao, Mingyan, Robert P. Futrelle: „Recognition and Classification of Figures in PDF Documents." In: *Graphics Recognition Ten Years Review and Future Perspectives*. Hrsg. von David Hutchison, Takeo Kanade. Berlin, Heidelberg: Springer, 2006. 235.

4 Schneeberger 2014: 177.

5 Klampfl, Stefan, Michael Granitzer, Kris Jack: „Unsupervised document structure analysis of digital scientific articles." In: *International Journal on Digital Libraries* 14,3-4 (2014): 84; und Constantin, Alexandru, Steve Pettifer, Andrei Voronkov: „PDFX. Fully-automated PDF-to-XML Conversion of Scientific Literature." In: *DocEng'13 – Proceedings of the 2013 ACM symposium on Document engineering*. Florence 2013. 177.

6 Ramakrishnan, Cartic, Abhishek Patnia, Eduard Hovy: „Layout-aware text extraction from full-text PDF of scientific articles." In: *Source code for biology and medicine* 7,1 (2012): 7.

7 Klampfl et al. 2014: 83

8 Luong, Minh-Thang, Thuy Dung Nguyen, Min-Yen Kan: „Logical Structure Recovery in Scholarly Articles with Rich Document Features." In: *International Journal of Digital Library Systems* 1,4 (2010): 2; und Klampfl, Stefan, Roman Kern: „Machine Learning Techniques for Automatically Extracting Contextual Information from Scientific Publications." In: *Semantic Web Evaluation Challenges*. Hrsg. von Fabien Gandon, Elena Cabrio, Milan Stankovic. 108. Cham 2015.

9 Berg, Øyvind Raddum, Stephan Oepen, Jonathon Read: „Towards High-quality Text Stream Extraction from PDF: Technical Background to the ACL 2012 Contributed Task." In: *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. 99. Stroudsburg, PA, 2012; und Mohemad, Rosmayati, Abdul Razak

sich diese Tools nur bei born-digital PDFs anwenden lassen.[10]

In den Studien wird die Performanz der Tools evaluiert und teilweise mit anderen Tools verglichen. Bei einigen Studien werden dazu dieselben Testsets verwendet oder die eigenen zu Vergleichszwecken zur Verfügung gestellt.[11] In der Literatur finden sich dazu zwei Ansätze. Es kann ein Testset genutzt werden, welches bereits in beiden Formaten, PDF und XML, vorliegt, womit ein automatisierter Abgleich möglich ist.[12] Verschiedene Algorithmen, wie die „Ratcliff/Obershelp string comparison method“[13] oder der „Needleman-Wunsch algorithm“[14] werden dabei genutzt. Die zweite Möglichkeit besteht darin, das Testset manuell zu analysieren und mit dem Output der Tools zu vergleichen.[15] Dieser Aufwand schränkt allerdings den Umfang des Testsets ein. Ob die Erkennung der Strukturelemente erfolgreich ist, wird über Precision und Recall gemessen. Die Maßeinheit F1 setzt diese beiden Werte in Beziehung zueinander und lässt so einen einfachen Vergleich zu.[16]

Der Output wird häufig als Basis für die weitere Analyse genutzt, ist aber mit Fehlern behaftet, die die Gesamtperformanz beeinflussen.[17] Als mögliche Verbesserung wird in einigen Quellen vorgeschlagen, im Output der Tools Position und Layout jedes Buchstabens auszugeben. Ein weiterer Vorschlag besteht darin, die Informationen aus der OCR um die Informationen aus den eingebetteten Objekten in born-digital PDFs zu ergänzen, um die verschiedenen technischen Ansätze zu verbinden.[18]

Fehler bei der Texterkennung und -zuordnung zu Abschnitten treten häufig bei Formeln und grafischen Darstellungen auf.[19] Hier ist eine Weiterentwicklung oder Integration von bestehenden Tools[20] wünschenswert, insbesondere für Texte im naturwissenschaftlichen Bereich.

Klampfl stellt fest, dass es bei Tools, die überwachtes Machine Learning verwenden, zu Performanz-Problemen kommen kann. Daher verwenden die Autoren dieser Studie unüberwachte Machine-Learning-Techniken, die nicht für jedes Datenset neu trainiert werden müssen, sondern bei jeder einzelnen Datei dazulernen. Diese Tools werden mit einem Testset trainiert. Bei PDFs, die sich im Aufbau von denen des Testsets unterscheiden, kann es zu schlechteren Ergebnissen kommen.[21]

2 Auswahl der Tools

Insgesamt wurden sieben Tools auf ihre Eignung für diese Untersuchung analysiert, wovon zwei ausgewählt werden konnten.

