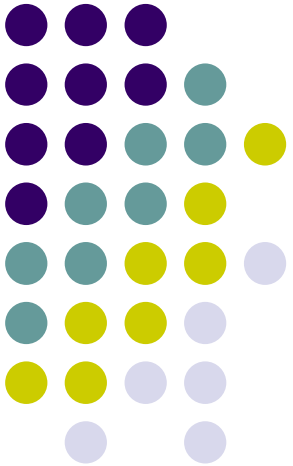
Unsupervised Discovery of Scenario-Level Patterns for Information Extraction

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, Silja Huttunen (2000)



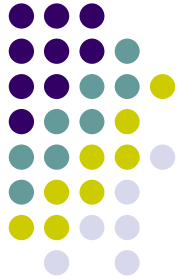
Universität Dortmund

Seminar „Informationsextraktion“

Bianca Selzam

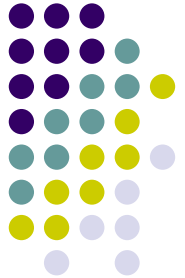
19. November 2002

Übersicht



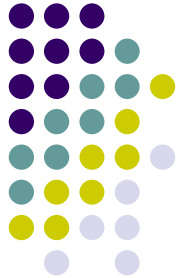
1. Grundlagen der Informationsextraktion
2. Pattern Matching
3. Vorstellung des IE-Systems von Yangarber, Grishman, Tapanainen, Huttunen
4. Algorithmus zur Pattern-Generierung
5. Auswertung und Ergebnisse der MUC-6
6. Stellungnahme
7. Literaturverzeichnis

1. Aufgabe der Informationsextraktion



- Selektive Extraktion der Semantik aus natürlichsprachlichen Texten
- Unterteilung in semantische Objekte:
  - Einheiten
  - Beziehungen
  - Ereignisse
- Speicherung der extrahierten Informationen in relationaler Datenbank

1. Begriffe aus der IE-Literatur



- Subject domain

  Klasse von Textdokumenten, die verarbeitet werden sollen

- Scenario

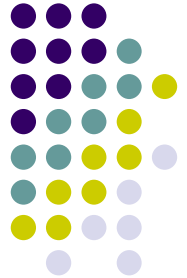
  Festgelegtes Thema, das innerhalb einer Domain von Interesse ist

  Beispiel: *Management succession (MUC-6)*

- MUC

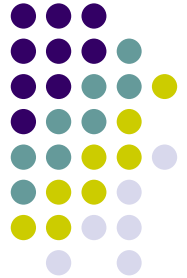
  Message Understanding Conference

2. Pattern Matching



- Pattern = Regulärer Ausdruck
  - Universale Komponente
  - Domain- und Szenario-spezifische Komponente
- Speicherung in Pattern Base
- Probleme:
  - Übertragbarkeit
  - Leistung

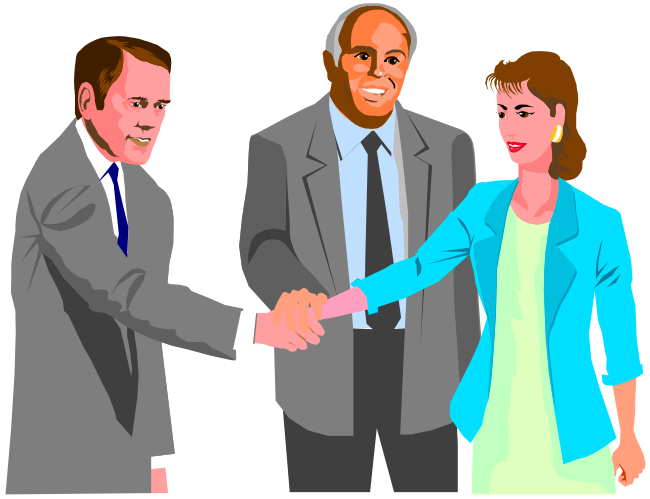
2. Arbeitsweise herkömmlicher Pattern-Matching-Systeme



