Information Extraction and Question-Answering Systems

Foundations and methods

Dr. Günter Neumann
LT-Lab, DFKI
neumann@dfki.de

What the lecture will cover

Machine Learning for IE

Lexical processing

Evaluation Methods

Basic Terms & Examples

Parsing of Unrestricted Text

Generic NL Core system

Domain Modelling

Question/Answering Core components

Advanced Topics

2

Extraktion und Induktion von Ontologien und Lexikalisch Semantischen Relationen

Extraktion und Induktion von Ontologien und Lexikalisch Semantischen Relationen

- **Learning to Extract Symbolic Knowledge from the WWW**
- **Discovering Conceptual Relations from Text**
- **Extracting Semantic Relationships between Terms**

4

Learning to Extract Symbolic Knowledge from the WWW

CMU World Wide Knowledge Base (Web->KB) project

The approach explored in this research is to develop a trainable system that can be taught to extract various types of information by automatically browsing the Web.
This system accepts two types of inputs:

1. An ontology specifying the classes and relations of interest.
2. Training examples that represent instances of the ontology classes and relations.

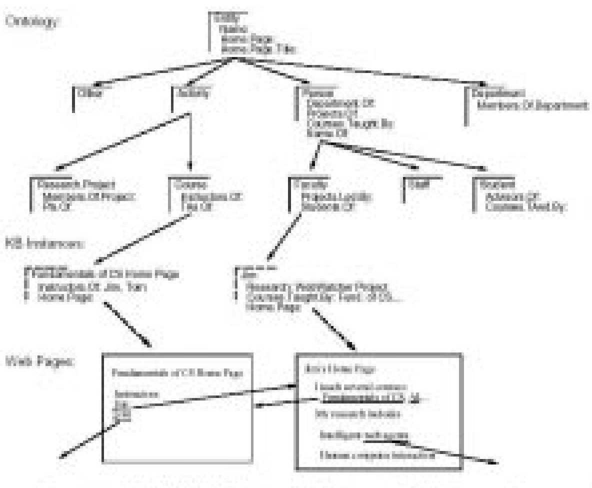

Figure 1: The inputs and outputs of the WebKB system. The top part of the figure shows an ontology that defines the classes and relations of interest. The bottom part shows two Web pages identified as training examples of the classes Course and Faculty. Together, these two pages also constitute a training example for the relations Instructors.Of and Courses.Taught.By. Given the ontology and a set of training data, WebKB learns to interpret additional Web pages and hyperlinks to add new instances to the knowledge base, such as those shown in the middle of the figure.

Assumptions about the mapping between the ontology and the Web:

1. Each Instance of an ontology class is represented by one or more *contiguous segments of hypertext* on the Web
    1. Web page
    2. Contiguous string of text within a web page
    3. Collection of web pages interconntected by hyperlinks
2. Each instance **R(A,B)** of a relation **R** is represented on the Web in one of three ways:
    1. the instance **R(A,B)** may be represented by a segment of hypertext that connects the segment representing **A** to the segment representing **B.**
    2. the instance **R(A,B)** may alternatively be represented by a contiguous segment of text representing **A** that contains the segment that represents **B**.
    3. the instance **R(A,B)** may be represented by the fact that the hypertext segment for **A** satisfies some learned model for relatedness to **B**

7

Experimental Testbed

- **Domain:** computer science departments
- **Ontology includes the following classes:** Department, Faculty, Staff, Student, Research.Project, Course, Other.
- Each of the classes has a set of slots defining relations that exist among instances of the given class and other class instances in the ontology.
- **Two data sets:**
    - 4.127 pages and 10.945 hyperlinks drawn from four CS departments
    - 4.120 additional pages from numerous other CS departments

8

Recognizing Class Instances

The first task for the system is to identify new instances of ontology classes from the text sources on the Web.

