Joint Models for Concept-to-Text Generation

Ioannis Konstas

Institute for Language, Cognition and Computation
University of Edinburgh

University of Washington, Seattle
22 October 2013

Outline

1. **Grammar-based Generation**
   - Unsupervised Concept-to-text Generation w/ Hypergraphs, NAACL 2012
   - A Global Model for Concept-to-Text Generation, JAIR 2013, In press.
2. **Inducing Document Plans**
   - Inducing Document Plans for Concept-to-text Generation, ACL 2013
3. **Discriminative Reranking: An exploratory study**
   - Concept-to-text Generation via Discriminative Reranking, ACL 2012

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

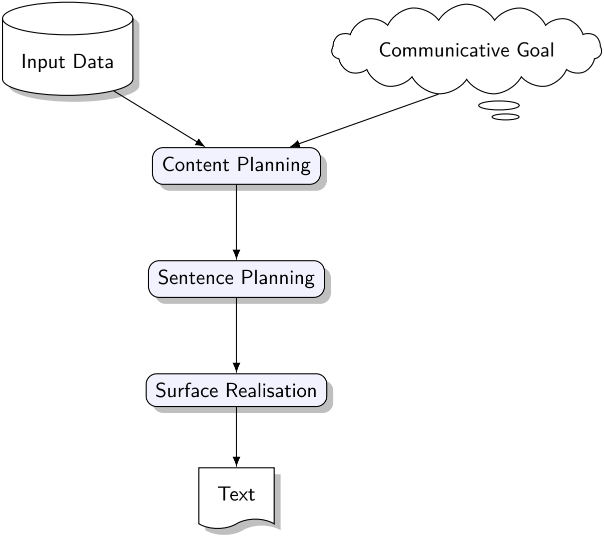
Cmd	Name	Type
double-click	users and passwords	item

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

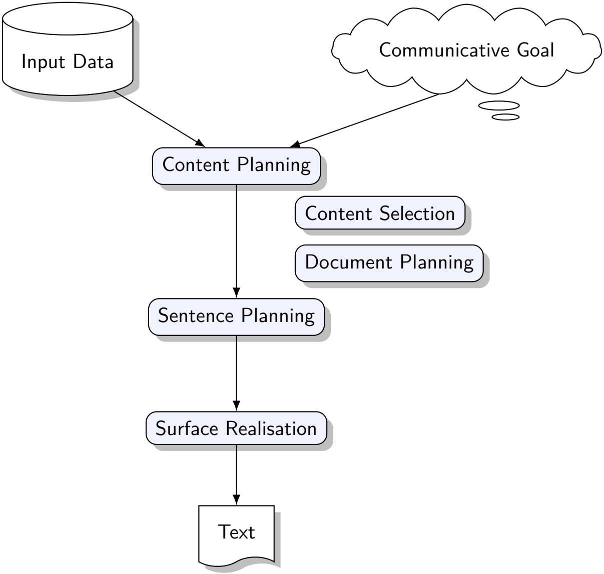
Double-click users and passwords.

On the advanced tab, click advanced.

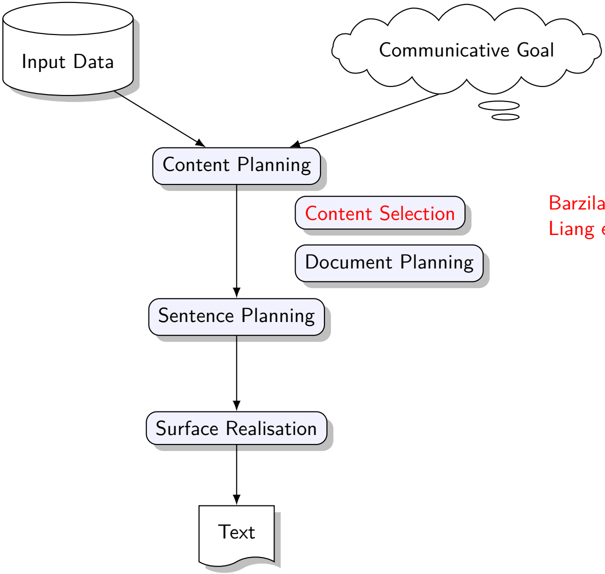
Traditional NLG Pipeline



Traditional NLG Pipeline

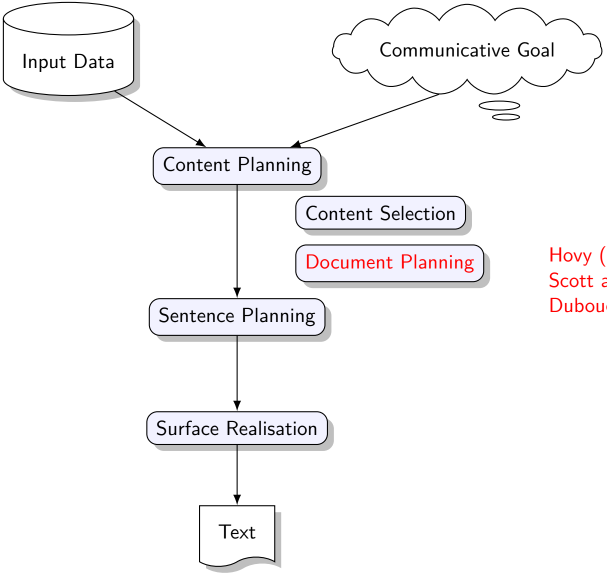


Traditional NLG Pipeline



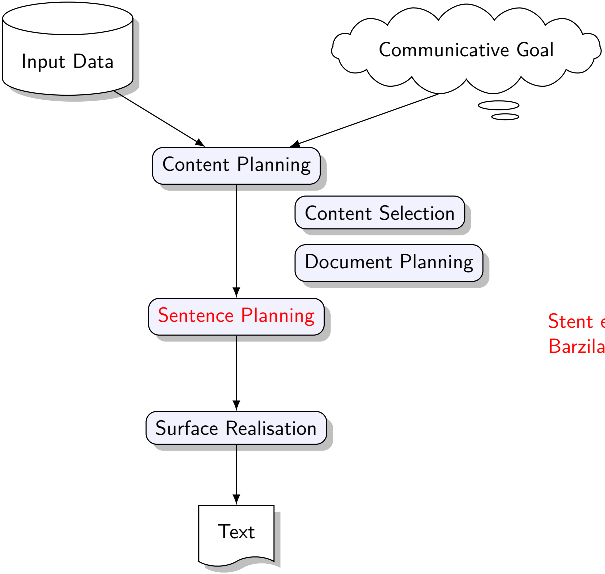
Barzilay and Lapata (2005)
Liang et al. (2009)

Traditional NLG Pipeline



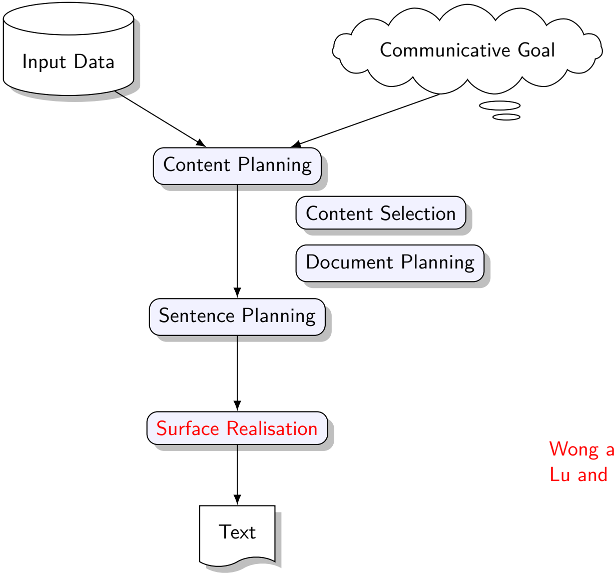
Hovy (1993)
Scott and de Souza (1990)
Duboue and Mckeown (2002)

