

Discriminative Parsing of the Chinese Treebank Using a Global Discriminative Model



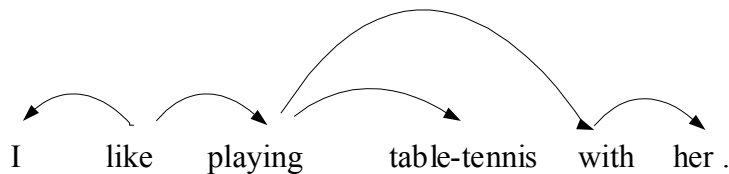
Yue Zhang and Stephen Clark
Oxford University Cambridge University

Introduction

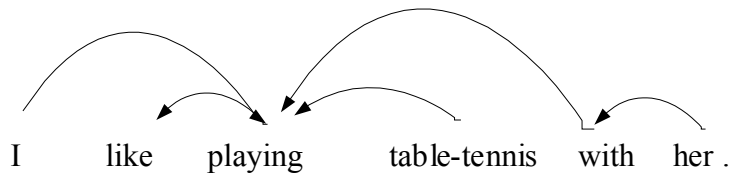
- Transition-based parsing
 - Construct a parse tree by using transition-actions
 - SHIFT, REDUCE, ...
 - Examples:
 - Nivre (2006),
 - Sagae and Lavie (2006),
 - Wang (2006)
 - Contrast to Graph-based parsing

Introduction

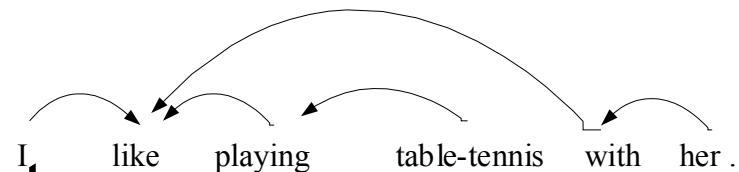
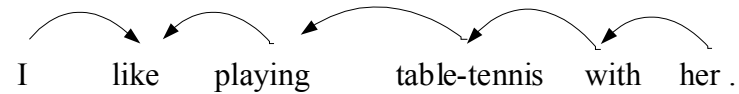
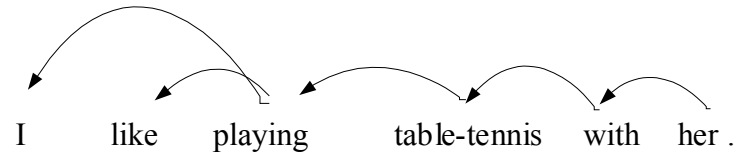
- Graph-based parsing example



...



...



- Score each possible output
- Often use dynamic programming to explore search space

Introduction

- Transition-based parsing example



I like playing table-tennis with her .

Next? ... SHIFT

Introduction

- Transition-based parsing example

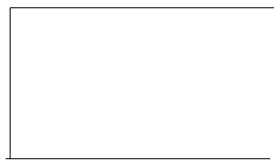
I

like playing table-tennis with her .

Next? ... ARC-LEFT

Introduction

- Transition-based parsing example



I like playing table-tennis with her .

An arrow points from the word "like" down to the character "I".

Next? ... SHIFT

Introduction

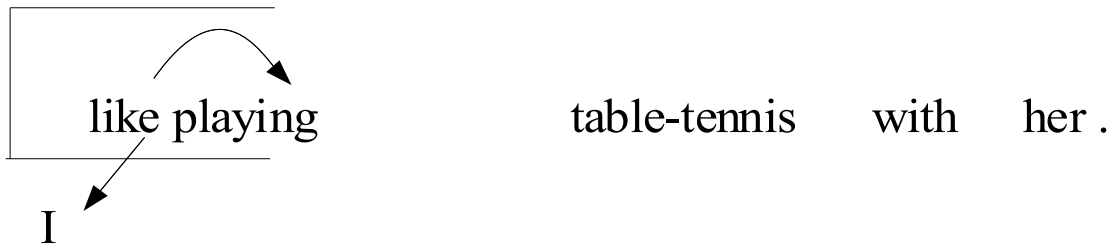
- Transition-based parsing example



Next? ... ARC-RIGHT

Introduction

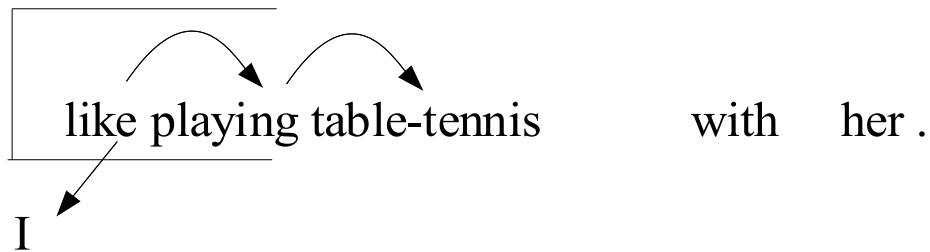
- Transition-based parsing example



Next? ... ARC-RIGHT

Introduction

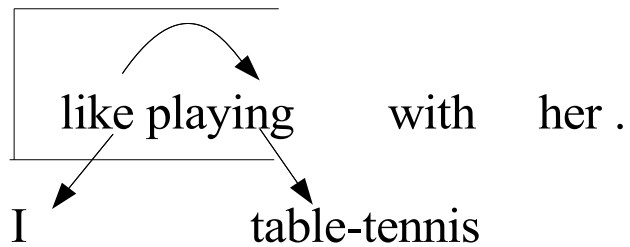
- Transition-based parsing example



Next? ... REDUCE

Introduction

- Transition-based parsing example



Next? ... ARC-RIGHT

Introduction

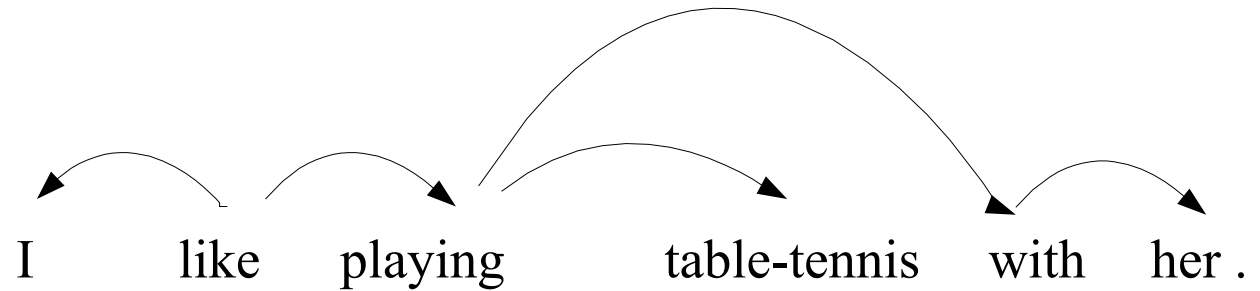
- Transition-based parsing example

...

