HUMBOLDT-UNIVERSITÄT ZU BERLIN

Exposé der Studienarbeit:

Aufbau eines Flexionslexikons für die Katalogbereinigung

Eingereicht von: **Johannes Kozakiewicz**
Institut für Informatik
Humboldt-Universität zu Berlin
Matr.Nr.: 186778
kozakiewicz@gmx.de

Erstbetreuer **Herr Prof. Dr. Ulf Leser**
Institut für Informatik
Humboldt-Universität zu Berlin
Knowledge Management in Bioinformatics
leser@informatik.hu-berlin.de

Betreuer: **Herr Dr. Michael Eimermacher**
EITCO, Berlin
MEimermacher@eitco.de

Berlin, den 20.07.2007

Exposé Aufbau eines Flexionslexikons für die Katalogbereinigung2

1. Einleitung

Der Eitco*Scout* bündelt eine Reihe von Technologien aus Statistik und Künstlicher Intelligenz für individuelle Onlineberatung, Diagnoseunterstützung, intelligente Suche etc. Da in derartigen Anwendungen oft große Kataloge eingesetzt werden, wird zurzeit von der EITCO GmbH eine weitere EitcoScout-Komponente entwickelt, die der Optimierung der Struktur großer Kataloge dienen soll. Dabei sollen u.a. ähnliche Artikel identifiziert und Dubletten nach unterschiedlichen Strategien gesperrt werden können.
Im Rahmen der Vorverarbeitung der Katalogdaten wird jedoch zusätzliches lexikalisches Wissen benötigt. In diesem Bereich ist die Studienarbeit angesiedelt.

Da gleiche oder ähnliche Artikel auf unterschiedliche Art und Weise beschrieben werden können, ist es notwendig alle bedeutungstragenden Wörter auf einheitliche Begriffe abzubilden. Da für wird ein Flexionslexion benötigt, welches mittels verschiedener Generatoren/Filter von mir erzeugt wird. Es sollte nach Fertigstellung in tabellarischer Form vorliegen und möglichst viele Wörter mit der korrekten Grundform enthalten. So werden z.B. alle denkbaren Flexionsformen (z.B. sage, sagte, sagst …) zu einer Grundform (z.B. sagen) von dem **Regelbasierten Generator** automatisch erzeugt. Und mit einem weiteren „Stoppwort"-Lexikon werden alle „überflüssigen" Wörter ausgefiltert.

Wie die vorhandenen Katalogdaten wird auch das Flexionslexikon mit sämtlichen erzeugten Wortformen in eine *Oracle*-Datenbank über die bereits existierende Importfunktion in das Gesamtsystem importiert. Meine Studienarbeit wird auf bereits vorhandene frei verfügbare Lexiken aufbauen, aber das dabei entstehende Lexikon soll wesentlich mehr Wörter und Wortformen enthalten.

Dabei sind Regeln zu definieren, wie Wortarten und Wortgruppen innerhalb einer Wortart verändert werden dürfen. Diese morphologischen Flexionsregeln definieren korrekte Worttransformationen, die über einfaches Abschneiden oder Anfügen von Wortendungen (Suffixen) hinausgehen (so genanntes „Stemming") [MEM]. Vor allem Änderungen im Wortstamm (z.B. „der Vogel", aber „die Vögel") und Ausnahme-Regelungen (z.B. Präteritum von „senden" ist „sendeten" und „sandten"), wie sie in natürlichen Sprachen häufig vorkommen, müssen beachtet werden. Hinzu kommt, dass die neue Rechtschreibung beachtet werden muss [DUD].

2. Aufgabenstellung

Die Aufgabe des Flexionslexikons ist Überprüfung von Tokens aus den Artikelbeschreibungen der einzelnen Kataloge auf Schreibfehler und anschließender Flexionsreduktion (Lemmatisierung) der Wortformen zur Vereinheitlichung der Tokens.

Ziel meiner Studienarbeit ist der Aufbau dieses Lexikons, welches, um einen schnellen Abgleich mit einem Token zu ermöglichen, alle Wortformen als Vollformen enthält. Platzeffizienter wäre natürlich die Grundform mit einem Satz an global definierten

Flexionsregeln aufzunehmen, allerdings verringert dies die Verarbeitungsgeschwindigkeit erheblich, da alle Formen zur Laufzeit erzeugt werden und im Arbeitsspeicher gehalten werden müssen.

3. Vorgehen

Zur Erzeugung des Lexikons ist ein Basissatz an Wörtern notwendig. Dazu sollen frei verfügbare Daten sowie bereits existierende Lexiken und Stammformenlisten genutzt werden. Eine Grundlage meines Lexikons ist Ispell [LASR], welches zurzeit das noch am häufigsten benutzte Spellchecker-Programm für Unix und Linux ist.

Dessen Lexikon besteht aus zwei Teilen

- Wortformen mit Flags für Transformationsregeln
- Transformationsregeln beschreiben, wie die Wortformen verändert werden, um aus Grund- oder Stammformen Flexionsformen zu generieren.

Damit entsprechen die Flags indirekt Flexionsreihen. Die Beziehung ist jedoch nicht immer eindeutig, daher muss zunächst geklärt werden, welche Flags eindeutig auf eine Wortart weisen und welche allgemein angewendet werden können.

Für die eindeutigen Fälle werden folgende Schritte realisiert:

- Zuordnung der Wortart
- Generieren der Flexionsformen für diese Wortart
- Speichern der Relation Flexionsform – Wortart – Grundform.

