Visualisierung 2 LVA: 186.833

Stephan Rotheneder
932 - 0625931*

1 Abstract

Dieses Dokument gehört zum zweiten Abschnitt des Übungsteils der LVA Visualisierung 2. Im ersten Teil dieses Dokuments werde ich auf einige Details der Implementierung eingehen und im zweiten Teil werde ich das Programm vorstellen.

Dies ist eine Einzelarbeit, weshalb nicht der volle Umfang der Möglichkeiten für die Implementierung ausgeschöpft wurde.

2 Technologie Wahl

Für die umsetzung des Implementierungsvorschlages wurde die Technologie Jython gewählt.

Diese Wahl wird ist dadurch begründet dass zum Beginn der Umsetzung zwei Bibliotheken zur Auswahl standen die für das Lesen und Manipulieren von PDF-Dateien genutzt werden können. Diese Bibliotheken waren zum einen *PyPDF* [Fenniak 2008] (ein Python Modul) und *iText* [iTe] (eine Java Library).

Durch den unterschiedlichen Ursprung der beiden Bibliotheken wäre durch die Verwendung von konventionellem Java bzw. CPython jeweils nur eine der beiden Bibliotheken verfügbar gewesen was eine Entscheidung für eine der beiden Bibliotheken noch vor Beginn der tatsächlichen Implementation notwendig gemacht hätte. Durch Verwendung von Jython [jyt 2011] (ein Python - Interpreter für die JVM) stehen Python- und Java-Bibliotheken zur Verfügung.

3 Programm Aufbau und Ablauf

Das Programm besitzt mehrere Komponenten. Diese Komponenten sind eine Organisations-Klasse welche die Komponenten GUI, PDF-Parsing und Text-Analyse triggert.

Die Steuer-Klasse heißt PDFReader und wird einmal instanziert. Diese Instanz von PDFReader legt zunächst für die verwendeten TextFeatures Instanzen an und speichert diese ab. Danach startet der PDFReader eine Instanz der GUI und ist damit bis zu einem Callback der GUI fertig initialisiert.

Die Haupt-GUI-Klasse heißt Window und bietet nach der Instanzierung dem Benutzer ein Fenster mit *File*-Menü in dem ein PDF-File geöffnet werden kann. Wenn eine Datei ausgewählt wurde wird bei der Instanz von PDFReader das Fileparsing ausgelöst.

Die PDFReader-Instanz generiert nun einen Parser für das angegebene PDF-File. Für jede Seite im PDF-File wird bei der GUI ein Seiten-Objekt (RenderPage mit Instanz von MyCanvas und tabellarischer Ansicht) angelegt und ein TextAnalyzer gestartet der den Text der jeweiligen Seite in Sätze und Blöcke untergliedern soll. Danach werden die gegliederten Seiten den TextFeatures zur Analyse übergeben und das Ergebnis wird gespeichert. Danach wird bei der Window-Instanz die erste Seite des Dokuments als markiert gesetzt.

Die Window-Instanz zeigt einerseits den Seitenüberblick links und die ausgewählte Seite zentriert an. Dem Benutzer stehen hier folgende Interaktionen zur Verfügung: Erstens kann der Benutzer eine Seite im Überblick anklicken, worauf die zentrierte Ansicht (egal ob tabellarisch oder Seitenansicht) auf diese Seite wechselt. Zweitens kann im Menü *Analysis* mit der Option *toggle MainView* zwischen der Seitenansicht und der tabellarischen Ansicht gewechselt werden. Drittens kann mit dem *File*-Menü ein anderes PDF geöffnet werden.

4 Resourcen

Es gibt zwei Möglichkeiten dieses Programm zu erhalten.

Erstens, man bekommt ein .zip-File via E-Mail zugeschickt weil man zum Beispiel Georg Molzer heißt und Tutor für Visualisierung 2 ist.