Next? ...

Introduction

- Transition-based parsing example



Done.

Introduction

- Transition vs. Graph
 - For English Treebank, transition-based behind the best graph-based
 - For Chinese Treebank, transition-based outperform graph-based
 - Constituent parsing (Wang et al, 2006)
 - Dependency (Duan, 2007; Zhang and Clark, 2008)

Introduction

- Our approach is based on the idea of our transition-based dependency parser (EMNLP 2008)
 - Transition-based parsing
 - Global linear model
 - Beam search
- We use Wang et al. (2006)'s shift-reduce transition-based process

Introduction

- Wang et al. (2006) is local
 - Deterministic parsing
 - SVM training (locally trained by action)
- Ours is global, idea from our dep. parser
 - Beam-search
 - Perceptron training, global linear models
- Achieved better accuracy with gold POS; comparable accuracy with auto POS

The shift-reduce parsing process

- A state item = a pair <stack, queue>
 - Stack: holds the partial parse trees already built
 - Queue: holds the incoming words with POS

The shift-reduce parsing process

- Transition actions
 - SHIFT
-

stack



queue

NR 布朗 VV 访问 NR 上海

The shift-reduce parsing process

- Transition actions
 - SHIFT
-

stack

NR 布朗

queue

VV 访问 NR 上海

The shift-reduce parsing process

- Transition actions
 - REDUCE-UNARY-X
-

stack

NR 布朗

queue

VV 访问 NR 上海

The shift-reduce parsing process

- Transition actions
 - REDUCE-UNARY-X
-

stack



NR 布朗

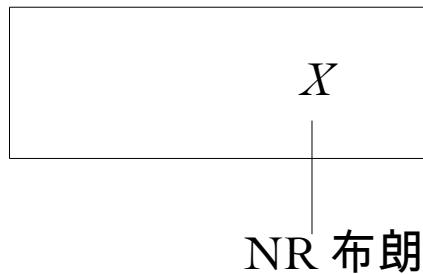
queue

VV 访问 NR 上海

The shift-reduce parsing process

- Transition actions
 - REDUCE-UNARY- X
-

stack

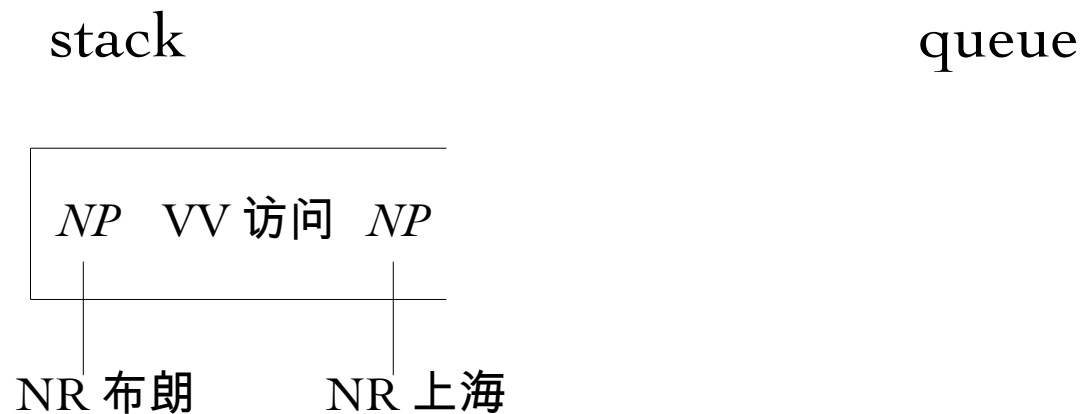


queue

VV 访问 NR 上海

The shift-reduce parsing process

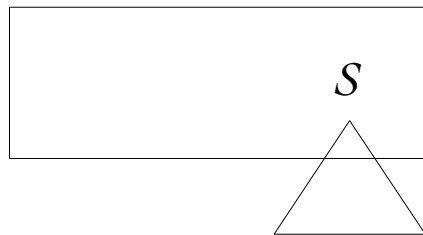
- Transition actions
 - REDUCE-BINARY-{L/R}-X