There are different approaches:

- Statistical Text Classification
- First-Order Text Classification
- Identifying Multi-Page Segments

Statistical Text Classification

A document *d* belongs to class *c'* according to the following rule (compute score for each class c and choose maximum)

$$c' = \underset{c}{\operatorname{argmax}} \left(\frac{\log \Pr(c)}{n} + \sum \Pr(w_i|d) \log \left[\frac{\Pr(w_i|c)}{\Pr(w_i|d)} \right] \right)$$

- n = the number of words in *d*
- T = the size of the vocabulary
- w_i = the i-th word in the vocabulary
- $\Pr(w_i|c)$ = probability of drawing w_i given a document from class *c*
- $\Pr(w_i|d)$ = the frequency of occurrence of w_i in document *d*

- vocabulary limited to 2000 words in this experiment

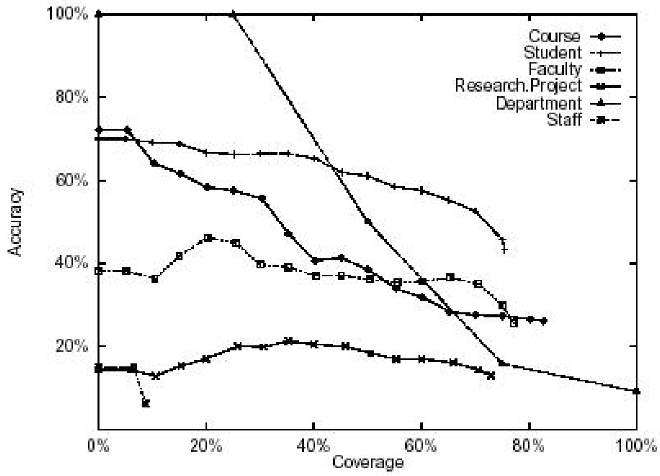

Figure 2: Accuracy/coverage for statistical classifiers.

First Order Text Classification

- Used Algorithm ist FOIL (Quinlan & Cameron-Jones 1993)
- FOIL is algorithm for learning function-free Horn clauses.
- The representation provided to the learning algorithm constits of the following background relations:

- **has_*word*(Page):** Each of these Boolean predicates indicates the pages in which the word *"word"* occurs.

- **link_to(Page, Page):** represents the hyperlinks that interconnect the pages.

Two of the rules learned by FOIL for classifying pages, and their test-set accuracies:

Student(A):- not(has_*data*(A)), not(has_*comment*(A)), Link_to(B,A), has_*jame*(B), has_*paul*(B), not(has_*mail*(B)).

Test Set: 126 Pos, 5 Neg

Faculty(A):- has_*professor*(A), has_*ph*(A), link-to(B,A), has_*faculti*(B).

Test Set: 18 Pos, 3 Neg

13

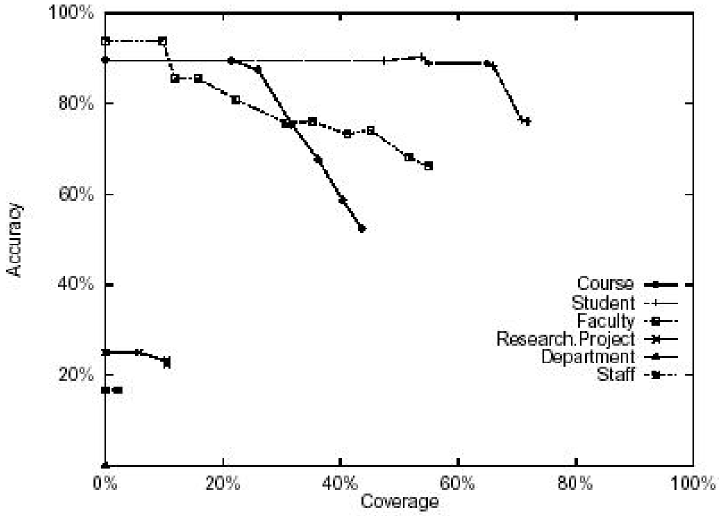

Figure 3: Accuracy/coverage for FOIL classifiers.

14

Identifying Multi-Page Segments

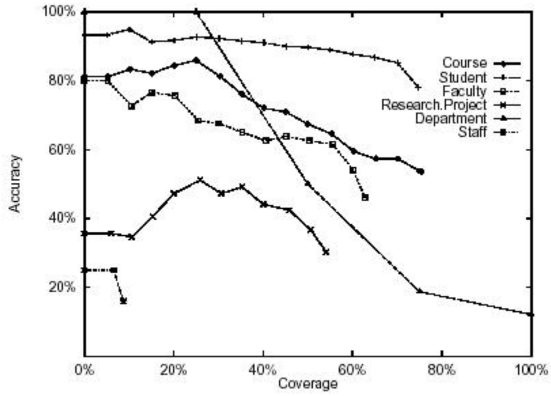

Figure 5: Accuracy/coverage for the statistical text classifiers after the application of URL heuristics.

