Learning to generate: Concept-to-text generation using machine learning

Ioannis Konstas

Institute for Language, Cognition and Computation
University of Edinburgh

Aberdeen, NLG Summer School
21 July 2015
**Introduction**

**Motivation**

**Sensor Data**

**Full Descriptor**

<table>
<thead>
<tr>
<th>Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETTING;VENTIL;FiO2 (36%)</td>
<td>10.30</td>
</tr>
<tr>
<td>MEDICATION;Morphine</td>
<td>10.44</td>
</tr>
<tr>
<td>ACTION;CARE;TURN/CHANGE POSITION;SUPINE</td>
<td>10.46-10.47</td>
</tr>
<tr>
<td>ACTION;RESP;HAND BABY</td>
<td>10.47-10.51</td>
</tr>
<tr>
<td>SETTING;VENTIL;FiO2 (60%)</td>
<td>10.47</td>
</tr>
<tr>
<td>ACTION;RESP;INTUBATE</td>
<td>10.51-10.52</td>
</tr>
</tbody>
</table>

**Action Records**
**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).

<table>
<thead>
<tr>
<th>Wind Chill</th>
<th>Temperature</th>
<th>Wind Speed</th>
<th>Wind Direction</th>
<th>Gust</th>
<th>Precipitation Potential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Min Mean Max</td>
<td>Time Min Mean Max</td>
<td>Time Min Mean Max</td>
<td>Time Mode</td>
<td>Time Mode</td>
<td>Time Min Mean Max</td>
</tr>
<tr>
<td>06-21 0 0 0</td>
<td>06-21 52 61 70</td>
<td>06-21 11 22 29</td>
<td>06-21 S</td>
<td>06-21 0 20 39</td>
<td>06-21 26 81 100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sky Cover</th>
<th>Rain Chance</th>
<th>Snow Chance</th>
<th>Sleet Chance</th>
<th>Freezing Rain Chance</th>
<th>Thunder Chance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Percent (%)</td>
<td>Time Mode</td>
<td>Time Mode</td>
<td>Time Mode</td>
<td>Time Mode</td>
<td>Time Mode</td>
</tr>
<tr>
<td>06-21 75-100</td>
<td>06-21 Def</td>
<td>06-21 Def</td>
<td>06-21 Def</td>
<td>06-21 Def</td>
<td>06-21 Def</td>
</tr>
<tr>
<td>06-09 75-100</td>
<td>06-09 Lkly</td>
<td>06-09 Lkly</td>
<td>06-09 Lkly</td>
<td>06-09 Lkly</td>
<td>06-09 Lkly</td>
</tr>
<tr>
<td>06-13 50-75</td>
<td>06-13 Def</td>
<td>06-13 Def</td>
<td>06-13 Def</td>
<td>06-13 Def</td>
<td>06-13 Chc</td>
</tr>
<tr>
<td>09-21 75-100</td>
<td>09-21 Def</td>
<td>09-21 Def</td>
<td>09-21 Def</td>
<td>09-21 Def</td>
<td>09-21 Def</td>
</tr>
<tr>
<td>13-21 75-100</td>
<td>13-21 Def</td>
<td>13-21 Def</td>
<td>13-21 Def</td>
<td>13-21 Def</td>
<td>13-21 Def</td>
</tr>
</tbody>
</table>

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).

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

What has been done so far?
Introduction

What has been done so far?

- Expert knowledge deployed for the creation of hand-crafted rules - single domain
Introduction

What has been done so far?

- Expert knowledge deployed for the creation of hand-crafted rules - single domain
- Manually annotated corpora - discourse relations, alignments
Introduction

What has been done so far?

- Expert knowledge deployed for the creation of hand-crafted rules - single domain
- Manually annotated corpora - discourse relations, alignments
- Breakdown of process into a pipeline of modules
What we will look into today?
What we will look into today?

- Recast NLG into a generative model
Introduction

What we will look into today?

- Recast NLG into a generative model
- Learn parameters from (un)-annotated data - multiple domains
What we will look into today?

