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

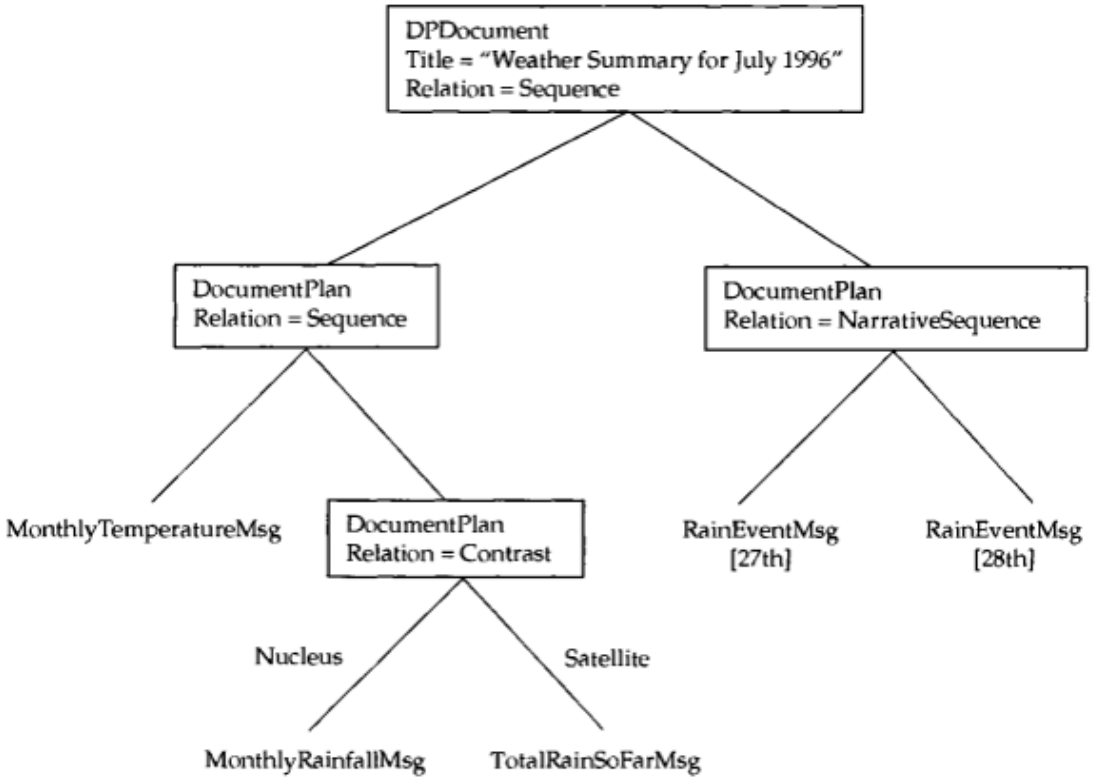
- Input: communication goal ("explain", "describe", "relate")
- Output: discourse (tree) structure – content plan tree

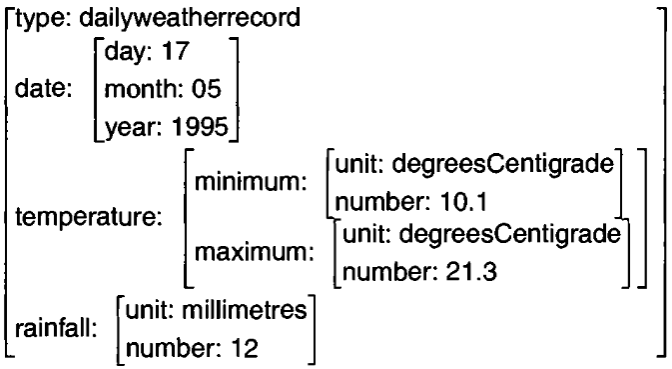
Possible approaches

- 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)

