Informatik
Hauptcampus

H O C H
S C H U L E
T R I E R

Implementierung einer Enterprise Search Engine für das Dietrich Online Projekt

Implementation of an enterprise search engine for the Dietrich Online project

Florian Reitz

Bachelor-Abschlussarbeit

Betreuer: Professor Doktor Christoph Schmitz

Trier, den 15.10.2019 15.3.2020

Vorwort

Ein Vorwort ist nicht unbedingt nötig. Falls Sie ein Vorwort schreiben, so ist dies der Platz, um z.B. die Firma vorzustellen, in der diese Arbeit entstanden ist, oder einigen Leuten zu danken, die in irgendeiner Form positiv zur Entstehung dieser Arbeit beigetragen haben. Auf keinen Fall sollten Sie im Vorwort die Aufgabenstellung näher erläutern oder vertieft auf technische Sachverhalte eingehen.

Kurzfassung

In der Kurzfassung soll in kurzer und prägnanter Weise der wesentliche Inhalt der Arbeit beschrieben werden. Dazu zählen vor allem eine kurze Aufgabenbeschreibung, der Lösungsansatz sowie die wesentlichen Ergebnisse der Arbeit. Ein häufiger Fehler für die Kurzfassung ist, dass lediglich die Aufgabenbeschreibung (d.h. das Problem) in Kurzform vorgelegt wird. Die Kurzfassung soll aber die gesamte Arbeit widerspiegeln. Deshalb sind vor allem die erzielten Ergebnisse darzustellen. Die Kurzfassung soll etwa eine halbe bis ganze DIN-A4-Seite umfassen.

Hinweis: Schreiben Sie die Kurzfassung am Ende der Arbeit, denn eventuell ist Ihnen beim Schreiben erst vollends klar geworden, was das Wesentliche der Arbeit ist bzw. welche Schwerpunkte Sie bei der Arbeit gesetzt haben. Andernfalls laufen Sie Gefahr, dass die Kurzfassung nicht zum Rest der Arbeit passt.

The same in english.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1

Einleitung und Problemstellung

Begonnen werden soll mit einer Einleitung zum Thema, also Hintergrund und Ziel erläutert werden.

Weiterhin wird das vorliegende Problem diskutiert: Was ist zu lösen, warum ist es wichtig, dass man dieses Problem löst und welche Lösungsansätze gibt es bereits. Der Bezug auf vorhandene oder eben bisher fehlende Lösungen begründet auch die Intention und Bedeutung dieser Arbeit. Dies können allgemeine Gesichtspunkte sein: Man liefert einen Beitrag für ein generelles Problem oder man hat eine spezielle Systemumgebung oder ein spezielles Produkt (zum Beispiel in einem Unternehmen), woraus sich dieses noch zu lösende Problem ergibt.

Im weiteren Verlauf wird die Problemstellung konkret dargestellt: Was ist spezifisch zu lösen? Welche Randbedingungen sind gegeben und was ist die Zielsetzung? Letztere soll das beschreiben, was man mit dieser Arbeit (mindestens) erreichen möchte.

2

Vergleich der Enterprise Search Engines

In ersten Schritt werden diverse Enterprise Search Engines evaluiert. Dafür wurde eine Anforderungsliste mit den Mitarbeitern erstellt. Die Systeme welche bei diesen Vergleich nach Features am besten abschneiden werden anschließend aufgesetzt, getestet und genauer verglichen.

