Projekt Maschinelle Übersetzung (Teil I)

Walther v. Hahn, Cristina Vertan
Universität Hamburg • Fachbereich Informatik
AB. Natürlichsprachliche Systeme

WWW: http://nats-www.informatik.uni-hamburg.de/
E-Mail: {vhahn, vertan}@informatik.uni-hamburg.de

Aims

- To provide an overview on the basic principles, problems and solutions in MT-Systems
- To go into details in one of the modern MT-approaches (EBMT)
- To make a „hands-on“ exercise in EBMT in a project group-work scenario
- To show what evaluation of MT-Systems imply

Ablauf

Basic Methods of Natural Language Processing for MT

Walther v. Hahn
University of Hamburg • Informatics Department
Natural Language Systems Group

WWW: http://nats-www.informatik.uni-hamburg.de/~vhahn/
E-Mail: cri@informatik.uni-hamburg.de

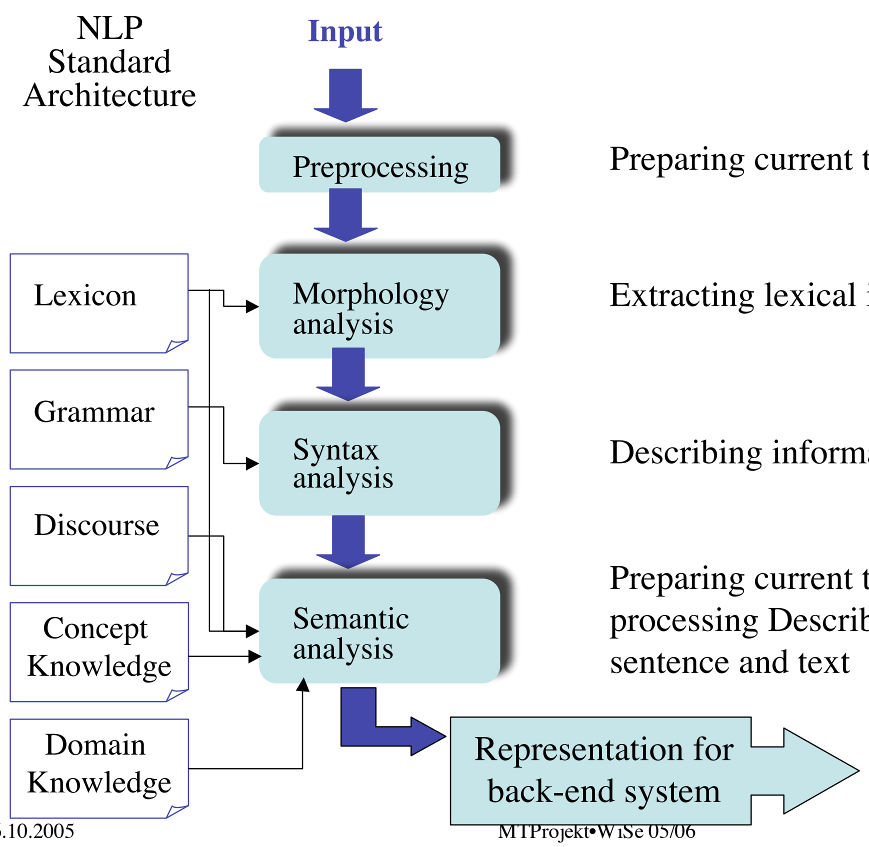
Why starting with NLP

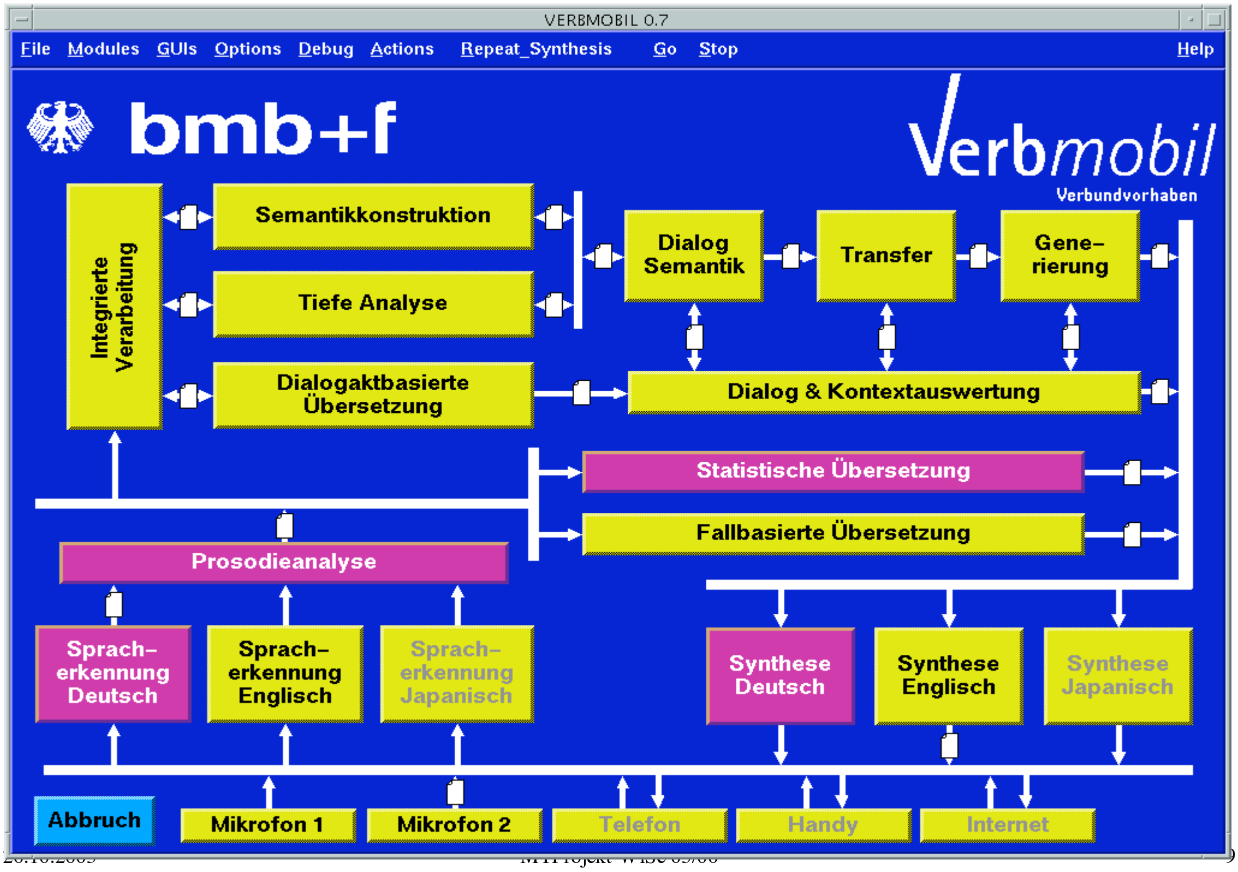
- MT has much in common with all other language technologies:
  - The basic features of natural languages
  - Missing situational data
  - Limited modal channels
  - Limited technology
  - New words/names
  - Non-deterministic processing

Typical Features of Natural Language

- Unclear focus of analysis, esp. with spoken input (the whole text only?)
- Ambiguity on all levels
- Self-reference, meta language capacity (I meant "lion")
- Valencies, i.e. syntactic/semantic co-occurrences of categories
- Multi-word lexemes and idioms with non-compositional meaning
- Hierarchical syntax in non-linear order
- Long distance dependencies
- Discontinuous components