The shift-reduce parsing process

- Transition actions
 - TERMINATE

stack



queue

The shift-reduce parsing process

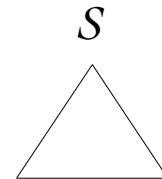
- Transition actions
 - TERMINATE
-

stack



queue

ans



The shift-reduce parsing process

- Example

- SHIFT

stack



queue

NR 布朗 VV 访问 NR 上海

The shift-reduce parsing process

- Example

- REDUCE-UNARY-NP

stack

NR 布朗

queue

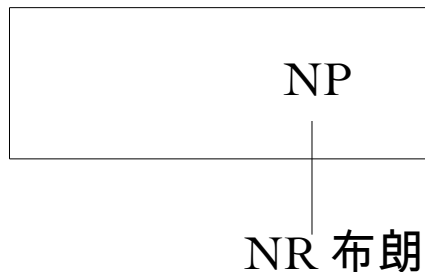
VV 访问 NR 上海

The shift-reduce parsing process

- Example

– SHIFT

stack



queue

VV 访问 NR 上海

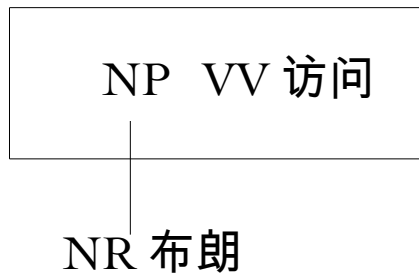
The shift-reduce parsing process

- Example

- SHIFT

stack

queue



NR 上海

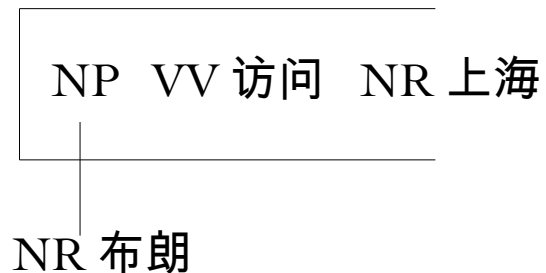
The shift-reduce parsing process

- Example

- REDUCE-UNARY-NP

stack

queue



The shift-reduce parsing process

- Example

- REDUCE-BINARY-L-VP

stack

queue



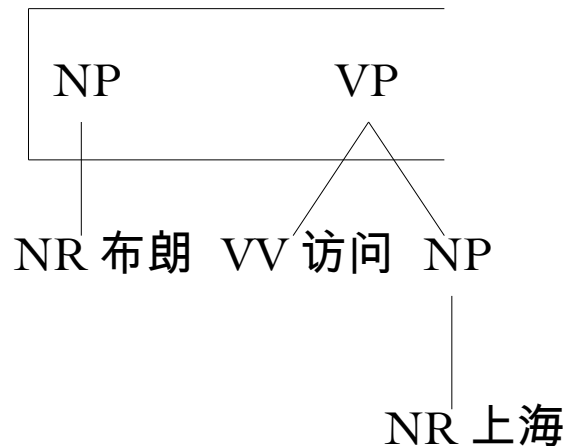
The shift-reduce parsing process

- Example

- REDUCE-BINARY-R-IP

stack

queue



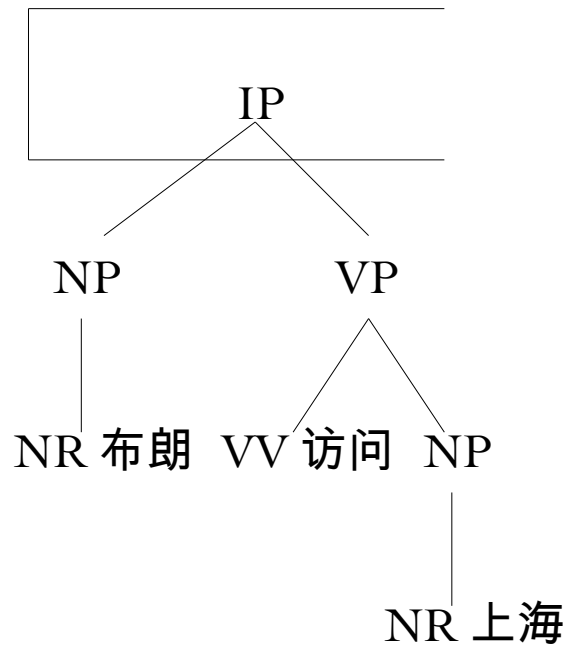
The shift-reduce parsing process

- Example

– TERMINATE

stack

queue



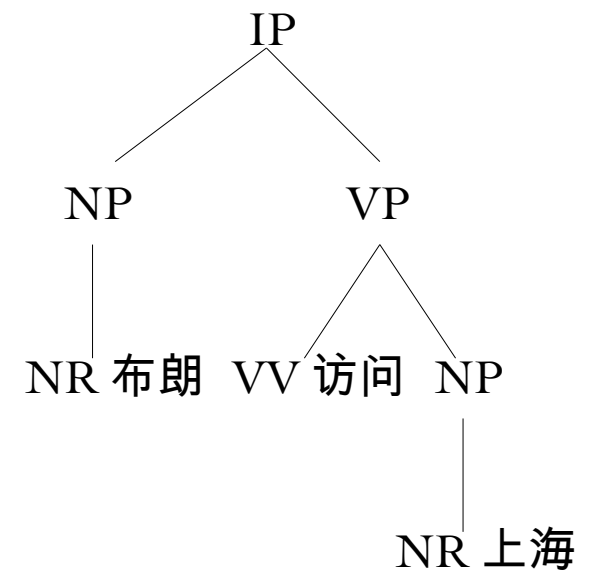
The shift-reduce parsing process

- Example

stack



queue

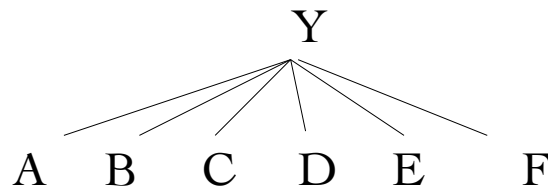


Grammar binarization

- The shift-reduce parser require binarized trees
- Chinese Treebank trees are not binarized
- CTB \leftrightarrow Parser
 - Binarize CTB data to make training data
 - Unbinarize parser output back to CTB format
 - Reversible

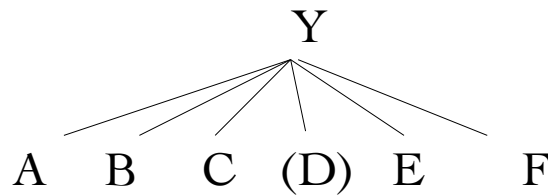
Grammar binarization

- The binarization process
 - Find head
 - Binarize left nodes
 - Binarize right nodes



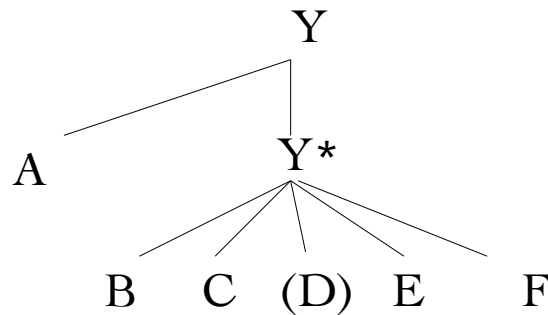
Grammar binarization

- The binarization process
 - Find head
 - Binarize left nodes
 - Binarize right nodes



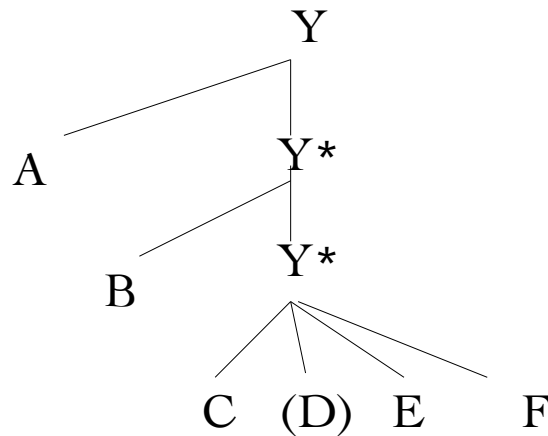
Grammar binarization

- The binarization process
 - Find head
 - Binarize left nodes
 - Binarize right nodes



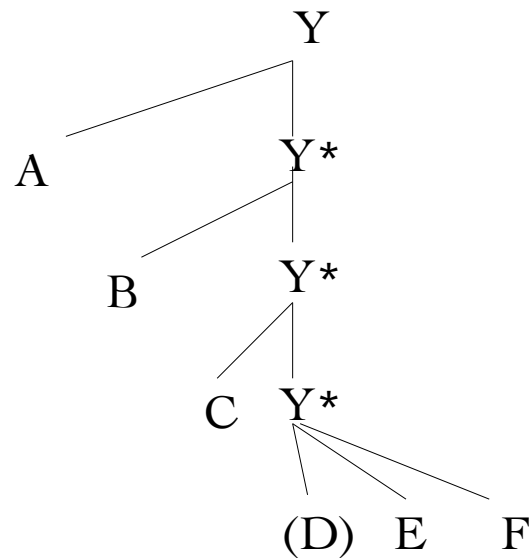
Grammar binarization

- The binarization process
 - Find head
 - Binarize left nodes
 - Binarize right nodes



