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

Ioannis Konstas

Institute for Language, Cognition and Computation
University of Edinburgh

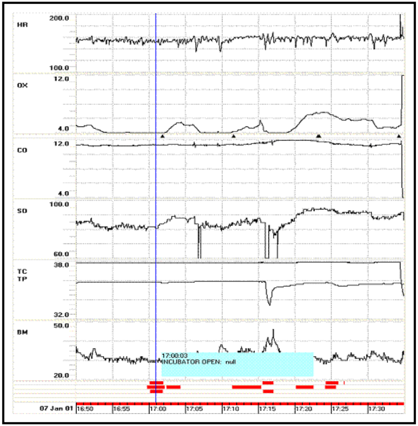
Aberdeen, NLG Summer School
21 July 2015

Introduction

Introduction

- ROOT
  - exhibit
    - statue
      - kouros
        - exhibit7
        - exhibit12
      - complex-statue
      - portrait
      - imperial-portrait
    - coin
    - jewel
    - relief
    - ...
  - a-location
    - museum
      - museum1
    - archeological-site
  - ...

Introduction



Sensor Data

Full Descriptor	**Time**
SETTING;VENTIL;FiO2 (36%)	10.30
MEDICATION;Morphine	10.44
ACTION;CARE;TURN/ CHANGE POSITION;SUPINE	10.46-10.47
ACTION;RESP;HAND BABY	10.47-10.51
SETTING;VENTIL;FiO2 (60%)	10.47
ACTION;RESP;INTUBATE	10.51-10.52

Action Records

Introduction

Concept-to-text generation refers to the task of automatically producing textual output from nonlinguistic input (Reiter and Dale, 2000)

Introduction

Concept-to-text generation refers to the task of automatically producing textual output from nonlinguistic input (Reiter and Dale, 2000)

Wind Chill

Time	Min	Mean	Max
06-21	0	0	0

Temperature

Time	Min	Mean	Max
06-21	52	61	70

Wind Speed

Time	Min	Mean	Max
06-21	11	22	29

Wind Direction

Time	Mode
06-21	S

Gust

Time	Min	Mean	Max
06-21	0	20	39

Precipitation Potential

Time	Min	Mean	Max
06-21	26	81	100

Sky Cover

Time	Percent (%)
06-21	75-100
06-09	75-100
06-13	50-75
09-21	75-100
13-21	75-100

Rain Chance

Time	Mode
06-21	Def
06-09	Lkly
06-13	Def
09-21	Def
13-21	Def

Snow Chance

Time	Mode
06-21	–
06-09	–
06-13	–
09-21	–
13-21	–

Sleet Chance

Time	Mode
06-21	–
06-09	–
06-13	–
09-21	–
13-21	–

Freezing Rain Chance

Time	Mode
06-21	–
06-09	–
06-13	–
09-21	–
13-21	–

Thunder Chance

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

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%.

Introduction

Concept-to-text generation refers to the task of automatically producing textual output from nonlinguistic input (Reiter and Dale, 2000)

Desktop

Cmd	Name	Type
left-click	start	button

Start

Cmd	Name	Type
left-click	settings	button

Location

Name	Type
start menu	button
control panel	window

Start Target

Cmd	Name	Type
left-click	control panel	button

Navigate Window

Cmd	Name	Type
left-click	accounts and users	window

Context Menu

Cmd	Name	Type
left-click	advanced	tab

Action Context Menu

Cmd	Name	Type
left-click	advanced	button

Window Target

Cmd	Name	Type
double-click	users and passwords	item

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

Introduction

What we will look into today?

Introduction

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

Introduction

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 $r \in \mathbf{d}$ has a type $r.t$ and fields f
- Fields have values $f.v$ and types $f.t$ (integer, categorical, string)

Cloud Sky Cover	
Time	**Percent (%)**
06:00-09:00	25-50
09:00-12:00	50-75

mostly cloudy,

Input

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

Cloud Sky Cover	
Time	**Percent (%)**
06:00-09:00	25-50
09:00-12:00	50-75

mostly cloudy,

Input

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

Cloud Sky Cover	
Time	**Percent (%)**
06:00-09:00	25-50
09:00-12:00	50-75

mostly cloudy,

Input

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

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

mostly cloudy,

Input

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

Cloud Sky Cover	
Time	**Percent (%)**
06:00-09:00	25-50
09:00-12:00	50-75

mostly cloudy,

Key Idea

Temperature

Time	Min	Mean	Max
06:00-21:00	9	15	21

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

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

Wind Speed

Time	Min	Mean	Max
06:00-21:00	15	20	30

Wind Direction

Time	Mode
06:00-21:00	S

Key Idea

Temperature

Time	Min	Mean	Max
06:00-21:00	9	15	21

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

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

Wind Speed

Time	Min	Mean	Max
06:00-21:00	15	20	30

Wind Direction

Time	Mode
06:00-21:00	S

Key Idea

Temperature

Time	Min	Mean	Max
06:00-21:00	9	15	21

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

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

Wind Speed

Time	Min	Mean	Max
06:00-21:00	15	20	30

Wind Direction

Time	Mode
06:00-21:00	S

Key Idea

Temperature

Time	Min	Mean	Max
06:00-21:00	9	15	21

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

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

Wind Speed

Time	Min	Mean	Max
06:00-21:00	15	20	30

Wind Direction

Time	Mode
06:00-21:00	S

Key Idea

Temperature

Time	Min	Mean	Max
06:00-21:00	9	15	21

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

Wind Speed

Time	Min	Mean	Max
06:00-21:00	15	20	30

Wind Direction

Time	Mode
06:00-21:00	S

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

Key Idea

Temperature

Time	Min	Mean	Max
06:00-21:00	9	15	21

Cloud Sky Cover

Time	Percent (%)
06:00-09:00	25-50
09:00-12:00	50-75

Wind Speed

Time	Min	Mean	Max
06:00-21:00	15	20	30

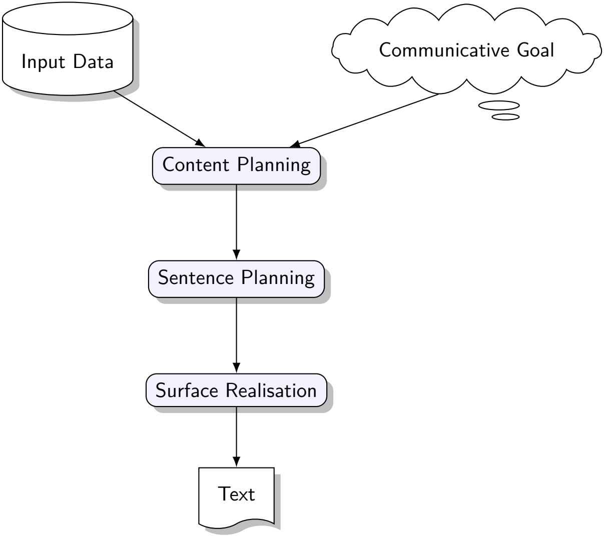
Wind Direction

Time	Mode
06:00-21:00	S

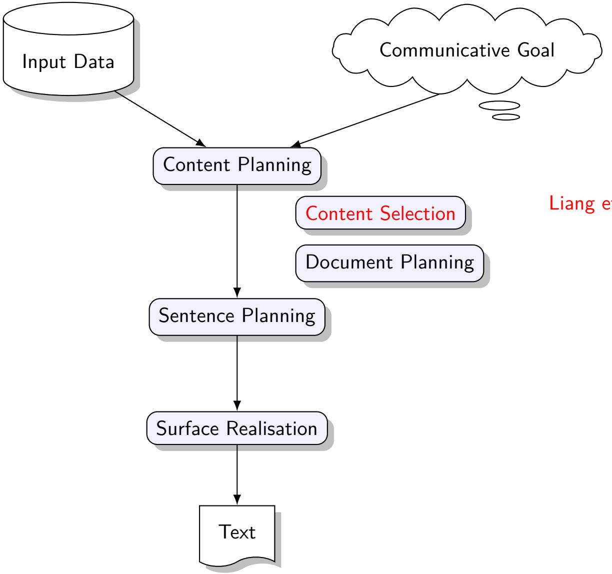
Cloudy, with a low around 10. South wind between 15 and 30 mph.