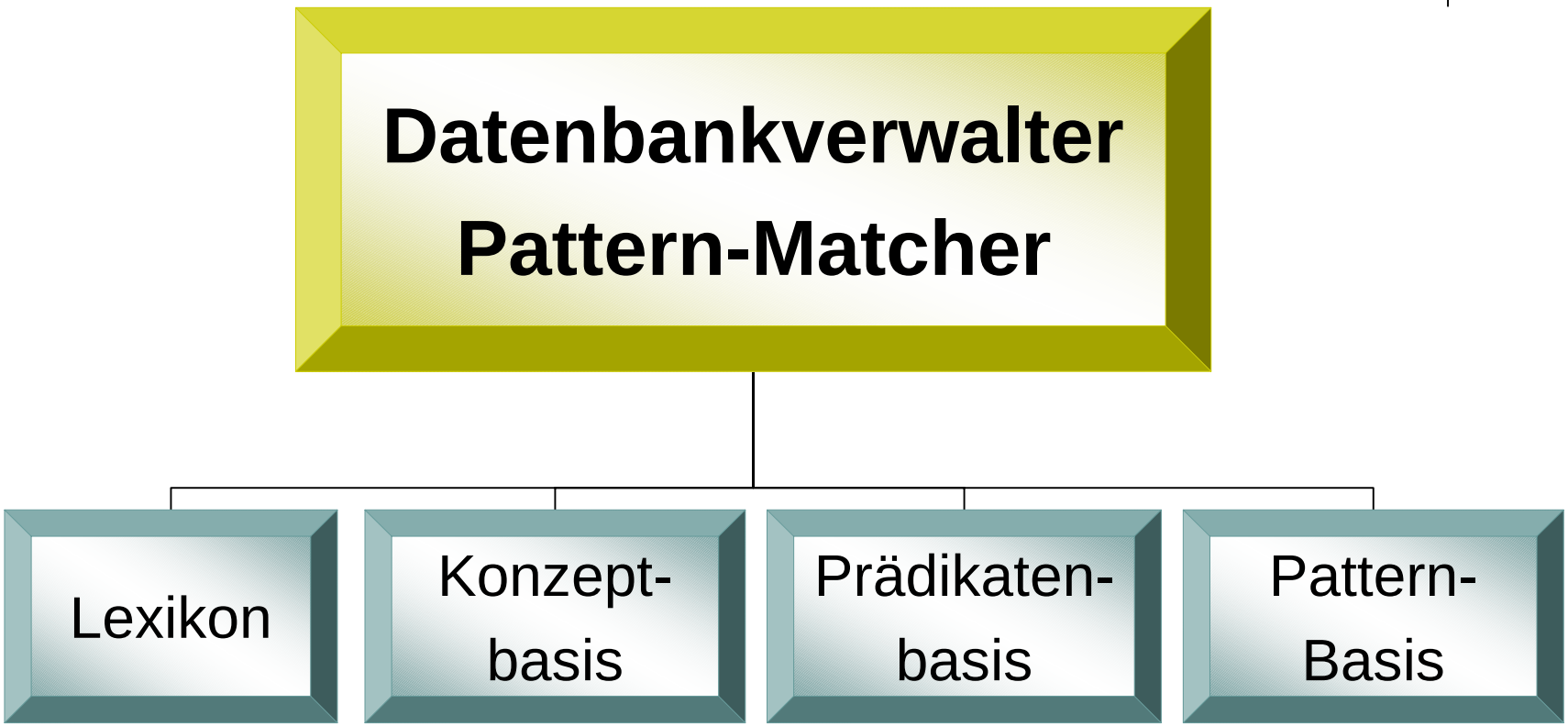
- Auswahl geeigneter Beispielsätze aus dem Text durch den Benutzer
- Generalisierung in Patterns durch das IE-System
- Probleme:
  - Verantwortung des Benutzers, zu jeder syntaktischen bzw. semantischen Konstruktion Beispiele zu finden
  - Sehr großer Zeitaufwand!

