CS447: Natural Language Processing
http://courses.engr.illinois.edu/cs447

Lecture 19: Dependency Grammars and Dependency Parsing

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center

Today's lecture

Dependency Grammars
Dependency Treebanks
Dependency Parsing

The popularity of Dependency Parsing

Currently the main paradigm for syntactic parsing.

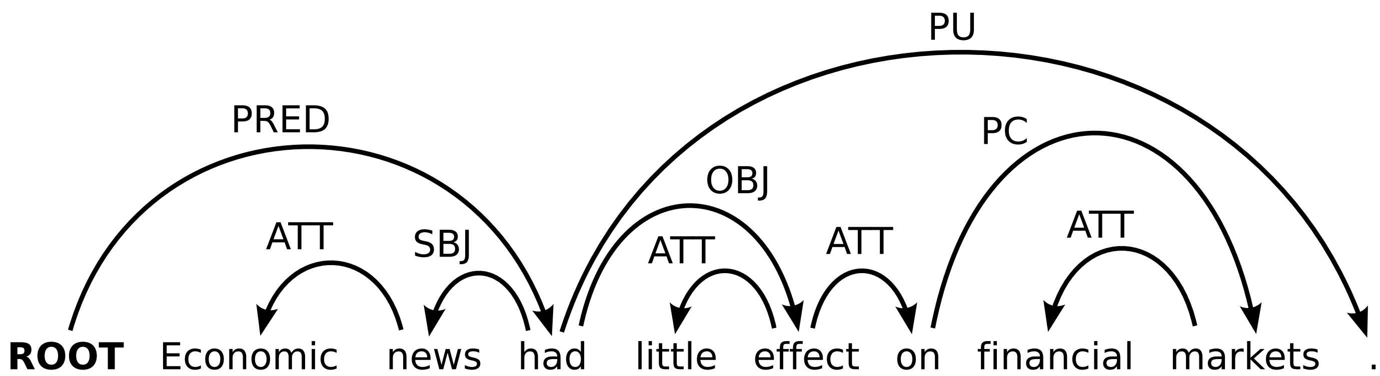
Dependencies are easier to use and interpret
for downstream tasks than phrase-structure trees

Dependencies are more natural for languages with free word order

Lots of dependency treebanks are available

Dependency Grammar

A dependency parse



Dependencies are (labeled) asymmetrical binary relations between two lexical items (words).

had —OBJ—> *effect* [*effect* is the object of *had*]

effect —ATT—> *little* [*little* is at attribute of *effect*]

We typically assume a special ROOT token as word 0

Dependency grammar

Word-word dependencies are a component of many (most/all?) grammar formalisms.

Dependency grammar assumes that syntactic structure consists *only* of dependencies.

Many variants. Modern DG began with Tesniere (1959).

DG is often used for **free word order languages**.

DG is **purely descriptive** (not generative like CFGs etc.), but some formal equivalences are known.

Dependency trees

Dependencies form a graph over the words in a sentence.
This graph is **connected** (every word is a node) and (typically) **acyclic** (no loops).

Single-head constraint:
Every node has at most **one incoming edge.**
Together with connectedness, this implies that the graph is a **rooted tree**.

Different kinds of dependencies

Head-argument: *eat sushi*
Arguments may be obligatory, but can only occur once.
The head alone cannot necessarily replace the construction.

Head-modifier: *fresh sushi*
Modifiers are optional, and can occur more than once.
The head alone can replace the entire construction.

Head-specifier: *the sushi*
Between function words (e.g. prepositions, determiners)
and their arguments. Syntactic head ≠ semantic head

Coordination: *sushi and sashimi*
Unclear where the head is.

There isn't one right dependency grammar

Lots of different ways to to represent particular constructions as dependency trees, e.g.:

Coordination (*eat sushi and sashimi, sell and buy shares*)
Prepositional phrases (*with wasabi*)
Verb clusters (*I will have done this*)
Relative clauses (*the cat I saw caught a mouse*)

Where is the head in these constructions?

Different dependency treebanks use different conventions for these constructions

Dependency Treebanks

Dependency Treebanks

Dependency treebanks exist for many languages:

- Czech
- Arabic
- Turkish
- Danish
- Portuguese
- Estonian
- ....