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

Traditional NLG Pipeline



Traditional NLG Pipeline



Liang et al. (2009)

Liang et al., ACL 2009

Learning Semantic Correspondences with Less Supervision

Alignment Task

Wind Chill

Time	Min	Mean	Max
06-21	0	0	0

Temperature

Time	Min	Mean	Max
06-21	52	61	70

Wind Speed

Time	Min	Mean	Max
06-21	11	22	29

Wind Direction

Time	Mode
06-21	S

Gust

Time	Min	Mean	Max
06-21	0	20	39

Precipitation Potential

Time	Min	Mean	Max
06-21	26	81	100

Sky Cover

Time	Percent (%)
06-21	75-100
06-09	75-100
06-13	50-75
09-21	75-100
13-21	75-100

Rain Chance

Time	Mode
06-21	Def
06-09	Lkly
06-13	Def
09-21	Def
13-21	Def

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%.

Snow Chance

Time	Mode
06-21	–
06-09	–
06-13	–
09-21	–
13-21	–

Thunder Chance

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

Freezing Rain Chance

Time	Mode
06-21	–
06-09	–
06-13	–
09-21	–
13-21	–

Sleet Chance

Time	Mode
06-21	–
06-09	–
06-13	–
09-21	–
13-21	–

Generative Story

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

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

$$p(\mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{w} | \mathbf{d}) = p(\mathbf{r}|\mathbf{d}) p(\mathbf{f}|\mathbf{r}) p(\mathbf{c}, \mathbf{w}|\mathbf{r}, \mathbf{f}, \mathbf{d})$$

Generative Story

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

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

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

$$p(\mathbf{f} \mid r_i.t) = \prod_{k}^{|r_i.\mathbf{f}|} p(r_i.f_k \mid r_i.f_{k-1})$$

$$p(\mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{w} | \mathbf{d}) = p(\mathbf{r}|\mathbf{d}) p(\mathbf{f}|\mathbf{r}) p(\mathbf{c}, \mathbf{w}|\mathbf{r}, \mathbf{f}, \mathbf{d})$$

Generative Story

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

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

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

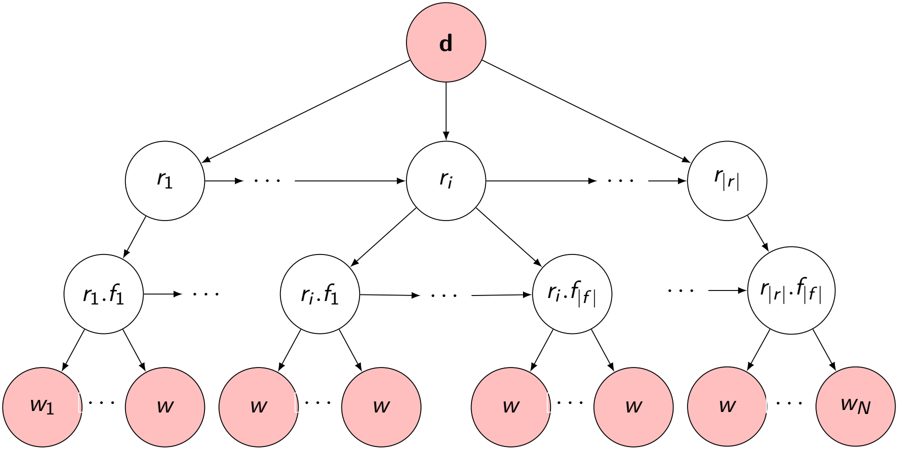
$$p(\mathbf{f} \mid r_i.t) = \prod_{k}^{|r_i.\mathbf{f}|} p(r_i.f_k \mid r_i.f_{k-1})$$

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

$$p(\mathbf{w} \mid r_i, r_i.f_k, r_i.f_k.t, c_{ik}) = \prod_{j}^{|\mathbf{w}|} p(w_j \mid r_i.t, r_i.f_k.v)$$

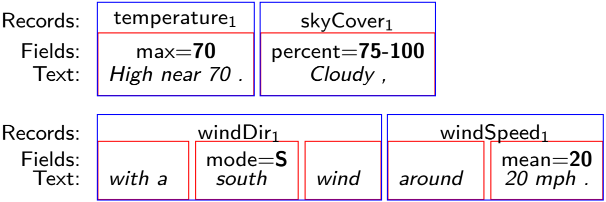
$$p(\mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{w} | \mathbf{d}) = p(\mathbf{r}|\mathbf{d}) p(\mathbf{f}|\mathbf{r}) p(\mathbf{c}, \mathbf{w}|\mathbf{r}, \mathbf{f}, \mathbf{d})$$

Hierarchical Semi-Markov Model (HSMM)



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

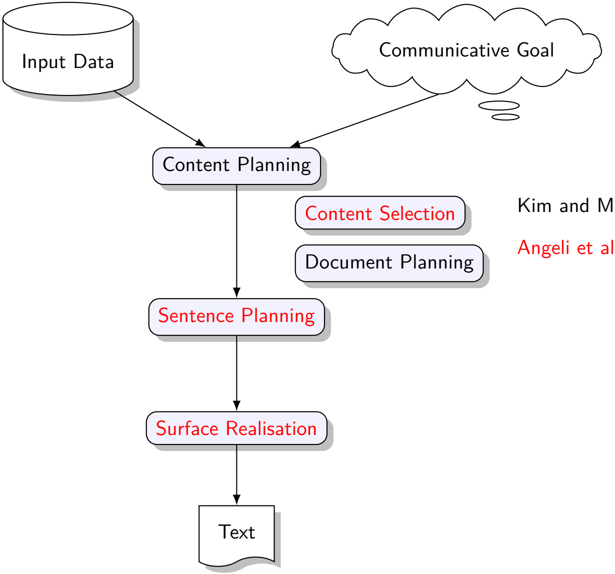
Aligned Output



Outline

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

Traditional NLG Pipeline



Kim and Mooney (2010)

Angeli et al. (2010)

Angeli et al., EMNLP 2010

A Simple Domain-Independent Probabilistic Approach to Generation

Generative Story

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

1. **choose** a record $r_i \in \mathbf{d}$

Generative Story

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

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

Generative Story

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

1. **choose** a record $r_i \in \mathbf{d}$
2. if $r_i = \textsc{stop}$: return
3. **choose** a field $f_j \in r_i.t.\mathbf{f}$

Generative Story

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

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

Generative Story

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

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

Generative Story

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

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

Each **decision** is governed by a set of feature templates

Feature Templates

Record	R1	list of $k = 1, 2$ record types	$r_2.t$=temp $\wedge$ $(r_1.t,\ r_0.t)$=(skyCover, START)
	R2	set of prev record types	$r_2.t$=temp $\wedge$ $\{r_1.t\}$=$\{$skyCover$\}$
	R3	record type already gen	$r_2.t$=temp $\wedge$ $r_j.t \neq$temp, $\forall j < 2$
	R4	field values	$r_2.t$=temp $\wedge$ r_2.v[min]=10, r_2.v[max]=20
	R5	STOP under LM	r_3.t=STOP $\times$ $p_{LM}($STOP$\mid$`degrees .`$)$

Feature Templates

Record	R1	list of $k = 1, 2$ record types	$r_2.t$=temp $\wedge$ $(r_1.t,\ r_0.t)$=(skyCover, START)
	R2	set of prev record types	$r_2.t$=temp $\wedge$ $\{r_1.t\}$=$\{$skyCover$\}$
	R3	record type already gen	$r_2.t$=temp $\wedge$ $r_j.t \neq$temp, $\forall j < 2$
	R4	field values	$r_2.t$=temp $\wedge$ r_2.v[min]=10, r_2.v[max]=20
	R5	STOP under LM	r_3.t=STOP $\times$ p_{LM}(STOP|degrees .)
Field	F1	field set	f_2= $\{$time, min, mean, max$\}$
	F2	field values	f_2= $\{$min, max$\}$ $\wedge$ f_2.v[min]=10, ...

