Universal Semantic Parsing with Neural Networks

Daniel Hershcovich
Advisors: Ari Rappoport and Omri Abend

PhD Lecture

February 5, 2019

Natural Language Processing: What's It Good For?

Natural Language Processing: What’s It Good For?

Machine translation:

After graduation, John moved to Copenhagen

↓

ג'ון עבר לקופנהגן אחרי שסיים את הלימודים

Natural Language Processing: What's It Good For?

Named entity recognition:

ג'ון עבר לקופנהגן אחרי שסיים את הלימודים

↓ ↓

Location Person

Natural Language Processing: What's It Good For?

Text simplification:

ג'ון עבר לקופנהגן אחרי שסיים את הלימודים

↓

ג'ון סיים את הלימודים. ג'ון עבר לקופנהגן.

Natural Language Processing: What's It Good For?

After graduation, John moved to Copenhagen

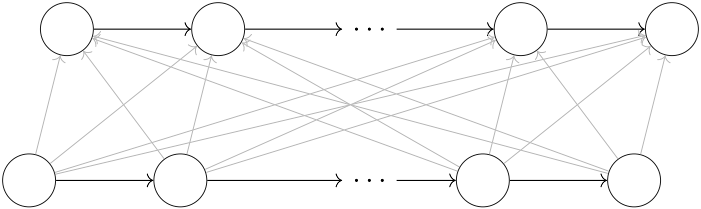
↓

ג'ון עבר לקופנהגן אחרי שסיים את הלימודים

↓

ג'ון סיים את הלימודים. ג'ון עבר לקופנהגן.

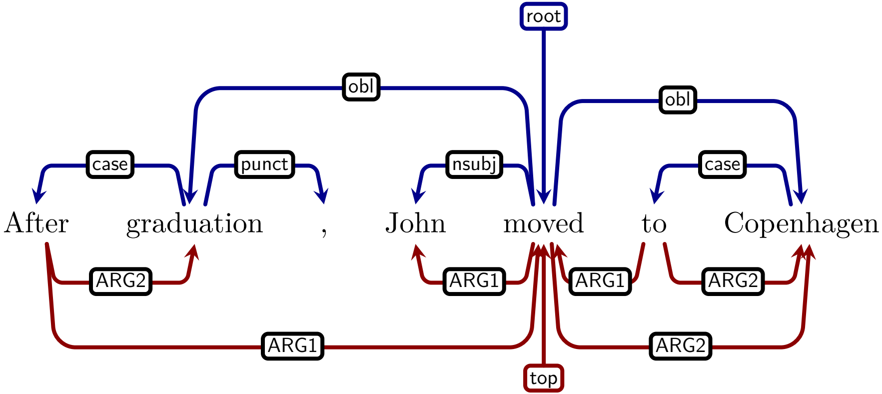
Sequence-to-sequence sometimes works, but lacks inductive bias.



Linguistic Structured Representations

Model explicit relations between words or concepts.

Example: syntactic/semantic bi-lexical dependencies.



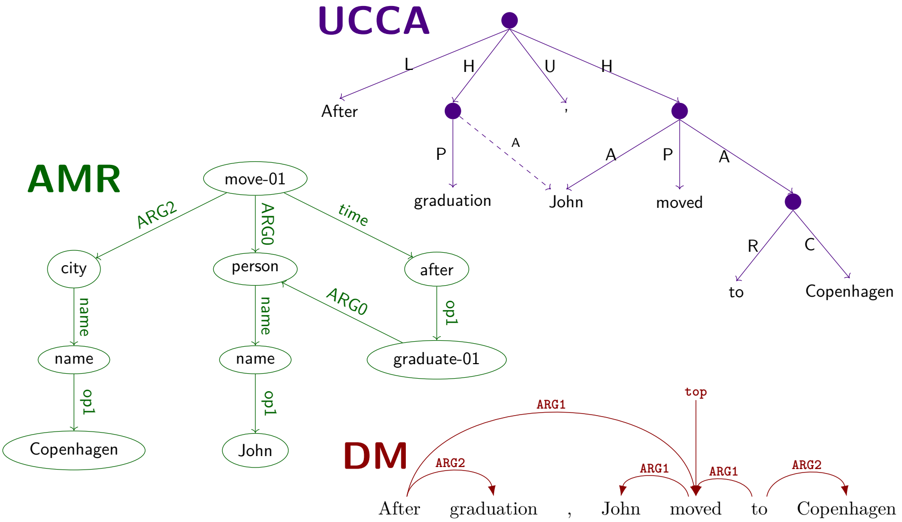
Semantic Representations

Abstract away from detail that does not affect meaning:

$$\boxed{\text{rest}} \approx \boxed{\text{take a break}}$$

$$\boxed{\text{graduation}} \approx \boxed{\text{סיים את הלימודים}}$$

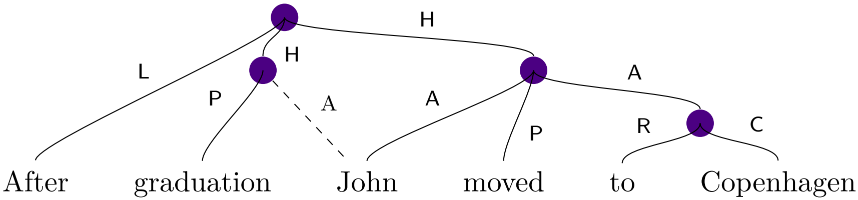
Semantic Representations



Outline

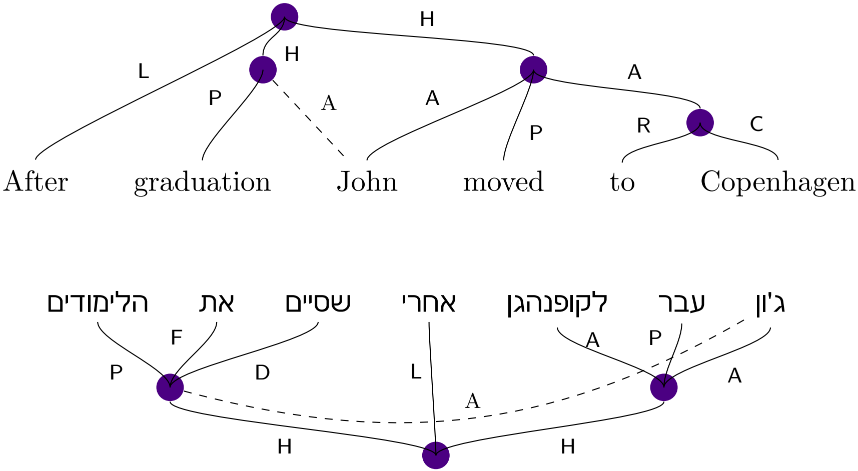
Universal Conceptual Cognitive Annotation (UCCA)

Supports rapid and intuitive annotation of linguistic semantic phenomena. [Abend and Rappoport, 2013]



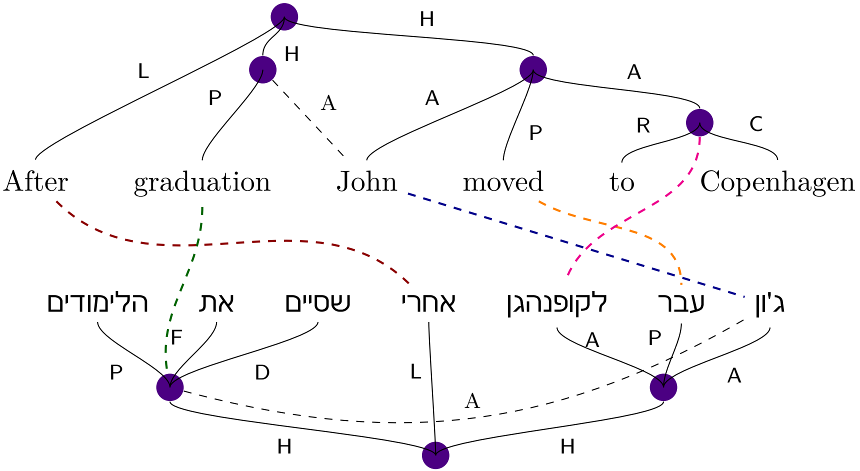
Universal Conceptual Cognitive Annotation (UCCA)

Supports rapid and intuitive annotation of linguistic semantic phenomena.
Cross-linguistically applicable and stable [Sulem et al., 2015].



Universal Conceptual Cognitive Annotation (UCCA)

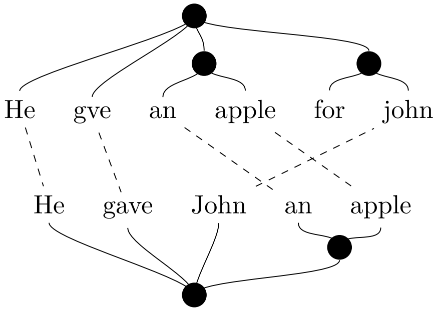
Supports rapid and intuitive annotation of linguistic semantic phenomena.
Cross-linguistically applicable and stable [Sulem et al., 2015].



