Joint Models for Concept-to-Text Generation

Ioannis Konstas

Institute for Language, Cognition and Computation
University of Edinburgh

University of Washington, Seattle
22 October 2013
Outline

1. **Grammar-based Generation**
   - Unsupervised Concept-to-text Generation w/ Hypergraphs, NAACL 2012

2. **Inducing Document Plans**
   - Inducing Document Plans for Concept-to-text Generation, ACL 2013

3. **Discriminative Reranking: An exploratory study**
   - Concept-to-text Generation via Discriminative Reranking, ACL 2012
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).

Showers and thunderstorms. High near 70. Cloudy, with a south wind around 20mph, with gusts as high as 40 mph. Chance of precipitation is 100%.
**Concept-to-text** generation refers to the task of automatically producing textual output from nonlinguistic input (Reiter and Dale, 2000).

<table>
<thead>
<tr>
<th>Desktop</th>
<th>Start</th>
<th>Location</th>
<th>Start Target</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd</strong></td>
<td><strong>Name</strong></td>
<td><strong>Type</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>left-click</td>
<td>start button</td>
<td>left-click settings button</td>
<td>start menu button</td>
</tr>
<tr>
<td>left-click control panel button</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Navigate Window</th>
<th>Context Menu</th>
<th>Action Context Menu</th>
<th>Window Target</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd</strong></td>
<td><strong>Name</strong></td>
<td><strong>Type</strong></td>
<td><strong>Cmd</strong></td>
</tr>
<tr>
<td>left-click</td>
<td>accounts and users window</td>
<td>left-click advanced tab</td>
<td>left-click advanced button</td>
</tr>
</tbody>
</table>

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
Traditional NLG Pipeline

- Content Planning
  - Content Selection
  - Document Planning
- Sentence Planning
- Surface Realisation
- Text
Related Work

Traditional NLG Pipeline

- Input Data
- Communicative Goal
- 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

Communicative Goal

Hovy (1993)
Scott and de Souza (1990)
Duboue and Mckeown (2002)
Traditional NLG Pipeline

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

Related Work
- Stent et al. (2004)
- Barzilay and Lapata (2006)

Concept-to-Text Generation
Related Work

Traditional NLG Pipeline

Input Data

Content Planning

Content Selection

Document Planning

Sentence Planning

Surface Realisation

Text

Communicative Goal

Wong and Mooney (2007)
Lu and Ng (2011)
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 (2012a, 2012b, 2013b) 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

Kim and Mooney (2010)
Konstas and Lapata (2012a, 2012b, 2013b)
Angeli et al. (2010)
**Related Work**

### Traditional NLG Pipeline

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

**References**

- Kim and Mooney (2010)
  - Pipeline approach
- Angeli et al. (2010)
  - Unified approach
- Konstas and Lapata (2012a, 2012b, 2013b)
  - 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)

### Cloud Sky Cover

<table>
<thead>
<tr>
<th>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

mostly cloudy,
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>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

mostly cloudy,
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>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

mostly cloudy,
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>Cloud Sky Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
</tr>
<tr>
<td>06:00-09:00</td>
</tr>
<tr>
<td>09:00-12:00</td>
</tr>
</tbody>
</table>

mostly cloudy,
### 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>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

- Mostly cloudy,
Konstas and Lapata, NAACL 2012
Unsupervised Concept-to-text Generation with Hypergraphs

Konstas and Lapata, JAIR 2013. In press
Cloudy, with a low around 10. South wind between 15 and 30 mph.

<table>
<thead>
<tr>
<th>Time</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>9</td>
<td>15</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>S</td>
</tr>
</tbody>
</table>
Cloudy, with a low around 10. South wind between 15 and 30 mph.
<table>
<thead>
<tr>
<th>Time</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>9</td>
<td>15</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

Cloudy, with a low around 10. South wind between 15 and 30 mph.
Cloudy, with a low around 10. South wind between 15 and 30 mph.