Phrase-structure treebanks (e.g. the Penn Treebank) can also be translated into dependency trees
(although there might be noise in the translation)

The Prague Dependency Treebank

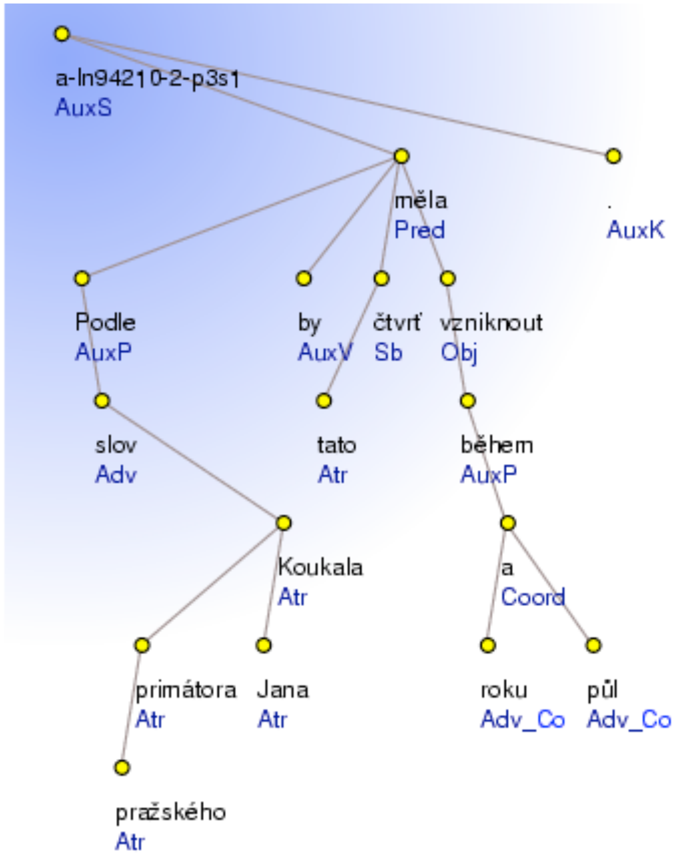
Three levels of annotation:

morphological: [<2M tokens]
Lemma (dictionary form) + detailed analysis
(15 categories with many possible values = 4,257 tags)

surface-syntactic ("analytical"): [1.5M tokens]
Labeled dependency tree encoding grammatical functions
(subject, object, conjunct, etc.)

semantic ("tectogrammatical"): [0.8M tokens]
Labeled dependency tree for predicate-argument structure,
information structure, coreference (not all words included)
(39 labels: agent, patient, origin, effect, manner, etc....)

Examples: analytical level



METU-Sabanci Turkish Treebank

Turkish is an agglutinative language with free word order.