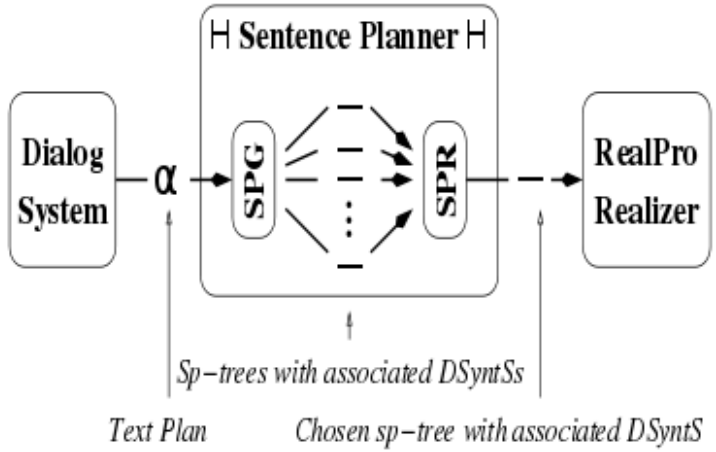




Example: SPoT

- 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

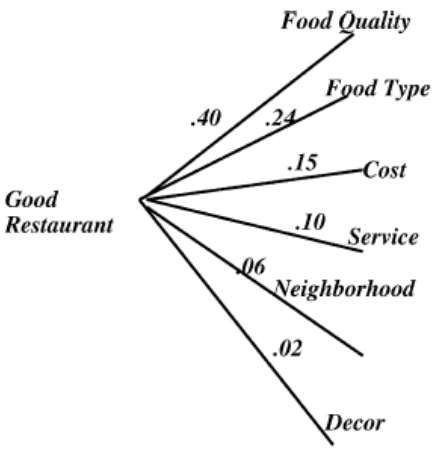
```
implicit-confirm(orig-city:NEWARK)
implicit-confirm(dest-city:DALLAS)
implicit-confirm(month:9)
implicit-confirm(day-number:1)
request(depart-time)
```



Alt	Realization	H	RB
0	What time would you like to travel on September the 1st to Dallas from Newark?	5	.85
5	Leaving on September the 1st. What time would you like to travel from Newark to Dallas?	4.5	.82
8	Leaving in September. Leaving on the 1st. What time would you, traveling from Newark to Dallas, like to leave?	2	.39

Example: MATCH

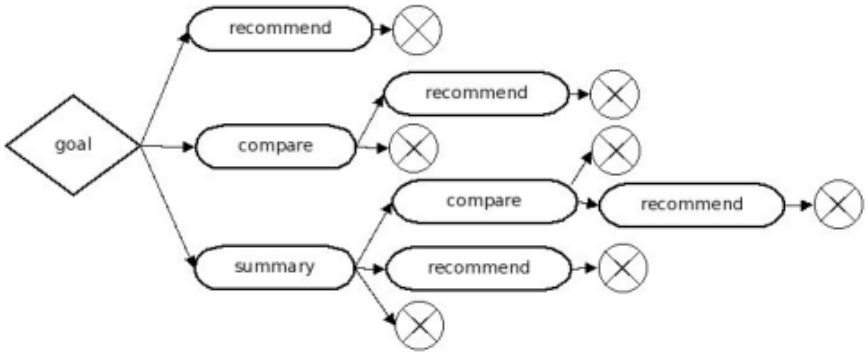
- NYC multimodal information system
- Presentation strategy based on user model (users answer initial questions)