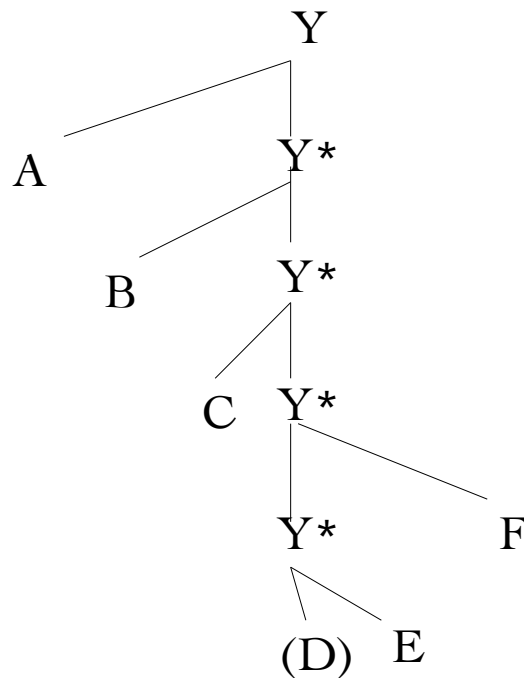
Grammar binarization

- The binarization process
 - Find head
 - Binarize left nodes
 - Binarize right nodes



Grammar binarization

- The binarization process
 - Find head
 - Binarize left nodes
 - Binarize right nodes



The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$

Initial item
stack: empty
queue: input

The statistical parser

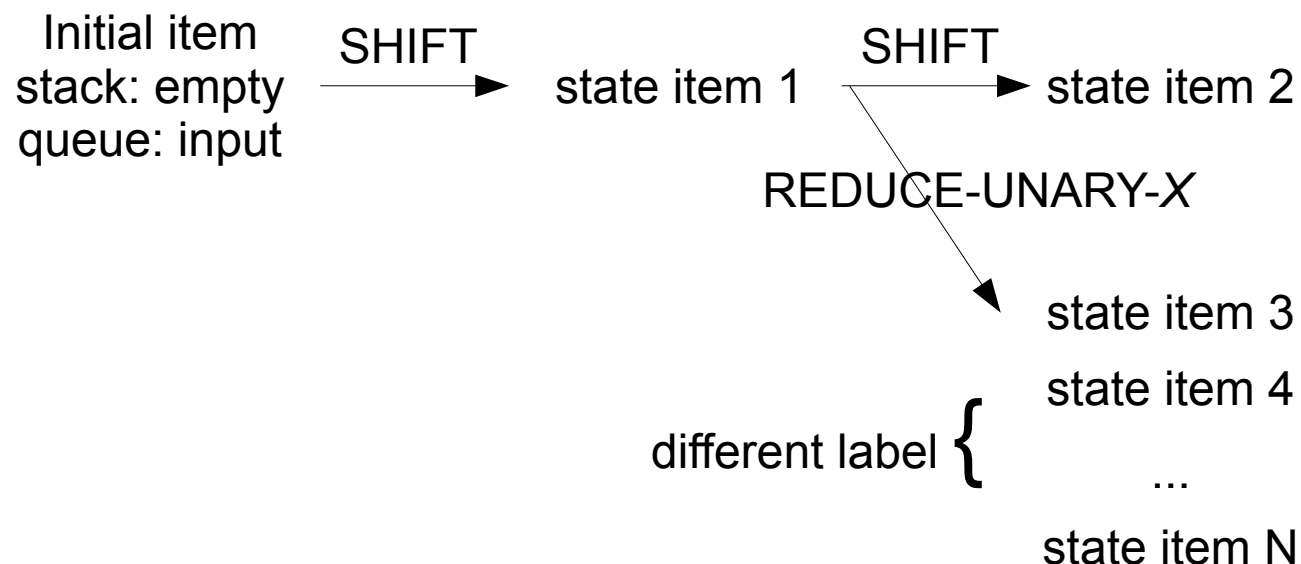
- Beam-search decoding
 - Deterministic parsing: $B=1$

Initial item
stack: empty
queue: input

SHIFT → state item 1

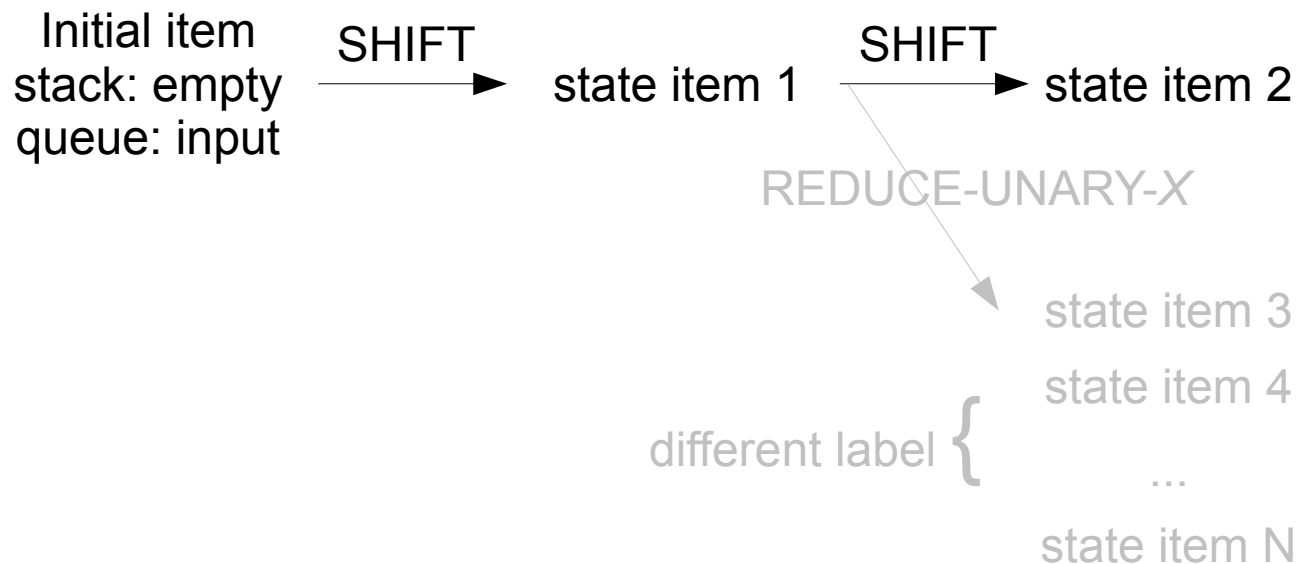
The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$