15

Recognizing Relation Instances

Relations among class instances are often represented by hyperlink paths. The task of learning to recognize relation instances involves rules that characterize the prototypical paths of the relation.

class(Page): for each class, the corresponding relation lists the pages that represent instances of *class.*

link_to(Hyperlink,Page, Page): represents the hyperlinks that interconnect the pages in the data set.

has_word(Hyperlink): indicates the words that are found in the anchor text of each hyperlink.

all_words-capitalized(Hyperlink): hyperlinks in which all of the words in the anchor text start with a capital letter.

has_alphanumeric_word(Hyperlink): hyperlinks which contain a word with both alphabetic and numeric characters.

has_neighborhood_word(Hyperlink): indicates the words that are found in the neighborhood of each hyperlink.

16

The search process consists of two phases:

1) the "path" part of the clause is learned

2) additional literals are added to the clause using a hill-climbing search.

Two of the rules learned for recognizing relation instances, and their test-set accuracies.

members_of_project(A,B):-research_project(A), person(B), link-to(C,A,D), link_to(E,D,B), neighborhood_word_people(C).

Test Set: 18 Pos, 0 Neg

Department_of_person(A,B) :- person(A), department(B), link-to(C,D,A), link_to(E,F,D), link_to(G,B,F), neighborhood_word_graduate(E).

Test Set: 371 Pos, 4 Neg

17

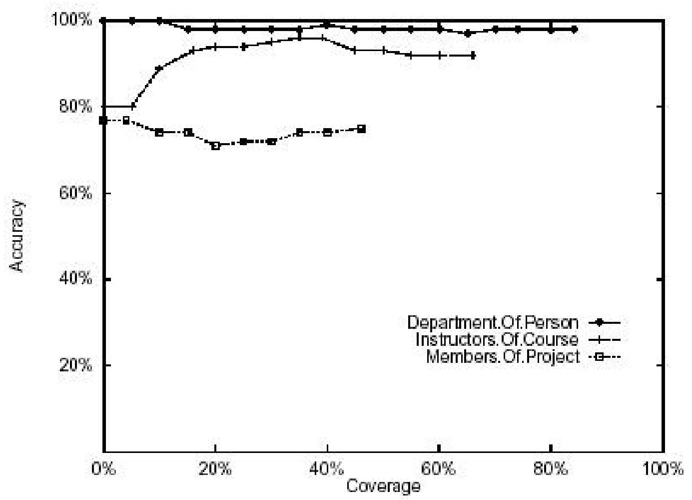

Figure 7: Accuracy/coverage for learned relation rules.

Extracting Text Fields

In some cases, the information will not be represented by Web pages or relations among pages, but it will be represented by small fragments of text embedded in pages.

Information-extraction learning algorithm SRV

- **Input:** a set of pages labeled to identify instances of the field wanted to extract
- **Output:** a set of information-extraction rules
- Positive example is a labeled text fragment – a sequence of tokens – in one of the training documents
- Negative example is any unlabeled token sequence having the same size as some positve example

19

The representation used by the rule learner includes the following relations:

length(Fragment, Relop, N): specify the length of a field, in terms of number of tokens, is less than, greater than, or equal to some integer.

some(Fragment, Var, Path, Attr, Value): posit an attribute-value test for some token in the sequence (e.g. capitalized token)

position(Fragment, Var, From, Relop, N): say something about the position of a token bound by asome-predicate in the current rule. The position is specified relative to the beginning or end of the sequence

relpos(Fragment, Var1, Var2, Relop, N): specify the ordering of two variables(introduced by some-predicates in the current rule) and distance from each other.

The data set consists of all Person pages in the data set. The unit of measurement in this experiment is an individual page. If SRV's most confident prediction on a page corresponds exactly to some instance of the page owner's name, or if it makes no prediction for a page containing no name, ist behavior is counted as correct.

