

Automatisierte Berechnung von Nährwerten einer großen, heterogenen Rezeptdatenbank - Hinweise zum Quellcode

Manuel Müller

Lehrstuhl für Informatik 8: Künstliche Intelligenz

22. Mai 2012

Inhaltsverzeichnis

1	Einleitung	1
2	Installationshinweise	1
3	Erläuterung der Klassen und ihrer wichtigsten Funktionen	2
3.1	Überblick	3
3.2	Klasse: Rezept	3
3.2.1	Beschreibung	3
3.2.2	Wesentliche Variablen	3
3.2.3	Wesentliche Funktionen	4
3.3	Klasse: Rezepteintrag	4
3.3.1	Beschreibung	4
3.3.2	Wesentliche Variablen	4
3.3.3	Wesentliche Funktionen	5
3.4	Klasse: Zutat	5
3.4.1	Beschreibung	5
3.4.2	Wesentliche Variablen	5
3.4.3	Wesentliche Funktionen	5
3.5	Klasse: Datenbank	6
3.5.1	Beschreibung	6
3.5.2	Wesentliche Variablen	6
3.6	Klasse: Regeln	6
3.6.1	Beschreibung	6
3.6.2	Funktionen	7
4	Anwendungsbeispiele	7

1 Einleitung

Im Folgenden wird zunächst eine kurze Anleitung zur Installation des Systems zur automatisierten Berechnung von Nährwerten einer großen, heterogenen Rezeptdatenbank gegeben.

Danach sollen zum besseren Verständnis des Quellcodes die Klassen und deren wichtigste Funktionen erläutert werden. Nicht erklärt wird hier allerdings das zugrunde liegende Verfahren, so muss die genaue Herangehensweise des Systems aus der Beschreibung dieser Studienarbeit, die ebenfalls auf der CD enthalten ist, entnommen werden.

Momentan läuft das System auf einem Server der Professur für künstliche Intelligenz. Ein Web-Interface zur Demonstration der Nährwertfindung kann unter <http://fau8183.informatik.uni-erlangen.de:8080/vamos/nutridemo/> erreicht werden, Abbildung 1 zeigt ein Bildschirmfoto dieser Seite.

apfel

Random

Submit

Apfel - Curry - Süppchen

Granatapfelhühnchen

Apfel - Rosinen - Muffins

Recipe Description

Best Database Match

2	Apfel	230.0 g	Apfel
1	Knoblauchzehe(n)	5.0 g	Knoblauch
½ TL	Ingwer, geriebener	2.5 g	Ingwerknolle
300 g	Lauch	300.0 g	Porree roh
1 Schuss	Wein, weiß	5.0 g	Weißwein trocken
1 TL	Currypulver, (mild)	5.0 g	Currypulver
500 ml	Hühnerbrühe	4.0 g	Hühnerbrühe/-bouillon gekörnt
125 ml	Sahne	125.0 g	Sahne
100 ml	Kokosmilch	100.0 g	Kokosmilch
	Butter	0.0 g	Butter

	Total	Per portion	Per 100g	% GDA
Energy (kcal)	637.9	159.5	82.2	6.4
Protein (g)	11.9	3.0	1.5	5.0
Fat (g)	41.6	10.4	5.4	13.0
Carbohydrates (g)	52.8	13.2	6.8	3.9
Fibres (g)	11.8	3.0	1.5	11.8

Abbildung 1: Bildschirmfoto des Web-Interfaces

2 Installationshinweise

Der gesamte Quellcode des Systems befindet sich in der auf dieser CD enthaltenen Datei *findnutr.py*. Zur Interpretation wird die Programmiersprache *Python* benötigt. Zusätzlich ist das Python-Interface *MySQLdb* vonnöten, das die Verbindung zu MySQL-Datenbanken ermöglicht.

Ferner wird eine Datenbank mit den drei Tabellen *IngredientsTable*, *IngredientsRulesTable* und *QuantityRulesTable*, die ebenfalls auf der CD aufzufinden sind, benötigt. Zugangsdaten und verwendete Tabellennamen müssen in der Klasse *Datenbank* eingetragen werden.