Genutzt wurde PDFX, da es ein leicht zu bedienendes Online-Tool ist. Eine Einschränkung im Hinblick auf die zu verarbeitenden Dateien ergibt sich, da das Tool nur online genutzt werden kann. Aus rechtlichen Gründen können daher nur Testdateien, die veröffentlicht sind, verwendet werden. Laut Beschreibung sollen viele Strukturelemente erkannt und ein XML-Output (nicht nur eine Ansicht der Ergebnisse) generiert werden können.

Als weiteres Tool wurde Adobe Pro getestet. Da Adobe zu den Entwicklern des PDF-Standards gehört, wird erwartet, dass das Format PDF gut verstanden wird. Insbesondere bei born-digital PDFs sollte eine Umsetzung in XML gut möglich sein.

Andere Tools wurden aus folgenden Gründen nicht verwendet:

- PDFExtract: das Tool wurde seit 2011 nicht weiterentwickelt bzw. angepasst und konnte nicht installiert werden.
- ParsCit / SectLabel: die Installation ist im Rahmen des Projektes zu aufwendig („we expect a UNIX-knowledgeable person to be able to get the installation working within 30 minutes to an hour“[22]).
- LA-PDFText: bei der Installation ist es zu Java-Fehlern gekommen, so dass die Installation nicht erfolgreich abgeschlossen werden konnte. Zusätzlich müssen bei der Anwendung von LA-PDFText eigene Regeln zur Verarbeitung erstellt werden, welches den Umfang des Projektes überschreitet.[23]
- Code-Annotator: die Demoversion, die online zur Verfügung steht, ist in ihrer Handhabung auf das Anno-

Hamdan, Zulaiha Ali Othman: „Automatic Recognition of Document Structure from PDF Files.“ In: *Software Engineering and Computer Systems*. Hrsg. von Jasni Mohamad Zain, Wan Maseri bt Wan Mohd, Eyas El-Qawasmeh. 280. Berlin, Heidelberg: Springer, 2011; und Constantin 2013: 177.

10 In dieser Untersuchung sollen der Umfang und die Qualität des Outputs und nicht die technische Vorgehensweise der Tools näher betrachtet werden.

11 Vgl. Constantin et al. 2013.

12 Ramakrishnan 2012: 4–5.

13 Constantin 2013: 179.

14 Ramakrishnan 2012: 5.

15 Luong 2010: 13; und Ramakrishnan 2012: 4; und Berg 2011: 77.

16 Ramakrishnan 2012: 4.

17 Klampfl 2014: 98.

18 Berg 2012: 103.

19 Berg 2011: 82.

20 Vgl. Lin, Xiaoyan, Liangcai Gao, Zhi Tang: “Mathematical Formula Identification in PDF Documents.” In: *Eleventh International Conference on Document Analysis and Recognition*. Beijing 2011.

21 Klampfl 2014: 98.

22 https://github.com/knmnyn/ParsCit (23.1.2017).

23 Ramakrishnan 2012: 6.

tieren und Anwendung von Taxonomien ausgerichtet. Die Ergebnisse der Strukturanalyse können in dieser Oberfläche nicht als XML-Output ausgegeben werden.
– System, vorgeschlagen von Stoffel, Spretke, Kinnemann und Keim:[24] wurde zwar patentiert, der Patentschutz ist jedoch ausgelaufen. Nach jetzigem Stand der Recherche wurde das System in keinem verfügbaren Tool angewandt.

3 Strukturelemente – Anforderungen aus der Langzeitarchivierung

Die Langzeitarchivierung stellt besondere Anforderungen an die Erkennung von Strukturelementen. Diese werden eingeteilt in Strukturelemente, die zwingend erkannt werden müssen (Mindestanforderung), und jene, die einen zusätzlichen Wert / Nutzen schaffen und somit die Anforderungen sehr gut erfüllen. Einige Elemente wie die Reading-Order können allerdings nur richtig oder falsch erkannt werden. In der Tabelle 1 ist eine Übersicht der Elemente und Anforderungen dargestellt, die im Folgenden erläutert werden.