UCCA Applications

Semantics-based **evaluation** of

- Machine translation [Birch et al., 2016].
- Text simplification [Sulem et al., 2018a].
- Grammatical error correction [Choshen and Abend, 2018].

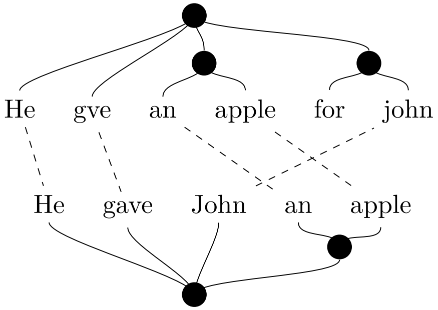


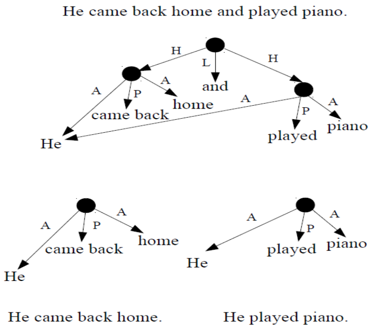
UCCA Applications

Semantics-based **evaluation** of

- Machine translation [Birch et al., 2016].
- Text simplification [Sulem et al., 2018a].
- Grammatical error correction [Choshen and Abend, 2018].

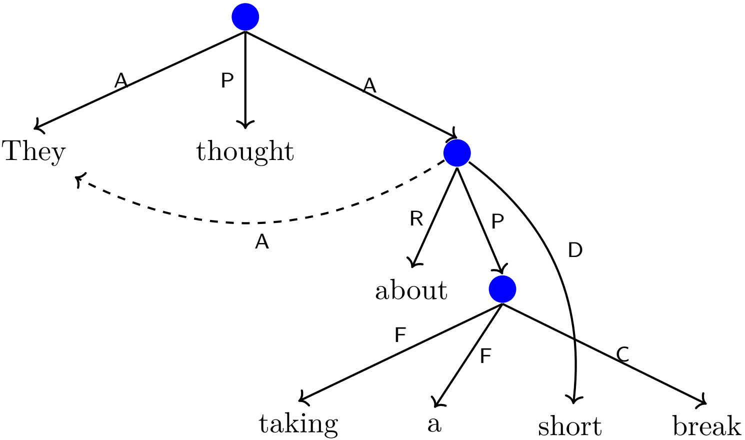
Sentence splitting for text simplification [Sulem et al., 2018b].





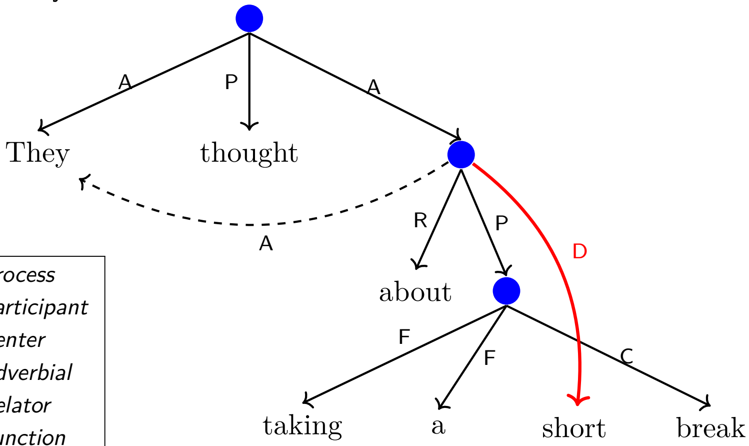
Graph Structure

UCCA structures are directed acyclic graphs (DAGs) with labeled edges.
Text tokens are terminals, complex units are non-terminal nodes.



Graph Structure

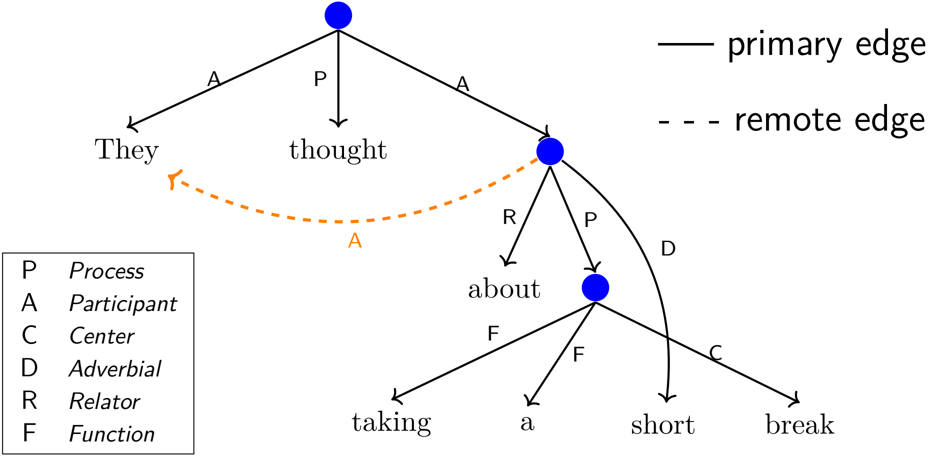
UCCA structures are directed acyclic graphs (DAGs) with labeled edges.
Text tokens are terminals, complex units are non-terminal nodes.
Phrases may be discontinuous.



P	*Process*
A	*Participant*
C	*Center*
D	*Adverbial*
R	*Relator*
F	*Function*

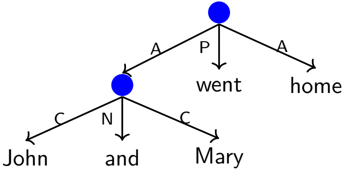
Graph Structure

UCCA structures are directed acyclic graphs (DAGs) with labeled edges.
Text tokens are terminals, complex units are non-terminal nodes.
Phrases may be discontinuous. *Remote edges* enable reentrancy.

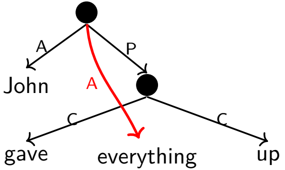


Structural Properties

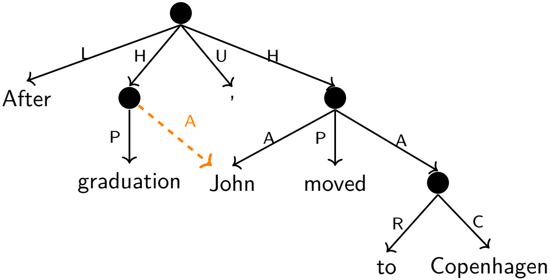
(1) non-terminal nodes



(2) discontinuity

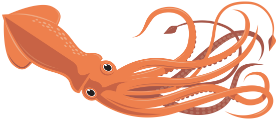


(3) reentrancy



UCCA Data

- English Wikipedia articles (Wiki).
- English-French-German parallel corpus from *Twenty Thousand Leagues Under the Sea* (20K).
- Reviews from the English Web Treebank (EWT).



Data Statistics

	Wiki	20K			EWT
	en	en	fr	de	en
# sentences	5,141	492	492	6,514	3,520
# tokens	158,739	12,638	13,021	144,529	51,042
# 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

Outline

1. Background: The UCCA Semantic Representation Scheme
2. A Transition-Based DAG Parser for UCCA (ACL'17)
3. Multitask Parsing across Semantic Representations (ACL'18)
4. Content Differences between Syntactic and Semantic Representations (under submission)

TUPA: 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.

TUPA: 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

TUPA: 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:

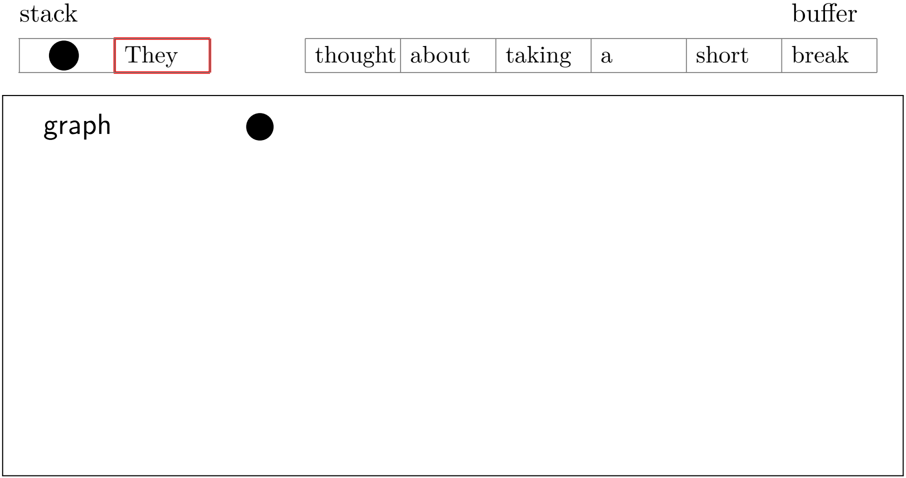
They	thought	about	taking	a	short	break

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

These transitions enable non-terminal nodes, reentrancy and discontinuity.