3 Erläuterung der Klassen und ihrer wichtigsten Funktionen

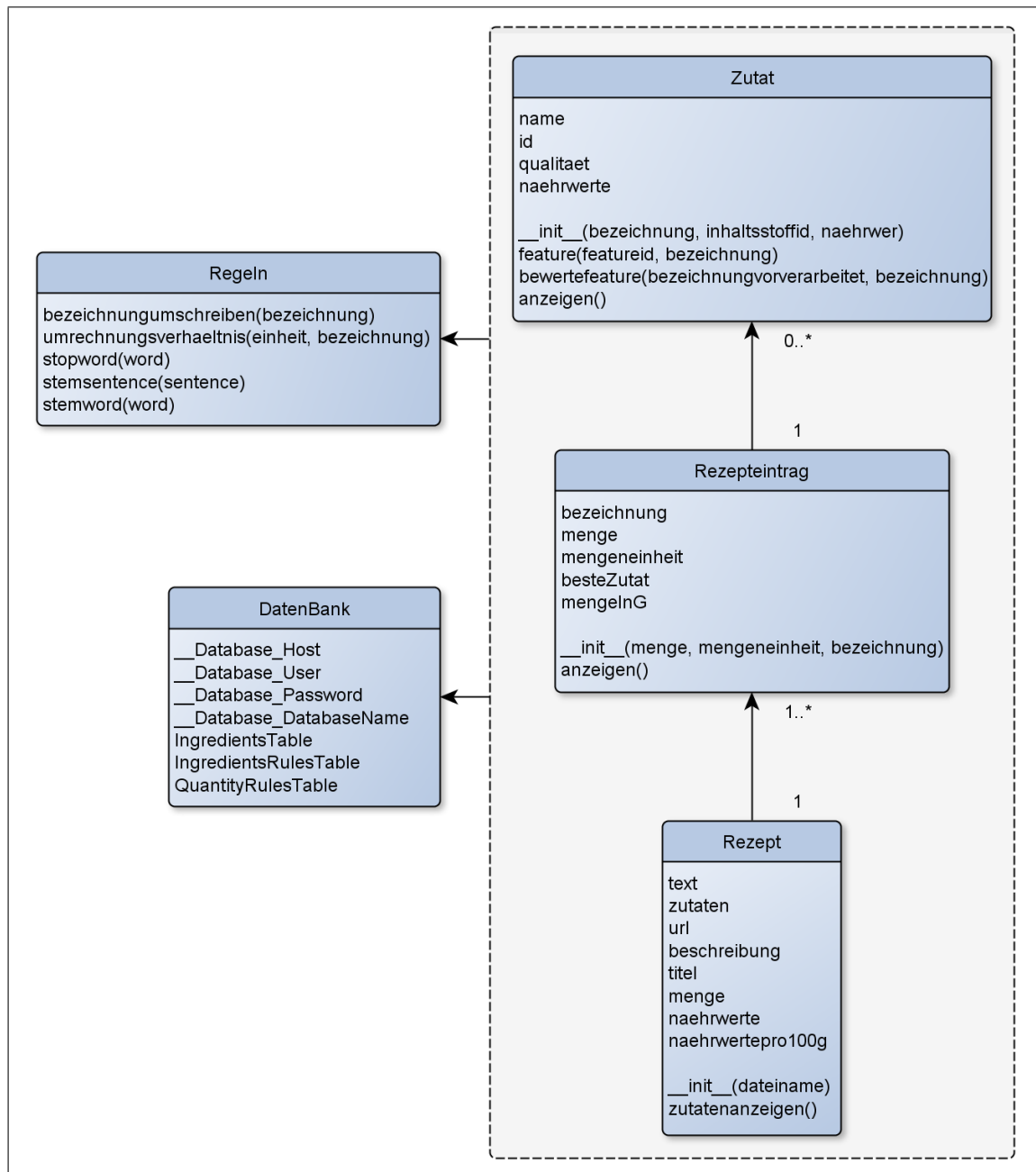


Abbildung 2: Klassendiagramm

3.1 Überblick

Das System wird durch die drei Klassen *Rezept*, *Rezepteintrag* und *Zutat* unterteilt. Ferner gibt es je eine globale Instanz der Klassen *DatenBank* und *Regeln*, die im Wesentlichen Schnittstellen zu den verschiedenen Datenbanken darstellen.

