

# Arc-Standard Algorithm

Two stacks:

- Stack with tokens that still have to be processed (Input Stack)
- Stack with tokens that already have been processed (Processed Stack)

One set:

- Set of predicted dependency pairs

Strategy used in MaltParser, cf. <http://www.maltparser.org/>

# Arc-Eager Parsing Algorithm

- Initialize the processed stack with the “0” token on top of it
- Initialize the input stack by putting all tokens (1 .. n) in the sentence on it
- While the input stack is not empty:
  - predict the probabilities of operations
  - remove not permissible operations
  - execute the most probable operation

# Arc-Eager Parser Operations

- **Leftarc**
  - Makes the top token of the **input stack** the head of the top token of the **processed stack**
  - Pops the top token from the **processed stack**
- **Rightarc**
  - Makes the top token of the **processed stack** the head of the top token of the **input stack**
  - Moves the top token of the **input stack** on top of the **processed stack**
- **Reduce**
  - Pops the top token from the **processed stack**, *if it already has been attached to all of its dependents to the right*
- **Shift**
  - Moves the top token of the **input stack** on top of the **processed stack**

# Parsing Input

1	He	PRP	-	-
2	said	VBD	-	-
3	he	PRP	-	-
4	will	MD	-	-
5	now	RB	-	-
6	consider	VB	-	-
7	those	DT	-	-
8	offers	NNS	-	-
9	.	.	-	-

# Parsing Output / Training Instance

1	He	PRP	2	SBJ
2	said	VBD	0	ROOT
3	he	PRP	4	SBJ
4	will	MD	2	OBJ
5	now	RB	4	TMP
6	consider	VB	4	VC
7	those	DT	8	NMOD
8	offers	NNS	6	OBJ
9	.	.	2	P

# Notation

wf := word form

p := POS tag

dt := dependency type

pr := processed stack

in := input stack

pn := n+1 element in stack

Examples:

pinp1=VBD ::= POS tag of second top element in input stack

dtpr=SBJ ::= dependency type of top element of processed stack

# Arc-Eager Training Example

[ ]

[0]

[9, 8, 7, 6, 5, 4, 3, 2, 1]

wfin=He pin=PRP pinp1=VBD wfinp2=he pinp2=PRP pinp3=MD pinp4=RB **shift**

0 He<sub>1</sub> said<sub>2</sub> he<sub>3</sub> will<sub>4</sub> now.<sub>5</sub> consider<sub>6</sub> those<sub>7</sub> offers<sub>8</sub> .<sub>9</sub>

# Arc-Eager Training Example

[ ]

[0, 1]

[9, 8, 7, 6, 5, 4, 3, 2]

wfin=said pin=VBD wfpr=He ppr=PRP dtpr=SBJ pinp1=PRP wfinp2=will pinp2=MD  
pinp3=RB pinp4=VB **leftarc**

0 He<sub>1</sub> said<sub>2</sub> he<sub>3</sub> will<sub>4</sub> now.<sub>5</sub> consider<sub>6</sub> those<sub>7</sub> offers<sub>8</sub> .<sub>9</sub>

# Arc-Eager Training Example

[(1,2)]

[0]

[9, 8, 7, 6, 5, 4, 3, 2]

wfin=said pin=VBD pinp1=PRP wfinp2=will pinp2=MD pinp3=RB pinp4=VB **rightarc**



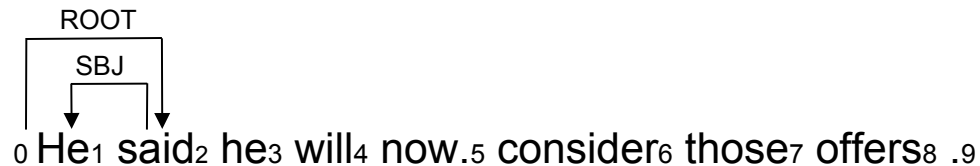
# Arc-Eager Training Example

[(1,2), (2,0)]

[0, 2]

[9, 8, 7, 6, 5, 4, 3]

wfin=he pin=PRP wfpr=said ppr=VBD dtpr=ROOT pinp1=MD wfinp2=now pinp2=RB  
pinp3=VB pinp4=DT **shift**



