Natural Language Generation
(Not Only) in Dialogue Systems

Ondřej Dušek

Institute of Formal and Applied Linguistics
Charles University in Prague

May 22, 2013
Introduction

Objective of NLG

Given (whatever) input and a communication goal, create a natural language string that is well-formed and human-like.

- Desired properties: variation, simplicity, trainability (?)

Usage

- Spoken dialogue systems
- Machine translation
- Short texts: Personalized letters, weather reports . . .
- Summarization
- Question answering in knowledge bases
Standard (Textbook) NLG Pipeline

[Input]

↓ Content/Text Planning ("what to say")

▶ Content selection, basic ordering

[Text plan]

↓ Sentence Planning/Realization ("how to say it")

↓ Microplanning: aggregation, lexical choice, referring...

[Sentence Plan(s)]

↓ Surface realization: linearization according to grammar

[Text]
Content Planning

Possible NLG Inputs

- Content plan (meaning, communication goal)
- Knowledge base (e.g. list of matching entries in database, weather report numbers etc.)
- User model (constraints, e.g. user wants short answers)
- Dialogue history (referring expressions, repetition)

Tasks of content planning

- Content selection according to communication goals
- Basic structuring (ordering)
Tasks of surface realization

Sentence planning (micro-planning)

- Word and syntax selection (e.g. choose templates)
- Dividing content into sentences
- Aggregation (merging simple sentences)
- Lexicalization
- Referring expressions

Surface realizer (proper)

- Creating linear text from (typically) structured input
- Ensuring syntactic correctness
Real NLG Systems

Few systems implement the whole pipeline

- Systems focused on content planning with trivial surface realization
- Surface-realization-only systems
- Word-order-only systems
- Input/intermediate data representation is incompatible

Possible approaches

- Template-based
- Grammar-based
- Statistical
- ... or a mix thereof
Content Planning

Workflow

1. Decide on information to be said
2. Construct discourse plan
3. “Chunk” into units of discourse

- Input: communication goal ("explain", "describe", "relate")
- Output: discourse (tree) structure – content plan tree

Possible approaches

- Schemas (observations about common text structures)
- Planning, rhetorical structure theory
- Machine learning
Example: WeatherReporter

- Generation of weather reports from raw data
- Rule-based (textbook example)
Example: SPoT

- Spoken Dialogue System in the flight information domain
- Rule-based sentence plan generator (clause combining operations)
- Statistical re-ranker (RankBoost) trained on hand-annotated sentence plan

```plaintext
implicit-confirm(orig-city:NEWARK)
implicit-confirm(dest-city:DALLAS)
implicit-confirm(month:9)
implicit-confirm(day-number:1)
request(depart-time)
```
Example: MATCH

- NYC multimodal information system
- Presentation strategy based on user model (users answer initial questions)
Example: RL-NLG

- Tested on MATCH corpus
- Reinforcement learning of presentation strategy
- Communicative Goal: Dialogue Act + desired user reaction
- Plan lower-level NLG actions to achieve goal
Surface Realization

Workflow

1. *Microplanning*: Select appropriate phrases and words
   - Content plan to text
   - Uses lexicons, grammars, ontologies...

Methods

- Canned text / template filling
- Rule- / grammar based
- Statistical / hybrid
Handcrafted realizers

Template-based

- Most common, also in commercial NLG systems
- Simple, straightforward, reliable (custom-tailored for domain)
- Lack generality and variation, difficult to maintain
- Enhancements for more complex utterances: rules

Grammar-based

- Hand-written grammars / rules
- Various formalisms
Example: Templates

- Just filling variables into slots
- Possibly a few enhancements, e.g. articles

```plaintext
inform(pricerange="{pricerange}"):
'It is in the {pricerange} price range.'

affirm() & inform(task="find") & inform(pricerange="{pricerange}"):
'Ok, you are looking for something in the' + ' {pricerange} price range.'

affirm() & inform(area="{area}"):
'Ok, you want something in the {area} area.'

affirm() & inform(food="{food}")
& inform(pricerange="{pricerange}"):
'Ok, you want something with the {food} food' + ' in the {pricerange} price range.'

inform(food="None"):
'I do not have any information' + ' about the type of food.'
```

Facebook templates

ALEX English templates
Examples: FUF/SURGE, KPML

**KPML**
- General purpose, multi-lingual
- Systemic Functional Grammar

**FUF/SURGE**
- General purpose
- Functional Unification Grammar

(EXAMPLE
   :NAME EX-SET-1
   :TARGETFORM "It is raining cats and dogs."
   :LOGICALFORM
     (A / AMBIENT-PROCESS :LEX RAIN
       :TENSE PRESENT-CONTINUOUS :ACTEE
         (C / OBJECT :LEX CATS-AND-DOGS :NUMBER MASS)))

Input Specification ($I_1$):

Output Sentence ($S_1$): “She hands the draft to the editor”
Example: OpenCCG

- General purpose, multi-lingual
- Combinatory Categorial Grammar
- Used in several projects
- With statistical enhancements

\[
\begin{align*}
(\Rightarrow) & \quad X/Y \quad Y \quad \Rightarrow \quad X \\
(\Leftarrow) & \quad Y \quad X\setminus Y \quad \Rightarrow \quad X \\
(\Rightarrow B) & \quad X/Y \quad Y/Z \quad \Rightarrow \quad X/Z \\
(\Leftarrow B) & \quad Y\setminus Z \quad X\setminus Y \quad \Rightarrow \quad X\setminus Z \\
(\Rightarrow T) & \quad X \quad \Rightarrow \quad Y/(Y\setminus X) \\
(\Leftarrow T) & \quad X \quad \Rightarrow \quad Y\setminus(Y\setminus X)
\end{align*}
\]

\[
\text{man} \vdash n \\
\text{that} \vdash (n\setminus n)/(s_{\text{form}=\text{fin}}/\text{np}) \\
\text{Bob} \vdash \text{np} \\
\text{saw} \vdash (s_{\text{tense}=\text{past, sform}=\text{fin}}\setminus \text{np})/\text{np}
\]
Example: SimpleNLG