3. Message Understanding Conference

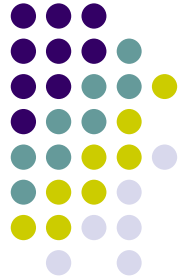
- MUC-6:
  - 15. - 17. November 1995
  - Columbia, Maryland, USA
  - Scenario: “Management Succession”



3. Aufbau des neu entwickelten IE-Systems



3. Lösungsansatz: Automatische Pattern-Generierung



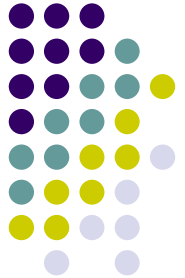
- Idee:

  1. Zum Szenario relevante Dokumente enthalten gute Patterns.

  2. Gute Patterns sind in zum Szenario relevanten Dokumenten zu finden.

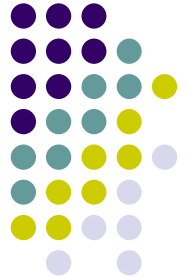
- Festlegung von wenigen Seed-Patterns
- Automatische Generierung neuer Patterns durch die initialen Seed-Patterns

4. Algorithmus: Vorgehensweise



- Gegeben:
  - Großer Korpus unkommentierter und unklassifizierter Dokumente
  - Menge von initialen Seed-Patterns
  - (möglicherweise leere) Menge von Konzeptklassen
- Partitionierung des Korpus durch die Seed-Patterns:
  - Menge R: mindestens ein Pattern gefunden
  - Menge ¬R: kein Pattern gefunden
- Generierung neuer Patterns

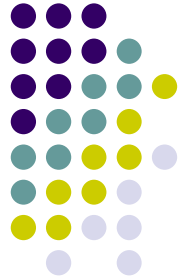
4. Algorithmus: Preprocessing



- Anwendung eines Namenserkennungsmoduls
- Ersetzung jedes Namens durch seinen Klassenbegriff

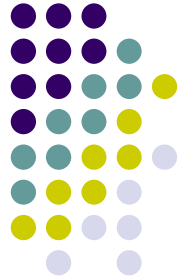
  Beispiele: *C-Person, C-Company, …*
- Ersetzung aller numerischen Werte durch einen Klassenbegriff

4. Algorithmus: Syntaktische Analyse



- Anwendung eines Syntaxanalyse-Tools
- Transformierung jedes Satzes in syntaktische Normalform, d. h. Prädikat-Argument-Struktur
- Repräsentation eines Satzes als Tupel:
  - Subjekt, ***z. B.*** *„**John** sleeps“*
  - Verb, ***z. B.*** *„John **sleeps**“*
  - Objekt, ***z. B.*** *„**John** is appointed by Company“*
  - Phrase bezogen auf Subjekt oder Objekt, ***z. B.*** “*Company named John Smith **president**“*

4. Algorithmus: Generalisierung

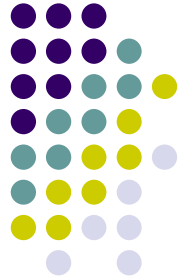


- Reduzierung der Tupel zu Paaren

  Beispiele: *Verb – Objekt, Subjekt – Verb, …*
- Suche nach szenario-relevanten Paaren
- Erstellen oder Erweiterung von Konzeptklassen

  Beispiel: *company {hire / fire / expel} person*
- Neue Partitionierung der Dokumentenmenge durch die neue Patternsammlung
- Wiederholung, bis keine neuen Patterns mehr gefunden werden

4. Algorithmus: Suche nach neuen Patterns



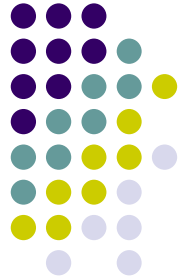
- MUC-6: Szenario „Management Succession“
- Vorgegebene Seed Patterns:

Subjekt	**Verb**	**Direktes Objekt**
C-Company	C-Appoint	C-Person
C-Person	C-Resign	---

C-Appoint = {appoint, elect, promote, name}

C-Resign = {resign, depart, quit, step-down}

4. Algorithmus: Suche nach neuen Patterns



- Berechnung des Scores nach jedem Iterationsschritt:

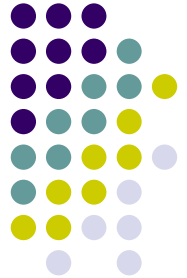
$$L(p) = P_c(p) \cdot \log|H \cap R|$$

H = {Dokumente, in denen p gefunden wird}

R = {relevante Dokumente}

$$P_c(p) = \frac{|H \cap R|}{|H|}$$ (bedingte Wahrscheinlichkeit)

4. Algorithmus: Suche nach neuen Patterns



- Auswahlkriterien:
  - Verwerfen zu häufiger Patterns, für die gilt:

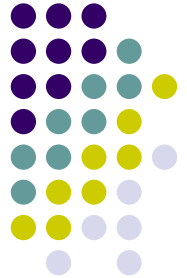
    $$|H \cap R| > \frac{|U|}{10}$$

  - Verwerfen zu seltener Patterns, für die gilt:

    $$|H \cap R| < 2$$

- Auswahl des Patterns mit dem höchsten Score
- Hinzufügen zu den Seed Patterns
- Iteration des Verfahrens

4. Algorithmus: Bewertung der Dokumente



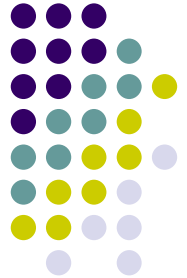
- Seed Patterns: Relevanz 1
- Zuweisung eines Precision-Maßes nach i Iterationen:

$$\mathrm{Prec}^{i+1}(p) = \frac{1}{|H(p)|} \cdot \sum_{d \in H(p)} \mathrm{Rel}^{i}(d)$$

- Precision-Maß für Klassen von Patterns:

$$\mathrm{Prec}^{i+1}(K) = \frac{1}{|H(K)|} \cdot \sum_{d \in H(K)} \mathrm{Rel}^{i}(d)$$

4. Algorithmus: Bewertung der Dokumente

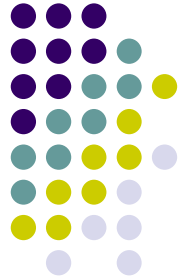


- Anpassen der Relevanz-Scores nach Hinzunahme des neuen Patterns:

$$\mathrm{Rel}^{i+1}(d) = \max\left(\mathrm{Rel}^{i}(d), \mathrm{Prec}^{i+1}(K_d)\right)$$

- Motivation:
  - Monotones Wachstum der Relevanz-Scores

5. Wichtige Bewertungsmaße



- Precision:

$$\text{Pre} = \frac{|H \cap R|}{|H|}$$

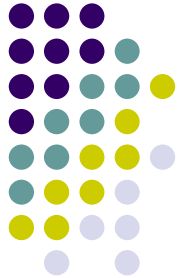
- Recall:

$$\text{Rec} = \frac{|H \cap R|}{|R|}$$

- F-Maß:

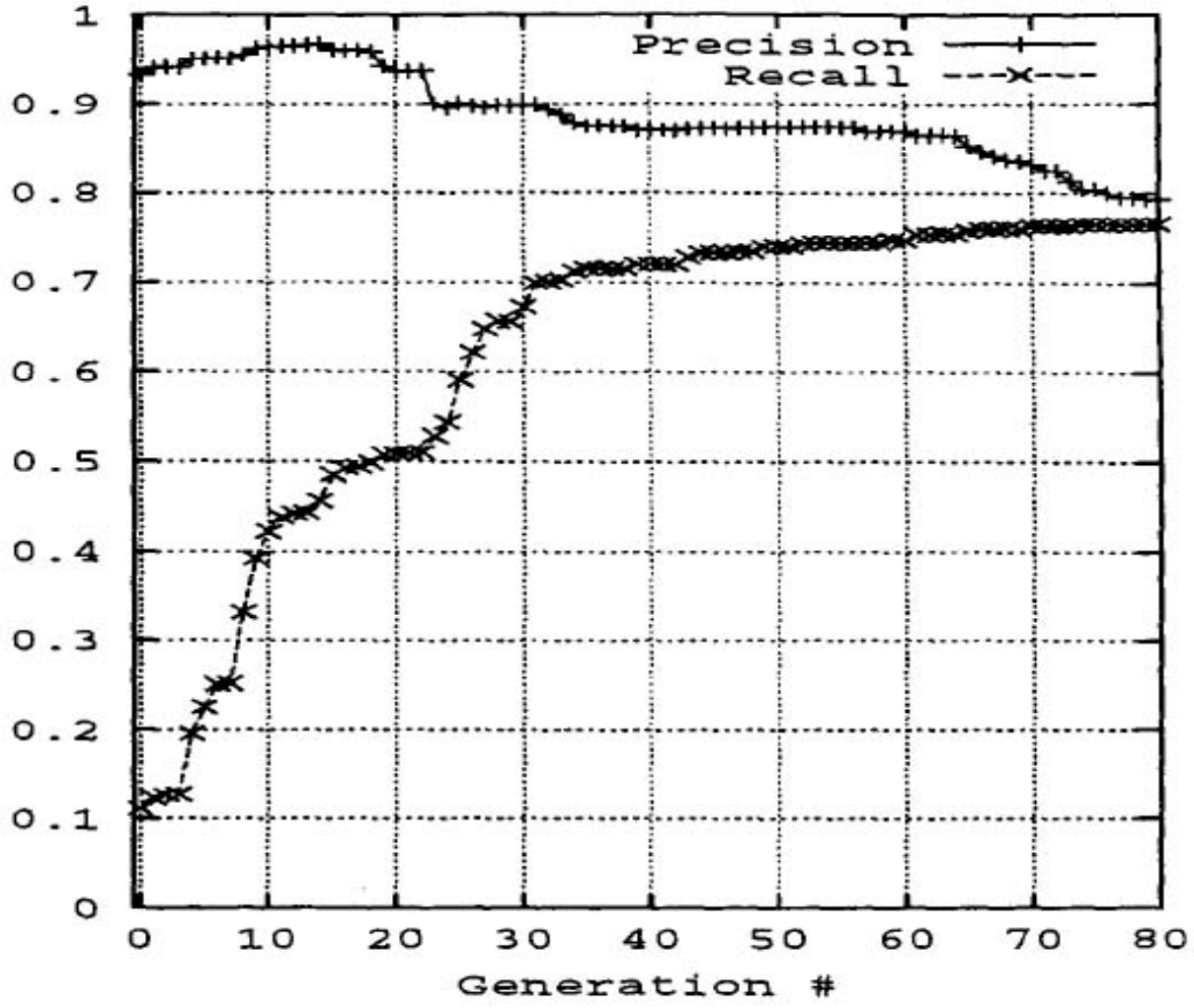
$$\text{F}_\beta = \frac{(\beta^2 + 1) \cdot pre \cdot rec}{\beta^2 \cdot pre + rec}$$