- Recast NLG into a generative model
- Learn parameters from (un)-annotated data - multiple domains
- Search for the best parameters that fit the input and **decode** into text
Outline

- Problem Formulation
- Learning Alignments
- Pipeline Approach
- Joint Approaches
Outline

- Problem Formulation
- Learning Alignments
- Pipeline Approach
- Joint Approaches
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**

- **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)
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)
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)
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>

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 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>

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.
### Key Idea

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

<table>
<thead>
<tr>
<th>Cloud Sky Cover</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
</tr>
<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 Speed</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>15</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>06:00-21:00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wind Direction</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
</tr>
<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.
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>

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

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

#### 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>

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

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

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

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

<table>
<thead>
<tr>
<th>Wind Direction</th>
<th>Time</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<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.

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.
### 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>

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

#### 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>

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

Key Idea

Traditional NLG Pipeline

Input Data

Content Planning

Sentence Planning

Surface Realisation

Text

Communicative Goal

Konstas (ILCC)
Traditional NLG Pipeline

Input Data

Communicative Goal

Content Planning

Content Selection

Document Planning

Sentence Planning

Surface Realisation

Text

Liang et al. (2009)
Liang et al., ACL 2009
Learning Semantic Correspondences with Less Supervision
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%.
Generative Story

1. Record choice: choose a sequence of records \( r = (r_1, \ldots, r_{|r|}) \)

\[
p(r \mid d) = \prod_{i} p(r_i.t \mid r_{i-1}.t) \frac{1}{|s(r_i.t)|}
\]
Generative Story

1. Record choice: choose a sequence of records \( r = (r_1, \ldots, r_{|r|}) \)

\[
p(r \mid d) = \prod_i p(r_i.t \mid r_{i-1}.t) \frac{1}{|s(r_i.t)|}
\]

2. Field choice: for each chosen record \( r_i \), select a sequence of fields \( f_i = (f_{i1}, \ldots, f_{i|f_i|}) \)

\[
p(f \mid r_i.t) = \prod_k p(r_i.f_k \mid r_i.f_{k-1})
\]

\[
p(r, f, c, w \mid d) = p(r \mid d)p(f \mid r)p(c, w \mid r, f, d)
\]
Generative Story

1. Record choice: choose a sequence of records \( r = (r_1, \ldots, r_{|r|}) \)

\[
p(r \mid d) = \prod_{i} p(r_i.t \mid r_{i-1}.t) \frac{1}{|s(r_i.t)|}
\]

2. Field choice: for each chosen record \( r_i \), select a sequence of fields \( f_i = (f_{i1}, \ldots, f_{i|f_i|}) \)

\[
p(f \mid r_i.t) = \prod_{k} p(r_i.f_k \mid r_i.f_{k-1})
\]

3. Word choice: for each chosen field \( f_{ik} \), choose a number \( c_{ik} > 0 \) uniformly, and generate a sequence of \( c_{ik} \) words.

\[
p(w \mid r_i, r_i.f_k, r_i.f_k.t, c_{ik}) = \prod_{j} p(w_j \mid r_i.t, r_i.f_k.v)
\]

\[
p(r, f, c, w \mid d) = p(r \mid d)p(f \mid r)p(c, w \mid r, f, d)
\]
Hierarchical Semi-Markov Model (HSMM)

EM Training: dynamic program similar to the inside-outside algorithm
Aligned Output

Records: temperature\_1  
Fields: max=70 
Text: High near 70.

Records: skyCover\_1 
Fields: percent=75-100 
Text: Cloudy,

Records: windDir\_1 windSpeed\_1 
Fields: mode=S mean=20 
Text: with a south around 20 mph.
Outline

- Problem Formulation
- Learning Alignments
- **Pipeline Approach**
- Joint Approaches
Traditional NLG Pipeline

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

Kim and Mooney (2010)
Angeli et al. (2010)
Angeli et al., EMNLP 2010
A Simple Domain-Independent Probabilistic Approach to Generation
for $i = 1, 2, \ldots$:

1. choose a record $r_i \in d$
Generative Story

for $i = 1, 2, \ldots$:

1. choose a record $r_i \in d$
2. if $r_i = \text{STOP}$: return
Generative Story

for $i = 1, 2, \ldots$:

1. **choose** a record $r_i \in d$
2. if $r_i = \text{STOP}$: return
3. **choose** a field $f_j \in r_i.t.f$
for $i = 1, 2, \ldots$:

1. **choose** a record $r_i \in d$
2. if $r_i = \text{STOP}$: return
3. **choose** a field $f_j \in r_i.t.f$
4. **choose** a template $T_k \in r_i.t.f_j.T$
for $i = 1, 2, \ldots$:

1. **choose** a record $r_i \in d$
2. if $r_i = \text{STOP}$: return
3. **choose** a field $f_j \in r_i.t.f$
4. **choose** a template $T_k \in r_i.t.f_j.T$
for $i = 1, 2, \ldots$:

1. **choose** a record $r_i \in d$
2. if $r_i = \text{STOP}$: return
3. **choose** a field $f_j \in r_i.t.f$
4. **choose** a template $T_k \in r_i.t.f_j.T$

Each decision is governed by a set of feature templates
# Feature Templates

<table>
<thead>
<tr>
<th>Record</th>
<th>Description</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>list of $k = 1, 2$ record types</td>
<td>$r_2.t = \text{temp} \land (r_1.t, r_0.t) = (\text{skyCover}, \text{START})$</td>
</tr>
<tr>
<td>R2</td>
<td>set of prev record types</td>
<td>$r_2.t = \text{temp} \land {r_1.t} = {\text{skyCover}}$</td>
</tr>
<tr>
<td>R3</td>
<td>record type already gen</td>
<td>$r_2.t = \text{temp} \land r_j.t \neq \text{temp}, \forall j &lt; 2$</td>
</tr>
<tr>
<td>R4</td>
<td>field values</td>
<td>$r_2.t = \text{temp} \land r_2.v[\text{min}] = 10, r_2.v[\text{max}] = 20$</td>
</tr>
<tr>
<td>R5</td>
<td>STOP under LM</td>
<td>$r_3.t = \text{STOP} \times p_{LM}(\text{STOP}</td>
</tr>
</tbody>
</table>
## Feature Templates

<table>
<thead>
<tr>
<th>Record</th>
<th>( R_1 )</th>
<th>list of ( k = 1, 2 ) record types</th>
<th>( r_2.t = \text{temp} \land (r_1.t, r_0.t) = (\text{skyCover}, \text{START}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( R_2 )</td>
<td>set of prev record types</td>
<td>( r_2.t = \text{temp} \land {r_1.t} = {\text{skyCover}} )</td>
</tr>
<tr>
<td></td>
<td>( R_3 )</td>
<td>record type already gen</td>
<td>( r_2.t = \text{temp} \land r_j.t \neq \text{temp}, \forall j &lt; 2 )</td>
</tr>
<tr>
<td></td>
<td>( R_4 )</td>
<td>field values</td>
<td>( r_2.t = \text{temp} \land r_2.v[\text{min}] = 10, r_2.v[\text{max}] = 20 )</td>
</tr>
<tr>
<td></td>
<td>( R_5 )</td>
<td>STOP under LM</td>
<td>( r_3.t = \text{STOP} \times p_L M(\text{STOP}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>( F_1 )</th>
<th>field set</th>
<th>( f_2 = {\text{time, min, mean, max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( F_2 )</td>
<td>field values</td>
<td>( f_2 = {\text{min, max}} \land f_2.v[\text{min}] = 10, \ldots )</td>
</tr>
</tbody>
</table>
# Feature Templates

<table>
<thead>
<tr>
<th>Record</th>
<th>R1</th>
<th>list of $k = 1, 2$ record types</th>
<th>$r_2.t = \text{temp} \land (r_1.t, r_0.t) = (\text{skyCover}, \text{START})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R2</td>
<td>set of prev record types</td>
<td>$r_2.t = \text{temp} \land {r_1.t} = {\text{skyCover}}$</td>
</tr>
<tr>
<td></td>
<td>R3</td>
<td>record type already gen</td>
<td>$r_2.t = \text{temp} \land r_j.t \neq \text{temp}, \forall j &lt; 2$</td>
</tr>
<tr>
<td></td>
<td>R4</td>
<td>field values</td>
<td>$r_2.t = \text{temp} \land r_2.v[\text{min}] = 10, r_2.v[\text{max}] = 20$</td>
</tr>
<tr>
<td></td>
<td>R5</td>
<td>STOP under LM</td>
<td>$r_3.t = \text{STOP} \times p_{LM}(\text{STOP}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>F1</th>
<th>field set</th>
<th>$f_2 = {\text{time, min, mean, max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F2</td>
<td>field values</td>
<td>$f_2 = {\text{min, max}} \land f_2.v[\text{min}] = 10, \ldots$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Template</th>
<th>W1</th>
<th>base/coarse</th>
<th>$B(T_2) = \langle \text{with a low around [min]} \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W2</td>
<td>field values</td>
<td>$C(T_2) = \langle \text{with a [time] around [min]} \rangle$</td>
</tr>
<tr>
<td></td>
<td>W3</td>
<td>$1_{st}$ word of T under LM</td>
<td>$p_{LM}(\text{with}</td>
</tr>
</tbody>
</table>
# Feature Templates