Für alle mehrdeutigen Fälle sind weitere Filter nötig. Das gilt insbesondere für Stammformen: Stammformen werden in diesem Bestand oft nicht aus ihrer Grundform abgeleitet, wenn sie „stark“ sind, also im Bereich des Stammes von der Grundform abweichen. Dann erhalten sie einen eigenen Eintrag ohne Bezug zur Grundform.
Zur Überprüfung der Rechtschreibung reicht dies aus, zur Reduktion von natürlich-sprachlichen Texten allerdings nicht. Hierfür wird eine weitere Liste benötigt, die die korrekte Zuordnung von Stammformen auf ihre Grundform ermöglicht.
Ein weiteres Problem ist, dass Ispell die Worte allein aufgrund syntaktischer Ähnlichkeit generiert, d.h. die Semantik von Wörtern nicht berücksichtigt. Zum Beispiel wird aus dem Substantiv „Stör“ auch „Störung“ abgeleitet oder aus dem Adjektiv „bar“ auch die flektierte Verbform „gebar“ erzeugt. Solche Zuordnungen lassen sich nur schwer oder gar nicht automatisch erkennen, sodass hier manuell korrigiert werden muss.

Sofern die Zuordnung Stammform ⇨ Grundform innerhalb des Bestandes nicht möglich ist, sind folgende Schritte zu prüfen:

- Abgleich mit einer existierenden Tabelle für starke Flexionsformen
- Prüfen, ob die Wortform als Kompositum zerlegt werden kann. Für diesen Fall: Ableitung des letzten Teilwortes auf das Kompositum übertragen.
- Ausgeben und Einlesen einer Liste mit den generierten Wortformen zur manuellen Überprüfung

Da unsere Quelle für starke Flexionsformen viele falsche Flexionsformen enthält, können wir diesen Bestand nicht einfach importieren, sondern können nur die Zuordnung Stammform ⇨ Grundform benutzen. Der Schnitt beider Quellen bildet dann wiederum eine korrekte Liste.

Automatische Korrekturen kann man dort benutzen wo spezifische Suffixe, Präfixe oder Substrings auftreten. So lassen sich

- bei Wörtern mit Suffixen wie –ung, -heit, -keit meistens auch die Suffixe –ungen, -heiten und –keiten generieren.
- bei Wörtern, die durch Ispell mit einem Präfix versehen wurden, eigenständige Basisformen samt dazugehöriger Flexionsformen erzeugen (falsche Flexionsformen müssen anhand einer Verbotsliste unterbunden werden)
- Wörter mit bestimmten Infixen einer anderen Basisform zuordnen, z.B. beim erweiterten Infinitiv mit –zu oder beim Partizip II durch den Infix –ge

Eventuell lassen sich während der Ausführung dieses Teils der Studienarbeit noch weitere Möglichkeiten finden, Korrekturen zu automatisieren.

Eine weitere Grundlage werden Wortlisten aus dem Morphy-Lexikon sein [IMS]. Diese wurden bereits nach Wortarten getrennt abgespeichert. Allerdings fehlt auch hier eine Zuordnung von im Stamm veränderter Wortformen zur korrekten Grundform. Da die Wörter aber nach Wortart getrennt sind, kann hier eine Zuordnung aufgrund grammatischer Regeln erfolgen. So kann eine automatische Zuordnung durch

- Überprüfung des Wortstamms nach vorheriger Umlaut-Reduktion („ä“ zu „a“ etc.) oder Austausch des 1.Vokals („i“ zu „o“ etc.) erfolgen
- Überprüfung des Suffixes (z.B. „en“ impliziert Grundform bei Verben) erfolgen
- Überprüfung, ob ein Präfix beliebiger Länge mit einem anderen Wort oder eines Präfixes des Wortes übereinstimmt, erfolgen

Eventuell können hier ebenfalls weitere Möglichkeiten zur Automatisierung gefunden werden. Da ein solch naiver Ansatz aber auch fehlerhafte Zuordnungen erzeugt, müssen die Listen vermutlich anschließend noch manuell korrigiert werden.

4. Technische Anforderungen

Da das Flexionslexikon auf mehreren Roh-Lexika und Listen beruht, ist eine schnelle Verarbeitung dieser Quellen und der erzeugten Zwischenergebnisse wünschenswert. Aus diesem Grund werden überwiegend die auf Hash-Funktionen basierten Java-eigenen Strukturen (wie Hashtable oder HashMap) verwendet, die ein schnelles Auffinden von bestimmten Lexikoneinträgen ermöglichen. Weiterhin kommen gewöhnliche Container, wie Sets (um komplette HashMaps abzubilden) oder Arraylisten (für dynamische Speicherallokation und variable Elementemengen), zum Einsatz.
Um einen schnellen Zugriff auf die Daten zu gewährleisten, ist es weiterhin erforderlich während des Erzeugungsvorgangs das Flexionslexikon sowie die aktuell zu verarbeitenden Daten komplett im Hauptspeicher zu halten.

5. Referenzliste

[MEM] Wortorientiertes Parsen - Dissertation
Eimermacher, M.; Berlin, 1988
(allgemeine Flexionsregeln zu Verben, Substantiven, Adjektiven und Partizipien)

[DUD] http://www.duden.de/deutsche_sprache/neue_rechtschreibung/

[LASR] http://lasr.cs.ucla.edu/geoff/ispell-dictionaries.html#German-dicts

[IMS] http://www.ims.uni-stuttgart.de/projekte/corplex/paper/lezius/molympic.ps.gz