Zweitens, man pulled sich die aktuelle Version vom entsprechenden BitBucket-Repository (*https://bitbucket.org/rothi01/vis2/overview*) welches natürlich erst nach Ende der Deadline am 13.06.2012 public gemacht wird. Hier finden Sie auch einen Installer für die verwendete Jython-Version, dieses Dokument und frühere Abgaben für diese LVA.

4.1 Laufzeit-Umgebung

Für die Verwendung dieser Applikation wird eine JVM mindestens Version 6 sowie eine Jython-Installation benötigt. Auf die Installation von Java wird in diesem Dokument nicht weiter eingegangen. Laden Sie Jython 2.5.3b1 (*http://sourceforge.net/projects/jython/files/jython-dev/2.5.3b1/jython_installer-2.5.3b1.jar/download*) herunter und installieren Sie dieses. Zur Installation von Jython muss bereits eine JRE auf dem Zielrechner installiert sein.

Bitte stellen Sie nach der Installation sicher dass die Installation vollständig ist und auch der Pfad zum Jython "Binaryrichtig im System gesetzt ist. Sie können dies überprüfen in dem Sie in der Kommandozeile oder Shell "jythonëingeben. Wenn jython startet ist alles richtig installiert und jython kann wieder mit quit() beendet werden.

5 Programm start

Um das Programm zu starten öffnen Sie eine Kommandozeile oder Shell im Programmordner *impl*. Der Programmordner enthält eine Datei start.py sowie einige Unterordner die Bibliotheken, Bildresourcen und natürlich Programmcode enthalten. Geben Sie nun in Ihre Kommandozeile oder Shell folgendes Kommando ein:

jython start.py

Nun sollte das Programm starten. In *Abbildung 1* sehen Sie das Fenster das nun bei Ihnen gestartet sein sollte.

In den folgenden Abbildungen *Abbildung 1* und *Abbildung 2* sieht man wie ein Dokument geöffnet und (nach kurzer Analysezeit) dargestellt wird. Bitte beachten Sie dass die Sätze bereits Farbkodiert hinterlegt werden. Diese Hintergrundfarbe ergibt sich aus dem gemittelten Ratings der einzelnen Features für diesen Satz und wird einem Farbverlauf von Blau über Weiß zu Rot entnommen.

*e-mail:stephan.rotheneder@student.tuwien.ac.at

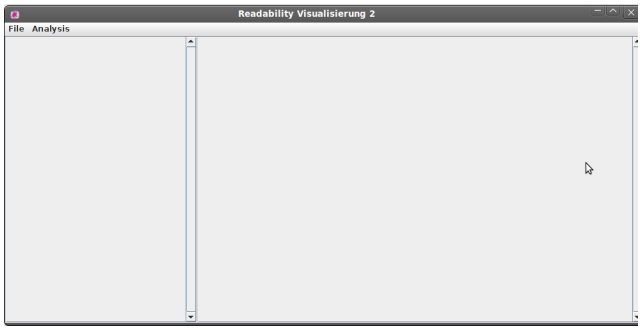

Abbildung 1: *Initialisiertes Fenster nach dem Startup.*

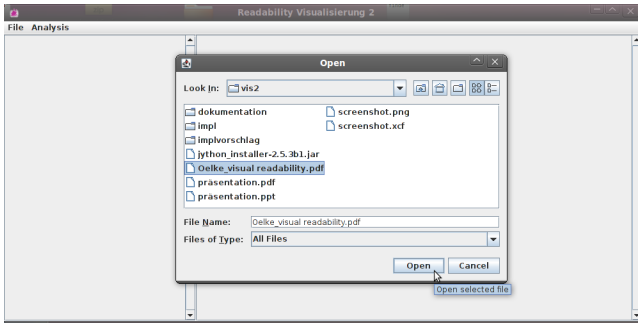

Abbildung 2: *Datei-Öffnen-Dialog*

Wechselt man im Menüpunkt *Analysis* mit *toggle MainView* auf die tabellarische Ansicht, sieht man die Aufschlüsselung jedes einzelnen Satzes nach den Ergebnissen der Features.