<table>
<thead>
<tr>
<th>Record</th>
<th>R1</th>
<th>list of $k = 1, 2$ record types</th>
<th>$r_2.t = \text{temp} \land (r_1.t, r_0.t) = (\text{skyCover}, \text{START})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R2</td>
<td>set of prev record types</td>
<td>$r_2.t = \text{temp} \land {r_1.t} = {\text{skyCover}}$</td>
</tr>
<tr>
<td></td>
<td>R3</td>
<td>record type already gen</td>
<td>$r_2.t = \text{temp} \land r_j.t \neq \text{temp}, \forall j &lt; 2$</td>
</tr>
<tr>
<td></td>
<td>R4</td>
<td>field values</td>
<td>$r_2.t = \text{temp} \land r_2.v[\text{min}] = 10, r_2.v[\text{max}] = 20$</td>
</tr>
<tr>
<td></td>
<td>R5</td>
<td>STOP under LM</td>
<td>$r_3.t = \text{STOP} \times p_{LM}(\text{STOP}</td>
</tr>
<tr>
<td>Field</td>
<td>F1</td>
<td>field set</td>
<td>$f_2 = {\text{time, min, mean, max}}$</td>
</tr>
<tr>
<td></td>
<td>F2</td>
<td>field values</td>
<td>$f_2 = {\text{min, max}} \land f_2.v[\text{min}] = 10, \ldots$</td>
</tr>
<tr>
<td>Template</td>
<td>W1</td>
<td>base/coarse</td>
<td>$B(T_2) = \langle \text{with a low around [min]} \rangle$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$C(T_2) = \langle \text{with a [time] around [min]} \rangle$</td>
</tr>
<tr>
<td></td>
<td>W2</td>
<td>field values</td>
<td>$p_{LM}(\text{with}</td>
</tr>
<tr>
<td></td>
<td>W3</td>
<td>1st word of T under LM</td>
<td></td>
</tr>
</tbody>
</table>

\[
p(c|d; \theta) = \prod_{j=1}^{|c|} p(c_j|c_{<j}; \theta)
\]
### Feature Templates

<table>
<thead>
<tr>
<th>Record</th>
<th>R1</th>
<th>list of $k = 1, 2$ record types</th>
<th>$r_2.t = \text{temp} \land (r_1.t, r_0.t) = (\text{skyCover}, \text{START})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R2</td>
<td>set of prev record types</td>
<td>$r_2.t = \text{temp} \land {r_1.t} = {\text{skyCover}}$</td>
</tr>
<tr>
<td></td>
<td>R3</td>
<td>record type already gen</td>
<td>$r_2.t = \text{temp} \land r_j.t \neq \text{temp}, \forall j &lt; 2$</td>
</tr>
<tr>
<td></td>
<td>R4</td>
<td>field values</td>
<td>$r_2.t = \text{temp} \land r_2.v[\text{min}] = 10, r_2.v[\text{max}] = 20$</td>
</tr>
<tr>
<td></td>
<td>R5</td>
<td>STOP under LM</td>
<td>$r_3.t = \text{STOP} \times p_{LM}(\text{STOP}</td>
</tr>
<tr>
<td>Field</td>
<td>F1</td>
<td>field set</td>
<td>$f_2 = {\text{time, min, mean, max}}$</td>
</tr>
<tr>
<td></td>
<td>F2</td>
<td>field values</td>
<td>$f_2 = {\text{min, max}} \land f_2.v[\text{min}] = 10, \ldots$</td>
</tr>
<tr>
<td>Template</td>
<td>W1</td>
<td>base/coarse</td>
<td>$B(T_2) = \langle\text{with a low around } [\text{min}]\rangle$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$C(T_2) = \langle\text{with a [time] around } [\text{min}]\rangle$</td>
</tr>
<tr>
<td></td>
<td>W2</td>
<td>field values</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W3</td>
<td>1st word of T under LM</td>
<td>$p_{LM}(\text{with}</td>
</tr>
</tbody>
</table>

\[
p(c|d; \theta) = \prod_{j=1}^{\lfloor c \rfloor} p(c_j|c_{<j}; \theta)
\]