20

```
ownername(Fragment) :- some(Fragment, B, [], in_title, true),
           length(Fragment, <, 3),
           some(Fragment, B, [prev_token], word, "gmt"),
           some(Fragment, A, [], longp, true),
           some(Fragment, B, [], word, unknown),
           some(Fragment, B, [], quadrupletonp, false)
```

```
Last-Modified:  Wednesday, 26-Jun-96 01:37:46 GMT

<title>Bruce Randall Donald</title>

<h1>
<img
src="ftp://ftp.cs.cornell.edu/pub/brd/images/brd.gif">
<p>
Bruce Randall Donald<br>
Associate Professor<br>
```

Figure 8: **Top:** An extraction rule for name of home page owner. This rule looks for a sequence of two tokens, one of which (A) is in a HTML title field and longer than four characters, the other of which (B) is preceded by the token gmt, unknown from the training data, and not a four-character token. **Bottom:** An example HTML fragment which the above rule matches.

21

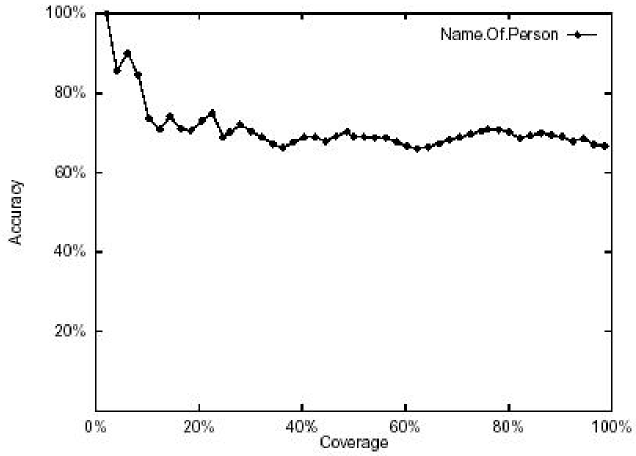

Figure 9: Accuracy/coverage for learned name-extraction rules.

22

The Crawler

- A Web-crawling system that populates a knowledge base with class and relation instances as it explores the Web.The system incorporates trained classifiers for the three learning tasks: recognizing class instances, recognizing relation instances and extracting text fields.
- The crawler employs a straightforward strategy to browse the Web.
- After exploring 2722 Web pages, the crawler extracted 374 new class instances and 361 new relation instances.

	Student	Faculty	Person	Project	Course	Dept.	Instruct.Of	Members.Of.Project	Department.Of
Extracted	180	66	246	99	28	1	23	125	213
Correct	130	28	194	72	25	1	18	92	181
Accuracy	72%	42%	79%	73%	89%	100%	78%	74%	85%

Table 1: Page and relation classification accuracy when exploring the CMU computer science department Web site.

Extracting Semantic Relationships between Terms:
Supervised vs. Unsupervised Methods

Michael Finkelstein-Landau
Emanuel Morin

25

Iterative Acquisition of Lexico-syntactic Patterns

- Supervised System PROMÉTHÉE for corpus-based information extraction
- extracts semantic relations between terms
- built on previous work on automatic extraction of hypernym links through shallow parsing(Hearst, 1992, 1998).
- Additionally the system incorporates a technique for the automatic generalization of lexico-syntactic patterns that relies on a syntactically-motivated distance between patterns

26

- The PROMÉTHÉE system has two functionalities:
  - The corpus-based acquisition of lexico-syntactic patterns with respect to a specific conceptual relation
  - The extraction of pairs of conceptual related terms through a database of lexico-syntactic patterns

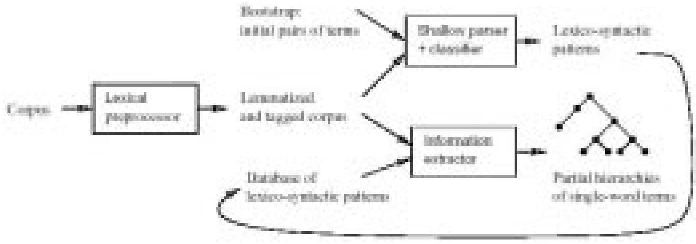

Figure 1: The information extraction system PROMÉTHÉE

27

Shallow Parser and Classifier

A shallow parser is complemented with a classifier for the purpose of discovering new patterns through corpus exploration. This purpose (Hearst1992,1998), is composed of 7 steps:

1.) Select manually a representative conceptual relation, for instance the hypernym relation.

