Universal Conceptual Cognitive Annotation

Cross-lingual Semantic Representation for NLP with UCCA

Daniel Hershcovich
University of Copenhagen

Parsing, Evaluation and Applications

Outline

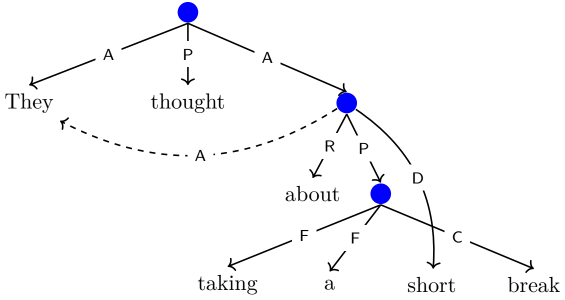
1. Parsing
2. Evaluation
3. Applications
   - Incorporating linguistically informed rules into NLP
   - Controlled NLG evaluation by explicit criteria

UCCA Parsing

The Task: Given plain text, predict its UCCA graph representation.

They thought about taking a short break

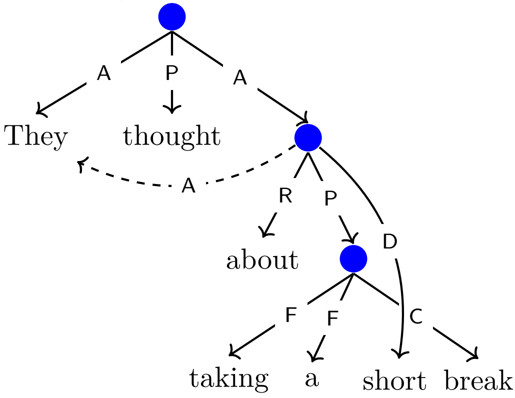
$\Downarrow$



Graph Structure

Labeled directed acyclic graphs (DAGs). Complex units are non-terminal nodes. Phrases may be discontinuous. *Remote edges* enable reentrancy.

A	Participant
C	Center
D	Adverbial
E	Elaborator
F	Function
G	Ground
H	Parallel scene
L	Linker
P	Process
R	Relator
S	State
U	Punctuation

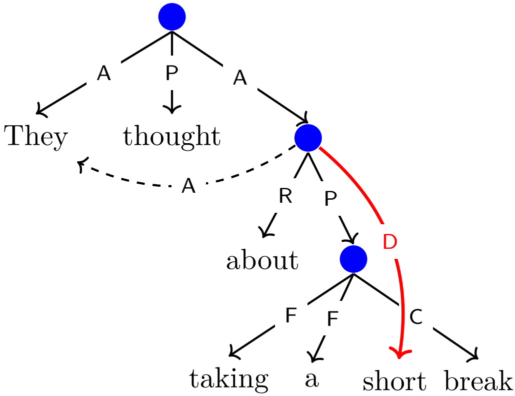


Graph Structure

Labeled directed acyclic graphs (DAGs). Complex units are non-terminal nodes. Phrases may be discontinuous.

Remote edges enable reentrancy.

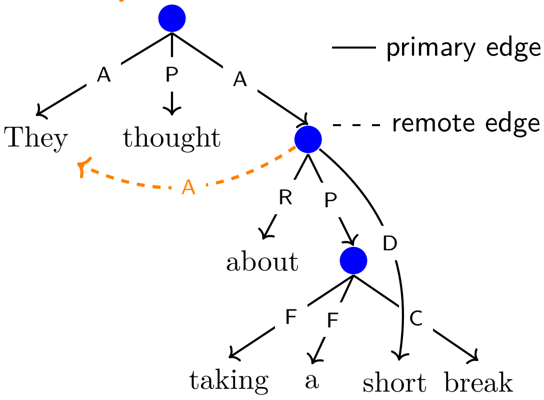
A	Participant
C	Center
D	Adverbial
E	Elaborator
F	Function
G	Ground
H	Parallel scene
L	Linker
P	Process
R	Relator
S	State
U	Punctuation



Graph Structure

Labeled directed acyclic graphs (DAGs). Complex units are non-terminal nodes. Phrases may be discontinuous. *Remote edges* enable reentrancy.

A	Participant
C	Center
D	Adverbial
E	Elaborator
F	Function
G	Ground
H	Parallel scene
L	Linker
P	Process
R	Relator
S	State
U	Punctuation

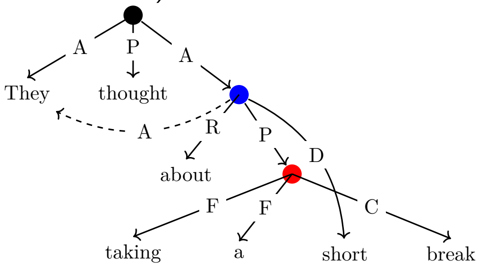


Data Statistics

	Wiki	20K			EWT
	en	en	fr	de	en
# sentences	5,141	492	492	6,514	3,813
# tokens	158K	12K	12K	144K	55K
# non-terminal nodes	62,002	4,699	5,110	51,934	18,156
% discontinuous	1.71	3.19	4.64	8.87	3.87
% reentrant	1.84	0.89	0.65	0.31	0.83
# edges	208,937	16,803	17,520	187,533	60,739
% primary	97.40	96.79	97.02	97.32	97.32
% remote	2.60	3.21	2.98	2.68	2.68

Transition-based UCCA Parser

A Transition-Based Directed Acyclic Graph Parser for UCCA (Hershcovich et al., 2017).



$\Updownarrow$

Shift, Right-Edge$_A$, Shift, Swap, Right-Edge$_P$, Reduce, Shift, Shift, Node$_R$, Reduce, Left-Remote$_A$, Shift, Shift, Node$_C$, Reduce, Shift, Right-Edge$_P$, Shift, Right-Edge$_F$, Reduce, Shift, Swap, Right-Edge$_D$, Reduce, Swap, Right-Edge$_A$, Reduce, Reduce, Shift, Reduce, Shift, Right-Edge$_C$, Finish

Transition-based UCCA Parser

Parses text $w_1 \ldots w_n$ to graph G incrementally by applying transitions to the parser state, consisting of: stack, buffer and constructed graph.

Initial state:

stack

buffer

They	thought	about	taking	a	short	break

Transitions:

$\{$Shift, Reduce, Node$_X$, Left-Edge$_X$, Right-Edge$_X$, Left-Remote$_X$, Right-Remote$_X$, Swap, Finish$\}$

Transition-based UCCA Parser

Parses text $w_1 \dots w_n$ to graph G incrementally by applying transitions to the parser state, consisting of: stack, buffer and constructed graph.

Initial state:

stack							buffer
●	They	thought	about	taking	a	short	break

Transitions:

{SHIFT, REDUCE, NODE$_X$, LEFT-EDGE$_X$, RIGHT-EDGE$_X$, LEFT-REMOTE$_X$, RIGHT-REMOTE$_X$, SWAP, FINISH}

Transition-based UCCA Parser

Parses text $w_1 \ldots w_n$ to graph G incrementally by applying transitions to the parser state, consisting of: stack, buffer and constructed graph.

Initial state:

stack

●

buffer

They	thought	about	taking	a	short	break

Transitions:

{Shift, Reduce, Node$_X$, Left-Edge$_X$, Right-Edge$_X$, Left-Remote$_X$, Right-Remote$_X$, Swap, Finish}