Traditional NLG Pipeline



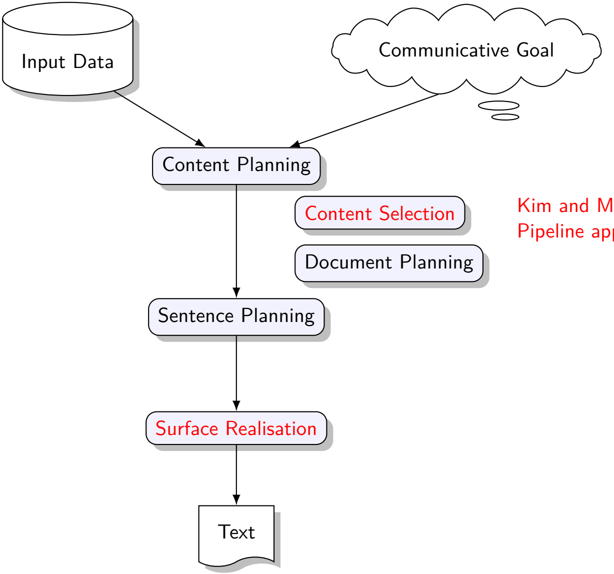
Stent et al. (2004)
Barzilay and Lapata (2006)

Traditional NLG Pipeline



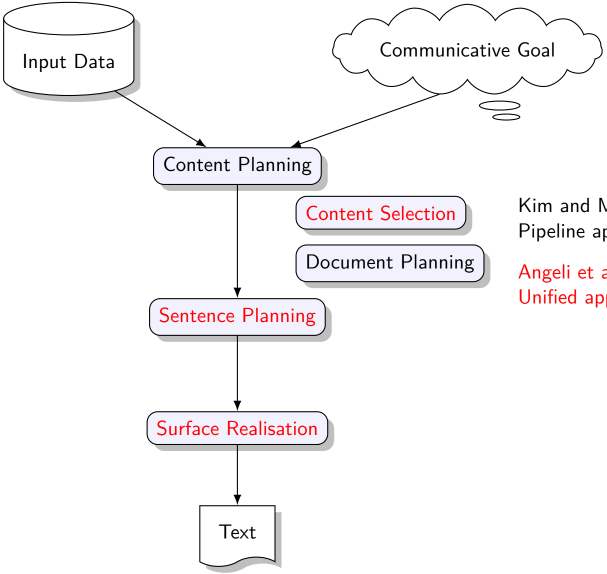
Wong and Mooney (2007)
Lu and Ng (2011)

Traditional NLG Pipeline



Kim and Mooney (2010)
Pipeline approach

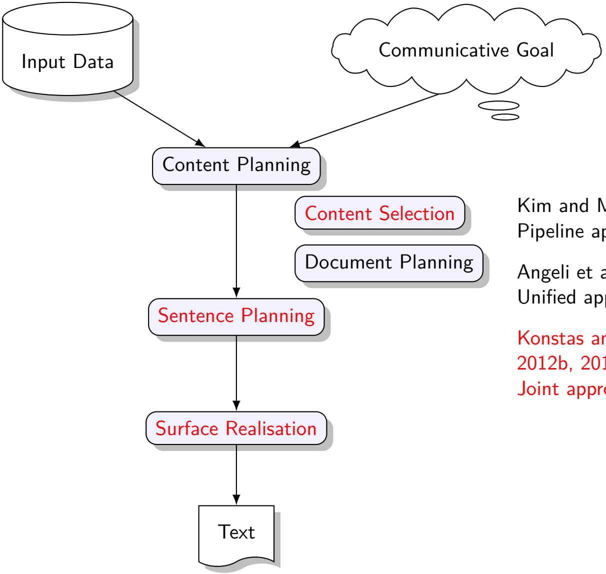
Traditional NLG Pipeline



Kim and Mooney (2010)
Pipeline approach

Angeli et al. (2010)
Unified approach

Traditional NLG Pipeline



Kim and Mooney (2010)
Pipeline approach

Angeli et al. (2010)
Unified approach

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

Input

- Input: database records **d**
- Output: words **w** corresponding to some records of **d**
- Each record $r \in \mathbf{d}$ 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,

Konstas and Lapata, NAACL 2012

Unsupervised Concept-to-text Generation with Hypergraphs

Konstas and Lapata, JAIR 2013. In press

A Global Model for Concept-to-Text Generation, JAIR 2013, In press.

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.

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.

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.

Grammar

Grammar

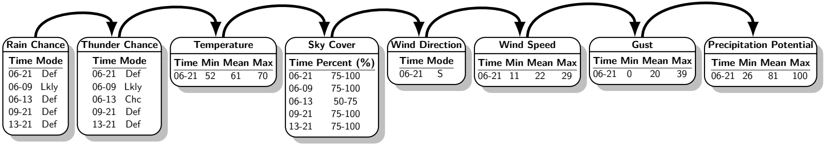
1. $\mathrm{S} \rightarrow \mathrm{R}(\mathit{start})$

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

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

Grammar

Rain Chance

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

Thunder Chance

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

Temperature

Time	Min	Mean	Max
06-21	52	61	70

Sky Cover

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

Wind Direction

Time	Mode
06-21	S

Wind Speed

Time	Min	Mean	Max
06-21	11	22	29

Gust

Time	Min	Mean	Max
06-21	0	20	39

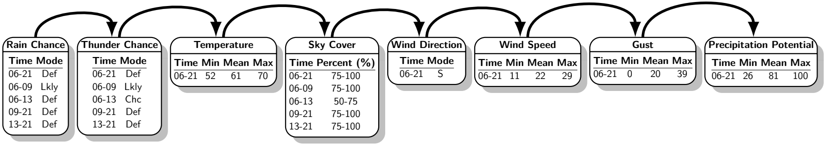
Precipitation Potential

Time	Min	Mean	Max
06-21	26	81	100

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

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

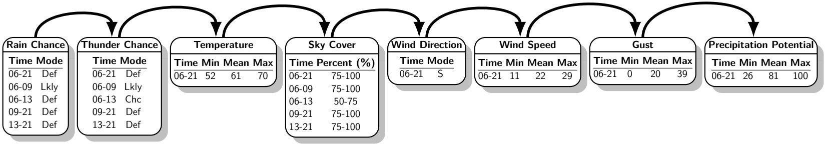
Grammar



1. $\mathrm{S} \rightarrow \mathrm{R}(start)$
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. $\mathrm{F}(r, r.f) \rightarrow \mathrm{W}(r, r.f)\mathrm{F}(r, r.f) \mid \mathrm{W}(r, r.f)$

