Semantik-gestützte Analyse von und Suche in Kundenspezifikationen im Maschinenbau

Martin Voigt, Daniel Hladky

Ontos AG
Mittelstrasse 24, CH-2560 Nidau
martin.voigt,daniel.hladky@ontos.com

Abstract: Die gezielte Suche von Informationen in großen Dokumentenmengen ist eine der wesentlichen Herausforderungen der heutigen Zeit. In diesem Papier wird beschrieben, wie wir die Analyse von und Suche in mehrsprachigen Kundenspezifikationen in einem aktuellen Kundenprojekt im Maschinenbau realisiert haben. Im Rahmen der Dokumentenanalyse kommen computerlinguistische und semantische Technologien zum Einsatz. Basis für die Suche bildet das Paradigma des Faceted Browsing.

1 Motivation und Problemstellung

In nahezu allen Unternehmensbereichen wachsen heutzutage die Datenmengen drastisch an, so dass es für den Nutzer zunehmend schwieriger wird, einen Überblick zu behalten bzw. die darin enthalten Informationen schnell und zuverlässig aufzufinden. Diese Herausforderungen wurden insbesondere auch im noch laufenden Kundenprojekt mit der AviComp Controls GmbH (ACC), einem Engineering-Spezialist für Steuerungslösungen für rotierende Maschinen, deutlich. In diesem Unternehmen muss der Vertrieb effizient bis zu 100 Spezifikationen je Anfrage von internationalen Kunden hinsichtlich der Passgenauigkeit auf das eigene Leistungsspektrum analysieren, Unklarheiten identifizieren und Angebote unterbreiten. Hierbei müssen teilweise auch bereits abgegebene Angebote oder hauseigene Spezifikationen aus dem gesamten Dokumentenstamm (> 100.000) hinzugezogen und analysiert werden. Die zeitaufwändige Informationssuche wurde bisher manuell mit der betriebssystemeigenen Suche oder mit Standardsoftware, wie Adobe Acrobat, durchgeführt. Als ein weiteres Problemfeld konnte die hohe Einstiegshürde für neue Vertriebsmitarbeiter identifiziert werden, da sie genau wissen müssen, welche (technischen) Konzepte bei der Prüfung von Spezifikationen relevant sind und welche nicht. Erfahrene Vertriebsingenieure bauen hier auf ihr angesammeltes, implizites Wissen, sind jedoch auch nicht davor gewappnet, die Prüfung bestimmter Fakten zu versäumen.

Um die genannten Problemstellungen zu lösen, wurde die Ontos Linked Data Information Workbench[1] (LDIW) entsprechend der Anforderungen konfiguriert und erweitert. Die drei wesentlichen Beiträge zur Unterstützung der Arbeit des Vertriebes von ACC sind:

[1] `http://www.ontos.com/products/ontosldiw/` (23.06.2014)

- Automatische Analyse, Verlinkung und Indexierung multilingualer (Text-) Dokumente, u. a. in Englisch, Deutsch und Russisch, in verschiedensten Formaten, wie etwa Microsoft Word und Excel, PDF, Bilder, ...
- Unterstützung der multilingualen Wissensstrukturierung von relevanten Konzepten
- Benutzerfreundliches, effizientes User Interface zur Suche von Dokumenten, deren Relationen untereinander sowie von wesentlichen Konzepten in den Spezifikationen

Im Folgenden wird das Konzept und die technische Lösung vorgestellt, ehe im Anschluss auf die gewonnenen Erkenntnisse und das bisherige Kundenfeedback sowie die weitere Entwicklung der Software eingegangen wird.

2 Konzept und technische Lösung

Basis für die technische Lösung ist die Ontos LDIW, deren Bestandteile für das Kundenprojekt in Abbildung 1 überblicksartig angezeigt und in den folgenden Abschnitten kurz erläutert werden. Der Zugang für die Nutzer erfolgt in einer Portal-artigen Webanwendung, die einen Workflow über die verschiedenen Arbeitsschritte vom Hochladen der Dokumente und der Informationsextraktion bis hin zur eigentlichen Suche bietet.