5. Auswertung

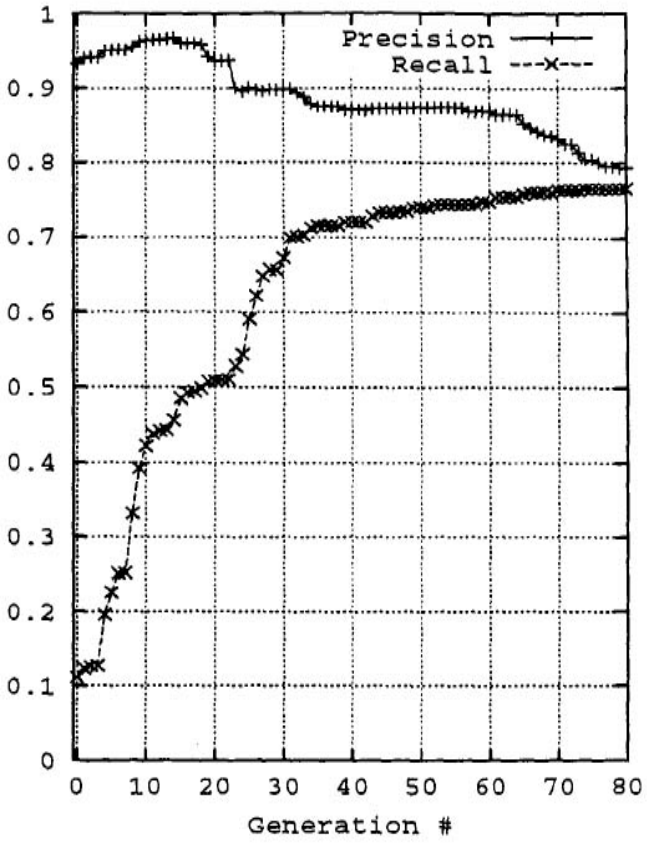


- Haupt-Entwicklungs-Korpus der MUC-6: 5963 Dokumente
- Bestimmung eines Test-Korpus von 100 Trainings-Dokumenten
- Zufällige Auswahl von 150 weiteren Dokumenten aus dem Haupt-Korpus
- Benutzte Seed-Patterns:
  - <C-Company> <C-Appoint> <C-Person>
  - <C-Person> <C-Resign>