Example: TUPA Transition Sequence

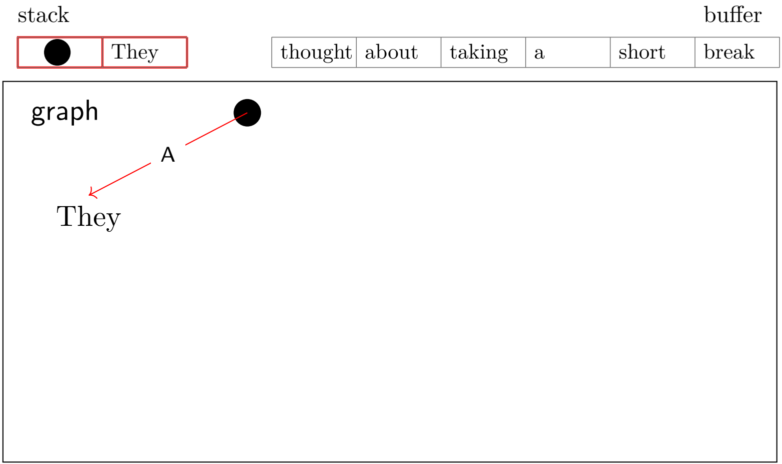
$\Rightarrow$ Shift

stack								buffer
●	They		thought	about	taking	a	short	break

graph ●

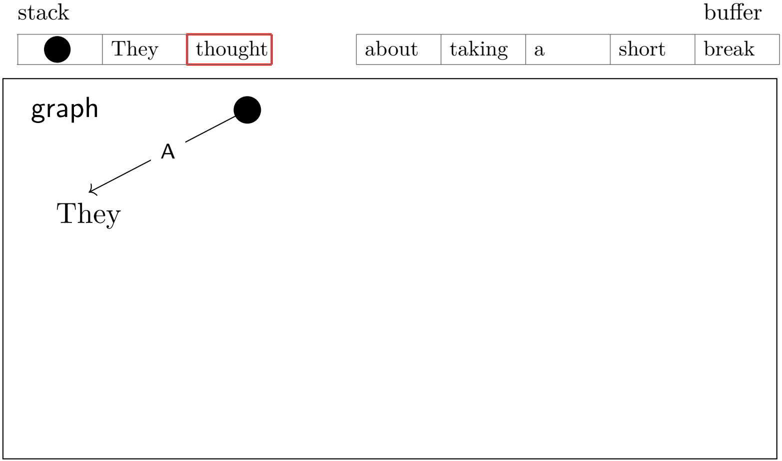
Example: TUPA Transition Sequence

$\Rightarrow$ RIGHT-EDGE$_A$



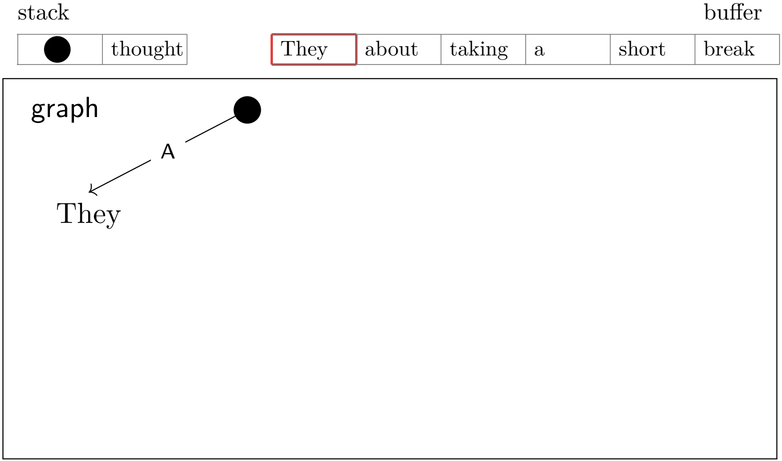
Example: TUPA Transition Sequence

$\Rightarrow$ SHIFT



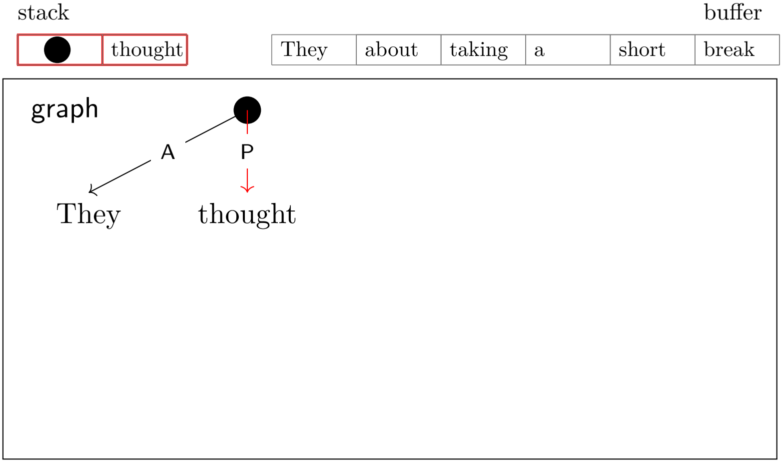
Example: TUPA Transition Sequence

$\Rightarrow$ SWAP



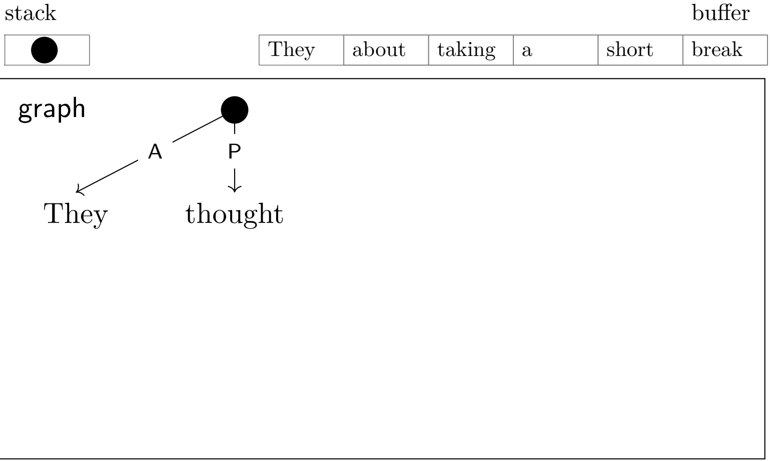
Example: TUPA Transition Sequence

$\Rightarrow$ RIGHT-EDGE$_P$



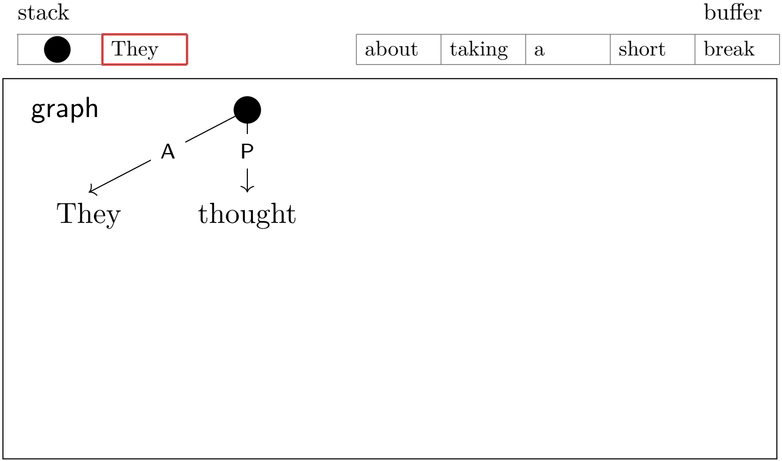
Example: TUPA Transition Sequence

$\Rightarrow$ REDUCE



Example: TUPA Transition Sequence

⇒ Shift



Example: TUPA Transition Sequence

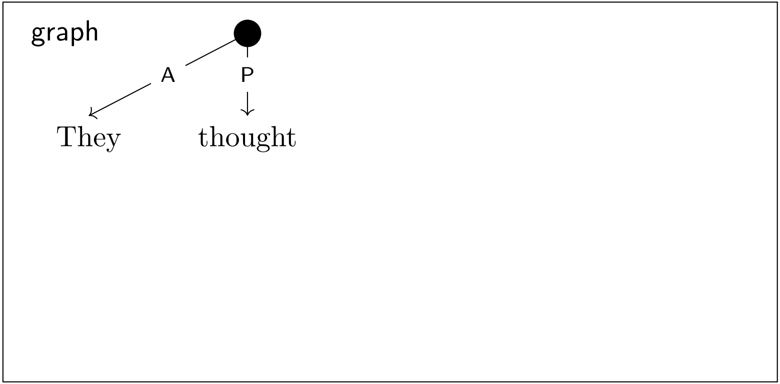
⇒ Shift

stack

	They	about

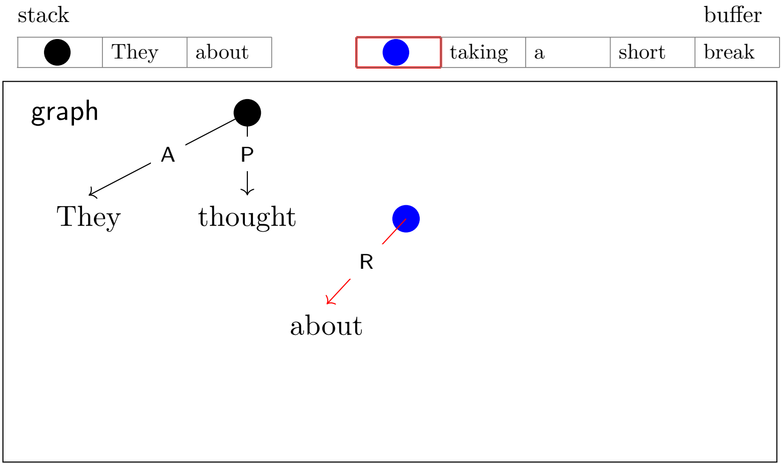
buffer

taking	a	short	break

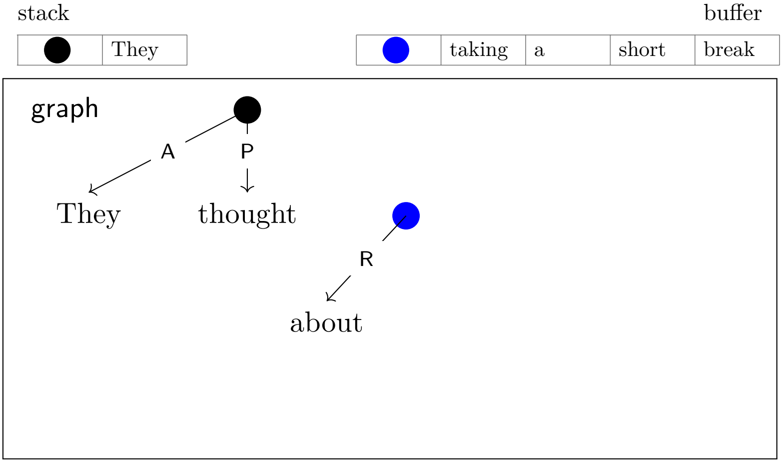


Example: TUPA Transition Sequence

⇒ NODE_R

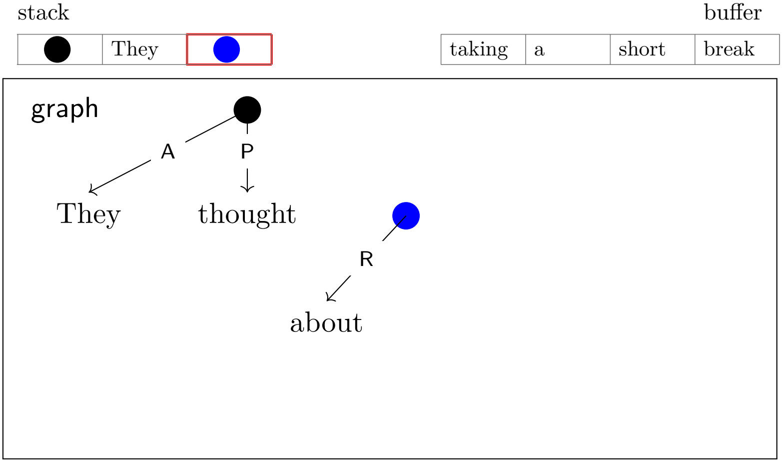


⇒ REDUCE


stack
They
buffer
taking
a
short
break
graph
A
P
They
thought
R
about

Example: TUPA Transition Sequence

⇒ SHIFT



Example: TUPA Transition Sequence

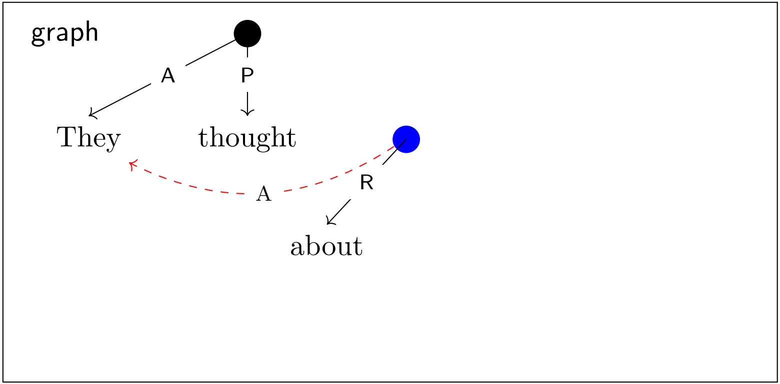
⇒ LEFT-REMOTE$_A$

stack

	They	

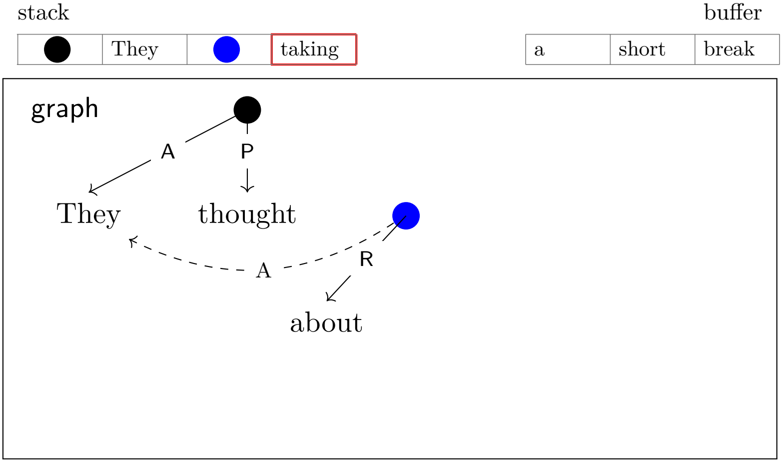
buffer

taking	a	short	break