- Open Source oder Kostenlos
- Unterstützung von Facetten
- Ranking der Suchergebnisse
- Volltextsuche
- Support für PDF, SQL, XML
- Logging-Möglichkeit

Des Weiteren sind die folgenden Funktionen auch wichtig, allerdings keine K.O. Kriterien:

- Support für PostgreSQL
- Backup Funktionen
- Auto-Korrektur und Auto-Vervollständigung
- Security Features
- PHP-Support
- bezahlter Support

Durch die begrenzten finanziellen Mittel und die lange Projektlaufzeit besteht die Notwendigkeit eine kostenfreie, im besten Fall sogar Open Source Suchmaschine zu finden. Auch äußerst wichtig ist der Support für Facetten, da viele Dietrich-Online als Suchmaschine den Nutzer einige Tools zum Verfeinern seiner Suchergebnisse zur Verfügung stellen will. Das Ranking der Suchergebnisse ist vor allem für die Transparenz wichtig, da hier mit vielen Daten gearbeitet wird, welche allerdings nicht alle gleichzeitig dargestellt werden können. Dieses Problem kann mit einer Gewichtung der Suchergebnisse behoben werden, welches dann auf der Seite für Transparenz veröffentlicht wird. Die Volltextsuche wird es möglich machen auch nach Schlüsselwörtern im Titel zu suchen. Der Support von den verschiedenen Dateiformaten ergibt sich dadurch, dass dieses Projekt stark gewachsen ist. Es gibt viele Prozessschritte, welche auf denselben Daten in verschiedenen Formen arbeiten. Darunter werden alle Einträge im XML Format bearbeitet, es gibt alle

Scans als PDF und für die Webseite sind alle Daten nochmals in der Datenbank vorhanden. Dadurch gibt sich auch das Problem, dass Daten mehrfach vorliegen welches sicherlich Später von größerer Bedeutung sein wird. Als letztes ist es noch wichtig, dass eine Logging-Möglichkeit geboten wird, damit schnell und effizient Probleme mit dem System erkannt und gelöst werden können.

Ein Support für PostgreSQL ist für dieses Projekt nicht so wichtig, allerdings könnte es sein, dass der Server später auch andere Datenbanken verwaltet. Der Server wird sowieso täglich durch einen Automatismus gesichert. Allerdings ist eine manuelle Backup-Lösung wünschenswert, um die Suchmaschine losgelöst von den Server zu sichern und gegebenenfalls auch einfach auf einen anderen Server umzuziehen. Auto-Korrektur und Auto-Vervollständigung sind beide sehr interessant, um den Nutzer mehr Komfort-Funktionen bieten zu können. Der Server sollte ja generell nur intern Ansprechbar sein. Allerdings gibt es für manche der Suchmaschinen eine Web-Oberfläche, da wäre es wichtig eine sichere Verbindung dorthin und gegebenenfalls eine Möglichkeit der Mehrfaktor-Authentifizierung zu haben. Einen PHP-Connector, welcher Objekte zum Umgang mit der Suchmaschine bietet, wäre wünschenswert. Allerdings bieten einige Suchmaschinen auch die Möglichkeit über JSON Anfragen an die Suchmaschine zu stellen. Allerdings sollte zumindest eine der beiden Möglichkeiten gegeben sein. Zuletzt wäre für akute Probleme ein bezahlter Support, der dafür schneller reagiert wünschenswert.

2.1 Apache Lucene Core

Lucene Core ist eine Open Source Enterprise Search Engine von der Apache Foundation geschrieben in Java.

Das Lucene Projekt wurde im Jahre 1997 vom Entwickler Doug Cutting gestartet. 2001 ist es dann der Apache Foundation als Teil des Jakarta-Projekts beigetreten und wurde 2005 ein eigenes Hauptprojekt der Foundation. [Wik19b]

Lucene Core erfüllt alle der Grundanforderungen. Für das Monitoring gibt es eine Klasse, die es auch ermöglicht, dass langsame Query's geloggt werden. Zudem bietet es Support für PostgreSQL und Auto-Korrektur/Auto-Vervollständigung. Da es keine Web-Oberfläche besitzt, gibt es auch keine weiteren Sicherheitsfunktionen. Einen PHP-Connector gibt es leider auch nicht, man müsste daher mit PHP direkte Systemaufrufe an Java machen. Bezahlten Support gibt es hier nicht, da dieses Projekt zur Apache Foundation gehört. [The19a]