Eine Mindestanforderung ist die korrekte Erfassung der Reading-Order (Lese-Reihenfolge). Eine Voraussetzung dafür ist die Erkennung des physischen Layouts eines Dokuments. Von Tools können u. a. Seiten, Worte (und deren Position), Schriftarten, Bilder, Spalten, Textblöcke etc. erkannt werden.[25] Gerade bei mehrspaltigen Publikationen, wie sie häufig bei Zeitschriften vorkommen, muss das Tool erkennen können, dass nicht Zeile für Zeile gelesen wird, sondern Spalte für Spalte.

Bei der Texterkennung gibt es zwei Mindestanforderungen. Die Textformatierung (z. B. Hochstellung zum

24 Stoffel, Andreas, David Spretke, Henrik Kinnemann, Daniel A. Keim: "Enhancing document structure analysis using visual analytics." In *Proceedings of the 2010 Symposium on Applied Computing*, March 22–28, 2010, Sierre, Switzerland. New York 2010.

25 Constantin 2013.

Tab. 1: Strukturelemente und die Anforderung aus der LZA

Strukturelement	Mindestanforderung	Anforderungen sehr gut erfüllt	Als Fehler gewertet
Reading-Order	Reading-Order korrekt erkannt		Falsche Reading-Order (durch Spalten)
Formatierung	Formatierung erkannt (Hochstellung, Unterstreichen …)		Formatierung nicht übernommen
Sonderzeichen im Fließtext	Sonderzeichen korrekt wiedergegeben		Sonderzeichen nicht erkannt
Überschrift	Überschrift im Fließtext	Überschriften als Überschrift in XML dargestellt	Text nicht erkannt/Text an falscher Stelle
Kopfzeile/Fußzeile	Kopfzeile/Fußzeile im Fließtext	Kopfzeile/Fußzeile als Kopfzeile/Fußzeile in XML dargestellt oder Kopfzeile/Fußzeile wird nicht wiederholt	Text nicht erkannt/Text an falscher Stelle
Fußnote	Fußnote im Fließtext	Fußnote als Fußnote in XML dargestellt	Text nicht erkannt/Text an falscher Stelle
Seitenumbruch und Seitenzahl	Seitenzahl als Ziffer im Text, Seitenumbruch in XML dargestellt	Seitenzahl als Seitenzahl in XML dargestellt, Seitenumbruch in XML dargestellt	Seitenumbruch nicht erkannt
Tabelle	Als Bild abgelegt und in XML referenziert	Als Tabelle in XML dargestellt (dadurch durchsuchbar)	Tabelle nicht erkannt, Tabelle nicht vollständig erkannt
Formel (nicht im Fließtext)	Als Bild abgelegt und in XML referenziert oder als Fließtext lesbar abgelegt	Als Formel in XML dargestellt (dadurch durchsuchbar)	Formel nicht erkannt, Formel nicht vollständig erkannt
Bild, graphische Darstellung	Als Bild abgelegt und in XML referenziert		Bild nicht erkannt, Bild nicht vollständig erkannt (oder geteilt in mehrere Bilder)

korrekten Identifizieren von Fußnoten, kursive Textpassagen, die Zitate darstellen können, etc.) und Sonderzeichen müssen korrekt erkannt werden. Als Sonderzeichen werden alle Zeichen gewertet, die nicht im lateinischen Alphabet vorkommen und im Fließtext stehen. Sobald ein Sonderzeichen auf der Seite nicht erkannt wurde, wurde es als Fehler gewertet, egal wie viele Sonderzeichen richtig erkannt wurden. Der Unterschied zu Formeln wurde anhand des Layouts definiert: Eine Formel hat in der Regel einen eigenen Absatz.

Wenn eine Überschrift erkannt wurde und in XML als Überschrift ausgezeichnet ist, sind die Anforderungen sehr gut erfüllt. Ebenso trifft dies auf die Erkennung und Darstellung von Kopfzeilen, Fußzeilen und Fußnoten zu.

Die Erkennung von Seitenumbrüchen und Seitenzahlen ist eine weitere Mindestanforderung. Dies ist besonders zum Zitieren und Nachverfolgen von Zitaten im wissenschaftlichen Kontext unabdingbar. Die Anforderung gilt als sehr gut erfüllt, wenn der Seitenumbruch an der richtigen Stelle und die Seitenzahl im XML als solche gekennzeichnet ist. Die Mindestanforderungen gelten als erfüllt, wenn die Seitenzahl als Text in XML abgebildet und der Seitenumbruch an der richtigen Stelle in XML dargestellt ist. Ist die Seitenzahl nur als Ziffer angegeben, ist die Mindestanforderung nicht erfüllt, da nicht eindeutig ist, ob die Zahl eine Seitenzahl ist oder eine andere Funktion erfüllt. Als Fehler zählt auch, wenn der Seitenumbruch nicht erkannt wurde.