**Temperature**

<table>
<thead>
<tr>
<th>Time</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>9</td>
<td>15</td>
<td>21</td>
</tr>
</tbody>
</table>

**Wind Speed**

<table>
<thead>
<tr>
<th>Time</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>15</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

**Cloud Sky Cover**

<table>
<thead>
<tr>
<th>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

**Wind Direction**

<table>
<thead>
<tr>
<th>Time</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>S</td>
</tr>
</tbody>
</table>
### Key Idea

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>9</td>
<td>15</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wind Speed</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>15</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cloud Sky Cover</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wind Direction</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>S</td>
</tr>
</tbody>
</table>

Cloudy, with a low around 10. South wind between 15 and 30 mph.
**Key Idea**

### Temperature

<table>
<thead>
<tr>
<th>Time</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>9</td>
<td>15</td>
<td>21</td>
</tr>
</tbody>
</table>

### Wind Speed

<table>
<thead>
<tr>
<th>Time</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>15</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

### Cloud Sky Cover

<table>
<thead>
<tr>
<th>Time</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-09:00</td>
<td>25-50</td>
</tr>
<tr>
<td>09:00-12:00</td>
<td>50-75</td>
</tr>
</tbody>
</table>

### Wind Direction

<table>
<thead>
<tr>
<th>Time</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00-21:00</td>
<td>S</td>
</tr>
</tbody>
</table>

Cloudy, with a low around 10. South wind between 15 and 30 mph.

Partly cloudy, with a low around 9. Breezy, with a south wind between 15 and 30 mph.
Grammar

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

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

$R(skyCover_{1.t}) \rightarrow FS(temperature_{1}, start)R(temperature_{1}.t)$
Grammar-based Generation

Key Idea

Grammar

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

\( R(skyCover_1.t) \rightarrow FS(temperature_1, start)R(temperature_1.t) \)
Grammar-based Generation

Key Idea

Grammar

Rain Chance

Time Mode

06-21 Def
06-09 Lkly
06-13 Def
09-21 Def
13-21 Def

Thunder Chance

Time Mode

06-21 Def
06-09 Lkly
06-13 Chc
09-21 Def
13-21 Def

Temperature

Time Min Mean Max

06-21 52 61 70
06-09 75-100
06-13 50-75
09-21 75-100
13-21 75-100

Sky Cover

Time Percent (%)

06-21 06-09 06-13 09-21 13-21
75-100 75-100 50-75 75-100 75-100

Wind Direction

Time Mode

06-21 S
06-09 06-13 09-21 13-21

Wind Speed

Time Min Mean Max

06-21 11 22 29
06-09 75-100
06-13 50-75
09-21 75-100
13-21 75-100

Gust

Time Min Mean Max

06-21 0 20 39
06-09 06-13 09-21 13-21

Precipitation Potential

Time Min Mean Max

06-21 26 81 100
06-09 06-13 09-21 13-21

1 S → R(start)

2 R(r_i . t) → FS(r_j , start) R(r_j . t) | FS(r_j , start)

3 FS(r , r.f_i) → F(r , r.f_j) FS(r , r.f_j) | F(r , r.f_j)

FS(wSpeed_1 , min) → F(wSpeed_1 , max) FS(wSpeed_1 , max)
Grammar

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

\( F(gust_1, \text{min}) \rightarrow W(gust_1, \text{mean})F(gust_1, \text{mean}) \)
S → R(start)

R(r_j.t) → FS(r_j, start)R(r_j.t) | FS(r_j, start)

FS(r, r.f_i) → F(r, r.f_j)FS(r, r.f_j) | F(r, r.f_j)

F(r, r.f) → W(r, r.f)F(r, r.f) | W(r, r.f)

W(r, r.f) → α | g(f.v)

W(skyCover_1, %) → cloudy [%v = ‘75-100’]
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
Decoding

\[ \hat{g} = f \left( \arg \max_{g,h} p(g) \cdot p(g, h | d) \right) \]
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 by linearly interpolating:
  - n-gram language model
  - dependency model (DMV; Klein and Manning, 2004)
- Implement using hypergraphs (Klein and Manning, 2001)
Leaf nodes $\epsilon$ emit a k-best list of words

$$W_{0,1}(\text{skyCover}_{1.t,\%})$$
Decoding
Decoding

Grammar-based Generation

Key Idea

Konstas (ILCC)

Concept-to-Text Generation

22 October 2013 12 / 50
Decoding

Grammar-based Generation

Key Idea

Konstas (ILCC)  Concept-to-Text Generation  22 October 2013  12 / 50
Experimental Setup

Data

- **RoboCup**: simulated sportscasting [214 words] (Chen and Mooney, 2008)
- **WeatherGov**: weather reports [4 sents, 345 words] (Liang et al., 2009)
- **ATIS**: flight booking [1 sent, 927 words] (Zettlemoyer and Collins, 2007)
- **WinHelp**: troubleshooting guides [4.3 sents, 629 words] (Branavan et al., 2009)
Experimental Setup

Data

- RoboCup: simulated sportscasting [214 words] (Chen and Mooney, 2008)
- WeatherGov: weather reports [4 sents, 345 words] (Liang et al., 2009)
- ATIS: flight booking [1 sent, 927 words] (Zettlemoyer and Collins, 2007)
- WinHelp: troubleshooting guides [4.3 sents, 629 words] (Branavan et al., 2009)

