Inducing Document Plans for Concept-to-Text Generation

Ioannis Konstas, Mirella Lapata

Institute for Language, Cognition and Computation
University of Edinburgh

EMNLP 2013, Seattle
20 October 2013
Concept-to-text generation refers to the task of automatically producing textual output from nonlinguistic input (Reiter and Dale, 2000).
Concept-to-text generation refers to the task of automatically producing textual output from nonlinguistic input (Reiter and Dale, 2000).

Click start, point to settings, and then click control panel. Double-click users and passwords. On the advanced tab, click advanced.
Traditional NLG Pipeline

Input Data → Content Planning → Sentence Planning → Surface Realisation → Text

Communicative Goal
Related Work

Traditional NLG Pipeline

Input Data

Content Planning

- Content Selection
- Document Planning

Sentence Planning

Surface Realisation

Text
Traditional NLG Pipeline

- Input Data
- Content Planning
  - Content Selection
  - Document Planning
- Sentence Planning
- Surface Realisation
- Text

Barzilay and Lapata (2005)
Liang et al. (2009)
Related Work

Traditional NLG Pipeline

Input Data → Content Planning → Content Selection → Document Planning → Sentence Planning → Surface Realisation → Text

Hovy (1993)
Scott and de Souza (1990)
Duboue and Mckeown (2002)
Related Work

Traditional NLG Pipeline

Input Data → Content Planning → Content Selection → Document Planning → Sentence Planning → Surface Realisation → Text

Communicative Goal

Stent et al. (2004)
Related Work

Traditional NLG Pipeline

- Input Data
- Communicative Goal
- Content Planning
  - Content Selection
  - Document Planning
- Sentence Planning
- Surface Realisation
- Text

Wong and Mooney (2007)
Lu and Ng (2011)
Traditional NLG Pipeline

- **Input Data**
- **Communicative Goal**
- **Content Planning**
  - **Content Selection**
  - **Document Planning**
- **Sentence Planning**
- **Surface Realisation**
- **Text**

Related Work:
- Kim and Mooney (2010) Pipeline approach
- Angeli et al. (2010) Unified approach
- Konstas and Lapata (2012) Joint approach

This work: Joint approach

Concept-to-Text Generation
Related Work

Traditional NLG Pipeline

Input Data → Content Planning → Content Selection → Document Planning → Sentence Planning → Surface Realisation → Text

Communicative Goal

Kim and Mooney (2010)
Pipeline approach

Angeli et al. (2010)
Unified approach

Konstas, Lapata (ILCC)
Concept-to-Text Generation

20 October 2013
Related Work

Traditional NLG Pipeline

- **Input Data**
- **Communicative Goal**
- **Content Planning**
  - **Content Selection**
  - **Document Planning**
- **Sentence Planning**
- **Surface Realisation**
- **Text**

**Kim and Mooney (2010)**
Pipeline approach

**Angeli et al. (2010)**
Unified approach

**Konstas and Lapata (2012)**
Joint approach
Related Work

Traditional NLG Pipeline

Input Data

Content Planning

- Content Selection
- Document Planning

Sentence Planning

Surface Realisation

Text

Communicative Goal

Kim and Mooney (2010)
Pipeline approach

Angeli et al. (2010)
Unified approach

Konstas and Lapata (2012)
Joint approach

This work
Joint approach
**Input**

- **Input:** database records \( d \)
- **Output:** words \( w \) corresponding to some records of \( d \)
- Each record \( r \in d \) has a type \( r.t \) and fields \( f \)
- Fields have values \( f.v \) and types \( f.t \) (integer, categorical, string)

<table>
<thead>
<tr>
<th>Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd</strong></td>
</tr>
<tr>
<td>left-click</td>
</tr>
</tbody>
</table>

*Konstas, Lapata (ILCC) Concept-to-Text Generation 20 October 2013*
Input

- Input: database records $\mathbf{d}$
- Output: words $\mathbf{w}$ corresponding to some records of $\mathbf{d}$
- Each record $r \in \mathbf{d}$ has a type $r.t$ and fields $f$
- Fields have values $f.v$ and types $f.t$ (integer, categorical, string)

<table>
<thead>
<tr>
<th>Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd</strong></td>
</tr>
<tr>
<td>left-click</td>
</tr>
</tbody>
</table>

click start,
Input: database records $\mathbf{d}$

Output: words $\mathbf{w}$ corresponding to some records of $\mathbf{d}$

Each record $r \in \mathbf{d}$ has a type $r.t$ and fields $f$

Fields have values $f.v$ and types $f.t$ (integer, categorical, string)
Input: database records $d$
Output: words $w$ corresponding to some records of $d$
Each record $r \in d$ has a type $r.t$ and fields $f$
Fields have values $f.v$ and types $f.t$ (integer, categorical, string)