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

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

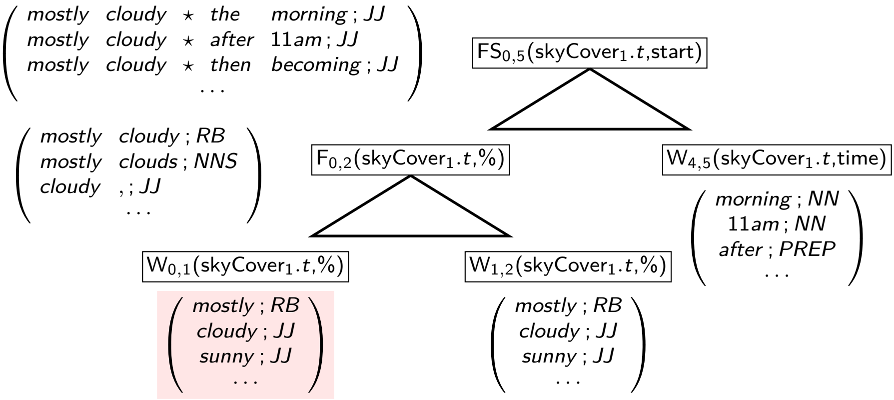
$$\boxed{W_{0,1}(\text{skyCover}_1.t,\%)}$$

$$\uparrow$$

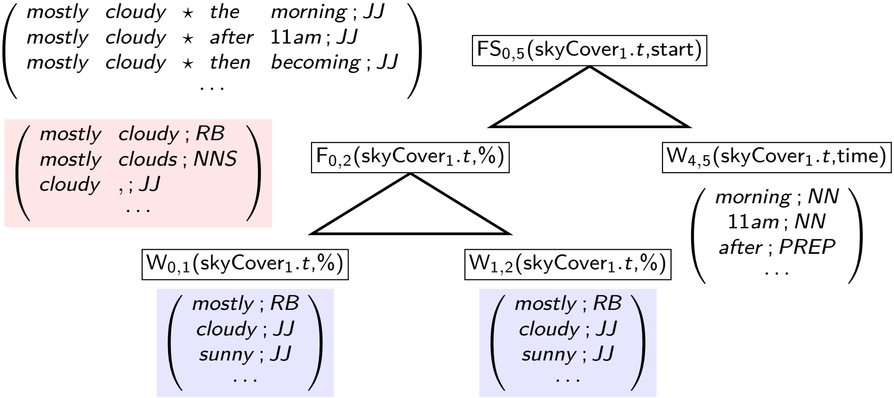
$$\epsilon$$

$$\begin{pmatrix} \textit{mostly} ; \textit{RB} \\ \textit{cloudy} ; \textit{JJ} \\ \textit{sunny} ; \textit{JJ} \\ \ldots \end{pmatrix}$$

Decoding

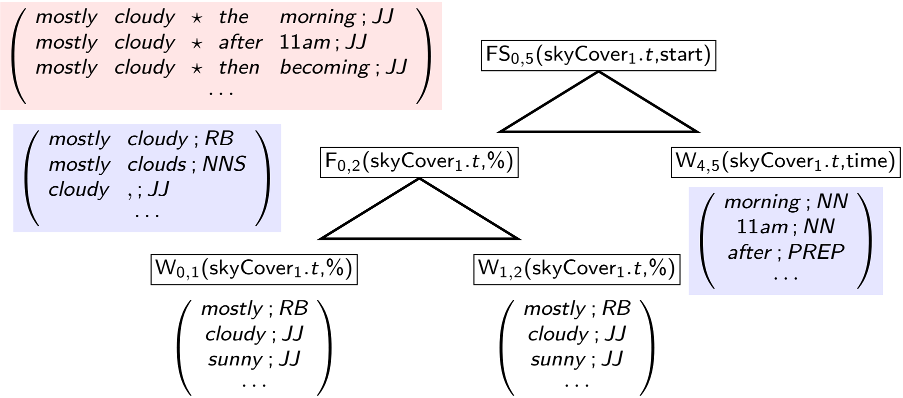

mostly cloudy ⋆ the morning ; JJ
mostly cloudy ⋆ after 11am ; JJ
mostly cloudy ⋆ then becoming ; JJ
...
mostly cloudy ; RB
mostly clouds ; NNS
cloudy , ; JJ
...
$FS_{0,5}(\text{skyCover}_1.t,\text{start})$
$F_{0,2}(\text{skyCover}_1.t,\%)$
$W_{4,5}(\text{skyCover}_1.t,\text{time})$
morning ; NN
11am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

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
...
$FS_{0,5}$(skyCover$_1$.t,start)
mostly cloudy ; RB
mostly clouds ; NNS
cloudy , ; JJ
...
$F_{0,2}$(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
...

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)