Evaluation

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

Data

- **RoboCup**: simulated sportscasting [214 words] (Chen and Mooney, 2008)
- **WeatherGov**: weather reports [4 sents, 345 words] (Liang et al., 2009)
- **ATIS**: flight booking [1 sent, 927 words] (Zettlemoyer and Collins, 2007)
- **WinHelp**: troubleshooting guides [4.3 sents, 629 words] (Branavan et al., 2009)

Evaluation

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

System Comparison

- 1-best, \texttt{k-Best-lm}, \texttt{k-Best-lm-dmv}
- Angeli et al. (2010)
Results: Automatic Evaluation

RoboCup

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Best</td>
<td>10.79</td>
</tr>
<tr>
<td>RoboCup - Best</td>
<td>28.7</td>
</tr>
<tr>
<td>RoboCup - Best-lm</td>
<td>30.9</td>
</tr>
<tr>
<td>RoboCup - Best-lm-dmv</td>
<td>29.73</td>
</tr>
</tbody>
</table>
Results: Automatic Evaluation

WEATHERGOV

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Best</td>
<td>8.64</td>
</tr>
<tr>
<td>Angeli</td>
<td>38.4</td>
</tr>
<tr>
<td>k-Best-LM</td>
<td>33.7</td>
</tr>
<tr>
<td>k-Best-LM-DMV</td>
<td>34.18</td>
</tr>
</tbody>
</table>

1. Angeli
2. Best-lm
3. Best-lm-dmv
Results: Automatic Evaluation

ATIS

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Best</td>
<td>11.85</td>
</tr>
<tr>
<td>Angeli</td>
<td>26.77</td>
</tr>
<tr>
<td>k-Best-lm</td>
<td>29.3</td>
</tr>
<tr>
<td>k-Best-lm-dmv</td>
<td>30.37</td>
</tr>
</tbody>
</table>
Results: Automatic Evaluation

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

- **1-Best**: 16.02
- **Angel**: 32.21
- **k-Best-LM**: 38.26
- **k-Best-LM-DMV**: 39.03

The chart compares the automatic evaluation of different models using BLEU-4 scores on the WinHelp dataset.
Output

**WeatherGov**

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Cloud Sky Cover</th>
<th>Chance of Rain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td><strong>Percent (%)</strong></td>
<td><strong>Mode</strong></td>
</tr>
<tr>
<td>06:00-21:00</td>
<td>06:00-21:00</td>
<td>06:00-11:00</td>
</tr>
<tr>
<td>Min 30</td>
<td>Mean 38</td>
<td>Slight Chance</td>
</tr>
<tr>
<td>Mean 38</td>
<td>Max 44</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wind Speed</th>
<th>Wind Direction</th>
<th>Precipitation Potential (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td><strong>Mode</strong></td>
<td><strong>Time</strong></td>
</tr>
<tr>
<td>06:00-21:00</td>
<td>ENE</td>
<td>06:00-21:00</td>
</tr>
<tr>
<td>Min 6</td>
<td></td>
<td>Min 9</td>
</tr>
<tr>
<td>Mean 6</td>
<td></td>
<td>Mean 20</td>
</tr>
<tr>
<td>Max 7</td>
<td></td>
<td>Max 35</td>
</tr>
</tbody>
</table>

**k-Best:** A chance of rain showers before 11am. Mostly cloudy, with a high near 44. East wind between 6 and 7 mph.

**Angeli:** A chance of showers. Patchy fog before noon. Mostly cloudy, with a high near 44. East wind between 6 and 7 mph. Chance of precipitation is 35%

**Human:** A 40 percent chance of showers before 10am. Mostly cloudy, with a high near 44. East northeast wind around 7 mph.
### ATIS

<table>
<thead>
<tr>
<th>Flight</th>
<th>Day</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>day</td>
<td>type</td>
</tr>
<tr>
<td>milwaukee</td>
<td>dep/ar/ret</td>
<td>what</td>
</tr>
<tr>
<td>to</td>
<td>saturday</td>
<td>query flight</td>
</tr>
</tbody>
</table>

**Input:**
- **from**: Milwaukee
- **to**: Phoenix
- **Day**: Saturday
- **Search type**: flight

**k-Best:** What are the flights from Milwaukee to Phoenix on Saturday

**ANGELI:** Show me the flights between Milwaukee and Phoenix on Saturday

**HUMAN:** Milwaukee to Phoenix on Saturday
Dependency Output

ATIS

ROOT

on

show

on

me

on

the

flights

on

from

on

Milwaukee

to

Phoenix

on

Phoenix

on

Saturday
Conclusions

- Generation as parsing problem
- Unsupervised end-to-end generation system
- Performance comparable to state-of-the-art
Conclusions

