Multilingual Temporal Parsing

Gabor Angeli
angeli@stanford.edu

Jakob Uszkoreit
uszkoreit@google.com

Example

Let's meet for a few hours next week , say August 12 ?

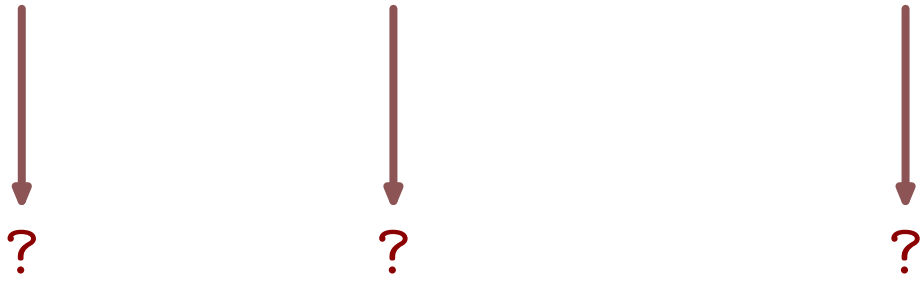
Example

Let's meet for **a few hours next week** *, say* **August 12** *?*

Detection Finding temporal *phrases* in a sentence.

Example

Let's meet for **a few hours next week** *, say* **August 12** *?*

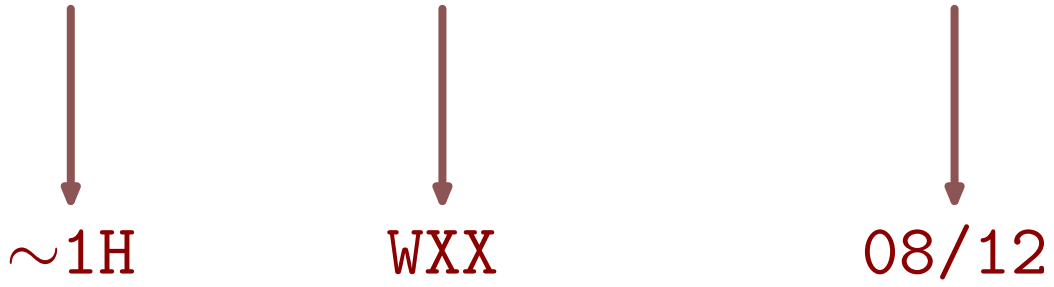


Detection Finding temporal *phrases* in a sentence.

Interpretation Finding the grounded *meaning* of a phrase

Example

Let's meet for **a few hours next week** *, say* **August 12** *?*



Detection Finding temporal *phrases* in a sentence.

Interpretation Finding the grounded *meaning* of a phrase

But, often incomplete information

Example

Detection Finding temporal *phrases* in a sentence.

Interpretation Finding the grounded *meaning* of a phrase

But, often incomplete information

Incorporate a `reference time`

Time In Information Extraction

Google

News

[The falcon] was found injured **last Thursday** *in . . .*

. . . she died **early Saturday morning**

Benjamin Franklin Federal Savings and Loan Association said it plans to restructure in the wake of a **third-quarter** *loss*

Time In Information Extraction

Google

News

[The falcon] was found injured **last Thursday** *in . . .*

. . . she died **early Saturday morning**

Benjamin Franklin Federal Savings and Loan Association said it plans to restructure in the wake of a **third-quarter** *loss*

Communication

Actually I am on vacation the **last three weeks of November**

I have some time available at the **end of next week**

System

Input (w,t) (*phrase* , `reference`)

System

Input (w,t) (*Last 2 days* , 2013-08-05)

System

Input (w,t) (*Last 2 days* , 2013-08-05)

Output τ^* `normalized time`

System

Input (w,t) (*Last 2 days* , 2013-08-05)

Output τ^* 2013-08-03 – 2013-08-05

System

Input (w,t) (*Last 2 days* , 2013-08-05)

Latent parse R

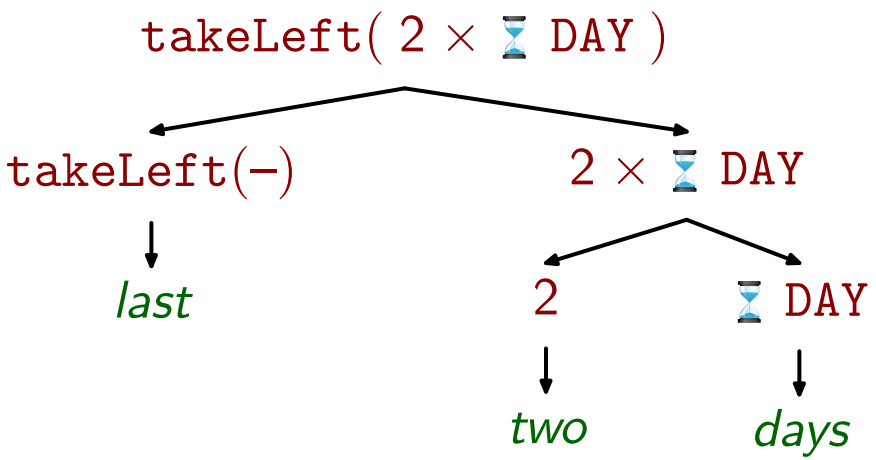
Output τ^* 2013-08-03 – 2013-08-05

System

Input (w,t) (*Last 2 days* , 2013-08-05)

↓

Latent parse R



↓

Output τ^* 2013-08-03 – 2013-08-05

Elements of Latent Parse

 `Range`	*the week of August 5, 2013*
 `Sequence`	*week*
⌛ `Duration`	*a week*
Nil	*this*
Functions	*last* [`takeLeft(−)`]

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 12th*

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 13th*

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 13th*

Domain pragmatics: *A year ago*

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 13th*

Domain pragmatics: *Sales are down from* **a year ago**

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 13th*

Domain pragmatics: *Remember? We got married* **a year ago**

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 13th*

Domain pragmatics: *Remember? We got married* **a year ago**

Rule engineering challenge

Always more rules: *2 days* **prior**, *the* **previous** *2 days*

```
/the/ /past|last/ (?: ($NUM) /to|-/ )? ($NUM)? ($TEUNITS)
/the/ /next|following/ (?: ($NUM) /to|-/ )? ($NUM)? ($TEUNITS)
...
```

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Prior Work

Google

Hand coded rules

Rigid

Syntax: *last Friday the 13th*

Domain pragmatics: *Remember? We got married* **a year ago**

Rule engineering challenge

Always more rules: *2 days* **prior**, *the* **previous** *2 days*

```
/the/ /past|last/ (?: ($NUM) /to|-/ )? ($NUM)? ($TEUNITS)
/the/ /next|following/ (?: ($NUM) /to|-/ )? ($NUM)? ($TEUNITS)
...
```

New set of rules for each language

Mani & Wilson (2000); Strötgen and Gertz (2010); Chang and Manning (2012)

Motivation

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Motivation

Google

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Hard:

pasado → `last(-)`

domingo →  `SUN`

```
($DOW) /pasad[oa]/
```

Motivation

Google

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Easier: annotate training data for *domingo pasado*

Motivation

Google

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Easier: annotate training data for *domingo pasado*

~ 100 grammar rules total

Same learning algorithm, hyperparameters

Motivation

Google

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Easier: annotate training data for *domingo pasado*

~ 100 grammar rules total

Same learning algorithm, hyperparameters

HeidelTime

English: ~ 2000 rules (185 combination rules)

Motivation

Google

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Easier: annotate training data for *domingo pasado*

~ 100 grammar rules total

Same learning algorithm, hyperparameters

HeidelTime

English: ~ 2000 rules (185 combination rules)

Spanish: ~ 1200 rules (167 combination rules)

Italian: ~ 1600 rules (156 combination rules)

...

Motivation

Google

Language independent

Arbitrary training data: **Last** *Sunday* / *domingo* **pasado**

Easier: annotate training data for *domingo pasado*

~ 100 grammar rules total

Same learning algorithm, hyperparameters

HeidelTime

English: ~ 2000 rules (185 combination rules)

Spanish: ~ 1200 rules (167 combination rules)

Italian: ~ 1600 rules (156 combination rules)

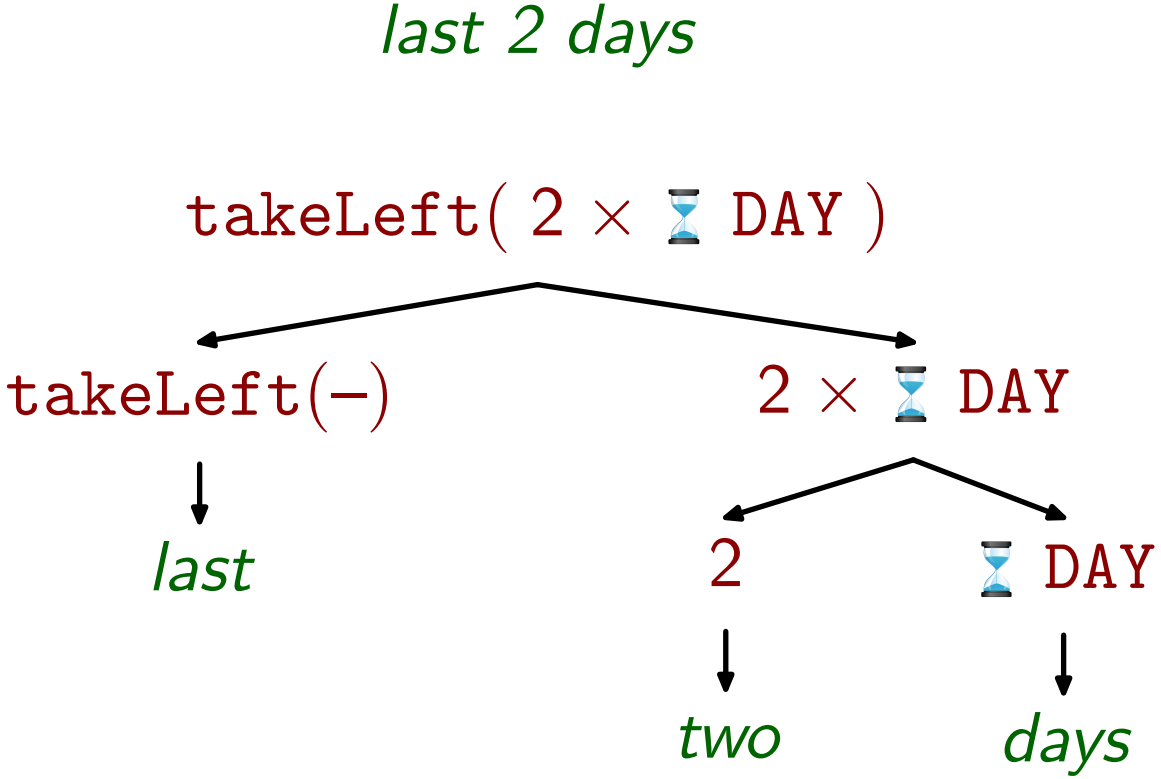
...