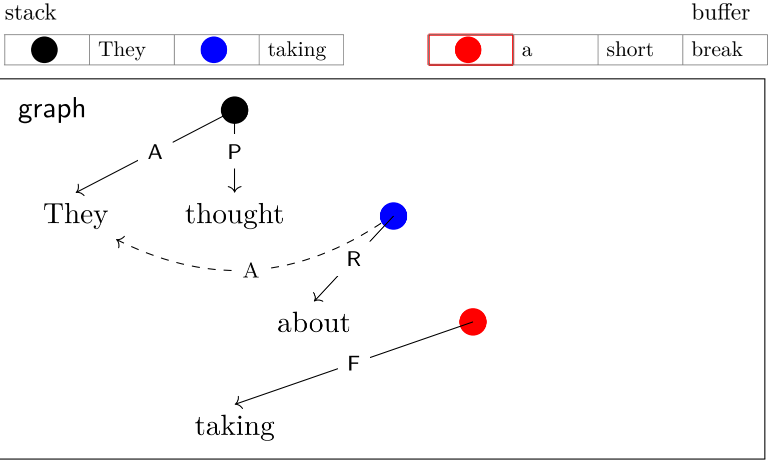
Example: TUPA Transition Sequence

⇒ SHIFT



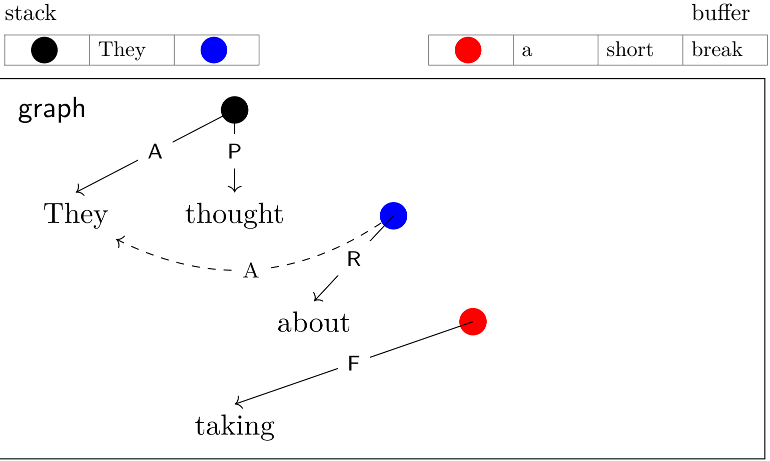
Example: TUPA Transition Sequence

⇒ $\textsc{Node}_C$



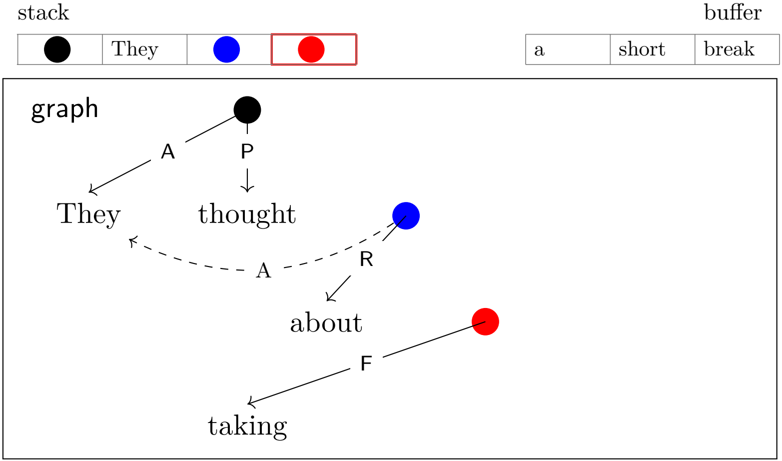
Example: TUPA Transition Sequence

⇒ REDUCE



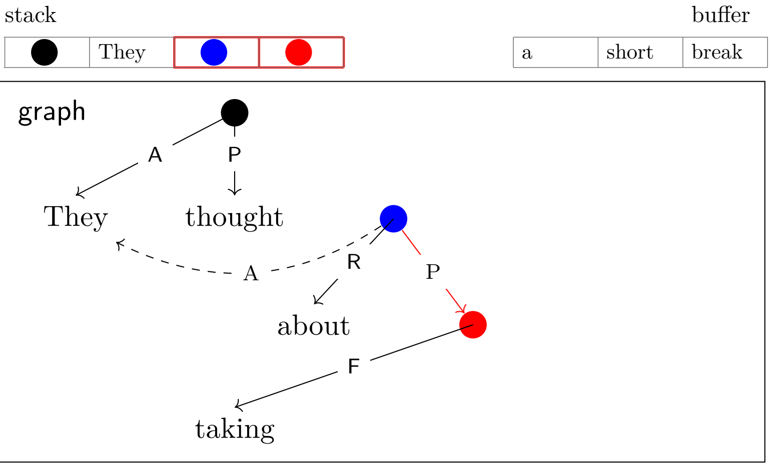
Example: TUPA Transition Sequence

⇒ SHIFT



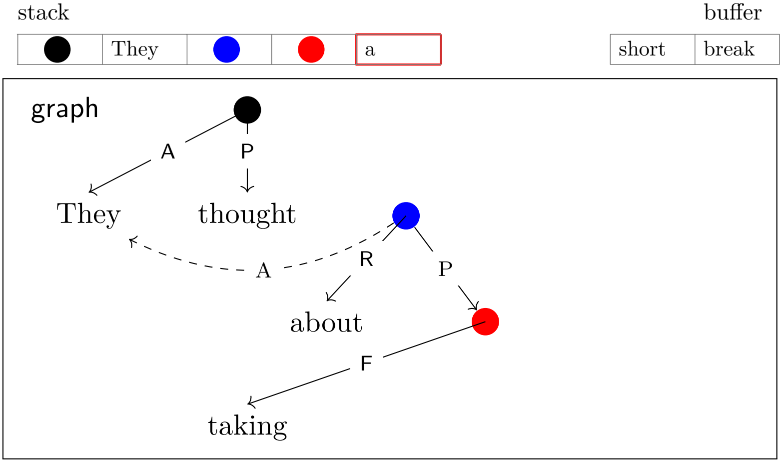
Example: TUPA Transition Sequence

⇒ RIGHT-EDGE$_P$



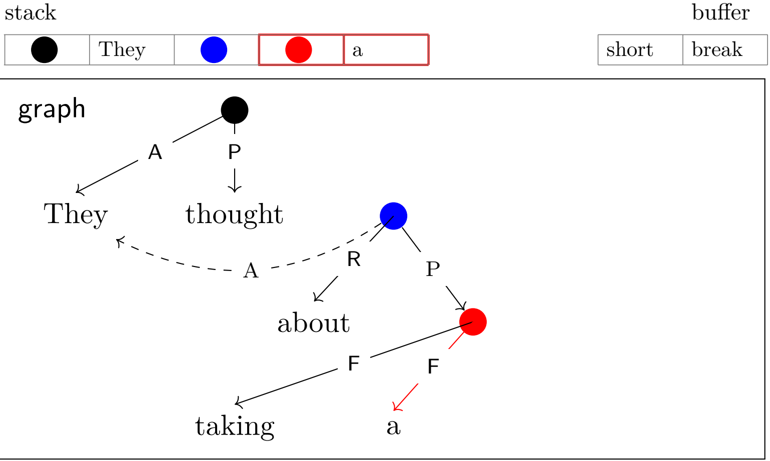
Example: TUPA Transition Sequence

⇒ Shift



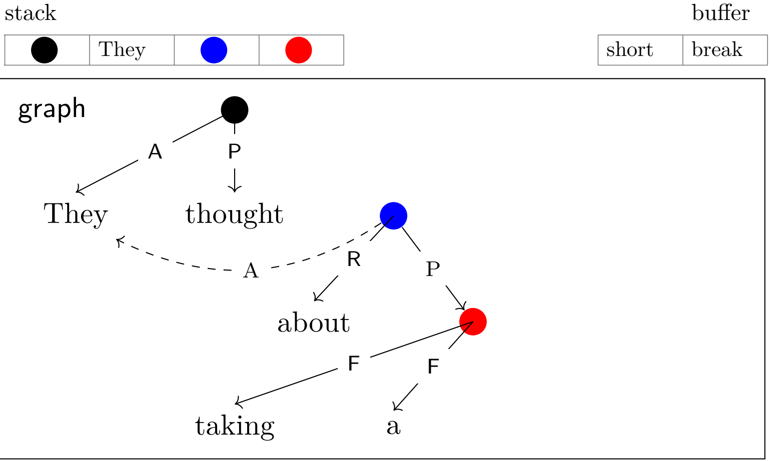
Example: TUPA Transition Sequence

$\Rightarrow$ RIGHT-EDGE$_F$



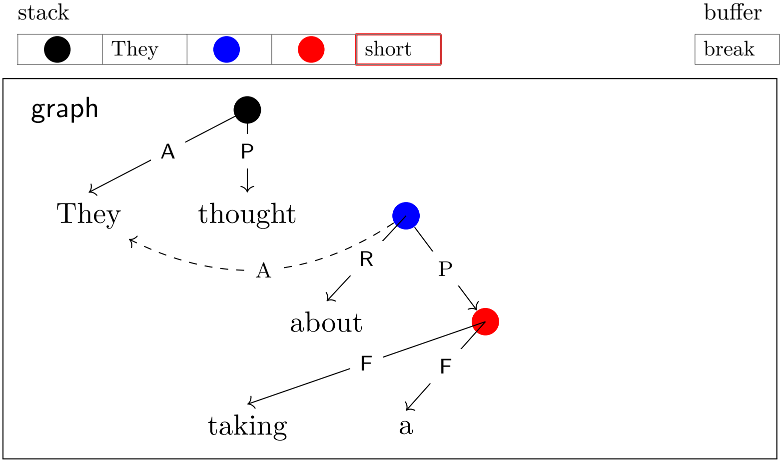
Example: TUPA Transition Sequence

⇒ REDUCE



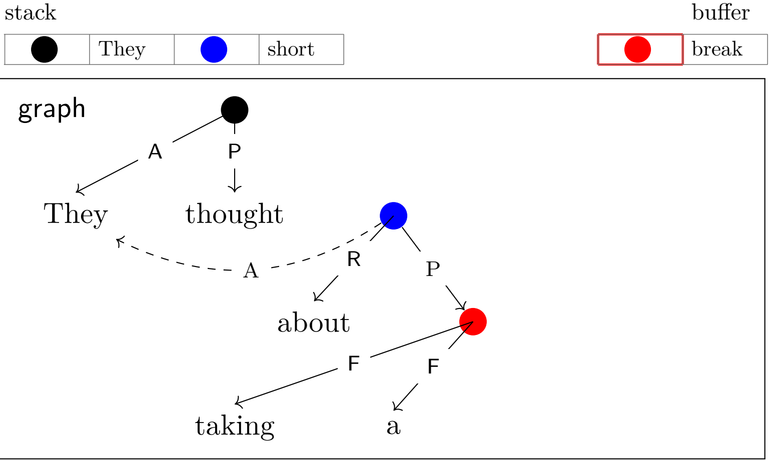
Example: TUPA Transition Sequence

⇒ SHIFT



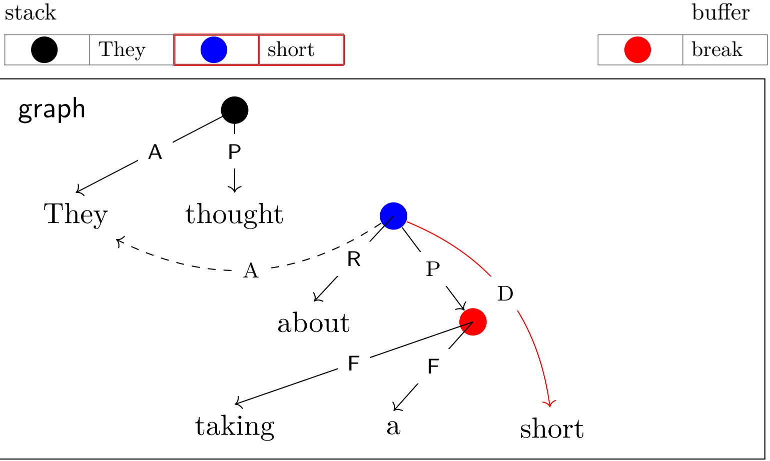
Example: TUPA Transition Sequence

⇒ SWAP



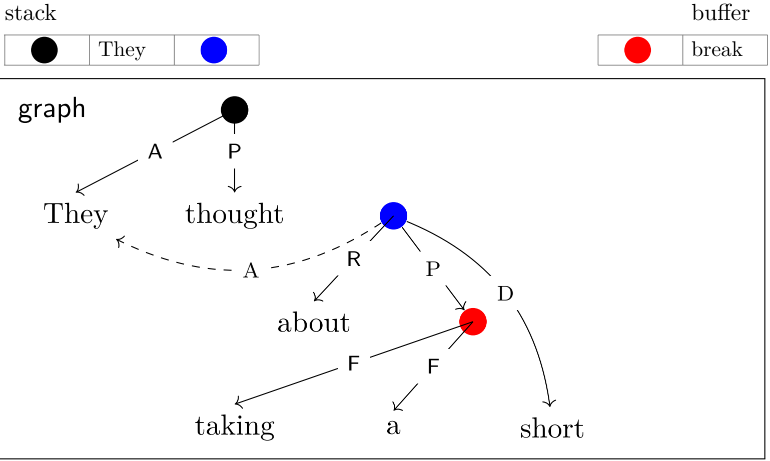
Example: TUPA Transition Sequence

$\Rightarrow$ Right-Edge$_D$



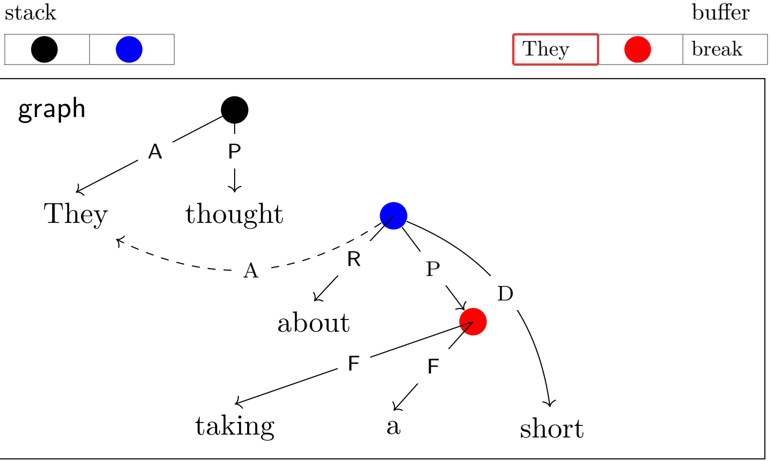
Example: TUPA Transition Sequence

⇒ Reduce



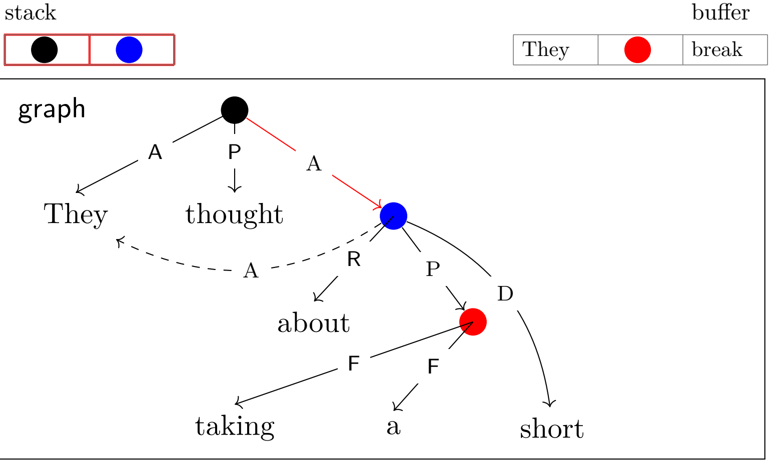
Example: TUPA Transition Sequence

$\Rightarrow$ Swap



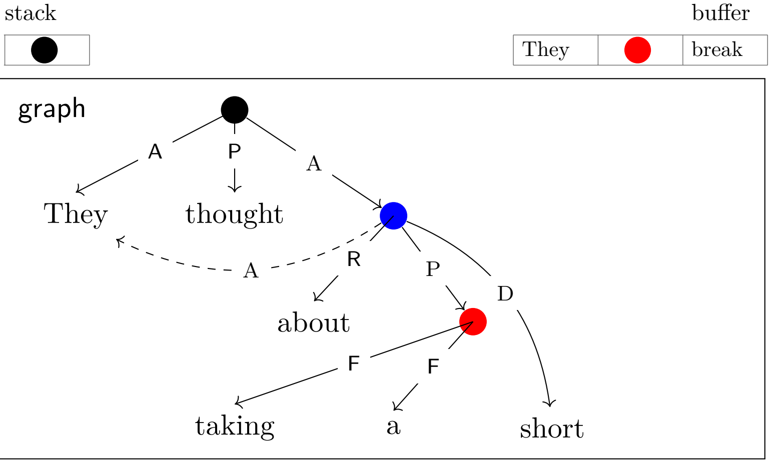
Example: TUPA Transition Sequence

$\Rightarrow$ RIGHT-EDGE$_A$