- Generation as parsing problem
- Unsupervised end-to-end generation system
- Performance comparable to state-of-the-art
- What about document planning?
Inducing Document Plans for Concept-to-text Generation, ACL 2013
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 (2012a, 2012b, 2013a)
Joint approach
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 (2012a, 2012b, 2013a)
Joint approach

Konstas and Lapata (2013a)
Joint approach
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

- Re-use the generation model based on a PCFG grammar of input
Key Idea: Grammar-based document plans

- 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

- 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

- 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
Key idea: Grammar on sequences of record types ($G_{RSE}$)
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.
Inducing Document Planning

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

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. \( S \rightarrow R(start) \)
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. \( W(r, r.f) \rightarrow \alpha \mid g(f.v) \)
Extended Grammar

1. \( D \rightarrow SENT(t_i, \ldots, t_j) \ldots SENT(t_l, \ldots, t_m) \)
2. \( SENT(t_i, \ldots, t_j) \rightarrow R(r_a \cdot t_i) \ldots R(r_k \cdot t_j) \cdot \)
3. \( R(r_i \cdot t) \rightarrow FS(r_j, start) \)
4. \( FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) | F(r, r.f_j) \)
5. \( F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) | W(r, r.f) \)
6. \( W(r, r.f) \rightarrow \alpha | g(f.v) | gen\_str(f.v, i) \)
Extended Grammar

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

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>Double-click users and passwords.</td>
<td></td>
</tr>
<tr>
<td>contextMenu</td>
<td>action-contextMenu</td>
<td>click advanced.</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>Double-click users and passwords.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**contextMenu**

On the advanced tab, click advanced.

Liang et al. (2009)

\[
\text{[ desktop start start-target || window-target || contextMenu action-contextMenu || ]}
\]
**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)

\[
\text{D} = \text{SENT(desk, start, start-target)} \quad \text{SENT(win-target)} \quad \text{SENT(contMenu, action-contMenu)}
\]

\[
\begin{align*}
\text{R(desk)} & \quad \text{R(start)} & \quad \text{R(start-target)} \\
\text{R(win-target)} & \quad \text{R(contMenu)} & \quad \text{R(action-contMenu)}
\end{align*}
\]
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,</td>
<td>point to settings,</td>
<td>and then click control panel.</td>
<td>Double-click users and passwords.</td>
</tr>
</tbody>
</table>

On the advanced tab, click advanced.

Liang et al. (2009)

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

\[
D
\]

\[
\text{SENT(desk, start, start-target)}
\]

\[
\text{R(desk)} \quad \text{R(start)} \quad \text{R(start-target)}
\]

\[
\text{SENT(win-target)}
\]

\[
\text{R(win-target)}
\]

\[
\text{SENT(contMenu, action-contextMenu)}
\]

\[
\text{R(contMenu)} \quad \text{R(action-contextMenu)}
\]

\[
\text{[SENT(win-target)-SENT(contMenu, action-contextMenu)]}
\]

\[
\text{R(start)} \quad \text{R(start-target)}
\]

\[
\text{SENT(start, start-target)}
\]

\[
\text{SENT(win-target)}
\]

\[
\text{SENT(contMenu, action-contextMenu)}
\]

\[
\text{R(contMenu)} \quad \text{R(action-contextMenu)}
\]
RST (Mann and Thompson, 1988)

D

Background[N][S]

Elaboration[N][S]

Open the control panel, and click on the sound settings.

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

- Background[N][S]
  - Open the control panel, and click on the sound settings.
  - The sound settings window allows you to control your sound devices.
  - Elaboration[N][S]
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.
Planning with Rhetorical Structure Theory

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

<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,</td>
<td>and then click control panel.</td>
<td>Double-click users and passwords.</td>
<td></td>
</tr>
<tr>
<td>contextMenu</td>
<td>action-contextMenu</td>
<td>click advanced.</td>
<td>Liang et al. (2009)</td>
</tr>
</tbody>
</table>
Inducing Document Planning

Planning with RST

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,</td>
<td>point to settings,</td>
<td>and then click control panel.</td>
<td>Double-click users and passwords.</td>
</tr>
<tr>
<td>contextMenu</td>
<td>action-contextMenu</td>
<td>click advanced.</td>
<td></td>
</tr>
</tbody>
</table>