Example: TUPA Transition Sequence

⇒ SHIFT



Example: TUPA Transition Sequence

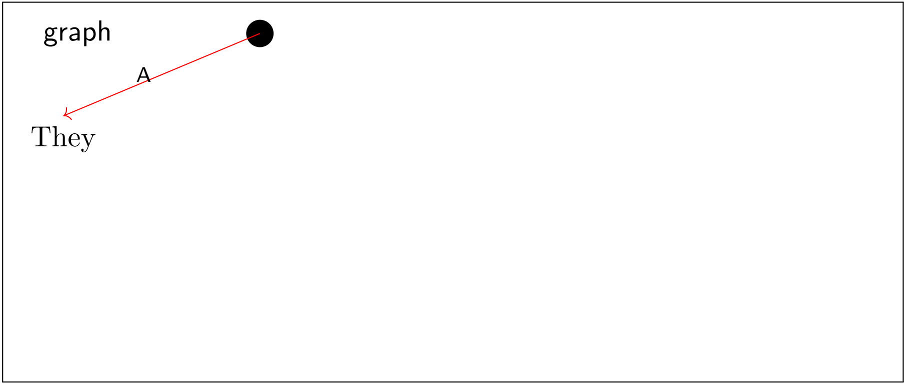
$\Rightarrow$ RIGHT-EDGE$_A$

stack

●	They

buffer

thought	about	taking	a	short	break

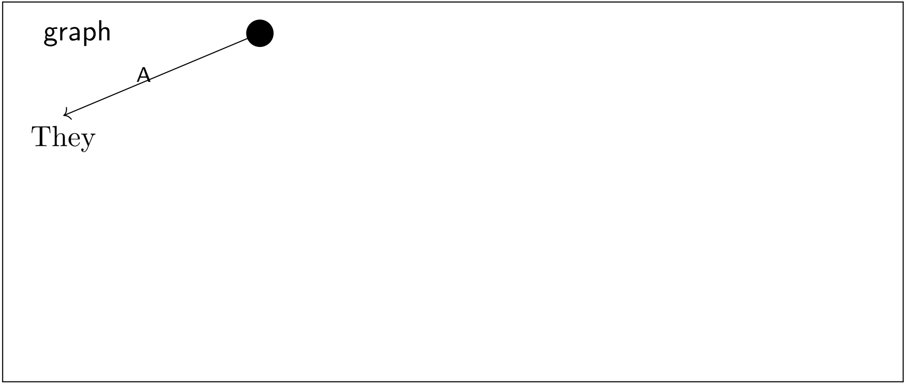


Example: TUPA Transition Sequence

⇒ Shift

stack | | | buffer

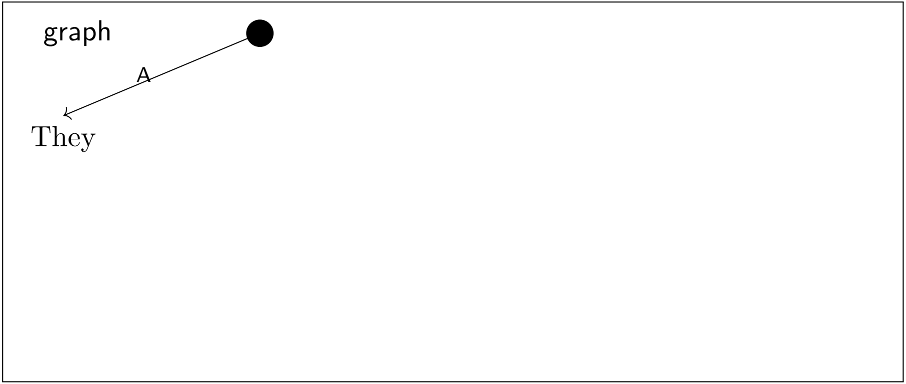
●	They	thought	about	taking	a	short	break