SUTime

English: ~ 900 rules

Language Independent

Latent semantic representation

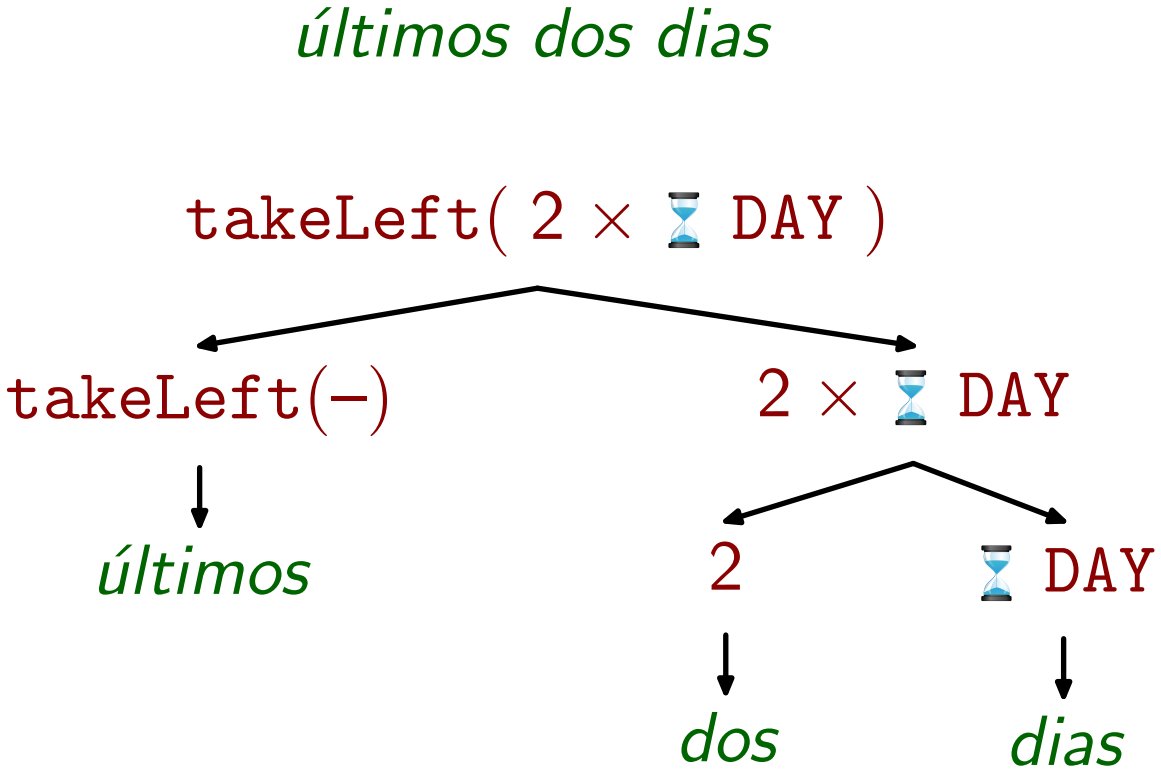


[2013-08-05] → 2013-08-03 − 2013-08-05

Language Independent

Latent semantic representation

Many languages share representation

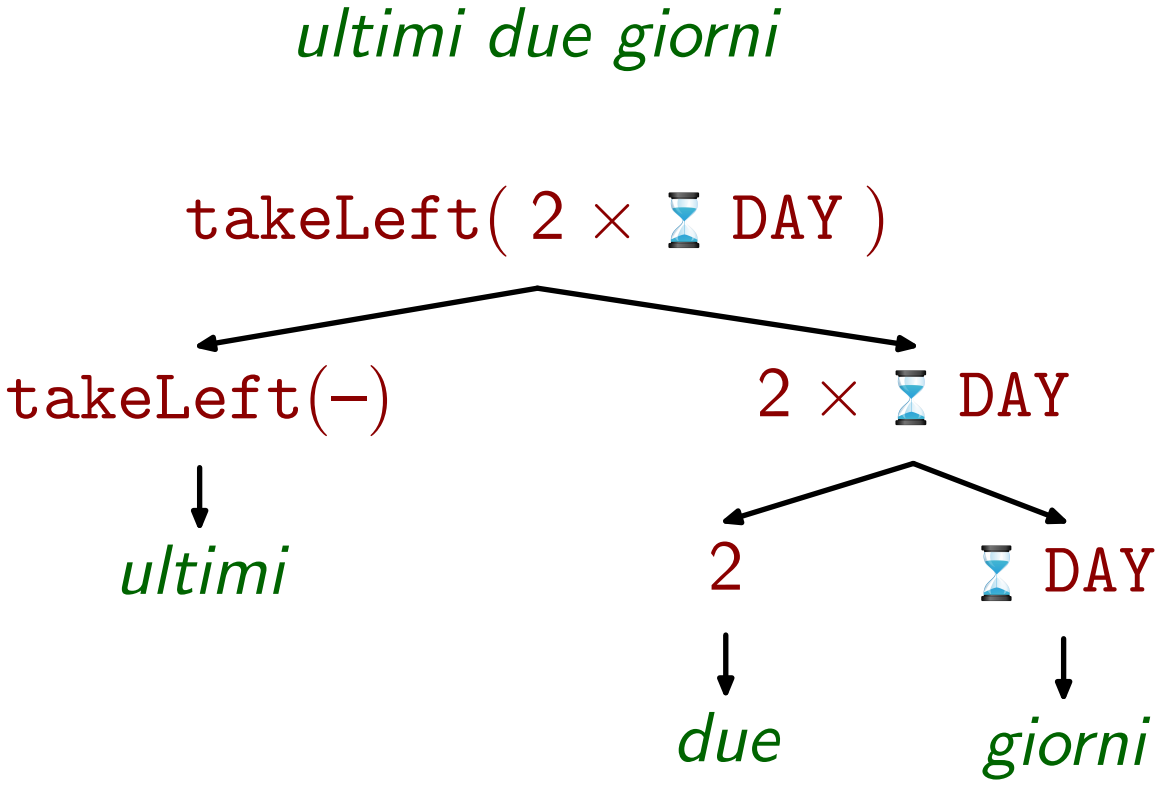


[2013-08-05] → 2013-08-03 – 2013-08-05

Language Independent

Latent semantic representation

Many languages share representation



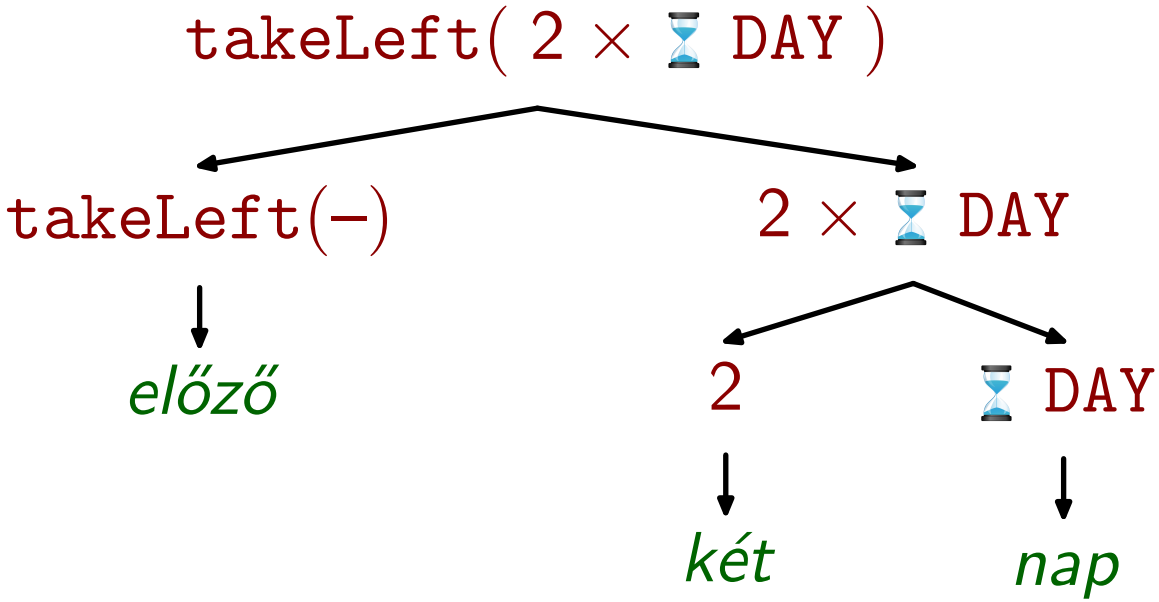
[2013-08-05] → 2013-08-03 – 2013-08-05

Language Independent

Latent semantic representation

Many languages share representation

előző két nap



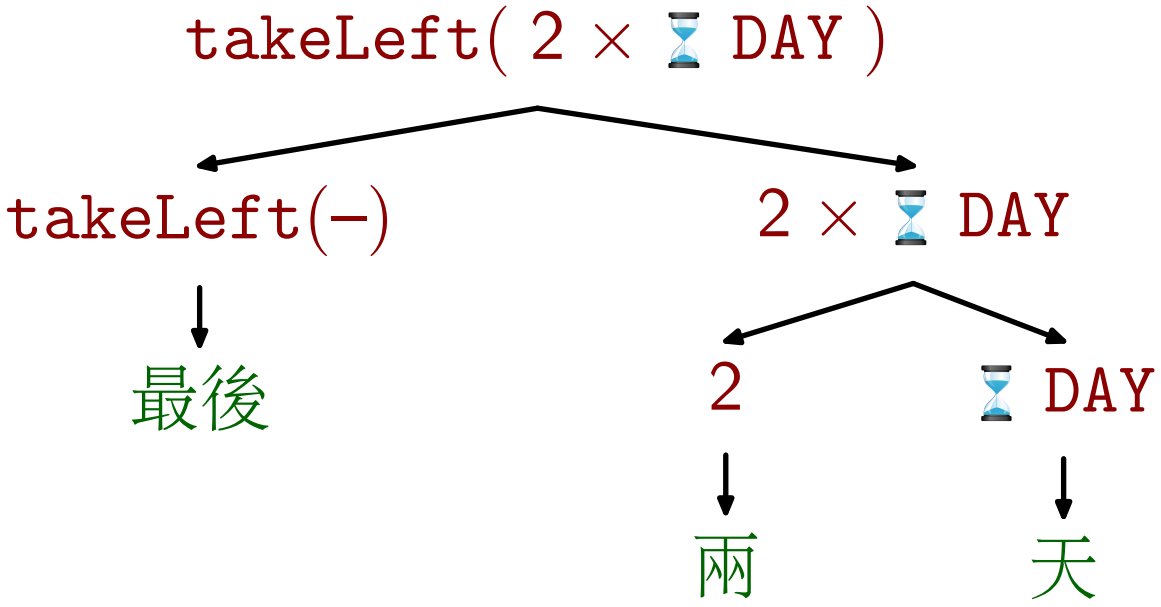
