Language Technology and the Semantic Web

Dr. Günter Neumann
http://www.dfki.de/~neumann
Language Technology-Lab
DFKI, Saarbrücken
Overview

- Language Technology
- Semantic Web
- Information Extraction
- Information Access
Human Language Technology

- Human Language Technology LT – covers
  - The design and implementation of algorithms, data and electronic devices for processing of natural language (text and speech), and
  - Their integration into real-world applications and products

- Language Technology defines the engineering part of computational linguistic
LT-methods cover many areas

Who won the ESC 2004?

Multi/cross-linguality is of great importance in all these areas!
LT as embedded part of applications

- Human-Machine Communication
- Data-oriented Knowledge Acquisition

Integration
- Modularity
- Multi-media
- Software-Engineering standards

High Performance
- Real-time
- Robustness
- Scalability
- Adaptation
- Evaluation

7/2004, GN
# Language Technology

## LT-Methods

### Core technology
- Efficient data structures
- Weighted finite state automata
- Machine learning
- Statistical inference

### Already a successful technology transfer
- Industry (Microsoft, IBM, Siemens, Telekom, ...) & Spin-offs, competence centers, ...
- Speech-systems, MT, Editors, Text-Mining, Knowledge-Mining Content-Management, ...

## Newest Technology Hype: the Semantic Web

- What role does it play for LT?
The Semantic Web (SW)

- Tim Berners-Lee, 1998:
  - “This document is a plan for achieving a set of connected applications for data on the Web in such a way as to form a consistent logical web of data (semantic web).”

- Tim Berners-Lee et al., 2001
  - “… an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”
The existing web will further emerge, so that computers can understand content on-line, to better help humans to organize, search, and exchange information.

SW exists of meta-data and links to global ontologies, which define the meaning of terms.

An ontology serves as a structural vocabulary for the interpretation of domain-specific terms.

The SW does not only consider Web-pages.

- Intelligent information search;
- Automatic support for the management of my personal information on the SW
RDF and OWL: Modeling data on the SW

1. RDF: Resource Description Framework
   RDF is a language for the representation of meta-data over web resources.
   RDF-statements are triples of the form (Subj, Pred, Obj).

2. XML & N3 sind alternative RDF-Syntaxen
   XML schematically: <Subj> <Pred> Obj </Pred> </Subj>
   N3:
   @prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#.
   @prefix contact: http://www.w3.org/2000/10/swpim/contact#.
   @prefix EM: http://www.w3.org/People/EM/contact#.
   EM:me rdf:type contact:person.
   EM:me contact:full-name "Eric Miller".
   EM:me contact:personalTitle "Dr.".
   EM:me contact:mailbox rdf:resource "mailto:em@w3.org"

3. OWL: Web Ontology Language
   • some RDF-statements have a fix interpretation (is-a, =, inverseOf, card, ...)
   • Sharing of information between individuals from multiple documents ⇒
     Web of data from heterogeneous sources
   • Semantic of OWL as basis for inference mechanism over these data structures.

4. Relevante Aspekte für das SW
   standardization, Web-globalization, distribution of resources

5. Ontology Mapping
   Mapping between distributed, local ontologies
The SW-pyramid

Current focus of major efforts

Established standards

Basic research
Relevance of LT for SW

1. During the transition from WWW to the SW, LT is a core technology.

2. As long as the human is in the "Internet Loop", NL will remain to be the core Human-SW communication device.

3. Humans will also in the future exchange knowledge via NL documents: Semantically annotated documents as Human-SW interface.

4. NL-generation of information in form of NL-Text, e.g., heterogeneous resources, dynamically created reports, newspapers, ...

Intelligent Information Extraction

Intelligent Information Access

CV
IE for semantic annotation

Identification of IE-sub-tasks:
- basic entities (e.g., proper names)
- binary relations between entities
- n-ary relations/events

Automatic Content Extraction (ACE)
- Spezification of an IE-core-ontology
- Annotation-specification & tools
- Templates as specializations of the IE-core-ontology (also multi-templates)

Machine learning!

IE as core for semantic annotation
- identification
- discovery
- validation
- evaluation
of semantic relationships & as basis for the automatic creation of meta data
IE for semantic annotation

IE-core system

Domain lexicon

IE-core ontology

Domain ontology

inference engine

NL-oriented ontology

{ <t1, rel?, t2> }<NP, VG, NP> <NE, ?, NP> <NE, ofPP, NE>

LT as basis of:
- concept identification
- determination of plausible structural relation candidates
Example for entities & their mentions


Person
Organization
Geopolitical Entity
LT-challenges

Identification of verbalizations/mentioning of concepts/instances

- Linking of domain ontology and NL-oriented ontology (e.g., WordNet)
- Paraphrasing
- Metonymy ("Peking organizes the Olympic Games 2008.")
- Reference identification ("Chancellor Schröder, Schröder, the German chancellor, he, …")
- Analysis of sublanguages as basis for adaptive IE (cf. Grishman, 2001)
Domain modeling in DFKI system SMES is realised using typed feature structures

○ Domain modeling via hierarchy of templates (black box), using the formalism TDL, which is also used to model hierarchies of linguistic objects (yellow boxes).

