Dialog systems

A few search/architecture problems

Some issues from Verbmobil

SEARCHING IN DIALOG SYSTEMS

Architectures

Several dedicated components (analysis, generation, discourse, translation, …)

In many parts organized in a pipe-line fashion

Some components with complementary/overlapping capabilities

Synchronisation of components may sometimes be time-critical

(processing of some components based on partial or incomplete results)

Some search issues

Orchestrating related components

- **Multiple analysis components (Verbmobil)**
- **Interleaving content specification and realization (WITAS)**

VERBMOBIL

Functionality

Speech-to-speech dialog translation

(German, English, Japanese)

Generation of multi-lingual summaries

Project goals

Speaker independent system for spontaneous dialogs

Three domains and more than 10,000 word forms

Average processing of four times of the input signal duration

Word recognition rate of more than 75% for spontaneous speech

More than 80% correct translations that preserve the speaker's intended effect

90% success rate for dialog tasks end-to-end evaluations

SHALLOW PROCESSING

Architecture

Rule-based system consisting of a cascade of weighted finite-state components

Tokenization, lexico-morphological analysis, part-of-speech filtering, named-entity recognition, sentence-boundary detection, chunk/subclause recognition

Functionality, coverage, and efficiency

30,000 words/sec processing speed

700,000 full-form lexicon entries

(compiled from 120,000 lemmas in a stem lexicon)

Part-of-speech diaambiguation rules compute preferred local readings

96% precision and 85% recall for recognizing proper names, quantities,, etc.

Reference resolution with dynamic lexicon of previously used abbreviations

Highly accurate contextual rules for filtering implausible punctuation signs

DEEP PROCESSING

Grammar

Large-scale HPSG grammar for German, further developed and extended

Hierarchy of 5,069 lexical and phrasal types, 23 rule schemata, 7 special verb movement and 17 domain-specific rules, 38,549 stem lexicon entries

Full sentences, maximal projections, analyses of fragments

Analysis system

Not only full form lexica and string input integrated

Input items can be complex, overlapping, and ambiguous (word graphs)

Dynamic creation of atomic symbols

Flexible interfaces to external systems possible

(morphology, tokenization, named-entity recognition>)

INTEGRATION (I)

Morphology and part-of-speech

Lexicon entries marked as preferred are assigned higher priorities

Finding correct reading early without excluding any reading

Input item not found in lexicon – create default entry with preferred reading

Increases robustness without increasing ambiguity

Named-entity recognition

Classes of shallow processing mapped onto HPSG types (for names)

Fine-tuning required when shallow and deep analyses differ in scope

Part-of-speech sequence used to retrieve pre-built feature structures

Produced by an offline processing with deep analysis and abstraction

More elaborate semantics without boosting the HPSG lexicon

INTEGRATION (II)

Lexical semantics

- **Most words unknown to HPSG are nouns (many of them in GermaNet)**
- **Mapping from fine-grained GermaNet semantic classification onto HPSG**
  - **Learning mapping from GermanNet concepts onto HPSG sorts**
  - **76% mappings correct, 97% within the first three hypotheses**

Stochastic topological parsing

- **Topological restructuring of the corpus analysis (pre-, middle-, post-fields)**
  - **Simple (non-lexicalized) and efficient (context-free part) model**
  - **Nevertheless, high values for accuracy and coverage**
- **Mapping of shallow topological and deep syntactic structures**
- **Changing priorities in the deep parser according to the topological structure**

EXPERIMENTAL RESULTS

Environment extensions

10,000 to 35,000 stems (corresponds to 350,000 full forms)

Test on a Newspaper corpus

	Deep	*Integrated*
# sentences	**20,568**	
avg. sentence length	**16.83**	
avg. lexical ambiguity	**2.38**	**1.98**
avg. # analyses	**16.19**	**18.53**
analysed sentences	**2,569**	**4,546**
lexical coverage	**28.6%**	**71.4%**
overall coverage	**12.5%**	**22.1%**

APPLICATION OF THE INTEGRATED ARCHITECTURE

Information extraction

Pattern-based rules yield good results for recognizing local relationships

Unification-based rules can handle free word order, passive, etc.

"According to X, Y was asked to take over the development sector."

Integrating shallow (S) and deep (D) results via unification-based rules

S: PERSON-IN (1), DIVISION (2), D: take-over (agent (1), theme (2))

Controled language checking

Local agreement rather easy to verify by checking systems

Problems with non-local dependencies, access to grammatical functions

Leads to frequent 'false alarm', since most input is correct

Shallow checker can cheaply identify initial candidates

False alarms eliminated based on richer annotations by the deep parse