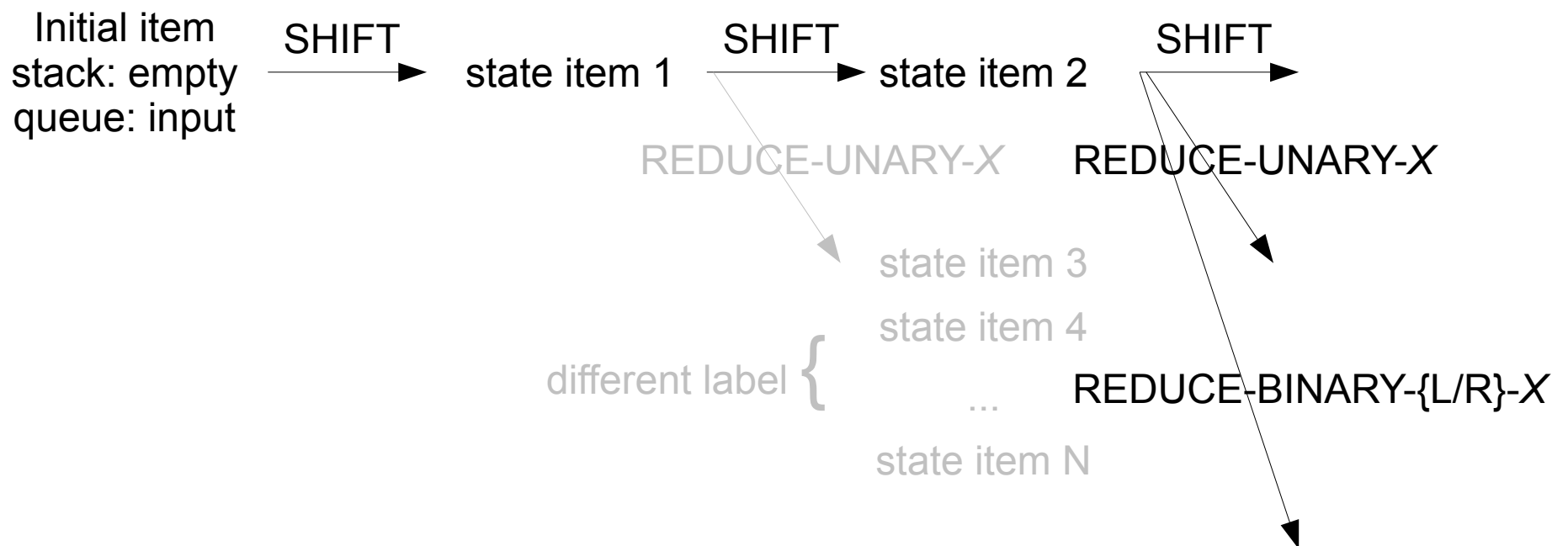
The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$



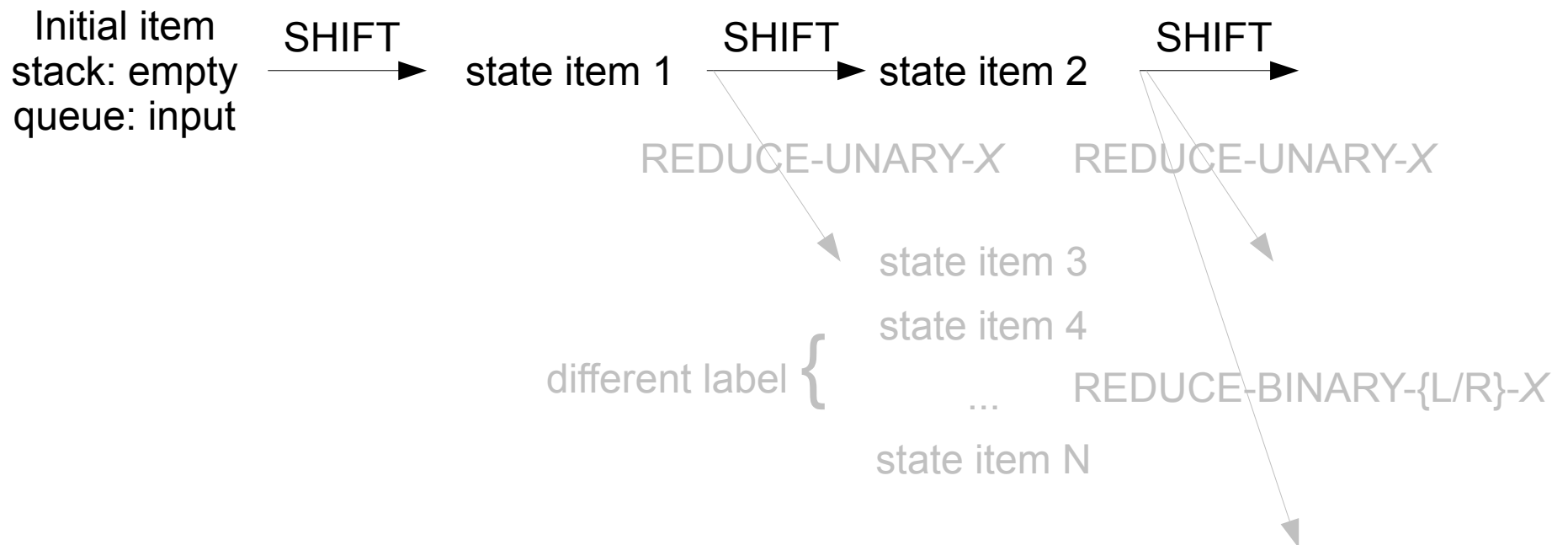
The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$



The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$



The statistical parser

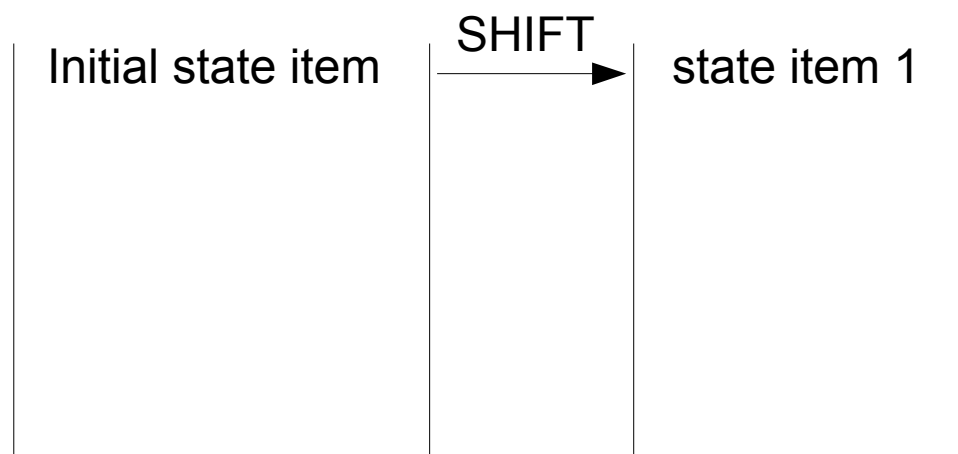
- Beam-search decoding
 - Deterministic parsing: $B=1$
 - Beam-search: $B>1$