2.2 Terrier

Terrier ist eine Open Source Enterprise Search Engine geschrieben in Java. Entwickelt und gepflegt wird diese von der University of Glasgow. Sie existiert bereit seit 10 Jahren und besitzt, laut Webseite, eine breite Nutzerbasis. Terrier erfüllt leider nicht alle Grundanforderungen, da es keine direkte Möglichkeit gibt SQL zu indexieren. Es gibt allerdings eine Möglichkeit das SQL in JSON zu konvertieren

und dieses dann in die Suchmaschine einzupflegen. Auch scheint es kein Support für Facetten gegeben. [MC19]

2.3 Sphinx

Sphinx ist eine Suchmaschine entwickelt von Andrew Aksyonoff. Das Akronym steht für „SQL Phrase Index“.[Sphb] Bis zur Version 2 wurde sie aktiv Open Source entwickelt. Ab Version 3 wurde die Entwicklung closed Source. Auf der Github-Seite steht: „The sources for 3.0 will also be posted here when we decide to make those publicly available.“ [sphc], also gibt es kein genaues Datum ob und wann die Version 3 Open Source geht. Version 3.1.1 wurde im Oktober 2018 veröffentlicht und seitdem lässt sich auch nichts mehr über das Projekt finden. Von daher ist davon auszugehen, dass das Projekt nicht mehr weitergeführt wird.

Zu den Features ist festzuhalten, dass es keinen nativen PDF-Support in der Open-Source Version existiert, ab der Version 3 jedoch wurde ein Dokumenten-Speicher eingebaut. Allerdings werden die anderen Anforderungen alle erfüllt. Es existiert, laut Webseite, sogar ein bezahlter Support, allerdings ist fraglich, ob man mit der Firma noch in Kontakt treten kann. [Spha]

2.4 Apache Solr

Apache Solr ist eine, auf Lucene Core 2.1 viel eingesetzte Search Engine von der Apache Foundation. Sie basiert auf Apache Lucene Core und erweitert dieses um ein grafische Benutzeroberfläche und einige Features. Die Entwicklung dafür begann 2004 als ein internes Projekt von CNET um eine bessere Suche für die eigene Webseite zu bieten. Später im Jahre 2006 hat CNET dann den Source Code an die Apache Foundation weitergegeben. Dadurch wurde es zu einem eigenen Projekt bei der Apache Foundation. Im Jahre 2009 wurden Solr dann in das Apache Lucene Projekt eingefügt. Dort wird es auch aktuell noch weiterentwickelt. [Wik19a]

Solr wird unter anderem von DuckDuckGo und Best Buy eingesetzt. Durch die Unterstützung von der Apache Foundation längerfristige Weiterentwicklung abzusehen.

Da Solr zur Apache Foundation gehört, ist es Open Source. Es bietet viele Funktionen von Haus, damit erfüllt es alle Grundanforderungen und besitzt darüber hinaus auch Support für fast alle Bonus-Features. Einzig und allein gibt es keinen bezahlten Support, dafür allerdings eine große Community, welche man durch einen Mailing Liste oder IRC erreichen kann.

[The19b]

2.5 ElasticSearch

Eine weitere großes Enterprise Search Engine ist ElasticSearch. Auch dieses Projekt arbeitet auf der Basis von Lucene. Zu den bekanntesten Kunden zählen Ebay

und Adobe. Gestartet wurde das Projekt in den jungen 2000ern von Shay Banon, um eine Verwaltung für die Rezepte seiner Frau zu schaffen. Im Juni 2012 haben sich dann Logstash, ein Logging Dienst, Kibana, ein UI für ElasticSearch, und ElasticSearch zusammengetan. Alle kamen zusammen in der ElasticSearch Incorporated. Seitdem wurden der Produktkatalog stetig erweitert und die Produkte weiterentwickelt. Viele der weiteren Produkte sind allerdings nicht mehr Open-Source oder kostenlos. Der ELK-Stack ist allerdings weiterhin kostenlos und ElasticSearch zudem auch als Open-Source Variante zu haben. Eine genauere Aussage, welche Features nur in der kostenlosen und nicht in der Open-Source Variante zu finden sind, finden sich in der Tabelle 2.1.