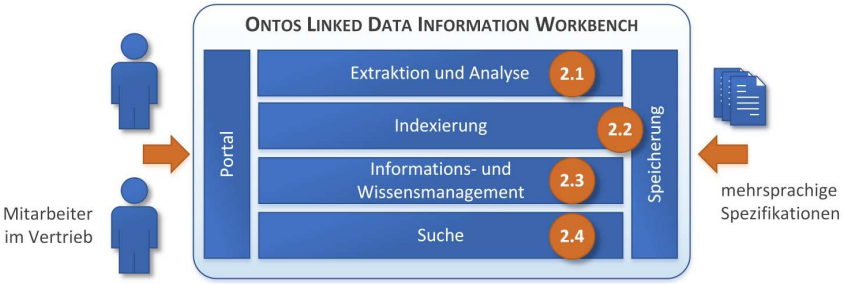

Abbildung 1: Ontos LDIW im Überblick mit Verweisen auf die beschreibenden Abschnitte

2.1 Extraktion und Datenanalyse

Die Verarbeitung der Spezifikationen läuft in parallelen Threads, wobei in jedem ein 3-stufiges Verfahren angewandt wird: 1) Homogenisierung und Textextraktion, 2) Vorverarbeitung und 3) Konzepterkennung. Nachfolgend werden diese näher erläutert.

Alle von den Vertriebsmitarbeitern bereitgestellten Dokumente werden zunächst in ein gemeinsames Format gebracht, um die Dokumente auch in Zukunft mit einem einheitlichen, verbreiteten Werkzeug öffnen und betrachten zu können. Hierbei fiel die Wahl auf das PDF-Format. Existieren Informationen nur in Bildern, was etwa durch den Scan alter Dokumente (z. B. Schaltpläne) bzw. aufgrund technischer Zeichnungen sehr oft der Fall ist, werden aus diesen mittels der freien OCR-Software tesseract-ocr[2] maschinenlesbare

[2] `https://code.google.com/p/tesseract-ocr/` (23.06.2014)

Texte extrahiert und diese ebenfalls als PDF bereitgestellt. Basierend auf dem homogenen Datenbestand werden anschließend mittels Apache Tika[3] die Texte seitenweise aus den Dokumenten gelesen.

Diese werden in zwei Schritten vorverarbeitet, um später eine möglichst genaue Analyse zu ermöglichen: Zunächst wird eine seitenweise Spracherkennung durchgeführt, da sich die Sprache nicht nur von Dokument zu Dokument sondern auch innerhalb einer Spezifikation ändern kann. Ein weiteres, in einem Vorverarbeitungsschritt zu lösendes Problem stellt die Entfernung von wiederkehrenden Texten, wie z. B. Kopf- oder Fußzeilen oder Kommentaren dar. Sind in diesen relevante Konzepte enthalten, würde deren Analyse ein Informationsrauschen erzeugen und so die Qualität der Suche verschlechtern.

Im letzten Schritt erfolgt die Erkennung der relevanten Konzepte in den vorverarbeiteten Texten. Grundlage ist ein multilingualer SKOS[4]-Thesaurus, dessen Erstellung in Abschnitt 2.3 näher erläutert wird. Aus diesem werden alle Bezeichner der definierten Konzepte in der Sprache des zu analysierenden Textes gelesen. Um ein effizientes Matching der Namen mit dem Text zu ermöglichen, wird der Aho-Corasick String Matching Algorithmus [AC75] eingesetzt.

2.2 Speicherung und Indexierung

Im Anschluss an die Datenanalyse erfolgt zunächst die Speicherung der beim Hochladen definierten Metainformationen, wie z. B. Name des zugehörigen Projektes oder die Nummer und Version des Dokumentes. Aber es werden auch die extrahierten Informationen, bspw. Sprache oder URIs der Konzepte, persistiert. Als Speicherlösung kommt der Ontos-eigene Triple Store OntoQUAD[5] zum Einsatz, der u. a. in Benchmarks aber auch in der Praxis seine Leistungsfähigkeit beweisen konnte [PPD+13]. Die Verwendung von RDF als Datenmodell und RDFS als Schemasprache bringt eine Vielzahl von Vorteilen mit sich, wie etwa eine leichte Erweiterbarkeit des Modells in späteren Projektphasen oder eine automatische Inferenz der Instanzdaten.