Feature Templates

Record	R1	list of $k = 1, 2$ record types	$r_2.t$=temp $\wedge$ $(r_1.t,\ r_0.t)$=(skyCover, START)
	R2	set of prev record types	$r_2.t$=temp $\wedge$ $\{r_1.t\}$=$\{$skyCover$\}$
	R3	record type already gen	$r_2.t$=temp $\wedge$ $r_j.t \neq$temp, $\forall j < 2$
	R4	field values	$r_2.t$=temp $\wedge$ r_2.v[min]=10, r_2.v[max]=20
	R5	STOP under LM	r_3.t=STOP $\times$ p_{LM}(STOP|`degrees .`)
Field	F1	field set	f_2= {time, min, mean, max}
	F2	field values	f_2= {min, max} $\wedge$ f_2.v[min]=10, ...
Template	W1	base/coarse	B(T_2) = ⟨`with a low around [min]`⟩
			C(T_2) = ⟨`with a [time] around [min]`⟩
	W2	field values	
	W3	1_{st} word of T under LM	p_{LM}(`with`|`cloudy ,`)

Feature Templates

Record	R1	list of $k = 1, 2$ record types	$r_2.t$=temp $\wedge$ $(r_1.t,\ r_0.t)$=(skyCover, START)
	R2	set of prev record types	$r_2.t$=temp $\wedge$ $\{r_1.t\}$=$\{$skyCover$\}$
	R3	record type already gen	$r_2.t$=temp $\wedge$ $r_j.t \neq$temp, $\forall j < 2$
	R4	field values	$r_2.t$=temp $\wedge$ r_2.v[min]=10, r_2.v[max]=20
	R5	STOP under LM	r_3.t=STOP $\times$ p_{LM}(STOP|`degrees .`)
Field	F1	field set	f_2= {time, min, mean, max}
	F2	field values	f_2= {min, max} $\wedge$ f_2.v[min]=10, ...
Template	W1	base/coarse	B(T_2) = ⟨`with a low around [min]`⟩ C(T_2) = ⟨`with a [time] around [min]`⟩
	W2	field values	
	W3	1_{st} word of T under LM	p_{LM}(`with`|`cloudy ,`)

$$p(\mathbf{c}|\mathbf{d};\theta) = \prod_{j=1}^{|\mathbf{c}|} p(c_j|c_{<j};\theta)$$

Feature Templates

Record	R1	list of $k = 1, 2$ record types	$r_2.t$=temp $\wedge$ $(r_1.t,\ r_0.t)$=(skyCover, START)
	R2	set of prev record types	$r_2.t$=temp $\wedge$ $\{r_1.t\}$=$\{$skyCover$\}$
	R3	record type already gen	$r_2.t$=temp $\wedge$ $r_j.t \neq$temp, $\forall j < 2$
	R4	field values	$r_2.t$=temp $\wedge$ r_2.v[min]=10, r_2.v[max]=20
	R5	STOP under LM	r_3.t=STOP $\times$ $p_{LM}(\text{STOP}\mid\texttt{degrees .})$
Field	F1	field set	f_2= {time, min, mean, max}
	F2	field values	f_2= {min, max} $\wedge$ f_2.v[min]=10, ...
Template	W1	base/coarse	B(T_2) = ⟨`with a low around [min]`⟩
			C(T_2) = ⟨`with a [time] around [min]`⟩
	W2	field values	
	W3	1_{st} word of T under LM	$p_{LM}(\texttt{with}\mid\texttt{cloudy ,})$

$$p(\mathbf{c}|\mathbf{d};\theta) = \prod_{j=1}^{|\mathbf{c}|} p(c_j|c_{<j};\theta)$$

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

Decoding

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

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

Conclusions

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

Conclusions

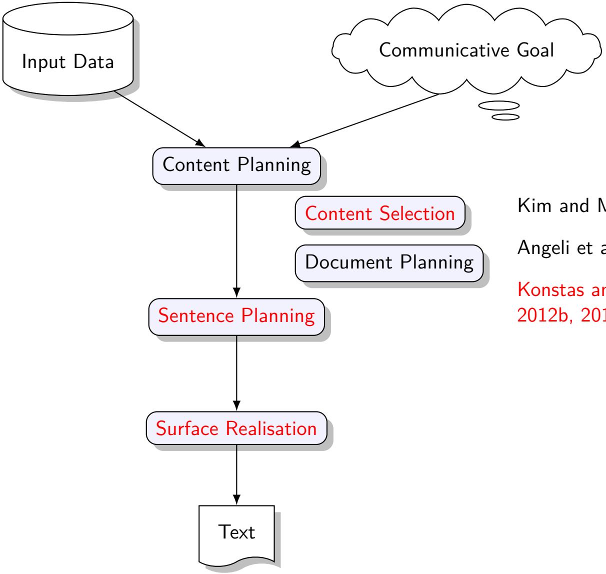
- Generation recast into a generative story
- Ensemble of local decisions
- Discriminatively trained end-to-end generation system

- How about we model generation jointly and learn without supervision?

Outline

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

Traditional NLG Pipeline



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

Grammar

Grammar

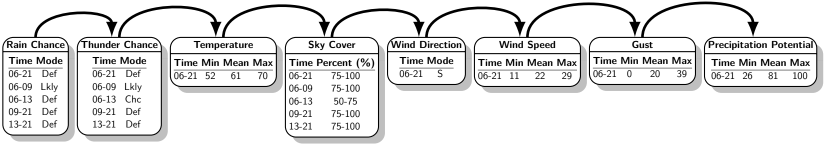
1. $\text{S} \rightarrow \text{R}(\textit{start})$

Grammar

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

$\text{R}(skyCover_1.t) \rightarrow \text{FS}(temperature_1, start)\text{R}(temperature_1.t)$

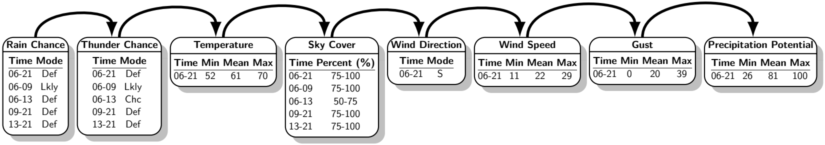
Grammar



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

$$\mathrm{R}(skyCover_1.t) \rightarrow \mathrm{FS}(temperature_1, start)\mathrm{R}(temperature_1.t)$$

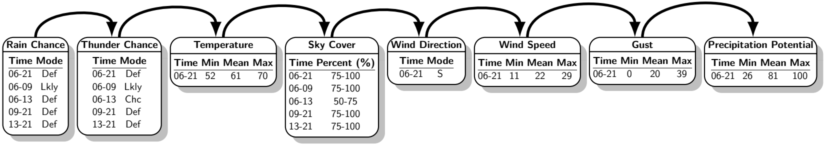
Grammar



1. $\mathrm{S} \rightarrow \mathrm{R}(start)$
2. $\mathrm{R}(r_i.t) \rightarrow \mathrm{FS}(r_j, start)\mathrm{R}(r_j.t) \mid \mathrm{FS}(r_j, start)$
3. $\mathrm{FS}(r, r.f_i) \rightarrow \mathrm{F}(r, r.f_j)\mathrm{FS}(r, r.f_j) \mid \mathrm{F}(r, r.f_j)$

$$\mathrm{FS}(wSpeed_1, min) \rightarrow \mathrm{F}(wSpeed_1, max)\mathrm{FS}(wSpeed_1, max)$$

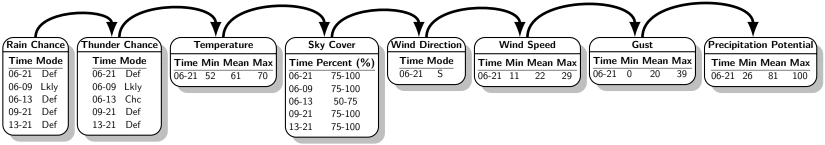
Grammar



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

$$\text{F}(gust_1, min) \rightarrow \text{W}(gust_1, mean)\text{F}(gust_1, mean)$$