2.) Collect a list of pairs of terms linked by the selected relation. The list of pairs of terms can be extracted from a thesaurus, a knowledge base or can be manually specified. For instance, the hypernym relation *neocortex IS-A vulnerable area* is used.

28

3. Find sentences in which conceptually related terms occur.

- sentences are lemmatized, and noun phrases are identified.
- sentences are represented as lexico-syntactic expressions.

For instance, the relation HYPERNYM(vulnerable area, neocortex) is used to extract from the corpus [Medic] the sentence:

Neuronal damage were found in the selectively <u>vulnerable areas</u> such as <u>neocortex</u>, striatum, hippocampus and thalamus.

The sentence is then transformed into the following lexico-syntactic expression:

NP find in <u>NP</u> such as <u>LIST</u>

29

4. Find a common environment that generalizes the lexico-syntactic expressions extracted at the third step. This environment is calculated with the help of a measure of similarity and a procedure of generalization that produce candidate lexico-syntactic pattern.

For instance, from the previous expression, and another similar one, the following candidate lexico-syntactic pattern is deduced:

<u>NP</u> such as <u>LIST</u>

5. Validate candidate lexico-syntactic patterns by an expert

6. Use new patterns to extract more pairs of candidate terms

7. Validate candidate pairs of terms by an expert, and go to step 3.

30

Through this technique, lexico-syntactic patterns are extracted from a technical corpus. These patterns are then exploited by the information extractor that produces pairs of conceptual related terms.

31

Automatic Classification of Lexico-syntactic Patterns

Step 4. of the described algorithm acquires automatically lexico-syntactic patterns by clustering similar patterns.
As indicated in step 3. the relation HYPERNYM(vulnerable area, neocortex) instantiate the pattern:

NP find in <u>NP</u> such as <u>LIST</u>

From the relation HYPERNYM(complication, infection) and the sentence:

therapeutic complications such as infection, recurrence, and loss of support of the articular surface have continued to plague the treatment of giant cell turmor is extracted through corpus exploration;

a second lexico-syntactic expression is produced:

<u>NP</u> such as <u>List</u> continue to plague NP

32

This lexico-syntactic expressions can be abstracted as:

$$A = A_1\ A_2\ \ldots\ A_j\ \ldots\ A_k\ \ldots A_n \text{ with} \begin{cases} \text{RELATION}(A_j, A_k) \\ k > j+1 \end{cases}$$

and:

$$B = B_1\ B_2\ \ldots\ B_{j'}\ \ldots\ B_{k'}\ \ldots B_{n'} \text{ with} \begin{cases} \text{RELATION}(B_{j'}, B_{k'}) \\ k' > j'+1 \end{cases}$$

33

Let *Sim*(A,B) be a function measuring the similarity of lexico-syntactic expressions *A* and *B* that relies on the following hypothesis:

Hypothesis 2.1 (Syntactic isomorphy)

If two lexico-syntactic expressions A and B indicate the same pattern then, the items A_j and $B_{j'}$, and the item A_k and $B_{k'}$ have the same syntactic function.

Let Win1(A) be the window built from the first through j-1 words, Win2 (A) be the window built from words ranking form j+1 through k-1th words, and Win3(A) be the window built from k+1th trough nth words(Fig.2). The similarity function is defined as follows:

34

$$Sim(A,B) = \sum_{i=1}^{3} Sim(Win_i(A), Win_i(B))$$

with

$$\begin{cases} Win_1(A) = A_1 A_2 \cdots A_{j-1} \\ Win_2(A) = A_{j+1} \cdots A_{k-1} \\ Win_3(A) = A_{k+1} \cdots A_n \end{cases} \text{ and } \begin{cases} Win_1(B) = B_1 B_2 \cdots B_{j'-1} \\ Win_2(B) = B_{j+1} \cdots B_{k'-1} \\ Win_3(B) = B_{k+1} \cdots B_{n'} \end{cases}$$

The function of similarity between lexico-syntactic patterns Sim(Wini(A), Wini(B)) is defined experimentally as function of the longest common string.