5. Precision-/Recall-Kurven

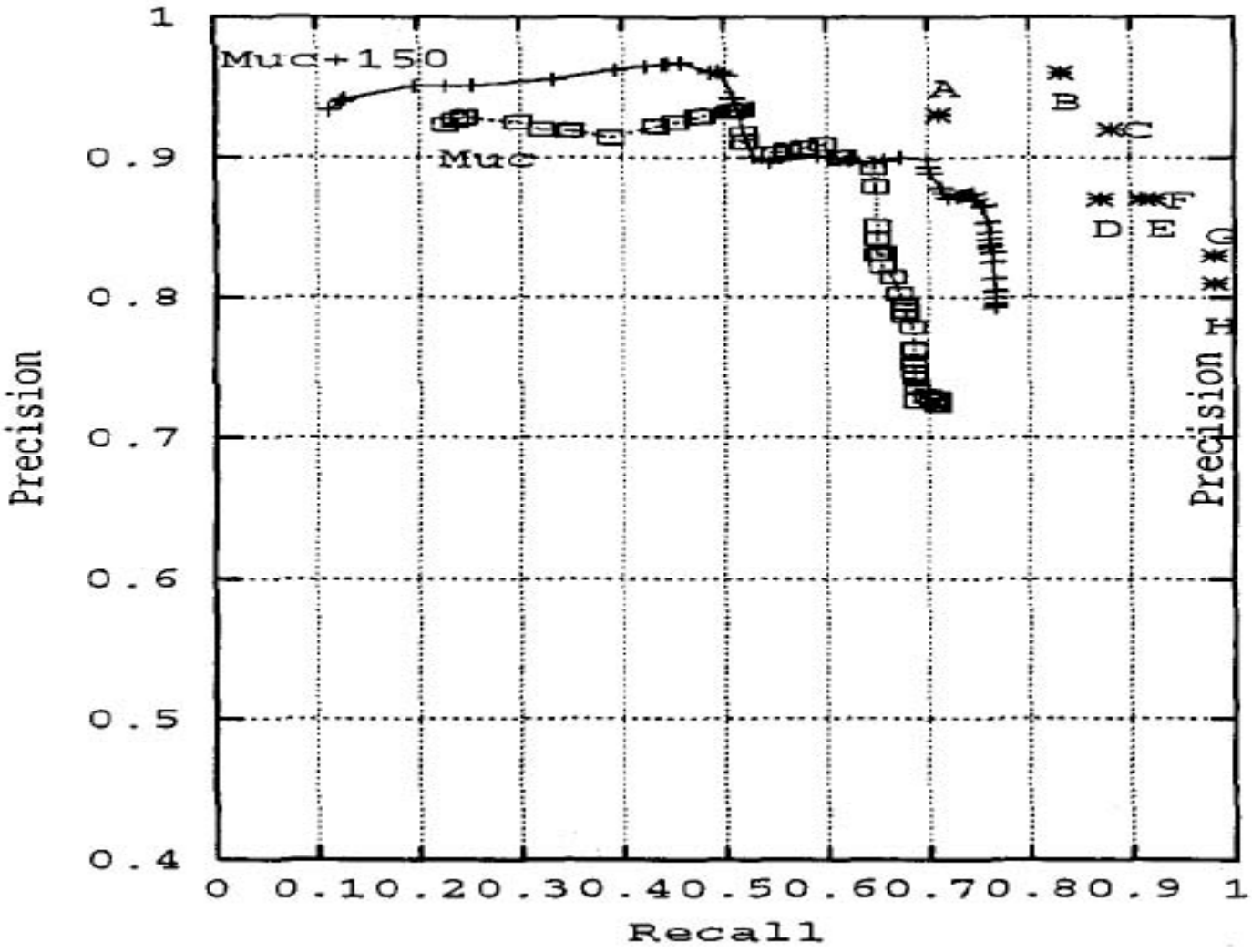


5. Precision-/Recall-Kurven

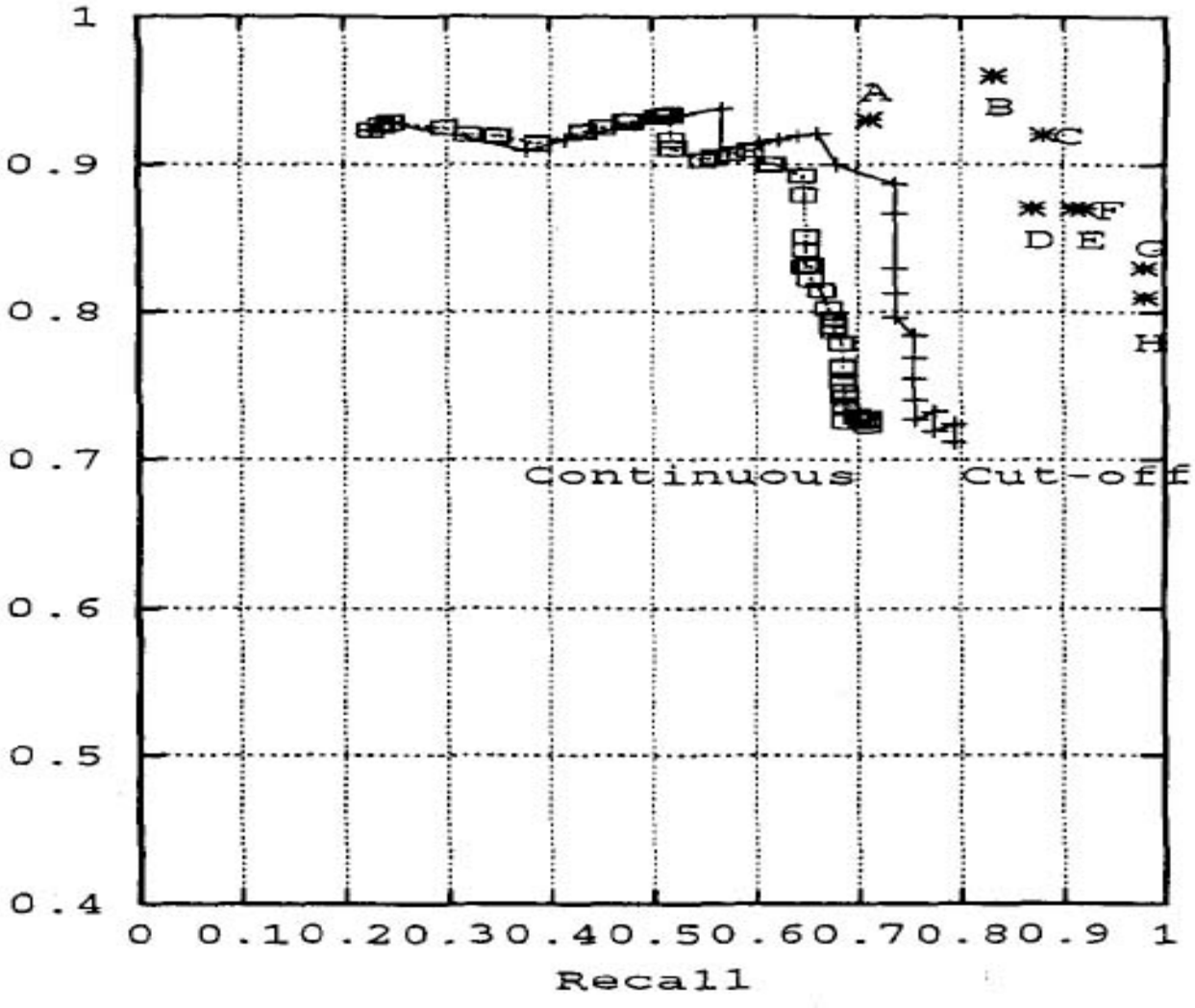


- Nach der ersten Iteration (Seed Patterns):
  - 184 von 5963 Dokumenten gefunden
  - Precision: 93%
  - Recall: 11%
- Nach 80 Iterationen:
  - 982 relevante Dokumente gefunden
  - Precision: 80%
  - Recall: 78%

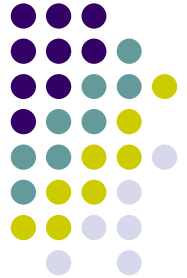
5. Precision-/Recall-Kurven



5. Precision-/Recall-Kurven



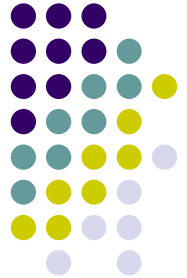
5. Auswertung der Patterns



- Einfaches Performance-Maß:

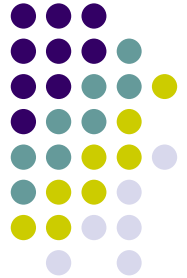
  Vergleich Anzahl gefundener Patterns vs. Systeme der anderen MUC-Teilnehmer
- 75 verschiedene Patterns
- 30 verschiedene Verben
- Exklusive Entdeckung von 8 Verben, z. B.:
  - *Company – bring – person – as – officer*
  - *Person – come – to – company – as – officer*
  - *Person – rejoin – company – as – officer*