Für die Strukturelemente Tabellen und Formeln gelten ähnliche Mindestanforderungen. Sie gelten als erfüllt, wenn die Tabelle oder Formel als Bild abgelegt und in XML referenziert wurde. Die Anforderung gilt als sehr gut erfüllt, wenn die Tabelle als Tabelle in XML dargestellt und die Formel als Formel abgelegt wurde. Eine Tabelle, die nur als Fließtext dargestellt ist, wurde als Fehler gewertet, da im Nachhinein nicht deutlich ist, wo die Trennung zwischen Spalten und Zeilen besteht. Für Formeln ist ein bestimmtes XML-Schema nötig, für die Darstellung kann beispielsweise MathML (vom W3C[26]) genutzt werden. Als Fehler wurde gewertet, wenn die Tabelle oder Formel nicht (vollständig) erkannt wurde.

Besondere Schwierigkeiten stellen grafische Darstellung oder sonstige Abbildungen dar. Diese können als Bild gespeichert und in XML referenziert werden. Dies ist die Mindestanforderung. Der Vorteil besteht darin, dass für die Langzeitarchivierung von Bildern Formate und Best-Practices zur Verfügung stehen. Wenn die getesteten Tools dies nicht vermögen, gibt es andere Tools, die Bilder aus PDFs herauslösen können. Wichtig ist hierbei, dass die Bilder an der richtigen Stelle im Textverlauf referenziert werden, bzw. im XML darauf verwiesen wird.

4 Erstellung des Testsets

Die Vielfalt der PDF-Versionen[27] kann in diesem Projekt nicht widergespiegelt werden. Es wurde aber versucht, das Testset in Hinblick auf die Versionen möglichst heterogen zu gestalten. Um die Heterogenität des Ausgangsmaterials abzubilden, werden für das Testset je 50 PDFs aus verschiedenen Quellen zusammengestellt. Da auch Online-Tools genutzt werden, müssen alle verwendeten PDFs bereits veröffentlicht sein.

Testset (A) Dissertationen

Es wurde ein Testset von 50 Dissertationen unterschiedlicher Disziplinen ausgewählt. Die PDFs wurden von unterschiedlichen Programmen erstellt und enthalten unterschiedliche PDF-Versionen.

Testset (B) Verlags-PDFs

An diesem Testset[28] lässt sich gut ablesen, welche anderen Faktoren sich auf die Ergebnisse der Tools auswirken, da alle Dateien dieselbe Version aufweisen. Testset (B) bildet daher eine gute Ergänzung zu Testset (A).

Für eine bessere Übersicht über das Testset wurde eine Liste mit Dateiname, PUID der PDF-Version, Seitenanzahl und Seite für manuelle Auswertung angelegt. Später wurde die Liste um die Ergebnisse erweitert.

5 Durchführung und Auswertung

Für jede PDF im Testset wurde eine Seite mit möglichst vielen Strukturelementen zur manuellen Auswertung ausgewählt. Zum visuellen Vergleich mit der XML-Datei wurde diese Seite ausgedruckt und die Strukturelemente wurden markiert.

26 W3C Math Home. http://www.w3.org/Math/ (19.1.2016).

27 Die für die Formatvalidierung genutzte Datenbank PRONOM führt 37 verschiedene PDF-Versionen auf und kennzeichnet diese mit dem PRONOM Unique Identifier (PUID). http://www.nationalarchives.gov.uk/PRONOM/ (23.1.2017)

28 Es werden PDFs vom Hindawi-Verlag (*Journal of Mathematics und Mathematical Problems in Engineering*) genutzt, da diese unter einer CC-Lizenz veröffentlicht sind.

Für das Tool PDFX mussten alle PDFs aus dem Testset (A) und zwei aus dem Testset (B) gekürzt werden, da nur 5 MB Upload und nicht mehr als 100 Seiten zugelassen sind. Gekürzt wurde mit dem Befehl „pdftk“ auf je 11 Seiten, 5 vor und 5 nach der zur manuellen Auswertung ausgewählten Seite. Für die Umwandlung mit Adobe wurden die Originale (ungekürzt) verwendet.