[2013-08-05] → 2013-08-03 – 2013-08-05

Language Independent

Latent semantic representation

Many languages share representation

最後兩天

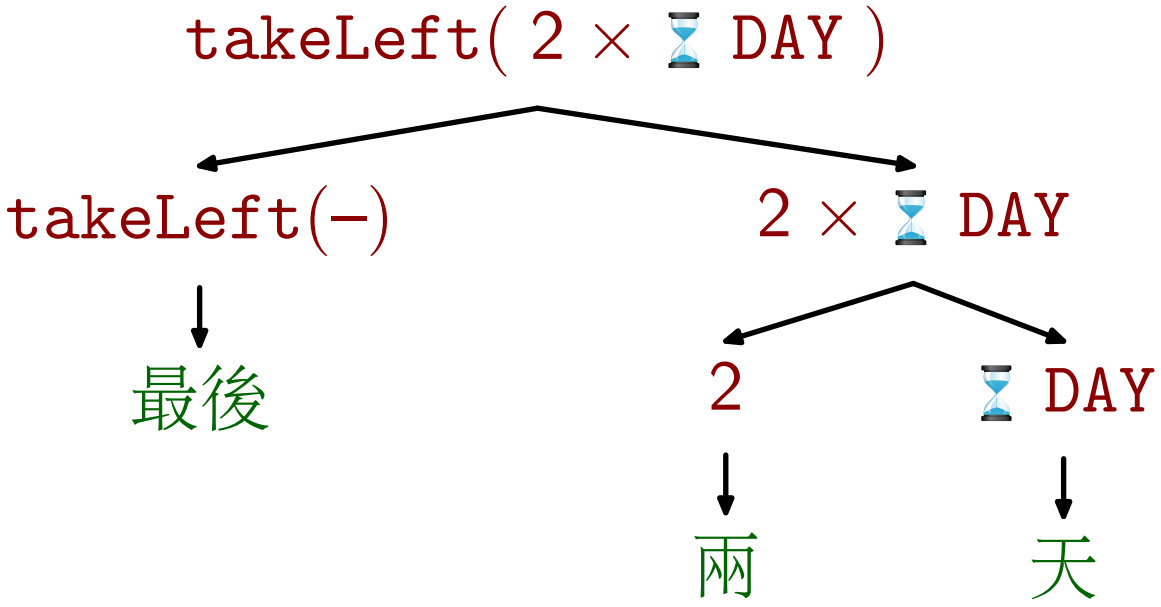


[2013-08-05] → 2013-08-03 – 2013-08-05

Language Independent

Latent semantic representation

Many languages share representation

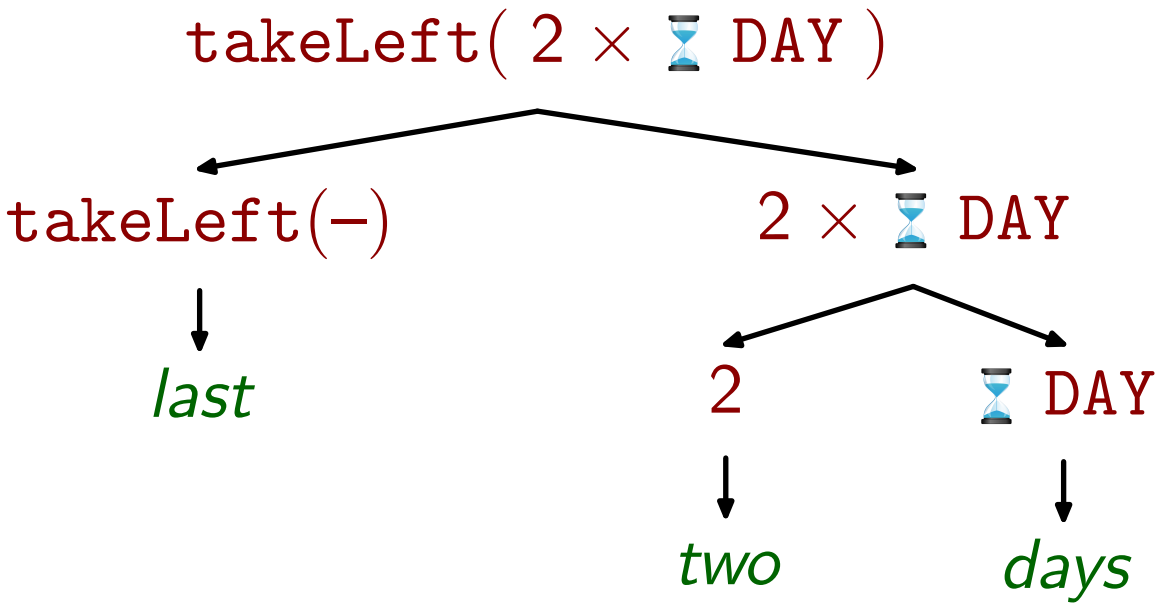


[2013-08-05] → 2013-08-03 – 2013-08-05

Parsing

Parsing temporal phrases

Explained in, but not specific to English

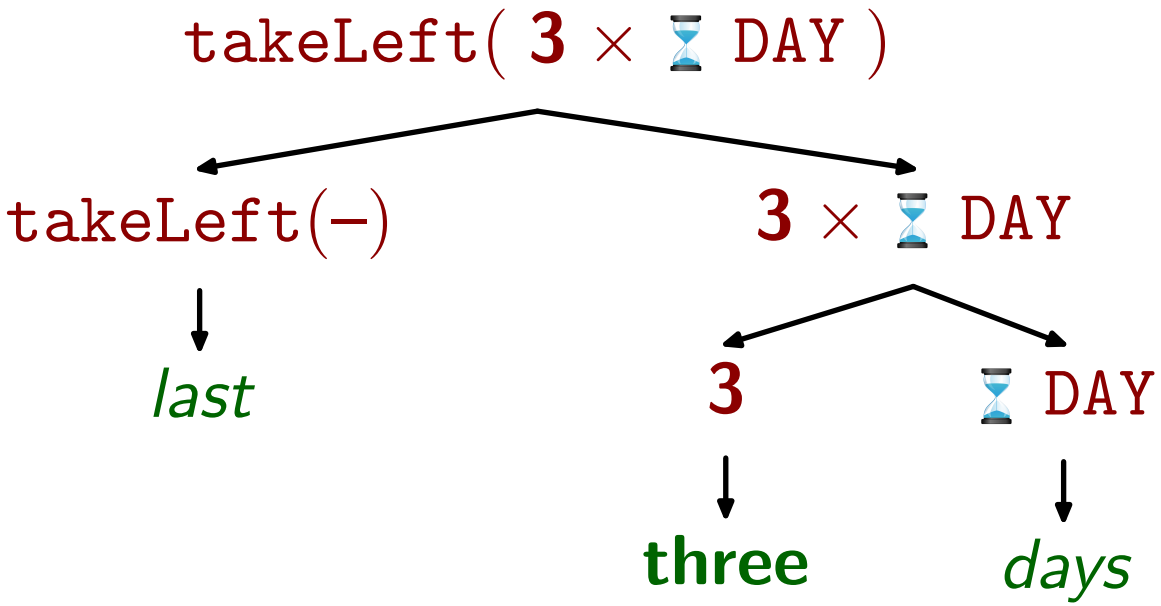


Parsing

Parsing temporal phrases

Explained in, but not specific to English

Naïvely, domain of nonterminals is large

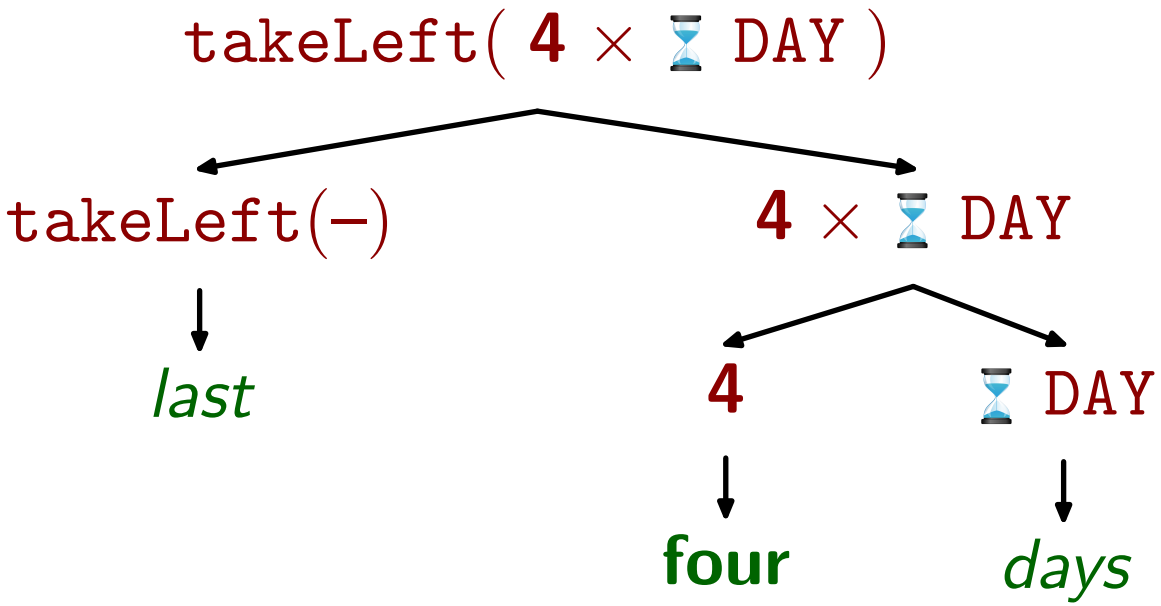