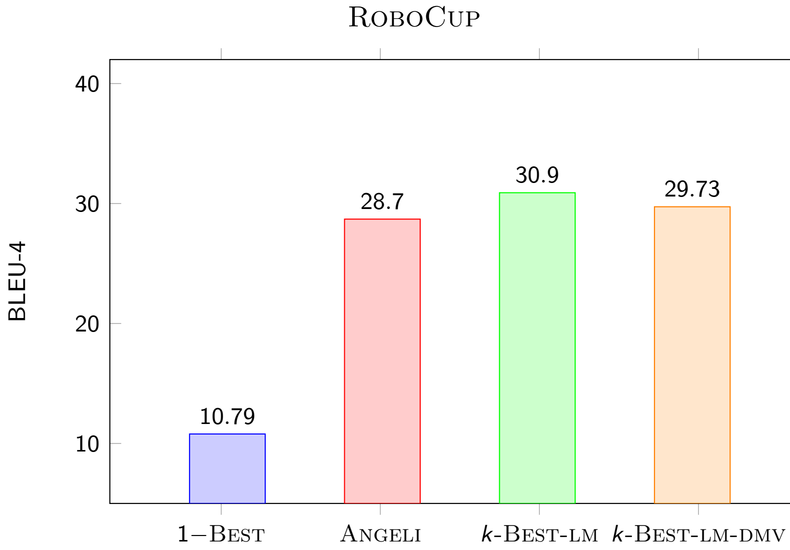
Evaluation

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

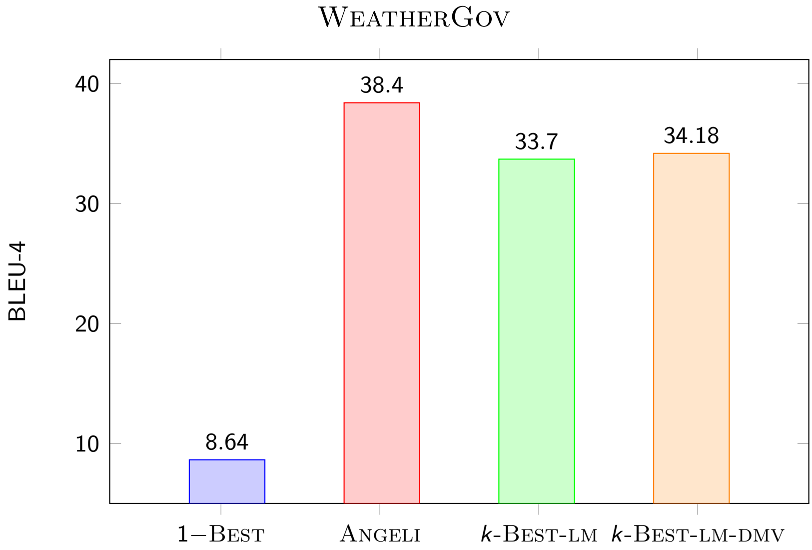
System Comparison

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

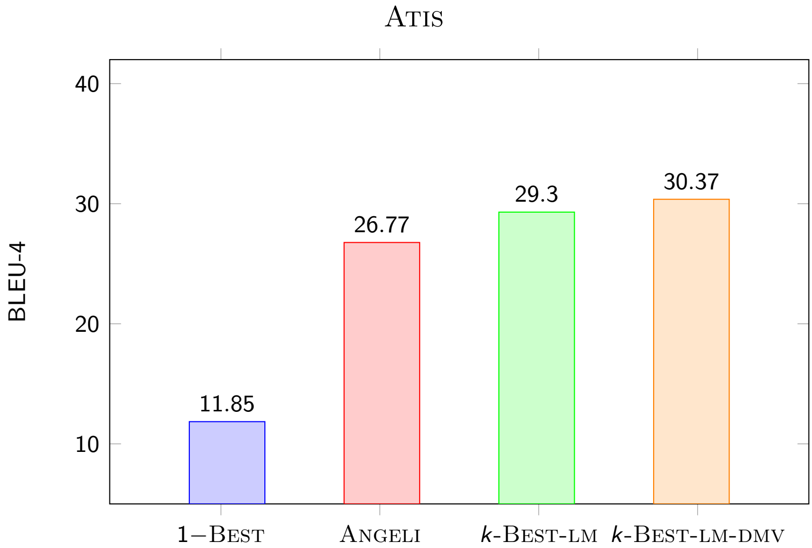
Results: Automatic Evaluation



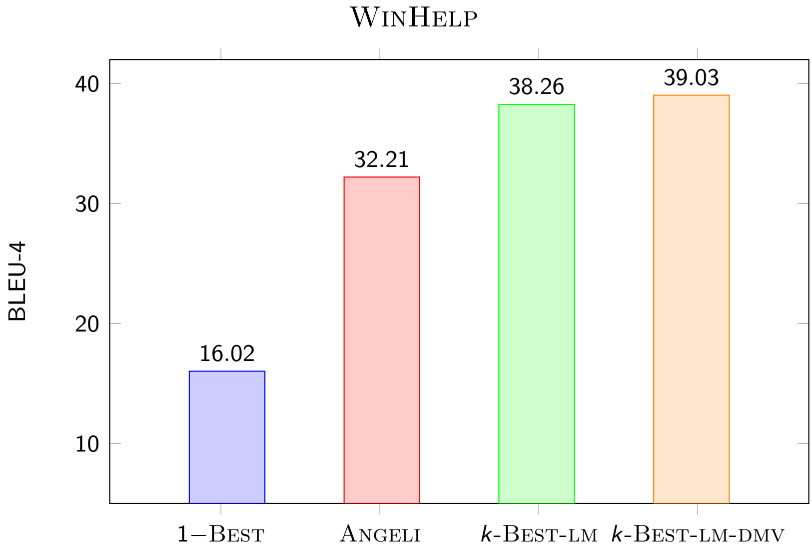
Results: Automatic Evaluation



Results: Automatic Evaluation



Results: Automatic Evaluation



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

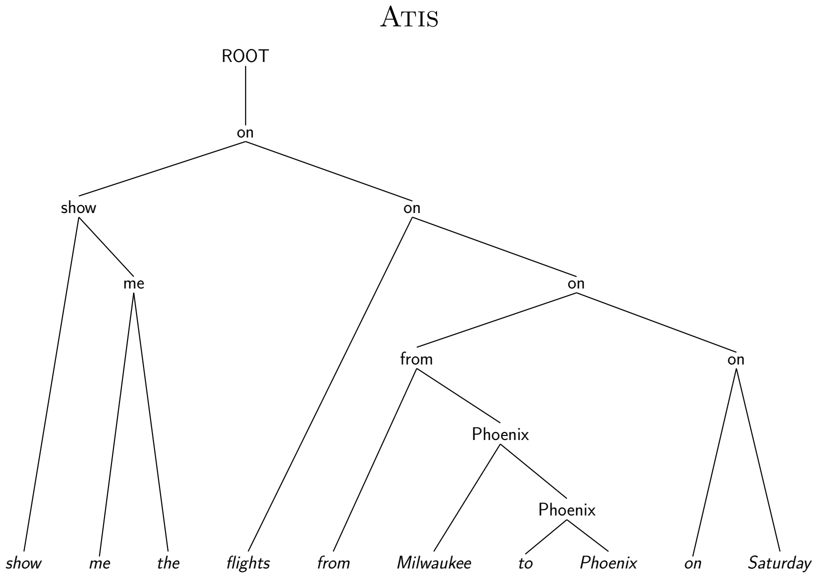
Search	
type	**what**
query	flight

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



Conclusions

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

Conclusions

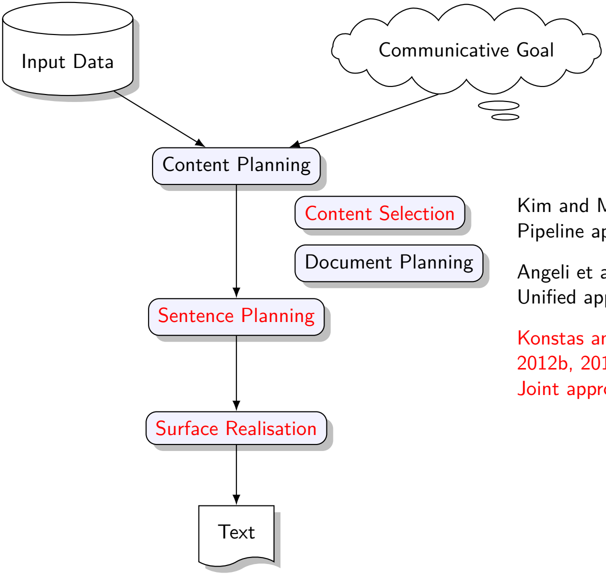
- Generation as parsing problem
- Unsupervised end-to-end generation system
- Performance comparable to state-of-the-art

- What about document planning?

Konstas and Lapata, ACL 2013

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

Traditional NLG Pipeline

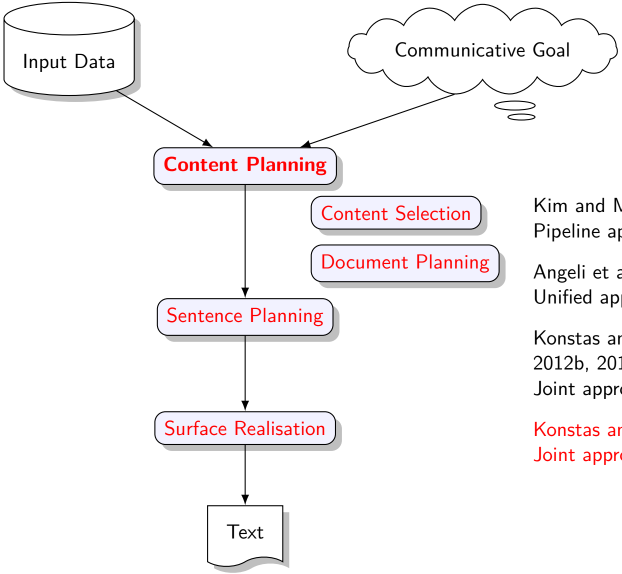


Kim and Mooney (2010)
Pipeline approach

Angeli et al. (2010)
Unified approach

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

Traditional NLG Pipeline



Kim and Mooney (2010)
Pipeline approach

Angeli et al. (2010)
Unified approach

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

Konstas and Lapata (2013a)
Joint approach

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

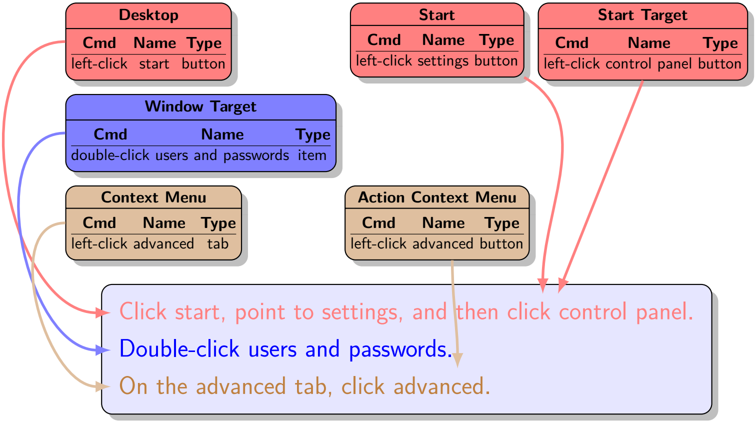
Cmd	Name	Type
left-click	advanced	button

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 (G_{RSE})

Planning with Record Sequences

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

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

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

Planning with Record Sequences

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

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

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

2. $\dfrac{\text{desktop} \mid \text{start} \mid \text{start-target}}{\text{Click start, point to settings, and then click control panel.}} \;\|$

   $\dfrac{\text{window-target}}{\text{Double-click users and passwords.}} \;\| \; \dfrac{\text{contextMenu} \mid \text{action-contextMenu}}{\text{On the advanced tab, click advanced.}} \;\|$

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

Planning with Record Sequences

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

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

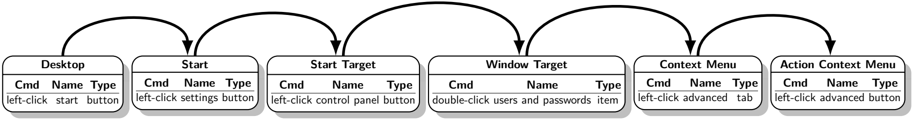
   Split a document into sentences, each terminated by a full-stop.

