4. Verwendete Technologien

4.1. Apache Lucene

Als Framework für das Backend unserer Suchmaschine nutzen wir das auf Java basierte Apache Lucene, wobei es sich um eine von der Apache Software Foundation unter der Apache Lizenz kostenlos zur Verfügung gestellte Programmbibliothek handelt, welche unter Anderem von Webseiten wie Twitter verwendet wird. Lucene bietet die Möglichkeit einen Index aus gecrawlten Daten aufzubauen, wobei es auf den von Stopwörtern befreiten Text das Tf-idf-Maß und ein Vektorraum-Modell anwendet, eine Funktion zur Suche auf dem erzeugten Index und darüber hinaus die Option Werte für das Ranking anzupassen.

4.2. Apache Tomcat

Zur Darstellung unsere Suchmaschine nutzen wir den Webserver Apache Tomcat, welcher ebenfalls von der Apache Software Foundation kostenlos unter der Apache Lizenz zur Verfügung gestellt wird, wodurch aufgrund von gleicher Programmiersprache eine verhätnismäßig simple Einbindung des Apache Lucene Codes möglich ist. Nach Start des Servers erzeugt Lucene auf Anfrage einen Index aus unserem Datensatz und Tomcat stellt eine Webseite bereit, über die man Queries an ein Servlet senden kann. Dieses gibt unter Einbezug des von Lucene erzeugten Indexes eine Menge an nach Relevanz für die gegebene Anfrage sortierten Dokumenten aus.

5. Aufbau und Funktion

5.4. User Interface

Der Nutzer hat die Möglichkeit die Suche über ein User Interface durchzuführen, welches aus einem Textfeld, einem Submit-Button, einem Button für das Durchführen der Indexierung und einer Checkbox für den Evaluationsmodus besteht. Nach Senden und Verarbeiten der Anfrage bekommt der User Ergebnisse in Form einer Liste über eine von Tomcat erzeugte HTML-Webseite ausgegeben. Nach einer erfolgreichen Suche werden die Ergebnisse mit Details wie ID, Titel und Inhalt der Rezension nacheinander dargestellt.

5.5. Evaluationsmodus

Man aktiviert den Evaluationsmodus indem man bevor man eine Suchanfrage sendet ein Häkchen im Feld „Evaluationsmodus“ setzt. Darauf hin wird ein Servlet zur Evaluation aufgerufen, welches eine Erweiterung des QueryServlets darstellt. Im Gegensatz zum normalen Suchmodus wird hier zu jedem Suchergebnis mithilfe von Buttons und einer geeigneten Darstellung via CSS zusätzlich ein Interface generiert, über das man die Suchergebnisse in Abhängigkeit zu einem gegeben Query als relevant oder nicht relevant bewerten kann. Dabei bekommen alle Dokumente anfangs den Status „nicht bewertet“ zugeordnet. Die Bewertung wird asynchron in einer vom Servlet erzeugten JSON Datei gespeichert und bei jedem erneuten Aufruf des Evaluationsmodus wird die aktuelle Bewertung des Suchergebnisses herangezogen und wiederhergestellt. Die erstellte JSON Datei beinhaltet jeweils Tripel aus den Werten Document ID, Topic ID und Relevanz. Beim Auslesen der JSON Datei erzeugt das EvaluationServlet eine Map, aus welcher dann wiederum eine HashMap mit Paaren von Queries und Topic IDs erstellt wird. Die abgegebenen Bewertungen haben Einfluss auf die Darstellung der Suchergebnisse, da unser QueryServlet eine auf Bubblesort basierende Sortierfunktion bereitstellt, welche zuvor als „nicht relevant“ bewertete Ergebnisse in der Darstellungsreihenfolge herunter setzt und als „relevant“ eingestufte Ergebnisse mit erhöhter Priorität anzeigt. Die Sortierfunktion haben wir in dieser Form gewählt da sie eine stabile Sortierfunktion ist, die Reihenfolge der Ergebnisse die nicht getauscht werden bleibt also erhalten. Das ist insofern wichtig, da zuvor eventuell noch andere Sortierfunktionen Einfluss auf die Ergebnisreihenfolge haben können und diese vor allem bei den nicht sortierten Ergebnissen logischerweise beibehalten werden sollte.