Das Testset wurde mit dem ausgewählten Tool in XML umgewandelt und die bei der Umwandlung ausgegebenen Fehlermeldungen wurden festgehalten. Anschließend wurde das Ergebnis manuell überprüft um festzustellen, ob ein Strukturelement falsch ist, den Mindestanforderungen entspricht oder sehr gut erkannt wurde.

Für die Auswertung wurde der Prozentsatz der Strukturelemente errechnet, die den Mindestanforderungen entsprechen, sowie der Prozentsatz aller Strukturelemente, die die Anforderungen sehr gut erfüllen:

- Min: Anzahl der Strukturelemente, die nur die Mindestanforderungen erfüllen
- Max: Anzahl der Strukturelemente, die die Anforderungen sehr gut erfüllen
- Original: Anzahl der Strukturelemente im Original

In der Auswertung wurde wie folgt vorgegangen:

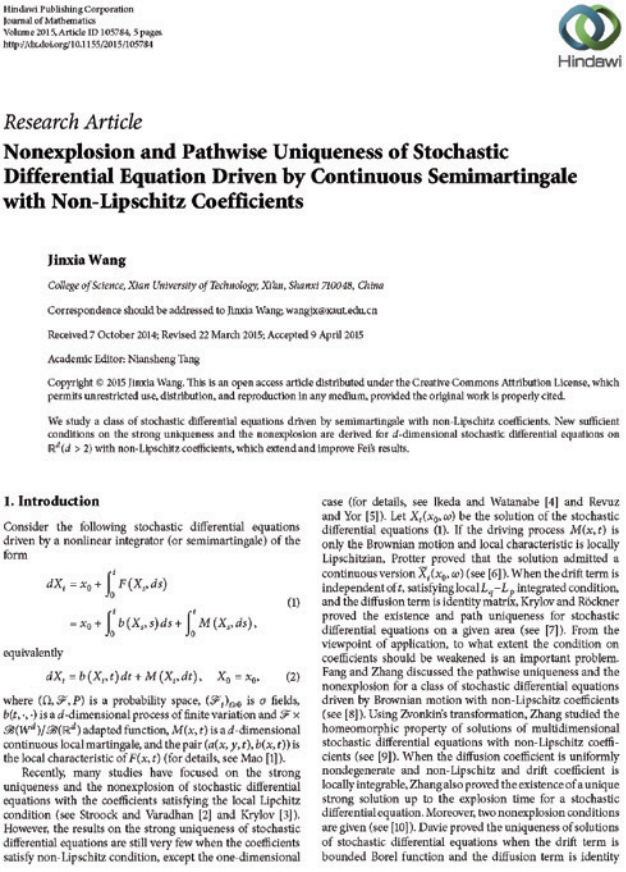
$$\frac{(\text{Min}+\text{Max})}{\text{Original}} = \text{Mindestanforderungen erfüllt}$$

$$\frac{\text{Max}}{\text{Original}} = \text{Anforderungen sehr gut erfüllt}$$

Mit der Angabe in Prozent lässt sich auf einen Blick vergleichen, welches Tool bei welchen Strukturelementen zu besseren Ergebnissen führt.

6 Ergebnisse

In diesem Kapitel wird der Output der Tools beispielhaft vorgestellt und analysiert. Dabei wird ausschnittsweise der Output derselben PDF gezeigt (siehe Abbildung 2). Anschließend werden die Ergebnisse verglichen.



Hindawi Publishing Corporation
Journal of Mathematics
Volume 2015, Article ID 105784, 5 pages
http://dx.doi.org/10.1155/2015/105784

Hindawi

Research Article

Nonexplosion and Pathwise Uniqueness of Stochastic Differential Equation Driven by Continuous Semimartingale with Non-Lipschitz Coefficients

Jinxia Wang

College of Science, Xian University of Technology, Xi'an, Shanxi 710048, China

Correspondence should be addressed to Jinxia Wang; wangjx@xaut.edu.cn

Received 7 October 2014; Revised 22 March 2015; Accepted 9 April 2015

Academic Editor: Nansheng Tang

We study a class of stochastic differential equations driven by semimartingale with non-Lipschitz coefficients. New sufficient conditions on the strong uniqueness and the nonexplosion are derived for d-dimensional stochastic differential equations on $\mathbb{R}^d (d > 2)$ with non-Lipschitz coefficients, which extend and improve Fei's results.