Liang et al. (2009)

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

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

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

Feng and Hirst (2012)
Extended Grammar

1. $G_{RST}$
2. $R(r_i.t) \rightarrow FS(r_j, \text{start})$
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. $W(r, r.f) \rightarrow \alpha \mid g(f.v) \mid \text{gen\_str}(f.v, i)$
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 (2012a)
- Angeli et al. (2010)
Results: Automatic Evaluation

![Comparison of BLEU-4 scores for different models](chart.png)

**WEATHERGOV**

- **ANGELI**: 38.4
- **K&L**: 33.7
- **G_{RSE}**: 35.6
- **G_{RST}**: 36.54
Results: Automatic Evaluation

![Bar chart showing BLEU-4 scores for different models]

- **ANGELI**: 32.21
- **K&L**: 38.26
- **G_{RSE}**: 40.92
- **G_{RST}**: 40.65

The bar chart above illustrates the BLEU-4 scores for different models in the WinHelp dataset. The scores indicate the quality of translation accuracy, with higher numbers representing better performance.
Results: Human Evaluation (Coherence)

WeatherGov

<table>
<thead>
<tr>
<th></th>
<th>ANGELI</th>
<th>K&amp;L</th>
<th>G_{RSE}</th>
<th>G_{RST}</th>
<th>HUMAN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.82</td>
<td>3.59</td>
<td>4.18</td>
<td>4.1</td>
<td>4.11</td>
</tr>
</tbody>
</table>

WinHelp

<table>
<thead>
<tr>
<th></th>
<th>ANGELI</th>
<th>K&amp;L</th>
<th>G_{RSE}</th>
<th>G_{RST}</th>
<th>HUMAN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.97</td>
<td>2.93</td>
<td>3.35</td>
<td>3.22</td>
<td>4.25</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GRSE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>K&amp;L</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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, and then click ok.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HUMAN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Click start, point to settings, click control panel, and then double-click network and dial-up connections. 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>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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 ($G_{RST}$)

How about a more sophisticated content selection model on the field level?
Konstas and Lapata, ACL 2012

Concept-to-text Generation via Discriminative Reranking
Discriminative Reranking Model

Original Model

- Joint model allows for more global decisions
- **Forest rescoring** allows for rescoring k-best trees at all internal nodes via LM+DMV integration
Discriminative Reranking Model

Original Model

- Joint model allows for more global decisions
- **Forest rescoring** allows for rescoring k-best trees at all internal nodes via LM+DMV integration

Discriminative Reranking Model
Discriminative Reranking Model

Original Model

- Joint model allows for more global decisions
- **Forest rescoring** allows for rescoring k-best trees at all internal nodes via LM+DMV integration

Discriminative Reranking Model

- Use decoder of the original model as a baseline
- Introduce lexical and structural features up to the field level
- **Discriminative reranking** reranks k-best trees at all internal nodes
- Train using an online learning algorithm
Hypergraph Reranking

Hidden correspondence \( h \) between database \( d \) and words \( w \)

Give me the flights leaving Seattle October 22nd coming back to New York
Hypergraph Reranking

Hidden correspondence $h$ between database $d$ and words $w$

$$(\hat{g}, \hat{h}) = \arg \max_{g,h} \alpha \cdot \Phi(d, \hat{g}, \hat{h})$$

- $\Phi = (\Phi_1, \ldots, \Phi_m)$: high dimensional feature representation
- $\alpha$: weight vector
- Learn $\alpha$ with averaged structured perceptron (Collins, 2002)
Hidden correspondence $\mathbf{h}$ between database $\mathbf{d}$ and words $\mathbf{w}$

$$ (\hat{\mathbf{g}}, \hat{\mathbf{h}}) = \arg \max_{g, h} \alpha \cdot \Phi(\mathbf{d}, \hat{\mathbf{g}}, \hat{\mathbf{h}}) $$

- $\Phi = (\Phi_1, \ldots, \Phi_m)$: high dimensional feature representation
- $\alpha$: weight vector
- Learn $\alpha$ with averaged structured perceptron (Collins, 2002)
Hidden correspondence $\mathbf{h}$ between database $\mathbf{d}$ and words $\mathbf{w}$

$$(\hat{\mathbf{g}}, \hat{\mathbf{h}}) = \arg \max_{\mathbf{g}, \mathbf{h}} \alpha \cdot \Phi(\mathbf{d}, \hat{\mathbf{g}}, \mathbf{h})$$

- $\Phi = (\Phi_1, \ldots, \Phi_m)$ : high dimensional feature representation
- $\alpha$ : weight vector
- Learn $\alpha$ with averaged structured perceptron (Collins, 2002)
Oracle Derivation