Parsing

Parsing temporal phrases

Explained in, but not specific to English

Naïvely, domain of nonterminals is large

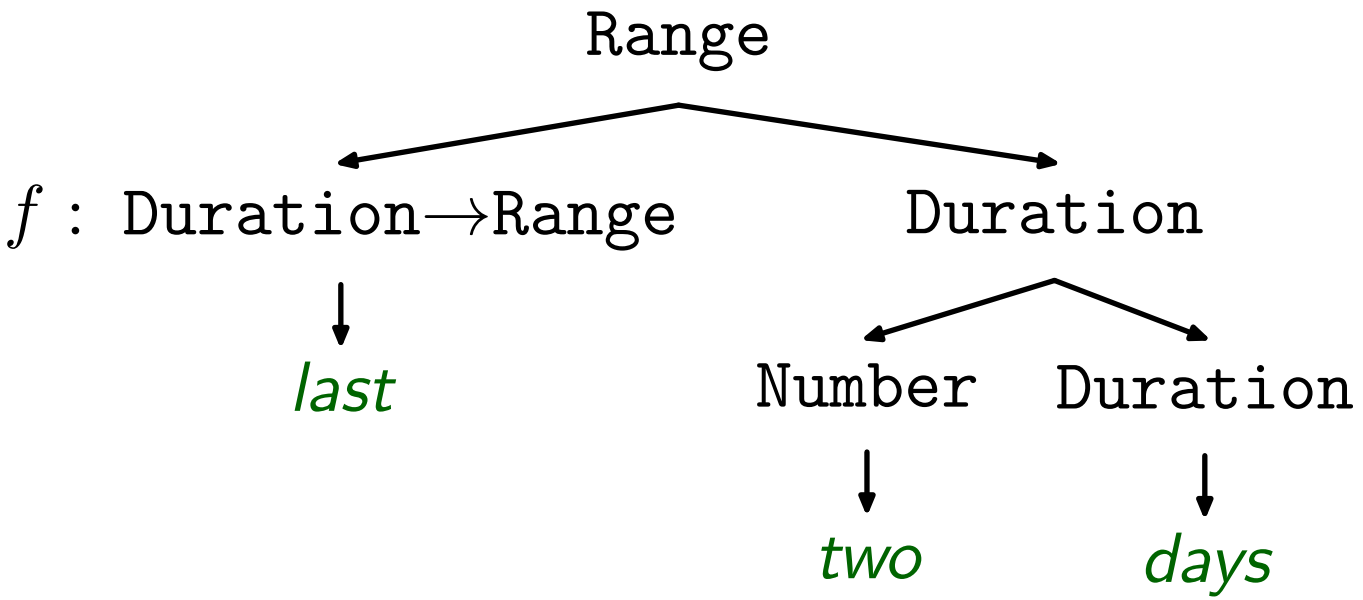


Parsing

Domain of nonterminals is large

Consider: *last 7 days* , *last 3 months* , etc.

Generative Grammar: Group nonterminals based on *types*

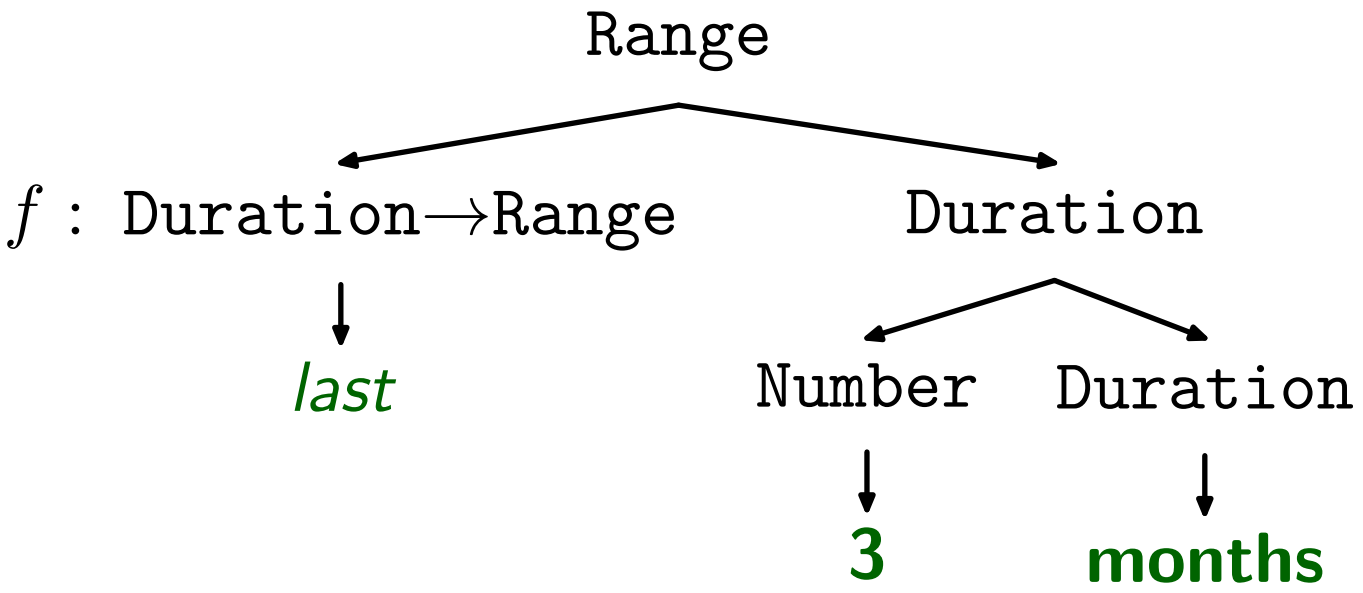


Parsing

Domain of nonterminals is large

Consider: *last 7 days* , *last 3 months* , etc.

Generative Grammar: Group nonterminals based on *types*

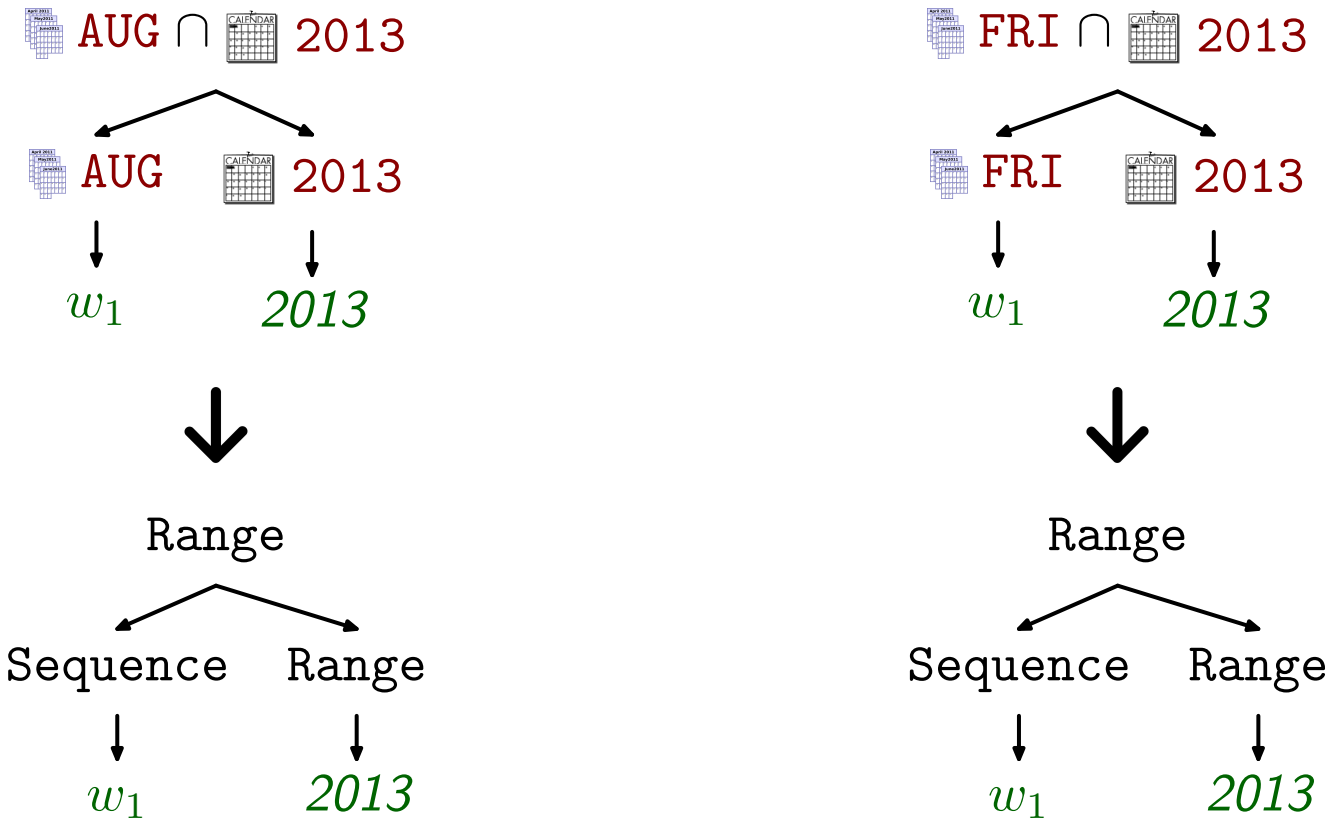


Parsing

Domain of nonterminals is large

Consider: *last 7 days* , *last 3 months* , etc.

