Language engineering systems 2

- Syntactic-semantic representation
  - Functional Dependency Grammar,
  - Feature-based grammars,
  - Semantic roles,
- Parsing and interpretation
  - Unification-based chart parsing,
  - Word sense disambiguation
- Generation
Typed feature structure

```
sentence
  head-comp-phr head-DTR
  head-DTR
  var ORTH "sover"
  NONHEAD-DTR head-comp-phr head-DTR
  prep ORTH "på"
  NONHEAD-DTR unary-phr head-DTR
  noun ORTH "bildskärmen"
NONHEAD-DTR unary-phr head noun ORTH "Nova"
```

Implementing cascades

- Composing finite-state transducers
  - (or similar modules using e.g. Perl)
- Enforcing same format on all communicated data
  - E.g. TIPSTER-architecture implemented in GATE
- Iterative constraint-based filtering

Type hierarchies

```
sign
  phrase
    <HEAD-DTR sign>
    word ORTH string
  unary-phrase
    <NONHEAD-DTR sign>
```

Functional Dependency Grammar

Parsing

- Based on Constraint Grammar (SWECG)
  - Rules are used to SELECT or REMOVE interpretations
  - The rules are applied if conditions on the context are fulfilled, e.g. REMOVE (V) IF (-1C DET)
  - Processing phases:
    - Morphological analysis (all possible readings collected from lexicon)
    - Morphological heuristics (words that are not in the lexicon)
    - Morphological disambiguation (remove faulty readings)
    - Morpho-syntactic mapping (verb, head, modifier)
    - Syntactic disambiguation (remove faulty readings)
  - Ambiguities may remain, but are kept at the lexical level

Analysis: Cascading processors

- Tokenisation
- Part-of-speech tagging of tokens
- Chunking of tokens
- Relating / joining chunks
- Solving coreference relations
- Building discourse structure
- …

Functional Dependency Grammar

- FDG make dependencies explicit in a tree structure
- Context tests in the rules use information about the heads and the depending words, and valency information, to create explicit dependencies
- Valency describes the number and types of modifiers a word can have
- Initially partial trees are constructed (for the verbs)
- Iterative application of the rules eliminates most faulty interpretations
Parsing unification-based grammars

- Chart-parser
  - A separate set of slides treat chart-parsing
  - Chapter 10 of Jurafsky & Martin
- Feature structure unifier
  - Chapter 11 in Jurafsky & Martin

Semantic roles

- Predicate-argument structure
- The surface structure of the argument of verbs are linked to roles in the semantic representation
- Alternation is the possible mappings between grammatical function and semantic roles
- Verb can have selectional restrictions that restrict the type of argument to a suitable type of concept

Chartparsing

Control

Agenda
(List of things to do)

Input
(String)

Chart
(monitor data structures, represents partial results)

Rules
(Grammar and Lexicon)

Semantic roles – Thematic roles

- From general (proto-agent and proto-patient) to very deep/specific (for each type of event/verb)
- Common thematic roles
  - Agent
  - Patient
  - Object
  - Instrument
  - Location
  - Source
  - Goal

Linguistic Knowledge Builder (LKB)

- Development environment
  - Open source (http://wiki.delph-in.net/moin/LkbTop)
  - Typed feature structures
  - Parsing and generation
  - Profiling
  - Resources
    - Large grammars
      - ERG (English)
      - JACY (Japanese)
    - Matrix grammars

Semantic roles – SR lists

- A list of the type of roles the arguments corresponds to
- Verbs can be categorised based on similar semantics and similar SR lists
- Example:
  - Break verbs: bend, fold, shatter, crack
  - Hit verbs: slap, strike, bump, stroke
  - Break: Agent, Instrument, Object
  - Hit: Agent, Instrument, Location
Semantic roles – Direct mapping

- Rules for syntactic realisation of argument structure
- Example
  - Agent in a (non-passive) sentence is in most cases Subject NP
  - Object is in most cases Subject if the verb is intransitive, and Direct object if the verb is transitive

---

Word sense disambiguation

- Most words are ambiguous
- Meaning depends on the context
- Approaches
  - Selectional restrictions (in semantic analysis)
  - Stand alone
    - Supervised ML
    - Bootstrapping ML
    - Unsupervised ML
    - Dictionaries
    - Tagger

---

WSD – Machine learning

- Input
  - Target word
  - Context
- Processing
  - POS tagging
  - Context modification
  - Stemming or morphological analysis
- Feature vector
  - Collocation
  - Co-occurrence

---

WSD – Selectional restrictions

- Use restrictions on thematic roles
- Utilise type hierarchies
- Example:
  I’m looking for a restaurant that serve vegetarian dishes
- Problems
  - Violations
  - Approximations
  - Metaphors

---

WSD – Supervised ML

- Input: vector + correct category/label
- \textit{Naive Bayes classifier}
  - Maximise the probability of a sense given the input vector
  - \[ \text{Sense}^* = \arg\max_s p(\text{vector} | \text{senses}) \]
- Decision list
  - A sequence of tests created due to accuracy
- Problem
  - Requires large amount of data
  - Requires key

---

WSD – Bootstrapping ML

- Small training set used as seeds
- Create an initial classifier through supervised ML
- Generate new data with the classifier
- Iterative development of classifiers with improving coverage and accuracy
WSD – Dictionaries

- Use definitions of senses to find overlaps of the word to be disambiguated and the words in the context
- Problem
  - Short definitions

NLG – Templates

- Congruency
  - [a, the] NOUN-SING is COLOUR
  - an W-NOUN-SING is COLOUR
  - [the]? NOUN-PL are COLOUR
  - en NOUN-N-SING är COLOUR-N-SING
  - ett NOUN-T-SING är COLOUR-T-SING
  - NOUN-PL är COLOUR-A
  - NOUN-DEF är COLOUR-A

Natural language generation

- NLG is the process of creating a written or spoken "text" to achieve a specific communicative goal
- Based on knowledge representation
  - A sentence
  - A paragraph
  - A text
  - An utterance in a dialogue
- NLG is not reversed parsing, has its own problems and solutions

NLG - Choices

- Content – user expertise, context
- Structures (rhetorical) – order, relations
- Choice of word, referring expressions, syntax, lexicalisation, aggregation, passive/active, etc.
- Realisation – morphology, linear order, punctuation, etc.

NLG – Techniques

- Canned text – ready made messages
- Templates – messages with holes that take different values
- Complete generation

NLG - Lexicalisation

- Lexeme
- Synonyms
- Referring expression
  - Mary’s car | her car | Mary’s new car
  - Expression to introduce and refer back
NLG - Aggregation

- Remove redundancy
  - Coordinate
  - Group similar information
  - Ellipsis
  - Embedded segment
  - Subordinate clause

NLG – Traditional reference architecture (Reiter and Dale, 2000)

NLG – RAGS architecture

- Defines high-level data types, data models, … but no specific pipe-line
- Considers 7 low level generation tasks
  - Lexicalisation
  - Aggregation
  - Rhetorical structuring
  - Generating referring expressions
  - Ordering
  - Segmentation
  - Centering/salience tracking