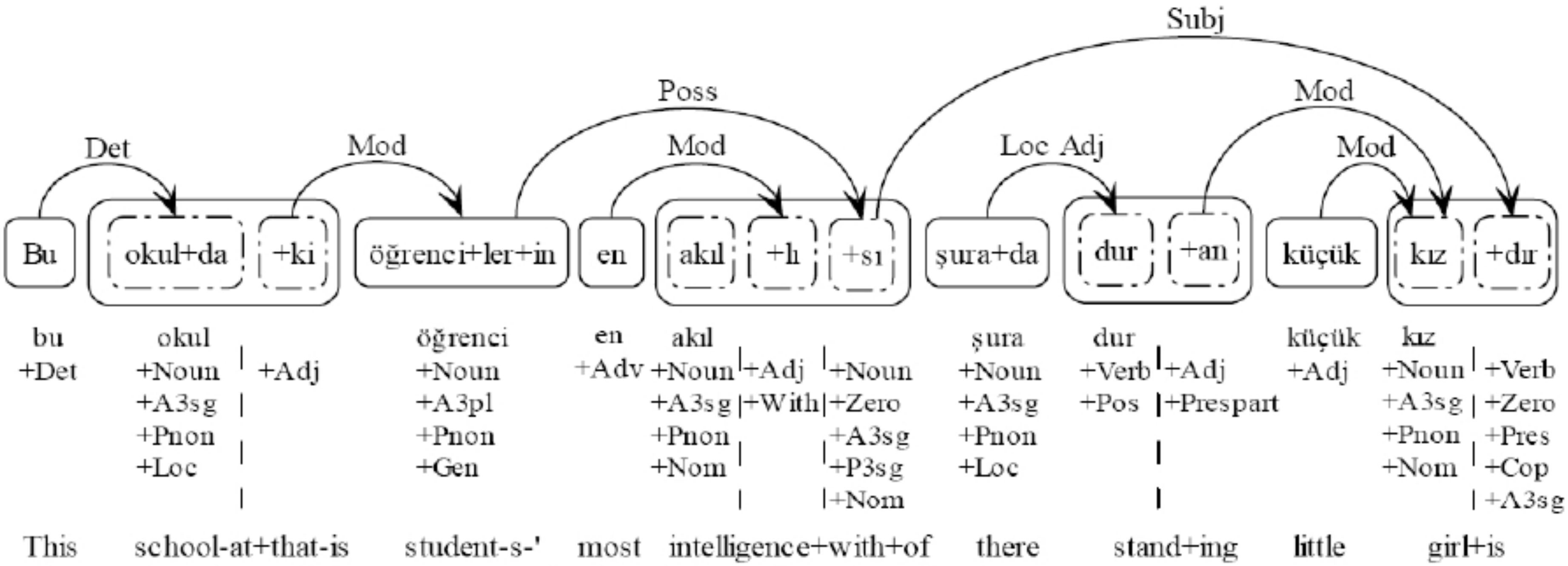
Rich morphological annotations

Dependencies (next slide) are at the morpheme level

- iyileştiriliyorken
  - (literally) while it is being caused to become good
  - while it is being improved
- iyi+Adj ^DB+Verb+Become^DB+Verb+Caus ^DB+Verb+Pass+Pos+Pres^DB+Adverb+While

Very small -- about 5000 sentences

METU-Sabanci Turkish Treebank



The most intelligent of the students in this school is the little girl standing there.

[this and prev. example from Kemal Oflazer's talk at Rochester, April 2007]

Universal Dependencies

37 syntactic relations, intended to be applicable to all languages ("universal"), with slight modifications for each specific language, if necessary.

http://universaldependencies.org

Universal Dependency Relations

Nominal core arguments: `nsubj` (nominal subject), `obj` (direct object), `iobj` (indirect object)
Clausal core arguments: `csubj` (clausal subject), `ccomp` (clausal object [“complement”])
Non-core dependents: `advcl` (adverbial clause modifier), `aux` (auxiliary verb),
Nominal dependents: `nmod` (nominal modifier), `amod` (adjectival modifier),
Coordination: `cc` (coordinating conjunction), `conj` (conjunct)

and many more…

From CFGs to dependencies

From CFGs to dependencies

Assume each CFG rule has **one head child** (bolded)
The other children are **dependents** of the head.

S	→ NP **VP**	**VP** is head, NP is a dependent
VP	→ **V** NP NP	
NP	→ DT **NOUN**	
NOUN	→ ADJ **N**	

The **headword** of a constituent is the terminal that is reached by recursively following the head child.

(here, V is the head word of S, and N is the head word of NP).

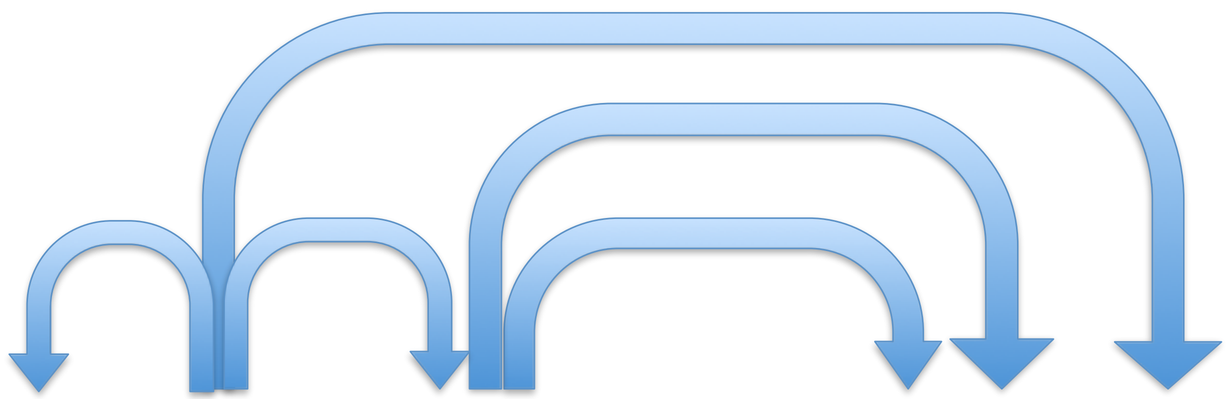
If in rule XP → **X** Y, X is head child and Y dependent, the headword of Y depends on the headword of X.

The **maximal projection** of a terminal *w* is the highest nonterminal in the tree that *w* is headword of.
Here, Y is a maximal projection.