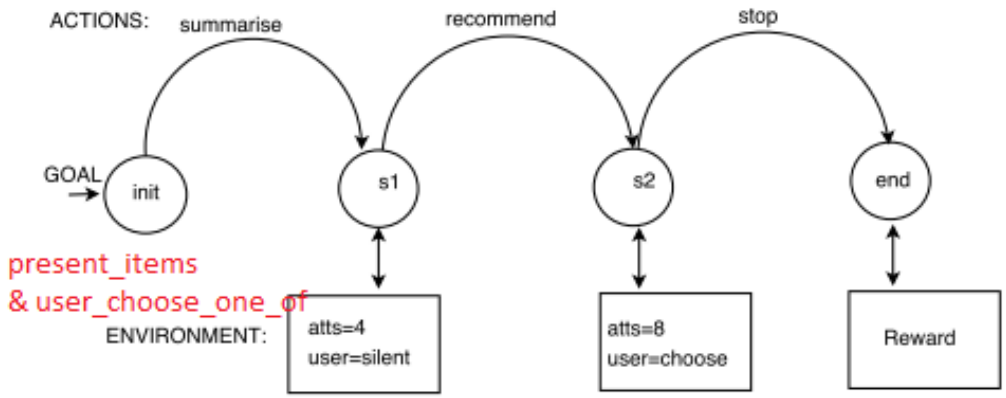


Show me Italian restaurants in the West Village.

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
2. *Realization*: Produce grammatically correct output.

- 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
- Enhacements 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

```
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.'
```

ALEX English templates

Facebook 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):

$$\begin{bmatrix} cat & clause \\ process & \begin{bmatrix} type & composite \\ relation & possessive \\ lex & \text{“}hand\text{”} \end{bmatrix} \\ partic & \begin{bmatrix} agent & \begin{bmatrix} cat & pers\_pro \\ gender & feminine \end{bmatrix} \\ affected & \boxed{1}\begin{bmatrix} cat & np \\ lex & \text{“}editor\text{”} \end{bmatrix} \\ possessor & \boxed{1} \\ possessed & \begin{bmatrix} cat & np \\ lex & \text{“}draft\text{”} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

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

$$
\begin{array}{lllll}
(>) & X/Y & Y & \Rightarrow & X \\
(<) & Y & X\backslash Y & \Rightarrow & X \\
(>\mathbf{B}) & X/Y & Y/Z & \Rightarrow & X/Z \\
(<\mathbf{B}) & Y\backslash Z & X\backslash Y & \Rightarrow & X\backslash Z \\
(>\mathbf{T}) & X & & \Rightarrow & Y/(Y\backslash X) \\
(<\mathbf{T}) & X & & \Rightarrow & Y\backslash(Y/X)
\end{array}
$$

$$
\begin{aligned}
&\mathit{man} \vdash \mathsf{n} \\
&\mathit{that} \vdash (\mathsf{n}\backslash\mathsf{n})/(\mathsf{s}_{vform=fin}/\mathsf{np}) \\
&\mathit{Bob} \vdash \mathsf{np} \\
&\mathit{saw} \vdash (\mathsf{s}_{tense=past,vform=fin}\backslash\mathsf{np})/\mathsf{np}
\end{aligned}
$$

```
be [tense=pres info=rh id=n1]
<Arg> flight [num=sg det=the info=th id=f2]
      <HasProp> cheapest [kon=+ id=n2]
<Prop> has-rel [id=n3]
       <Of> f2
       <Airline> Ryanair [kon=+ id=n4]
```

$$
\dfrac{\dfrac{\mathit{man}}{\mathsf{n}} \quad \dfrac{\mathit{that}}{(\mathsf{n}\backslash\mathsf{n})/(\mathsf{s}/\mathsf{np})} \quad \dfrac{\dfrac{\dfrac{\mathit{Bob}}{\mathsf{np}}}{\mathsf{s}/(\mathsf{s}\backslash\mathsf{np})}>\mathbf{T} \quad \dfrac{\mathit{saw}}{(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}}}{\mathsf{s}/\mathsf{np}}>\mathbf{B}}{\mathsf{n}\backslash\mathsf{n}}>}{\mathsf{n}}<
$$