○ The interface between domain knowledge and linguistic entities is specified via linking types (green box), which represent a close connection between concepts of the different layers, and which are accessible via the domain lexicon (brown & green box). Template filling is then realized via type expansion.

---

Diagram:

- **Fdescription**: [process, mods]
- **Phrase**: NP, PP
  - **LocNP**, **DateNP**
  - **LocPP**, **DatePP**
- **Linking Type**: [process=1, subj=2, templ=[action=1, slot=2, ...]]
- **Template**: [action, date]
  - **Move-T**: [from, to, unit]
  - **Loc-T**: [loc]
- **Fight-T**: [attacker attacked]
- **Meeting-T**: [visitor visited]
- **DomainLex**: shoot, Fight-Lex
- **Fight-Lex**: [process=1, subj=2, obj=3, templ=[action=1, attacker=2, attacked=3, ...]]
Central issues
1. Sentence-based NL analysis
2. NL-annotations for multimedia information segments

Processing of huge text collections:
1. Extraction of relevant sentences from texts.
2. Syntax analysis
3. Annotation of the texts with syntax

NL-Question
*Whose answer surprised Hillary?*

Answer related to *whom*
Haystack: the universal information client

http://haystack.lcs.mit.edu/

Motivation:
Semantic annotation should be a side-effect of daily use of computer.

Idea:
Personalized information portal for all relevant services, like email, documents, calender, Web-pages, ...

Collection of all data uniformly via RDF-database

Programming language Adenine for the manipulation of frequent (i.e., as support for the implementation of specific service programs).
Frage: What is the state bird of Alabama?

Antwort: Yellowhammer
Example: Linking of t-expressions & RDF

```xml
@prefix nl: http://www.ai.mit.edu/projects/infolab/start#

Add {  :Person
    rdf:type rdfs:Class ;
}

Add {  :homeAddress
    rdf:type rdfs:Property ;
    rdfs:domain :Person ;

    nl:annotation @(nl:subj "lives at" nl:obj) ;
    nl:annotation @(nl:subj "s home adress is" nl:obj) ;
    nl:annotation @(nl:subj "s apartment" nl:obj) ;

    nl:generation @(nl:subj "s home address is" nl:obj) ;
}
```

Remarks:
- NL-annotations as a means for controlling the paraphrasing potential of NL expressions
- Richer linguistic annotations are possible (e.g., fine-grained grammatical functions, agreement)
- Also relevant for user-oriented adaptation of service programs
Natural language annotations for the SW

- **NL used as meta-data**
  - Readability of RDF
  - Supports transition from WWW to SW
  - NL-annotation specifies which kind of (NL)-question a meta-data is able to answer
    ⇒ controlled question-answering systems

- **Information access (IA) within SW**
  - Development of programs, which help a user to locate, to collect, to compare and to link information

- **NL is the most natural way for user to perform IA**
  - SW should support in the same way IA using specialized languages/exchange formats & NL
Relevance

- Approach is open for future extensions:
  - statistical-based models (add weight to the NL-annotations)
- The current mechanism of NL-annotations is idiosyncratic, however at DFKI we plan the following:
  - Exploration of a linking mechanism between dependency structure and RDF/OWL
  - Foundation for novel template-based QA-strategies
Example for the processing of complex questions

- **Approach:**
  - Select templates via Q-Type & Q-Focus:
    - Definition question, list-question
    - Person: born-where, born-when, business-what ⇒ Ontology
  - Property P, select IR-Schema:
    - NL-based query-pattern
  - P might be:
    - From the set of known NE-types (person, location, date, …) ⇒ answer-type
    - NL-Phrase, which “describes” P, in case no a-type can be determined
  - Compute for each P für jede P one/ several IR-Query-terms, e.g.,
    - NE-type:person & text:<query term>

- Example for the processing of complex questions
  - "Wer ist Thomas Mann?"
  - Q-type=c-definition, focus=<Person, „Thomas Mann“>
  - IR-Schemata:
    - <PERSON> “geboren in” <LOCATION>
  - "(neTypes:LOCATION AND +geboren
    (text:"Thomas Mann" OR text:Mann))"
IE-based question answering

- Approach can also be used for template-based questions:
  - let $t \in T$, set of templates, which are known to the system – via IE-Ontology – e.g., “management-succession-Template”
  - for all properties $E$ of $t$, combine $E$ with NL-schema
    - E.g., “Person-In” $\Rightarrow (<PERS> \text{“is_successor_of”} <PERS>)$

- Answering of complex questions
  - As composition of the answering of – relative to the conceptual description – simple questions
  - Implementation of this approach as part of the DFKI project Quetal (prototype as part of DFKI’s qa@clef 2004 system)
  - Interactive online IE through close integration of IE & IA
Concluding remarks

- LT is a key technology for the construction of the Semantic Web

- Very high requirements on
  - Performance
  - Modularity & integration
  - scalability & on-demand availability
  - Domain & user adaptation

- Systematic evaluation of LT-methods
  - Driving power & revisions of future developments

- In the future, cognitive-based methods will be considered
  - as inspiration for more effective LT-methods