Generative Grammar: Group nonterminals based on *types*



Angeli *et al.* 2012

Parsing

Google

Domain of nonterminals is large

Consider: *last 7 days* , *last 3 months* , etc.

Generative Grammar: Group nonterminals based on *types*

Discriminative model

Coarse-grained features over types:  `Range`

Fine grained features over values:  `2013`

Adapt semantic parser (e.g., Liang *et al.* 2011)

Angeli *et al.* 2012

Features

Friday

of this *week*

Features

↓

Friday

of this *week*

`lex` <  `FRI` , *Friday* >

Parallel generative grammar

Features

Friday Nil

of this *week*

`lex` $<$ NIL , *of this* $>$

Parallel generative grammar

Features

FRI

↓

Friday Nil

↓

of this *week*

`lex` $<$ NIL , *of this* $>$

`lex` $<$ NIL , *of* $>$

`lex` $<$ NIL , *this* $>$

Features


FRI

↓

Friday

Nil

↓

of this

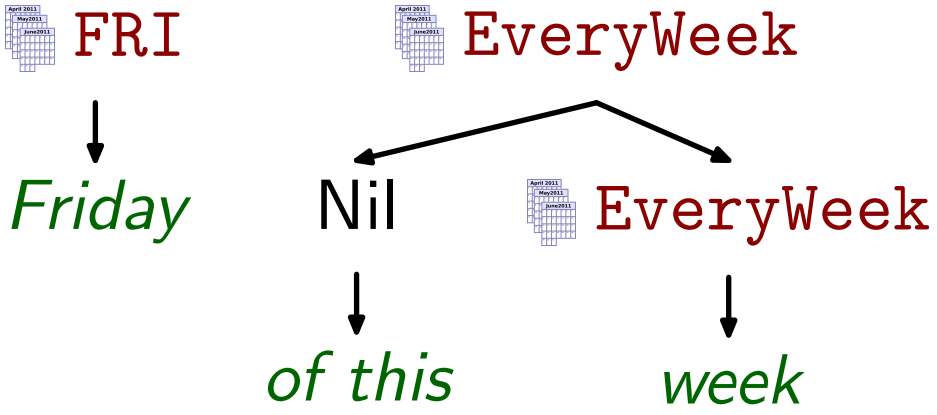

EveryWeek

↓

week

`lex` < 
`EveryWeek` , *week* >

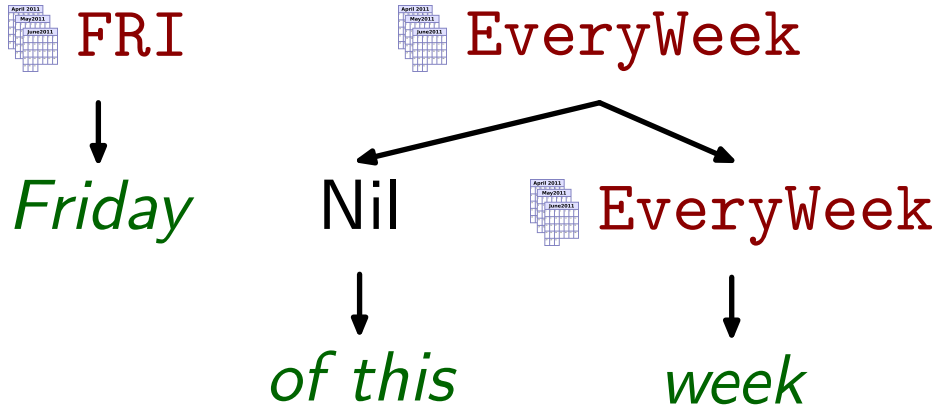
Features



`bracket` $< \textsc{Nil} , \textsc{Sequence} >$

Parallel generative grammar

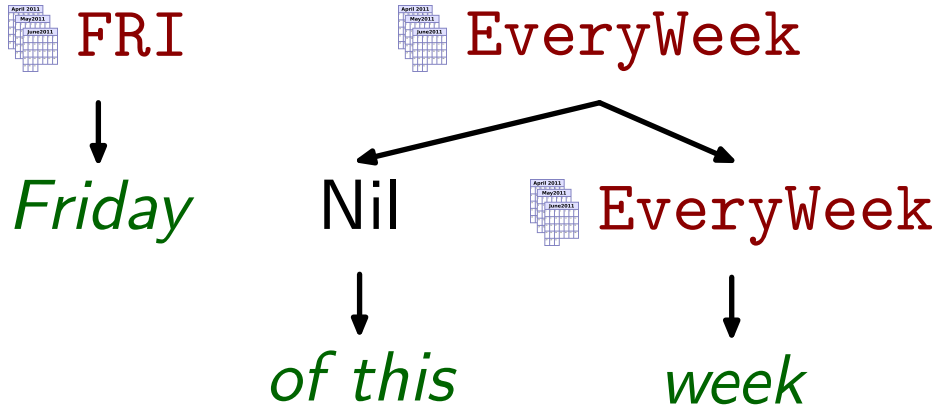
Features



bracket $< \textsc{Nil} , \textsc{Sequence} >$

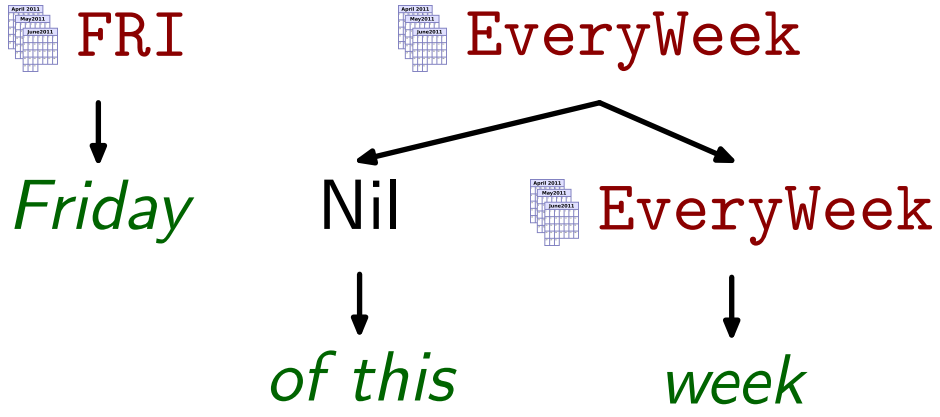
bracket $< \textsc{Nil} ,$ EveryWeek $>$

Features



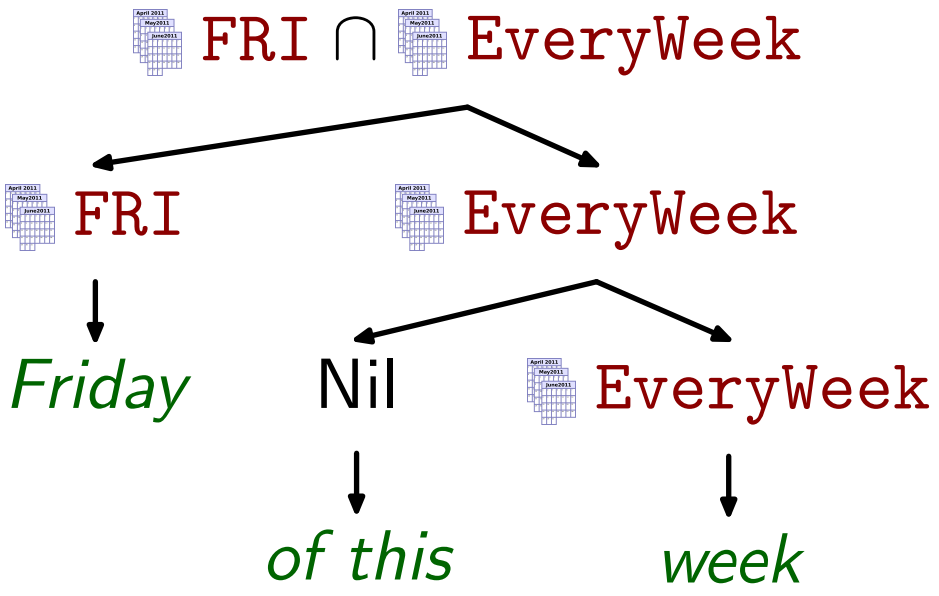
bracket	< Nil , Sequence >
bracket	< Nil , EveryWeek >
lex	< Nil_*of this* , Sequence >
lex	< Nil_*of* , Sequence >
lex	< Nil_*this* , Sequence >

Features



bracket	< Nil , Sequence >
bracket	< Nil , EveryWeek >
lex	< Nil_*of this* , EveryWeek >
lex	< Nil_*of* , EveryWeek >
lex	< Nil_*this* , EveryWeek >