Oracle derivation \((w^*, h^+))

- Use the decoder of the original model but observe the training text.
- \(w^*\): gold standard text
- \(h^+\): best latent configuration
Baseline Features

- **Baseline Model Feature (local)**: Log score of decoder of the original model
Baseline Features

- **Baseline Model Feature (local)**: Log score of decoder of the original model
- **Alignment Features (local)**: Count of each PCFG rule

\[
R(\text{search}_{1.t}) \quad \text{FS(\text{flight}_{1.t}, \text{start})} \quad R(\text{flight}_{1.t})
\]

\[
R(r_i.t) \rightarrow \text{FS}(r_j, \text{start}) R(r_j.t)
\]
Lexical Features

- Word Bigrams/Trigrams (non-local)
- Number of Words per Field (local)
- Consecutive Word/Bigram/Trigram (non-local)
Lexical Features

- **Word Bigrams/Trigrams (non-local)**
- **Number of Words per Field (local)**
- **Consecutive Word/Bigram/Trigram (non-local)**

\[
\begin{align*}
    F_{0,3}(\text{search}_{1}.t,\text{start}) & \\
    W_{0,1}(\text{search}_{1}.t,\text{type}) & \quad \ldots \quad W_{1,3}(\text{search}_{1}.t,\text{what}) \\
    \begin{pmatrix}
    \text{show} \\
    \text{me} \\
    \text{what} \\
    \ldots
    \end{pmatrix} & \quad \ldots \\
    \begin{pmatrix}
    \text{me} \\
    \text{the} \\
    \text{me flights} \\
    \text{the flights} \\
    \ldots
    \end{pmatrix}
\end{align*}
\]

\(<\text{show me the}>, <\text{show me flights}>, \text{etc.}\)
Lexical Features

- Word Bigrams/Trigrams (non-local)
- Number of Words per Field (local)
- Consecutive Word/Bigram/Trigram (non-local)

```
FS_{0,3}(search_1.t,start)
```

```
W_{0,1}(search_1.t,type)
```

```
W_{1,3}(search_1.t,what)
```

```
<show me the>, <show me flights>, etc.
```

```
<2 | from>
```

```
F_{2,4}(flight_1.t,from)
```

```
<2 words
```
Content Selection at the Field Level Features

- Field Bigrams/Trigrams (non-local)
- Number of Fields per Record (local)
- Fields with no Value (local)
Content Selection at the Field Level Features

- Field Bigrams/Trigrams (non-local)
- Number of Fields per Record (local)
- Fields with no Value (local)
k-best Decoding

- Bottom-up Viterbi search
- Keep k-best derivations at each node, cube pruning (Chiang, 2007)
- Score of \( j \)-th derivation: \( \alpha \cdot \Phi_L(e) + \alpha \cdot \Phi_N(<e,j>) \)
k-best Decoding

\[
\begin{align*}
\text{FS}_{0,5}(\text{search}_1.t, \text{start}) \\
\text{W}_{4,5}(\text{search}_1.t, \text{what}) \\
\text{F}_{0,2}(\text{search}_1.t, \text{type}) \\
\text{W}_{1,2}(\text{search}_1.t, \text{type}) \\
\text{W}_{0,1}(\text{search}_1.t, \text{type}) \\
\end{align*}
\]
k-best Decoding

(show me * the flights [type what])
(show me * what flights [type what])
(show me * all flights [type what])

FS0,5(search1.t,start)

(show me [type])
(show the [type])
(what are [type])

F0,2(search1.t,type)

W0,1(search1.t,type)
W1,2(search1.t,type)

W4,5(search1.t,what)

(show [∅])
(me [∅])
(what [∅])

(···)

(show [∅])
(me [∅])
(what [∅])

(···)
k-best Decoding

(show me * the flights [type what])
(show me * what flights [type what])
(show me * all flights [type what])
...

(show me [type])
(show the [type])
(what are [type])
...

FS_{0,5}(search_{1.t,start})

F_{0,2}(search_{1.t,type})

W_{0,1}(search_{1.t,type})

W_{1,2}(search_{1.t,type})

W_{4,5}(search_{1.t,what})

(show [∅])
(me [∅])
(what [∅])
...

(show [∅])
(me [∅])
(what [∅])
...