ElasticSearch erfüllt alle der Grundanforderungen, auch in der Open-Source Variante. Auch viele der optionalen Features kann man in der Open Source Variante genießen. Einzig die Sicherheitsfunktionen, wie rollen-basierte Authentifizierung sind der kostenlosen Variante vorbehalten. Eine Möglichkeit auf bezahlten Support besteht auch, dafür muss auf eine bezahlte Version gewechselt werden, was auch einige Funktionen wie IP-Filter mit sich bringt. Allerdings ist diese daraufhin auch nichtmehr Open-Source. [Ela]

2.6 Fess

Fess ist eine Enterprise Search Engine basierend auf ElasticSearch entwickelt von dem japanischen Unternehmen CodeLibs. Die Suchmaschine ist komplett Open-Source und wird unter der Apache-Lizenz entwickelt.

Die Suchmaschine erfüllt alle Grundanforderungen. Darüber hinaus bietet es Support für PostgreSQL, Backups (sogar über die Web-Oberfläche) und Auto-Korrektur und Vervollständigung. Es gibt keinen direkten PHP Support, allerdings können anfragen über JSON geschickt werden. Ein bezahlter Support ist auch möglich über die Firma N2SM Incorporated. [N2S] Bei dieser Arbeiten anscheinend auch einige der Entwickler von FESS. Sicherheitsfunktionen werden über rollenbasierte Authentifizierung mitgeliefert. [Cod19]

2.7 Algolia

Algolia ist eine cloud-basierte Search Engine, welche unter anderem von Twitch und Lacoste verwendet wird. Die Suchmaschine wird hierbei als SAAS (Software as a Service) angeboten. Hierbei lädt man die Daten auf Algolia Server und dann daraufhin der API Suchen auf den Daten ausführen.

Es erfüllt alle Grundanforderungen, wobei allerdings in der kostenlosen Variante grade einmal 10 Tausend Einträge und 50 Tausend Operationen im Monat erlaubt sind. Diese Einschränkung macht die kostenlose Variante dieser Suchmaschine für das Dietrich-Online Projekt unbrauchbar. Von den optionalen Anforderungen erfüllt Algolia auch alle. Der bezahlte Support wird ab der Starter Edition für 30 Dollar im Monat mitgeliefert. [Alg]

2.8 Manticore Search

Manticore Search Engine ist eine Open-Source Solution basierend auf Sphinx 2.3. Nachdem Sphinx Closed-Source gegangen ist, wurde auf der letzten offenen Version die erste Version von Manticore Search entwickelt. Zu den großen Kunden zählen unter anderem Craigslist und Boardreader.

Manticore erfüllt fast alle Grund Anforderungen, allerdings ist kein nativer PDF-Support gegeben. Es muss daher auf eine Konvertierung der Daten auf XML gesetzt werden. Es findet sich außerdem eine Unterstützung von PostgreSQL, sowie Auto-Korrektur und Vervollständigung. Es gibt auch einen Query-Log. Zuletzt gibt es noch eine Option auf bezahlten Support. Die Supportkosten sind dabei direkt auf der Webseite angegeben und belaufen sich auf 3000 Dollar im Jahr für den Standard Support. [Man]

2.9 Xapian

Xapian ist eine Open-Source Enterprise Suchmaschine, welche von Zeit-Online, der Universitätsbibliothek Köln und der Debian Webseite genutzt wird. Die Suchmaschine basiert auf Open Muscat, einer Suchmaschine, welche an der Cambridge Universität in den 1980ern von Dr. Martin Porter entwickelt wurde. In 2001, als Open Muscat Closed-Source ging, haben sich einige Entwickler die letzte offene Version geladen und diese weiterentwickelt.

