

3. Zwischenreport 13. 09. 2007

1.1. Einleitung

Das Projekt "Experimentelle Knowledge-Engine der österreichischen Zivilgesellschaft im Informationszeitalter" fokussiert auf experimentellen Technologien, mittels denen von traditionellen Knowledge Engines schlecht erfassten Datensätze, etwa Blogs und Email Listen, besser erfasst und genutzt werden können.

Die erste Phase des Projektes, die sich von Anfang Dezember 2006 bis Ende Januar 2007 erstreckte, war der Evaluation und dem Aufbau der technischen Basis-Infrastruktur gewidmet. In der zweiten Phase, die sich von Anfang Februar bis Mitte März erstreckte, wurden Informationen mittels standardisiertem Fragebogen und ExpertInneninterviews eingeholt, wie und in welchem Umfang "nicht-traditionelle" Medien von den Akteuren der Zivilgesellschaft genutzt werden. In der dritten Phase des Projects, die jetzt zu Ende geht, wurde der Search Engine eingerichtet, in eine Serverumgebung eingebaut, mit Backend-Schnittstellen zu Zope und mit Testdaten gefüttert. Der Search Engine ist nun in einer Alpha Version online. In dieser Version ist er erst mit Test-Daten gefüttert und noch nicht auf End-user ausgerichtet. Im Moment sind wir voll damit beschäftigt, die internen Prozesse des Search Engines, die Schnittstellen zum Backend und Frontend sowie die Data-Retrieval Prozesse zu optimieren. Diese Prozesse haben sich als technisch komplexer als erwartet herausgestellt, weil es vielzahl von unterschiedlichen Programmen angepasst und kompatibel gemacht werden müssen.

Dies wird zur Beta Version führen, die öffentlich vorgestellt wird.

Dieser 3 Zwischenreport fokussiert auf diese Arbeiten der Technologieentwicklung.

1.2. Temporäre Adresse des Engines, Alpha Version mit Testdaten

Seit Mitte August ist die Alpha Version für interne Entwicklungszwecke online (und auch von Außen erreichbar). Im aktuellen Stand sind erst Testdaten in der Datenbank, zum die Funktionalität des Engines und seine Einbindung in die Zope-Umgebung optimieren.

Url: <http://wii.t0.or.at/wii/retrieval>

2. Technische Beschreibung des Engines

Das WII Searchtool ist eine Text-retrieval Lösung um nach dem Bayesischen Modell der probabilistischen Inferenz Bedeutung in die automatische Klassifizierung von Dokumenten zu integrieren. Bedeutung wird meist in kurzen Wortmustern ausgedrückt und das WII Suchwerkzeug unterstützt automatische Erkennung von Mehrwortkonzepten, um diese als Grundlage für das Suchen zu verwenden. Isolierte einzelne Wörter sind in hohem Grade vieldeutig und führen zu Ergebnissen mit niedriger Präzision. Wenn klassisch mehrwortige "Phrasen" zum Suchen verwendet werden um die Suchgenauigkeit zu verbessern, geht dies auf Kosten der Retrieval Qualität, da jedes mögliches Dokument, das nicht die genaue Wortkombination enthält, ignoriert wird.

Verglichen zu Search Engines, die Indexe von einzelnen Keywords verwenden, bedarf es einer höheren Genauigkeit und einer besseren Retrievalrate durch entsprechende Klassifizierung von Resultaten. Für erhöhten inhaltlichen Vergleich und verbesserten Dokumentzugriff können passende Mehrwortphrasen identifiziert und ermittelt werden. Während traditionelle Systeme

die auf einfachen Schlüsselwörtern und Booleschen Logik basieren sehr exakt sind, ist, wenn die Zahl der Dokumente die aufgefunden werden zu groß ist vor allem die Fähigkeit, Dokumente, nach Wichtigkeit zu ordnen von großem Wert. Das Wahrscheinlichkeitsmodell erlaubt ein statistisches Modell der Keyword Frequenz über die gesamte Dokumentensammlung. Dieses Modell ermöglicht nicht nur genauere Ausgangsergebnisse, sondern es stellt auch einen Mechanismus für wiederholtes Suchen basierend auf Relevanz-Feedback.

Schlüsselpunkte sind Bedeutungsklassifizierung und Konzeptidentifizierung nach Shannon's Informationstheorie und Probabilistic Latent Semantic Indexing. Das letztere ist die Fähigkeit, Dokumente zu lokalisieren, die zur Benutzeranfrage Bedeutung haben, selbst wenn manche Wörter nicht im Benutzeranfragetext enthalten sind und beinhaltet gleichzeitig die Fähigkeit Dokumente zu ignorieren, die Wörter von der Benutzeranfrage enthalten, aber nicht relevant sind. Dies wird durch die Reihung der Relevanz der Dokumente der Ausgangssuche erzielt. Die entscheidenden Konzepte der relevantesten Dokumente werden extrahiert und die Suche erweitert, um eine Auswahl in Verbindung stehende Konzepte einzuschließen. Diese Funktionen lassen sich personalisieren und automatisieren. Die Klassifikationsfunktionen ermöglichen die Anwendung von Thesauren und Dictionaries sowie Experten-Taxonomien und Ontologien um Dokumente zu Beschreiben und zu Verknüpfen.