Context-free grammars

CFGs capture only **nested** dependencies

- The dependency graph is a **tree**
- The dependencies **do not cross**

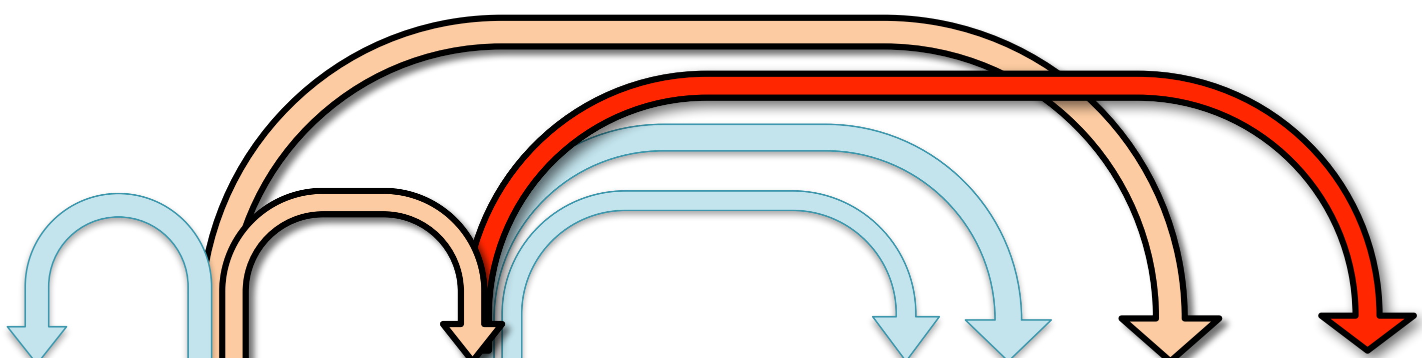


Beyond CFGs: Nonprojective dependencies

Dependencies: tree with crossing branches

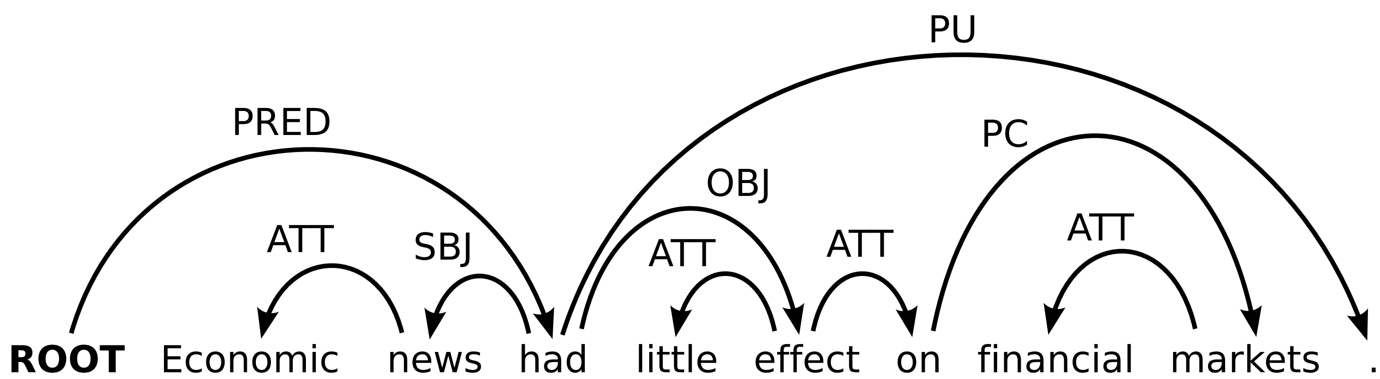
Arise in the following constructions

- (Non-local) **scrambling** (free word order languages)
  Die Pizza *hat Klaus* ***versprochen*** *zu* ***bringen***
- **Extraposition** (*The* ***guy*** *is* ***coming who is wearing a hat***)
- **Topicalization** (***Cheeseburgers****, I* ***thought*** *he* ***likes***)



Dependency Parsing

A dependency parse



Dependencies are (labeled) asymmetrical binary relations between two lexical items (words).

Parsing algorithms for DG

‘Transition-based’ parsers:

learn a sequence of actions to parse sentences

Models:

State = stack of partially processed items
+ queue/buffer of remaining tokens
+ set of dependency arcs that have been found already

Transitions (actions) = add dependency arcs; stack/queue operations

‘Graph-based’ parsers:

learn a model over dependency graphs

Models:

a function (typically sum) of local attachment scores

For dependency trees, you can use a minimum spanning tree algorithm

Transition-based parsing (Nivre et al.)

Transition-based parsing: assumptions

This algorithm works for projective dependency trees.

Dependency tree:

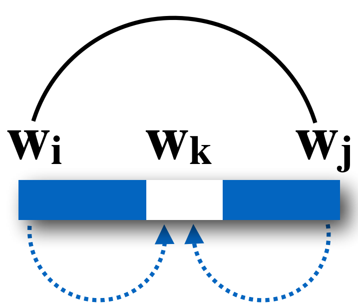
Each word has a single parent
(Each word is a dependent of [is attached to] one other word)

Projective dependencies:

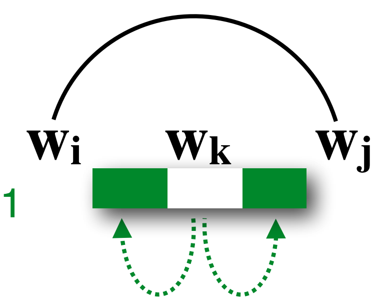
There are no crossing dependencies.