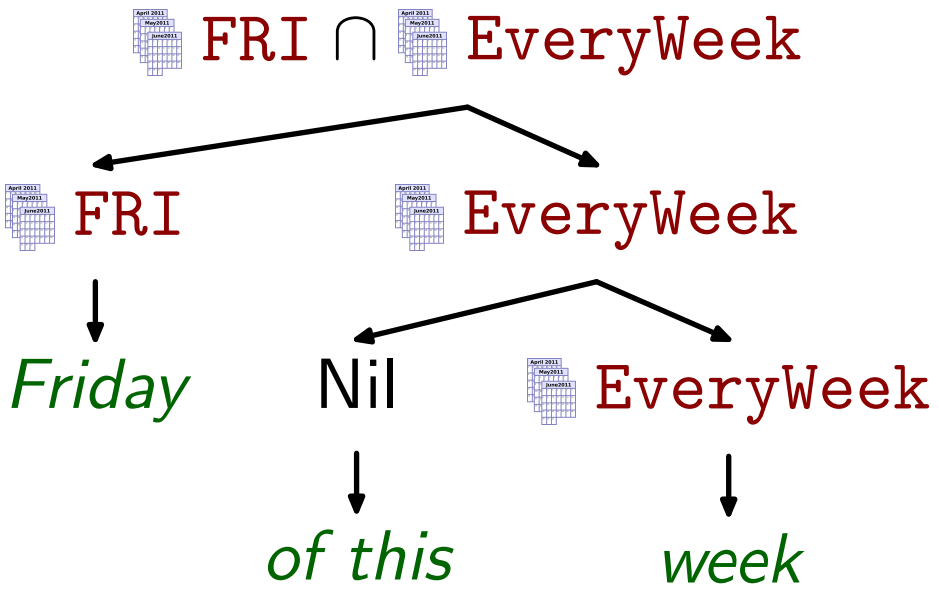
Features



bracket $<$ SEQUENCE , SEQUENCE $>$

bracket $<$ INTERSECT , FRI , EveryWeek $>$

Features



`bracket`	$<$ SEQUENCE , SEQUENCE $>$
`bracket`	$<$ INTERSECT , FRI , EveryWeek $>$
`valid`	$<$ IS_VALID $>$

(e.g., filter February 30th)

Training Setup

Given $\{ (x , y) \}$

Training Setup

Given { ((*Phrase* , `Reference`) , `Time`) }

Training Setup

Given { ((*Phrase* , `Reference`) , `Time`) }

Not given latent parse

Not given lexical or language cues

Training Setup

Given { ((*Phrase* , `Reference`) , `Time`) }

Not given latent parse

Not given lexical or language cues

Therefore, in general, multiple parses ground to same time

Training Setup

Given { ((*Phrase* , `Reference`) , `Time`) }

Not given latent parse

Not given lexical or language cues

Therefore, in general, multiple parses ground to same time

((w_1 w_2 , `2013-08-05`) , `2013-08-12`)

Training Setup

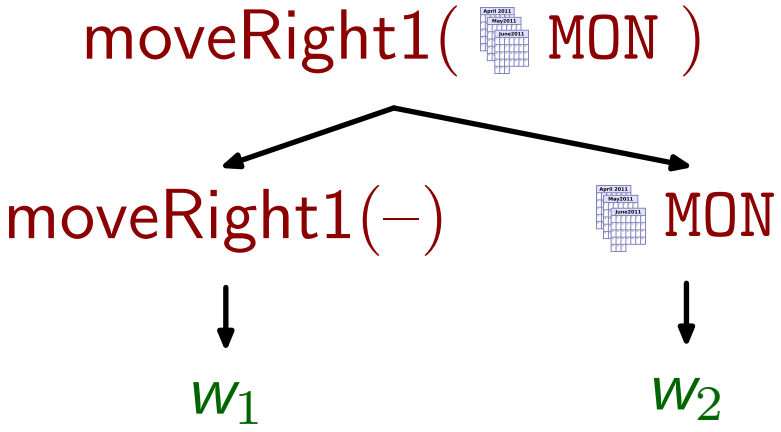
Given { ((*Phrase* , `Reference`) , `Time`) }

Not given latent parse

Not given lexical or language cues

Therefore, in general, multiple parses ground to same time

((w_1 w_2 , `2013-08-05`) , `2013-08-12`)



e.g., $w_1 = \textit{next}$, $w_2 = \textit{Monday}$

Training Setup

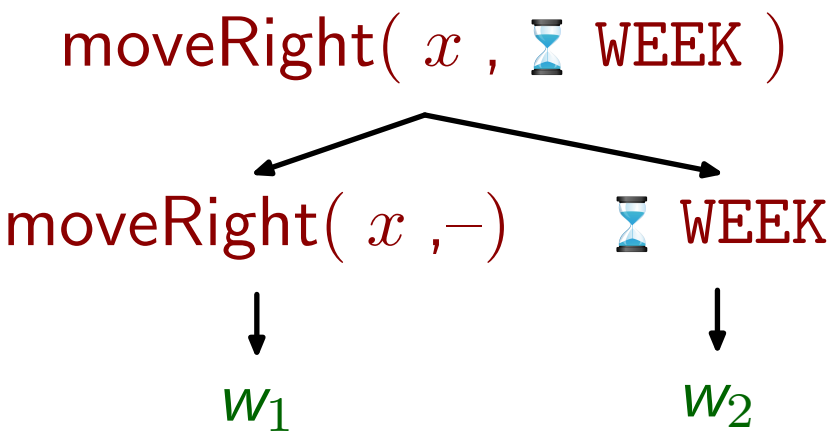
Given { ((*Phrase* , `Reference`) , `Time`) }

Not given latent parse

Not given lexical or language cues

Therefore, in general, multiple parses ground to same time

((w_1 w_2 , `2013-08-05`) , `2013-08-12`)



e.g., $w_1 = next$, $w_2 = Monday$

e.g., $w_1 = next$, $w_2 = week$

Training Setup

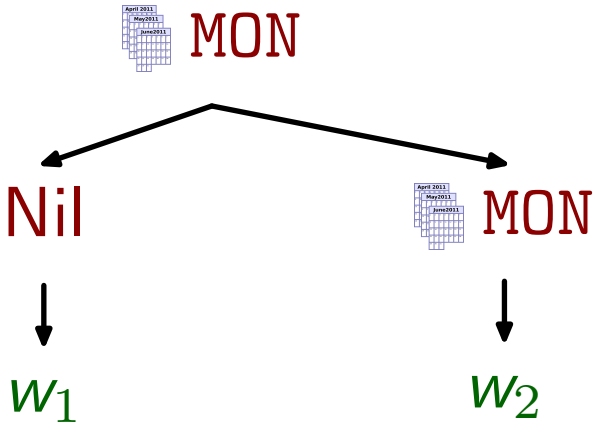
Given { ((*Phrase* , `Reference`) , `Time`) }

Not given latent parse

Not given lexical or language cues

Therefore, in general, multiple parses ground to same time

((w_1 w_2 , `2013-08-05`) , `2013-08-12`)



e.g., $w_1 = \textit{next}$, $w_2 = \textit{Monday}$

e.g., $w_1 = \textit{next}$, $w_2 = \textit{week}$

e.g., $w_1 = \textit{the}$, $w_2 = \textit{Monday}$

Training

For each example:

Get k-best parses for phrase

Training

For each example:

Get k-best parses for phrase

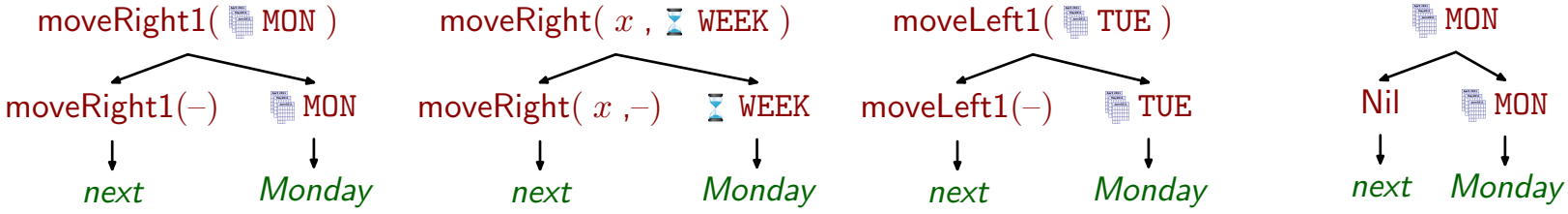
((*next Monday* , 2013-08-05) , 2013-08-12)

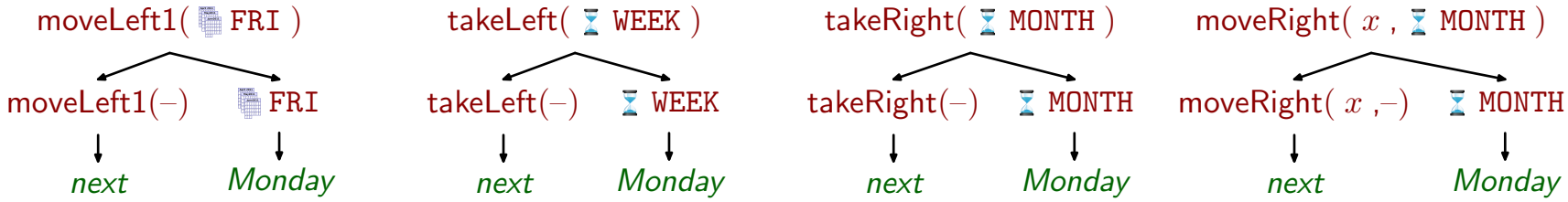
Training

For each example:

Get k-best parses for phrase

((*next Monday* , 2013-08-05) , 2013-08-12)



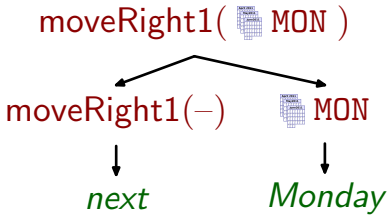


Training

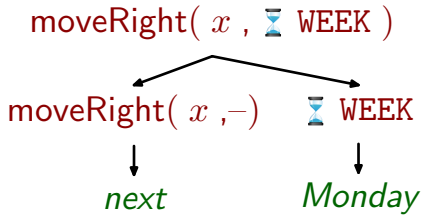
For each example:

Get k-best parses for phrase

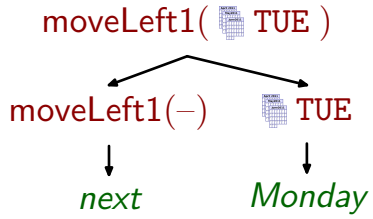
((*next Monday* , 2013-08-05) , 2013-08-12)