6 Ausblick

Diese Sektion behandelt viele Dinge die nicht im Programm sind aber eine Verbesserung des Programms darstellen würden.

6.1 Textparsing

Obwohl bereits ganz passable Ergebnisse erziehlt werden, lässt das bisherige Textparsing noch zu wünschen übrig. Graphen im PDF werden Teilweise als Sätze erkannt. Wortabteilungen werden noch nicht richtig korrigiert.

Leider ist Textverständnis ein sehr umfangreiches Feld und kann nicht leicht in ein kurzes Stück Code abgebildet werden.

6.2 Sortierung

Um den Benutzer bei seiner Arbeit mit dem Programm zu unterstützen wäre eine Sortierung der Sätze nach Rating in der tabellarischen Ansicht wünschenswert.

6.3 Darstellungsmodus

Obwohl im Code bereits vorgesehen und für die Seitenansicht bereits Umgesetzt fehlt noch ein Zugang (und umsetzung für tabellarische Ansicht) zum Rendermodus Block. Hier werden nicht Sätze sondern ganze Blöcke beurteilt und Farbkodiert hinterlegt.

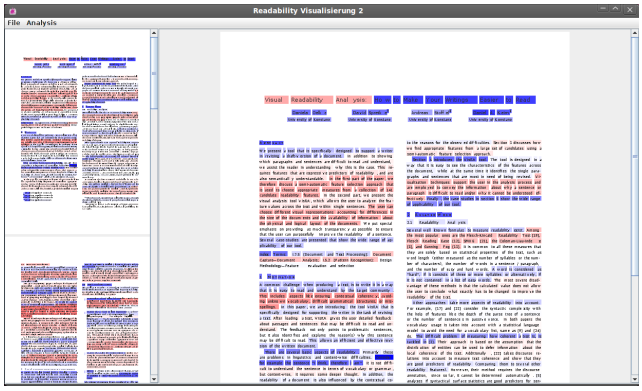

Abbildung 3: *PDF-Darstellung, Seitenansicht.*

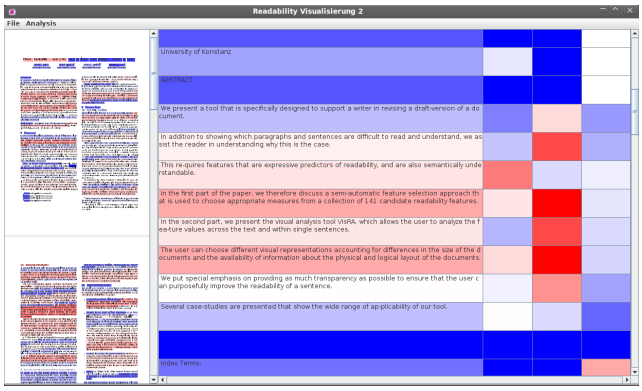

Abbildung 4: *Tabellarische Ansicht - Rating der Sätze für jedes Feature*

6.4 GUI Erweiterungen

Weitere Optionen im Menü Analysis, Parametrisierung der Features, Aktivieren/Deaktivieren der Features, Tooltips die das Rating anzeigen sind nur ein paar der Verbesserungen die mir einfallen.

Literatur

FENNIAK, M. 2008. *Pure-Python library built as a PDF toolkit*, 1.12 ed. http://pybrary.net/pyPdf/, Sept.

ITEXT SOFTWARE CORP. *The core iText*, 5.2.2 ed. http://api.itextpdf.com/itext/. iText is available in Java as well as in C.

2011. *Jython Documentation*, 2.5.2 ed. http://www.jython.org/docs/index.html, Mar.

OELKE, D., STOFFEL, D. S. A., AND KEIM, D. A. 2010. Visual readability analysis: How to make your writings easier to read. *IEEE Conference on Visual Analytics Science and Technology* (Oct.), 123–130.