Example: TUPA Transition Sequence

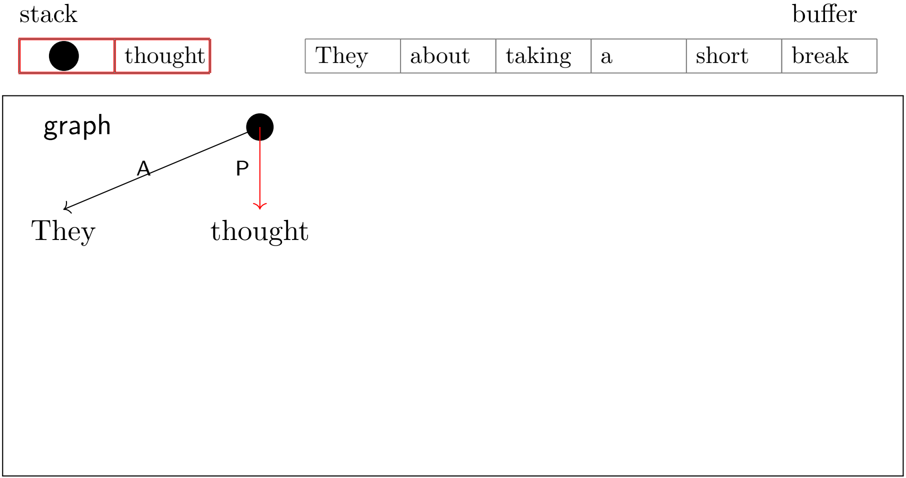
⇒ Swap

stack			buffer					
●	thought		They	about	taking	a	short	break



Example: TUPA Transition Sequence

$\Rightarrow$ $\text{R}\textsc{ight}\text{-E}\textsc{dge}_P$



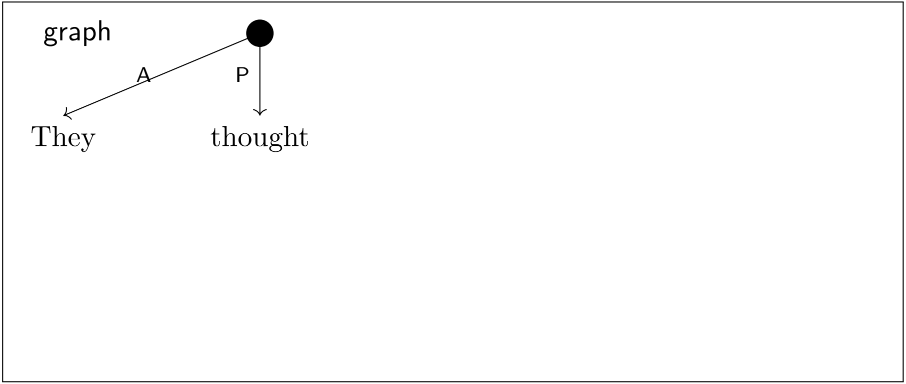
Example: TUPA Transition Sequence

⇒ Reduce

stack

buffer

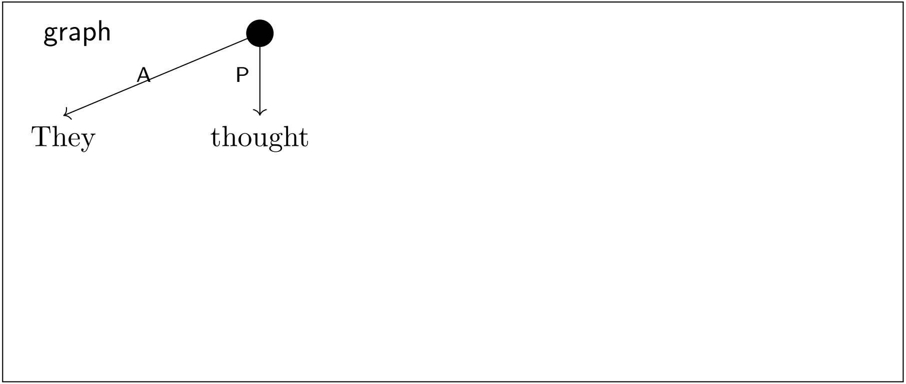
They	about	taking	a	short	break



Example: TUPA Transition Sequence

$\Rightarrow$ SHIFT

stack			buffer				
●	They		about	taking	a	short	break

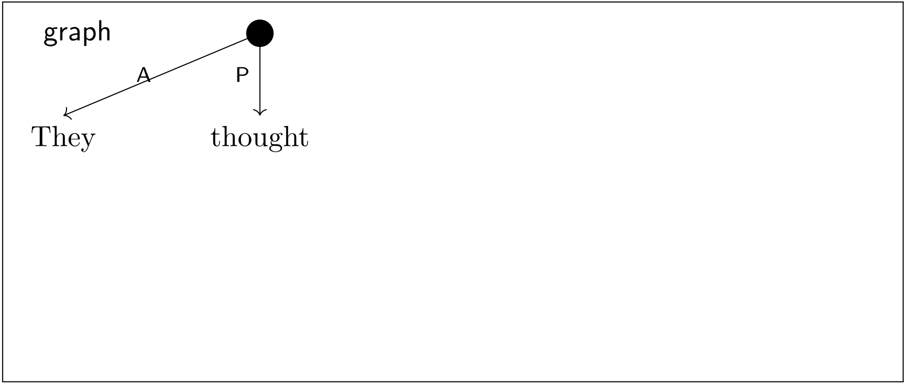


Example: TUPA Transition Sequence

⇒ SHIFT

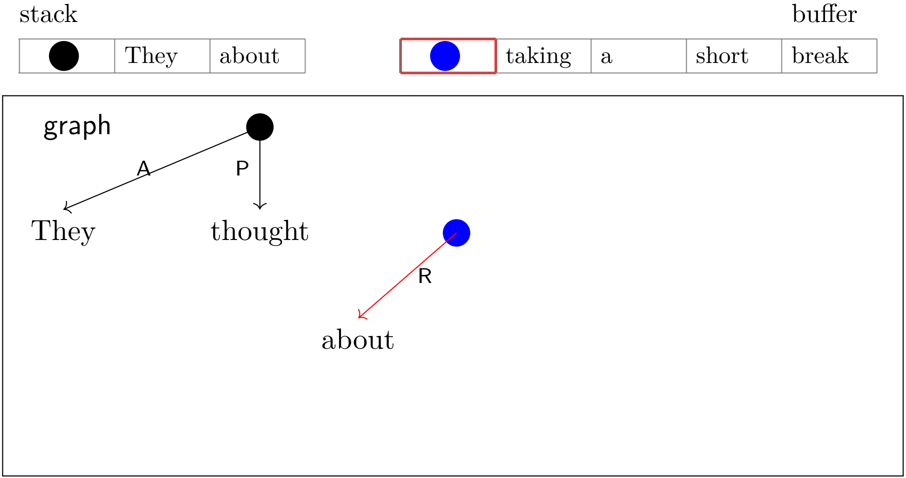
stack | buffer

●	They	about	taking	a	short	break



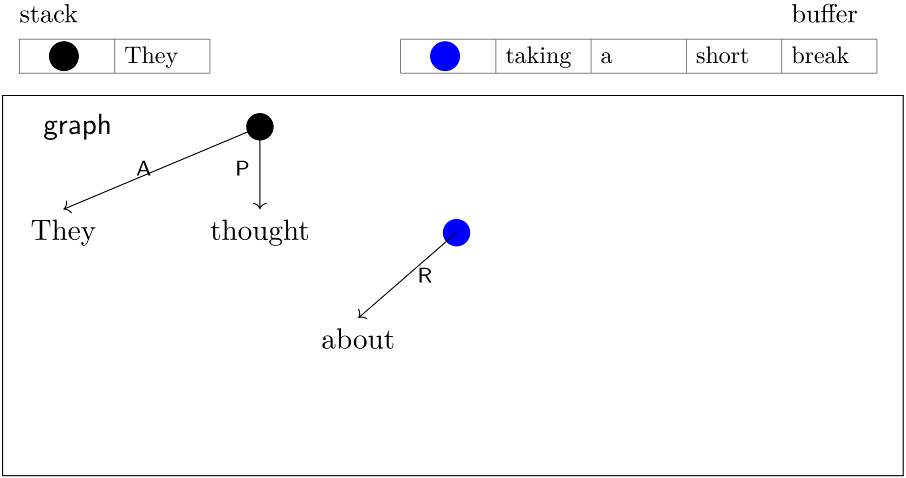
Example: TUPA Transition Sequence

$\Rightarrow$ NODE$_R$



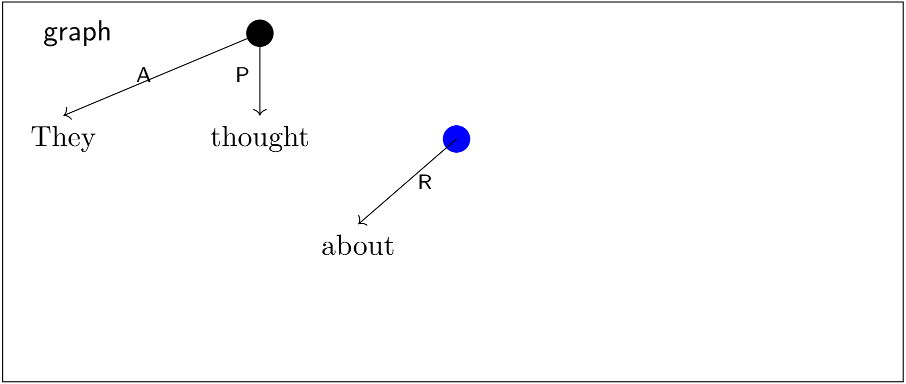
Example: TUPA Transition Sequence

⇒ REDUCE



Example: TUPA Transition Sequence

⇒ SHIFT



Example: TUPA Transition Sequence

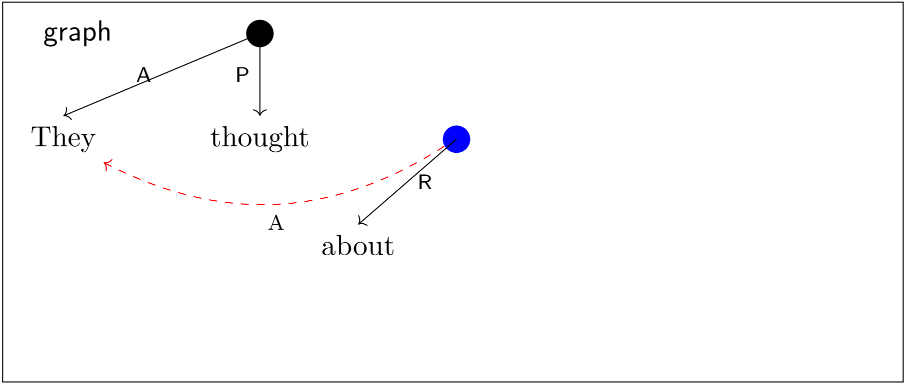
⇒ LEFT-REMOTE$_A$

stack

●	They	●

buffer

taking	a	short	break



