Text Generation in the SmartWeb Multimodal Dialogue System

Ralf Engel, Daniel Sonntag

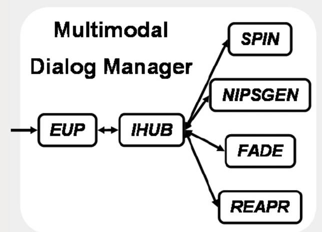
DFKI GmbH, Saarbrücken, Stuhlsatzenhausweg 3, Germany, ralf.engel@dfki.de, daniel.sonntag@dfki.de

Smartweb Requirements

- Multimodal dialogue with question answering functionality.
- Speech is dominant input modality for interaction.
- Multimodal recognition for speech or gestures.
- Result rendering for Semantic Web data content: text, images, videos, graphics, and synthesis of speech.
- Control the message flow in the system.
- Develop a context-aware, mobile, multimodal user interface.
- Use a smartphone as interaction device.

Text Generation

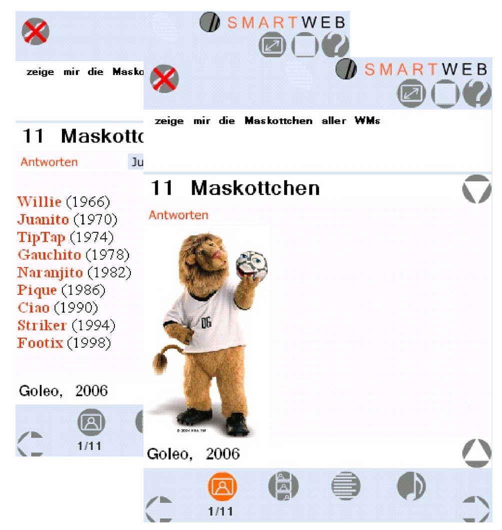
- Generates full utterances for the speech synthesiser and typography-enriched texts and tables presented on the mobile device.
- Generates a semantic paraphrase representing how the system has interpreted the user input in a human readable manner.
- Module called NipsGen is responsible for text generation.
- Input are RDF/S instances based on the SmartWeb ontology (SWintO).
- Interface layer integrates NipsGen into the information hub (IHUB) and translates between SWintO and NipsGen's own internal representation.



- Outlook: (1) Tool that generates automatically the types needed for the TAG derivation tree. (2) Multiple alternative texts which are tested against layout and presentation constraints.

.

"Show me the mascots of the Football World Cup"



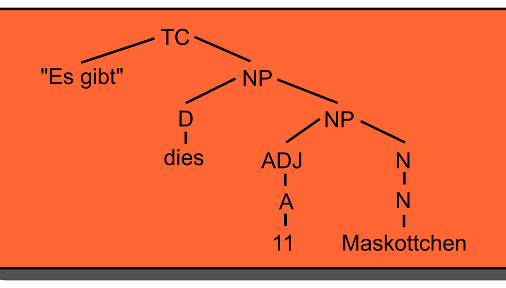
NipsGen module

- Transforms ontology instances into text
- Combines existing components:
  - SPIN parser (essentially a rewriting system for TFS)
  - a TAG framework for German (based on XTAG)
- Allows combination of dynamically generated and canned text

Result(target: synthesis,
number: 11,
type: Mascot(...))

TC(c: ["Es gibt" ,
NP(det: dies, num: pl,
adj: AdjP(lex: 11),
lex: Veranstaltung)])

TC(c: ["Es gibt",
aN(I_11: Maskottchen,
a_0: bDnx(I_11: dies),
a_1: bADJnx(I_111: 11),
fvp: Fvp(num: pl))])



Es gibt diese 11 Maskottchen.

Step 1: Construction of the intermediate syntactic representation

SPIN rules (exemplary)

Result(target: synthesis,
type: $T)
-> TC(c: ["Es gibt",
NP(o: $T)])

NP(o: Mascot())
-> NP(lex: Maskottchen)

Step 2: Transformation to the extended TAG derivation tree

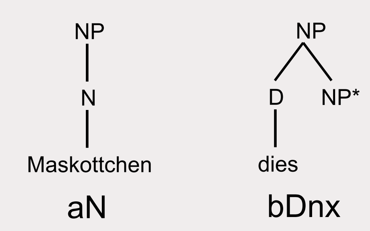
SPIN rule (exemplary)

NP(lex: $L,
%det: $D,
%adjP: $AdjP)
-> aN(I_11: $Lex,
a_0: bDnx(
I_11: $Det),
a_1: $AdjP)

Step 3: Construction of the derived TAG tree

Elementary trees of TAG grammar are combined and features are propagated

Elementary trees (exemplary)



Step 4: Morphosyntactic adaptations of the lexical leafs and linearization

- Full form lexicon
- Support for HTML tags with features *formatBegin* and *formatEnd*

This work was funded by the German Federal Ministry of Education and Research (BMBF) in the framework of the SmartWeb project under Grant 01 IM D0 1A. The responsibility for the contents lies with the authors.