Sie erfüllt alle der Grundanforderungen, wenn auch Logging nur im Grundsinne erfüllt wird, da nur Errors geschmissen werden. Des Weiteren bietet die Suchmaschine Support für PostgreSQL. Auch eine Replikations-Funktion wird mitgeliefert. Sie bietet auch Auto-Korrektur und Auto-Vervollständigung. Ein Login-System mit Sicherheitsfunktionen gibt es durch das Fehlende Frontend Administration nicht. Es gibt allerdings die Möglichkeit mit Omega eine CGI-Suche zu nutzen. Diese Suche bietet allerdings keine Administration, sondern nur eine grafische Oberfläche für Suchanfragen.

Auch gibt es eine Möglichkeit für bezahlten Support. Auf der Webseite werden zwei Firmen angegeben, welche bezahlten Support bieten. Allerdings funktioniert der Link aktuell nur für eine der beiden Firmen aktuell. Zudem ist ein PHP-Connector für die Suchmaschine vorhanden, was die Einbindung ist das Projekt vereinfacht. [XAP19]

2.10 Vorauswahl

Alle Suchmaschinen die zumindest die Grundanforderungen erfüllen, werden hier in der Tabelle nun nochmals aufgeführt für einen leichteren Vergleich.

Nach einem ersten Überblick wurden nun Aufgrund der Auswahlkriterien diese Systeme zum genaueren Vergleich ausgewählt: Apache Solr, Manticore Search, ElasticSearch (in der kostenlosen Version) und Xapian. Lucene Core wird nicht genauer untersucht, da Solr ein umfassenderes Paket bietet, welches den gestellten Anforderungen mehr entspricht.

	LC	**SH**	**AS**	**ES**	**FE**	**AG**	**XP**
Open Source oder Kostenlos	x	x	x	x	x	x	x
Unterstützung von Facetten	x	x	x	x	x	x	x
Ranking der Suchergebnisse	x	x	x	x	x	x	x
Volltextsuche	x	x	x	x	x	x	x
Support für PDF, SQL, XML	x	x*	x	x	x	x	x
Monitoring / Logging	x	x	x	x	x	x	x?
Support für PostgreSQL	x	x	x	x	x	x	x
Backup	-	-	x	x	x	x+	-
Auto-Korrektur und Vervollständigung	x	x	x	x	x	x	x
Security Features	-	-	x-	x*	x	x	-
PHP Support	-	x	x	x	-	x	x
bezahlter Support	-	x	-	x	x	x	x
unter aktiver Entwicklung**	x	-	x	x	x	x	x
offizelles Docker Image	-	-	x	x	x	-	-
Synonym Support	x	x	x	x	x	x	x
Web-Interface	-	-	x	x	x	x	-
Plugin Support	-	x	x	x	x	-	-
JSON oder RESTful API	-	x*	x	x	x	-	x–
SQL-Like Query Support	-	x	x	x	-	-	-

Tabelle 2.1. Feature-Vergleich der verschiedenen Enterprise Suchmaschinen

* = Feature nur in der kostenlosen Variante verfügbar.
** = Update innerhalb des letzen halben Jahres
– = Nur mit Omega CGI installiert
+ = Anbieter kümmert sich um das Feature
- = Funktion nur per Plugin Implementiert

Die Tabelle vergleicht einige Features der ausgewählten Search Engines. Dabei wurden die Namen aus Platzgründen wie folgt abgekürzt:

- LC = Lucene Core 2.1
- SH = Sphinx 2.3
- AS = Apache Solr 2.4
- ES = ElasticSearch 2.5
- FE = Fess 2.6
- AG = Algolia 2.7
- XP = Xapian 2.9