Die Klasse *Rezept* entspricht dabei einem Rezept der Internetseite *chefkoch.de*. Jedes Rezept besteht wiederum aus den in der Zutaten-Liste des Rezeptes eingetragenen Zutaten-Beschreibungen, welche jeweils Instanzen der Klasse *Rezepteintrag* sind. Jede Instanz der Klasse *Rezepteintrag* besteht wiederum aus verschiedenen Einträgen der Nährwerttabelle, die eine Auswahl aller für eine Zuordnung potentiell in Frage kommender Inhaltsstoffe darstellen. Jeder Eintrag in der Nährwerttabelle ist eine Instanz der Klasse *Zutat*. Für den Rezepteintrag *1 Prise Salz* kommen beispielsweise die Nährwertdatenbank-Einträge *Speisesalz*, *Steinsalz*, *Meersalz*, *jodiertes Salz* etc. in Frage.

3.2 Klasse: Rezept

3.2.1 Beschreibung

Eine Instanz der Klasse *Rezept* repräsentiert ein Rezept der verwendeten Rezept-Seite *chefkoch.de*. Im HTML-Format vorliegende Rezepte können geparkt werden. Gespeichert werden URL des Chefkoch-Rezeptes, Titel, Zubereitungs-Beschreibung und eine Liste aller Zutaten, jeweils bestehend aus einer Mengeneinheit (zum Beispiel *Gramm*; *Liter*), einem Mengenquantum (zum Beispiel *3*; *5*) und einer Bezeichnung (zum Beispiel *Zucker*; *Ei(er)*, *möglichst frisch*). Die Klasse liefert dann die Gesamtmasse aller im Rezept enthaltenen Zutaten, die enthaltenen Nährwerte aller Zutaten werden aufaddiert und sind ebenfalls abrufbar.

3.2.2 Wesentliche Variablen

- **text:** Der Quelltext des Rezeptes im HTML-Format.
- **url:** Die URL des Rezeptes.
- **beschreibung:** Die Zubereitungs-Beschreibung des Rezeptes.
- **titel:** Der Titel des Rezeptes.
- **zutaten:** Eine Liste aller im Rezept enthaltenen Zutaten. Jede Zutat ist dabei eine Instanz der Klasse *Rezepteintrag*.
- **menge:** Die Gesamtmasse des Rezeptes, also die Summe der Masse aller enthaltenen Zutaten.

- **naehrwerte:** Eine Liste aller im Rezept enthaltenen Nährwerte. Enthaltene Kilokalorien sind beispielsweise über den Wert *naehrwerte['GCAL']* abrufbar, enthaltene Proteine in Milligramm über den Wert *naehrwerte['ZE']*. Die Abkürzungen und Einheiten der verschiedenen Nährwerte können dem Handbuch des Bundeslebensmittelschlüssels (Seite 24-28) entnommen werden.
- **naehrwerte:** Eine Liste aller im Rezept enthaltenen Nährwerte pro 100g.

3.2.3 Wesentliche Funktionen

- **__init__(dateiname):** Zum Instanzieren eines Objektes wird dieser Funktion Pfad und Dateiname eines Rezeptes im HTML-Format übergeben. Alle Informationen werden dann automatisch ermittelt.
- **zutatenanzeigen():** Diese Funktion kann zur Ausgabe der ermittelten Informationen verwendet werden.

3.3 Klasse: Rezepteintrag

3.3.1 Beschreibung

Jede Instanz dieser Klasse verkörpert einen Eintrag der Zutaten-Liste, bestehend aus Bezeichnung (zum Beispiel *Salz*, *Mehl*), Mengeneinheit (zum Beispiel *Gramm*, *Liter*) und Mengenquantum (zum Beispiel *3*, *5*). Die Klasse speichert alle Einträge aus der Nährwertdatenbank, die für eine Zuordnung in Frage kommen, als Instanzen der Klasse *Zutat*. Zu jedem Eintrag der Nährwerttabelle wird die Eignung der Zuordnung bestimmt, um den am besten passenden Eintrag auswählen zu können. Danach wird versucht, die Masse des Rezepteintrags in Gramm zu bestimmen.