Initial state item



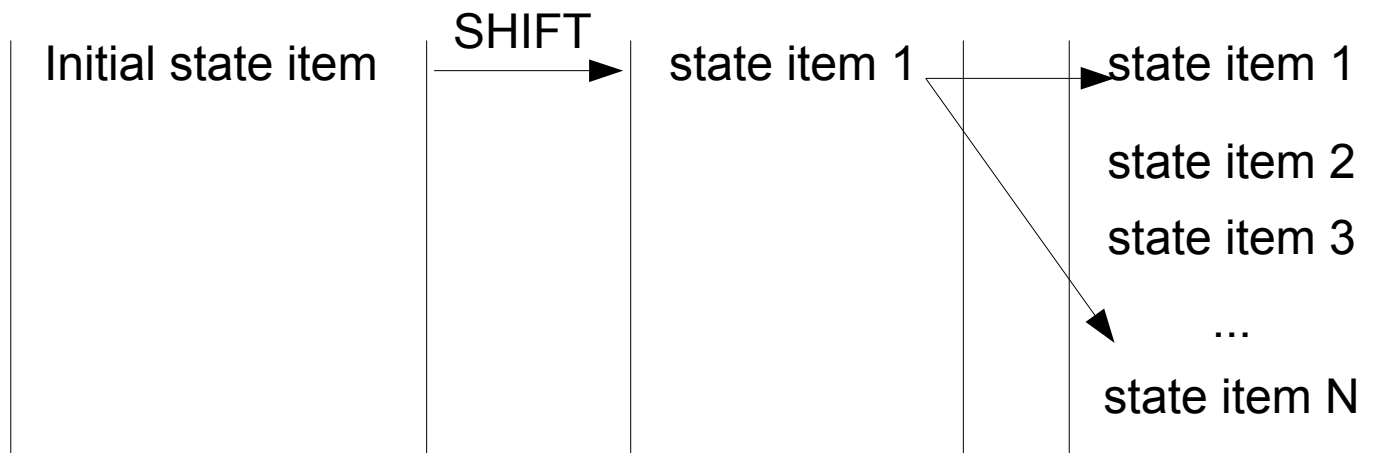
The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$
 - Beam-search: $B>1$



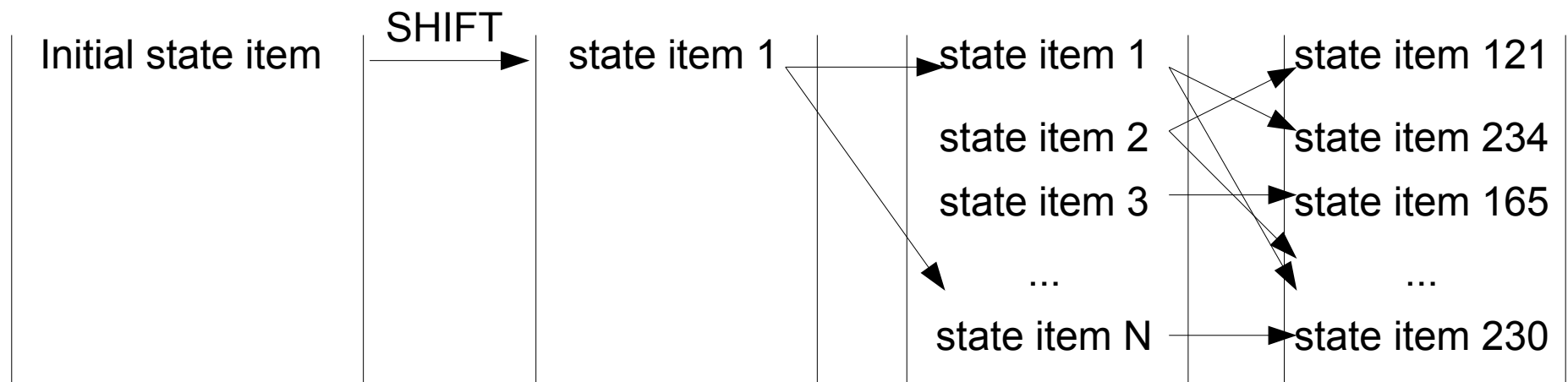
The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$
 - Beam-search: $B>1$



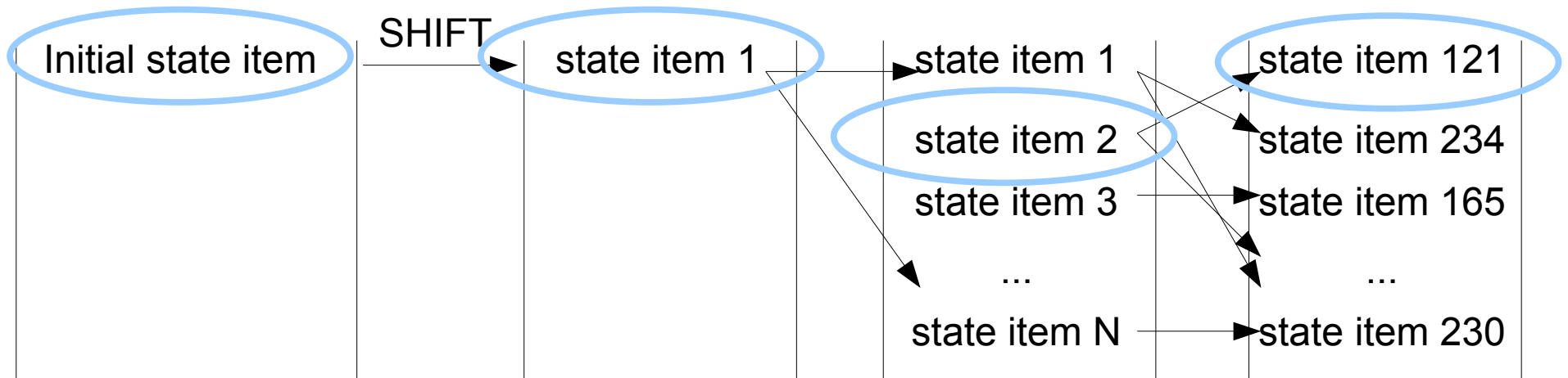
The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$
 - Beam-search: $B>1$



The statistical parser

- Beam-search decoding
 - Deterministic parsing: $B=1$
 - Beam-search: $B>1$



The statistical parser

- Discriminative model
 - Global linear model
$$Score(Y) = \Phi(Y) \cdot \vec{w}$$
 - $\Phi(Y)$ is the global feature vector
 - Features extracted from all actions in state
 - Each elem represents a global count
 - \vec{w} is the parameter vector of the model
 - Trained using the perceptron algorithm

The statistical parser

- The perceptron algorithm

Inputs: training examples (x_i, y_i)

Initialization: set $\vec{w} = \text{all zeros}$

Algorithm:

for $t=1..T, i=1..N$

calculate $z_i = \operatorname{argmax}_{y \in \text{GEN}(x_i)} \Phi(y) \cdot \vec{w}$

if $z_i \neq y_i$

$\vec{w} = \vec{w} + \Phi(y_i) - \Phi(z_i)$

Output: \vec{w}

(t – training iteration; i – training example.)