For any i, j, k with $i < k < j$: if there is a dependency between w_i and w_j, the parent of w_k is a word w_l between (possibly including) i and j: $i \leq l \leq j$, while any child w_m of w_k has to occur between (excluding) i and j: $i<m<j$

the parent of w_k:
one of $w_i \ldots w_j$



any child of w_k:
one of $w_{i+1} \ldots w_{j-1}$



Transition-based parsing

Transition-based shift-reduce parsing processes the sentence $S = w_0 w_1 \dots w_n$ from left to right.
Unlike CKY, it constructs a **single tree**.

Notation:

w_0 is a special ROOT token.

$V_S = \{w_0, w_1, \dots, w_n\}$ is the vocabulary of the sentence

R is a set of dependency relations

The parser uses three data structures:

σ: a **stack** of partially processed words $w_i \in V_S$

β: a **buffer** of remaining input words $w_i \in V_S$

A: a **set of dependency arcs** $(w_i, r, w_j) \in V_S \times R \times V_S$

Parser configurations (σ, β, A)

The **stack $\boldsymbol{\sigma}$** is a list of partially processed words

We push and pop words onto/off of $\boldsymbol{\sigma}$.

$\boldsymbol{\sigma|w}$: $\mathbf{w}$ is on top of the stack.

Words on the stack are not (yet) attached to any other words.

Once we attach $\mathbf{w}$, $\mathbf{w}$ can't be put back onto the stack again.

The **buffer $\boldsymbol{\beta}$** is the remaining input words

We read words from $\boldsymbol{\beta}$ (left-to-right) and push them onto σ

$\mathbf{w}|\boldsymbol{\beta}$: $\mathbf{w}$ is on top of the buffer.

The **set of arcs $\mathbf{A}$** defines the current tree.

We can add new arcs to $\mathbf{A}$ by attaching the word on top of the stack to the word on top of the buffer, or vice versa.

Parser configurations (σ, β, A)

We start in the **initial configuration** $([w_0], [w_1,..., w_n], \{\})$

(Root token, Input Sentence, Empty tree)

We can attach the first word (w_1) to the root token w_0,
or we can push w_1 onto the stack.
(w_0 is the only token that can't get attached to any other word)

We want to end in the **terminal configuration** $([], [], A)$

(Empty stack, Empty buffer, Complete tree)

Success!
We have read all of the input words (empty buffer) and have attached all input words to some other word (empty stack)

Transition-based parsing

We process the sentence $S = w_0 w_1 \ldots w_n$ from left to right ("incremental parsing")

In the parser configuration $(\sigma | w_i, w_j | \beta, A)$:

- w_i is on top of the stack. w_i may have some children
- w_j is on top of the buffer. w_j may have some children
- w_i precedes w_j ($i < j$)

We have to either attach w_i to w_j, attach w_j to w_i, or decide that there is no dependency between w_i and w_j

If we reach $(\sigma | w_i, w_j | \beta, A)$, all words w_k with $i < k < j$ have already been attached to a parent w_m with $i \leq m \leq j$

Parser actions

(σ, β, A): Parser configuration with stack σ, buffer β, set of arcs A
(w, r, w'): Dependency with head w, relation r and dependent w'

SHIFT: Push the next input word w_i from the buffer β onto the stack σ

$$(\sigma, w_i|\beta, A) \Rightarrow (\sigma|w_i, \beta, A)$$

LEFT-ARC$_r$: … w_i…w_j… (dependent precedes the head)

Attach dependent w_i (top of stack σ) to head w_j (top of buffer β) with relation r from w_j to w_i. Pop w_i off the stack.

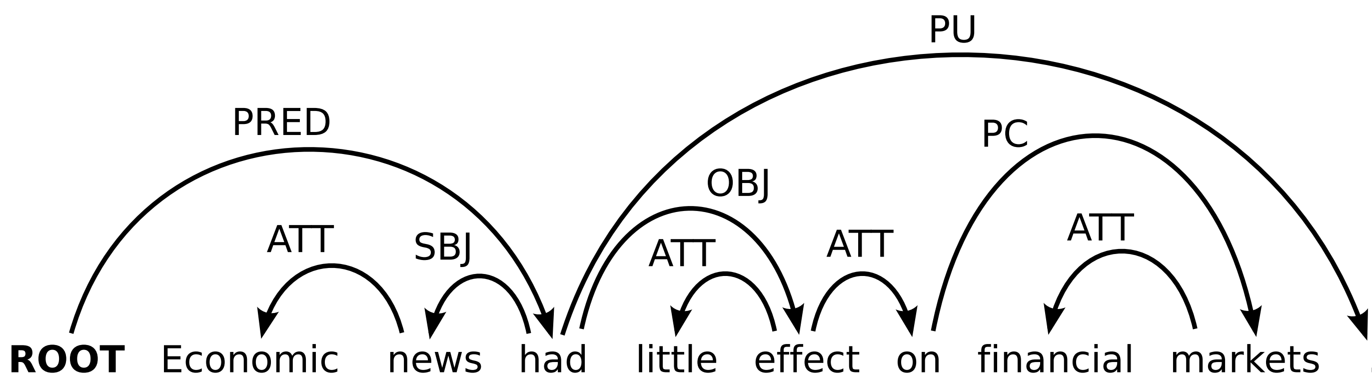
$$(\sigma|w_i, w_j|\beta, A) \Rightarrow (\sigma, w_j|\beta, A \cup \{(w_j, r, w_i)\})$$