- Ellipses
- Paraphrases
- Coherence
- Understanding by word knowledge

(Imagine these phenomena in programming languages …)





Some Central Decisions of Analysis

- Which type of language is expected?
  - Spoken input may contain errors
  - Spoken language style (even in written transcripts) may have syntactic and semantic "errors"
  - Written language has no prosody, but is supposed to be correct
- Which type of output?
  - Table, representation expressions, slot filler, classification
- Which domain?
  - Technical, social, leisure,
- Which pragmatics?
  - Question answering, action control, information,

Result: GUS
Semantic Slot Fillers

"I want to go to San Diego on May 28"

↓

(Client Declare
  (Case for *want* / e (Tense Present)
    Agent = Dialog.Client.Person
    Event = (Case for *go* (Tense Present)
      Agent = Dialog.Client.Person
      To-Place = (Case for City
        Name = *San Diego*)
      Date = (Case for Date
        Month = *May*
        Day = *28*))))

Winograd

Result: DB-Interface
Expression

"List the names of all suppliers, who deliver at least the parts that are delivered by supplier S2"

↓

```
SELECT          UNIQUE  S#
FROM            SP  SP X
WHERE           NOT EXISTS
                (SELECT *
                FROM SP SP Y
                WHERE S# = ´S2´
                AND             NOT EXISTS
                                (SELECT *
                                FROM SP
                                WHERE S# = SP X. S#
                                AND P# = SP Y. P#))
```