Um eine schnelle Suche in und Filterung von Dokumenten zu gewährleisten, wird die Apache Solr[6] Suchplattform eingesetzt. Sie bietet u. a. Volltextsuche, Faceted Browsing, oder die Indexierung von PDF-Dokumenten. Diese werden zusammen mit den auch in OntoQUAD gespeicherten Metainformationen hochgeladen, indexiert und stehen quasi in Echtzeit zur Suche bereit. Solr wurde durch das lucene-skos-Plug-In[7] erweitert, welches das Feature der Labelexpansion implementiert. Hierbei werden die Bezeichner des SKOS-Thesaurus erneut eingelesen und bei der Volltextsuche genutzt. Diese bringt bspw. den Vorteil, englischsprachige Begriffe auch in russischsprachigen Dokumenten zu finden. Weiterhin kommt bei der Indexierung das Semantic Vectors Package[8] zur Anwendung, was verschiedene Algorithmen, wie das Latent Semantic Indexing, beinhaltet. Diese be-

[3] http://tika.apache.org/ (23.06.2014)
[4] http://www.w3.org/2004/02/skos/ (23.06.2014)
[5] http://www.ontos.com/products/ontoquad/ (23.06.2014)
[6] http://lucene.apache.org/solr/ (23.06.2014)
[7] https://github.com/behas/lucene-SKOS (23.06.2014)
[8] https://code.google.com/p/semanticvectors/ (23.06.2014)

rechnen Vektorwortmodelle, mit denen Ähnlichkeiten zwischen Dokumenten anhand von gewählten Wörtern berechnet werden können.

2.3 Informations- und Wissensmanagement

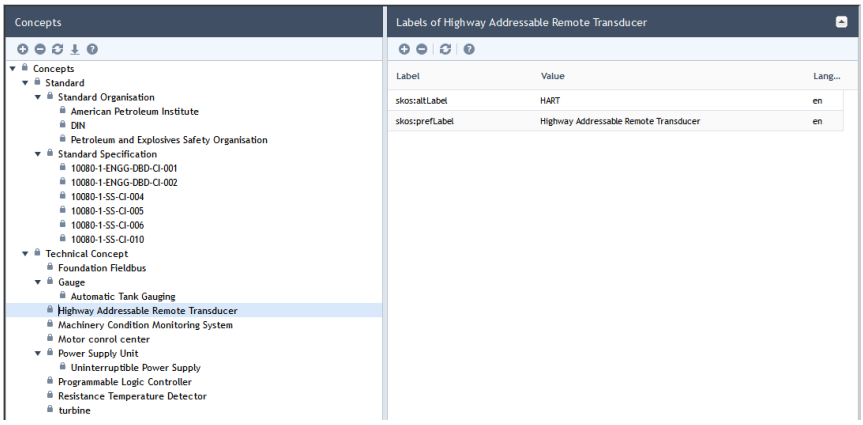

Abbildung 2: Beispielhafter, SKOS-basierter Thesaurus entwickelt in MiniDix

Für das Wissensmanagement kommt der webbasierte, mehrsprachige Ontologie-Editor OntoDix[9] zur Verwendung, der für die Erstellung von SKOS-basierten Thesauri deutlich vereinfacht wurde, um so auch eine höheren Akzeptanzgrad bei Nutzern ohne Semantic-Web-Erfahrung zu erzeugen. Wie Bild 2 zeigt, besteht er im Wesentlichen aus zwei Bereichen. Im linken Widget kann man neue Konzepte erstellen, existierende in der Hierarchie umsortieren oder auch entfernen. Auf der rechten Seite kann man für das jeweils gewählte Konzept die Bezeichner verwalten, wobei in bevorzugte (*skos:prefLabel*) und in alternative (*skos:altLabel*) Bezeichner unterschieden wird. Weiterhin können diese hier für unterschiedliche Sprachen definiert werden, was bei der späteren Suche den Mitarbeitern mit fehlenden Sprachkenntnissen zugutekommt.