2. $\underbrace{\text{Click start, point to settings, and then click control panel.}}_{}$ with labels: desktop | start | start-target ‖ Double-click users and passwords. (window-target) ‖ On the advanced tab, click advanced. (contextMenu | action-contextMenu) ‖

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

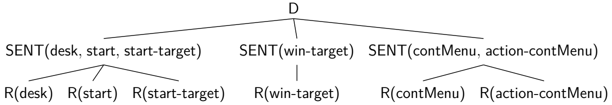
3. Goal: Learn patterns of record type sequences **within** and **among** sentences

Extended Grammar



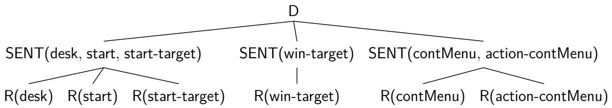
1. $\cancel{\mathrm{S} \rightarrow \mathrm{R}(start)}$
2. $\cancel{\mathrm{R}(r_i.t) \rightarrow \mathrm{FS}(r_j, start)\mathrm{R}(r_j.t) \mid \mathrm{FS}(r_j, start)}$
3. $\mathrm{FS}(r, r.f_i) \rightarrow \mathrm{F}(r, r.f_j)\mathrm{FS}(r, r.f_j) \mid \mathrm{F}(r, r.f_j)$
4. $\mathrm{F}(r, r.f) \rightarrow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

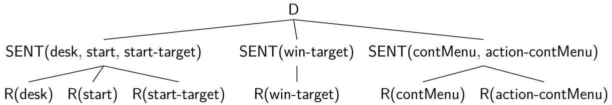
Extended Grammar



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

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 ‖]

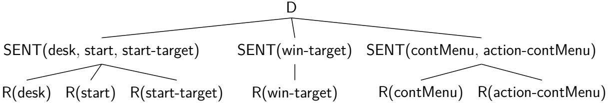
Grammar Extraction

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 ‖]



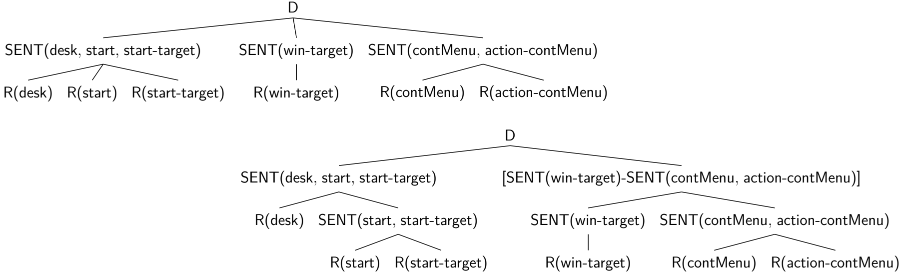
Grammar Extraction

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.

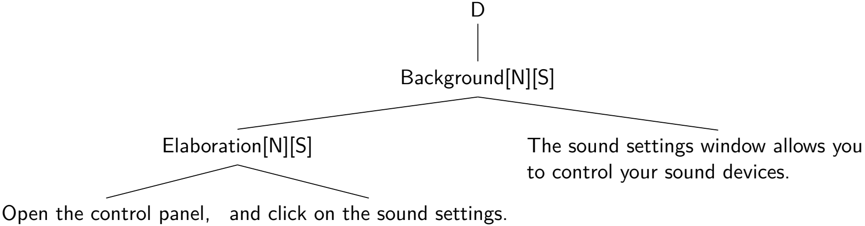
Liang et al. (2009)

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



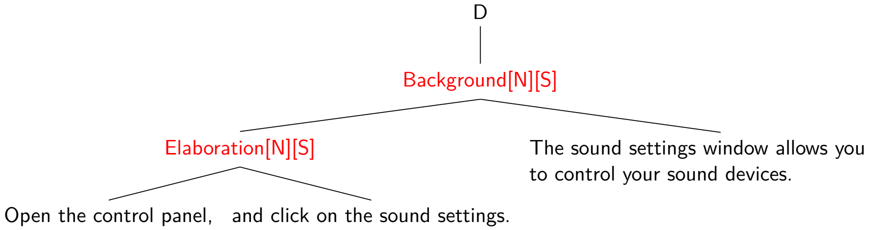
Planning with Rhetorical Structure Theory

RST (Mann and Thompson, 1988)



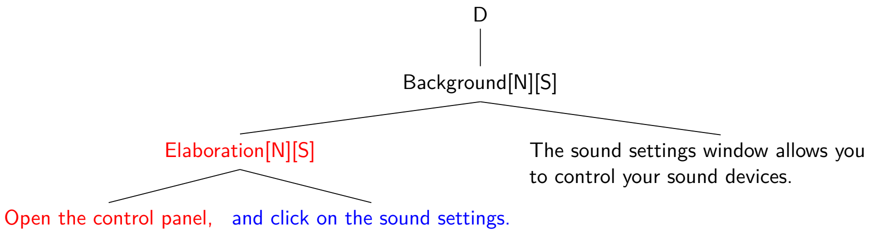
Planning with Rhetorical Structure Theory

RST (Mann and Thompson, 1988)



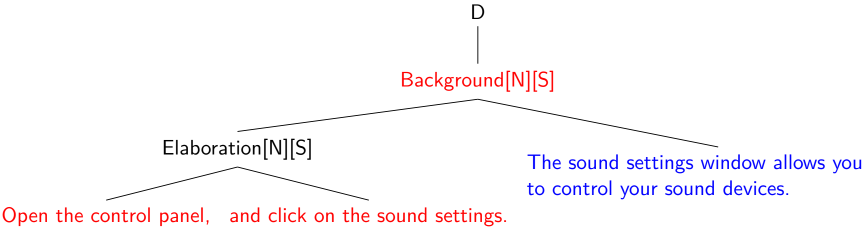
Planning with Rhetorical Structure Theory

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

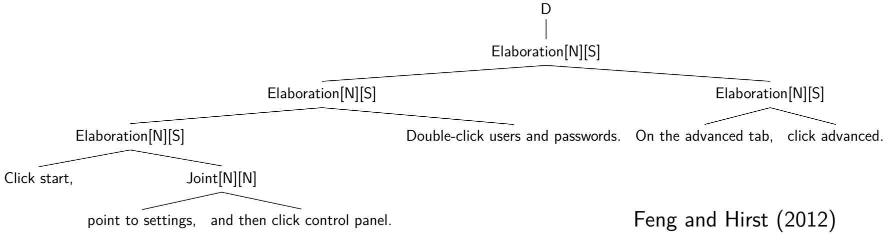
desktop	start	start-target	window-target
Click start,	point to settings,	and then click control panel.	Double-click users and passwords.