(flights [∅])
(flight [∅])
(airline [∅])
...
Experimental Setup

Data

- ATIS: flight booking [1 sent, 927 words] (Zettlemoyer and Collins, 2007)
Experimental Setup

Data
- ATIS: flight booking [1 sent, 927 words]
  (Zettlemoyer and Collins, 2007)

Evaluation
- Automatic evaluation: BLEU-4
- Human evaluation: Fluency, Semantic correctness
Experimental Setup

Data

- **ATIS**: flight booking [1 sent, 927 words]  
  (Zettlemoyer and Collins, 2007)

Evaluation

- Automatic evaluation: BLEU-4
- Human evaluation: Fluency, Semantic correctness

System Comparison

- Baseline: 1-BEST+BASE+ALIGN
- k-best (+Lexical): k-BEST+BASE+ALIGN+LEX
- k-best (+Structural): k-BEST+BASE+ALIGN+LEX+STR
- Angeli et al. (2010)
Results: Automatic Evaluation

**WinHELP**

<table>
<thead>
<tr>
<th>BLEU-4</th>
<th>Baseline</th>
<th>Angeli</th>
<th>+Lexical</th>
<th>+Structural</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.93</td>
<td></td>
<td>26.77</td>
<td>28.66</td>
<td>31.62</td>
</tr>
</tbody>
</table>
Conclusions

- Discriminative reranking using the structured perceptron
- Introduced local and non-local features
- More sophisticated content selection on the field level
Future Work

Where do we go from here?

- More challenging factual domains: biographies from Wikipedia
- More sophisticated sentence planning: aggregation, coreference resolution
- Real induction for document planning grammar $G_{RSE}$: ID/LP grammars
- Discriminative reranking: use of parallelisable online learning algorithms
- More engineering: scaling can be an issue for large documents
- Apply document planning grammars to summarisation
Thank you
<table>
<thead>
<tr>
<th>System</th>
<th>Fluency</th>
<th>SemCor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-BEST</td>
<td>2.47</td>
<td>2.33</td>
</tr>
<tr>
<td>k-BEST</td>
<td>4.31</td>
<td>3.96</td>
</tr>
<tr>
<td>ANGELI</td>
<td>4.03</td>
<td>3.70</td>
</tr>
<tr>
<td>HUMAN</td>
<td>4.47</td>
<td>4.37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Fluency</th>
<th>SemCor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-BEST</td>
<td>1.82</td>
<td>2.05</td>
</tr>
<tr>
<td>k-BEST</td>
<td>3.92</td>
<td>3.30</td>
</tr>
<tr>
<td>ANGELI</td>
<td>4.26</td>
<td>3.60</td>
</tr>
<tr>
<td>HUMAN</td>
<td>4.61</td>
<td>4.03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Fluency</th>
<th>SemCor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-BEST</td>
<td>2.40</td>
<td>2.46</td>
</tr>
<tr>
<td>k-BEST</td>
<td>4.01</td>
<td>3.87</td>
</tr>
<tr>
<td>ANGELI</td>
<td>3.56</td>
<td>3.33</td>
</tr>
<tr>
<td>HUMAN</td>
<td>4.10</td>
<td>4.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Fluency</th>
<th>SemCor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-BEST</td>
<td>2.57</td>
<td>2.10</td>
</tr>
<tr>
<td>k-BEST</td>
<td>3.41</td>
<td>3.05</td>
</tr>
<tr>
<td>ANGELI</td>
<td>3.57</td>
<td>2.80</td>
</tr>
<tr>
<td>HUMAN</td>
<td>4.15</td>
<td>4.04</td>
</tr>
</tbody>
</table>
Discriminative Reranking Results: Human Evaluation

<table>
<thead>
<tr>
<th>System</th>
<th>Fluency</th>
<th>SemCor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-BEST</td>
<td>2.70</td>
<td>3.05</td>
</tr>
<tr>
<td>k-BEST</td>
<td>4.02</td>
<td>4.04</td>
</tr>
<tr>
<td>ANGELI</td>
<td>3.74</td>
<td>3.17</td>
</tr>
<tr>
<td>HUMAN</td>
<td>4.18</td>
<td>4.02</td>
</tr>
</tbody>
</table>

- **k-BEST** significantly better than **1-BEST** and **ANGELI** ($\alpha < 0.01$)
- **k-BEST** and **HUMAN** are not significantly different
Hypergraphs

Definition
An ordered hypergraph \( H \) is a tuple \( \langle N, E, t, R \rangle \), where \( N \) is a finite set of nodes, \( E \) is a finite set of hyperarcs, \( t \in N \) is a target node and \( R \) is the set of weights. Each hyperarc \( e \in E \) is a triple \( e = \langle T(e), h(e), f(e) \rangle \), where \( h(e) \in N \) is its head node, \( T(e) \in N^* \) is a set of tail nodes and \( f(e) \) is a monotonic weight function \( R_{|T(e)|} \) to \( R \).
Definition