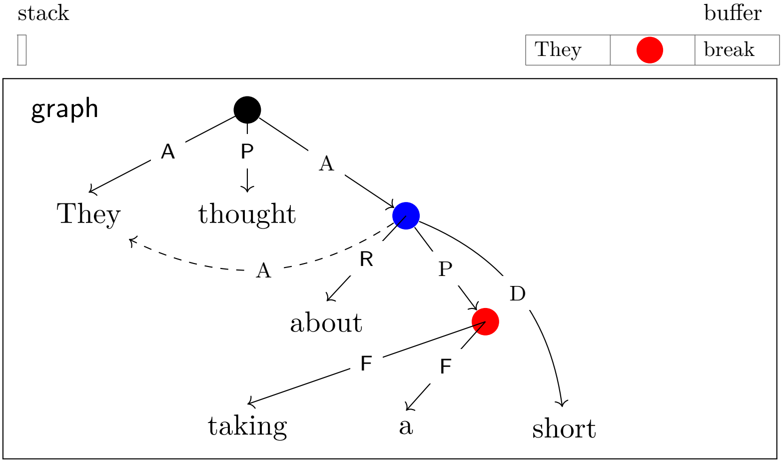
Example: TUPA Transition Sequence

⇒ Reduce



Example: TUPA Transition Sequence

⇒ Reduce



Example: TUPA Transition Sequence

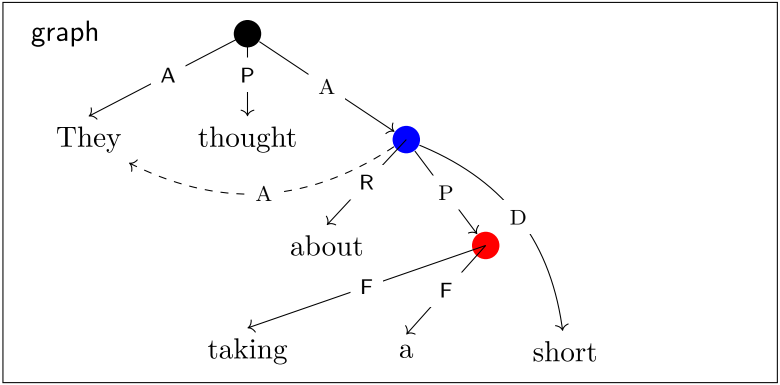
⇒ SHIFT

stack

They

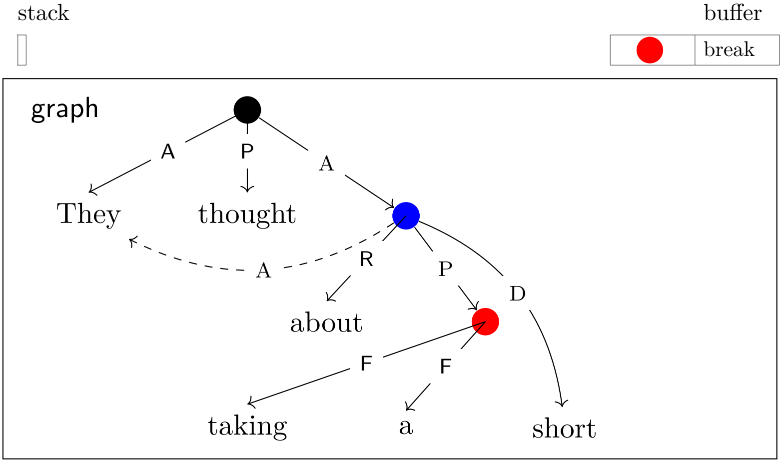
buffer

●	break



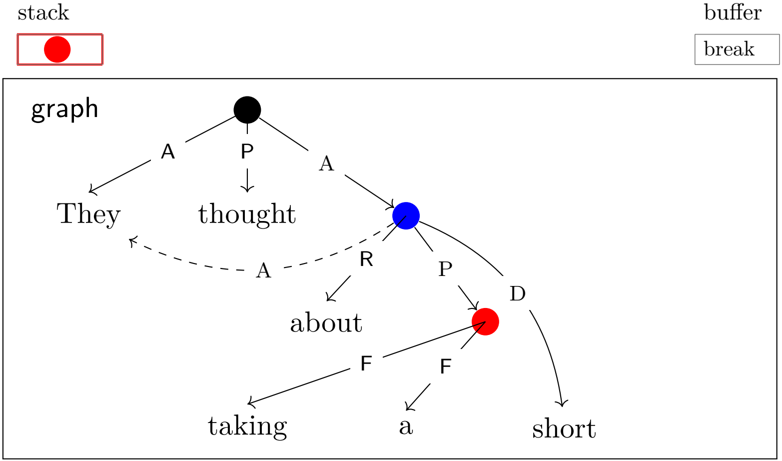
Example: TUPA Transition Sequence

⇒ Reduce



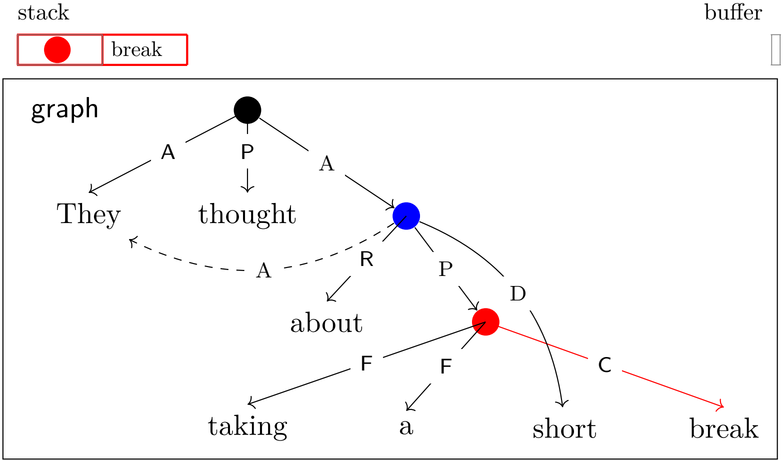
Example: TUPA Transition Sequence

⇒ Shift



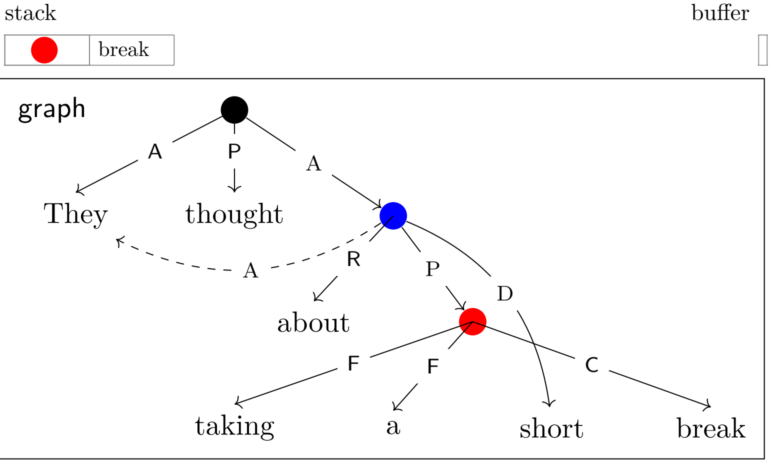
Example: TUPA Transition Sequence

$\Rightarrow$ RIGHT-EDGE$_C$



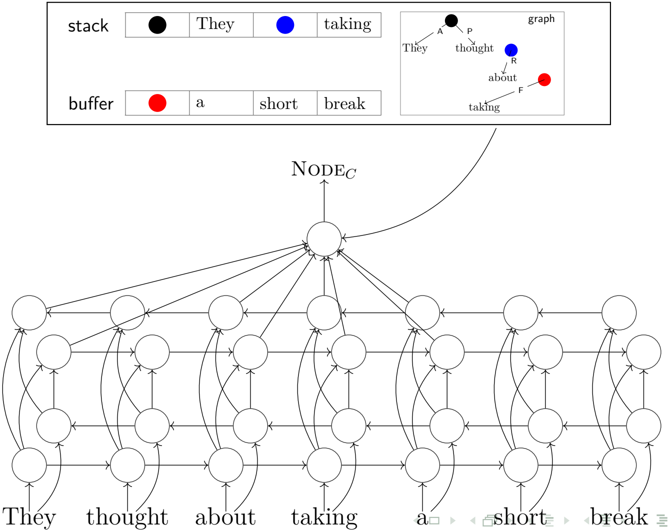
Example: TUPA Transition Sequence

⇒ FINISH



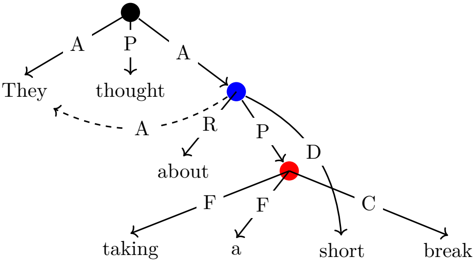
TUPA model

Learns to predict next transition based on current state.



Training

An *oracle* provides the transition sequence given the correct graph:



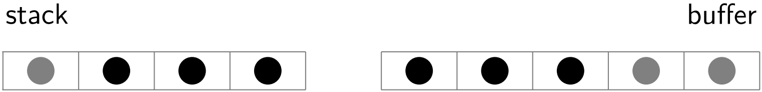
$\Updownarrow$

SHIFT, RIGHT-EDGE$_A$, SHIFT, SWAP, RIGHT-EDGE$_P$, REDUCE, SHIFT, SHIFT, NODE$_R$, REDUCE, LEFT-REMOTE$_A$, SHIFT, SHIFT, NODE$_C$, REDUCE, SHIFT, RIGHT-EDGE$_P$, SHIFT, RIGHT-EDGE$_F$, REDUCE, SHIFT, SWAP, RIGHT-EDGE$_D$, REDUCE, SWAP, RIGHT-EDGE$_A$, REDUCE, REDUCE, SHIFT, REDUCE, SHIFT, RIGHT-EDGE$_C$, FINISH

TUPA Model

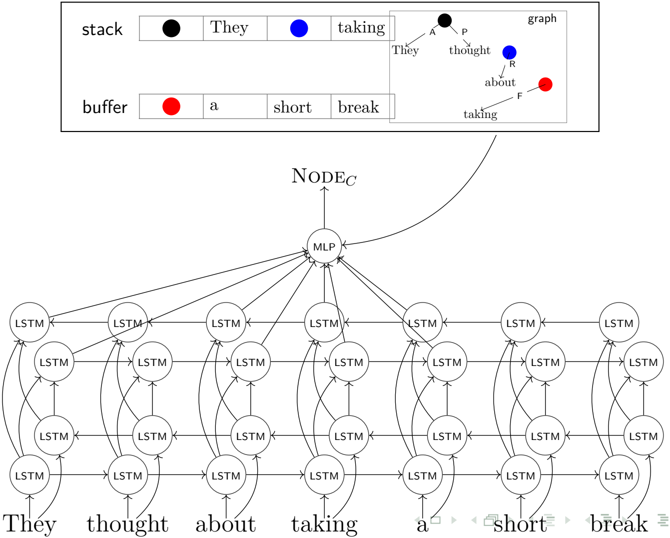
Learns to greedily predict transition based on current state.