- General purpose
- English, adapted to several other languages
- Java implementation (procedural)

```java
Lexicon lexicon = new XMLLexicon("my-lexicon.xml");
NLGFactory nlgFactory = new NLGFactory(lexicon);
Realiser realiser = new Realiser(lexicon);

SPhraseSpec p = nlgFactory.createClause();
p.setSubject("Mary");
p.setVerb("chase");
p.setObject("the monkey");
p.setFeature(Feature.TENSE, Tense.PAST);

String output = realiser.realiseSentence(p);
System.out.println(output);

>>> Mary chased the monkey.
```
Trainable Surface Realizers: Overgenerate and Rank

- Require a hand-crafted realizer, e.g. CCG realizer
- Input underspecified $\rightarrow$ more outputs possible
- Overgenerate
- Then use a statistical re-ranker
- Ranking according to:
  - **NITROGEN, HALOGEN**: $n$-gram models
  - **FERGUS**: Tree models (XTAG grammar)
  - **Nakatsu and White**: Predicted Text-To-Speech quality
  - **CRAG**: Personality traits (extraversion, agreeableness...) + alignment (repeating words uttered by dialogue counterpart)
- Provides variance, but at a greater computational cost
Trainable Surface Realizers: Parameter Optimization

- Still require a hand-crafted realizer
- Train hand-crafted realizer parameters
- No overgeneration
- Realizer needs to be “flexible”

Examples

- *Paiva and Evans*: linguistic features annotated in corpus generated with many parameter settings, correlation analysis
- *PERSONAGE-PE*: personality traits connected to linguistic features via machine learning
Fully Statistical Surface Realizers

- Few, rather limited, based on supervised learning

Phrase-based

- Hierarchical: semantic stacks / records \(\downarrow\) fields \(\downarrow\) templates
- Limited domain
- *Mairesse et al.*: Bayesian networks
- *Angeli et al.*: log-linear model

Syntax-based

- *Bohnet et al.*: general realizer based on SVMs
- Deep syntax/semantics \(\rightarrow\) surface syntax \(\rightarrow\) linearization \(\rightarrow\) morphologization
Natural Language Generation at ÚFAL

- Procedural, for Czech (and partially Russian)
  - *Ptáček and Žabokrtský*: Generating from PDT (t-trees with functors)
  - *TectoMT*: Generating from t-trees with formemes
  - Word form selection: *Hajič*’s morphological dictionary

- ReverseNumberNounDependency
- InitMorphcat
- FixPossessiveAdjs
- MarkSubject
- Impose{PronZ, RelPron, Subjpred, Attr, Compl}Agr
- DropSubjPersProns
- Add{Prepos, Subconjs, ReflexParticles}
- AddAuxVerb{CompoundPassive, Modal, CompoundFuture, Conditional, Past}
- AddClausalExpletivePronouns
- ResolveVerbs
- ProjectClauseNumber
- AddParentheses
- Add*Punct
- ChooseMlemmaForPersPron
- GenerateWordforms
- DeleteSuperfluousAuxCP
- MoveCliticsToWackernagel
- VocalizePrepos
- CapitalizeSentStart
- ConcatenateTokens
Czech NLG for ÚFAL Dialogue Systems

- Partial tecto-templates
  - Simpler specification (improvements due)
- Using statistical word form generator
  - Levenshtein distance edit-scripts
  - Logistic regression model


- do: doing  >0-ing
- llegar: llegó  >2-ó
- Mann: Männer  >0-er,3:1-ä
- jenž: jež  >2:1-
- mantener: mantindran  >0-an,2:1-d,4:1-i
- sparen: gespart  >2-t,<ge
- vědět: nevíme  >4-íme,<ne
- be: is  *is
References

Angeli

Bohnet
Bohnet, B. et al. 2010. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. *COLING*

CRAG
Isard, A. et al. 2006. Individuality and alignment in generated dialogues. *INLG*

FERGUS

FUF/SURGE

Hajič

HALOGEN
Langkilde-Geary, I. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. *INLG*

KPML

OpenCCG

Mairesse
Mairesse, F. et al. 2010. Phrase-based statistical language generation using graphical models and active learning. *ACL*

MATCH
References

Nakatsu&White  Nakatsu, C. and White, M. 2006. Learning to say it well: reranking realizations by predicted synthesis quality. COLING-ACL


Rl-Nlg   Rieser, V. and Lemon, O. 2010. Natural language generation as planning under uncertainty for spoken dialogue systems. EMNLP


SPoT   Walker, M. et al. 2001. SPoT: A trainable sentence planner. NAACL

TectoMt   Žabokrtský, Z. et al. 2008. TectoMT: highly modular MT system with tectogrammatics used as transfer layer. WMT


Further Links
C. DiMarco's slides: https://cs.uwaterloo.ca/~jchampai/CohenClass.en.pdf
J. Moore's NLG course: http://www.inf.ed.ac.uk/teaching/courses/nlg/
NLG Systems Wiki: http://www.nlg-wiki.org