Grammar



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

$\mathrm{W}(skyCover_1, \%) \rightarrow \text{cloudy } [\%.v = \text{'75–100'}]$

Grammar

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

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

Decoding

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

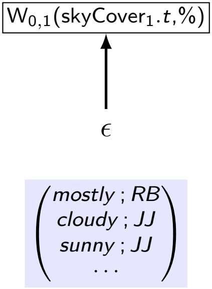
Decoding

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

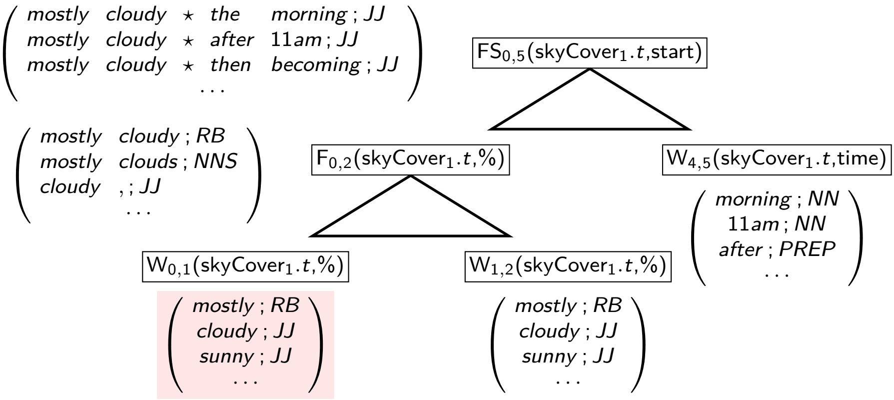
- Bottom-up Viterbi search
- Keep k-best derivations at each node, cube pruning (Chiang, 2007)
- $p(g)$ rescores derivations by linearly interpolating:
  - n-gram language model
  - dependency model (DMV; Klein and Manning, 2004)
- Implement using hypergraphs (Klein and Manning, 2001)

Decoding

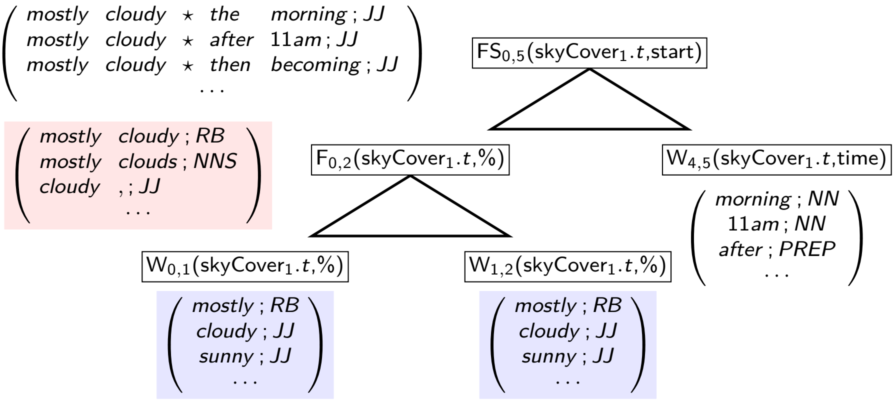
Leaf nodes ϵ emit a k-best list of words



Decoding

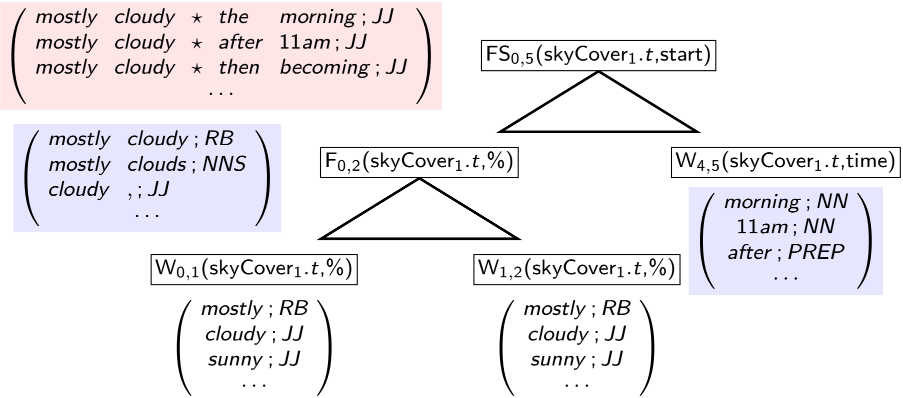

mostly cloudy ⋆ the morning ; JJ
mostly cloudy ⋆ after 11am ; JJ
mostly cloudy ⋆ then becoming ; JJ
...
FS0,5(skyCover1.t,start)
mostly cloudy ; RB
mostly clouds ; NNS
cloudy , ; JJ
...
F0,2(skyCover1.t,%)
W4,5(skyCover1.t,time)
morning ; NN
11am ; NN
after ; PREP
...
W0,1(skyCover1.t,%)
W1,2(skyCover1.t,%)
mostly ; RB
cloudy ; JJ
sunny ; JJ
...
mostly ; RB
cloudy ; JJ
sunny ; JJ
...

Decoding



mostly cloudy ⋆ the morning ; JJ
mostly cloudy ⋆ after 11am ; JJ
mostly cloudy ⋆ then becoming ; JJ
...
mostly cloudy ; RB
mostly clouds ; NNS
cloudy , ; JJ
...
$FS_{0,5}$(skyCover$_1$.t,start)
$F_{0,2}$(skyCover$_1$.t,%)
$W_{4,5}$(skyCover$_1$.t,time)
morning ; NN
11am ; NN
after ; PREP
...
$W_{0,1}$(skyCover$_1$.t,%)
$W_{1,2}$(skyCover$_1$.t,%)
mostly ; RB
cloudy ; JJ
sunny ; JJ
...
mostly ; RB
cloudy ; JJ
sunny ; JJ
...

Decoding


mostly cloudy ⋆ the morning ; JJ
mostly cloudy ⋆ after 11am ; JJ
mostly cloudy ⋆ then becoming ; JJ
. . .
mostly cloudy ; RB
mostly clouds ; NNS
cloudy , ; JJ
. . .
FS0,5(skyCover1.t,start)
F0,2(skyCover1.t,%)
W4,5(skyCover1.t,time)
morning ; NN
11am ; NN
after ; PREP
. . .
W0,1(skyCover1.t,%)
W1,2(skyCover1.t,%)
mostly ; RB
cloudy ; JJ
sunny ; JJ
. . .
mostly ; RB
cloudy ; JJ
sunny ; JJ
. . .

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)

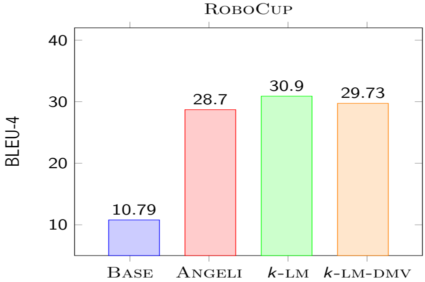
Evaluation

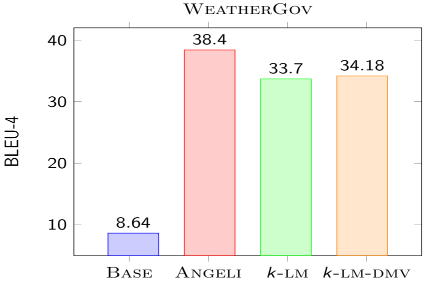
- Automatic evaluation: BLEU-4
- Human evaluation: Fluency, Semantic Correctness