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

Liang et al. (2009)

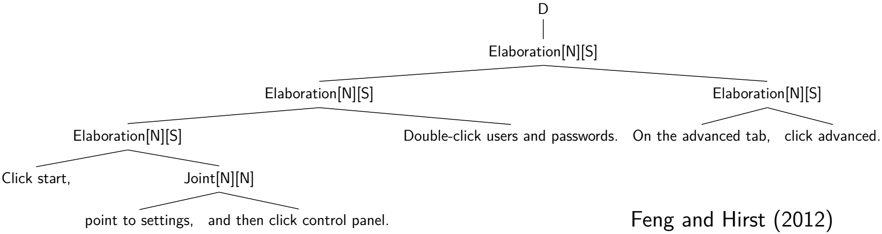
[Click start,]^{desktop} [point to settings,]^{start} [and then click control panel.]$^{start-target}$ [Double-click users and passwords.]$^{window-target}$ [On the advanced tab,]^{contextMenu} [click advanced.]$^{action-contextMenu}$

Grammar Extraction

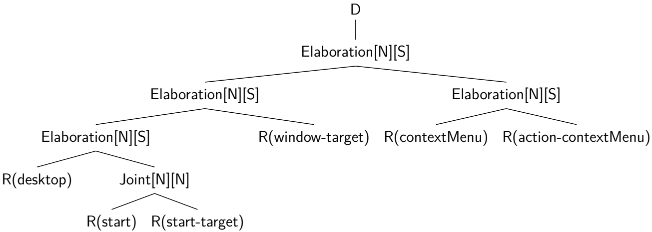


Feng and Hirst (2012)

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)

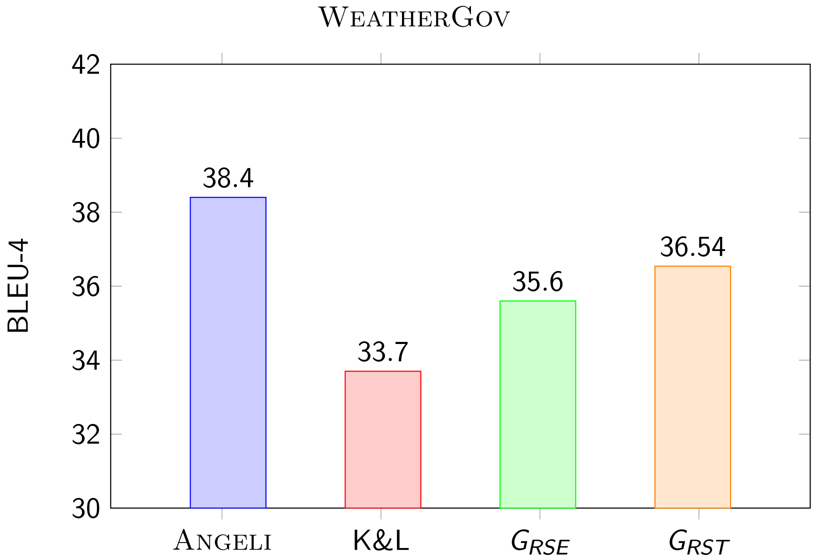
Evaluation

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

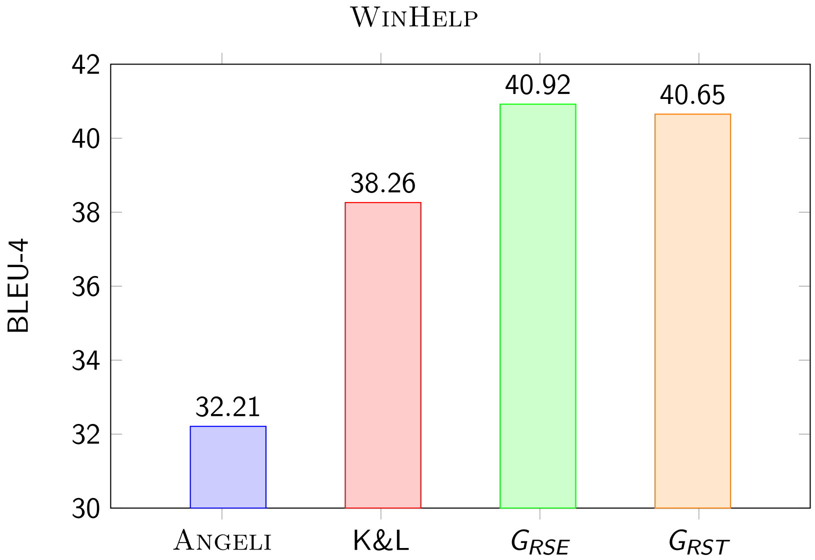
System Comparison

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

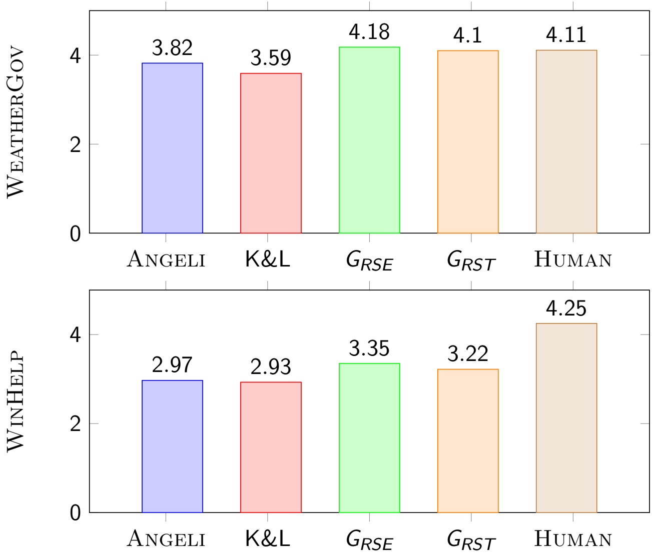
Results: Automatic Evaluation



Results: Automatic Evaluation



Results: Human Evaluation (Coherence)



Output

GRSE	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})
- How about a more sophisticated content selection model on the field level?

Konstas and Lapata, ACL 2012

Concept-to-text Generation via Discriminative Reranking

Discriminative Reranking Model

Original Model

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

Discriminative Reranking Model

Original Model

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

Discriminative Reranking Model

Discriminative Reranking Model

Original Model

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

Discriminative Reranking Model

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

Hypergraph Reranking

Flight

from	**to**
seattle	new york

Day Number

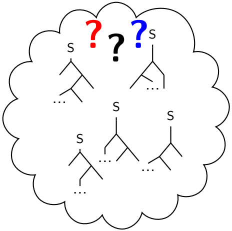
number	**dep/ar**
23	departure

Month

month	**dep/ar**
october	departure

Search

type	**what**
query	flight



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

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

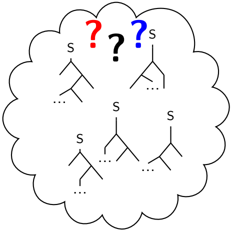
Hypergraph Reranking

Flight	
from	**to**
seattle	new york

Day Number	
number	**dep/ar**
23	departure

Month	
month	**dep/ar**
october	departure

Search	
type	**what**
query	flight



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

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

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

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

Hypergraph Reranking

Flight

from	**to**
seattle	new york

Day Number

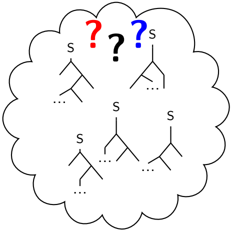
number	**dep/ar**
23	departure

Month

month	**dep/ar**
october	departure

Search

type	**what**
query	flight



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

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

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

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

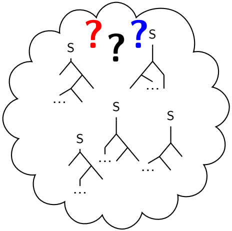
Hypergraph Reranking

Flight	
from	**to**
seattle	new york

Day Number	
number	**dep/ar**
23	departure

Month	
month	**dep/ar**
october	departure

Search	
type	**what**
query	flight



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

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

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

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

Oracle Derivation

Oracle derivation $(\mathbf{w}^*, \mathbf{h}^+)$

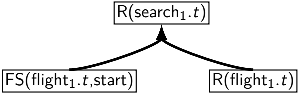
- Use the decoder of the original model but observe the training text.
- $\mathbf{w}^*$: gold standard text
- $\mathbf{h}^+$: best latent configuration