The statistical parser

- Averaging parameters

- Purpose is reducing overfitting
- Averaged parameter vector

N training examples and T training iterations

$$\vec{y} = \frac{1}{NT} \sum_{i=1..N; t=1..T} \vec{w}^{n,t}$$

- It is used instead of the non-averaged vector

The statistical parser

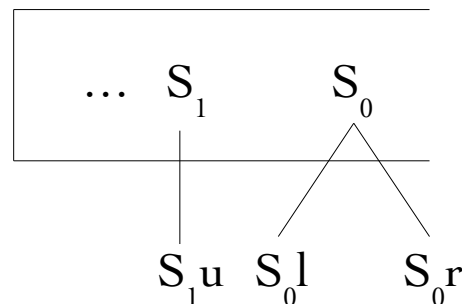
- Early update
 - Stop parsing when no state item in the agenda is correct
 - Update parameters using the current top state item as the parser output, and the corresponding gold-standard state item as the standard parse
 - “Corresponding” means the same number of transitions

The statistical parser

- Features

- Extracted from top nodes from the stack S_0, S_1, S_2, S_3 , the left and right or single child of S_0 and S_1 , and the first words from the queue N_0, N_1, N_2, N_3 .

stack



queue

$N_0 \dots$

The statistical parser

- Features
 - Manually combine word and constituent information
 - Unigrams

*S_{0tc}, S_{0wc}, S_{1tc}, S_{1wc},
S_{2tc}, S_{2wc}, S_{3tc}, S_{3wc},
N_{0wt}, N_{1wt}, N_{2wt}, N_{3wt},
S_{0lwc}, S_{0rwc}, S_{0uwc},
S_{1lwc}, S_{1rwc}, S_{1uwc},*

The statistical parser

- Features

- Manually combine of word and constituent information

- Bigrams

*S₀wS₁w, S₀wS₁c, S₀cS₁w, S₀cS₁c,
S₀wN₀w, S₀wN₀t, S₀cN₀w, S₀cN₀t,
N₀wN₁w, N₀wN₁t, N₀tN₁w, N₀tN₁t,
S₁wN₀w, S₁wN₀t, S₁cN₀w, S₁cN₀t,*

The statistical parser

- Features

- Manually combine of word and constituent information

- Trigrams

$S_0cS_1cS_2c, S_0wS_1cS_2c,$

$S_0cS_1wS_2c, S_0cS_1cS_2w,$

$S_0cS_1cN_0t, S_0wS_1cN_0t,$

$S_0cS_1wN_0t, S_0cS_1cN_0w$

The statistical parser

- Features

- Special features

- Brackets

- Forward brackets: (《 “

- Backward brackets:) 》 ”

- b values indicates the status of the last left bracket on stack

- 1 no matching right bracket on stack

- 2 matching right bracket on stack

- 3 matching right bracket used to be on stack

The statistical parser

- Features
 - Special features

- Brackets

S_0wb, S_0cb

$S_0wS_1cb, S_0cS_1wb, S_0cS_1cb$

$S_0wN_0tb, S_0cN_0wb, S_0cN_0tb$

The statistical parser

- Features

- Special features

- Separators

- Punctuation: , . 、 ;

- p values (1,2,3,4) indicate a particular punctuation

- q values indicate the count of any separator between S_0 and S_1

$S_0wp, S_0wcp, S_0wq, S_0wcq,$

$S_1wp, S_1wcp, S_1wq, S_1wcq$

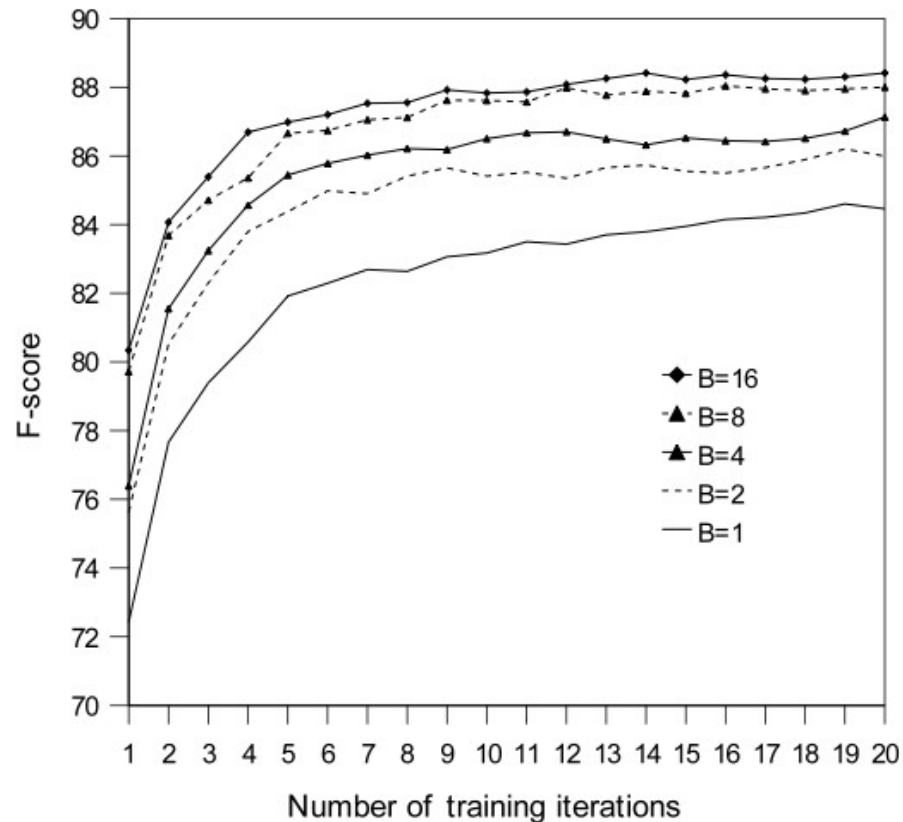
S_0cS_1cp, S_0cS_1cq

Experiments

- Two sets of experiments using CTB2 and CTB5
- Standard evaluation using the evalb tool

Experiments

- Development tests
 - Influence of beam-size



Experiments

- Final test using CTB2
 - Gold-standard POS

Model	<i>LR</i>	<i>LP</i>	<i>F1</i>
Bikel Thesis	80.9%	84.5%	82.7%
Wang 2006 SVM	87.2%	88.3%	87.8%
Wang 2006 Stacked	88.3%	88.1%	88.2%
Our parser	89.4%	90.1%	89.8%