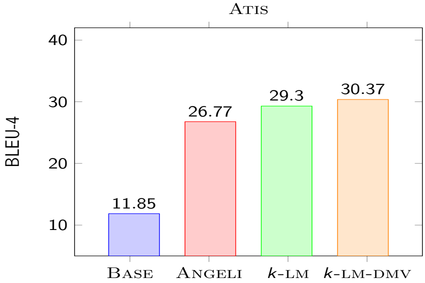
System Comparison

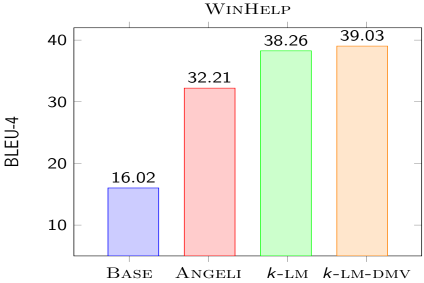
- 1−best, *k*-Best-lm, *k*-Best-lm-dmv
- Angeli et al. (2010)

Results: Automatic Evaluation

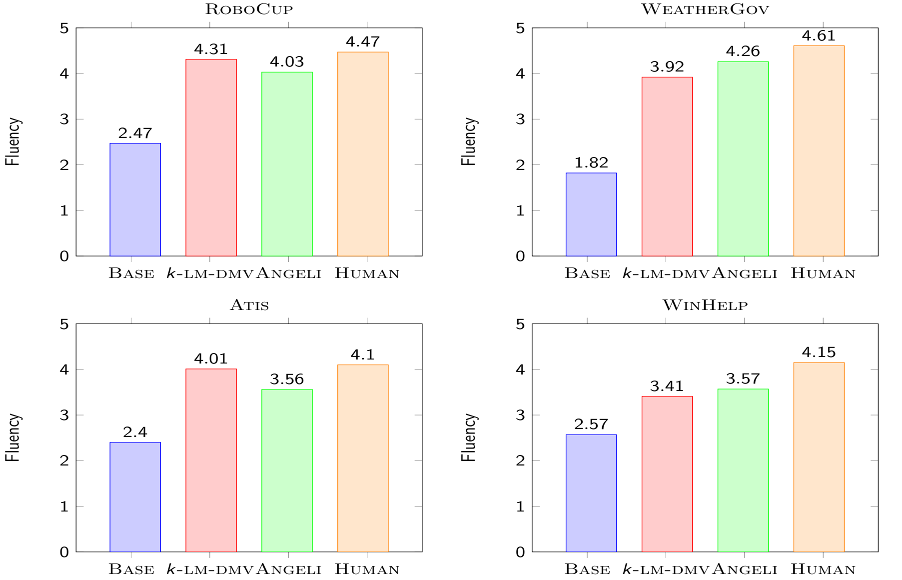








Results: Human Evaluation (Fluency)



Output

WeatherGov

Temperature

Time	Min	Mean	Max
06:00-21:00	30	38	44

Cloud Sky Cover

Time	Percent (%)
06:00-21:00	75-100

Chance of Rain

Time	Mode
06:00-11:00	Slight Chance

Wind Speed

Time	Min	Mean	Max
06:00-21:00	6	6	7

Wind Direction

Time	Mode
06:00-21:00	ENE

Precipitation Potential (%)

Time	Min	Mean	Max
06:00-21:00	9	20	35

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.

Output

ATIS

Input:

Flight	
from	**to**
milwaukee	phoenix

Day	
day	**dep/ar/ret**
saturday	departure

Search	
type	**what**
query	flight

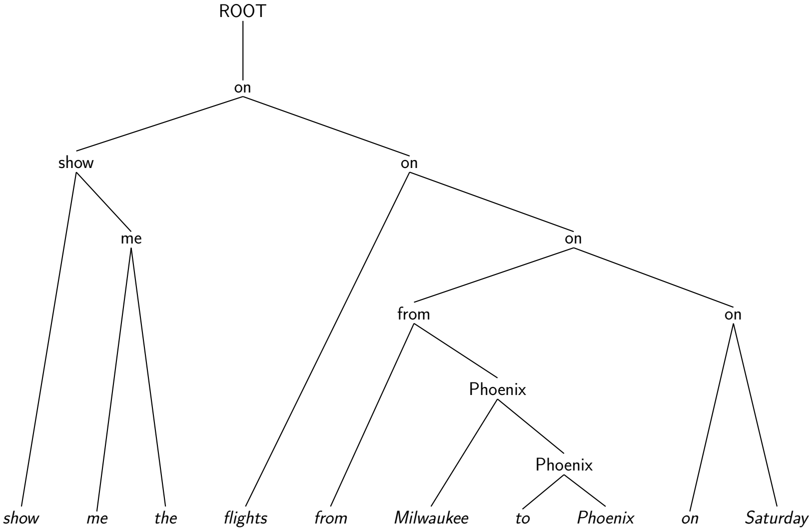
k-BEST: **What are the flights from Milwuakee to Phoenix on Saturday**

ANGELI : Show me the flights between Milwuakee and Phoenix on Saturday

HUMAN: Milwuakee to Phoenix on Saturday

Dependency Output

ATIS



Conclusions

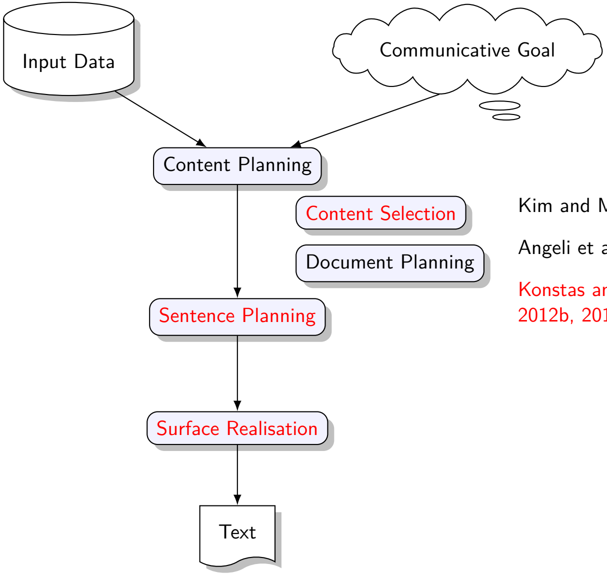
- Generation as parsing problem
- Unsupervised end-to-end generation system
- Performance comparable to state-of-the-art

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

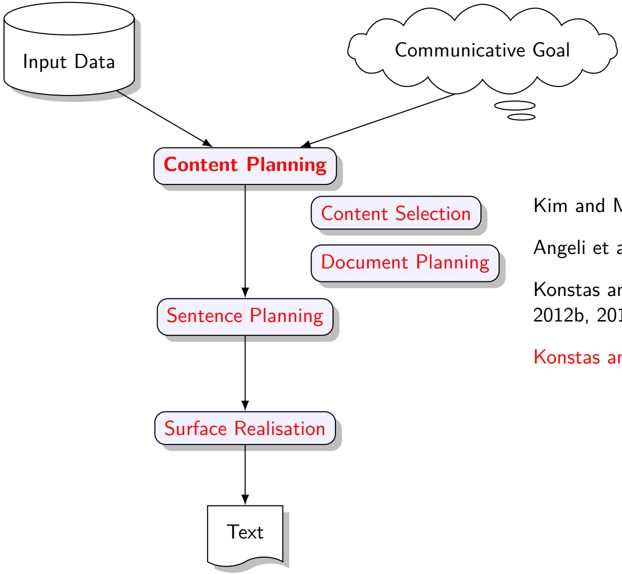


Kim and Mooney (2010)

Angeli et al. (2010)

Konstas and Lapata (2012a, 2012b, 2013a)

Traditional NLG Pipeline



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

Key Idea

Desktop

Cmd	Name	Type
left-click	start	button

Start

Cmd	Name	Type
left-click	settings	button

Location

Name	Type
start menu	button
control panel	window

Start Target

Cmd	Name	Type
left-click	control panel	button

Navigate Window

Cmd	Name	Type
left-click	accounts and users	window

Context Menu

Cmd	Name	Type
left-click	advanced	tab

Action Context Menu

Cmd	Name	Type
left-click	advanced	button

Window Target

Cmd	Name	Type
double-click	users and passwords	item

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

Double-click users and passwords.

On the advanced tab, click advanced.