All lexico-syntactic expressions are compared two by two previous similarity measure, and similar lexico-syntactic expressions are clustered. Each cluster is associated with a candidate lexico-syntactic pattern. For instance, the sentences introduced earlier generate the unique candidate lexico-syntactic pattern:

<u>NP</u> such as <u>LIST</u>

35

Term Level Text Mining

- The unsupervised system combines ideas of term identification and term and term relationship extraction for term-level text mining.
- The overall purpose of the system:
  - find interesting relationships between terms and to label these relationships.
- The system uses NLP techniques in order to increase confidence in both extracting terms and identifiying relationships.
  - Lemmatizing
  - shallow parsing

36

Multi-word term recognition methods are used for finding relationships between terms. In particular, similar association measures that in previous literature(Daille, 1996; Smadja, 1923) were used for term and collocation extraction are implemented in this work for extracting relations between terms.

The system requires only few manual definitions and avoids the need to know the relevant lexico-syntactic pattern in advance.

37

Term Extraction

- The system extracts various term patterns from the corpus:
  - Simple term patterns: Adjective-Noun(ADJ-N), Noun-Sequence(NSEQ), Noun-Preposition-Noun(N-PREP-N) and Proper Name(PN).
  - Syntactic relations: Verb-Object(VB-OBJ) and Subject-Verb(SUBJ-VB).
  - Semantic relations: IsA and Has A.
- The extraction process is preceded by a module for tagging, lemmatizing and shallow parsing the documents.

38

Term Typing an Filtering

- This stage is intended to determine which terms become in focus, since the extraction process yields enormous mumber of term candidates.
- Using a predefined list of term types, some terms are typed and become in focus regardless of their distributional properties.
- Others are scored according to classical scoring criteria in order to filter out non-relavant combinations

39

Defined Term Types

1. Merger terms:
   1. terms containing the substring „merge“, which refer to merger events.
   2. Example: merger agreement, merger of airline, and announce merger.
2. Product terms:
   1. terms that form a product.
   2. Example, the object in a VB-OBJ term where VB = „produce“(oil in produce oil) and the last noun in a N-PREP-N term where the first noun is „procuction“ and PREP = „of“ (camera in production of camera).
3. Company-Name terms:
   1. proper names containing substrings that tend to appear within company names like „Ltd“ „Corp“, „Co“ and „Inc“.
   2. For example: Lloyds Bank NZA Ltd. And Utah International Inc.

The assumption is that finding term types is not difficult using local cues andpredefined list of types.

40

Term Associations and Labeling Associations

- Relationships between terms are identified according to co-occurrences association calculation. The relationships differ by two factors:
  - Types of co-occurrences: some relations are better identified using term co-occurrences in joint sentences, while for others co-occurrences in joint documents give better results.
  - Types of scores: Mutual Information for example, discriminates in favor of rare events while Log Likelihood behaves in an opposite way, thus different association measures can identify different conceptual relations.

Company-Name	Product	Score
U.S. Group CPC International	corn	47.07
Lex Vehicle Lease Ltd.	car	18.59
McDonnell Douglas Corp	aircraft	15.86
Heinz-UFE Ltd.	cereal	13.68
Alliant Computer System Corp	software	10.32
Honda Motor Corp	car	9.28
McDonnell Douglas Corp	jet	8.50

Table 1: Relationships between Company-Name terms and Product terms

Company-Name	Company-Name	Merger	Score
Alpha Health Systems Corp	REPH Acquisition Co	merger proposal	30.88
MTS Acquisition Corp	Seton Co	plan of merger	30.88
Bristol-Myers Co	SciMed Life Systems Inc	agreement of merger	16.43
Comdata Network Inc	Lambert Inc	finance merger	16.43
Electrospace Systems Inc	Chrysler Corp	merger agreement	16.43
Industries Inc	Hayes-Albion Corp	vote on merger	16.43
Trust Co	Independence Ban Corp	complete merger	16.43

Table 2: Relationships between Merger terms and Company-Name terms

43

The Merge Relation

Merge relation Merge(CN_1, CN_2), where
CN_1 and CN_2 are both Company-Name terms
that participates some merger event (merger in progress, actual, etc.).

The first experiment evaluated the performance of PROMETHEE system as a stand-alone system.