An ordered hypergraph $H$ is a tuple $\langle N, E, t, R \rangle$, where $N$ is a finite set of nodes, $E$ is a finite set of hyperarcs, $t \in N$ is a target node and $R$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$ is its head node, $T(e) \in N^*$ is a set of tail nodes and $f(e)$ is a monotonic weight function $\mathbb{R}_{|T(e)|}$ to $\mathbb{R}$. 
An ordered hypergraph \( H \) is a tuple \( \langle N, E, t, R \rangle \), where \( N \) is a finite set of nodes, \( E \) is a finite set of hyperarcs, \( t \in N \) is a target node and \( R \) is the set of weights. Each hyperarc \( e \in E \) is a triple \( e = \langle T(e), h(e), f(e) \rangle \), where \( h(e) \in N \) is its head node, \( T(e) \in N^* \) is a set of tail nodes and \( f(e) \) is a monotonic weight function \( R|_{T(e)} \) to \( R \).
Hypergraphs

Definition

An ordered hypergraph $H$ is a tuple $\langle N, E, t, R \rangle$, where $N$ is a finite set of nodes, $E$ is a finite set of hyperarcs, $t \in N$ is a target node and $R$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$ is its head node, $T(e) \in N^*$ is a set of tail nodes and $f(e)$ is a monotonic weight function $R_{|T(e)|}$ to $R$. 

\[
\text{Diagram:}
\begin{array}{c}
\text{t} \\
\quad \downarrow \\
\text{f(e)} \\
\quad \downarrow \\
\text{b}
\end{array}
\]
Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, R \rangle$

\[ f(e) = f(\text{FS}_{5,7}(\text{flight}_1.t, \text{start})) \otimes f(\text{R}_{7,9}(\text{flight}_1.t)) \otimes w(\text{R}(\text{search}_1.t) \rightarrow \text{FS}(\text{flight}_1, \text{start}) \ \text{R}(\text{flight}_1.t)) \]

\[ \text{R}(r_i.t) \rightarrow \text{FS}(r_j, \text{start}) \ \text{R}(r_j.t) \]
Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, R \rangle$

$$f(e) = f(FS_{5,7}(flight_1.t, start)) \otimes f(R_{7,9}(flight_1.t)) \otimes w(R(search_1.t) \rightarrow FS(flight_1, start) R(flight_1.t))$$

$$R(r_i.t) \rightarrow FS(r_j, start) R(r_j.t)$$
Hypergraph Construction

Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, R \rangle$

\[
f(e) = f(FS_{5,7}(flight_1.t, start)) \times f(R_{7,9}(flight_1.t)) \times\]
\[
w(R(search_1.t) \rightarrow FS(flight_1, start) R(flight_1.t))
\]

\[
R(r_i.t) \rightarrow FS(r_j, start) R(r_j.t)
\]
Hypergraph Construction

Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, R \rangle$

$$f(e) = f(FS_{5,7}(flight_1.t, start)) \otimes f(R_{7,9}(flight_1.t)) \otimes w(R(search_1.t) \rightarrow FS(flight_1, start) R(flight_1.t))$$

$$R(r_i.t) \rightarrow FS(r_j, start) R(r_j.t)$$
Hypergraph Construction

Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, R \rangle$

$$f(e) = f(FS_{5,7}(flight_1.t, start)) \otimes f(R_{7,9}(flight_1.t)) \otimes w(R(search_1.t) \rightarrow FS(flight_1, start) \ R(flight_1.t))$$

$$R(r_i.t) \rightarrow FS(r_j, start) \ R(r_j.t)$$
Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, R \rangle$

$$f(e) = f(FS_{5,7}(flight_1.t, start)) \otimes f(R_{7,9}(flight_1.t)) \otimes w(R(search_1.t) \rightarrow FS(flight_1, start) \ R(flight_1.t))$$

$$R(r_i.t) \rightarrow FS(r_j, start) R(r_j.t)$$
Hypergraph Example

\[
\begin{align*}
S_0,7 & \rightarrow R_{0,7}(\text{start}) \\
FS_{0,1}(\text{skyCover}_1, \text{start}) & \rightarrow F_{0,1}(\text{skyCover}_1, \%)
\end{align*}
\]
Determining Text Length

- Train a linear regression model
- Idea: The more records and fields that have values in the database → the more facts need to be uttered
- Input to the model: Flattened version of the database input, i.e. each feature is a record-field pair
- Feature values: Values vs Counts of Fields