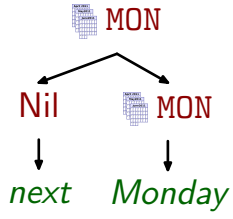
$\phi_1 \cdot w$



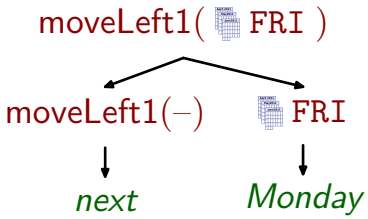
$\phi_2 \cdot w$



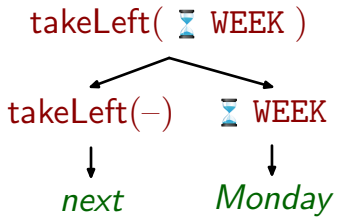
$\phi_3 \cdot w$



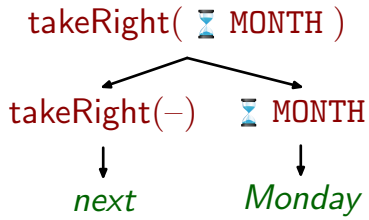
$\phi_4 \cdot w$



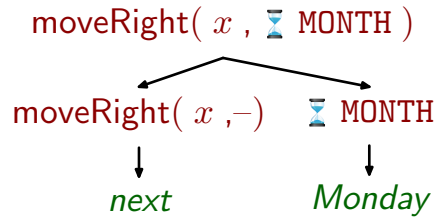
$\phi_5 \cdot w$



$\phi_6 \cdot w$



$\phi_7 \cdot w$



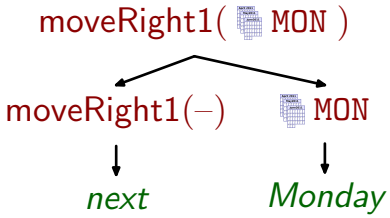
$\phi_8 \cdot w$

Training

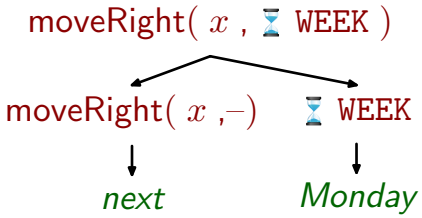
For each example:

- Get k-best parses for phrase
- **Re-weight correct parses as distribution**

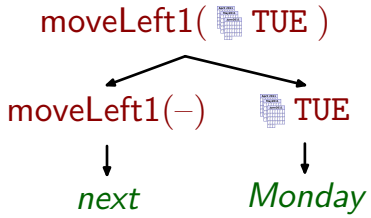
((*next Monday* , 2013-08-05) , 2013-08-12)



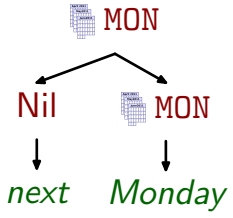
$\phi_1 \cdot w$



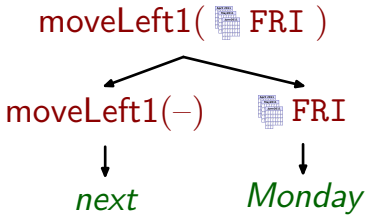
$\phi_2 \cdot w$



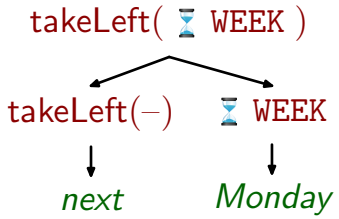
$\phi_3 \cdot w$



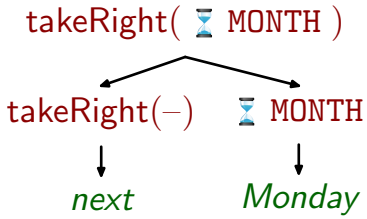
$\phi_4 \cdot w$



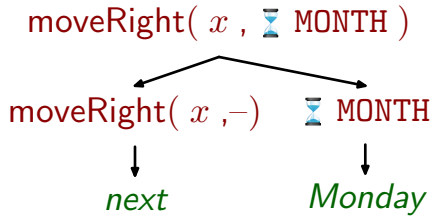
$\phi_5 \cdot w$



$\phi_6 \cdot w$



$\phi_7 \cdot w$



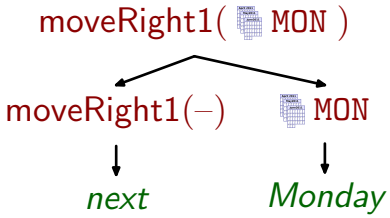
$\phi_8 \cdot w$

Training

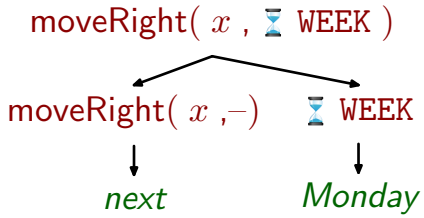
For each example:

- Get k-best parses for phrase
- **Re-weight correct parses as distribution**

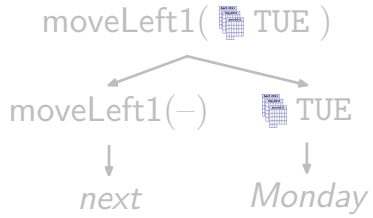
((*next Monday* , 2013-08-05) , 2013-08-12)



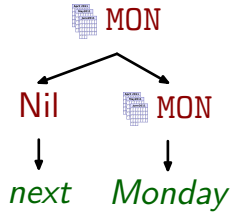
$\phi_1 \cdot w$



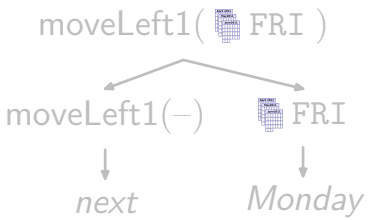
$\phi_2 \cdot w$



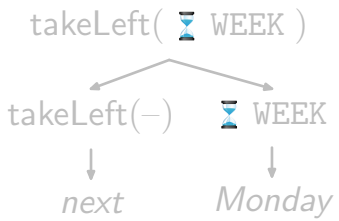
$\phi_3 \cdot w$



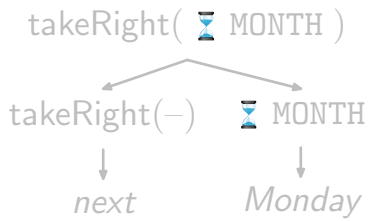
$\phi_4 \cdot w$



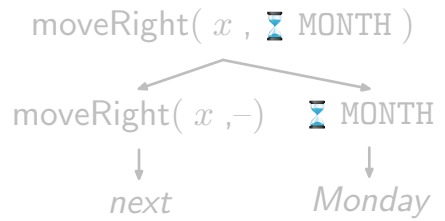
$\phi_5 \cdot w$



$\phi_6 \cdot w$



$\phi_7 \cdot w$



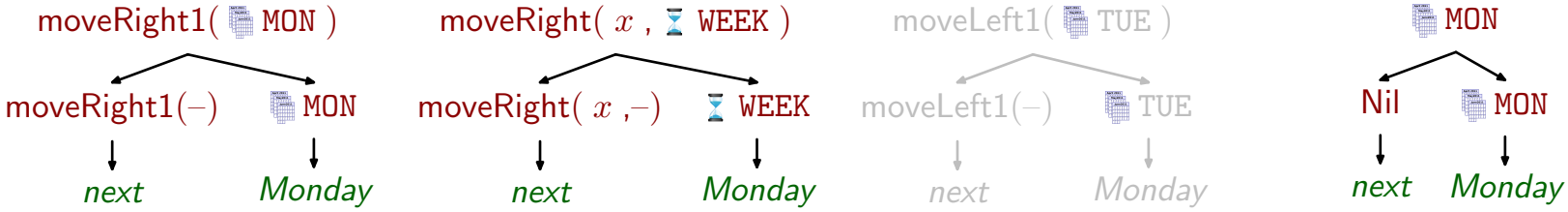
$\phi_8 \cdot w$

Training

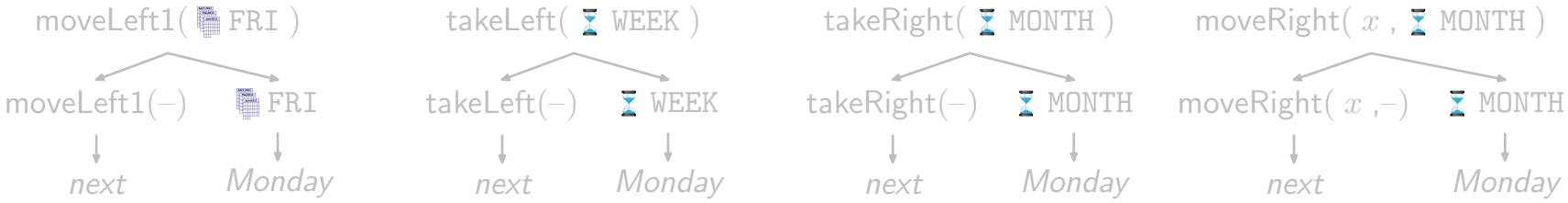
For each example:

- Get k-best parses for phrase
- **Re-weight correct parses as distribution**

((*next Monday* , 2013-08-05) , 2013-08-12)