<table>
<thead>
<tr>
<th>Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd</strong></td>
</tr>
<tr>
<td>left-click</td>
</tr>
</tbody>
</table>

```
click start,
```
Input: database records \( \mathbf{d} \)

Output: words \( \mathbf{w} \) corresponding to some records of \( \mathbf{d} \)

Each record \( r \in \mathbf{d} \) has a type \( r.t \) and fields \( f \)

Fields have values \( f.v \) and types \( f.t \) (integer, categorical, string)

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>left-click</td>
<td>start</td>
<td>button</td>
</tr>
</tbody>
</table>

"click start,\"
Click start, point to settings, and then click control panel. Double-click users and passwords. On the advanced tab, click advanced.
Click start, point to settings, and then click control panel. Double-click users and passwords. On the advanced tab, click advanced.
Click start, point to settings, and then click control panel. Double-click users and passwords. On the advanced tab, click advanced.
Click start, point to settings, and then click control panel.

Double-click users and passwords.

On the advanced tab, click advanced.
Click start, point to settings, and then click control panel. 

**Double-click users and passwords.**

On the advanced tab, click advanced.
Key Idea: Grammar-based document plans
Key Idea: Grammar-based document plans

- Extend the model of Konstas and Lapata (2012)
Key Idea: Grammar-based document plans

- Extend the model of Konstas and Lapata (2012)
- Re-use the generation model based on a PCFG grammar of input
Key Idea: Grammar-based document plans

- Extend the model of Konstas and Lapata (2012)
- Re-use the generation model based on a PCFG grammar of input
- Replace existing locally coherent Content Selection model and incorporate global Document Planning (explore two solutions):
Key Idea: Grammar-based document plans

- Extend the model of Konstas and Lapata (2012)
- Re-use the generation model based on a PCFG grammar of input
- Replace existing locally coherent Content Selection model and incorporate global Document Planning (explore two solutions):
Key Idea: Grammar-based document plans

- Extend the model of Konstas and Lapata (2012)
- Re-use the generation model based on a PCFG grammar of input
- Replace existing locally coherent **Content Selection** model and incorporate global **Document Planning** (explore two solutions):

Patterns of record sequences *within* a sentence and *among* sentences

Rhetorical Structure Theory (Mann and Thompson, 1988) inspired plans
Grammar
Grammar

1. $S \rightarrow R\left(start\right)$
Grammar

1. $S \rightarrow R(start)$
2. $R(r_i.t) \rightarrow FS(r_j, start)R(r_j.t) \mid FS(r_j, start)$

$R(desktop_1.t) \rightarrow FS(start_1, start)R(start_1.t)$
Grammar

1. $S \rightarrow R(start)$

2. $R(r_i.t) \rightarrow FS(r_j, start)R(r_j.t) \mid FS(r_j, start)$

$R(desktop_1.t) \rightarrow FS(start_1, start)R(start_1.t)$
Grammar

1. \[ S \rightarrow R(start) \]
2. \[ R(r_i.t) \rightarrow FS(r_j, start)R(r_j.t) \mid FS(r_j, start) \]
3. \[ FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j) \]

\[ FS(desktop_1, cmd) \rightarrow F(desktop_1, name)FS(desktop_1, name) \]
Grammar

1. \( S \rightarrow R(start) \)
2. \( R(r_i.t) \rightarrow FS(r_j, start)R(r_j.t) \mid FS(r_j, start) \)
3. \( FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j) \)
4. \( F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f) \)

\[ F(desktop_1, cmd) \rightarrow W(desktop_1, cmd)F(desktop_1, cmd) \]
Grammar

1. $S \rightarrow R(start)$
2. $R(r_i.t) \rightarrow FS(r_j, start)R(r_j.t) \mid FS(r_j, start)$
3. $FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j)$
4. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f)$
5. $W(r, r.f) \rightarrow \alpha \mid g(f.v)$

$W(desktop_1, cmd) \rightarrow \text{click} \ [\text{cmd.v} = \text{‘left-click’}]$
Grammar

1. $S \rightarrow R(start)$
2. $R(r_i.t) \rightarrow FS(r_j, start)R(r_j.t) \mid FS(r_j, start)$
3. $FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j)$
4. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f)$
5. $W(r, r.f) \rightarrow \alpha \mid g(f.v)$

**EM Training**: dynamic program similar to the inside-outside algorithm
Key idea: Grammar on sequences of record types ($G_{RSE}$)
Planning with Record Sequences

Key idea: Grammar on sequences of record types ($G_{RSE}$)

1. Click start, point to settings, and then click control panel. || Double-click users and passwords. || On the advanced tab, click advanced. ||

Split a document into sentences, each terminated by a full-stop.
Planning with Record Sequences

Key idea: Grammar on sequences of record types ($G_{RSE}$)

1. Click start, point to settings, and then click control panel. || Double-click users and passwords. || On the advanced tab, click advanced. ||

Split a document into sentences, each terminated by a full-stop.