Das Modul basiert auf einer offenen Architekturstrategie und ermöglicht eine einfache Integration von Anwendungsschnittstellen (APIs) auf der Basis von XML und Webservices. Plattformkompatibilität über Webservices und Programm-Anwendung Schnittstellen (APIs) basieren auf XML erlauben transparenten Zugang zu den System Internals einschließlich dem statistischen Profil von Keywords und Auswertungen.

3. WII-Searchtool Backend / Zope Interface

Für eine erste Implementierung des auf Python/Zope/Plone basierenden Frontends waren einige Hürden zu überwinden. Vor allem die Anbindung an den Query-Server via Web Service verursachte Schwierigkeiten, denn der Query-Server der Search-Engine implementiert entgegen anders lautender Angaben den XML Standard SOAP nur sehr eingeschränkt.

Wenn das Frontend auf eine Methode in der API der Search-Engine zugreifen will, müssen zuerst mehrere Argumente an den Query-Server gesendet werden, bei erfolgreicher und fehlerfreier Interpretation der Query wird ein mehr oder minder komplexes Resultat-Set retourniert. Diesen Prozess könnte die Python-Library "SOAPpy" inklusive Typ-Konvertierung einfach und vollständig abwickeln. Leider funktioniert die SOAP-Implementierung des Query-Servers anders: Die einzelnen Query-Argumente und Resultat-Komponenten können nicht etwa vollständig via SOAP-Typen/Methoden abgewickelt werden, sondern es kann nur ein einziges String-Argument an den Query-Server übergeben werden, welches die effektiven Argumente in Form von XML enthält. Bei erfolgreicher und fehlerfreier Interpretation des XML-String-Arguments wird wiederum ein einziger Result-String retourniert, der ebenfalls die einzelnen Resultset-Komponenten in Form von XML enthält.

Da das SOAP-Protokoll auch in Form von XML kommuniziert, muss der XML-String der Query-Argumente in einen CDATA-Abschnitt gewrapped werden. Diese Einbettung von XML in XML führt zu verschiedenen Problemen - insbesondere im Zusammenhang mit Zeichensätzen. Außerdem müssen für jede Methode in der API der Search-Engine individuelle XML-Generatoren/Parser erstellt werden. Diese Schnittstellen- und Standardisierungsprobleme haben zu einer nicht unwesentlichen Verzögerung des Projekts geführt, sind nun aber gelöst und die Einpassung des WII Searchtool backend in das Zope Frontend weitgehend reibungsfrei.

4. Datenquellen

Um die zivilgesellschaftlichen Datenquellen, die durch den Search Engine besser zugänglich gemacht werden sollen, Email Listen und Blogs, zu erfassen sind 3 Schritte notwendig. Ersten müssen die bestehenden Archive der Listen und der Blogs, indiziert werden, zweitens müssen die jeweils aktuell dazukommenden Beiträge in Echtzeit erfasst werden und drittens müssen die Daten in XML konvertiert werden, damit sie mit den relevanten Metadaten in die Datenbank eingepflegt werden können (Zur XML Struktur, siehe das Schema im ersten Zwischenbericht "2.3.3 Installieren und testen der realtime update scripts").

Eine Liste der ersten Sammlung an Blogs und Emailisten, die in der Alpha Version des Search Engines bereits erfasst ist, ist online.

<http://wii.t0.or.at/wii/retrieval/indexed>

Nach längeren Recherchen und testen diverser Tools und Bibliotheken wird nun eine hybride Strategie eingesetzt. Mit klassischen UNIX Kommandozeilen Werkzeugen wie curl und wget, textbasierten Browsern wie lynx und w3m werden die meist höchst unterschiedlichen Layouts von Weblogs umgangen und der Content als Plaintext durch das WII Searchtool indiziert. Dazu waren weitere Python Programme nötig die an die einzelnen Layouts angepasst wurden. In weiterer Folge werden die hierbei gesammelten Erfahrungen genutzt, um diesen Prozess weitestgehend zu automatisieren. Die Einträge und E-Mails werden in zwei Formaten (plain text und xml) lokal gespeichert. Der Vorteil eines lokalen Backups dieser Inhalte in einem modifizierten mbox Format einerseits und als XML andererseits besteht außerdem darin dass die bereits vorhandenen Daten ohne größeren Aufwand auch in andere Datenbanksysteme oder Contentindizierungssoftware eingebucht werden kann. Weiters bleiben die Texte auch nach dem Löschen durch Betreiber von Listen oder Weblogs für die Suche erhalten.

Mit RSS feeds wird analog verfahren. Mit der Python Bibliothek Feedparser (<http://feedparser.org/>) steht ein leistungsfähiges Framework zur Konvertierung von live Updates zur Verfügung. UNIX cronjobs stellen alle 5 Minuten sicher dass die Datenbank aus den gewählten Quellen auf dem neuesten Stand ist.