Example: TUPA Transition Sequence

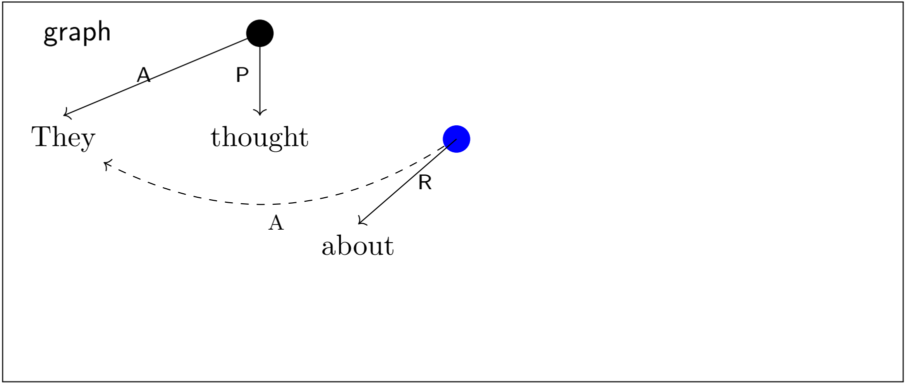
⇒ SHIFT

stack

●	They	●	taking

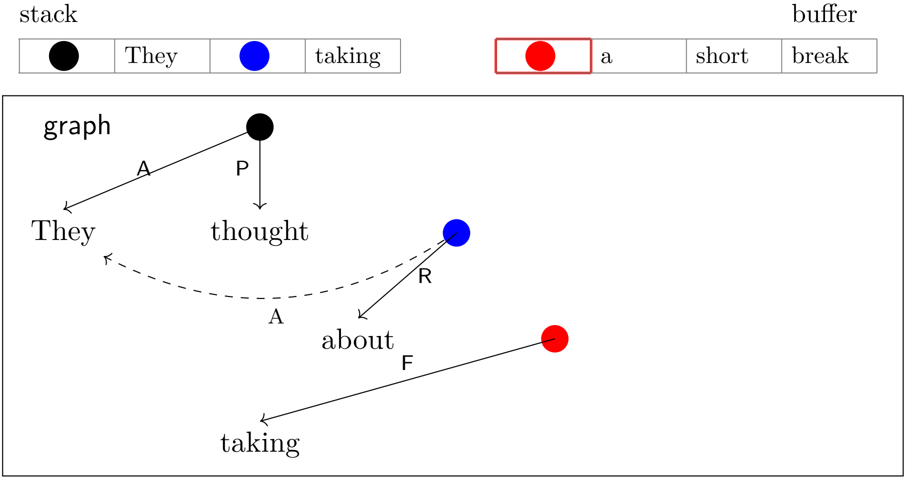
buffer

a	short	break



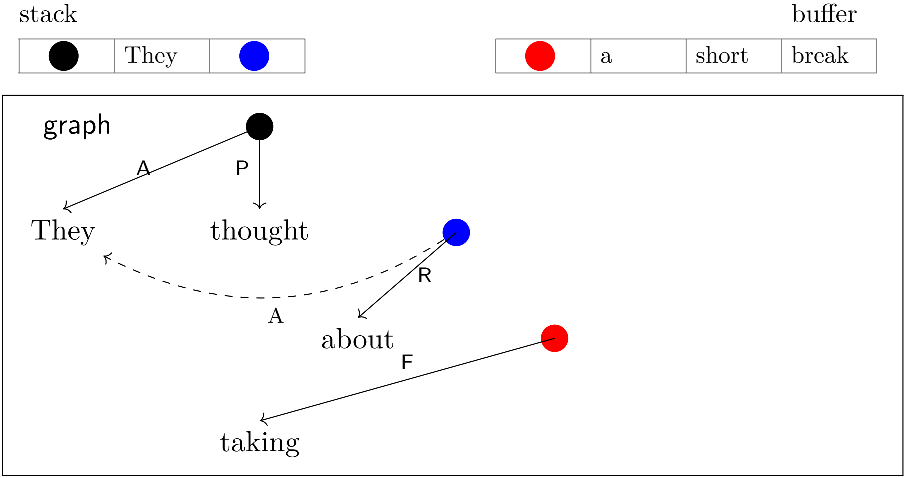
Example: TUPA Transition Sequence

$\Rightarrow$ NODE$_C$



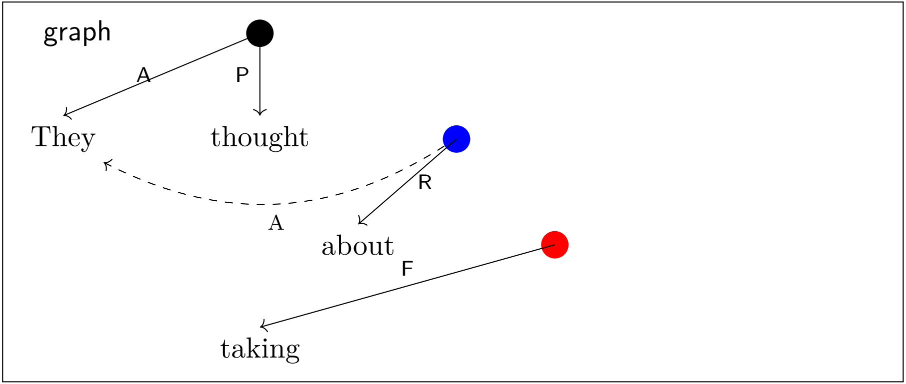
Example: TUPA Transition Sequence

⇒ REDUCE



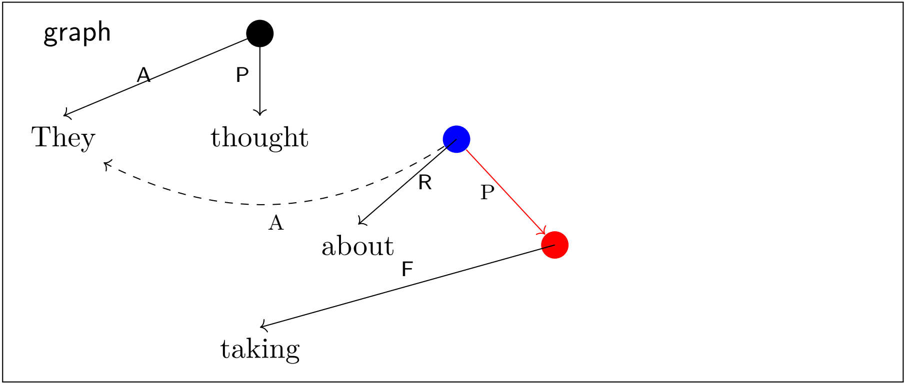
Example: TUPA Transition Sequence

⇒ SHIFT



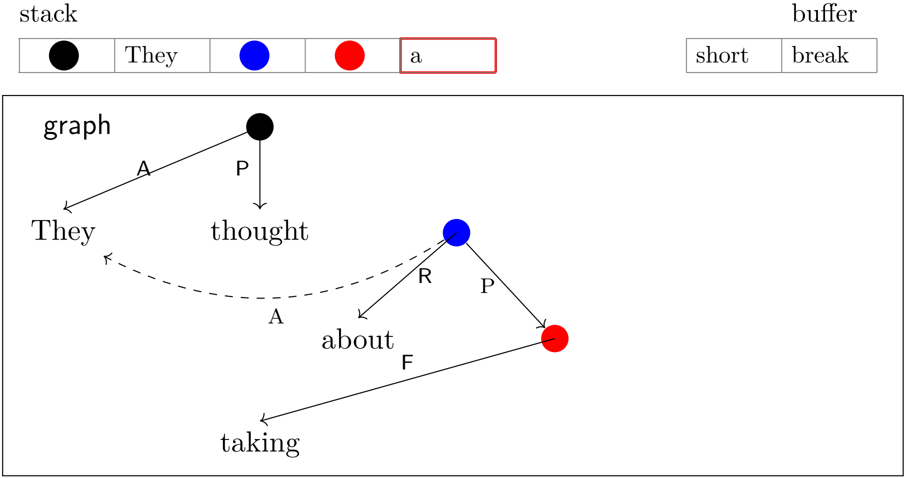
Example: TUPA Transition Sequence

⇒ RIGHT-EDGE$_P$



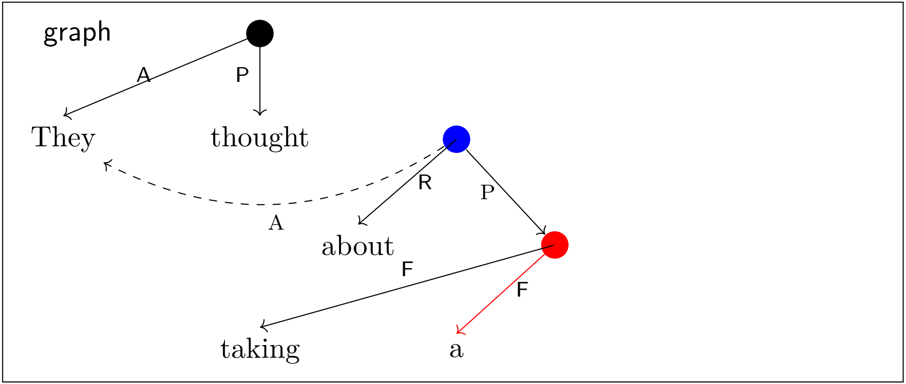
Example: TUPA Transition Sequence

⇒ SHIFT



Example: TUPA Transition Sequence

⇒ RIGHT-EDGE$_F$



Example: TUPA Transition Sequence

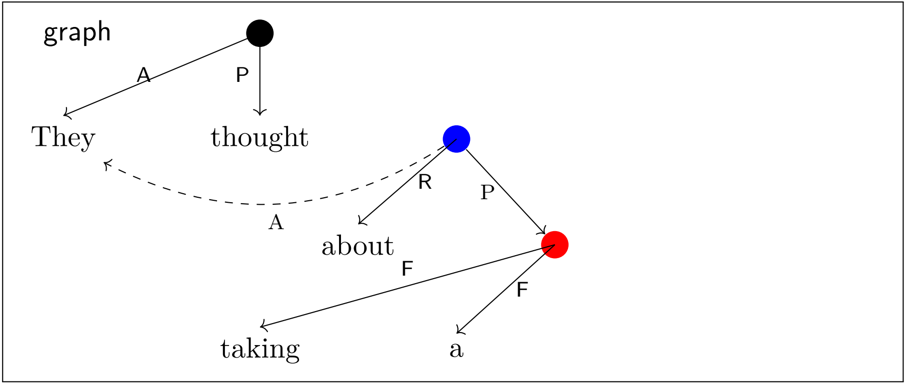
⇒ Reduce

stack

●	They	●	●

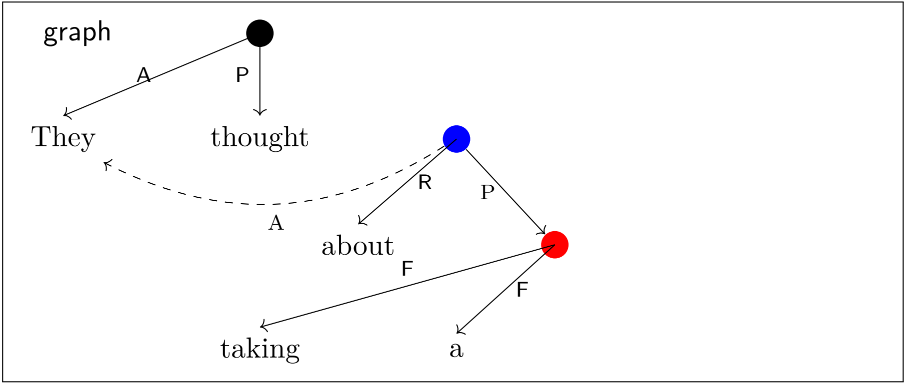
buffer

short	break



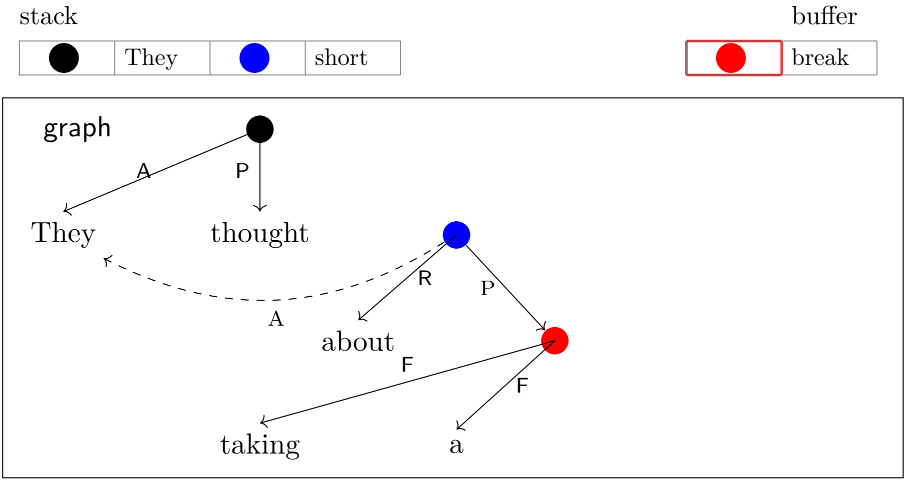
Example: TUPA Transition Sequence

⇒ SHIFT



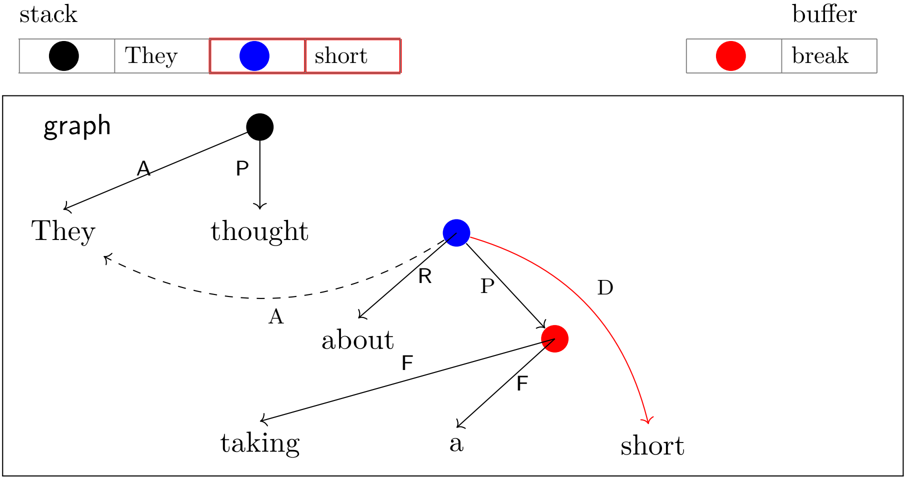
Example: TUPA Transition Sequence

⇒ SWAP



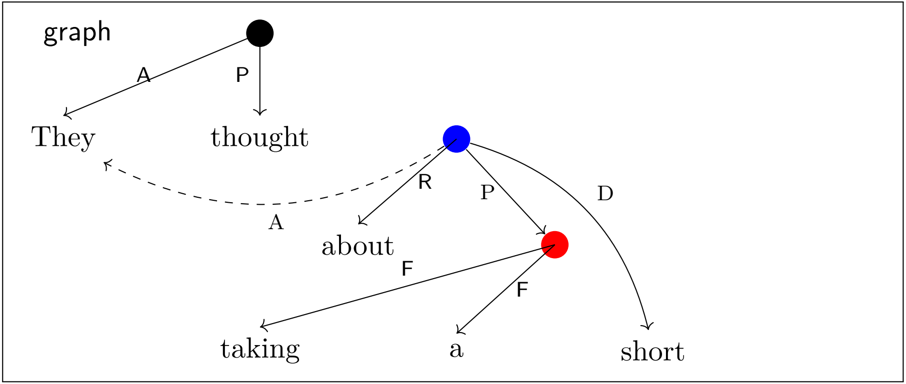
Example: TUPA Transition Sequence