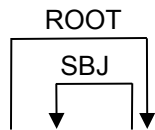
# Arc-Eager Training Example

[(1,2), (2,0)]

[0, 2, 3]  
5, 4]

[9, 8, 7, 6,

wfin=will pin=MD wfpr=he ppr=PRP dtpr=SBJ wfprp1=will pprp1=MD pinp1=RB  
wfinp2=consider pinp2=VB pinp3=DT pinp4=NNS **leftarc**



0 He<sub>1</sub> said<sub>2</sub> he<sub>3</sub> will<sub>4</sub> now.<sub>5</sub> consider<sub>6</sub> those<sub>7</sub> offers<sub>8</sub> .<sub>9</sub>

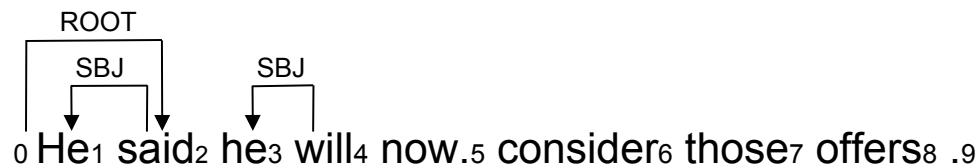
# Arc-Eager Training Example

[(1,2), (2,0), (3,4)]

[0, 2]

[9, 8, 7, 6, 5, 4]

wfin=will pin=MD wfpr=said ppr=VBD dtpr=ROOT pinp1=RB wfinp2=consider pinp2=VB  
pinp3=DT pinp4=NNS **rightarc**



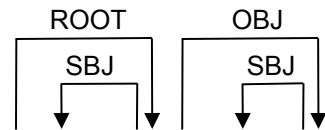
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2)]

[0, 2, 4]  
6, 5]

[9, 8, 7,

wfin=now pin=RB wfpr=will ppr=MD dtpr=OBJ wfprp1=now pprp1=RB pinp1=VB  
wfinp2=those pinp2=DT pinp3=NNS pinp4=. **rightarc**



0 He<sub>1</sub> said<sub>2</sub> he<sub>3</sub> will<sub>4</sub> now<sub>.5</sub> consider<sub>6</sub> those<sub>7</sub> offers<sub>8</sub> .<sub>9</sub>

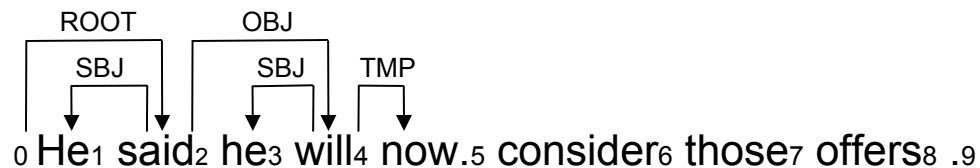
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4)]

[0, 2, 4, 5]

[9, 8, 7, 6]

wfin=consider pin=VB wfpr=now ppr=RB dtpr=TMP wfprp1=consider pprp1=VB  
pinp1=DT wfinp2=offers pinp2=NNS pinp3=. **reduce**



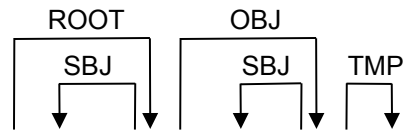
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4)]

[0, 2, 4]  
7, 6]

[9, 8,

wfin=consider pin=VB wfpr=will ppr=MD dtpr=OBJ wfprp1=now pprp1=RB pinp1=DT  
wfinp2=offers pinp2=NNS pinp3=. **rightarc**



0 He<sub>1</sub> said<sub>2</sub> he<sub>3</sub> will<sub>4</sub> now<sub>.5</sub> consider<sub>6</sub> those<sub>7</sub> offers<sub>8</sub> .<sub>9</sub>