1. Introduction

Consider the following stochastic differential equations driven by a nonlinear integrator (or semimartingale) of the form

$$dX_t = x_0 + \int_0^t F(X_s, ds) = x_0 + \int_0^t b(X_s, s)\,ds + \int_0^t M(X_s, ds), \quad (1)$$

equivalently

$$dX_t = b(X_t, t)\,dt + M(X_t, dt), \quad X_0 = x_0. \quad (2)$$

where $(\Omega, \mathcal{F}, P)$ is a probability space, $(\mathcal{F}_t)_{t\geq 0}$ is σ fields, $b(t, \cdot, \cdot)$ is a d-dimensional process of finite variation and $\mathcal{F} \times \mathcal{B}(W^d)/\mathcal{B}(\mathbb{R}^d)$ adapted function, $M(x, t)$ is a d-dimensional continuous local martingale, and the pair $(a(x, y, t), b(x, t))$ is the local characteristic of $F(x, t)$ (for details, see Mao [1]).

Recently, many studies have focused on the strong uniqueness and the nonexplosion of stochastic differential equations with the coefficients satisfying the local Lipchitz condition (see Stroock and Varadhan [2] and Krylov [3]). However, the results on the strong uniqueness of stochastic differential equations are still very few when the coefficients satisfy non-Lipschitz condition, except the one-dimensional case (for details, see Ikeda and Watanabe [4] and Revuz and Yor [5]). Let $X_t(x_0, \omega)$ be the solution of the stochastic differential equations (1). If the driving process $M(x, t)$ is only the Brownian motion and local characteristic is locally Lipschitzian, Protter proved that the solution admitted a continuous version $\overline{X}_t(x_0, \omega)$ (see [6]). When the drift term is independent of t, satisfying local L_q–L_p integrated condition, and the diffusion term is identity matrix, Krylov and Röckner proved the existence and path uniqueness for stochastic differential equations on a given area (see [7]). From the viewpoint of application, to what extent the condition on coefficients should be weakened is an important problem. Fang and Zhang discussed the pathwise uniqueness and the nonexplosion for a class of stochastic differential equations driven by Brownian motion with non-Lipschitz coefficients (see [8]). Using Zvonkin's transformation, Zhang studied the homeomorphic property of solutions of multidimensional stochastic differential equations with non-Lipschitz coefficients (see [9]). When the diffusion coefficient is uniformly nondegenerate and non-Lipschitz and drift coefficient is locally integrable, Zhang also proved the existence of a unique strong solution up to the explosion time for a stochastic differential equation. Moreover, two nonexplosion conditions are given (see [10]). Duvie proved the uniqueness of solutions of stochastic differential equations when the drift term is bounded Borel function and the diffusion term is identity

Abb. 2: Seite zur Auswertung aus Testset B

6.1 Ergebnisse Adobe

In Abbildung 3 sieht man einen Auszug aus dem XML-Output von Adobe. Man sieht in den untersten Zeilen, dass jeweils auf ein Bild verwiesen wird und im Anschluss der erkannte Text ohne Leerzeichen wiedergegeben wird. In dem Tag <x:xmpmeta> sind Metadaten wie das Entstehungsdatum des PDFs oder die erstellende Software enthalten.

Abb. 3: XML-Output von Adobe

Abb. 4: XML-Output von PDFX

In einem Fall sind Quellen, die im Text unterstrichen waren, in XML als Referenzen gekennzeichnet worden.

Aufgefallen ist, dass bei allen Dokumenten aus Testset (B) und bei drei Dokumenten aus Testset (A) die Leerzeichen im Text fehlen. Dies behindert die Les- und Interpretierbarkeit, sowohl für Mensch als auch für Maschine. Dieser Fehler ist nur bei Adobe aufgetreten. Um dennoch einen Vergleich durchführen zu können, wurde bei betroffenen Dokumenten trotzdem die XML-Datei auf die Strukturelemente hin untersucht.

Bei den Dissertationen konnte eine Datei nicht umgewandelt werden. Drei weitere wurden nicht vollständig umgewandelt. Die Erkennung der Strukturelemente wurde somit als Fehler gewertet. Da verschiedene PDF-Versionen davon betroffen waren, sind keine Rückschlüsse auf eine bestimmte Version möglich.