Key Idea

Desktop

Cmd	Name	Type
left-click	start	button

Start

Cmd	Name	Type
left-click	settings	button

Location

Name	Type
start menu	button
control panel	window

Start Target

Cmd	Name	Type
left-click	control panel	button

Navigate Window

Cmd	Name	Type
left-click	accounts and users	window

Context Menu

Cmd	Name	Type
left-click	advanced	tab

Action Context Menu

Cmd	Name	Type
left-click	advanced	button

Window Target

Cmd	Name	Type
double-click	users and passwords	item

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

Key Idea

Desktop

Cmd	Name	Type
left-click	start	button

Start

Cmd	Name	Type
left-click	settings	button

Start Target

Cmd	Name	Type
left-click	control panel	button

Window Target

Cmd	Name	Type
double-click	users and passwords	item

Context Menu

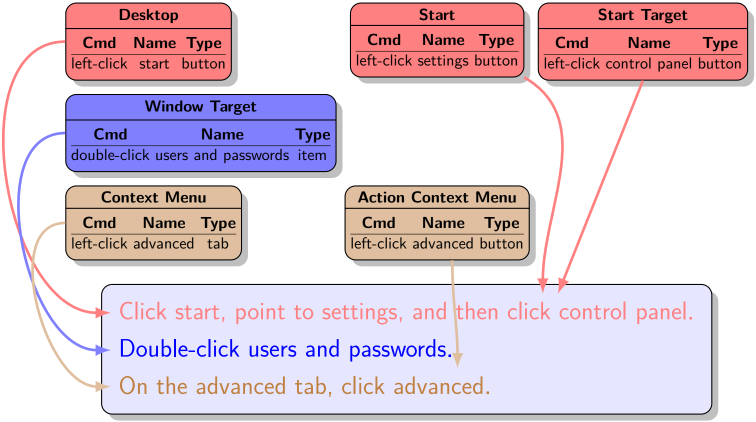
Cmd	Name	Type
left-click	advanced	tab

Action Context Menu

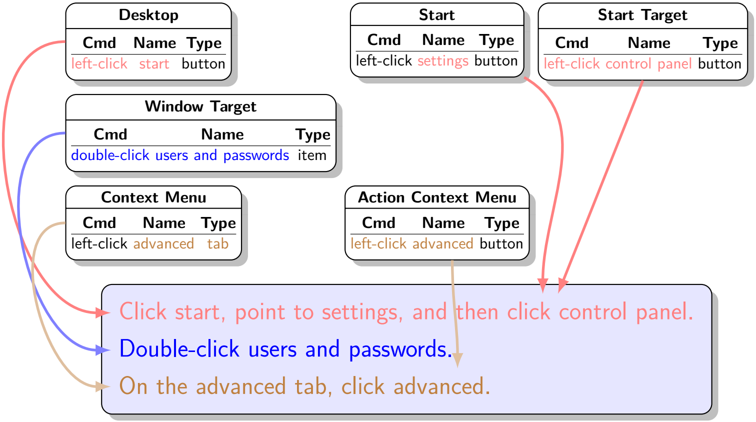
Cmd	Name	Type
left-click	advanced	button

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

Key Idea


Desktop
Cmd Name Type
left-click start button
Start
Cmd Name Type
left-click settings button
Start Target
Cmd Name Type
left-click control panel button
Window Target
Cmd Name Type
double-click users and passwords item
Context Menu
Cmd Name Type
left-click advanced tab
Action Context Menu
Cmd Name Type
left-click advanced button
Click start, point to settings, and then click control panel.
Double-click users and passwords.
On the advanced tab, click advanced.

Key Idea


Desktop
Cmd Name Type
left-click start button
Start
Cmd Name Type
left-click settings button
Start Target
Cmd Name Type
left-click control panel button
Window Target
Cmd Name Type
double-click users and passwords item
Context Menu
Cmd Name Type
left-click advanced tab
Action Context Menu
Cmd Name Type
left-click advanced button
Click start, point to settings, and then click control panel.
Double-click users and passwords.
On the advanced tab, click advanced.

Key Idea

Key Idea: Grammar-based document plans

Key Idea

Key Idea: Grammar-based document plans

- Re-use the generation model based on a PCFG grammar of input

Key Idea

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

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

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

Planning with Record Sequences

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.

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

   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. ‖

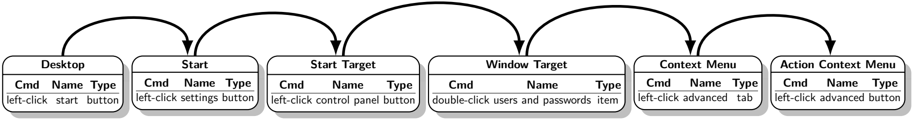
   Split a document into sentences, each terminated by a full-stop.

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

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

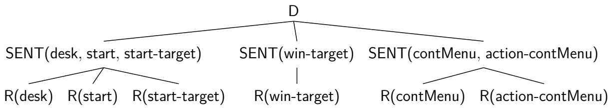
3. Goal: Learn patterns of record type sequences within and among sentences

Extended Grammar



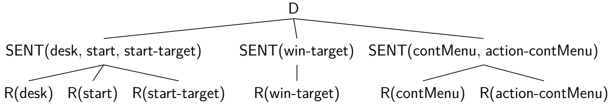
1. ~~$\mathrm{S} \rightarrow \mathrm{R}(start)$~~
2. ~~$\mathrm{R}(r_i.t) \rightarrow \mathrm{FS}(r_j, start)\mathrm{R}(r_j.t) \mid \mathrm{FS}(r_j, start)$~~
3. $\mathrm{FS}(r, r.f_i) \rightarrow \mathrm{F}(r, r.f_j)\mathrm{FS}(r, r.f_j) \mid \mathrm{F}(r, r.f_j)$
4. $\mathrm{F}(r, r.f) \rightarrow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

Extended Grammar



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

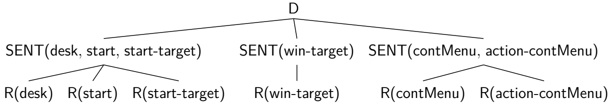
Extended Grammar



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

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

Extended Grammar



1. $D \rightarrow SENT(t_i, \ldots, t_j) \ldots SENT(t_l, \ldots, t_m)$
2. $SENT(t_i, \ldots, t_j) \rightarrow R(r_a.t_i) \ldots R(r_k.t_j) \cdot$
3. $\text{R}(r_i.t) \rightarrow \text{FS}(r_j, start)$
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. $\text{W}(r, r.f) \rightarrow \alpha \mid \text{g}(f.v) \mid \text{gen\_str}(f.v, i)$

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

Plan B: Extract grammar rules from training data

Grammar Extraction

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

contextMenu	action-contextMenu
On the advanced tab ,	click advanced.

Liang et al. (2009)

Grammar Extraction

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

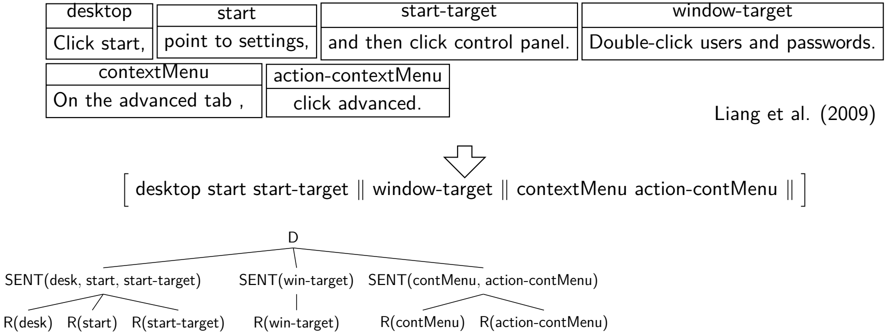
contextMenu	action-contextMenu
On the advanced tab ,	click advanced.