2. \text{desktop | start | start-target}
   \underline{Click start, point to settings, and then click control panel.} ||
   \text{window-target}
   \underline{Double-click users and passwords.} ||
   \text{contextMenu | action-contextMenu}
   \underline{On the advanced tab, click advanced.} ||

Then split a sentence further into a sequence of record types.
Planning with Record Sequences

Key idea: Grammar on sequences of record types ($G_{RSE}$)

1. Click start, point to settings, and then click control panel. || Double-click users and passwords. || On the advanced tab, click advanced. ||

Split a document into sentences, each terminated by a full-stop.

2. Click start, point to settings, and then click control panel. ||
   window-target

Double-click users and passwords. ||
   contextMenu | action-contextMenu

On the advanced tab, click advanced. ||

Then split a sentence further into a sequence of record types.

3. Goal: Learn patterns of record type sequences within and among sentences
Extended Grammar

1. $S \rightarrow R(\text{start})$
2. $R(r_j.t) \rightarrow FS(r_j, \text{start})R(r_j.t) \mid FS(r_j, \text{start})$
3. $FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j)$
4. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f)$
5. $W(r, r.f) \rightarrow \alpha \mid g(f.v)$
Extended Grammar

\[ D \rightarrow \text{SENT}(t_i, \ldots, t_j) \ldots \text{SENT}(t_l, \ldots, t_m) \]

\[ \text{SENT}(t_i, \ldots, t_j) \rightarrow R(r_{a.t_i}) \ldots R(r_{k.t_j}) \cdot \]

\[ R(r_{i.t}) \rightarrow \text{FS}(r_j, \text{start}) \]

\[ \text{FS}(r, r.f_i) \rightarrow \text{F}(r, r.f_j)\text{FS}(r, r.f_j) | \text{F}(r, r.f_j) \]

\[ \text{F}(r, r.f) \rightarrow \text{W}(r, r.f)\text{F}(r, r.f) | \text{W}(r, r.f) \]

\[ \text{W}(r, r.f) \rightarrow \alpha | g(f.v) | \text{gen}_{-\text{str}}(f.v, i) \]
Extended Grammar

$D \rightarrow \text{SENT}(t_i, \ldots, t_j) \ldots \text{SENT}(t_l, \ldots, t_m)$

$\text{SENT}(t_i, \ldots, t_j) \rightarrow R(r_{a.t_i}) \ldots R(r_{k.t_j})$.

$R(r_i.t) \rightarrow \text{FS}(r_j, \text{start})$

$\text{FS}(r, r.f_i) \rightarrow \text{F}(r, r.f_j) \text{FS}(r, r.f_j) | \text{F}(r, r.f_j)$

$\text{F}(r, r.f) \rightarrow \text{W}(r, r.f) \text{F}(r, r.f) | \text{W}(r, r.f)$

$\text{W}(r, r.f) \rightarrow \alpha | g(f.v) | \text{gen\_str}(f.v, i)$

Straightforward solution: Embed the parameters with the original grammar and train using EM
Extended Grammar

1. \[ D \rightarrow \text{SENT}(t_i, \ldots, t_j) \ldots \text{SENT}(t_l, \ldots, t_m) \]
2. \[ \text{SENT}(t_i, \ldots, t_j) \rightarrow R(r_a.t_i) \ldots R(r_k.t_j) \cdot \]
3. \[ R(r_i.t) \rightarrow \text{FS}(r_j, \text{start}) \]
4. \[ \text{FS}(r, r.f_i) \rightarrow F(r, r.f_j) \text{FS}(r, r.f_j) \mid F(r, r.f_j) \]
5. \[ F(r, r.f) \rightarrow W(r, r.f) F(r, r.f) \mid W(r, r.f) \]
6. \[ W(r, r.f) \rightarrow \alpha \mid g(f.v) \mid \text{gen_str}(f.v, i) \]

Straightforward solution: Embed the parameters with the original grammar and train using EM

