Ralph Grishman
„Information Extraction: Techniques and Challenges“

Referat von Felix Jungermann

12.11.2002

1. Einleitung

1.1 Über den Artikel

- Autor Ralph Grishman
- Professor an der Universität von New York
- Mitbegründer des Proteus Projekts
- Verfasst im Jahr 1997

1.2 Was versteht man unter I.E.?

- Gezielt Informationen aus grossen Textbeständen
- Identifikationen von Ereignissen und Beziehungen
- Strukturierte Repräsentation (ähnlich Datenbank)
- Grosses Interesse durch MUC
- MUC-3: Terrorismus
- Wer, was, wann, wo, mit welchen Folgen?

19 March – A bomb went off this morning near a power tower in San Salvador leaving a large part of the city without energy, but no casualties have been reported. According to unofficial sources, the bomb – allegedly detonated by urban guerrilla commandos – blew up a power tower in the northwestern part of San Salvador at 0650 (1250 GMT).

INCIDENT TYPE	bombing
DATE	March 19
LOCATION	El Salvador: San Salvador (city)
PERPETRATOR	urban guerrilla commandos
PHYSICAL TARGET	power tower
HUMAN TARGET	-
EFFECT ON PHYSICAL TARGET	destroyed
EFFECT ON HUMAN TARGET	no injury or death
INSTRUMENT	bomb

- Andere MUC: JointVentures oder Positionswechsel

- Keineswegs volles Textverständnis
- Volles Textverständnis = alle Informationen
- I.E. : Bestimmung von Semantik der Ausgabe

1.3 Wieso besteht Interesse an I.E.?

- Viele Informationen NUR in natürlichsprachlichen Texten
- Aktuelle Möglichkeiten: Textarchiv, Internet
- Aktuelle Möglichkeiten stossen auf Grenzen!
- Grosse Vorteile bei Verarbeitung techn. Texte
- Beispiel Krankenblatt

- Effizienz immer noch schlecht!
- Systeme mit schlechter Performanz trotzdem von Vorteil
- Informationen müssen „gut“ vorliegen, damit aktuelle Systeme gut arbeiten

2. I.E. am Beispiel MUC

- Erhalt des „training corpus“
- Systeme werden bearbeitet
- Abgabe des „test corpus“
- Vergleich zwischen „answer key“ und „test corpus“
- *precision* und *recall* (F-Note = (2*p*r)/(p+r))

3. Grundlegende Techniken der I.E.

3.1 Einführung

- Prozess besteht aus zwei grundlegenden Teilen
- Lokale Textanalyse
- Analyse der erarbeiteten Bestandteile

- Fakten ins Ausgabeformat konvertieren

- Fakten werden mithilfe von Mustern extrahiert
- Muster dürfen keinen konkreten Wortstücken oder -abfolgen entsprechen
- Daher: Strukturierung der Eingabe!
- Lexikalische Analyse
- Namenserkennung
- Syntaxanalyse

3.2 Mustererkennung und Strukturaufbau

- Beispiel:

Sam Schwartz retired as executive vice president of the famous hot dog manufacturer, Hupplewhite Inc.
He will be succeeded by Harry Himmelfarb.

- Um endgültiges template zu generieren, benötigt man semantische Strukturen
- „entity“
- „event“
- Diese werden aus der Syntax extrahiert

3.3 Lexikalische Analyse

- Zerlegung des Textes in Sätze
- Worte der Sätze werde im Lexikon
  „nachgeschlagen“
- Proteus benutzt „Complex Syntax dictionary“
  und andere Lexika

3.4 Namenserkennung

- Erkennung echter Namen sowie anderer spezieller Formen, wie z.B. Datumsangaben
- Verschiedene Merkmale für Namenserkennung
- Personennamen <> Firmennamen
- Firmenlexikon ist hilfreich!

- Das Beispiel momentan:

[name type: person Sam Schwartz] retired as executive vice president of the famous hot dog manufacturer,
[name type: company Hupplewhite Inc.]
He will be succeeded by [name type: person Harry Himmelfarb].

- Erkennen von Aliasnamen

- Aliasnamen sind hilfreich als Referenz für Namen

3.5 Syntaktische Struktur

- Komplette Syntaxidentifikation ist problematisch
- Einige Systeme bilden komplette Syntaxstrukturen
- Proteus (und andere) gehen Kompromisse ein:
- Substantive und linke nähere Bestimmung
- Prädikatgruppen mit Hilfsverben

- Das Beispiel unterteilt in Substantiv- (ng) und Prädikatgruppen (vg) :

[ng entity: e1 Sam Schwartz] [vg retired] as [ng entity e2 executive vice president] of [ng entity: e3 the famous hot dog manufacturer], [ng entity: e4 Hupplewhite Inc.] [ng entity: e5 He] [vg will be succeeded] by [ng entity: e6 Harry Himmelfarb].

- Informationen der Gruppen werden noch untersucht

- Für jede Substantiv-Gruppe wird eine sogenannte semantische entity erstellt

entity e1	type: person	name: „Sam Schwartz“
entity e2	type: position	value: „executive vice president“
entity e3	type: manufacturer	
entity e4	type: company	name: „Hupplewhite Inc.“
entity e5	type: person	
entity e6	type: person	name: “Harry Himmelfarb”

- Grössere Substantiv-Gruppen werden gebildet
- Verbindung von zwei Gruppen
- Entity enthält dann hinzugefügte Informationen
- Aufstellen der isa-Hierarchie

- Es ergibt sich folgende Markierung für das Beispiel:

[ng entity: e1 Sam Schwartz] [vg retired] as [ng entity e2 executive vice president of the famous hot dog manufacturer Hupplewhite Inc.] [ng entity: e5 He] [vg will be succeeded] by [ng entity: e6 Harry Himmelfarb].