L-BFGS learning: Use Liang et al. (2009) alignments to compute features
Decoding

\[ \hat{c}_j = \arg \max_{c_j} p(c_j | c_{<j}; \theta) \]

- Greedy search: choose the best decision \( \hat{c}_j \) until the \texttt{STOP} record is drawn.
Decoding

\[ \hat{c}_j = \arg \max_{c_j} p(c_j | c_{<j}; \theta) \]

- Greedy search: choose the best decision \( \hat{c}_j \) until the STOP record is drawn
- Alternatively, sample from the distribution \( p(c_j | c_{<j}; \theta) \);
Decoding

\[ \hat{c}_{j} = \arg \max_{c_{j}} p(c_{j}|c_{<j}; \theta) \]

- Greedy search: choose the best decision \( \hat{c}_{j} \) until the \texttt{STOP} record is drawn
- Alternatively, sample from the distribution \( p(c_{j}|c_{<j}; \theta) \);
- Viterbi search over \( \arg \max_{c_{j}} p(c_{j}|d; \theta) \)
Conclusions

- Generation recast into a generative story
- Ensemble of local decisions
- Discriminatively trained end-to-end generation system
Conclusions

- Generation recast into a generative story
- Ensemble of local decisions
- Discriminatively trained end-to-end generation system
- How about we model generation \textit{jointly} and learn \textit{without} supervision?
Outline

- Problem Formulation
- Learning Alignments
- Pipeline Approach
- Joint Approaches
Traditional NLG Pipeline

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

Communicative Goal

- Kim and Mooney (2010)
- Angeli et al. (2010)
- Konstas and Lapata (2012a, 2012b, 2013b)
Konstas and Lapata, NAACL 2012
Unsupervised Concept-to-text Generation with Hypergraphs

Konstas and Lapata, JAIR 2013
A Global Model for Concept-to-Text Generation
Joint Approaches

Grammar-based Generation

Grammar

<table>
<thead>
<tr>
<th>Rain Chance</th>
<th>Time Mode</th>
<th>06-21 Def</th>
<th>06-09 Lkly</th>
<th>06-13 Def</th>
<th>13-21 Def</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thunder Chance</td>
<td>Time Mode</td>
<td>06-21 Def</td>
<td>06-09 Chc</td>
<td>06-13 Def</td>
<td>13-21 Def</td>
</tr>
<tr>
<td>Temperature</td>
<td>Time Min</td>
<td>Mean</td>
<td>Max</td>
<td>06-21 52</td>
<td>61</td>
</tr>
<tr>
<td>Sky Cover</td>
<td>Time Percent (%)</td>
<td>06-21 75-100</td>
<td>06-09 75-100</td>
<td>06-13 50-75</td>
<td>09-21 75-100</td>
</tr>
<tr>
<td>Wind Direction</td>
<td>Time Mode</td>
<td>06-21 S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind Speed</td>
<td>Time Min</td>
<td>Mean</td>
<td>Max</td>
<td>06-21 11</td>
<td>22</td>
</tr>
<tr>
<td>Gust</td>
<td>Time Min</td>
<td>Mean</td>
<td>Max</td>
<td>06-21 0</td>
<td>20</td>
</tr>
<tr>
<td>Precipitation Potential</td>
<td>Time Min</td>
<td>Mean</td>
<td>Max</td>
<td>06-21 26</td>
<td>81</td>
</tr>
</tbody>
</table>
\( S \rightarrow R(start) \)
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

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

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. $FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j)$

$FS(wSpeed_1, min) \rightarrow F(wSpeed_1, max)FS(wSpeed_1, max)$
Grammar

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

F(gust_1, min) → W(gust_1, mean)F(gust_1, mean)
Joint Approaches
Grammar-based Generation