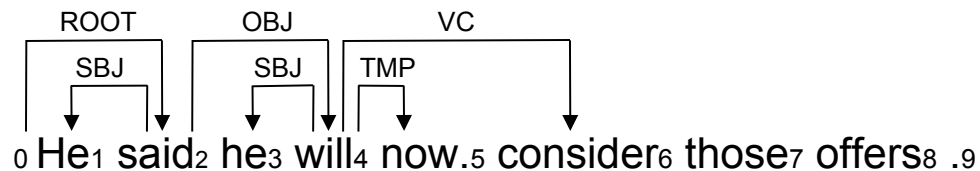
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4)]

[0, 2, 4, 6]

[9, 8, 7]

wfin=those pin=DT wfpr=consider ppr=VB dtpr=VC wfprp1=those pprp1=DT pinp1=NNS  
wfinp2=. pinp2=. **shift**



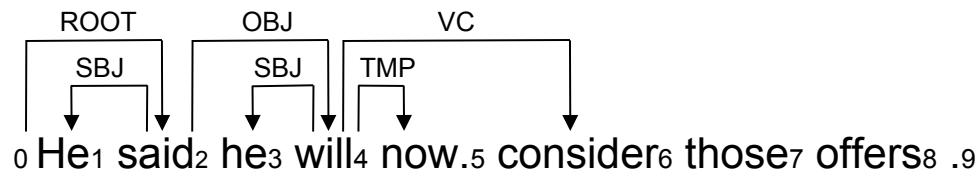
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4)]

[0, 2, 4, 6, 7]

[9, 8]

wfin=offers pin=NNS wfpr=those ppr=DT dtpr=NMOD wfprp1=offers pprp1=NNS pinp1=.  
**leftarc**



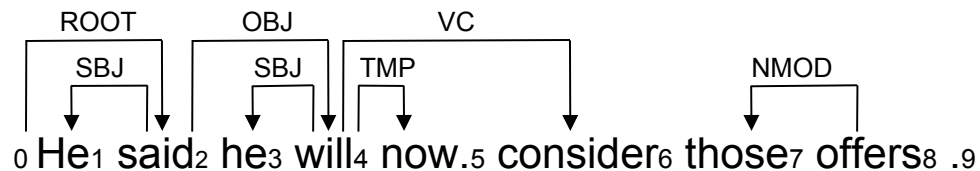
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4), (7,8)]

[0, 2, 4, 6]

[9, 8]

wfin=offers pin=NNS wfpr=consider ppr=VB dtpr=VC wfprp1=those pprp1=DT pinp1=.  
**rightarc**



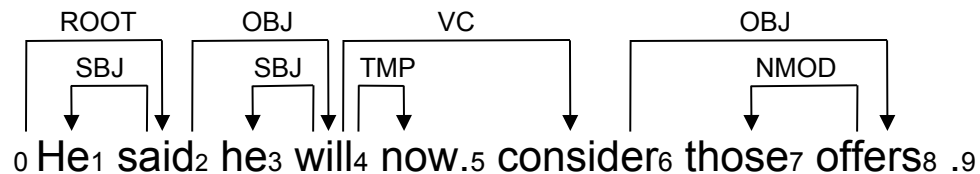
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4), (7,8), (8,6)]

[0, 2, 4, 6, 8]

[9]

wfin=. pin=. wfpr=offers ppr=NNS dtpr=OBJ wfprp1=. pprp1=. **reduce**



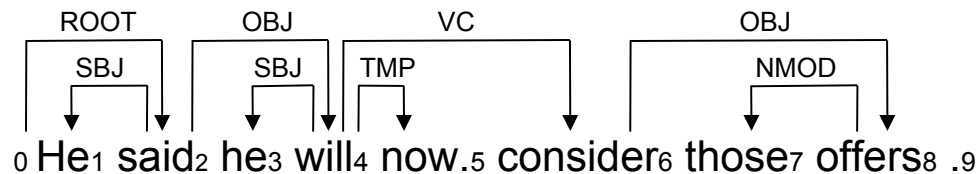
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4), (7,8), (8,6)]

[0, 2, 4, 6]

