Extraktion, Zuordnung und Normierung der Arbeitsschritte einer großen, heterogenen Rezeptdatenbank - Hinweise zum Quellcode

Manuel Müller

18. Dezember 2012

Inhaltsverzeichnis

1 Einleitung

Im Folgenden wird zunächst eine kurze Anleitung zur Installation des Systems zur *Extraktion, Zuordnung und Normierung der Arbeitsschritte einer großen, heterogenen Rezeptdatenbank* gegeben.

Danach soll die Nutzung und Funktionsweise anhand einiger Beispiele erläutert werden.

Momentan läuft das System auf einem Server der Professur für künstliche Intelligenz. Ein Web-Interface zur Demonstration der Nährwertfindung kann unter *http://faui8183.informatik.uni-erlangen.de:8080/vamos/nutridemo/* erreicht werden.

2 Voraussetzungen

- Die Programmiersprache *Python* zur Interpretation des Systems
- Ein *MySql-Server* zur Verwaltung der Datenbanktabellen
- Das Python-Interface *MySQLdb*, das die Verbindung zu der Datenbank ermöglicht
- Der *Treetagger* des *Instituts für Maschinelle Sprachverarbeitung Stuttgart*, um Wörtern Wortarten zuzuordnen
- Der *Treetagger Python-Wrapper*, der ebenfalls auf den Seiten des *Instituts für Maschinelle Sprachverarbeitung Stuttgart* bezogen werden kann

3 Installation

Zuerst müssen alle Tabellen, die auf dieser CD im Verzeichnis *tables* gespeichert sind, importiert werden. Die Verzeichnisse *src* und *lists* können dann an einen geeigneten Ort kopiert werden. Danach lässt sich die Datei *config.ini* im Verzeichnis *src* folgendermaßen anpassen:

Die Einträge *abbreviations*, *stopwords*, *containerwords* und *nouns* geben an, wo genau die im Verzeichnis *lists* enthaltenen Dateien gespeichert sind.

Zugangsdaten für die zu verwendende Datenbank werden mittels der Einträge *Database_Host*, *Database_User*, *Database_Password* und *Database_DatabaseName* spezifiziert.

Die Namen der Datenbanktabellen werden in den Einträgen *IngredientsTable*, *IngredientsRulesTable*, *QuantityRulesTable*, *ProcedureListMainGroup*, *ProcedureListSubGroup*, *ToolsListMainGroup*, *ToolsListSubGroup*, *DescriptionReplacemen-*

tRules, *TimeQuantities* und *TimeUnits* angegeben. Falls die Tabellen-Namen beim Import in die Datenbank nicht verändert wurden, können die Einträge unverändert übernommen werden.

Durch *tagdir* wird angegeben, in welchem Verzeichnis der *Treetagger* gefunden werden kann.

Auch Kommentare sind in der Konfigurationsdatei möglich, das Zeichen # dient zu ihrer Kennzeichnung.

Im Verzeichnis *raw* dieser CD sind einige Beispielrezepte gespeichert, anhand derer die Funktion des Systems nun getestet werden kann.

4 Anwendungs-Beispiele

Im Folgenden sollen einige Anwendungsbeispiele zeigen, wie das System verwendet werden kann.

Zunächst wird das System importiert:

```
from readprep import *
```

Nun kann eine Rezept-Datei geladen werden:

```
rezept = ZubereitungsBeschreibung("1.raw")
```

Über den Werkzeugmanager können nun Ober- und Untergruppe aller gefundenen Werkzeuge ausgegeben werden:

```
for werkzeuge in rezept.werkzeugMan.werkzeuge:
        print werkzeuge.untergruppe
        print werkzeuge.obergruppe
```

Der Zeitmanager speichert einerseits die in den Rezepten explizit angegebenen Zeitkosten:

```
print rezept.zeitMan.zeitDirekt
```

Andererseits kann die Zeitkostenschätzung aller nötigen Arbeitsvorgänge ausgelesen werden:

```
print rezept.zeitMan.vorgangsZeitverbrauch
```

Alle gefundenen Zutaten können über den Zutatenmanager ausgelesen werden:

```
for zutaten in rezept.zutatenMan.zutaten:
        print zutaten.wortlaut
```

Jede Zutat speichert zusätzlich, welche Vorgänge in Relation stehen:

```
for alleGefundenenZutaten in rezept.zutatenMan.zutaten:
        print alleGefundenenZutaten.wortlaut
        for zugeordneteVorgaenge in alleGefundenenZutaten
           .vorgangsRelationen:
                print "␣–␣" + rezept.vorgangsMan.
                   getVorgang(zugeordneteVorgaenge).
                   wortlaut
```

Natürlich kann umgekehrt auch von allen gefundenen Vorgängen ausgegangen werden, um dann auszulesen, welche Zutaten in Relation stehen:

```
for alleGefundenenVorgaenge in rezept.vorgangsMan.
   vorgaenge:
        print alleGefundenenVorgaenge.wortlaut
        for zugeordneteZutaten in alleGefundenenVorgaenge
           .zutatenRelationen:
                print "␣–␣" + rezept.zutatenMan.getZutat(
                   zugeordneteZutaten).wortlaut
```