Grammar

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

\(W(skyCover_1, \%) \rightarrow \text{cloudy [\%}.v = '75-100']\)
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. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f)$
5. $W(r, r.f) \rightarrow \alpha \mid g(f.v)$

**EM Training**: dynamic program similar to the inside-outside algorithm
Decoding

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

\[
W_{0,1}(\text{skyCover}_1.t,\%)
\]

\[\epsilon\]

\[
(\text{mostly} ; \text{RB} \\
\text{cloudy} ; \text{JJ} \\
\text{sunny} ; \text{JJ} \\
\ldots)
\]
Decoding

\[
\left(\text{mostly cloudy } \star \text{ the morning ; JJ} \right) \left(\text{mostly cloudy } \star \text{ after 11am ; JJ} \right) \left(\text{mostly cloudy } \star \text{ then becoming ; JJ} \right) \ldots
\]

\[
\left(\text{mostly cloudy ; RB} \right. \left(\text{mostly clouds ; NNS} \right. \left(\text{cloudy ; ; JJ} \right) \ldots
\]

\[
\text{W}_{0.1}(\text{skyCover}_t.\text{t,} \%) \quad \text{W}_{0.2}(\text{skyCover}_t.\text{t,} \%) \quad \text{W}_{4.5}(\text{skyCover}_t.\text{t,} \text{time})
\]

\[
\left(\text{mostly ; RB} \right. \left(\text{cloudy ; JJ} \right. \left(\text{sunny ; JJ} \right) \ldots
\]

\[
\left(\text{morning ; NN} \right. \left(\text{11am ; NN} \right. \left(\text{after ; PREP} \right. \ldots
\]
Decoding

Joint Approaches

Grammar-based Generation

Konstas (ILCC)  Concept-to-Text Generation  21 July 2015  28 / 56
Joint Approaches

Decoding

```
(mostly cloudy, the morning, JJ)
(mostly cloudy, after 11am, JJ)
(mostly cloudy, then becoming, JJ)
...
```

```
(mostly cloudy, RB)
(mostly clouds, NNS)
(cloudy, ;, JJ)
...
```

```
W_{0,1}(skyCover_1.t, %)
```

```
F_{0,2}(skyCover_1.t, %)
```

```
W_{1,2}(skyCover_1.t, %)
```

```
FS_{0,5}(skyCover_1.t, start)
```

```
W_{4,5}(skyCover_1.t, time)
```

```
(morning, NN)
(11am, NN)
(after, PREP)
...
```
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, k-BEST-LM, k-BEST-LM-DMV
- Angeli et al. (2010)
Results: Automatic Evaluation

**RoboCup**

- **Base**: 10.79
- **Angeli**: 28.7
- **k-lm**: 30.9
- **k-lm-dmv**: 29.73

**WeatherGov**

- **Base**: 8.64
- **Angeli**: 38.4
- **k-lm**: 33.7
- **k-lm-dmv**: 34.18

**Atis**

- **Base**: 11.85
- **Angeli**: 26.77
- **k-lm**: 29.3
- **k-lm-dmv**: 30.37

**WinHelp**

- **Base**: 16.02
- **Angeli**: 32.21
- **k-lm**: 38.26
- **k-lm-dmv**: 39.03
Results: Human Evaluation (Fluency)

**RoboCup**
- Base: 2.47
- $k$-lm-dmv: 4.31
- Angeli: 4.03
- Human: 4.47

**WeatherGov**
- Base: 1.82
- $k$-lm-dmv: 3.92
- Angeli: 4.26
- Human: 4.61

**Atis**
- Base: 2.4
- $k$-lm-dmv: 4.01
- Angeli: 3.56
- Human: 4.1

**WinHelp**
- Base: 2.57
- $k$-lm-dmv: 3.41
- Angeli: 3.57
- Human: 4.15
### 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>Min</strong></td>
<td><strong>Mean</strong></td>
</tr>
<tr>
<td>06:00-21:00</td>
<td>30</td>
<td>38</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>Min</strong></td>
<td><strong>Mean</strong></td>
</tr>
<tr>
<td>06:00-21:00</td>
<td>6</td>
<td>6</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

**Input:**

<table>
<thead>
<tr>
<th>Flight</th>
<th>Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>day</td>
</tr>
<tr>
<td>milwaukee</td>
<td>dep/ar/ret</td>
</tr>
<tr>
<td>to</td>
<td>saturday</td>
</tr>
</tbody>
</table>

