Projektseminar Mediengestaltung
Informationsvisualisierung
Sarah Wagner und Peter Zeitlhöfler

Dokumentation: OPAC-Datensatz

1. Datenaufbereitung

Der Datensatz den wir ursprünglich erhalten haben, lag in einem unübersichtlichen xml-Format vor, welches zuvor aus einer *.unimarc Datei erstellt wurde. Um den Datensatz übersichtlicher zu gestalten und nur die für die Visualisierung wichtigen Informationen heraus zu filtern, haben wir mit Hilfe von XSLT und Saxon eine Transformation in eine neue XML Datei vorgenommen. Diese enthält jetzt nur noch die für uns relevanten Informationen und ist übersichtlich strukturiert. So konnte sie auch leicht mit einem Tool (MoorXmlToCsvConverter) in eine *.csv Datei konvertiert werden. Die Datei enthält nun folgende Daten: Titel, Untertitel, Ort, Verlag, Jahr, Seitenzahl, ob die Ressource eine elektronische Ressource ist und die Signaturen. Anschließend haben wir mit Java Script und der Library Papaparse die *.csv Datei noch weiter aufbereitet, denn einige Einträge waren schwer zu verarbeiten oder unterschiedlich formatiert. Aus den Signaturen haben wir die Standortkennzeichen herausgefiltert und den unterschiedlichen Bibliotheken zugeordnet. Die Seitenzahlen waren unterschiedlich formatiert (z.B. „283 p. S. VII. 12Cm“ → „283“) genau so wie die Jahreszahlen (z.B. „20[07]“ → „2007“), was behoben werden musste. Außerdem haben wir alle Einträge in Kleinbuchstaben umgewandelt, um sie später leichter filtern zu können. Zum Schluss haben wir die Kodierung von Umlauten und Sonderzeichen noch ersetzt (z.B. „Éu“ → „ue“).

2. Erstellung des Tools

Angefangen haben wir damit, das Serverscript zu verfassen und somit die Kommunikation zwischen Server und Client zu ermöglichen. Da für unsere Filterfunktionen alle Daten benötigt werden, wird die *.csv Datei serverseitig durchsucht. Die Daten werden in Form eines JSON Strings vom Server zurückgeliefert und enthalten zum einen die geschickte Anfrage, sowie eine Anzahl der Treffer auf die Anfrage, welche serverseitig ermittelt wurde. Als die richtigen Daten vom Server geliefert wurden, haben wir zuerst das Grundgerüst der Seite mit html und css erstellt und somit das User Interface aufgebaut, dabei haben wir das Framework Bootstrap zur Hilfe genommen. Danach ermöglichten wir eine Interaktion, indem wir Funktionen hinter die UI-Elemente gelegt haben, damit der Benutzer selbst Anfragen an den Server schicken kann. Diese sind URLs, welche aus den Inputeingaben des Nutzers erstellte werden. Danach haben wir uns mit der Visualisierung von Daten mit d3.js befasst. Schnell wurde aber klar, dass die Library für unsere Visualisierungen zu umfangreich und zu aufwendig zu benutzen ist, weshalb wir die Darstellung mit Hilfe von googleCharts realisiert haben. Hauptsächlich werden unsere Daten in Balkendiagrammen dargestellt, nur die Verteilung der Sprachen und der Medien wird in einem Tortendiagramm realisiert und Vergleiche in einem Flächendiagramm. Im Anschluss daran haben wir den Datensatz unter der Verwendung der von uns erstellten Filterfunktion nach interessanten Mustern durchsucht und so statische Visualisierungen der Daten in die Anwendung integriert. Viel Aufwand war es dann noch, kleine Fehler in unserem Tool zu identifizieren und zu beheben. Am Ende haben wir uns dann doch dazu entschieden, die Visualisierungen nicht mit googleCharts, sondern mit highCharts vorzunehmen und haben den Code dahingehend modifiziert, denn highCharts bietet eine ansprechendere Darstellung und eine bessere Möglichkeit die Visualisierung im Nachhinein zu bearbeiten.

3. Aufgabenverteilung

Datenaufbereitung: Peter Zeitlhöfler: xslt Transform in neues xml Dokument
Sarah Wagner: Parsen der *.csv Datei mit Javascript
Erstellung des Tools: Hier ist die Aufgabenteilung schwer zu bestimmen, denn die größten Probleme haben wir zusammen gelöst und in gemeinsamen Sitzungen den Code erstellt. Peter Zeitlhöfler hat die Darstellung der Visualisierungen mit highCharts vorgenommen.
Präsentation: Auch diese haben wir zusammen erstellt und ausgearbeitet.
Dokumentation: Erstellt von Sarah Wagner

4. Installationsanweisung

Navigieren Sie im Terminal in den „NodeServer“ Ordner unseres Projektes und geben Sie folgenden Befehl ein: „node server.js“. So starten sie das Serverscript. Nach ungefähr 10 Sekunden erscheint die Ausgabe „data loaded“, das bedeutet die *.csv Datei wurde geladen und unsere Anwendung ist benutzbar. Sollte der Befehl nicht funktionieren muss der Ordner „node-modules“ gelöscht werden und folgende node-Module neu installiert werden:

- npm install node
- npm install express
- npm install path
- npm install fs
- npm install cors
- npm install csvtojson

Nach der Neuinstallation sollte der Befehl node server.js funktionieren.

Hinweis: Bei der Installation auf dem Uniserver (132.199.139.24) funktionierte die neu installierte Version von csvtojson nicht, weshalb wir den Ordner csvtojson aus dem Mensaapp-Beispiel der Vorlesung verwendeten.