Experiments

- Final test using CTB2
 - Automatic POS

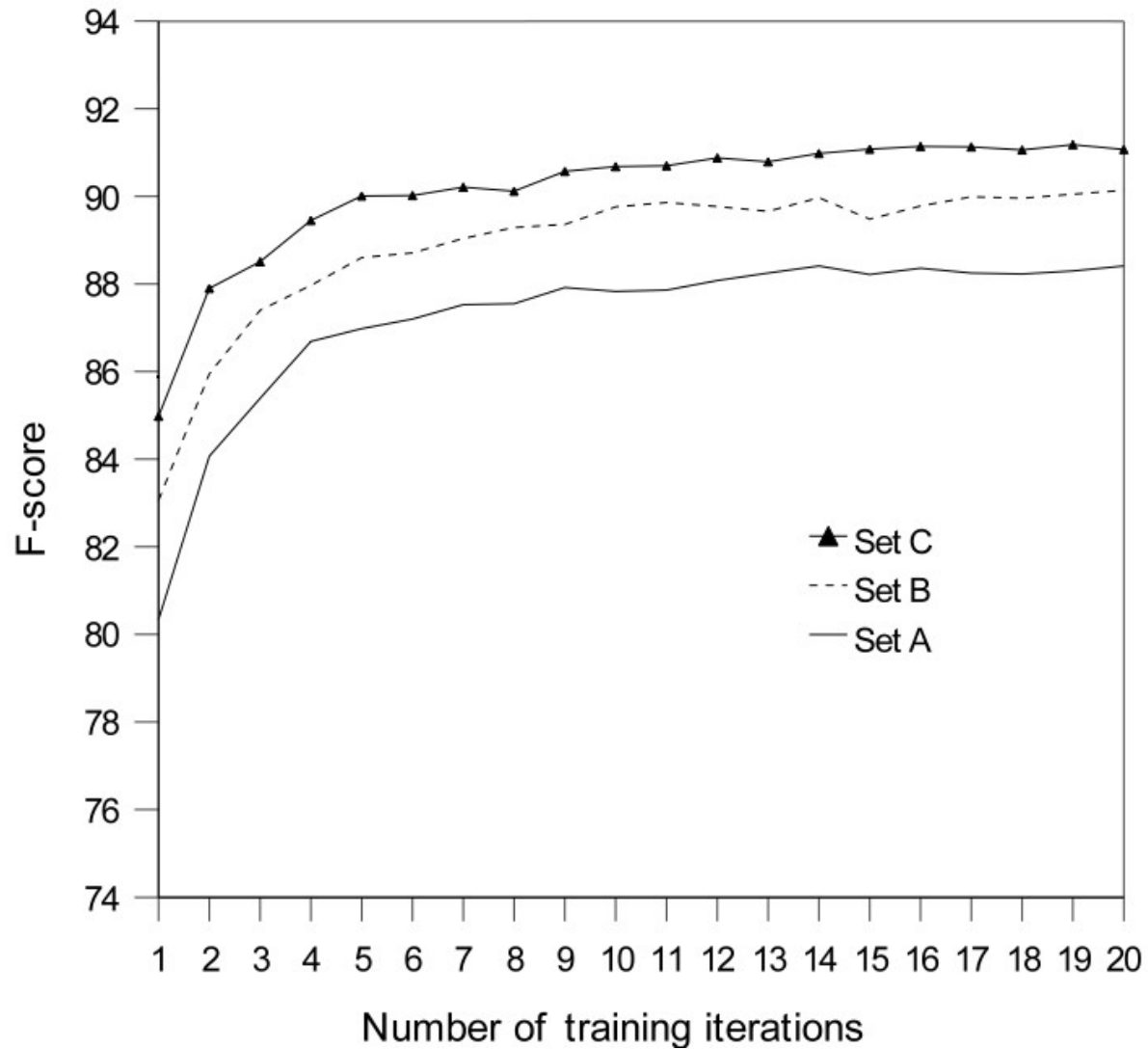
	≤ 40 words				≤ 100 words				Unlimited			
	<i>LR</i>	<i>LP</i>	<i>F1</i>	<i>POS</i>	<i>LR</i>	<i>LP</i>	<i>F1</i>	<i>POS</i>	<i>LR</i>	<i>LP</i>	<i>F1</i>	<i>POS</i>
Bikel 2000	76.8%	77.8%	77.3%	-	73.3%	74.6%	74.0%	-	-	-	-	-
Levy 2003	79.2%	78.4%	78.8%	-	-	-	-	-	-	-	-	-
Xiong 2005	78.7%	80.1%	79.4%	-	-	-	-	-	-	-	-	-
Bikel Thesis	78.0%	81.2%	79.6%	-	74.4%	78.5%	76.4%	-	-	-	-	-
Chiang 2002	78.8%	81.1%	79.9%	-	75.2%	78.0%	76.6%	-	-	-	-	-
Wang 2006 SVM	78.1%	81.1%	79.6%	92.5%	75.5%	78.5%	77.0%	92.2%	75.0%	78.0%	76.5%	92.1%
Wang 2006 Stacked	79.2%	81.1%	80.1%	92.5%	76.7%	78.4%	77.5%	92.2%	76.2%	78.0%	77.1%	92.1%
Our parser	80.2%	80.5%	80.4%	93.5%	76.5%	77.7%	77.1%	93.1%	76.1%	77.4%	76.7%	93.0%

Experiments

- Development test
 - Change the size of training data
 - Motivation – CTB2 small
 - Motivation – accuracy fluctuation
 - Motivation – more data increases accuracy

	Sections	Sentences	Words
Set A	001–270	3,484	84,873
Set B	Set A; 400–699	6,567	161,893
Set C	Set B; 700–931	9,707	236,051

Experiments



Experiments

- Final test on CTB5
 - Reporting accuracies

≤ 40 words				Unlimited			
<i>LR</i>	<i>LP</i>	<i>F1</i>	<i>POS</i>	<i>LR</i>	<i>LP</i>	<i>F1</i>	<i>POS</i>
87.9%	87.5%	87.7%	100%	86.9%	86.7%	86.8%	100%
80.2%	79.1%	79.6%	94.1%	78.6%	78.0%	78.3%	93.9%

- Extracting dependencies

	Non-root	Root	Complete
Zhang-Clark 2008	86.21%	76.26%	34.41%
This paper	86.95%	79.19%	36.08%

Conclusion and future work

- Conclusion
 - A global discriminative model for transition parser
 - Comparable accuracy with Wang et al. (2006)
- Future work
 - Investigate how the accuracy with auto POS could be improved for our parser
 - The combination of graph-based features into the transition-based parser

Thank you!

<http://sourceforge.net/projects/zpar/>