Plan B: Extract grammar rules from training data
### Grammar Extraction

<table>
<thead>
<tr>
<th>desktop</th>
<th>start</th>
<th>start-target</th>
<th>window-target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click start, point to settings, and then click control panel.</td>
<td></td>
<td></td>
<td>Double-click users and passwords.</td>
</tr>
<tr>
<td>contextMenu</td>
<td>action-contextMenu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On the advanced tab, click advanced.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Liang et al. (2009)
Grammar Extraction

<table>
<thead>
<tr>
<th>desktop</th>
<th>start</th>
<th>start-target</th>
<th>window-target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click start, point to settings, and then click control panel.</td>
<td></td>
<td>Double-click users and passwords.</td>
<td></td>
</tr>
<tr>
<td>contextMenu</td>
<td>action-contextMenu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On the advanced tab, click advanced.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Liang et al. (2009)

\[
\text{[ desktop start start-target || window-target || contextMenu action-contextMenu || ]}
\]
Grammar Extraction

Liang et al. (2009)

Click start, point to settings, and then click control panel. Double-click users and passwords. On the advanced tab, click advanced.

D

SENT(desk, start, start-target)  SENT(win-target)  SENT(contMenu, action-contMenu)
Grammar Extraction

Click start, point to settings, and then click control panel. Double-click users and passwords. On the advanced tab, click advanced.

Liang et al. (2009)
RST (Mann and Thompson, 1988)

D

Background[N][S]

Elaboration[N][S]

The sound settings window allows you to control your sound devices.

Open the control panel, and click on the sound settings.
RST (Mann and Thompson, 1988)

The sound settings window allows you to control your sound devices.

Open the control panel, and click on the sound settings.
RST (Mann and Thompson, 1988)

- **Background**
  - The sound settings window allows you to control your sound devices.
  - Open the control panel, and click on the sound settings.
RST (Mann and Thompson, 1988)

D

Background\[N][S]\n
Elaboration\[N][S]\n
The sound settings window allows you to control your sound devices.

Open the control panel, and click on the sound settings.
Planning with Rhetorical Structure Theory

Key idea: Grammar using RST relations ($G_{RST}$)
Planning with Rhetorical Structure Theory

Key idea: Grammar using RST relations \((G_{RST})\)

**Assumption**

Each record in the database input corresponds to a unique non-overlapping span in the collocated text, and can be therefore mapped to an EDU.
Grammar Extraction

Click start, point to settings, and then click control panel.
Double-click users and passwords.

On the advanced tab, click advanced.

Liang et al. (2009)
**Grammar Extraction**

<table>
<thead>
<tr>
<th>desktop</th>
<th>start</th>
<th>start-target</th>
<th>window-target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click start, point to settings, and then click control panel.</td>
<td>Double-click users and passwords.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Liang et al. (2009)**

\[
\text{[Click start,]}^{desktop}\ [\text{point to settings,}]^{start}\ [\text{and then click control panel.}]^{start-target}\ \\
\text{[Double-click users and passwords.]}^{window-target}\ [\text{On the advanced tab,}]^{contextMenu}\ \\
\text{[click advanced.]}^{action-contextMenu}
\]
On the advanced tab, click advanced.

Double-click users and passwords.

Click start, point to settings, and then click control panel.

Feng and Hirst (2012)
Double-click users and passwords. On the advanced tab, click advanced.

Click start, point to settings, and then click control panel.

Feng and Hirst (2012)
Extended Grammar

1. $G_{RST}$
2. $R(r_i.t) \rightarrow \text{FS}(r_j, \text{start})$
3. $\text{FS}(r, r.f_i) \rightarrow F(r, r.f_j) \text{FS}(r, r.f_j) \mid F(r, r.f_j)$
4. $F(r, r.f) \rightarrow \text{W}(r, r.f) F(r, r.f) \mid \text{W}(r, r.f)$
5. $\text{W}(r, r.f) \rightarrow \alpha \mid g(f.v) \mid \text{gen_str}(f.v, i)$
Experimental Setup

Data

- **WEATHERGOV**: weather reports [4 sents, 345 words]  
  (Liang et al., 2009)

- **WINHELP**: troubleshooting guides [4.3 sents, 629 words]  
  (Branavan et al., 2009)
Experimental Setup

Data

- WEATHERGOV: weather reports [4 sents, 345 words] (Liang et al., 2009)
- WINHELP: troubleshooting guides [4.3 sents, 629 words] (Branavan et al., 2009)

Evaluation

- Automatic evaluation: BLEU-4
- Human evaluation: Fluency, Semantic Correctness, Coherence
Experimental Setup