⇒ $\text{R}{\scriptsize\text{IGHT}}\text{-E}{\scriptsize\text{DGE}}_D$



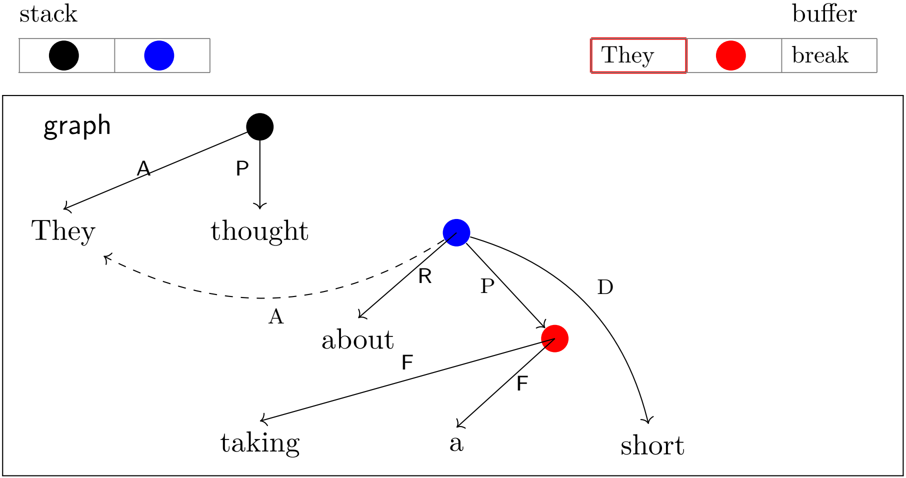
Example: TUPA Transition Sequence

⇒ Reduce



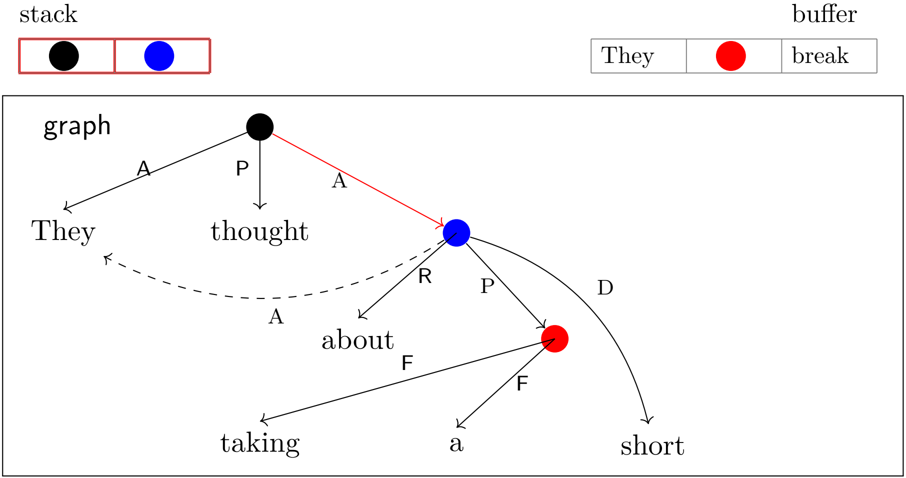
Example: TUPA Transition Sequence

⇒ SWAP



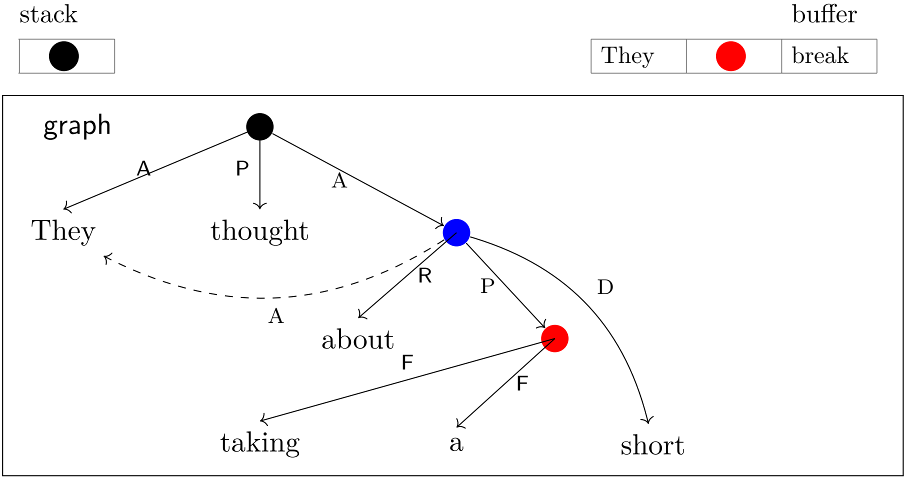
Example: TUPA Transition Sequence

$\Rightarrow$ Right-Edge$_A$



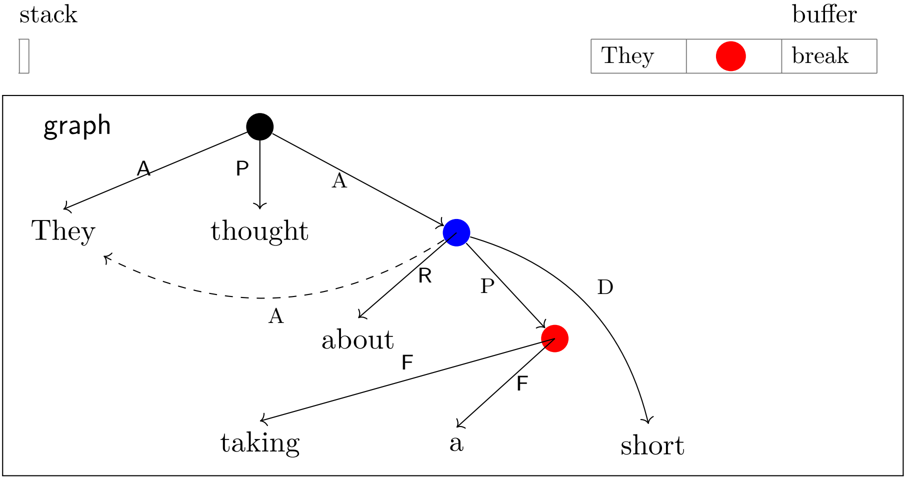
Example: TUPA Transition Sequence

⇒ Reduce



Example: TUPA Transition Sequence

⇒ REDUCE



Example: TUPA Transition Sequence

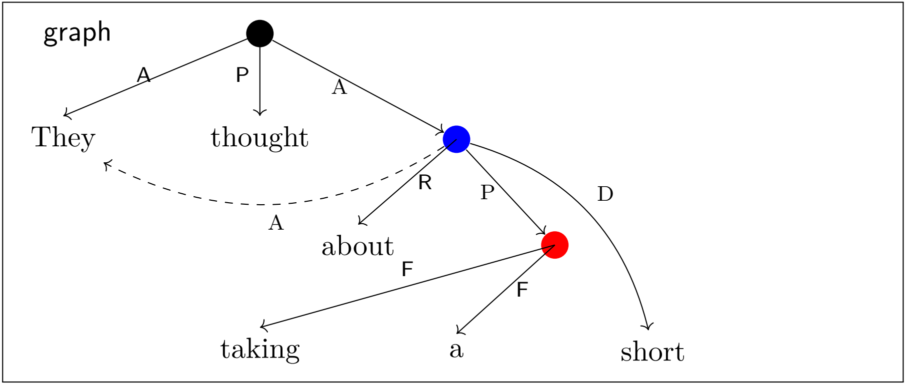
⇒ Shift

stack

They

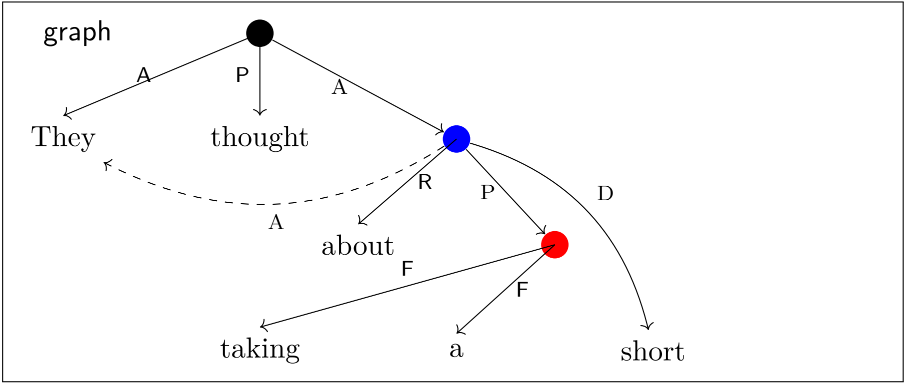
buffer

●	break



Example: TUPA Transition Sequence

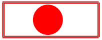
⇒ REDUCE



Example: TUPA Transition Sequence

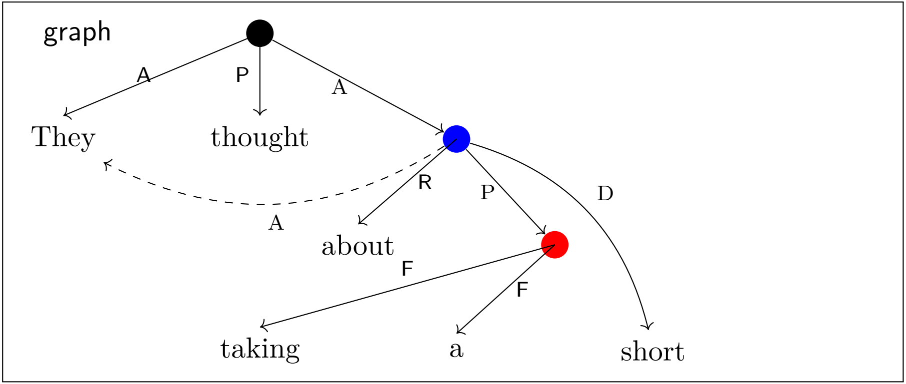
⇒ Shift

stack



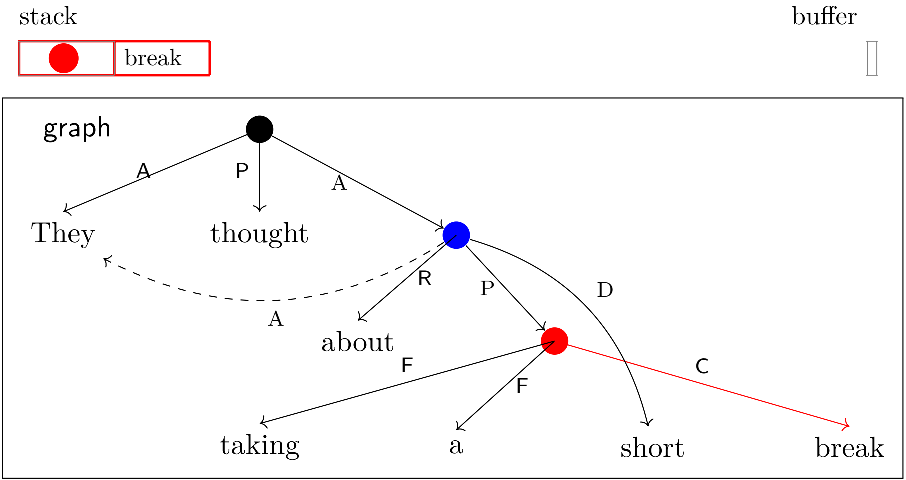
buffer

break



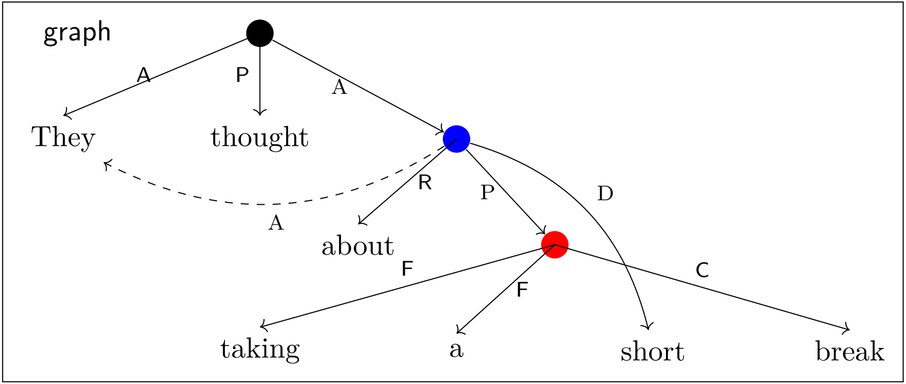
Example: TUPA Transition Sequence

⇒ RIGHT-EDGE$_C$



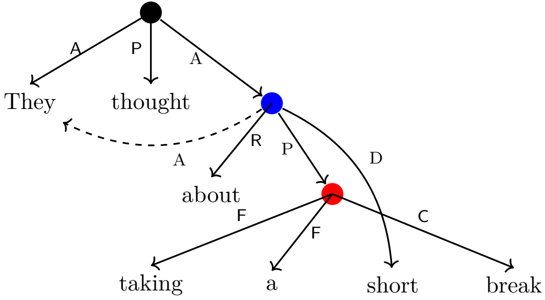
Example: TUPA Transition Sequence

⇒ FINISH



Training