**Search**

<table>
<thead>
<tr>
<th>type</th>
<th>what</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>flight</td>
</tr>
</tbody>
</table>

**$k$-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
Joint Approaches

Results

Dependency Output

ATIS

ROOT

on

show

me

the

flights

from

Milwaukee

to

Phoenix

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?
Traditional NLG Pipeline

Input Data

Content Planning

Content Selection

Document Planning

Sentence Planning

Surface Realisation

Text

Communicative Goal

Kim and Mooney (2010)
Angeli et al. (2010)
Konstas and Lapata (2012a, 2012b, 2013a)
Joint Approaches
Inducing Document Planning

Traditional NLG Pipeline

Input Data

Communicative Goal

Content Planning

Content Selection

Document Planning

Sentence Planning

Surface Realisation

Text

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

Inducing Document Plans for Concept-to-text Generation, EMNLP 2013
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
Planning with Record Sequences

Key idea: Grammar on sequences of record types

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.
Key idea: Grammar on sequences of record types

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. desktop | start | start-target
   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.
Planning with Record Sequences

Key idea: Grammar on sequences of record types

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. \texttt{desktop | start | start-target} \\
   \texttt{Click start, point to settings, and then click control panel.} || \texttt{window-target} \\
   \texttt{Double-click users and passwords.} || \texttt{contextMenu | action-contextMenu} \\
   \texttt{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 \textit{within} and \textit{among} sentences
Extended 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. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f)$
5. $W(r, r.f) \rightarrow \alpha \mid g(f.v)$
Joint Approaches
Planning with Record Sequences

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. $FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j) \mid F(r, r.f_j)$
5. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f) \mid W(r, r.f)$
6. $W(r, r.f) \rightarrow \alpha \mid g(f.v) \mid gen\_str(f.v, i)$
Extended Grammar

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

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

5 \[ F(r, r.f) \rightarrow \text{W}(r, r.f) \cdot F(r, r.f) \mid \text{W}(r, r.f) \]

6 \[ \text{W}(r, r.f) \rightarrow \alpha \mid \text{g}(f.v) \mid \text{gen_str}(f.v, i) \]

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. \[ \text{FS}(r, r.f_i) \rightarrow \text{F}(r, r.f_j) \text{FS}(r, r.f_j) \mid \text{F}(r, r.f_j) \]
5. \[ \text{F}(r, r.f) \rightarrow \text{W}(r, r.f) \text{F}(r, r.f) \mid \text{W}(r, r.f) \]
6. \[ \text{W}(r, r.f) \rightarrow \alpha \mid g(f.v) \mid \text{gen}_\text{str}(f.v, i) \]

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>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>contextMenu</th>
<th>action-contextMenu</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the advanced tab,</td>
<td>click advanced.</td>
</tr>
</tbody>
</table>

Liang et al. (2009)

\[
\text{desktop} \text{ start} \text{ start-target} \parallel \text{window-target} \\
\text{contextMenu} \text{ action-contextMenu} \\
\text{On the advanced tab} \text{ click advanced.}
\]

\[
\downarrow
\]

\[
\left[ \text{desktop start start-target} \parallel \text{window-target} \parallel \text{contextMenu action-contextMenu} \parallel \right]
\]
### Grammar Extraction

Liang et al. (2009)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>desktop</td>
<td>start</td>
<td>start-target</td>
<td>window-target</td>
</tr>
<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>
<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>

\[
\text{D} \left[ \text{desktop start start-target} \parallel \text{window-target} \parallel \text{contextMenu action-contMenu} \parallel \right]
\]

- \(\text{SENT(desk, start, start-target)}\)
- \(\text{SENT(win-target)}\)
- \(\text{SENT(contMenu, action-contMenu)}\)
- \(\text{R(desk)}\)
- \(\text{R(start)}\)
- \(\text{R(start-target)}\)
- \(\text{R(win-target)}\)
- \(\text{R(contMenu)}\)
- \(\text{R(action-contMenu)}\)
Liang et al. (2009)

\[
\text{desktop} \quad \text{start} \quad \text{start-target} \quad \text{window-target}
\]

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

Double-click users and passwords.