Two manually defined lexico-syntactic patterns:

merger of CN_1 with CN_2

Dixons Group Plc said shareholders at a special meeting of Cyclops Corp approve the previously announced merger of Cyclops with Dixons

merger of CN_1 and CN_2

Hoechst Celanese was formed Feb 27 by the merger of Celanese Corp and American Hoechst Corp

44

Then, all instances of those patterns were extracted from the corpus, and PROMETHEE incrementally learned more patterns for the Merge relation. The new patterns learned were:

CN_1 said it complete * acquisition of CN_2
Chubb Corp said it completed the previously announced acquisition of Sovereign Corp

CN_1 said it shareholder * CN_2 approve * merger of the two company
INTERCO Inc said its shareholders and shareholders of the Lane Co approved the merger of the two companies

CN_1 said it shareholder approve * merger with CN_2
Fair Lanes Inc said its shareholders approved the previously announced merger with Maricorp Inc a unit of Northern Pacific Corp

CN_1 said it agree * to (acquire|buy|merge with) CN_2
Datron Corp said it agreed to merge with GGFH Inc a Florida-based company formed by the four top officers of the company

45

101 pairs of terms (class A) conceptually related have been extracted from the corpus.

The second experiment was performed on the integrated system.
At first, Merger terms and Company-Name terms were extracted from the corpus. For the 350 Merger terms (*e.g. merger talk*, *approve merger*, *merger transaction*) and 4500 Company-Name terms (*e.g. Texas Bancshare Inc*, *Bank of England*) that were found, a ranked list of 263 conceptually related triples within the Merge relation was generated using an automatic relationship identification module.
Each triple included the merger description and two companies. The triples became pairs by leaving only the two related company names to be given as initial training input to the learning system (class C). The PROMETHEE system discovered again patterns 1, 3,4, 5, 6, and a new pattern:

CN_1 said it sign * to (acquire|buy|merge with) CN_2
Dauphin Deposit Corp said it signed a definitive agreement to acquire Colonial Bancorp Inc

46

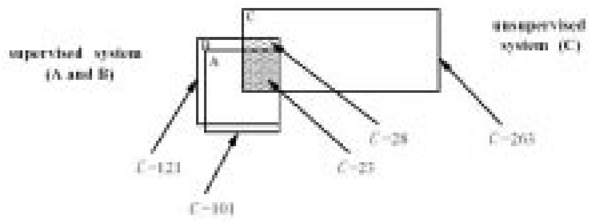

Figure 3: Overlap of the pairs of terms for the Merge relation

47

---

neumann:

Not covered in coruse

Discovering Conceptual Relations from Text

Alexander Maedche and Steffen Staab

A new approach to discover non-taxonomic conceptual relations from text building on shallow text processing techniques. A generalized association rule algorithm is used, that does not only detect relations between concepts, but also determines the appropriate level of abstraction at which to define relations.

Architecture of SMES

- Tokenizer
- Lexicon
- Lexical Analyses
- Chunk Parser
- Dependency Relations
- Heuristics

49

Learning Algorithm

Consists of the following rules:

Set of concepts	C:= $\{a_i\}$
Set of concept pairs:	CP:=$\{(a_{i,1},a_{i,2})\|a_{i,j}$ € C}
Set of transactions:	T:=$\{t_i\|i = 1 ...n\}$ where each transaction t_i consists of a
Set of items:	t_i:=$\{a_{i,j}\|j= 1 ... m_i, a_{i,j}$ € C} and each item $a_{i,j}$ is from a set of concepts C
Taxonomic relation:	$H \subset C \times C$

50

The algorithm computes association rules

$X_k \Rightarrow Y_k (X_k, Y_k \subset C,\ X_k \cap Y_k = \{\})$ such that measures for support and confidence exceed user-defined thresholds.

Support of a rule $X_k \Rightarrow Y_k$ is the percentage of transactions that contain $X_k \cup Y_k$ as a subset.