5. Vergleich verschiedener Patterns



Pattern-Basis	Recall	Precision	F-Maß
Seed Patterns	28%	78%	41,32
Seed Patterns + generierte Patterns	51%	76%	61,18
von Hand - MUC	54%	71%	61,93
von Hand – nun im System benutzt	69%	79%	73,91

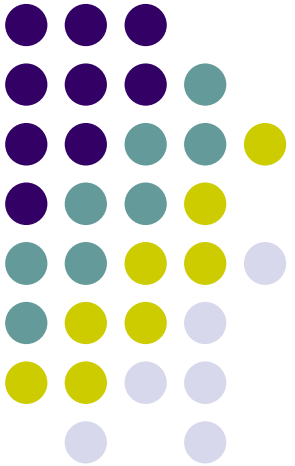
6. Stellungnahme



- Pattern Matching verbreitetes Mittel zur Informationsextraktion
- **Neu:** Lernverfahren zur automatischen Pattern-Generierung
- Unmarkierte Texte
- Nachteil der mangelnden Universalität
- Interessanter Ansatz für zukünftige IE-Systeme

Vielen Dank …

… für Eure Aufmerksamkeit!



7. Literaturverzeichnis

- R. Yangarber, R. Grishman, P. Tapanainen, S. Huttunen: „*Unsupervised discovery of scenario-level patterns for information extraction*“. Proceedings of the Sixth Conference on Applied Natural Language Processing, (ANLP-NAACL 2000), 2000.
- R. Yangarber, R. Grishman: „*Machine Learning of Extraction Patterns from Unannotated Corpora: Position Statement*.“