3.3.2 Wesentliche Variablen

- **bezeichnung:** Der Wortlaut, mit dem die Zutat in der Zutatenliste des Rezeptes umschrieben wird, zum Beispiel *Ei(er)*, *frisch*.
- **mengeneinheit:** Die Einheit der Menge aus der Zutatenliste, beispielsweise *Stück*, *Gramm* oder *Liter*.
- **menge:** Das Quantum der in der Zutatenliste umschriebenen Menge, bei der Zutat *3 Stück Ei(er)*, *frisch* ist dies zum Beispiel *3*.
- **besteZutat:** Die Zutat aus der Nährwerttabelle, die sich im Ranking als am besten geeignet herausgestellt hat, um die Beschreibung aus der Zutaten-Liste zu repräsentieren. Bei der Zutat *3 Stück Ei(er)*, *frisch* wäre dies beispielsweise *Hühnerei roh*.

- **mengeInG:** Die unter Berücksichtigung des Mengenquantums, der Mengeneinheit und der am besten passenden Zutat aus der Nährwerttabelle berechnete Masse des Rezepteintrags in der Einheit Gramm.

3.3.3 Wesentliche Funktionen

- **__init__(menge, mengeneinheit, bezeichnung):** Instanziiert wird ein Objekt der Klasse, indem Mengenquantum, Mengeneinheit und Zutaten-Beschreibung aus der Zutaten-Liste übergeben werden. Ein möglicher Aufruf wäre also: `Rezepteintrag('3', 'Stück', 'Ei(er), frisch')`.
- **anzeigen():** Zur Ausgabe der ermittelten Informationen.

3.4 Klasse: Zutat

3.4.1 Beschreibung

Instanzen dieser Klasse stellen einen Eintrag aus der Nährwerttabelle dar, welcher aus einer Beschreibung (zum Beispiel *Hühnerei roh*) und Nährwertangaben besteht. Für jede Instanz kann eine Eignung bestimmt werden, die angibt, wie gut der Eintrag aus der Nährwerttabelle zum Rezepteintrag passt. Objekte dieser Klasse werden ausschließlich von der Klasse *Rezepteintrag* benötigt, ansonsten ist es nicht sinnvoll, die Klasse *Zutat* zu instanziiieren oder deren Funktionen aufzurufen.

3.4.2 Wesentliche Variablen

- **name:** Die Beschreibung des Eintrags aus der Nährwerttabelle, beispielsweise *Hühnerei roh*.
- **id:** Die ID des jeweiligen Eintrags des Bundeslebensmittelschlüssels.
- **qualitaet:** Die berechnete Eignung des Eintrages der Nährwerttabelle zum Rezepteintrag.
- **naehrwerte:** Die Nährwerte des Eintrags, wie sie im Bundeslebensmittelschlüssel angegeben sind.

3.4.3 Wesentliche Funktionen

- **__init__(bezeichnung, inhaltsstoffid, naehrwer):** Instanziiert wird die Klasse unter Angabe der Beschreibung aus der Nährwertdatenbank (zum Beispiel *Hühnerei roh*), der ID des Eintrags im Bundeslebensmittelschlüssel und der Nährwerte, die im Bundeslebensmittelschlüssel angegeben sind.

- **feature(featureid, bezeichnung):** Für das Ranking berechnet diese Funktion eine Komponente des Merkmalsvektors zu einer bestimmten Beschreibung aus der Zutatenliste. Die Nummer der Komponente kann durch die *featureid* übergeben werden. Ein möglicher Aufruf ist also `feature(1, 'Eier frisch')`.
- **bewertefeature(bezeichnungvorverarbeitet, bezeichnung):** Diese Funktion wird von der Klasse *Rezepteintrag* aufgerufen, um die verschiedenen Komponenten der Merkmalsvektoren zu berechnen und zu gewichten. Übergeben wird dabei die Zutaten-Beschreibung aus der Zutaten-Liste des Rezeptes, sowohl verarbeitet als auch unangetastet.
- **anzeigen():** Zur Ausgabe der ermittelten Informationen.

3.5 Klasse: Datenbank

3.5.1 Beschreibung

Diese Klasse bildet ein Interface zu den verwendeten Datenbank-Tabellen. Ein Objekt der Klasse wird global instanziiert, so dass alle Klassen darauf zugreifen können.