2.4 Suche

Wie bereits genannt, wurde Apache Solr in die LDIW integriert und als Suchmaschine genutzt. Da die Anpassung von Solr an die Anforderungen des Projektes leicht zu realisieren waren, lag der Implementierungsschwerpunkt vor allem im Suchinterface, das neben einer Volltextsuche eine leichte Filterung über die Metadaten basierend auf dem Paradigma des Faceted Browsing [YSLH03] bieten soll. Um dem Kunden frühzeitig ein erstes User Interface bereitstellen zu können, das die wesentlichen Funktionen beinhaltet, wurde auf das in Solr integrierte, Template-basierte Framework Apache Velocity[10] zurückgegriffen. Bild 3 zeigt die Benutzerschnittstelle, die in vier eingefärbte Bereiche untergliedert ist: (1)

[9] `http://dix.ontos.ru/dix/` (23.06.2014)
[10] `http://velocity.apache.org/` (23.06.2014)

Feld zur Volltextsuche mit autocomplete-Funktion, (2) eine Auflistung der zur Filterung genutzten Facettenwerte, (3) Facetten und deren Facettenwerte sowie (4) eine Ergebnisliste. Letztere zeigt die Suchtreffer samt des von Solr automatisch aus den PDFs extrahierten Texts, der für die Volltextsuche genutzt wird, und relevante Metainformation, wie die gefundenen semantischen Konzepte, an. Suchtreffer werden automatisch hervorgehoben und helfen dem Nutzer bei der Bewertung der Ergebnisse. Hinsichtlich des User Interfaces konnten wir feststellen, dass die Suche schnell (Filteranfragen über ca. 2500 Testdokumente liegen im Schnitt bei ca. 40ms) und flüssig durchgeführt werden kann.

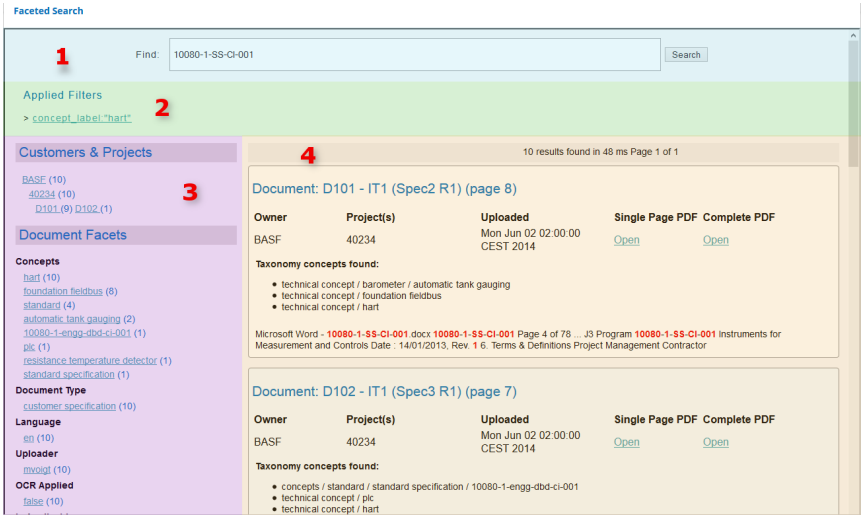

Abbildung 3: Facettierte Suche nach relevanten Dokumenten

3 Erfahrungen und geplante Weiterentwicklungen

Für die Umsetzung des ACC-Projektes wurde ein agiles, iteratives Vorgehen gewählt, so dass frühzeitig Prototypen vorliegen mit denen auch die Mitarbeiter des Kunden experimentieren und so hilfreiches Feedback geben können. Trotz anfänglicher Skepsis, die bspw. im Mehraufwand durch das Hochladen von Dokumenten und in der Einarbeitung in das neue Werkzeug begründet ist, wurde den Testern schnell der Mehrwehrt der neuen Suchlösung deutlich.

