Informationsextraktion

22.12.09

Computerlinguistik

Referenten: Alice Holka, Sandra Pyka

Informationsextraktion (IE)

1. Einleitung
2. Ziel der IE
3. Funktionalität eines IE-Systems
4. Beispiel
5. Übung
6. Aufbau
7. Architektur
8. Evaluation von IE-Systemen
9. Linguistische Tiefe
10. Schwierigkeiten von IE-Aufgaben
11. Literaturverzeichnis

1. Einleitung

- Durch Ausweitung des Internets sind immer mehr Texte online verfügbar

  → Informationsüberflutung

  - Es wird immer schwieriger, relevante Informationen zu finden, zu extrahieren und zu repräsentieren

- Informationsextraktionssysteme (IE-Systeme) werden entwickelt, um Informationsüberflutung adäquat meistern zu können

2. Ziel der IE

- Relevante Informationen aus freien, elektronischen Texten sollen gezielt aufgespürt und strukturiert werden
  - Analyse von Textpassagen, die relevante Informationen enthalten
  - ≠ keine umfassende Analyse des gesamten Inhaltes der Textdokumente
- Irrelevante Informationen werden gleichzeitig „Überlesen“

3. Funktionalität eines IE-Systems

- **Eingabe:**
  - Spezifikation des Typs der relevanten Informationen in Form von Templates (Menge von Merkmalen)
    - Durch domänspezifische Regeln wird dem System fest vorgegebenen, was als relevant gilt
    - Die Regeln müssen detailliert und präzise festlegen, welche Typen von Informationen von dem IE-System extrahiert werden sollen
  - Menge von freien Textdokumenten

3. Funktionalität eines IE-Systems

- **Ausgabe:**
  - Antwortmuster werden erzeugt
    - → Menge von instanziierten Templates
      - Strukturen in Form von Merkmal/Wert-Paaren (Tabelle)
  - Templates mit als relevant bestimmten Textabschnitten gefüllt

4. Beispiel

- Extraktion von Informationen über **Personalwechsel** aus Online- Dokumenten

Aufgabe:

Was soll extrahiert werden?

- wer hat verlassen **(PersonOut)**
- welche Position **(Position)**
- welcher Organization **(Organization)**
- wann wurde die Position verlassen **(TimeOut)**
- von wem neuen wurde die Position besetzt **(PersonIn)**
- wann wurde die Position besetzt **(TimeIn)**

4. Beispiel

- Template mit der Menge von Merkmalen:

PersonOut

PersonIn

Position

Organization

TimeOut

TimeIn

4. Beispiel

- Text:

  Dr. Hermann Wirth, bisheriger Leiter der Musikhochschule München, verabschiedete sich heute aus dem Amt. Der 65jährige tritt seinen wohlverdienten Ruhestand an. Als seine Nachfolgerin wurde Sabine Klinger benannt. Ebenfalls neu besetzt wurde die Stelle des Musikdirektors. Annelie Häfner folgt Christian Meindl nach.

4. Beispiel

- Gefülltes instanziiertes Template

PersonOut	Dr. Hermann Wirth
PersonIn	Sabine Klinger
Position	Leiter
Organisation	Musikhochschule München
TimeOut	Heute
TimeIn	Partielle Instanz, da Merkmal nicht mit Wert belegt wird

5. Übung 1

- Was geschieht mit dem 4. & 5. Satz des Textes?
  - → „Ebenfalls neu besetzt wurde die Stelle des Musikdirektors. Annelie Häfner folgt Christian Meindl nach.“
- **Aufgabe**: Erstelle eine weitere Templateeinstanz!

PersonOut	Christian Meindl
PersonIn	Annelie Häfner
Position	Musikdirektor
Organisation	Musikhochschule München
TimeOut	Partielle Instanz
TimeIn	Partielle Instanz

5. Übung 2

- Einzelne Merkmale können auch eine eigene Templatestruktur besitzen

Aufgabe:

Erstelle eine Templatestruktur für den Personennamen Dr. Hermann Wirth!

Nachname	Wirth
Vorname	Hermann
Titel	Doktor

6. Aufbau eines IE-Systems

- Zwei Ansätze :
  - Automatisch trainierte Systeme
  - „Knowledge Engineering Approach“

6.1 Automatisch trainierte Systeme

- 3 Methoden

1. Lernen aus Regeln eines annotierten Korpus
2. Lernen aus Regeln in Interaktion mit dem Benutzer
3. Verwendung statistischer Methoden

6.1 Automatisch trainierte Systeme

Lernen aus Regeln eines annotierten Korpus

- Trainingsmenge von bereits mit den Ergebnissen annotierten Textdokumenten
- **Ziel**:
  - automatisch Regeln zum Füllen von Vorlagen zu induzieren

Lernen aus Regeln in Interaktion mit dem Benutzer

- System macht eine Hypothese
- Benutzer bewertet die Hypothese (richtig oder falsch)
- System korrigiert ggf. seine Regeln

6.2. Knowledge Engineering Approach

- Entwicklung einer Grammatik von einem „K.E“
- Trainingsdaten, um das System zu testen
- Iteratives Verfahren

7. Architektur eines IE-Systems

1. **Tokenscanner**
   - Wortsegmentierung
2. **Morphologische und lexikalische Analyse**
   - Part of Speech Tagging
   - Word Sense Tagging
3. **Syntaktische Analyse**
   - Parsing
4. **Domänenanalyse**
   - Konferenz
   - Merging Partial

8. Evaluation von IE-Sytemen

- **Message Understanding Conference“ (MUC)**
  - Initiiert und finanziert von der DARPA
  - Evaluierungsveranstaltung, die jährlich stattfindet
  - IE-Systeme werden wettbewerbsmäßig
  - systematisch evaluiert

8. Evaluation von IE-Sytemen

- **Evaluationskriterien**
  - Maße Präzision (P)
  - Vollständigkeit (V)
  - F-Maß

9. Linguistische Tiefe

- völlig unterschiedliche linguistische Tiefe aufweisen
  - reiner Satzfilterung, wo lediglich semantische Orientierung in Form der Wortliste gegeben
  - bis hin zu Systemen mit Analysemodulen für sämtliche Ebenen der Sprache (Phonologie, Morphologie, Syntax, Semantik, ev. auch Pragmatik)

10. Schwierigkeiten von IE-Aufgaben

Verschiedene Schwierigkeiten können bei einer IE auftreten

- Die Allgemeinheit von Domänen
- Die Unstrukturriertheit von Texten
- Die Komplexität von zu extrahierenden Informationen
- Eigenschaften einer Sprache

  (deutsche Sprache unterscheidet sich von englischer Sprache in der Groß- und Kleinschreibung)

  → im englischen: meist nur Eigennamen und Satzanfänge groß geschrieben und dadurch wird Eigennamenerkennung erleichtert

11. LITERATURVERZEICHNIS

Neumann, Günter (2001) ”Informationsextraktion” in Carstensen, Kai-Uwe et al. *Computerlinguistik und Sprachtechnologie. Eine Einführung, Heidelberg, Berlin: Spektrum. 448-455.*

<http://duepublico.uni-duisburg-essen.de/servlets/.../informationsextraktion.pdf/>.

<http://*quui.de/fsteeg/files/spinfo-ie-ha.pdf/*>.