Es ist aufgefallen, dass bei dem Testset (A) die Kopf- und Fußzeilen nicht in XML abgebildet wurden. Die Anforderung gilt daher als nicht erfüllt.

6.2 Ergebnisse PDFX

Der Output von PDFX für dasselbe PDF ist umfangreicher annotiert (siehe Abbildung 4). So sind nicht nur fortlaufende IDs für die einzelnen Objekte vergeben worden, sondern auch die Angabe, auf welcher Seite (fortlaufend gezählt, unabhängig von der Original-Seitenzählung) sich das Objekt befindet (bei 92 von 100 insgesamt). Hinzu kommen Angaben wie type="page_nr", die die Original-Seitenzählung enthalten.

Durch die fortlaufende Zählung der Seiten ist eine eindeutige Zuordnung der Strukturelemente zu einer Seite möglich. Daher werden Strukturelemente in solchen Dokumenten als richtig gewertet, selbst wenn das Objekt nicht an der erwarteten Stelle steht. Bei einigen Dokumenten wurden die Seiten nicht fortlaufend gezählt. Die Formatversion konnte als Ursache ausgeschlossen werden, da verschiedene Versionen betroffen waren. Nicht ausgeschlossen werden konnte, dass es an Einstellungen innerhalb des PDFs liegt oder an der Software, mit der das PDF erstellt wurde.

Die Erkennung von Überschriften (und Abschnittsüberschriften) und deren Kennzeichnung in XML erfüllen die Anforderungen sehr gut. Gleiches gilt für die Erkennung und Darstellung von Kopf- und Fußzeilen und Fußnoten.

Aufgefallen ist, dass bei Testset (A) zwei Dokumente nicht vollständig umgewandelt wurden. Daher werden die enthaltenen Strukturelemente als Fehler gewertet. Beim Testset (B) traten allerdings nicht dieselben Fehler wie bei Adobe auf.

Bei einem PDF waren Quellen gekennzeichnet. Im Output von Adobe waren diese markiert, beim PDFX-Output ist diese Information nicht in den Strukturelementen abgebildet.

Als Fehler wurde gewertet, wenn eine Überschrift zwar als Header in XML dargestellt wurde, es aber im Fließtext keinen Hinweis darauf gab, an welcher Stelle die Überschrift eingebunden werden soll.

Zweimal sind zwar Bild und Bildunterschrift korrekt erkannt worden, der Fließtext jedoch nicht.

Werden Formeln erkannt, werden diese im XML in eine Formular-Box gebettet. Allerdings wird auch in dieser Formular-Box nicht dargestellt, ob ein Bruch vorhanden ist oder etwas hoch- oder tiefgestellt ist. Die Anforderungen gelten nur dann als sehr gut erfüllt, wenn die Formel korrekt dargestellt ist.

6.3 Ergebnisse in der Erkennung der Strukturelemente im Vergleich

Um die Ergebnisse der Untersuchung besser vergleichen zu können, wurde in der folgenden Tabelle dargestellt, ob und zu welchem Anteil die Strukturelemente erkannt wurden.

Aufgrund der geringen Datenmenge in der Untersuchung lässt sich zur Erkennung der Strukturelemente nur eine generelle Tendenz feststellen. Die Reading-Order wird grundsätzlich gut erkannt (wobei man beachten muss, dass bei Adobe das gesamte Testset (B) keine Leerzeichen enthält). Auch die Erkennung von Überschriften und Kopf- und Fußzeilen entspricht in den meisten Fällen den Erwartungen. Beide Tools haben zudem die Bilder, die erkannt wurden, in Ordner abgelegt und im XML darauf referenziert.

Sonderzeichen wurden dann als richtig gewertet, wenn alle Sonderzeichen im Test richtig erkannt wurden, und als fehlerhaft, sobald ein Sonderzeichen nicht richtig erkannt wurde.

Viele Seiten aus dem Testset (B) enthalten nur das © und das @ als Sonderzeichen, welche mit beiden Tools richtig erkannt und somit gewertet wurden. Dies führt allerdings zu einem Ungleichgewicht in der Bewertung und zu dem hohen Anteil an richtig erkannten Sonderzeichen (46 Prozent bei Adobe, 64 Prozent bei PDFX).