Data

- **WeatherGov**: weather reports [4 sents, 345 words] (Liang et al., 2009)
- **WinHelp**: troubleshooting guides [4.3 sents, 629 words] (Branavan et al., 2009)

Evaluation

- Automatic evaluation: BLEU-4
- Human evaluation: Fluency, Semantic Correctness, Coherence

System Comparison

- $G_{RSE}$, $G_{RST}$
- Konstas and Lapata (2012)
- Angeli et al. (2010)
Results: Automatic Evaluation

**WeatherGov**

![Graph showing BLEU-4 scores for different models]

- **Angeli**: 38
- **K&L**: ~30
- **G_{RSE}**: ~30
- **G_{RST}**: ~30

(BLEU-4 scores for Konstas, Lapata (ILCC))
Results: Automatic Evaluation

WeatherGov

<table>
<thead>
<tr>
<th>BLEU-4</th>
<th>ANGELI</th>
<th>K&amp;L</th>
<th>G_{RSE}</th>
<th>G_{RST}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>38</td>
<td>34</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>
Results: Automatic Evaluation

![WeatherGov BLEU-4 chart]

- **Angeli**: 38
- **K&L**: 34
- **G_{RSE}**: 36
- **G_{RST}**: 34

Konstas, Lapata (ILCC)  Concept-to-Text Generation  20 October 2013  19 / 23
Results: Automatic Evaluation

WeatherGov

<table>
<thead>
<tr>
<th>BLEU-4</th>
<th>ANGELI</th>
<th>K&amp;L</th>
<th>G_{RSE}</th>
<th>G_{RST}</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>38</td>
<td>34</td>
<td>36</td>
<td>36</td>
</tr>
</tbody>
</table>
Results: Automatic Evaluation

WinHELP

![Graph showing BLEU-4 scores for different models. The graph compares performance for models Angeli, K&L, $G_{RSE}$, and $G_{RST}$, with Angeli having the highest score.]
Results: Automatic Evaluation

![Graph showing BLEU-4 scores for different methods]

- **WinHELP**
  - **ANGELI**: BLEU-4 score 32
  - **K&L**: BLEU-4 score 38
  - **G_{RSE}**: BLEU-4 score
  - **G_{RST}**: BLEU-4 score
Results: Automatic Evaluation

**WinHELP**

![Bar chart showing BLEU-4 scores for different models.](image)

- **ANGELI**: Score of 32
- **K&L**: Score of 38
- **G_{RSE}**: Score of 42
- **G_{RST}**: Score of 42
Results: Automatic Evaluation

![BLEU-4 Evaluation Chart]

- **Angeli**: BLEU-4 Score 32
- **K&L**: BLEU-4 Score 38
- **G_{RSE}**: BLEU-4 Score 42
- **G_{RST}**: BLEU-4 Score 41

**WinHELP**
Results: Human Evaluation (Coherence)

![Chart showing human evaluation results for different systems.]

- **WEATHERGOV**
  - Angeli: 4
  - K&L: 4
  - G_{RSE}: 4
  - G_{RST}: 4
  - Human: 4

- **WINHELP**
  - Angeli: 2
  - K&L: 2
  - G_{RSE}: 4
  - G_{RST}: 4
  - Human: 4
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GRSE</strong></td>
<td>Click start, point to settings, and then click control panel. Double-click network and dial-up connections. Right-click local area connection, and then click properties. <strong>Click install, and then click add.</strong> Click network monitor driver, and then click ok.</td>
</tr>
<tr>
<td><strong>K&amp;L</strong></td>
<td>Click start, point to settings, and then click control panel. Double-click network and dial-up connections. Double-click network and dial-up connections. Right-click local area connection, <strong>and then click ok.</strong></td>
</tr>
<tr>
<td><strong>HUMAN</strong></td>
<td>Click start, point to settings, click control panel, <strong>and then double-click network and dial-up connections.</strong> Right-click local area connection, and then click properties. <strong>Click install, click protocol, and then click add.</strong> Click network monitor driver, and then click ok.</td>
</tr>
</tbody>
</table>
End-to-end generation system that incorporates document planning

Grammar-based approach allows for document planning naturally: all we need is a discourse grammar

Provide two solutions for document plans:
- Linguistically naive record sequence grammar ($G_{RSE}$)
- RST-inspired grammar ($G_{RST}$)

Future work: more challenging domains (financial, biographies)
Thank you
Decoding

\[ \hat{g} = f \left( \arg \max_{g,h} p(g) \cdot p(g, h | d) \right) \]

- Bottom-up Viterbi search
- Keep k-best derivations at each node, cube pruning (Chiang, 2007)
- \( p(g) \) rescores derivations using an n-gram language model