RIGHT-ARC$_r$: …w_i…w_j … (dependent follows the head)

Attach dependent w_j (top of buffer β) to head w_i (top of stack σ) with relation r from w_i to w_j. Move w_i back to the buffer

$$(\sigma|w_i, w_j|\beta, A) \Rightarrow (\sigma, w_i|\beta, A \cup \{(w_i, r, w_j)\})$$

An example sentence & parse



Economic news had little effect on financial markets .

Economic news had little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	∅)

Economic news had little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	∅)

Economic **news** had little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	∅)
SH ⇒	([ROOT, Economic],	[news, . . . , .],	∅)

Economic news had little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	∅)
SH ⇒	([ROOT, Economic],	[news, . . . , .],	∅)
LA_{ATT} ⇒	([ROOT],	[news, . . . , .],	A_1 = {(news, ATT, Economic)})

Economic **news** **had** little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
$\text{LA}_{\text{ATT}} \Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)

Economic **news** **had** little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
$\text{LA}_{\text{ATT}} \Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)
$\text{LA}_{\text{SBJ}} \Rightarrow$	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)

Economic news **had** **little** effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, ..., .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, ..., .],	$A_1 = \{(\text{news, ATT, Economic})\})$
SH $\Rightarrow$	([ROOT, news],	[had, ..., .],	$A_1)$
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\})$
SH $\Rightarrow$	([ROOT, had],	[little, ..., .],	$A_2)$

Economic news had **little** **effect** on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, ..., .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, ..., .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, ..., .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, ..., .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, ..., .],	A_2)

Economic news **had** **little** **effect** on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, . . . , .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, . . . , .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)

Economic news had little **effect** **on** financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, . . . , .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, . . . , .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, . . . , .],	A_3)

Economic news had little effect **on** **financial** markets .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, ..., .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, ..., .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, ..., .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, ..., .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, ..., .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, ..., .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, ..., .],	A_3)
SH $\Rightarrow$	([ROOT, ... on],	[financial, markets, .],	A_3)

Economic news had little effect on **financial** **markets** .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, ..., .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, ..., .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, ..., .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, ..., .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, ..., .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, ..., .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, ..., .],	A_3)
SH $\Rightarrow$	([ROOT, ... on],	[financial, markets, .],	A_3)
SH $\Rightarrow$	([ROOT, ..., financial],	[markets, .],	A_3)

Economic news had little effect **on financial markets** .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, ..., .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, ..., .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, ..., .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, ..., .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, ..., .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, ..., .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, ..., .],	A_3)
SH $\Rightarrow$	([ROOT, ... on],	[financial, markets, .],	A_3)
SH $\Rightarrow$	([ROOT, ..., financial],	[markets, .],	A_3)
LA_{ATT} $\Rightarrow$	([ROOT, ... on],	[markets, .],	$A_4 = A_3 \cup \{(\text{markets, ATT, financial})\}$)

Economic news had little **effect** **on** financial **markets** .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	∅)
SH ⇒	([ROOT, Economic],	[news, . . . , .],	∅)
LA_{ATT} ⇒	([ROOT],	[news, . . . , .],	A_1 = {(news, ATT, Economic)})
SH ⇒	([ROOT, news],	[had, . . . , .],	A_1)
LA_{SBJ} ⇒	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup$ {(had, SBJ, news)})
SH ⇒	([ROOT, had],	[little, . . . , .],	A_2)
SH ⇒	([ROOT, had, little],	[effect, . . . , .],	A_2)
LA_{ATT} ⇒	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup$ {(effect, ATT, little)})
SH ⇒	([ROOT, had, effect],	[on, . . . , .],	A_3)
SH ⇒	([ROOT, . . . on],	[financial, markets, .],	A_3)
SH ⇒	([ROOT, . . . , financial],	[markets, .],	A_3)
LA_{ATT} ⇒	([ROOT, . . . on],	[markets, .],	$A_4 = A_3 \cup$ {(markets, ATT, financial)})
RA_{PC} ⇒	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup$ {(on, PC, markets)})