Features include:
{words, parts of speech, syntactic dependencies, edge labels}
from the stack and buffer + parents, children, grandchildren.



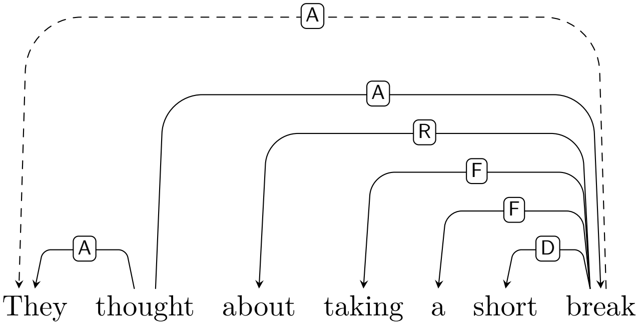
TUPA Model

Learns to greedily predict transition based on current state.



Comparing to Dependency Parsers

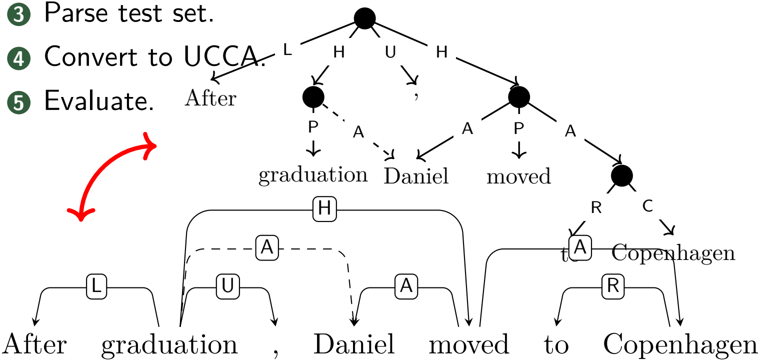
Using conversion-based approximation as baseline,
with bi-lexical DAG parsers and transition-based tree parsers.



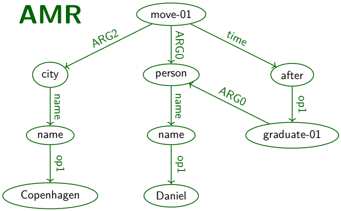
UCCA bi-lexical DAG approximation.

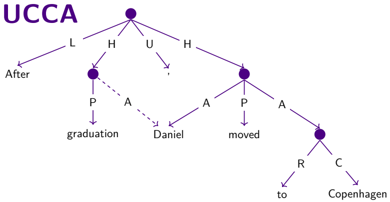
Bi-lexical Graph Approximation

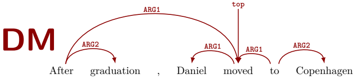
1. Convert UCCA to bi-lexical DAGs.
2. Train bi-lexical parsers.
3. Parse test set.
4. Convert to UCCA.
5. Evaluate.



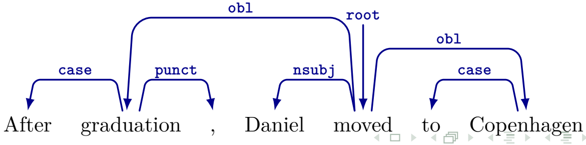
Other Semantic/Syntactic Representations







UD (Universal Dependencies)



Data

UCCA training data is scarce

(English)

and domains are limited.

UCCA	AMR	DM	UD
Wikipedia	blogs	news	blogs
books	news		news
reviews	emails		emails
	reviews		reviews
			Q&A

Data

UCCA training data is scarce

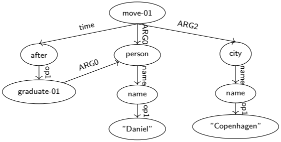
(English)

and domains are limited.

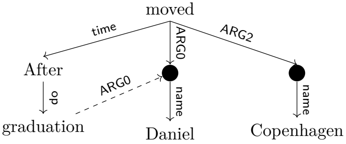
UCCA	AMR	DM	UD
Wikipedia	blogs	news	blogs
books	news		news
reviews	emails		emails
	reviews		reviews
			Q&A

Conversion

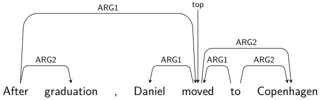
AMR



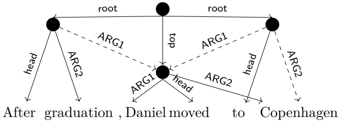
⇔



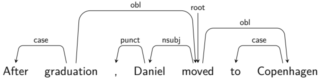
DM



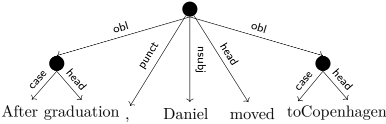
⇔



UD

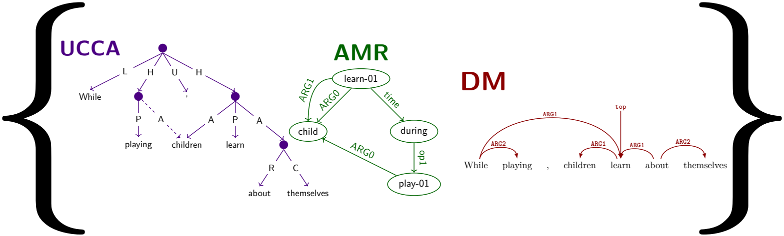


⇔



Sharing for Better Generalization

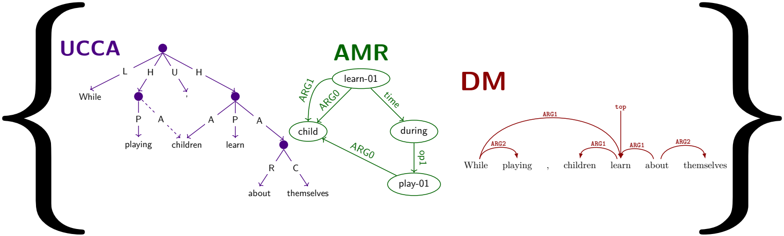
Multitask Parsing Across Semantic Representations
(Hershcovich et al., 2018)



Improved UCCA parsing in English, French and German.

Sharing for Better Generalization

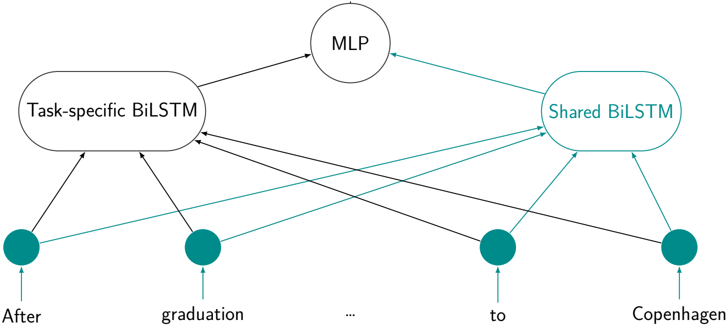
Multitask Parsing Across Semantic Representations
(Hershcovich et al., 2018)



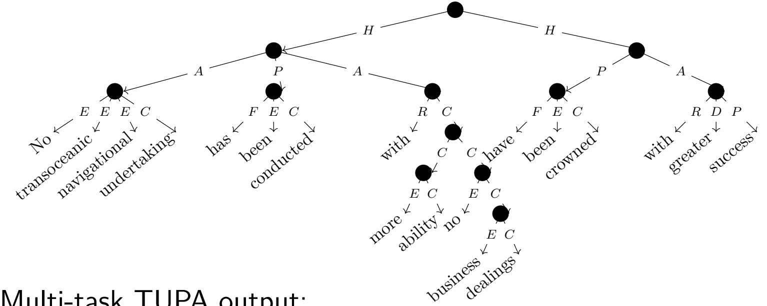
Improved UCCA parsing in English, French and German.

Multi-task

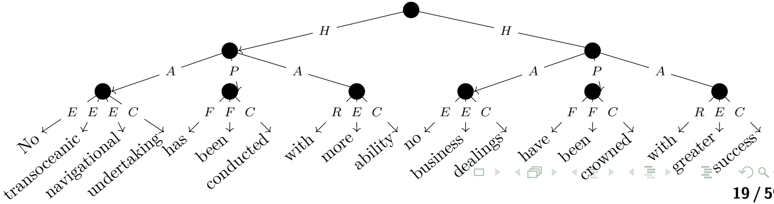
Multi-task TUPA model:



TUPA output:

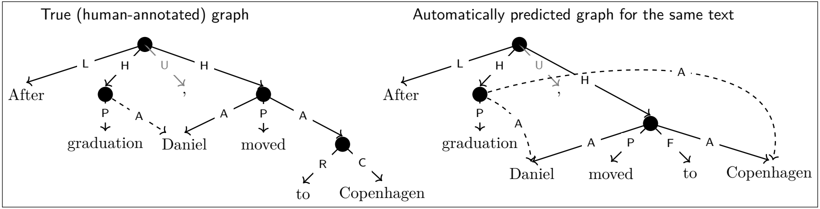


Multi-task TUPA output:
(+AMR+DM+UD)



Outline

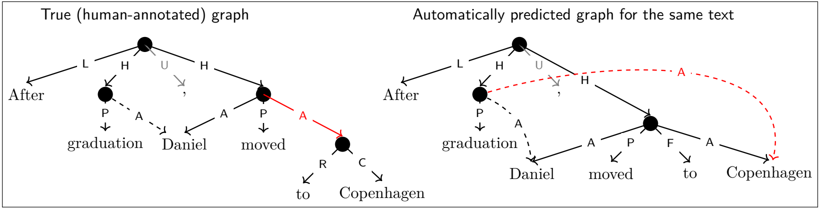
Unlabeled Evaluation of UCCA Parsing



1. Match terminal yield of primary edges.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

Primary			Remote		
UP	**UR**	**UF**	**UP**	**UR**	**UF**
$\frac{7}{7} = 100\%$	$\frac{7}{8} \approx 87\%$	$\approx 93\%$	$\frac{1}{2} = 50\%$	$\frac{1}{1} = 100\%$	$\approx 67\%$

Unlabeled Evaluation of UCCA Parsing



1. Match terminal yield of primary edges.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

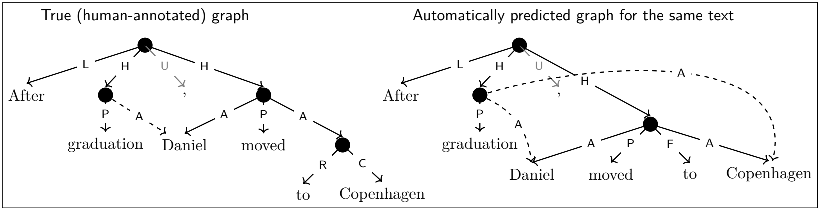
Primary

UP	UR	UF
$\frac{7}{7} = 100\%$	$\frac{7}{8} \approx 87\%$	$\approx 93\%$

Remote

UP	UR	UF
$\frac{1}{2} = 50\%$	$\frac{1}{1} = 100\%$	$\approx 67\%$

Unlabeled Evaluation of UCCA Parsing



1. Match terminal yield* of primary edges.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

Primary

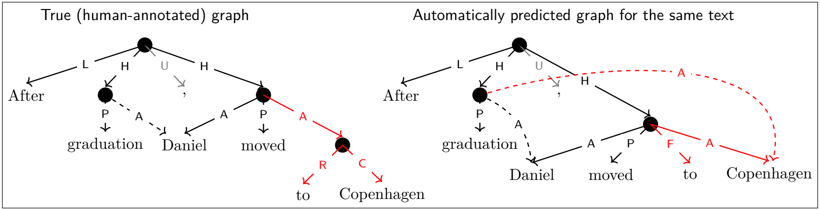
UP	UR	UF
$\frac{7}{7} = 100\%$	$\frac{7}{8} \approx 87\%$	$\approx 93\%$