0.57 0.36 0.00 0.07

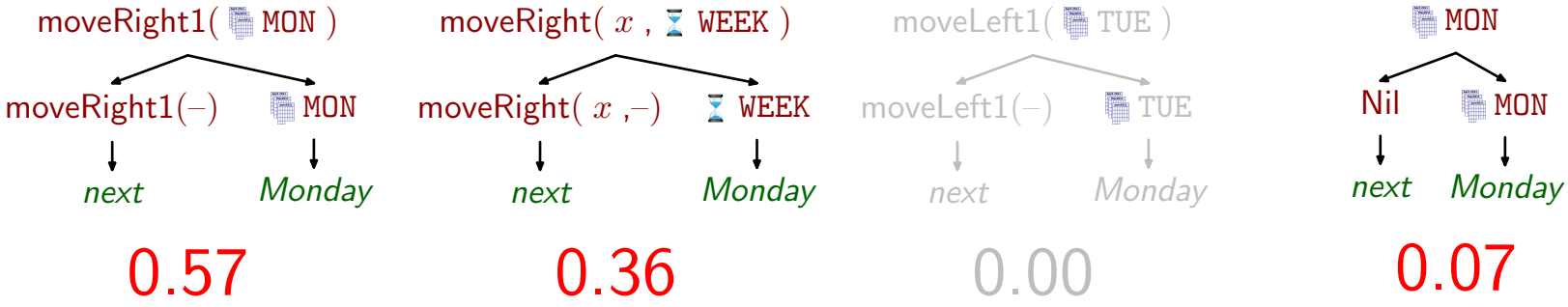


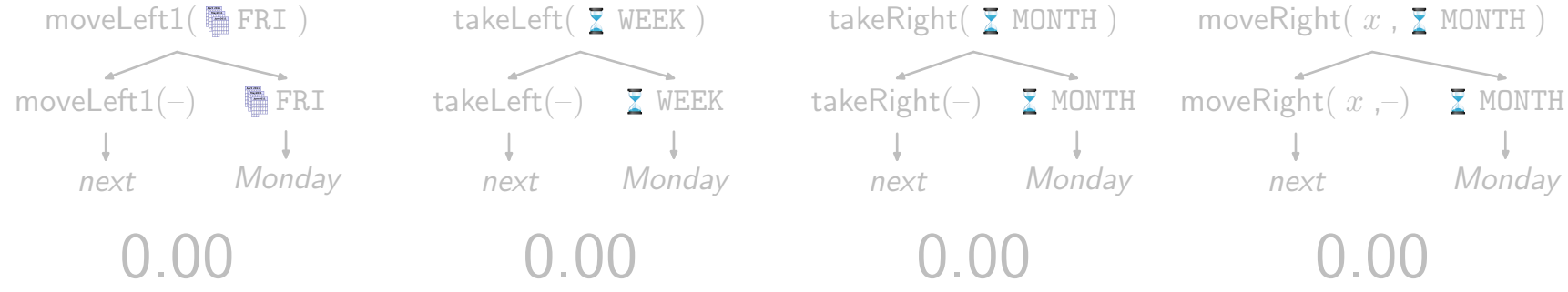
0.00 0.00 0.00 0.00

Training

For each example:

- Get k-best parses for phrase
- Re-weight correct parses as distribution
- **Gradient update on multiclass hinge loss**





Dataset

TempEval2

Newswire annotated for temporal expressions

Dataset

TempEval2

Newswire annotated for temporal expressions

6 languages: English, Spanish, Italian, Chinese, Korean, French

Dataset

Google

TempEval2

Newswire annotated for temporal expressions

6 languages: **English**, **Spanish**, Italian, Chinese, Korean, French

Various sizes: **1052**, **1092**, 523, 659, 247, 206 examples

Dataset

Google

TempEval2

Newswire annotated for temporal expressions

6 languages: **English**, **Spanish**, Italian, Chinese, Korean, French

Various sizes: **1052**, **1092**, 523, 659, 247, 206 examples

Evaluation

Type: Accuracy over result's temporal type

 `August 5, 2013` =  `August 12, 2013`

Dataset

TempEval2

Newswire annotated for temporal expressions

6 languages: **English**, **Spanish**, Italian, Chinese, Korean, French

Various sizes: **1052**, **1092**, 523, 659, 247, 206 examples

Evaluation

Type: Accuracy over result's temporal type

 `August 5, 2013` =  `August 12, 2013`

Value: Accuracy over result's value, if types match

 `August 5, 2013` ≠  `August 12, 2013`

Dataset

TempEval2

Newswire annotated for temporal expressions

6 languages: **English**, **Spanish**, Italian, Chinese, Korean, French

Various sizes: **1052**, **1092**, 523, 659, 247, 206 examples

Evaluation

Type: Accuracy over result's temporal type

 `August 5, 2013` =  `August 12, 2013`

Value: Accuracy over result's value, if types match

 `August 5, 2013` ≠  `August 12, 2013`

Constrained to guess on each example; no contextual cues

Results

English (all expressions; gold detection)

- **GUTime** (Mani and Wilson, 2000)
- **SUTime** (Chang and Manning, 2012)
- **HeidelTime** (Strötgen and Gertz, 2010)
- **ParsingTime** (Angeli *et al.*, 2012)

Results

Google

English (all expressions; gold detection)

- **GUTime** (Mani and Wilson, 2000)
- **SUTime** (Chang and Manning, 2012)
- **HeidelTime** (Strötgen and Gertz, 2010)
- **ParsingTime** (Angeli *et al.*, 2012)

Spanish

- **UC3M** (Vincente-Díez *et al.* 2010)

Results

English (all expressions; gold detection)

System	Type	Value
GUTime	0.80	0.42
SUTime	**0.94**	0.71
HeidelTime	0.85	0.71
ParsingTime	0.88	0.72

Results

English (all expressions; gold detection)

System	Type	Value
GUTime	0.80	0.42
SUTime	**0.94**	0.71
HeidelTime	0.85	0.71
ParsingTime	0.88	0.72
This Work	0.91	**0.76**

Results

English (all expressions; gold detection)

System	Type	Value
GUTime	0.80	0.42
SUTime	**0.94**	0.71
HeidelTime	0.85	0.71
ParsingTime	0.88	0.72
This Work	0.91	**0.76**

Spanish

System	Type	Value
UC3M	0.79	0.72
This Work	**0.92**	**0.76**

Results

Google

All Languages (value accuracy)

Language	# Examples	Train	Test
English	1052	0.81	0.76
Spanish	1092	0.84	0.76

Results

Google

All Languages (value accuracy)

Language	# Examples	Train	Test
English	1052	0.81	0.76
Spanish	1092	0.84	0.76
Italian	523	0.85	0.38
Chinese	659	0.73	0.60
Korean	247	0.67	0.42
French	206	0.76	0.35

First results on remaining 4 languages

Results

Google

All Languages (value accuracy)

Language	# Examples	Train	Test
English	1052	0.81	0.76
Spanish	1092	0.84	0.76
Italian	523	0.85	0.38
Chinese	659	0.73	0.60
Korean	247	0.67	0.42
French	206	0.76	0.35

First results on remaining 4 languages

Test accuracy correlates with training size

Analysis

What are we still missing?

Analysis

29% **Pragmatics**

Next Saturday? → 2013-08-10 or 2013-08-17

Analysis

29% **Pragmatics**

Next Saturday? → 2013-08-10 or 2013-08-17

Last year? → a day? a quarter? a year?

Analysis

29% **Pragmatics**

Next Saturday? → 2013-08-10 or 2013-08-17

Last year? → a day? a quarter? a year?

Hard even for humans

Analysis

29% **Pragmatics**

16% **Type error**

Just given *day*

The past 5 days

Analysis

29% **Pragmatics**

16% **Type error**

10% **Incorrect number**

Drop the number

Spelled out names: *seventeen seventy-six* → 17 76

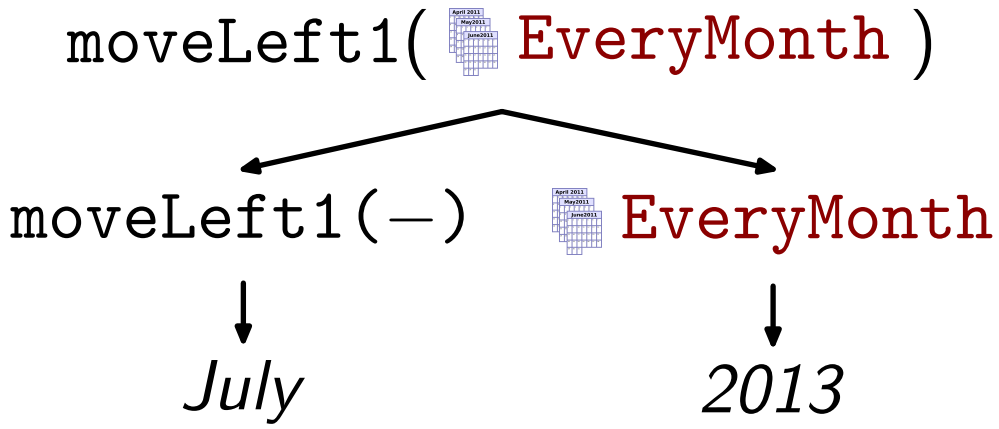
Analysis

29% **Pragmatics**

16% **Type error**

10% **Incorrect number**

7% **Absolute versus relative ambiguity**



Analysis

29%	**Pragmatics**
16%	**Type error**
10%	**Incorrect number**
7%	**Absolute versus relative ambiguity**
19%	**Other parse errors**

Analysis

29%	**Pragmatics**
16%	**Type error**
10%	**Incorrect number**
7%	**Absolute versus relative ambiguity**
19%	**Other parse errors**
19%	**Out of scope**

Analysis

29% **Pragmatics**

16% **Type error**

10% **Incorrect number**

7% **Absolute versus relative ambiguity**

19% **Other parse errors**

19% **Out of scope**

16% Missing context

That time

From time to time

Analysis

29% **Pragmatics**

16% **Type error**

10% **Incorrect number**

7% **Absolute versus relative ambiguity**

19% **Other parse errors**

19% **Out of scope**

- 16% Missing context
  - *That time*
  - *From time to time*
- 3% Bad reference time
- Annotation error
- Reference time is not publication time

Conclusion

Multilingual temporal parsing

- Compositional grammar of time
- Results competitive with state-of-the-art

Conclusion

Google

Multilingual temporal parsing

- Compositional grammar of time
- Results competitive with state-of-the-art

Takeaway points

- Multilingual, with no language-specific tuning
- Rich features over *types* and *values*
- Learns pragmatics of training domain

Thank You!

(Time for questions)