Bei der Umsetzung zeigten sich jedoch drei Probleme, denen im Vorfeld keine große Bedeutung beigemessen wurde, deren Behebung jedoch viel Zeit in Anspruch nahm:

Qualität der Spezifikationen Nach den ersten Tests der Verarbeitungspipeline mit realen Spezifikationen stellte sich heraus, dass deren textuelle Inhalte sich stark in der Qualität unterscheiden und so auch die Qualität der späteren Suche beeinflussen. So beinhalten manche Dokumente nur Verweise auf andere und sind somit weniger wichtig, wiederkehrende Kopf- und Fußzeilen oder kopierte Textpassagen sorgen für Rauschen, Sprachen werden im Dokument oder gar in einer Seite vermischt.

Erstellung einer Wissensbasis Die Entwicklung einer semantischen Wissensbasis stellte ebenfalls eine große Hürde dar. So fällt es auch sehr erfahrenen Ingenieuren schwer, das implizite Wissen zu externalisieren. Hierzu gehört einerseits die Identifikation wesentlicher Konzepte aber andererseits auch deren Strukturierung. Zur Unterstützung des Prozesses wurde der Ontologie-Editor OntoDix auf das Notwendigste reduziert. Weiterhin wurde ein einführender Workshop zum Thema Wissensstrukturierung mit Mitarbeitern durchgeführt.

Entwicklung des Suchinterfaces Letztlich wurde angenommen, dass die facettierte Suche schnell zu realisieren ist. Wie oben beschrieben, bot sich Apache Velocity sehr gut an. Jedoch wurde bei Tests mit Mitarbeitern und Echtdaten sofort deutlich, dass eine individuellere Funktionalität notwendig ist. So wird bspw. die Menge der Metainformationen, aus denen die Facettenwerte für die Filterung extrahiert werden, so groß sein, dass sie nicht über einfache Listen abbildbar sind. Weiterhin wird auch eine direkte Vorschau der Suchergebnisse im PDF benötigt, so dass die Ingenieure den Kontext besser erfassen können. Diese und weitere clientseitge Funktionen wie die Integration des Latent Semantic Indexing lassen sich durch die fehlende AJAX-Unterstützung in Velocity schwierig oder nicht realisieren.

Die zuletzt genannte Problemstellung stellt auch den aktuellen Arbeitsschwerpunkt dar. Auf Basis des Ajax-Solr-Frameworks[11] wird eine Widget-basierte, leicht konfigurier- aber auch erweiterbare Suchoberfläche entwickelt, die Apache Velocity ersetzen wird. Nach deren Fertigstellung wird dieses User Interface und das Gesamtsystem einem breiteren, längeren Nutzertest sowie einem Lasttest mit großen Datenmengen unterzogen.

Weitere künftige Erweiterungen in der LDIW sind die Konzeption und Realisierung von Mechanismen zur Disambiguierung von gefundenen semantischen Konzepten sowie die Integration von Crawling-Methoden, um automatisch Ausschreibungen aus dem Web dem System zuzuführen. Und es wird auch eine Integration der Daten aus dem ERP sowie des CRM erwogen, was durch die Nutzung von W3C-Standards und semantischen Technologien gut realisierbar ist. Die Kundendaten sollen durch im Web verfügbare Nachrichten mittels der genannten Crawling-Mechanismen extrahiert und angereichert werden.

Literatur

[AC75] Alfred V. Aho und Margaret J. Corasick. Efficient String Matching: An Aid to Bibliographic Search. *Commun. ACM*, 18(6):333–340, Juni 1975.

[PPD+13] Alexander Potocki, Anton Polukhin, Grigory Drobyazko, Daniel Hladky, Victor Klintsov und Jörg Unbehauen. OntoQuad: Native High-Speed RDF DBMS for Semantic Web. In *Knowledge Engineering and the Semantic Web*, Jgg. 394 of *Communications in Computer and Information Science*, Seiten 117–131. Springer Berlin Heidelberg, 2013.

[YSLH03] Ka-Ping Yee, Kirsten Swearingen, Kevin Li und Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '03, Seiten 401–408, New York, NY, USA, 2003. ACM.

[11] `https://github.com/evolvingweb/ajax-solr` (23.06.2014)