Date

Result: Table of chemical reactions in SIE

REF	SCALE	PHASE	YIELD	TEMP
para.1	small	solid	77%	-78 to 20
TIME	ENERGY	APPARATUS	FEATURES	
	cooling		IR. NMR. MS	
REG. NO	FUNCTION	AMT.	AUTHOR ID	
78624-62-1	product	2.70 g	7a	
78624-61-0	reactant		6a	
13274-48-6	reactant	1.24 g	N-methyltriazolinedione	
	solvent	80 ml	pentane	
	solvent	40 ml	ethyl acetate	

Result: Semantic Representation

"who is the man that leads the LOKI project?"

[request (referent(_5747))
presuppose (exists (_4340))]
some (_4340)
[unique (_4407)
single (_4407)
instance (_4407, person)
propval (person,_4407,sex,male)
[some (_4725)
[unique (_5033)
single (_5033)
instance (_5033,project)
propval (project,_5033,name,str (LOKI))]

instance (_4725,leading)
propval (leading,_4725,theta,_5033)
propval (leading,_4725,alpha,_4407)
topic (_4407)]]
instance (_4340,identity)
propval (identity,_4340,alpha,_4407)
propval (identity,_4340,theta,_5747)
topic (_4407)
[some (_5747)
single (_5747)
instance (_5747,person)]

First Step: Preprocessing

- Separate text from non-text (images, code, analyze tables ...)
- Lemmatization (Splitting)
- Normalize writing (e.g. Ablaut)

} Lemmatizer

- Join separable suffixes (esp. in German)
  - "Er fing die Maus ein" ⇒ *einfangen*
  - *(*He caught the mouse in)*
- Separation of compounds
- Block multiword terms and idioms,

} Tokenizer

- (Attach PoS PoS-Tagger)

Second Step: Consulting the Lexicon

- Full form lexicon
  - Every form of a word is an entry in the Lexicon
  - Result: no morphological processes after tokenizing
  - For real life applications sometimes too large
  - Difficult for languages with strong composition
- Stem lexicon
  - Only stems are entries
  - Additional information about inflexion class
  - a morphological generator is necessary
  - Result: small resources

All further processes, except for names,
rely on lexicon information, at least on part of speech tags

Resources: Lexicon

- **Dictionary**
  - Pronunciation
  - Definitions
  - etymological information
  - stylistic information
  - PoS (part of speech)
  - Few sub-classification features (usually gender, plr.)
  - Translation (in bi- or multilingual dictionaries)
- **Lexicon**
  - PoS (noun, Verb, etc.)
  - Sub-classification features (verb transitive/intransitive, Genus etc.)
  - Inflexion classes
  - semantic information (e.g. if a verb requires an alive Subject)
  - a link to translation equivalents in other lexicons, or a mark for lexical gap (in bi- or multilingual dictionaries)

Resources: Lexicon

Representation and Encoding - 1 -

- There is a huge number of lexicon formats according to:
  - the encoded linguistic information (which features, in which order)
  - the encoding schema (distributed lexicon, delimiters between linguistic categories or entries, pointer to entries in other lexicons)
- The lexicon design is an extremely time consuming process, therefore „re-usability“ has high priority on the agenda of lexicon developers,,
- Several standard models have been proposed (PAROLE / SIMPLE, MILE) as well as standard encoding schemas based on XML (SALT, OLIF)

Resources: Lexicon
Representation and Encoding - 2 -

- Most existing standard models are very complicated because they intend to cover a large spectrum of linguistic features but
  - they still did not succeed to model all linguistic phenomena of the European languages,
  - Existing lexicons, which do not follow these standards, cannot be re-used.
  - MANAGELEX (Univ. Hamburg) is a tool for reformatting, editing, developing and merging of lexicons (under development)

Pattern matching - 1 -

- Pattern = a syntactic frame for lexical-semantic equivalence classes.
- Patterns describe frequent expressions of a language

Wo is president / mayor / king of Germany / Hamburg / Denmark ?

Pattern matching - 2 -

- Patterns specify
  - A fixed word order
  - Places for variable words
  - A clause can be processed, if there is (at least) one matching pattern
- Patterns can be filled by categorical, syntactic or semantic constraints

  " _any word_" / "_any noun_" / "_any noun+sing_" / "_any noun with attr +living_" / "any noun ⊃ class MUSHROOMS_"
- For one sentence more than one pattern may be applied

Problems: syntactically or semantically deviant sentences can be accepted, if the equivalence classes are not defined tightly:
  - z. B: Who is the pencils of football league ?