[9]

wfin=. pin=. wfpr=consider ppr=VB dtpr=VC wfprp1=those pprp1=DT **reduce**



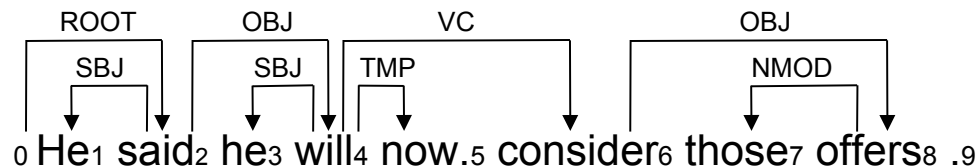
# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4), (7,8), (8,6)]

[0, 2]

[9]

wfin=. pin=. wfpr=said ppr=VBD dtpr=ROOT **rightarc**

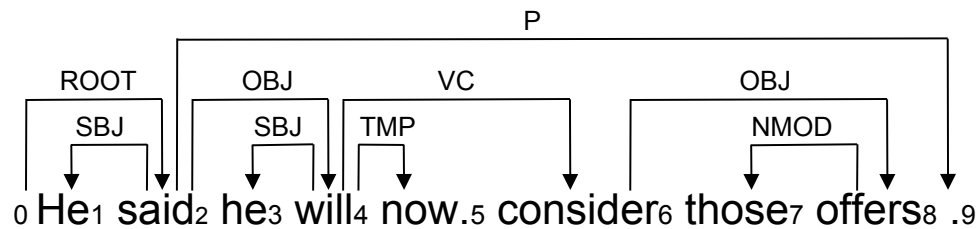


# Arc-Eager Training Example

[(1,2), (2,0), (3,4), (4,2), (5,4), (6,4), (7,8), (8,6), (9,2)]

[0, 2, 9]

[ ]



# Output

wfin=He pin=PRP pinp1=VBD wfinp2=he pinp2=PRP pinp3=MD pinp4=RB shift  
wfin=said pin=VBD wfpr=He ppr=PRP dtpr=SBJ pinp1=PRP wfinp2=will pinp2=MD pinp3=RB  
pinp4=VB leftarc  
wfin=said pin=VBD pinp1=PRP wfinp2=will pinp2=MD pinp3=RB pinp4=VB rightarc  
wfin=he pin=PRP wfpr=said ppr=VBD dtpr=ROOT pinp1=MD wfinp2=now pinp2=RB pinp3=VB  
pinp4=DT shift  
wfin=will pin=MD wfpr=he ppr=PRP dtpr=SBJ wfprp1=will pprp1=MD pinp1=RB wfinp2=consider  
pinp2=VB pinp3=DT pinp4=NNS leftarc  
wfin=will pin=MD wfpr=said ppr=VBD dtpr=ROOT pinp1=RB wfinp2=consider pinp2=VB pinp3=DT  
pinp4=NNS rightarc  
wfin=now pin=RB wfpr=will ppr=MD dtpr=OBJ wfprp1=now pprp1=RB pinp1=VB wfinp2=those  
pinp2=DT pinp3=NNS pinp4=. rightarc  
wfin=consider pin=VB wfpr=now ppr=RB dtpr=TMP wfprp1=consider pprp1=VB pinp1=DT  
wfinp2=offers pinp2=NNS pinp3=. reduce  
wfin=consider pin=VB wfpr=will ppr=MD dtpr=OBJ wfprp1=now pprp1=RB pinp1=DT wfinp2=offers  
pinp2=NNS pinp3=. rightarc  
wfin=those pin=DT wfpr=consider ppr=VB dtpr=VC wfprp1=those pprp1=DT pinp1=NNS wfinp2=.  
pinp2=. shift  
wfin=offers pin=NNS wfpr=those ppr=DT dtpr=NMOD wfprp1=offers pprp1=NNS pinp1=. leftarc  
wfin=offers pin=NNS wfpr=consider ppr=VB dtpr=VC wfprp1=those pprp1=DT pinp1=. rightarc  
wfin=. pin=. wfpr=offers ppr=NNS dtpr=OBJ wfprp1=. pprp1=. reduce  
wfin=. pin=. wfpr=will ppr=MD dtpr=OBJ wfprp1=now pprp1=RB reduce  
wfin=. pin=. wfpr=said ppr=VBD dtpr=ROOT rightarc