Remote

UP	UR	UF
$\frac{1}{2} = 50\%$	$\frac{1}{1} = 100\%$	$\approx 67\%$

*Ignoring remotes; collapsing unary children

Labeled Evaluation of UCCA Parsing

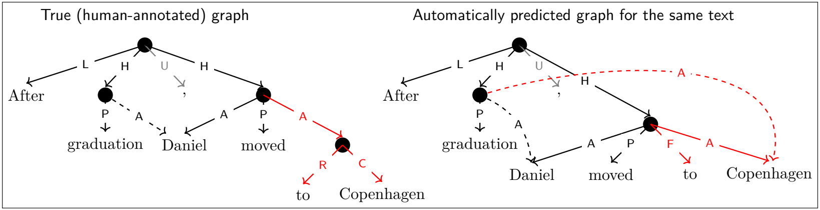


1. Match terminal yield* + **label** of primary edges.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

Primary			Remote		
LP	LR	LF	LP	LR	LF
$\frac{5}{7} \approx 71\%$	$\frac{5}{8} \approx 62\%$	$\approx 67\%$	$\frac{1}{2} = 50\%$	$\frac{1}{1} = 100\%$	$\approx 67\%$

*Ignoring remotes; collapsing unary children

Labeled Evaluation of UCCA Parsing



1. Match terminal yield* + **label** of primary edges.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

Primary

LP	LR	LF
$\frac{5}{7} \approx 71\%$	$\frac{5}{8} \approx 62\%$	$\approx 67\%$

Remote

LP	LR	LF
$\frac{1}{2} = 50\%$	$\frac{1}{1} = 100\%$	$\approx 67\%$

*Ignoring remotes; collapsing unary children

The Fine Print of UCCA Evaluation

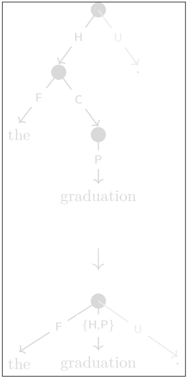
Punctuation edges ignored.

Yield excludes remotes, punctuation.

Unary children collapsed.
Correct yield $\Leftrightarrow$ label overlap ≥ 1.

Normalization:

1. Nested Centers flattened.
2. Common Functions attached to root.



The Fine Print of UCCA Evaluation

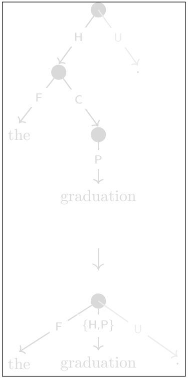
Punctuation edges ignored.

Yield excludes remotes, punctuation.

Unary children collapsed.
Correct yield $\Leftrightarrow$ label overlap ≥ 1.

Normalization:

1. Nested Centers flattened.
2. Common Functions attached to root.



The Fine Print of UCCA Evaluation

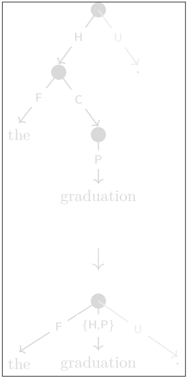
Punctuation edges ignored.

Yield excludes remotes, punctuation.

Unary children collapsed.
Correct yield $\Leftrightarrow$ label overlap ≥ 1.

Normalization:

1. Nested Centers flattened.
2. Common Functions attached to root.



The Fine Print of UCCA Evaluation

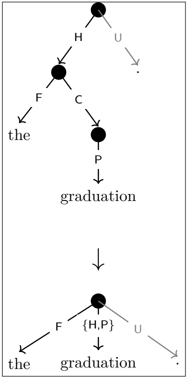
Punctuation edges ignored.

Yield excludes remotes, punctuation.

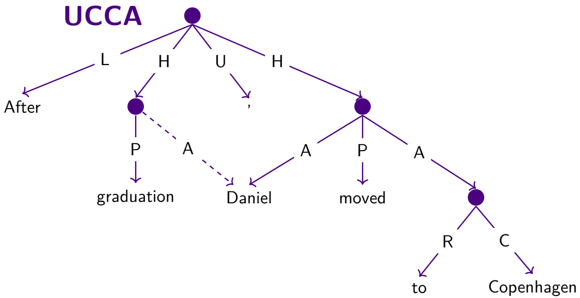
Unary children collapsed.
Correct yield $\Leftrightarrow$ label overlap ≥ 1.

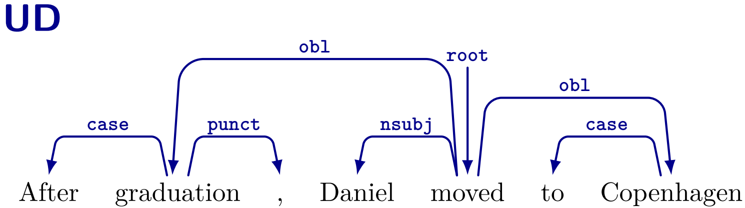
Normalization:

1. Nested Centers flattened.
2. Common Functions attached to root.



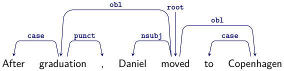
UCCA vs. UD


UCCA
L
H
U
H
After
P
A
graduation
,
A
P
A
Daniel
moved
R
C
to
Copenhagen

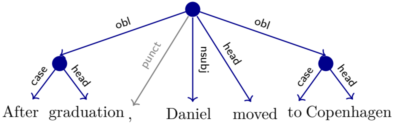

UD
obl
root
obl
case
punct
nsubj
case
After
graduation
,
Daniel
moved
to
Copenhagen

Assimilating the Graph Structures

UD



$\Rightarrow$



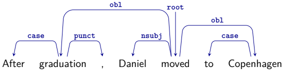
Unlabeled evaluation by matching (Hershcovich et al., 2019).

UCCA

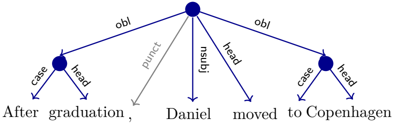
UP	UR	UF
$\frac{7}{8} \approx 87\%$	$\frac{7}{8} \approx 87\%$	$\approx 87\%$

Assimilating the Graph Structures

UD



$\Rightarrow$



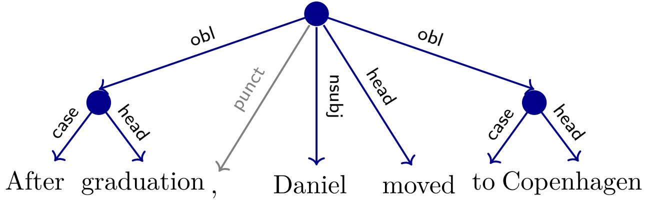
Unlabeled evaluation by matching (Hershcovich et al., 2019).

UCCA

UP	UR	UF
$\frac{7}{8} \approx 87\%$	$\frac{7}{8} \approx 87\%$	$\approx 87\%$

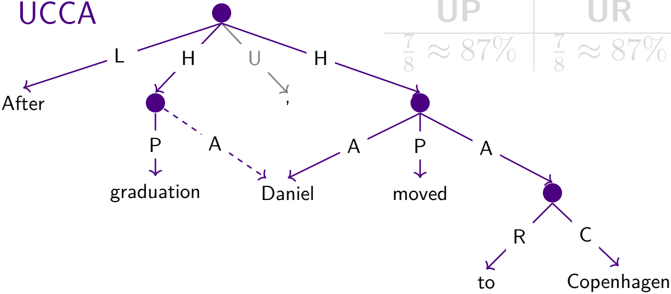
Assimilating the Graph Structures

UD



Unlabeled evaluation by matching (Hershcovich et al., 2019).

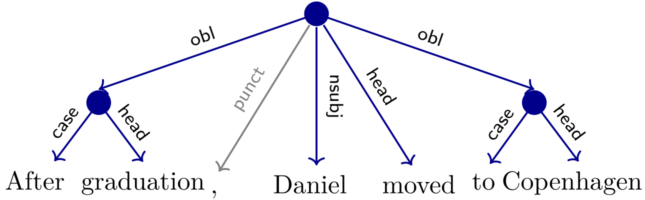
UCCA



UP	UR	UF
$\frac{7}{8} \approx 87\%$	$\frac{7}{8} \approx 87\%$	$\approx 87\%$

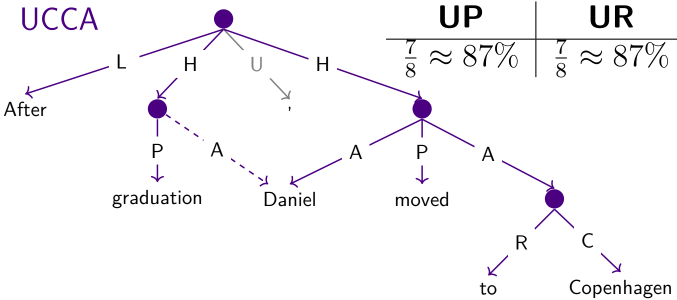
Assimilating the Graph Structures

UD



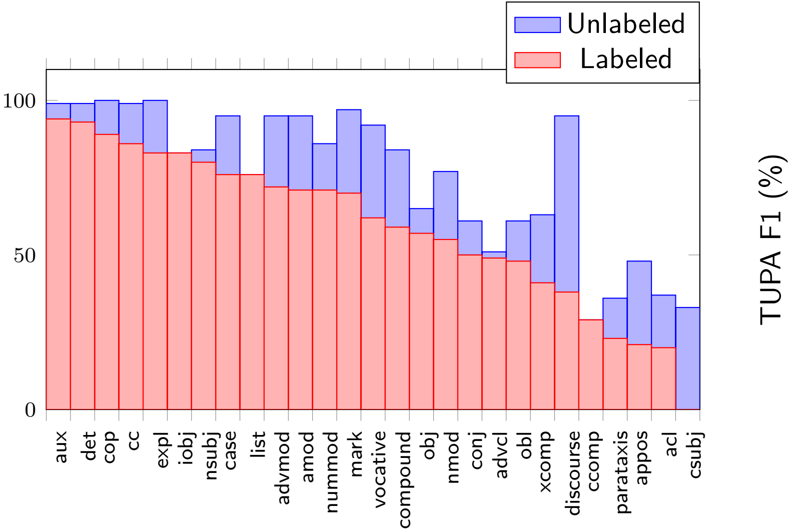
Unlabeled evaluation by matching (Hershcovich et al., 2019).

UCCA



UP	UR	UF
$\frac{7}{8} \approx 87\%$	$\frac{7}{8} \approx 87\%$	$\approx 87\%$

Fine-grained UCCA Parsing Evaluation



Shared Tasks: Parsing Competitions

SemEval 2019 Task 1

- UCCA parsing in English, French and German.

MRP 2019: Cross-Framework Meaning Representation Parsing

- DM, PSD, EDS, UCCA and AMR parsing in English.

MRP 2020: Cross-Framework and Cross-Lingual MRP

- EDS, PTG, UCCA, AMR and DRG parsing in English, Czech, German and Chinese.

Shared Tasks: Parsing Competitions

SemEval 2019 Task 1

- UCCA parsing in English, French and German.

MRP 2019: Cross-Framework Meaning Representation Parsing

- DM, PSD, EDS, UCCA and AMR parsing in English.

MRP 2020: Cross-Framework and Cross-Lingual MRP

- EDS, PTG, UCCA, AMR and DRG parsing in English, Czech, German and Chinese.

Shared Tasks: Parsing Competitions

SemEval 2019 Task 1

- UCCA parsing in English, French and German.

MRP 2019: Cross-Framework Meaning Representation Parsing

- DM, PSD, EDS, UCCA and AMR parsing in English.

MRP 2020: Cross-Framework and Cross-Lingual MRP

- EDS, PTG, UCCA, AMR and DRG parsing in English, Czech, German and Chinese.

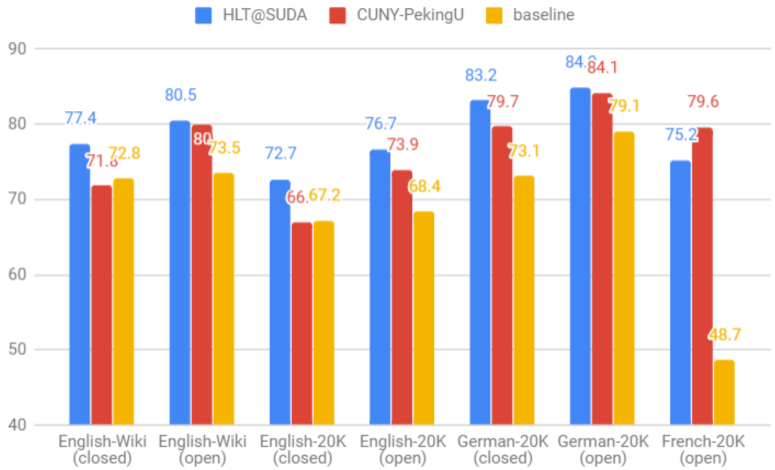
SemEval 2019 Task 1: Cross-lingual Semantic Parsing with UCCA

- UCCA parsing in English, French and German.
- 8 teams participated.
- Evaluation metric: UCCA graph score.
- Baseline: TUPA.