Liang et al. (2009)

⇩

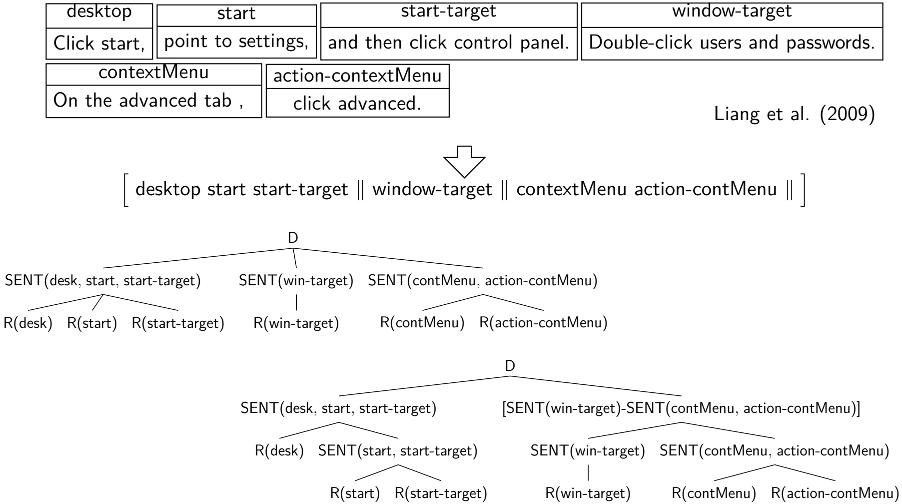
[desktop start start-target ‖ window-target ‖ contextMenu action-contMenu ‖]

Grammar Extraction



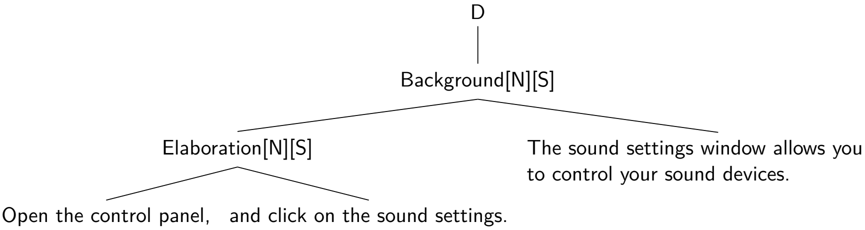
Liang et al. (2009)

Grammar Extraction



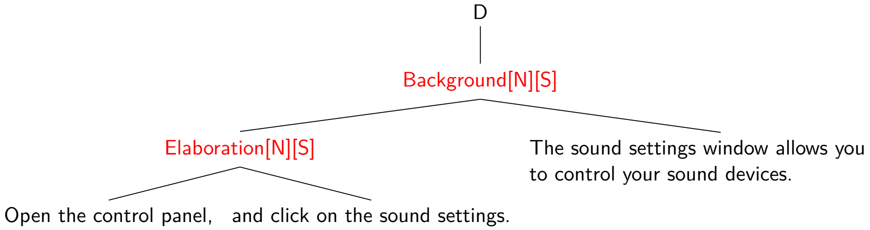
Planning with Rhetorical Structure Theory

RST (Mann and Thompson, 1988)



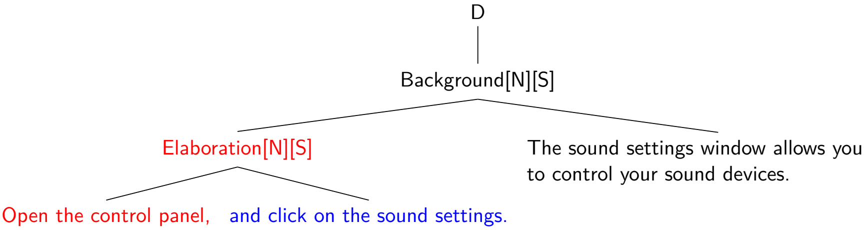
Planning with Rhetorical Structure Theory

RST (Mann and Thompson, 1988)



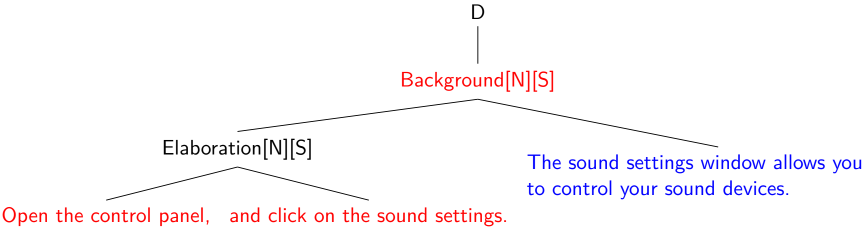
Planning with Rhetorical Structure Theory

RST (Mann and Thompson, 1988)



Planning with Rhetorical Structure Theory

RST (Mann and Thompson, 1988)



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

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

contextMenu	action-contextMenu
On the advanced tab ,	click advanced.

Liang et al. (2009)

Grammar Extraction

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

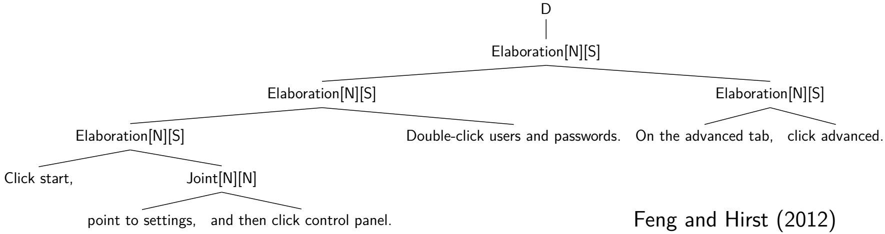
contextMenu	action-contextMenu
On the advanced tab ,	click advanced.

Liang et al. (2009)

⇩

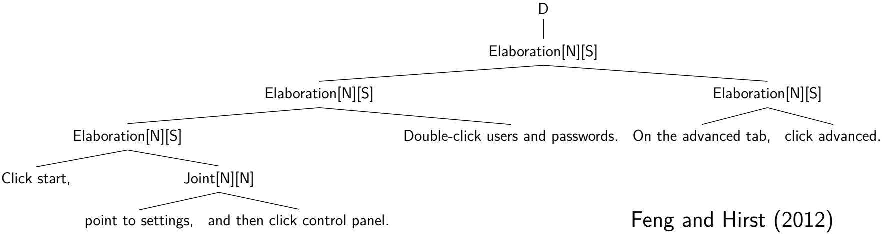
[Click start,]^{desktop} [point to settings,]^{start} [and then click control panel.]$^{start-target}$ [Double-click users and passwords.]$^{window-target}$ [On the advanced tab,]^{contextMenu} [click advanced.]$^{action-contextMenu}$

Grammar Extraction

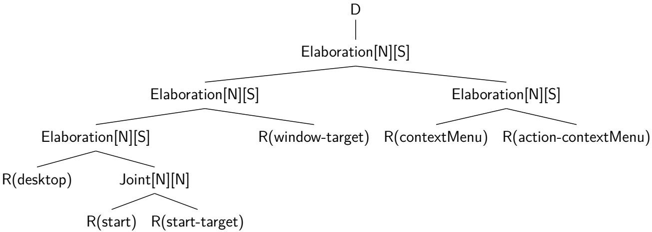


Feng and Hirst (2012)

Grammar Extraction



Feng and Hirst (2012)



Extended Grammar

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

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)