3.5.2 Wesentliche Variablen

- **__Database_Host:** Die Adresse des Datenbank-Servers, zum Beispiel *localhost*.
- **__Database_User:** Der Benutzername für den Zugang zur Datenbank.
- **__Database_Passwort:** Das Passwort für den Zugang zur Datenbank.
- **__Database_DatabaseName:** Der Name der verwendeten Datenbank.
- **IngredientsTable:** Der Name der Nährwert-Tabelle.
- **IngredientsRulesTable:** Der Name Tabelle für die Bezeichnungs-Regeln.
- **QuantityRulesTable:** Der Name Tabelle für die Mengen-Regeln.

3.6 Klasse: Regeln

3.6.1 Beschreibung

Diese Klasse stellt verschiedene Hilfsfunktionen zur Verfügung. Ein Objekt der Klasse wird global instanziiert, so dass alle Klassen darauf zugreifen können.

3.6.2 Funktionen

- **bezeichnungumschreiben(bezeichnung):** Der Funktion wird eine Bezeichnung übergeben, die dann anhand der Bezeichnungs-Regeln umgeschrieben und zurückgegeben wird.
- **umrechnungsverhaeltnis(einheit, bezeichnung):** Anhand übergebener Mengeneinheit und einer Zutaten-Beschreibung aus der Nährwerttabelle liefert diese Funktion ein Umrechnungsverhältnis für die Masse der Zutat in Gramm. Beispielsweise ergibt der Aufruf *umrechnungsverhaeltnis('Stück', 'Hühnerei roh')* das Verhältnis *50*, woraus dann folgt, dass *3 Stück Hühnerei roh* eine Masse von 3×50 Gramm haben. Falls keine Menge ermittelt werden kann, wird 0 zurückgegeben, falls es sich um eine konstante Menge handelt, die nicht mehr mit dem Mengenquantum multipliziert werden sollte (z.B. bei *Prise Salz*), wird der zurückgegebene Wert mit -1 multipliziert, so dass dieser kleiner 0 ist.
- **stopword(word):** Die Funktion liefert *true*, falls ein übergebenes Wort in der verwendeten Stopwortliste enthalten ist, ansonsten wird *false* zurückgegeben.
- **stemword(word):** Diese Funktion verwendet den auf <http://snowball.tartarus.org/algorithms/german/stemmer.html> beschriebenen *Snowball-Stemmer*, um ein übergebenes Wort auf dessen Wortstamm zurückzuführen, der dann zurückgeliefert wird.
- **stemsentence(sentence):** Diese Funktion führt alle Wörter in einem übergebenen Satz auf deren Wortstamm zurück und liefert den umgeformten Satz zurück.

4 Anwendungsbeispiele

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

Zunächst wird das System importiert.

```
from findnutr import *
```

Nun kann eine Rezept-Datei geladen werden,

```
Beispielrezept = Rezept('raw/1.raw')
```

um bestimmte Nährwertangaben auszulesen

```
print "Kcal:_ " + str(Beispielrezept.naehrwerte['GCAL'])
print "Fett:_ " + str(Beispielrezept.naehrwerte['ZF'])
```

oder die Gesamtmasse der im Rezept enthaltenen Zutaten zu ermitteln.

```
print "Masse:_ " + str(Beispielrezept.menge) + " _Gramm"
```

Leicht kann ein Überblick über alle zugeordneten Zutaten und die im Rezept enthaltenen Nährwerte erzeugt werden.

```
Beispielrezept.zutatenanzeigen()
```

Falls nur eine einzelne Zutat zugeordnet werden soll, empfiehlt sich eine Instanz der Klasse *Rezepteintrag*. Zu beachten ist dabei, dass die Zutaten-Angaben in Unicode kodiert sein müssen.

```
Beispielzutat = Rezepteintrag(u'5', u'Stück', u'Ei(er)')
Beispielzutat.anzeigen()
```

Zum Schluss werden die Kalorien der Zutat *2 Liter Milch* ausgegeben.

```
Beispielzutat2 = Rezepteintrag(u'2', u'Liter', u'Milch')
if (Beispielzutat2.besteZutat != ''):
    kalorien = Beispielzutat2.besteZutat.naehrwerte['
        GCAL']*Beispielzutat2.mengeInG/100.0
    print "Kcal:_ " + str(kalorien)
else:
    print "passender _Nährwerteintrag _konnte _nicht _
        gefunden _werden"
```