Economic news **had** little **effect** **on** financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, . . . , .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, . . . , .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, . . . , .],	A_3)
SH $\Rightarrow$	([ROOT, . . . on],	[financial, markets, .],	A_3)
SH $\Rightarrow$	([ROOT, . . . , financial],	[markets, .],	A_3)
LA_{ATT} $\Rightarrow$	([ROOT, . . . on],	[markets, .],	$A_4 = A_3 \cup \{(\text{markets, ATT, financial})\}$)
RA_{PC} $\Rightarrow$	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup \{(\text{on, PC, markets})\}$)
RA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, .],	$A_6 = A_5 \cup \{(\text{effect, ATT, on})\}$)

Economic news **had** little **effect** on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, ..., .],	$\emptyset$)
$\text{LA}_{\text{ATT}} \Rightarrow$	([ROOT],	[news, ..., .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, ..., .],	A_1)
$\text{LA}_{\text{SBJ}} \Rightarrow$	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, ..., .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, ..., .],	A_2)
$\text{LA}_{\text{ATT}} \Rightarrow$	([ROOT, had],	[effect, ..., .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, ..., .],	A_3)
SH $\Rightarrow$	([ROOT, ... on],	[financial, markets, .],	A_3)
SH $\Rightarrow$	([ROOT, ..., financial],	[markets, .],	A_3)
$\text{LA}_{\text{ATT}} \Rightarrow$	([ROOT, ... on],	[markets, .],	$A_4 = A_3 \cup \{(\text{markets, ATT, financial})\}$)
$\text{RA}_{\text{PC}} \Rightarrow$	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup \{(\text{on, PC, markets})\}$)
$\text{RA}_{\text{ATT}} \Rightarrow$	([ROOT, had],	[effect, .],	$A_6 = A_5 \cup \{(\text{effect, ATT, on})\}$)
$\text{RA}_{\text{OBJ}} \Rightarrow$	([ROOT],	[had, .],	$A_7 = A_6 \cup \{(\text{had, OBJ, effect})\}$)

Economic news **had** little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, ..., .],	∅)
SH ⇒	([ROOT, Economic],	[news, ..., .],	∅)
LA_{ATT} ⇒	([ROOT],	[news, ..., .],	A_1 = {(news, ATT, Economic)})
SH ⇒	([ROOT, news],	[had, ..., .],	A_1)
LA_{SBJ} ⇒	([ROOT],	[had, ..., .],	$A_2 = A_1 \cup$ {(had, SBJ, news)})
SH ⇒	([ROOT, had],	[little, ..., .],	A_2)
SH ⇒	([ROOT, had, little],	[effect, ..., .],	A_2)
LA_{ATT} ⇒	([ROOT, had],	[effect, ..., .],	$A_3 = A_2 \cup$ {(effect, ATT, little)})
SH ⇒	([ROOT, had, effect],	[on, ..., .],	A_3)
SH ⇒	([ROOT, ... on],	[financial, markets, .],	A_3)
SH ⇒	([ROOT, ..., financial],	[markets, .],	A_3)
LA_{ATT} ⇒	([ROOT, ... on],	[markets, .],	$A_4 = A_3 \cup$ {(markets, ATT, financial)})
RA_{PC} ⇒	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup$ {(on, PC, markets)})
RA_{ATT} ⇒	([ROOT, had],	[effect, .],	$A_6 = A_5 \cup$ {(effect, ATT, on)})
RA_{OBJ} ⇒	([ROOT],	[had, .],	$A_7 = A_6 \cup$ {(had, OBJ, effect)})
SH ⇒	([ROOT, had],	[.],	A_7)

Economic news **had** little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
$LA_{ATT} \Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)
$LA_{SBJ} \Rightarrow$	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, . . . , .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, . . . , .],	A_2)
$LA_{ATT} \Rightarrow$	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, . . . , .],	A_3)
SH $\Rightarrow$	([ROOT, . . . on],	[financial, markets, .],	A_3)
SH $\Rightarrow$	([ROOT, . . . , financial],	[markets, .],	A_3)
$LA_{ATT} \Rightarrow$	([ROOT, . . . on],	[markets, .],	$A_4 = A_3 \cup \{(\text{markets, ATT, financial})\}$)
$RA_{PC} \Rightarrow$	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup \{(\text{on, PC, markets})\}$)
$RA_{ATT} \Rightarrow$	([ROOT, had],	[effect, .],	$A_6 = A_5 \cup \{(\text{effect, ATT, on})\}$)
$RA_{OBJ} \Rightarrow$	([ROOT],	[had, .],	$A_7 = A_6 \cup \{(\text{had, OBJ, effect})\}$)
SH $\Rightarrow$	([ROOT, had],	[.],	A_7)
$RA_{PU} \Rightarrow$	([ROOT],	[had],	$A_8 = A_7 \cup \{(\text{had, PU, .})\}$)