SemEval 2019 Task 1

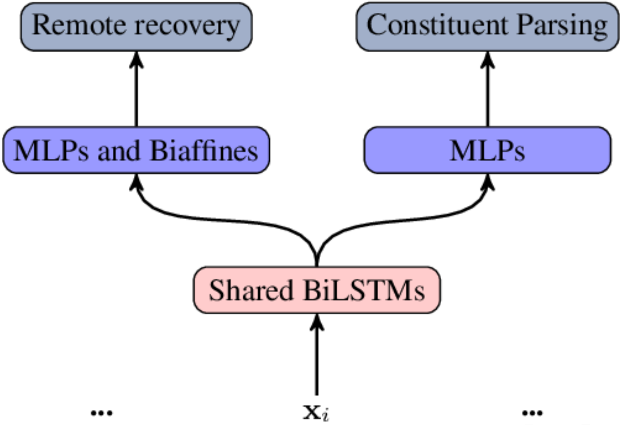
- English {in-domain/out-of-domain} × {open/closed}
- German in-domain {open/closed}
- French *low-resource* (only 15 training sentences)

SemEval 2019 Task 1



SemEval 2019 Task 1

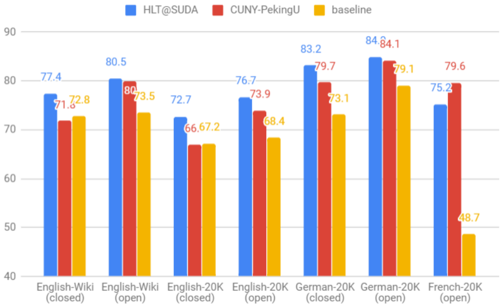
Winning system: HLT@SUDA (Jiang et al., 2019).
Neural constituency parser + multi-task + BERT.
Multilingual training with language embedding.



SemEval 2019 Task 1

Results in French were close to English and German

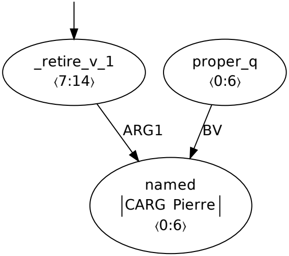
- Demonstrates viability of cross-lingual UCCA parsing
- Is this because of UCCA's stability in translation?



More in part 6: cross-linguistic studies!

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
-



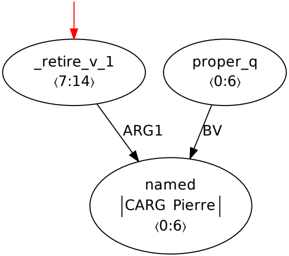
Pierre retired.

Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
- tops,



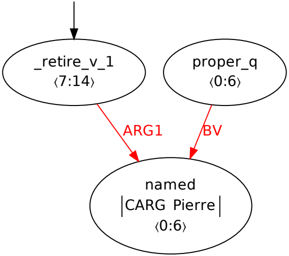
Pierre retired.

Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
- tops, (labeled) edges,



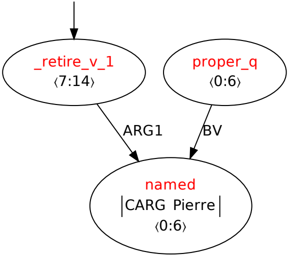
Pierre retired.

Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
- tops, (labeled) edges, labels,



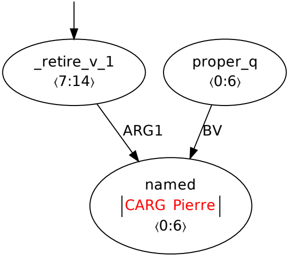
Pierre retired.

Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
- tops, (labeled) edges, labels, properties,



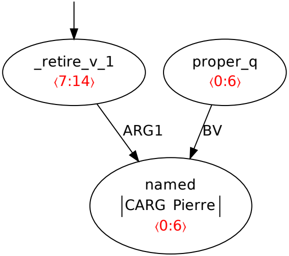
Pierre retired.

Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
- tops, (labeled) edges, labels, properties, anchors,



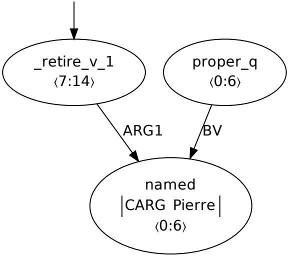
Pierre retired.

Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

Cross-Framework Evaluation: MRP

- Break down graphs into per-type and overall F_1.
- tops, (labeled) edges, labels, properties, anchors, attributes.



Pierre retired.

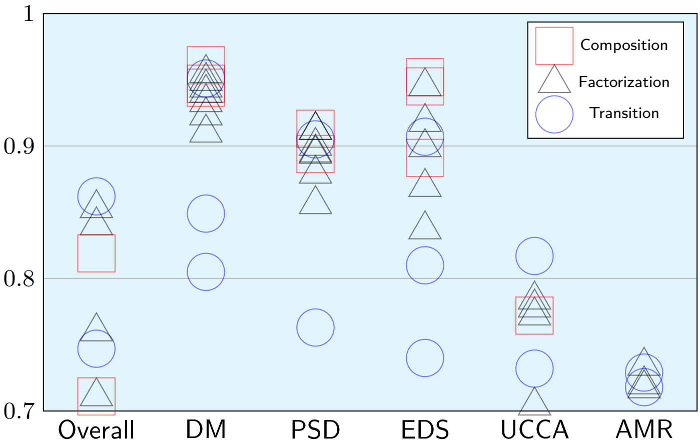
Types of Semantic Graph 'Atoms'

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Labeled Edges	✓	✓	✓	✓	(✓)
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchoring	✓	(✓)	(✓)	✗	✗
Edge Attributes	✗	✓	✓	✗	✗

MRP 2019

- **DM, PSD, EDS, UCCA and AMR** parsing in English.
- 18 teams participated.
- Evaluation metric: MRP score.
- Baseline: TUPA (generalized beyond UCCA).

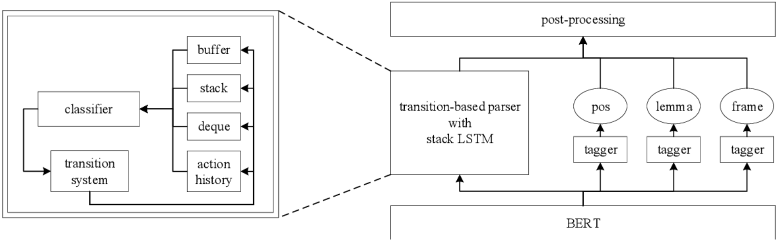
MRP 2019



1
0.9
0.8
0.7
Overall
DM
PSD
EDS
UCCA
AMR
Composition
Factorization
Transition

MRP 2019

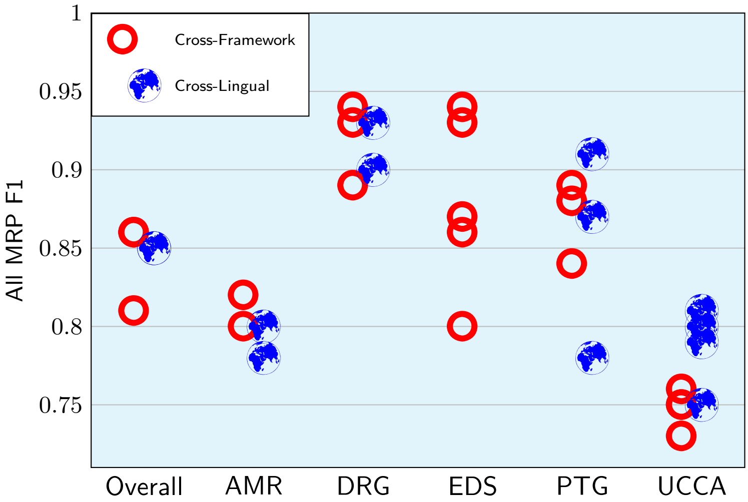
Winning system: HIT-SCIR (Che et al., 2019).
Transition-based parser + efficient training + BERT.



MRP 2020

- EDS (English),
  PTG (English and Czech),
  UCCA (English and **German**),
  AMR (English and Chinese) and
  DRG (English and German) parsing.
- 8 teams participated.
- Evaluation metric: MRP score.

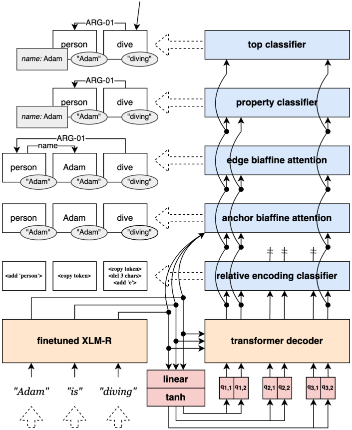
MRP 2020

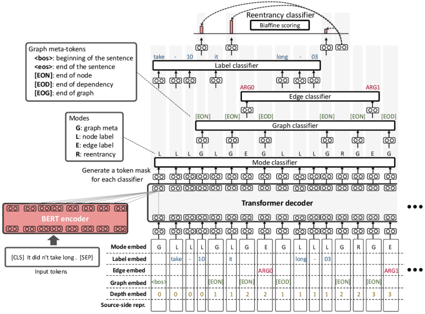


1
0.95
0.9
0.85
0.8
0.75
All MRP F1
Cross-Framework
Cross-Lingual
Overall
AMR
DRG
EDS
PTG
UCCA

MRP 2020

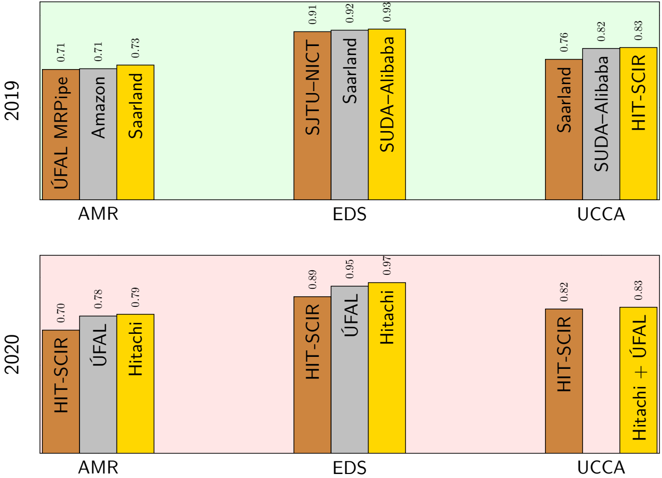
Winning systems: encoder-decoder with transformers.
ÚFAL (Samuel and Straka, 2020), Hitachi (Ozaki et al., 2020).





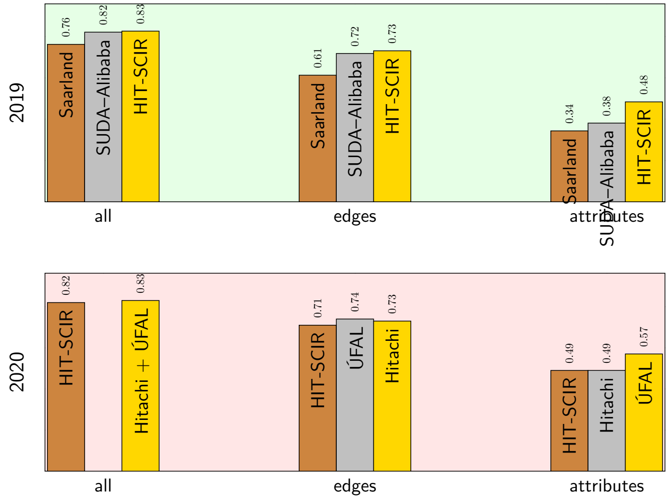
State of the Art in MRP

MRP F1 on English *The Little Prince* evaluation set:



State of the Art in MRP

UCCA **fine-grained** MRP F1 on English *The Little Prince*:*



*Attribute `remote=True` in 5% of edges.