- Nun ergeben sich die entities wie folgt:

entity e1	type: person	name: „Sam Schwartz“
entity e2	type: position	value: „executive vice president“
		company: e3
entity e3	type: manufacturer	name: „Hupplewhite Inc.“
entity e5	type: person	
entity e6	type: person	name: “Harry Himmelfarb”

3.6 Szenario-Mustererkennung

- Bis jetzt Vorbereitung für Szenario-Mustererkenn.
- Dem zu untersuchenden Positionswechsel liegen zwei Muster zugrunde:
- person retires as position
- person is succeeded by person

- Ereignis-Klauseln (events) werden aufgestellt

[clause event: e7 Sam Schwartz retired as executive vice president of the famous hot dog manufacturer Hupplewhite Inc.] [clause event: e8 He will be succeeded by Harry Himmelfarb.]

- Nun werden die events zusätzlich verzeichnet:

entity e1	type: person	name: „Sam Schwartz“	
entity e2	type: position	value: „executive vice president“	
		company: e3	
entity e3	type: manufacturer	name: „Hupplewhite Inc.“	
entity e5	type: person		
entity e6	type: person	name: “Harry Himmelfarb”	
event e7	type: leave-job	person: e1	position: e2
event e8	type: succeed	person: e6	person2: e5

- Pronomen werden geprüft
- Verbindungen des Pronomens werden auf eine eine kurz zuvor benutzte entity des Typs person übertragen

- Also folgt:

entity e1	type: person	name: „Sam Schwartz“
entity e2	type: position	value: „executive vice president“
		company: e3
entity e3	type: manufacturer	name: „Hupplewhite Inc.“
entity e6	type: person	name: “Harry Himmelfarb”
event e7	type: leave-job	person: e1 position: e2
event e8	type: succeed	person: e6 person2: e1

- Weiteres Nutzen der isa-Hierarchie
- Über mehrere Sätze verstreute Informationen müssen kombiniert werden
- Schlussfolgerungen über Informationen
- Was impliziert zum Beispiel „succeed“?

-Beispiele: Sam was president. He was succeeded by Harry.
Sam will be president; he succeeds Harry.

- leave-job(X-person, Y-job) & succeed(Z-person, X-person)
  -> start-job(Z-person, Y-job)

- start-job(X-person, Y-job) & succeed(X-person, Z-person)
  -> leave-job(Z-person, Y-job)

...

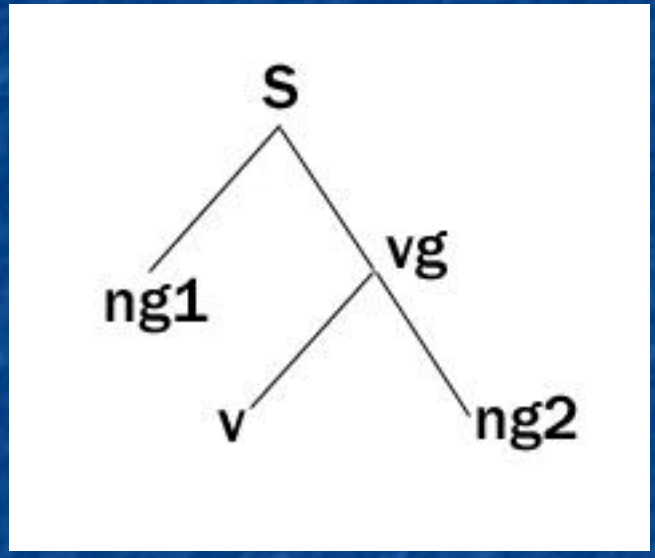
event e7	type: leave-job	person: e1	position: e2
event e8	type: succeed	person: e6	person2: e1
event e9	type: start-job	person: e6	position: e2

4. Probleme im Aufbau

4.1 Partielle oder vollständige Analyse

- Frühere Systeme führten komplette syntaktische Analysen durch
- Man benötigt jedoch nur Struktur in Hinsicht aufs Szenario!

- Proteus arbeitet mit Metaregeln
- subject=company verb=hired object=person



Beispiel-Syntaxbaum

- Folgende Strukturen sind denkbar:

  v ng1 ng2?
  ng1 v ng2
  ng2 v ng1
  etc.

- Diese werden dann von dem System erstellt:

  hired company person?
  company hired person
  person was hired by company
  person, who was hired by company
  person, hired by company
  etc.

- Aktuelle Systeme arbeiten mit Werten um 80%
  (mit handgeklammerten Texten trainiert!!!)

4.2 Portabilität

- Umstellen der Systeme ist problematisch!
- Umstellen muss leichter und automatisiert werden
- AutoSlog für MUC-4
- Systeme mithilfe ML wurden entwickelt
- Viele Beispiele <> wenige bearbeitete Beispiele
- Proteus arbeitet mit interaktivem Tool

4.3 Performanz-Probleme

- MUC-6: beste Systeme erreichten F von nur 51-56
- Ähnliches Design
- Mittlerer Level schnell zu erreichen
- Steigerungen „sehr teuer“
- Unwissen über aktuelles Szenario
- Je mehr Extraktionen, umso besser