Baseline Features

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

Baseline Features

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



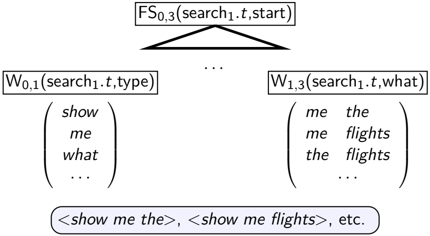
$\mathrm{R}(r_i.t) \rightarrow \mathrm{FS}(r_j, start)\mathrm{R}(r_j.t)$

Lexical Features

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

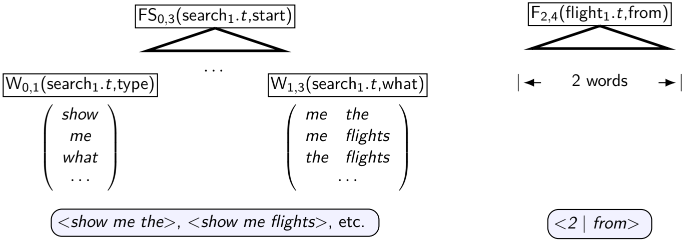
Lexical Features

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



Lexical Features

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

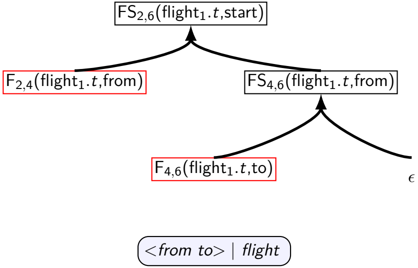


Content Selection at the Field Level Features

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

Content Selection at the Field Level Features

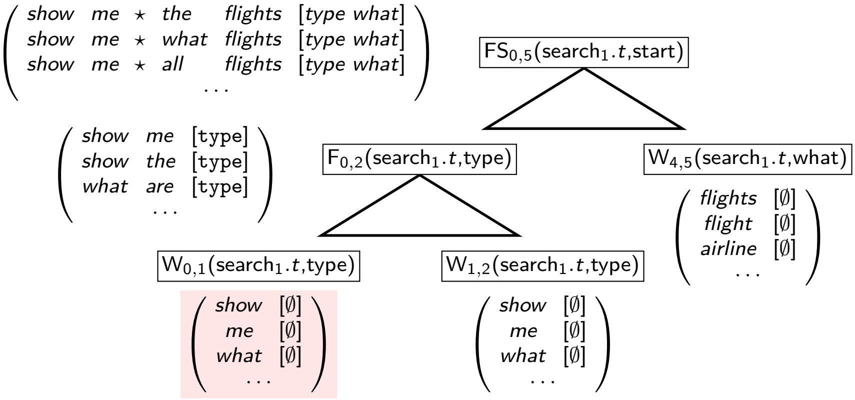
- Field Bigrams/Trigrams (non-local)
- Number of Fields per Record (local)
- Fields with no Value (local)



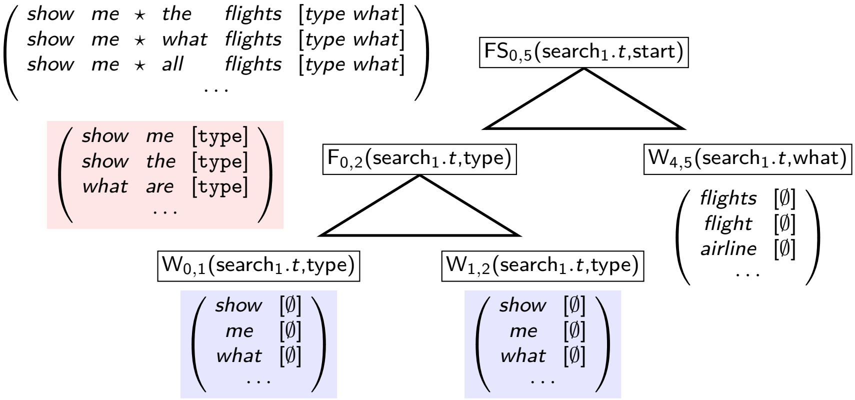
k-best Decoding

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

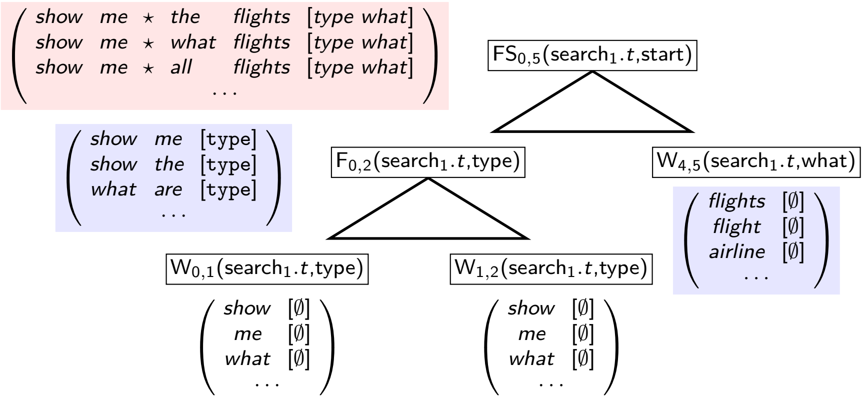
k-best Decoding



k-best Decoding



k-best Decoding



Experimental Setup

Data

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

Experimental Setup

Data

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

Evaluation

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

Experimental Setup

Data

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

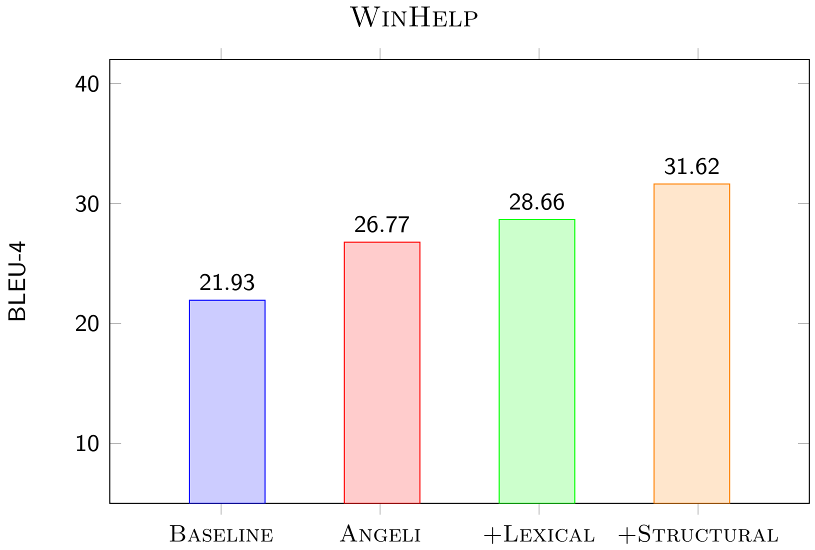
Evaluation

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

System Comparison

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

Results: Automatic Evaluation



Conclusions

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

Where do we go from here?

- More challenging factual domains: biographies from Wikipedia
- More sophisticated sentence planning: aggregation, coreference resolution
- Real induction for document planning grammar G_{RSE}: ID/LP grammars
- Discriminative reranking: use of parallelisable online learning algorithms
- More engineering: scaling can be an issue for large documents
- Apply document planning grammars to summarisation

Thank you

Questions ?