Confidence for $X_k \Rightarrow Y_k$ is defined as the percentage of transactions that Y_k is seen when X_k appears in a transaction

$$(3)\quad \text{support}(X_k \Rightarrow Y_k) = \frac{|\{t_i \mid X_k \cup Y_k \subseteq t_i\}|}{n}$$

$$(4)\quad \text{confidence}(X_k \Rightarrow Y_k) = \frac{|\{t_i \mid X_k \cup Y_k \subseteq t_i\}|}{|\{t_i \mid X_k \subseteq t_i\}|}$$

51

The Algorithm, described in the following four steps, summarizes the learning module:

1. Determine:
   $T := \{\{a_{i,1}, a_{i,2}, \ldots, a_{i,m'i}\} \mid (a_{i,1}, a_{i,2}) \in CP \wedge l \geq 3 \rightarrow ((a_{i,1}, a_{i,1}) \in H \vee (a_{i,2}, a_{i,1}) \in H)\}$
2. Determine support for all association rules $X_k \Rightarrow Y_k$, wobei $|X_k| = |Y_k| = 1$.
3. Determine confidence for all association rules $X_k \Rightarrow Y_k$ that exceed user-defined support in step 2.
4. Output association rules that exceed user-defined confidence in step 3 and that are not pruned bz ancestral rules with higher or equal confidence and support.

52

Example Sentences

a. *Mecklenburg's* schönstes *Hotel* liegt in Rostock.

b. Ein besonderer Service für unsere Gäste ist der *Frisörsalon* in unserem *Hotel.*

c. Das Hotel Mercure hat *Balkone* mit direktem *Strandzugang*.

d. Alle *Zimmer* sind mit *TV*, Telefon, Modem und Minibar ausgestattet**.**

Table 1. Examples for linguistically related pairs of concepts

Term_1	$a_{i,1}$	Term_2	$a_{i,2}$
Mecklenburgs	area	*hotel*	hotel
hairdresser	hairdresser	*hotel*	hotel
balconies	balcony	*access*	access
room	room	*TV*	television

53

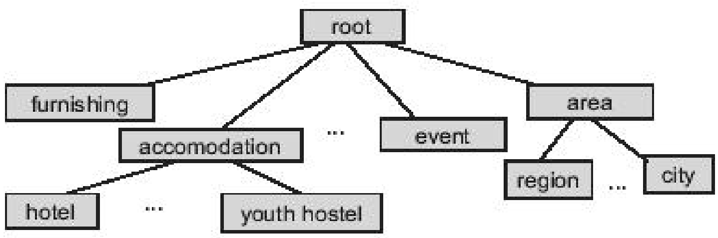

Figure 1. An example scenario

Table 2. Examples of discovered relations

Discovered relation	Confidence	Support
(area, accomodation)	0.38	0.04
~~(area, hotel)~~	~~0.1~~	~~0.03~~
(room, furnishing)	0.39	0.03
~~(room, television)~~	~~0.29~~	~~0.02~~
(accomodation, address)	0.34	0.05
(restaurant, accomodation)	0.33	0.02

54

Evaluation

- Analyzed HTML documents: 2234, 16 million words and HTML tags
- 51.000 linguistically pairs (as in Table 1)
- The modeled Ontology contained 284 concepts and 88 non-taxonomic conceptual relations.

Generic Relation Learning Accuracy (RLA)
Defined to capture intuitve notions for relation like:
"utterly wrong", "rather bad", "near miss" and *"direct hit".*
RLA ist the averaged accuracy that the instances *d* of dicovered relations *D* match against their best counter´-parts from *R*.

55

Table 3. Evaluation Results — number of discovered relations, $\overline{\text{RLA}}$, recall, precision

	Confidence			
Support	0.01	0.1	0.2	0.4
0.0001	2429 / 0.55 66% / 2%	865 / 0.57 31% / 3%	485 / 0.57 18% / 3%	238 / 0.51 2% / 1%
0.0005	1544 / 0.57 59% / 3%	651 / 0.59 30% / 4%	380 / 0.58 17% / 4%	198 / 0.5 1% / 1%
0.002	889 / 0.6 47% / 5%	426 / 0.61 27% / 6%	245 / 0.61 16% / 6%	131 / 0.52 1% / 1%
0.01	342 / 0.64 31% / 8%	225 / 0.64 19% / 8%	143 / 0.64 14% / 8%	74 / 0.53 1% / 1%
0.04	98 / **0.67** **13%** / **11%**	96 / **0.67** 11% / 10%	70 / 0.65 6% / 7%	32 / 0.51 0% / 0%
0.06	56 / 0.63 6% / 9%	56 / 0.63 6% / 9%	48 / 0.62 3% / 6%	30 / 0.53 0% / 0%

56