Outline

What can meaning representation do for NLP?

- Incorporating linguistically informed rules
- Controlled evaluation by explicit criteria
- Inductive bias to facilitate learning
- Explainable models by design

Outline

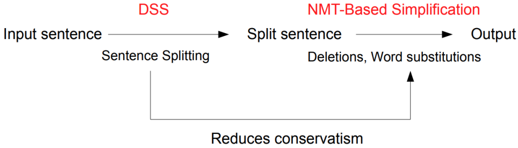
Sentence Splitting for Text Simplification

Last year I read the book Daniel authored →

Daniel wrote a book. I read the book.

MT-based simplifcation is *overly conservative*.

Direct Semantic Splitting before MT-based simplification to place each scene in its own sentence (Sulem et al., 2018c).

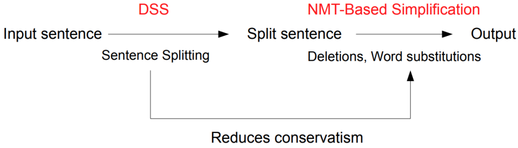


Sentence Splitting for Text Simplification

Last year I read the book Daniel authored →
Daniel wrote a book. I read the book.

MT-based simplifcation is *overly conservative*.

Direct Semantic Splitting before MT-based simplification to place each scene in its own sentence (Sulem et al., 2018c).

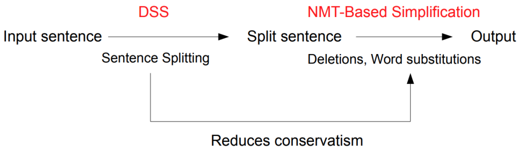


Sentence Splitting for Text Simplification

Last year I read the book Daniel authored $\rightarrow$
Daniel wrote a book. I read the book.

MT-based simplifcation is *overly conservative*.

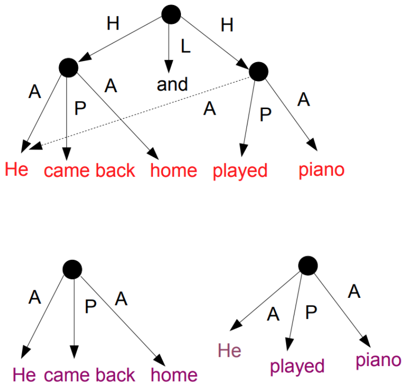
Direct Semantic Splitting before MT-based simplification to place each scene in its own sentence (Sulem et al., 2018c).



Rule 1:

The Semantic Rules

Parallel Scenes



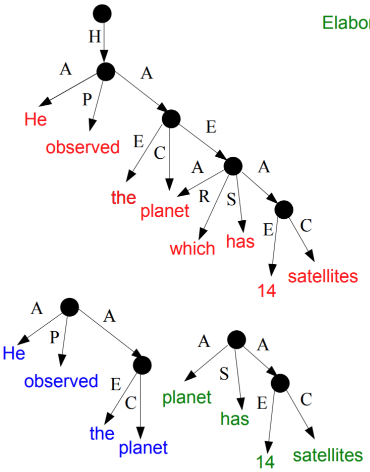
He came back home and played piano.

He came back home. He played piano.

Rule 2:

The Semantic Rules

Elaborator Scenes



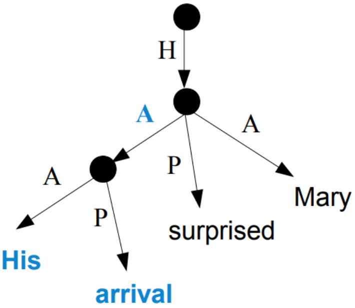
He observed the planet which has 14 satellites.

He observed the planet. Planet has 14 satellites.

29

Sentence Splitting for Text Simplification

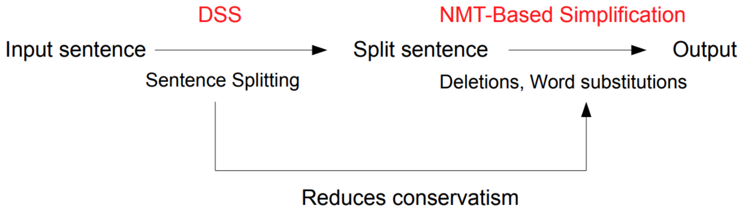
Participant scenes are not split.



Sentence Splitting for Text Simplification

He observed the planet. Planet has 14 satellites.

Neural MT methods to fix grammaticality.

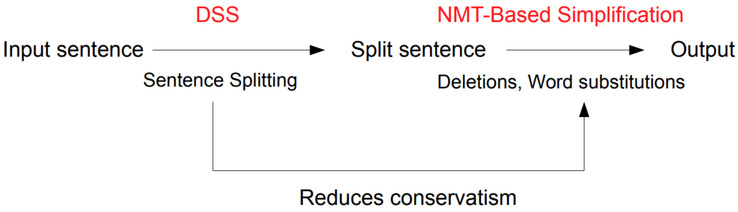


He observed the planet. **The planet** has 14 satellites.

Sentence Splitting for Text Simplification

He observed the planet. Planet has 14 satellites.

Neural MT methods to fix grammaticality.

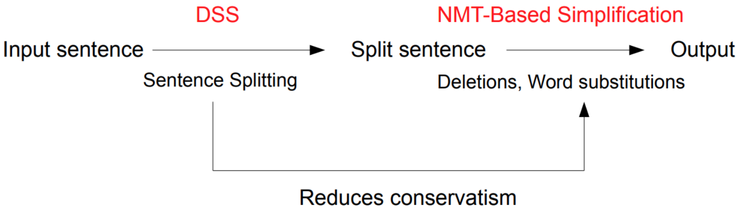


He observed the planet. **The planet** has 14 satellites.

Sentence Splitting for Text Simplification

He observed the planet. Planet has 14 satellites.

Neural MT methods to fix grammaticality.



He observed the planet. **The planet** has 14 satellites.

Outline

BLEU is Not Suitable for the Evaluation of Text Simplification

BLEU: reference-based evaluation metric for MT, also widely used to evaluate text simplification.

With sentence splitting, not correlated with grammaticality or meaning preservation (Sulem et al., 2018a).

Negatively correlated with simplicity!

BLEU is Not Suitable for the Evaluation of Text Simplification

BLEU: reference-based evaluation metric for MT, also widely used to evaluate text simplification.

With sentence splitting, not correlated with grammaticality or meaning preservation (Sulem et al., 2018a).

Negatively correlated with simplicity!

BLEU is Not Suitable for the Evaluation of Text Simplification

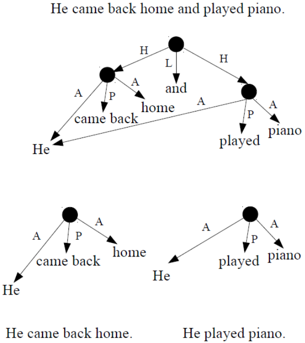
BLEU: reference-based evaluation metric for MT, also widely used to evaluate text simplification.

With sentence splitting, not correlated with grammaticality or meaning preservation (Sulem et al., 2018a).

Negatively correlated with simplicity!

Semantic Structural Evaluation for Text Simplification

SAMSA: reference-less measure of *structural simplicity* and *meaning preservation* (Sulem et al., 2018b).
Same principle: *one scene per sentence*.



Semantic Structural Evaluation for Text Simplification

Word alignment

Scene John gave Mary a call

Sentence John called Mary

UCCA annotation

[John]$_A$ [gave$_F$]$_{P-}$ **[Mary]**$_A$ [a$_E$ **call**$_C$]$_{-P(CONT.)}$

Suppose the Scene *Sc* is matched to the sentence *Sen*:

$$Score_{Sen}(Sc) = \frac{1}{2}\left(Score_{Sen}(MR) + \frac{1}{K}\sum_{i=1}^{K} Score_{Sen}(Par_k)\right)$$

MR - **Minimal center** of the Main Relation (Process / State)
Par_k - **Minimal center** of the kth Participant

$$Score_{Sen}(u) = \begin{cases} 1 & u \text{ is aligned to a word in } Sen \\ 0 & \text{otherwise} \end{cases}$$

19

- Average over the input Scenes
- Non-splitting penalty: $\frac{n_{out}}{n_{inp}}$

n_{out} → Number of output sentences

n_{inp} → Number of input Scenes

Grammatical Error Correction

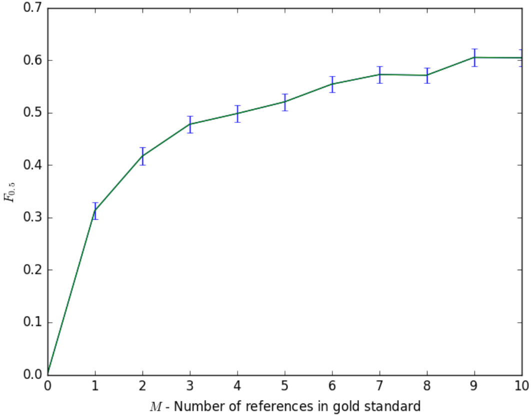
Another text-to-text generation task.

Ther is both sides of stories $\rightarrow$

There are two sides to every story

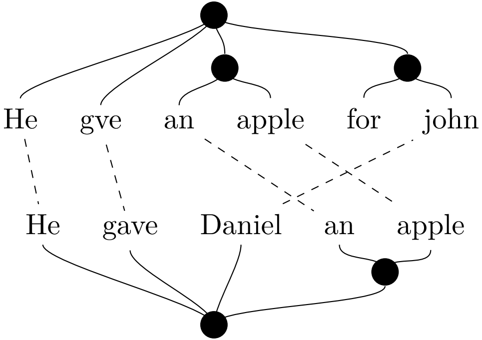
Inherent Biases in Reference-based Evaluation for GEC and Text Simplification

Using *references* for GEC evaluation underestimates performance (Choshen and Abend, 2018a).



Reference-less Measure of Faithfulness for Grammatical Error Correction

- UCCA is *applicable* to ungrammatical learner language!
- UCCA is *stable* with respect to grammar corrections



Reference-less Measure of Faithfulness for Grammatical Error Correction

USim measures meaning preservation automatically *without references* (Choshen and Abend, 2018b).

Variation on standard UCCA evaluation, using unit *alignment* between the source and target graphs.

Sensitive to faithfulness, not overly conservative.

Source	the good student must know how to understand and work hard to get the iede.
Reference	A good student must be able to understand and work hard to get the idea.
Corrector	The good student must know how to understand and work hard to get on.

Reference-less Measure of Faithfulness for Grammatical Error Correction

USim measures meaning preservation automatically *without references* (Choshen and Abend, 2018b).

Variation on standard UCCA evaluation, using unit *alignment* between the source and target graphs.

Sensitive to faithfulness, not overly conservative.

Source	the good student must know how to understand and work hard to get the iede.
Reference	A good student must be able to understand and work hard to get the idea.
Corrector	The good student must know how to understand and work hard to get on.

Reference-less Measure of Faithfulness for Grammatical Error Correction

USim measures meaning preservation automatically *without references* (Choshen and Abend, 2018b).

Variation on standard UCCA evaluation, using unit *alignment* between the source and target graphs.

Sensitive to faithfulness, not overly conservative.

Source	the good student must know how to understand and work hard to get the iede.
Reference	A good student must be able to understand and work hard to get the idea.
Corrector	The good student must know how to understand and work hard to get on.

Universal Conceptual Cognitive Annotation

Cross-lingual Semantic Representation for NLP with UCCA

Daniel Hershcovich
University of Copenhagen

Parsing, Evaluation and Applications

References

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 76 – 85, Hong Kong, China.

Leshem Choshen and Omri Abend. 2018a. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Melbourne, Australia. Association for Computational Linguistics.

Leshem Choshen and Omri Abend. 2018b. Reference-less measure of faithfulness for grammatical error correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 124–129, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL*, pages 1127–1138.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proc. of ACL*, pages 373–385.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2019. Content differences in syntactic and semantic representation. In *Proc. of NAACL-HLT*.

Wei Jiang, Zhenghua Li, Yu Zhang, and Min Zhang. 2019. HLT@SUDA at SemEval-2019 task 1: UCCA graph parsing as constituent tree parsing. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 11–15, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Hiroaki Ozaki, Gaku Morio, Yuta Koreeda, Terufumi Morishita, and Toshinori Miyoshi. 2020. Hitachi at MRP 2020: Text-to-graph-notation transducer. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 40 – 52, Online.

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53 – 64, Online.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.