Maintenance problems with large number of (partly overlapping) patterns

⇒ Patterns do not deliver a symbolic description

Basic Syntactic Decisions

- **System type** Parser&Grammar vs separate grammar modules
- **Result of parser** Full vs selective
- **Type of grammar** Dependency vs constituency
- **Formalism** any sort
- **Architecture** parallel vs sequential
- **Start point** top-down vs bottom-up
- **Rule application** deterministic vs non-deterministic
- **Rule choice** first guess vs informed choice
- **Strategy** breadth first vs depth first
- **Scope** word by word vs phrase by phrase
- **Ambiguity handling** until success vs exhaustive

Syntactic analysis

- The output of the morphological analysis is parsed according to the chosen grammar
- i.e: parsing of a sequence of PoS symbols (retrieved by the morphological analysis)
  - the correct order of the PoS is proved
  - Iteratively a structural description is written into a data structure
  - <Structure= Art + N + ...>
- Very often a part of the input is abandoned for the moment because substructures have to be analyzed first.
- e.g *The books, which we bought yesterday are very interesting*

  <Art + N+ <Colon + sub-clause...> + V +Mod+Adj

  <Rel +Pron +V+Adv>

Resources: Grammars

- Grammars define the conditions of well-formed expressions in a language (syntax)
- Describe three basic relationships in sentences:
  - *sequence of words* (in English adjectives normally precede the nouns that they modify, whereas for e.g. in Spanish they normally follow it)
  - *categories*: e.g. a noun phrase may consist of a determiner and a noun or a determiner, an adjective and a noun.
  - *dependency* i.e. relations between categories: prepositions determine the case of the nouns which depend on them: e.g. „*mit*“ (germ.) „*con*“ (sp.) always require dative

Resources:Grammars

Grammar types

- Two basic types of grammatical representations are in common use, dependency grammars or constituency grammars.
- Sequence is optionally indicated in both types of representations
- Dependency relations are represented
  - in a dependency grammar by a word tree starting with the verb
  - In a constituency grammar by a tree of constituents
- Categories are explicitly represented in a phrase structure tree, in s dependency grammar categories are subtrees

Resources: Grammars

- Example -

S→ NP VP

NP→N

NP→AdjP NP

NP→Det NP

AdjP→Mod Adj

VP→VNP

N→Hamburg, Stadt

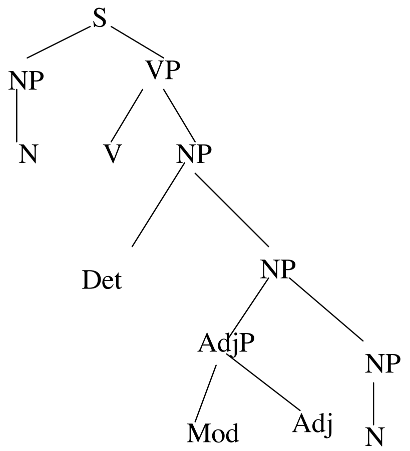
V→ ist

Det →eine

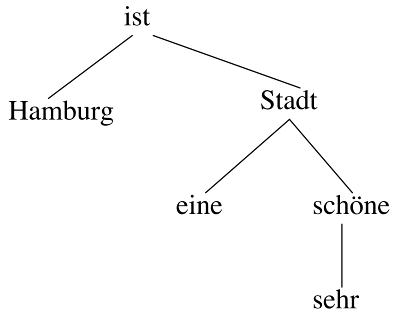
Mod→ sehr

Adj→schöne

Constituent structure



Dependency structure



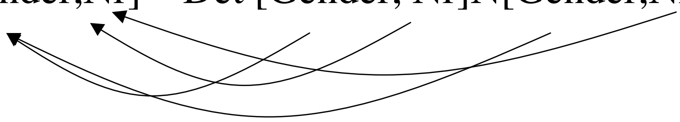
Hamburg ist eine sehr schöne Stadt

Resources: Grammars

Feature -based representations

- To shift grammatical features to higher nodes or inherit features from them
- Linguistic features are represented as attribute-value-pairs
- Additionally, rules for combining features must be specified (e.g.for correspondence)
- Features can be inserted both in constituent and dependency structures
- E.g

  NP [Gender,Nr]→Det [Gender, Nr]N[Gender,Nr]