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



$\Downarrow$

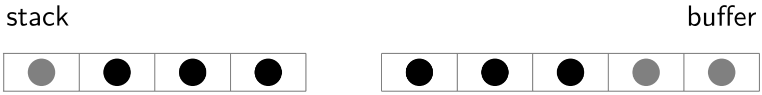
Shift, Right-Edge$_A$, Shift, Swap, Right-Edge$_P$, Reduce, Shift, Shift, Node$_R$, Reduce, Left-Remote$_A$, Shift, Shift, Node$_C$, Reduce, Shift, Right-Edge$_P$, Shift, Right-Edge$_F$, Reduce, Shift, Swap, Right-Edge$_D$, Reduce, Swap, Right-Edge$_A$, Reduce, Reduce, Shift, Reduce, Shift, Right-Edge$_C$,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

TUPA Model

Learns to greedily predict transition based on current state.
Experimenting with three classifiers:

Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + bidirectional RNN + MLP.

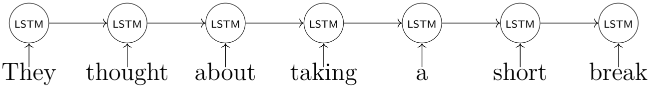
Features include:
{words, parts of speech, syntactic dependencies, existing edge labels}
from the stack and buffer + parents, children, grandchildren.



TUPA Model

Learns to greedily predict transition based on current state.
Experimenting with three classifiers:

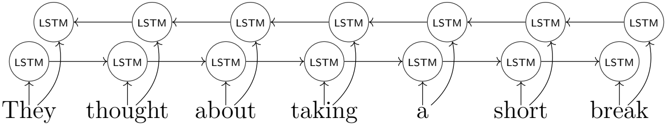
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.



TUPA Model