Example: SimpleNLG

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

```
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-crafed 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 ↘ fields ↘ templates
- Limited domain
- *Mairesse et al.*: Bayesian networks
- *Angeli et al.*: log-linear model

Syntax-based

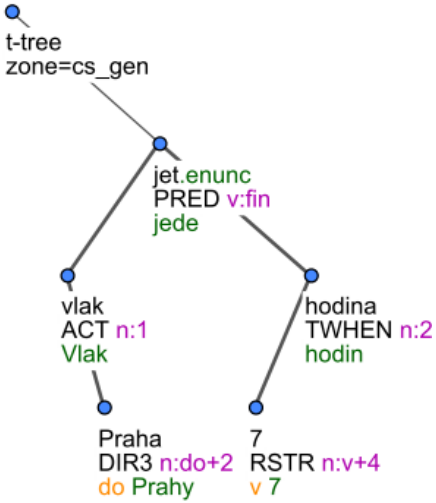
- *Bohnet et al.*: general realizer based on SVMs
- Deep syntax/semantics → surface syntax → linearization → morphologization

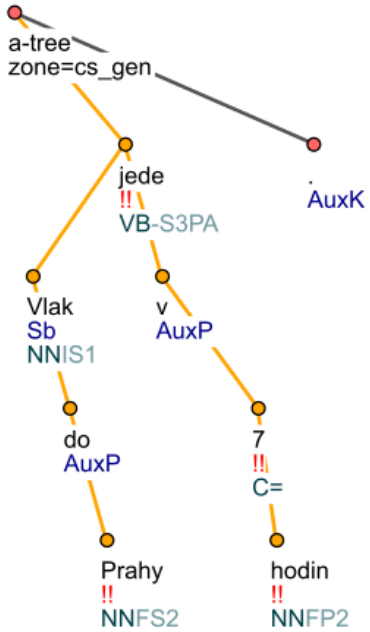
Natural Language Generation at ÚFAL

- 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

```
Vlak [Praha|n:do+2|gender:fem] jede
v [[7|adj:attr] hodina|n:4|gender:fem].
```

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

References

Angeli — Angeli, G. et al. 2010. A Simple Domain-Independent Probabilistic Approach to Generation. *EMNLP*

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 — Bangalore, S. and Rambow, O. 2000. Exploiting a probabilistic hierarchical model for generation. *COLING*

FUF/SURGE — Elhadad, M. and Robin, J. 1996. An overview of SURGE: A reusable comprehensive syntactic realization component.
`http://www.cs.bgu.ac.il/surge/`

Hajič — Hajič, J. 2004. *Disambiguation of Rich Inflection – Computational Morphology of Czech*. Karolinum

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

KPML — Bateman, J. A. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*
`http://purl.org/net/kpml`

OpenCCG — White, M. and Baldrige, J. 2003. Adapting Chart Realization to CCG. *ENLG*
Moore, J. et al. 2004. Generating Tailored, Comparative Descriptions in Spoken Dialogue. *FLAIRS*
`http://openccg.sourceforge.net/`

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

MATCH — Walker, M. et al. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*

References

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

NITROGEN Langkilde, I. and Knight, K. 1998. Generation that exploits corpus-based statistical knowledge. *ACL-COLING*

Paiva&Evans Paiva, D. S. and Evans, R. 2005. Empirically-based control of natural language generation. *ACL*

PERSONAGE-PE Mairesse, F. and Walker, M. 2008. Trainable generation of big-five personality styles through data-driven parameter estimation. *ACL*

Ptáček&Žabokrtský Ptáček, J. and Žabokrtský, Z. 2006. Synthesis of Czech Sentences from Tectogrammatical Trees. *TSD*

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

SimpleNLG Gatt, A. and Reiter, E. 2009. SimpleNLG: A realisation engine for practical applications. *ENLG*

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*

Textbook Reiter, E. and Dale, R. 2000. *Building natural language generation systems*. Cambridge Univ. Press

Further Links

C. DiMarco's slides: `https://cs.uwaterloo.ca/~jchampai/CohenClass.en.pdf`
F. Mairesse's slides: `http://people.csail.mit.edu/francois/research/papers/ART-NLG.pdf`
J. Moore's NLG course: `http://www.inf.ed.ac.uk/teaching/courses/nlg/`
NLG Systems Wiki: `http://www.nlg-wiki.org`
Wikipedia: `http://en.wikipedia.org/wiki/Natural_language_generation`