5. Interface

Datumseingrenzung

Eines der wesentlichen Probleme mit bestehenden Knowledge Engines ist, dass sie die Dokumente, die sie indizieren zeitlich nicht einordnen können. Dies führt dazu, dass auch auf aktuelle Queries veraltete Daten als relevant eingeschätzt werden. In unserem experimentellen Knowledge Engine wurde die Datumsindizierung als zentrales Element in ins Design eingebaut und es ist nun möglich, die Suche zeitlich einzugrenzen und die Ergebnisse nicht nur nach Relevanz, sondern auch nach Datum ausgeben zu lassen.

Related Topics

Oftmals ist es so, dass ein Suchbegriff, den der Nutzer eingibt, den relevanten Suchraum nur ungenau abdeckt, sei es, dass er zu spezifisch, einseitig gewichtet, oder zu ungenau ist. Das ist

besonders relevant, bei heterogenen Datensätzen, wie wir sie verarbeiteten, die oftmals dieselben Begriffe unterschiedlich benutzen. Um dem Nutzer zu helfen, seine oder ihre Suche zu verbessern ("query expansion"), bieten wir so genannte "related topics" an, die vom Knowledge Engine aufgrund der Analyse der gefundenen Dokumente in Echtzeit erstellt wird.

In der Alpha Version haben wir die Analysekriterien, nach denen diese Topics erstellt werden sichtbar gemacht. Hinter jedem Topic sind 3 Variablen dargestellt.

- The "wrf" [1. variable] element shows the number of times the terms appears in the current page of the hitlist (within results frequency).
- The "wcf" element [2.variable] shows the number of documents indexed by this term (within collection frequency).
- The "tw" [3. variable] shows the standard probabilistic weighting for this term.

In der Endversion werden diese Variablen nicht mehr im Interface dargestellt werden, weil sie für den Endnutzer nicht lesbar sind. Für die Entwicklung ist es aber sinnvoll sie anzuzeigen.

6. Ausblick: Final Stage und Final Report

Für den Abschluss der Arbeiten am Knowledge Engine im Rahmen des IPA Projektes werden wir uns auf folgende Elemente konzentrieren:

Backend-Optimierung

Optimierung des Scripts, die die Datensätze einholen, formatieren und zur Indizierung bereit stellen. Ziel ist es, die Effizienz dieser Skripte zu steigern und damit System schneller und skalierbarer zu machen

Optimierung der Features des Knowledge Engine, insbesondere der Parameter, die die "related topics" generieren

Verbreiterung der indizierten Datensätze

Redaktionelle Auswahl

Wir werden zunächst noch weitere, von uns redaktionell ausgewählte Datensätze indizieren und bearbeiten lassen.

Offenes Formular, um relevante Quellen zur Indizierung vorzuschlagen

In so vielschichtigen Themenfelder, wie diejenigen, die von der Österreichischen Zivilgesellschaft bearbeitet werden, nicht sinnvoll ist, eine zentrale Auswahl aller relevanten Datenquellen zu machen. Um dieser Diversität, und dem darin enthalten spezifischen Fachwissen Rechnung zu tragen, werden wir ein Formular anbieten, mittels dessen die interessierte Öffentlichkeit weitere Quellen zur Indizierung vorschlagen kann. Die Auswahl wird aber nach wie vor redaktionell bearbeitet, um den spezifischen Charakter des Knowledge Engines zu bewahren.

Personalisierungsoptionen

Wir arbeiten aktuell an einer Option, die es erlaubt, individuelle Searches als RSS-Feed zu abonnieren, und somit bequem jeweils die aktuellste Diskussion zu einem Thema quer über aller Quellen mitverfolgen zu können.

Redesign

Die aktuelle Alpha Version ist noch nicht graphisch ausgeführt, da sie primär internen Testzwecken dient. Die Betaversion wird neu designed, um sie benutzerfreundlicher zu machen.

Soft Launch

Die Betaversion wird "sanft" lanciert. Da es eine Beta Version ist, wird die Nützlichkeit für reine Anwender noch eingeschränkt sein, aber wir werden interessierte Kreise in der Österreichischen Zivilgesellschaft ansprechen, um sie zur experimentellen Nutzung und Weiterentwicklung zu gewinnen.

Mit dem Soft Launch wird auch der Final Report veröffentlicht werden. Die Erfahrungen an diesem Projekt haben uns gezeigt, dass es ein großes Potential gibt, spezialisierte Knowledge Engines zu entwickeln. Die Zusammenarbeit mit der Firma Matrixware hat sich bewährt und es besteht beiderseitiges Interesse, hier weiter zusammen zu arbeiten. Dazu werden wir in nächster Zukunft ein neues Projekt starten, im Zuge dessen die experimentelle Beta Version des Knowledge Engines zu einer auch für die Endnutzer attraktiven Version 1 ausgebaut werden kann.