On the advanced tab, click advanced.

[ desktop start start-target || window-target || contextMenu action-contMenu || ]
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)

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)

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]

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

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

<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>

- contextMenu
- On the advanced tab,  
  - action-contextMenu  
    - 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)

```latex
\begin{align*}
\text{[Click start.]}^\text{desktop} & \text{[point to settings, ]}^\text{start} \text{[and then click control panel.]}^\text{start-target} \\
\text{[Double-click users and passwords.]}^\text{window-target} & \text{[On the advanced tab,] }^\text{contextMenu} \\
\text{[click advanced.] }^\text{action-contextMenu}
\end{align*}
```
Grammar Extraction

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

On the advanced tab, click advanced.

Double-click users and passwords.

Feng and Hirst (2012)
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)
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

- \textsc{WeatherGov} : weather reports [4 sents, 345 words]  
  (Liang et al., 2009)

- \textsc{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 (BLEU-4)

![Bar chart showing BLEU-4 scores for different systems and datasets.]

- **WEATHERGOV**
  - ANGELI: 38.4
  - K&L: 33.7
  - $G_{RSE}$: 35.6
  - $G_{RST}$: 36.54

- **WINHELP**
  - ANGELI: 32.21
  - K&L: 38.26
  - $G_{RSE}$: 40.92
  - $G_{RST}$: 40.65
### Results: Human Evaluation (Coherence)

**WEATHERGOV**

<table>
<thead>
<tr>
<th></th>
<th>ANGELE</th>
<th>K&amp;L</th>
<th>G_{RSE}</th>
<th>G_{RST}</th>
<th>HUMAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGELE</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>ANGELE</th>
<th>K&amp;L</th>
<th>G_{RSE}</th>
<th>G_{RST}</th>
<th>HUMAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGELE</td>
<td>2.97</td>
<td>2.93</td>
<td>3.35</td>
<td>3.22</td>
<td>4.25</td>
</tr>
<tr>
<td></td>
<td>Output</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>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>
</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, and then click ok.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HUMAN</strong></td>
<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>
</tr>
</tbody>
</table>
Conclusions

- 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 ($G_{RST}$)
Recap

- Recast NLG into a generative model
  - History-based local decisions - Add more features
  - Hierarchical joint model - Add more layers
- Learn parameters from (un)-annotated data - multiple domains
- **Decoding**: greedy search, $k$-best Viterbi search
Where do we go from here?

- Generate from more open-ended formalisms: AMR
- More challenging factual domains: biographies from Wikipedia
- More sophisticated sentence planning: aggregation, referring expressions
- More engineering: address sparsity, with Deep Learning
- Apply document planning grammars to summarisation
Thank you

Questions?
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 $R$. 
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 $R$. 

\begin{center}
\begin{tikzpicture}
\node (t) at (0,0) [draw, circle] {t};
\node (a) at (1,1) [draw, circle] {a};
\node (b) at (1,-1) [draw, circle] {b};
\draw (t) edge [->] node [below] {f(e)} (b);
\draw (t) edge [->] node [right] {} (a);
\end{tikzpicture}
\end{center}
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 \( R \).
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 $R$. 

![Hypergraph Diagram]
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 $H : \langle N, E, t, R \rangle$

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

$$R(r_i.t) \rightarrow FS(r_j, start) R(r_j.t)$$
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 $H : \langle N, E, t, R \rangle$

\[ f(e) = f(FS_{5,7}(flight_1.t, start)) \otimes f(R_{7,9}(flight_1.t)) \otimes w(R(\text{search}_1.t) \rightarrow FS(\text{flight}_1, start) \ R(\text{flight}_1.t)) \]

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

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

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

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

$\text{R(search}_1.t) \rightarrow \text{FS(flight}_1.t, \text{start}) \land \text{R(flight}_1.t)$

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

$\text{R(r}_i.t) \rightarrow \text{FS(r}_j, \text{start}) \land \text{R(r}_j.t)$
Hypergraph Example
Hypergraph Example
Hypergraph Example

S\(_0\), R\(_0\) (start)

FS\(_0\), 1 (skyCover\(_1\), start)

F\(_0\), 1 (skyCover\(_1\), %)
W\(_0\), 1 (skyCover\(_1\), %)

sunny

with

Konstas (ILCC)
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