3

Nutzung des Open Archives Initiative Protokolls für Metadaten

Das Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) ist ein Protokoll zum Austausch von Metadaten. Dabei werden Anfragen per GET oder POST-Request angefragt. Als Antwort erhält man im Folgenden ein XML-Dokument. So können die Metadaten mit bestimmten Facetten abgefragt werden (zum Beispiel Autor). Dabei geht es allerdings darum primär darum Änderungen weiterzugeben. So können durch dieses Protokoll neue Einträge oder Änderungen in der Datenbank weitergeben werden. [Deu]

3.1 OAI Harverester

Ein OAI Harverester ist ein Programm, welches durchgehend einen Abgleich der Daten vollführt. Dabei lässt es sich die Änderungen mit einem List-Befehl von dem Server geben und gleicht diese danach mit der eigenen Struktur ab. Sollten dabei Unterschiede festgestellt werden, werden daraufhin die Änderungen auch beim Harverester eingefügt. So steht der Harverester immer mit dem Server auf einen Stand. [Deu]

3.2 Support der Enterprise Search Engines

Bei den vorhin genannten Enterprise Search Engines gibt es keine mit nativen OAI Harverester Support. Es gibt die Möglichkeit für manche der Suchmaschinen ein solches Verhalten mithilfe von Plugins zu implementieren. Allerdings sind die meisten dieser Add-ons auch schon veraltet.

3.3 Auswertung

Durch eine Fehlende Basisimplementierung des Protokolls in den einzelnen Suchmaschinen und der Möglichkeit eines direkten Zugriffs auf die Datenbank, sehe ich keinen Grund dieses Protokoll einzubauen. Es müsste ein Server vor die Datenbank installiert werden und ein Harverester vor der ESE. Dies ist ein großer Mehraufwand. Sollte allerdings die ESE ein übergreifendes System werden, kann darüber nachgedacht werden, die anderen Datenbanken per Harverester anzusprechen.

4

Zusammenfassung und Ausblick

In diesem Kapitel soll die Arbeit noch einmal kurz zusammengefasst werden. Insbesondere sollen die wesentlichen Ergebnisse Ihrer Arbeit herausgehoben werden. Erfahrungen, die zum Beispiel Benutzer mit der Mensch-Maschine-Schnittstelle gemacht haben oder Ergebnisse von Leistungsmessungen sollen an dieser Stelle präsentiert werden. Sie können in diesem Kapitel auch die Ergebnisse oder das Arbeitsumfeld Ihrer Arbeit kritisch bewerten. Wünschenswerte Erweiterungen sollen als Hinweise auf weiterführende Arbeiten erwähnt werden.

Literaturverzeichnis

Alg. *Fast, reliable and modern search and discovery.*

Cod19. *Fess Installation Guide*, 31.10.2019.

Deu. *OAI-Schnittstelle.*

Ela. *Elasticsearch: Verteilte RESTful-Suchmaschine und -Analytics Engine — Elastic.*

Man. *Manticore Search – open source text search engine for big data and stream filtering.*

MC19. McCreadie, Richard und Craig Macdonald and Jie Peng: *Terrier IR Platform - Homepage*, 25.01.2019.

N2S. *Professional Support — N2SM, Inc.*

Spha. *Sphinx — Open Source Search Engine.*

Sphb. *Sphinx — Open Source Search Server.*

sphc. *Sphinx search server.*

The19a. *Apache Lucene - Apache Lucene Core*, 26.07.2019.

The19b. *Apache Solr*, 26.07.2019.

Wik19a. *Apache Solr - Wikipedia*, 14.10.2019.

Wik19b. *Apache Lucene - Wikipedia*, 27.10.2019.

XAP19. *The Xapian Project*, 14.10.2019.

A

Glossar

ESE	Enterprise Search Engine
Facetten	Filter in Bibliothekarssprache
OAI	Open Archives Initiative

B

Erklärung der Kandidatin / des Kandidaten

☐ Die Arbeit habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Datum	Unterschrift der Kandidatin / des Kandidaten