Die Hoch- und Tiefstellung ist im Fachbereich Technik und Naturwissenschaften der TIB von besonderer Relevanz. Bei beiden Tools wurden die Formatierung, Sonderzeichen (außer © und @) und die Formeln nicht ausreichend erkannt. Nur in einem Fall wurde die Hochstellung von Fußnoten mit Adobe erkannt.

Tabellen im Text werden von Adobe besser erkannt. Bei PDFX waren hingegen Seitenumbruch und Seitenzahl

Tab. 2: Ergebnisse im Vergleich

Strukturelement	Adobe				PDFX				Im Original vorhanden
	Mindestanforderung erfüllt		Anforderungen sehr gut erfüllt		Mindestanforderung erfüllt		Anforderungen sehr gut erfüllt		
	Angaben in Prozent (absolut)								absolut
Reading-Order	96 Prozent		(51)		83 Prozent		(44)		53
Formatierung	1 Prozent		(1)		0 Prozent		(0)		88
Sonderzeichen im Fließtext	46 Prozent		(35)		54 Prozent		(41)		76
Überschrift	97 Prozent	(159)	20 Prozent	(32)	89 Prozent	(146)	69 Prozent	(113)	164
Kopfzeile/Fußzeile	61 Prozent	(55)	0 Prozent	(0)	92 Prozent	(83)	39 Prozent	(35)	90
Fußnote	57 Prozent	(13)	4 Prozent	(1)	61 Prozent	(14)	30 Prozent	(7)	23
Seitenumbruch und Seitenzahl	49 Prozent	(49)	0 Prozent	(0)	91 Prozent	(91)	81 Prozent	(81)	100
Tabelle	88 Prozent	(7)	88 Prozent	(7)	25 Prozent	(2)	25 Prozent	(12)	8
Formel (nicht im Fließtext)	3 Prozent	(3)	1 Prozent	(1)	18 Prozent	(20)	11 Prozent	(12)	110
Bild, graphische Darstellung	58 Prozent		(19)		36 Prozent		(12)		33

Legende	0–19 Prozent	80–100 Prozent

in 81 Prozent der Fälle in XML dargestellt worden und entsprechen somit den Anforderungen sehr gut. Generell lässt sich sagen, dass PDFX einen größeren Anteil der Strukturelemente sehr gut erkannt hat.

7 Zusammenfassung

Die Untersuchung hat gezeigt, dass beide Tools ähnliche Nachteile aufweisen. Dies betrifft das Erkennen von Formatierung, von Sonderzeichen und von Formeln. Dieses Ergebnis deckt sich mit den Fehlerquellen, die auch im Zuge der Literaturrecherche identifiziert wurden.

Da nur born-digital PDFs in den Testsets enthalten waren, lassen sich diese Probleme nicht auf die Qualität der OCR zurückführen. Verbesserungen können also nur über die Erkennung von Formatierung, Sonderzeichen und Formeln herbeigeführt werden. Hier sind weitere Entwicklungen nötig.

Vergleicht man Teilergebnisse der Untersuchung, lässt sich feststellen, dass der XML-Output von PDFX besser für die Langzeitarchivierung geeignet scheint. Hier wurden die Seitenumbrüche im XML besser gekennzeichnet und das XML ist an einen Standard angelehnt. Außerdem ist das Problem der fehlenden Leerzeichen bei PDFX nicht aufgetreten.

Beim aktuellen Stand der Technik und der Qualität der Umwandlung ist die Konvertierung von wissenschaftlichen Publikationen von PDF in XML noch keine sinnvolle Ergänzung zur Langzeitarchivierung im PDF Format. Aus Sicht der Langzeitarchivierung muss eine XML-Datei all jene Strukturinformationen enthalten, die für die Erhaltung des Informationsgehaltes notwendig sind.

Eine Alternative zu einem Konvertierungsverfahren könnte sein, die Produktionsdaten der Verlage, die in vielen Fällen im XML-Format vorliegen, in das Langzeitarchiv als Ergänzung zu den produzierten PDF-Dateien aufzunehmen. Auf Nachfrage werden diese von einigen Verlagen auch zur Verfügung gestellt.

Autoreninformationen

Thomas Bähr
Technische Informationsbibliothek
Welfengarten 1B
30167 Hannover
thomas.baehr@tib.eu
orcid.org/0000-0002-9337-7127

Merle Friedrichsen
Technische Informationsbibliothek
Welfengarten 1B
30167 Hannover
merle.friedrichsen@tib.eu
orcid.org/0000-0001-7158-8583