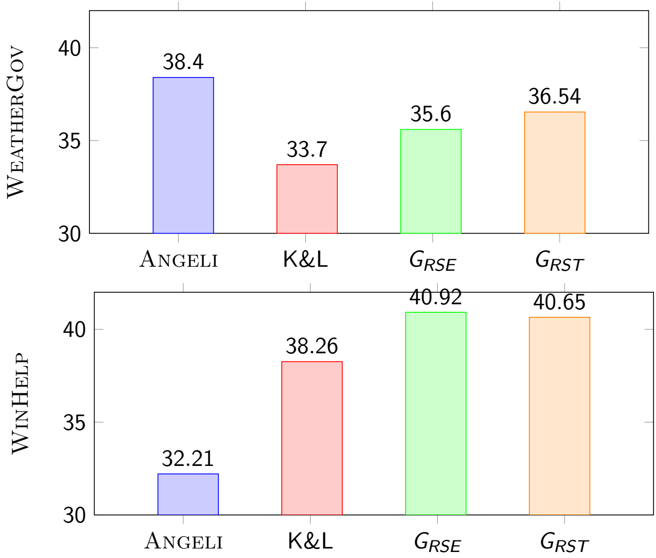
Evaluation

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

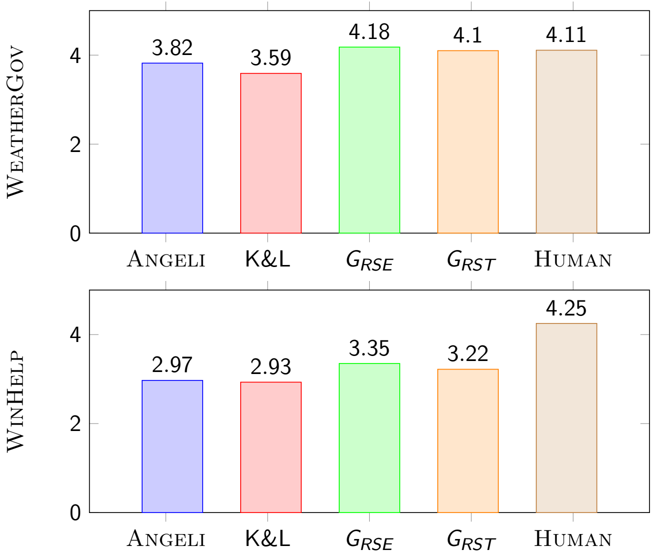
System Comparison

- G_{RSE}, G_{RST}
- Konstas and Lapata (2012a)
- Angeli et al. (2010)

Results: Automatic Evaluation (BLEU-4)



Results: Human Evaluation (Coherence)



Output

G_{RSE}	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. Click install, and then click add. Click network monitor driver, and then click ok.
K&L	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.
Human	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. Click install, click protocol, and then click add. Click network monitor driver, and then click ok.

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

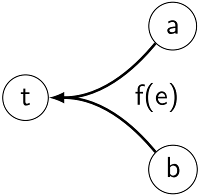
Thank you

Questions ?

Hypergraphs

Definition

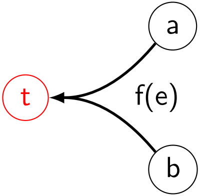
An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$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



Hypergraphs

Definition

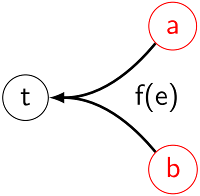
An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$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



Hypergraphs

Definition

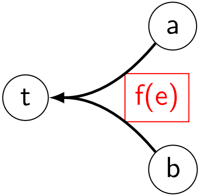
An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$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



Hypergraphs

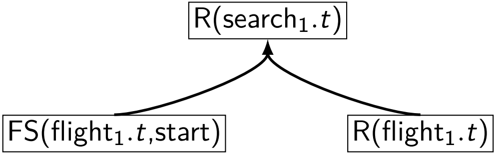
Definition

An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ 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 $\mathbf{R}_{|T(e)|}$ to $\mathbf{R}$.



Hypergraph Construction

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

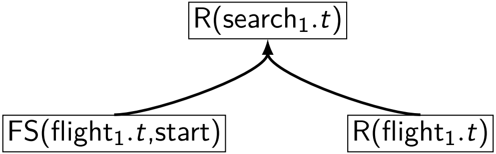


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\ w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

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

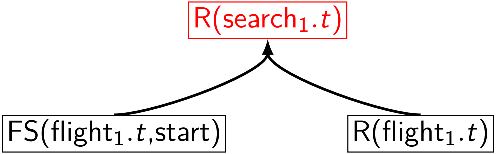


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes$$
$$w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

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

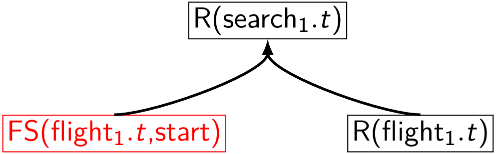


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\ w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

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

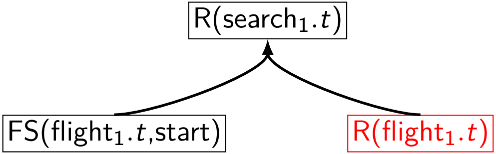


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\ w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

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

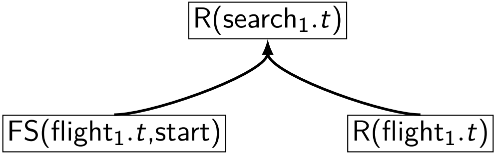


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\ w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

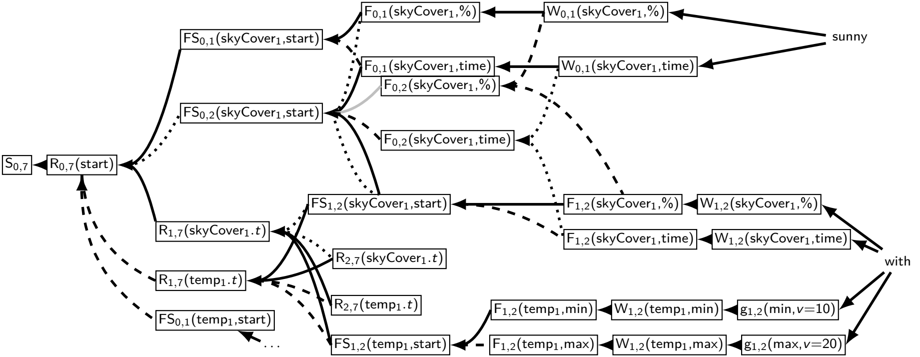
Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, \mathbf{R} \rangle$



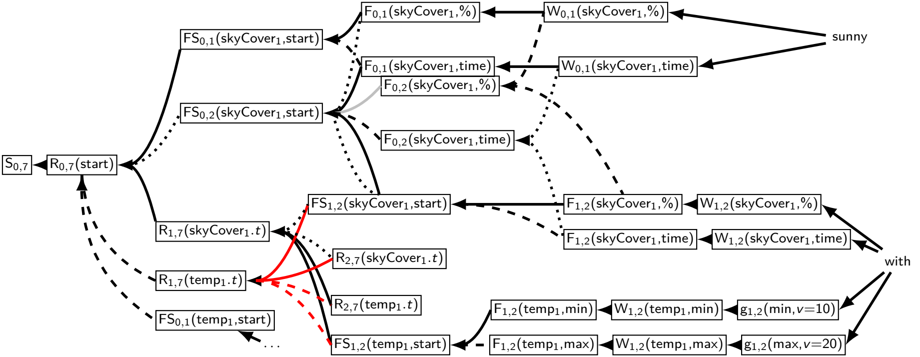
$$f(e) = f(\text{FS}_{5,7}(\text{flight}_1.t, start)) \otimes f(\text{R}_{7,9}(\text{flight}_1.t)) \otimes$$
$$w(\text{R}(\text{search}_1.t) \rightarrow \text{FS}(\text{flight}_1, start)\ \text{R}(\text{flight}_1.t))$$

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

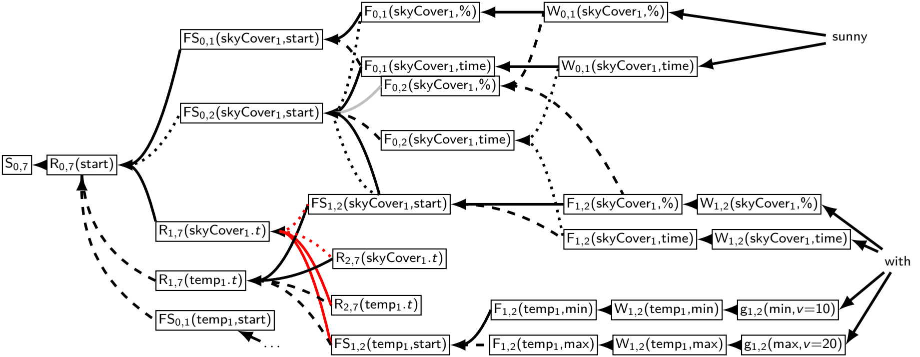
Hypergraph Example



Hypergraph Example



Hypergraph Example



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