Economic news **had** little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	∅)
SH ⇒	([ROOT, Economic],	[news, . . . , .],	∅)
LA_{ATT} ⇒	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH ⇒	([ROOT, news],	[had, . . . , .],	A_1)
LA_{SBJ} ⇒	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH ⇒	([ROOT, had],	[little, . . . , .],	A_2)
SH ⇒	([ROOT, had, little],	[effect, . . . , .],	A_2)
LA_{ATT} ⇒	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH ⇒	([ROOT, had, effect],	[on, . . . , .],	A_3)
SH ⇒	([ROOT, . . . on],	[financial, markets, .],	A_3)
SH ⇒	([ROOT, . . . , financial],	[markets, .],	A_3)
LA_{ATT} ⇒	([ROOT, . . . on],	[markets, .],	$A_4 = A_3 \cup \{(\text{markets, ATT, financial})\}$)
RA_{PC} ⇒	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup \{(\text{on, PC, markets})\}$)
RA_{ATT} ⇒	([ROOT, had],	[effect, .],	$A_6 = A_5 \cup \{(\text{effect, ATT, on})\}$)
RA_{OBJ} ⇒	([ROOT],	[had, .],	$A_7 = A_6 \cup \{(\text{had, OBJ, effect})\}$)
SH ⇒	([ROOT, had],	[.],	A_7)
RA_{PU} ⇒	([ROOT],	[had],	$A_8 = A_7 \cup \{(\text{had, PU, .})\}$)
RA_{PRED} ⇒	([ ],	[ROOT],	$A_9 = A_8 \cup \{(\text{ROOT, PRED, had})\}$)

Economic news had little effect on financial markets .

Transition	Configuration		
	([ROOT],	[Economic, . . . , .],	$\emptyset$)
SH $\Rightarrow$	([ROOT, Economic],	[news, . . . , .],	$\emptyset$)
LA_{ATT} $\Rightarrow$	([ROOT],	[news, . . . , .],	$A_1 = \{(\text{news, ATT, Economic})\}$)
SH $\Rightarrow$	([ROOT, news],	[had, . . . , .],	A_1)
LA_{SBJ} $\Rightarrow$	([ROOT],	[had, . . . , .],	$A_2 = A_1 \cup \{(\text{had, SBJ, news})\}$)
SH $\Rightarrow$	([ROOT, had],	[little, . . . , .],	A_2)
SH $\Rightarrow$	([ROOT, had, little],	[effect, . . . , .],	A_2)
LA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, . . . , .],	$A_3 = A_2 \cup \{(\text{effect, ATT, little})\}$)
SH $\Rightarrow$	([ROOT, had, effect],	[on, . . . , .],	A_3)
SH $\Rightarrow$	([ROOT, . . . on],	[financial, markets, .],	A_3)
SH $\Rightarrow$	([ROOT, . . . , financial],	[markets, .],	A_3)
LA_{ATT} $\Rightarrow$	([ROOT, . . . on],	[markets, .],	$A_4 = A_3 \cup \{(\text{markets, ATT, financial})\}$)
RA_{PC} $\Rightarrow$	([ROOT, had, effect],	[on, .],	$A_5 = A_4 \cup \{(\text{on, PC, markets})\}$)
RA_{ATT} $\Rightarrow$	([ROOT, had],	[effect, .],	$A_6 = A_5 \cup \{(\text{effect, ATT, on})\}$)
RA_{OBJ} $\Rightarrow$	([ROOT],	[had, .],	$A_7 = A_6 \cup \{(\text{had, OBJ, effect})\}$)
SH $\Rightarrow$	([ROOT, had],	[.],	A_7)
RA_{PU} $\Rightarrow$	([ROOT],	[had],	$A_8 = A_7 \cup \{(\text{had, PU, .})\}$)
RA_{PRED} $\Rightarrow$	([ ],	[ROOT],	$A_9 = A_8 \cup \{(\text{ROOT, PRED, had})\}$)
SH $\Rightarrow$	([ROOT],	[ ],	A_9)

Transition-based parsing in practice

Which action should the parser take under the current configuration?

We also need a parsing model that assigns a score to each possible action given a current configuration.

- Possible actions:
  SHIFT, and for any relation r: LEFT-ARC$_r$, or RIGHT-ARC$_r$
- Possible features of the current configuration:
  The top {1,2,3} words on the buffer and on the stack, their POS tags, distances between the words, etc.

We can learn this model from a dependency treebank.