Original Model Results: Human Evaluation (LM+DMV)

RoboCup

System	Fluency	SemCor
1-Best	2.47	2.33
k-Best	4.31	3.96
Angeli	4.03	3.70
Human	4.47	4.37

WeatherGov

System	Fluency	SemCor
1-Best	1.82	2.05
k-Best	3.92	3.30
Angeli	4.26	3.60
Human	4.61	4.03

Atis

System	Fluency	SemCor
1-Best	2.40	2.46
k-Best	4.01	3.87
Angeli	3.56	3.33
Human	4.10	4.01

WinHelp

System	Fluency	SemCor
1-Best	2.57	2.10
k-Best	3.41	3.05
Angeli	3.57	2.80
Human	4.15	4.04

Discriminative Reranking Results: Human Evaluation

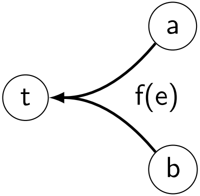
System	Fluency	SemCor
1-Best	2.70	3.05
k-Best	4.02	4.04
Angeli	3.74	3.17
Human	4.18	4.02

- *k*-Best significantly better than 1-best and Angeli ($\alpha < 0.01$)
- *k*-Best and Human are not significantly different

Hypergraphs

Definition

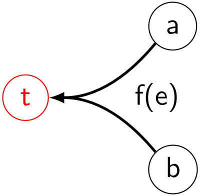
An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$ 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

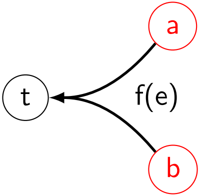
An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$ 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

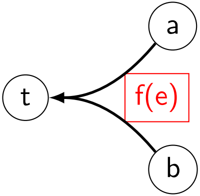
An ordered hypergraph H is a tuple $\langle N, E, t, \mathbf{R} \rangle$, where N is a finite set of nodes, E is a finite set of hyperarcs, $t \in N$ is a target node and $\mathbf{R}$ is the set of weights. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e) \rangle$, where $h(e) \in N$ 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

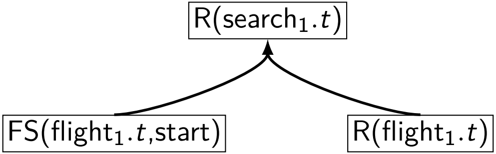
Definition

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 $H : \langle N, E, t, \mathbf{R} \rangle$

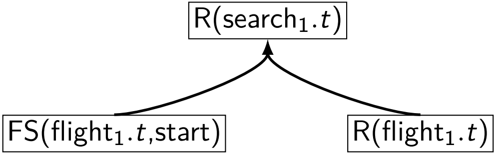


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes$$
$$w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

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

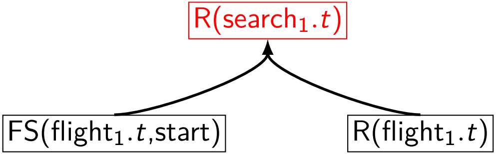


$$
\begin{aligned}
f(e) =\, & f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\
& w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))
\end{aligned}
$$

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

Hypergraph Construction

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

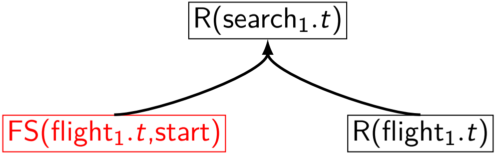


$$
\begin{aligned}
f(e) = & f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\
& w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))
\end{aligned}
$$

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

Hypergraph Construction

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

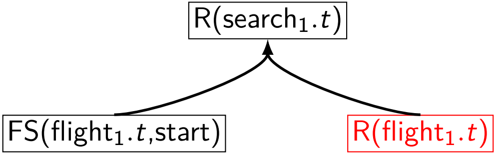


$$
\begin{aligned}
f(e) = & f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes \\
& w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))
\end{aligned}
$$

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

Hypergraph Construction

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

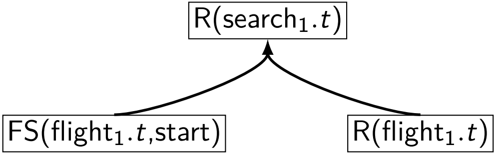


$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes$$
$$w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

Hypergraph Construction

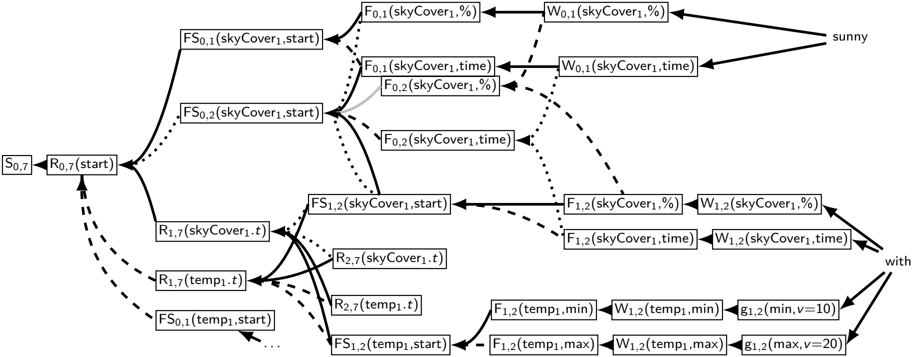
Map standard weighted CYK algorithm to hypergraph $H : \langle N, E, t, \mathbf{R} \rangle$



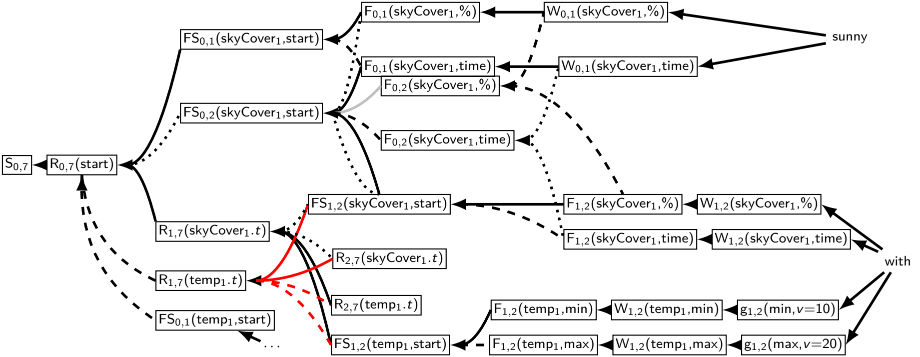
$$f(e) = f(\mathrm{FS}_{5,7}(\mathrm{flight}_1.t, start)) \otimes f(\mathrm{R}_{7,9}(\mathrm{flight}_1.t)) \otimes$$
$$w(\mathrm{R}(\mathrm{search}_1.t) \rightarrow \mathrm{FS}(\mathrm{flight}_1, start)\ \mathrm{R}(\mathrm{flight}_1.t))$$

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

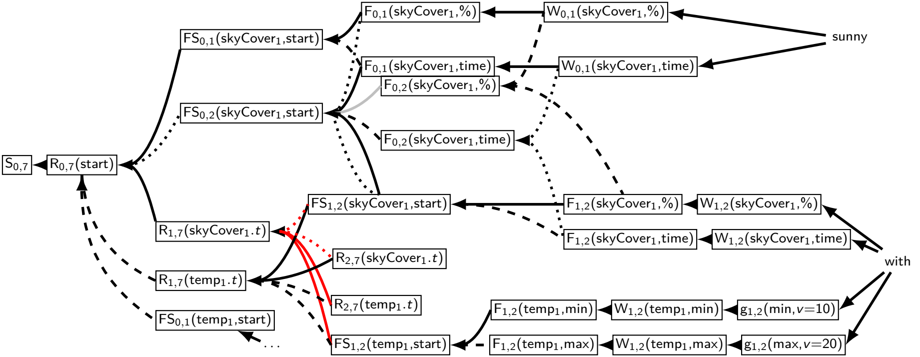
Hypergraph Example



Hypergraph Example



Hypergraph Example



Determining Text Length

- Train a linear regression model
- Idea: The more records and fields that have values
  in the database $\rightarrow$ 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