Natural Language Generation for Spoken Dialogue Systems

Grad. Seminar on Spoken Dialogues

Antoine Raux 11/20/2003
Outlook

- Introduction
Natural Language Generation

- Formal representation of information
- Applications:
  - Machine translation
  - Text generation (summarization, expert systems…)
  - Dialogue systems
  - …?
Where does NLG start and where does it end? (the slippery slope)

Dialogue Manager \( \rightarrow \) Content Planner \( \rightarrow \) Surface Realizer \( \rightarrow \) Speech Synthesizer

Decides the next dialogue act
And the main information to convey
Where does NLG start and where does it end? (the slippery slope)

Dialogue Manager → Content Planner → Surface Realizer → Speech Synthesizer

Decides what pieces of information to include in the utterance
Where does NLG start and where does it end? (the slippery slope)

- Dialogue Manager
- Content Planner
- Surface Realizer
- Speech Synthesizer

Constructs a valid utterance expressing the selected concepts and their relationship.
Where does NLG start and where does it end? (the slippery slope)

Produces the spoken utterance, with appropriate prosody
Where does NLG start and where does it end? (the slippery slope)

But, in reactive systems, generation might start as early as recognition: decision to stop speaking on barge-in, change the planned utterance according to the user input, etc…
Machine Translation Approach

- Starts from a “bag of concepts” and builds all possible syntactically valid sentences containing the concepts.
- Selects one (cf N-Gram paper).
- Eventually includes semantic constraints.
- Order of concepts not relevant (to deal with syntactic differences across languages).
MT Model for SDS?

- Adaptation of the model exists to account for *incrementality* in SDS:
  - Starts building an utterance with an incomplete set of concepts
  - Complete the utterance as new concepts are added
- No structure to the information
Text Generation

- Centered on information structure (discourse planning, reference…)
- Monologue: all the information must be provided in one single text
- No interaction: in SDS feedback from the user guides generation
- In SDS, utterances are short, with little information
- Conclusion: not appropriate for SDS…
Template-based Generation

- Most common in SDS: hand written templates for fixed sets of concepts
- Rigid: produces only a limited set of utterances
- Not scalable: hard to write/maintain large sets of utterances
- Fits well with limited domain speech synthesis
Planning Dialogue Contributions with New Information

- Very general definition of generation (even includes the DM)
- Based on observation of human-human dialogues
- Computational theory of generation in dialogue
The Three I’s of Generation in Dialogue

- **Incrementality:**
  - Old information vs new information

- **Immediacy:**
  - Speakers closely monitor their partner (backchannel, barge-in, etc...)

- **Interactivity:**
  - Some turns only play a role at the metalevel of interaction management (e.g. ask for a confirmation by repeating an already known piece of information)
Information Structure of Utterances

- Information Unit: a phrase with one or more accented words and a phrase accent at the end
- Central Concept: concept on which the participants are focusing:
  - goal or subgoal for task-based dialogues
  - current topic for descriptions
- NewInfo: piece of information newly presented by an utterance:
  - Can become CC in future turns
Generation as “Wrapping” of the NewInfo

- The DM provides the surface realizer with a communicative intention (dialogue act) and the NewInfo wrapped in further information:
  - Wrapping 1: prevent ambiguity by providing enough details
  - Wrapping 2: make sure that uncertain concepts are mutually known
  - Wrapping 3: decompose complex goals into subgoals
Conclusion/Discussion

- General computational theory of dialogue
- Handles both task-based dialogues and descriptions based on topic shifts
- Decomposition Task Manager/Dialogue Manager

- Problems: no implementation…
The Practical Value of N-Grams in Generation

- Combine a rule-based surface realizer with a statistical selector
- Study the contribution of statistical models to surface realization
Two-Tier Generation Module

Meaning

Symbolic Generator

Word lattice of possible renderings

Statistical Extractor

English string

lexicon
grammar
corpus
Symbolic Generator

- Gets feature structures as input with semantic relations (agent/patient, source/destination...)
- Produces a lattice of possible English word sequences
Symbolic Generator

- **Lexicon**: links a semantic concept to a root
- **Morphological KB**: rules and exceptions for plurals, conjugation...
- **Grammar rules mapping between**:
  - Semantic relations (input)
  - Deep syntactic relations (oblique, adjunct…)
  - Shallow syntactic relations (subject, object…)
- **Does not try to solve ambiguity**: produces all possible outputs
Statistical Extractor

- Bigram LM trained on 2 years of The Wall Street Journal
- From all the possible outputs of the symbolic generator, selects the one with the highest likelihood according to the LM
- Backs off to unigram when no bigrams
Conclusion/Discussion

- Simplicity: do not need very detailed grammar rules
- Robustness: LM works on any English sentence
- Generality: should work for any topic
- Weaknesses: bigrams only capture very local relations, sparseness issue with larger n...
Stochastic Language Generation for Spoken Dialogue Systems

- Specifically for dialogue systems
- Based on statistical methods
- Handles both content planning and surface realization
- Domain specific generation
Labeled Corpora

- In the context of Communicator (air travel reservation)
- Two corpora of human-human transcribed dialogues (travel agent/client)
- Each utterance classified by type: query_arrive_city, inform_flight, inform_price
- Each concept labeled: airline, arrive_city, hotel_price...
- Tagging done manually first and then semi-automatically
Content Planning

- Train models from labeled data to predict:
  - Number of attributes (concepts) in the utterance
  - Sequence of attributes
- Very simple models:
  - most frequent number of attributes for each class
  - Presence of each attribute only depends on the user’s previous utterance
- Deals with data sparseness
Surface Realization

- Generates random sequences of words according to the class LM (5-gram)
- Repeats until got an utterance with all attributes predicted by content planner (or timeout)
- Replaces word class names by actual words (e.g. depart_city → New York)
Evaluation

- No significant difference in user satisfaction between an old vs new approach and this approach for content planning (trend: old vs new)
- No significant difference in human preference judgment for surface realization between templates and this approach (trend: this approach)
Conclusion/Discussion

- Empirical approach to content planning (original) but too many simplifications? How to deal with data sparseness?
- Domain-specific statistical LM approach to surface realization: more convincing. Allows variations in the output
- Nice attempt to evaluation but not enough data?
Integrating Language Generation with Speech Synthesis in a Concept to Speech System

- Lower end of the process: link with synthesis
- Integrates prosody into generation
- System description
NLG Component

- Content planner, micro planner (sentence planning) and lexical chooser not detailed
- Produces a semantic structure (~ feature structure)
Speech Integrating Markup Language

- Keeps semantic/syntactic information with the utterance
- Converts into TTS system specific features