Basic Semantic Decisions

- Semantic inside the parser? ⇒ semantic parser / case grammar
- Separation of construction, resolution and evaluation? ⇒ Multi-phase processing vs extraction
- Frame oriented processing or compositional treatment
- Domain knowledge in semantics? ⇒ Reference semantics
- Conceptual knowledge separated from facts? ⇒ interaction
- User specific interpretation? ⇒ Partner model
- Time-dependent? ⇒ time logic

Semantic Strategies

- Elementary: Key word spotting
- Basic: Syntax looks only for semantic slot fillers
- Technical solution: The parser delivers already a semantic structure by looking for semantic roles and dependencies only
- Standard: Lexical semantic entries are amalgamated with parsing result ⟵ **Semantic case grammar**
- Advanced: A full logical representation of the proposition and presupposition is built up

Semantic Roles /Deep Case Semantics

Based on lexical semantics and syntax, sentence semantics delivers the semantic **potential** of an utterance

Often used: Semantic Roles (Deep Cases):

- Actor
- Instrument
- Object, etc.

specify "persons and items" of a sentence,

E.g., in an action:

An <Actor> moves an <Object> from a <Location1> to a <Location2> along a <Path> for a <Beneficiary>.

Grammarians propose up to 25 roles. Specific domains may have a very limited number of roles (e.g. weather reports)

Semantic Resolution

Define the current meaning by

- Using contextual knowledge:
  - Determination of current values
  - Disambiguation

} Discourse Semantics

- Using domain and every-day knowledge:
  - Identification of referred objects
  - domain data and state of affairs
  - Determination of relevant utterances
  - User specific inferences

} Reference Semantics

Resources: World Knowledge and Domain Knowledge

- Concepts and relations between concepts are represented in an ontology (semantic feature hierarchy)
- The lexemes (lexicon entries) in different languages can be mapped onto this ontology (details in knowledge-based MT)
- Sensors or time dependent expressions describe the state of affair (speaker t_1 = Prof. v. Hahn = "I")
- Usually this information is language independent.

Ambiguities on all levels

The central difference between formal and natural languages is the ambiguity. E.g.,

- Speech ambiguity "*Lead a ship*" *vs.* " *leadership*"
  "*peak*" *vs.* "*peek*"
- Lexical ambiguity *"Drive to the bank, please!"*
- Syntactic ambiguity *"I saw the Grand Canon flying to New York"*
- Pragmatic ambiguity *"Can I print some reports?"*
- Referential ambiguity *"He took some papers out of the envelopes and send them to his boss*

Discourse Coherence

So far, the scope of analysis was the sentence. However, many syntactic and semantic structures are super-segmental, especially in spoken language and in spoken style.

Important tasks: Anaphora resolution		"*He wanted all of the items*"
	Kataphorics	"*John did the following:*
	Ellipses	"*and those for 2005?*"

Pronouns can often be resolved (replaced by their antecedent) by searching for an adjacent noun in the previous sentence, which has the same grammatical and semantic features (role restrictions of the verb)

Kataphorics are much more difficult, but they are rare.

Ellipses can be completed by testing unification with previous sentences

Resources: Discourse memory

Minimal case:

- List of before-mentioned objects with gender information (for easy pronoun resolution)

Best case

- List of before-mentioned objects with gender information and semantic features (for elaborated pronoun resolution)
- Memory of syntactic structures (for ellipses reconstruction)
- Memory of propositions (for ellipses reconstruction)
- Memory of speech acts (for pragmatically adequate reactions)

Pragmatics reasoning

- Pragmatics is the linguistic field, which describes relations between language and action (planning). E.g.:
- Whenever I say "*can you do X?*" I assume, that
  - ➢ it is an order, not a yes/no-question
  - ➢ I want X
  - ➢ X can be done
  - ➢ I want it to be done immediately
  - ➢ I expect a rejection in case of disagreement
  - ➢ I am responsible for X, etc

All pragmatic presuppositions and implicatures must be fulfilled in the discourse. This is especially important with legal and economic translation. Counterexample:

~~"Can you ship these trains?" → "yes, we do it immediately!"~~

Generation

- Elementary solution in simple domains: Output patterns with the result values:
- "*Who is the president of the UN* ⇒
- "*Kofi Annan*"
- Better solution: The syntax for the answer is extracted from the question Processes:
  - Consider topicalization
  - Adjust word order
  - Omit trivial parts to be not too repetitive
- Best solution: Evaluation renders propositions (predicate-argument structures) and adequate grammar rules allow for free generation.