Learns to greedily predict transition based on current state.
Experimenting with three classifiers:

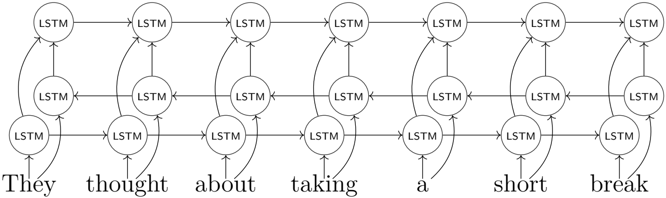
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.



TUPA Model

Learns to greedily predict transition based on current state.
Experimenting with three classifiers:

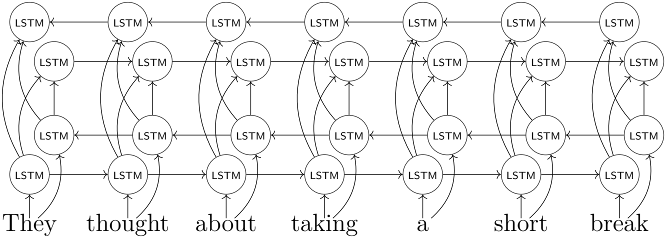
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.

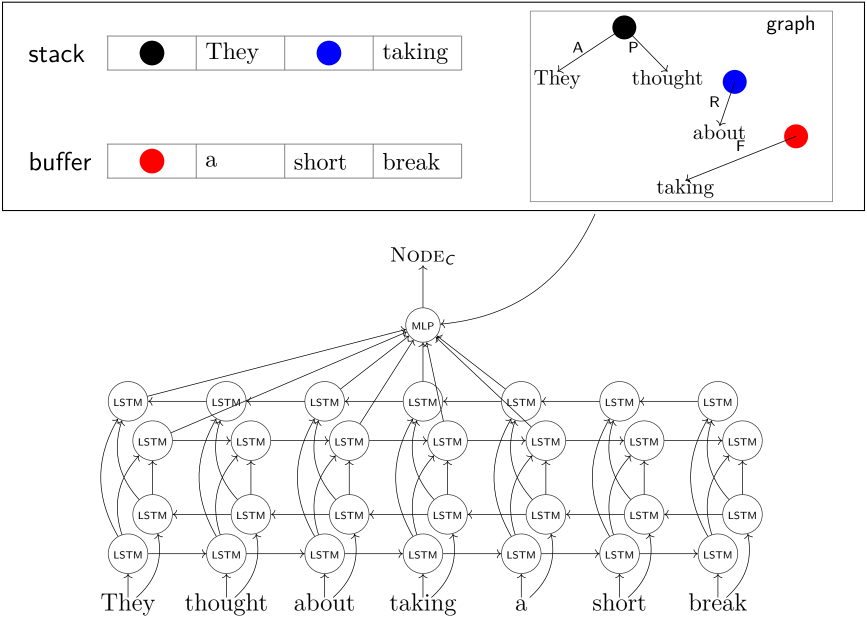


TUPA Model

Learns to greedily predict transition based on current state.
Experimenting with three classifiers:

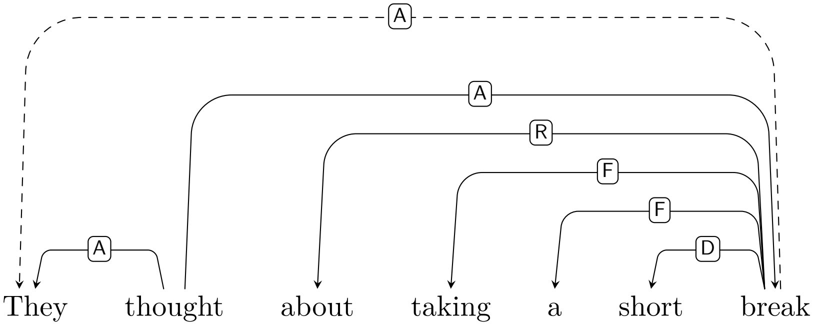
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.




stack
They
taking
buffer
a
short
break
graph
A
P
They
thought
R
about
F
taking
NODE$_C$
MLP
LSTM
They
thought
about
taking
a
short
break

Comparing to Existing Methods

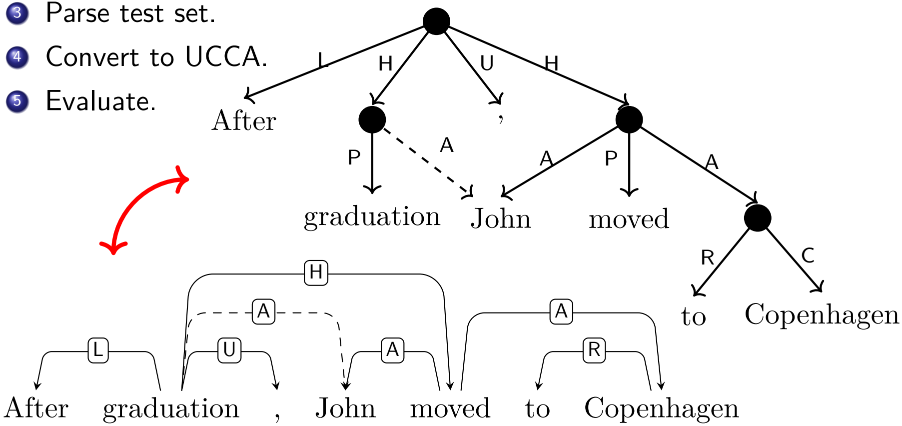
Using conversion-based approximation as baseline,
with bi-lexical DAG parsers and transition-based tree parsers.



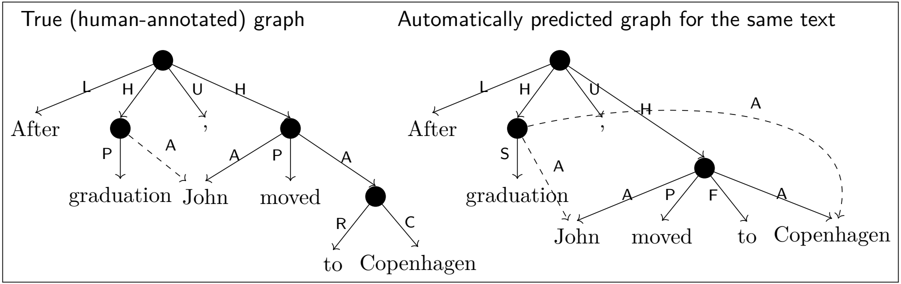
UCCA bi-lexical DAG approximation.

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.

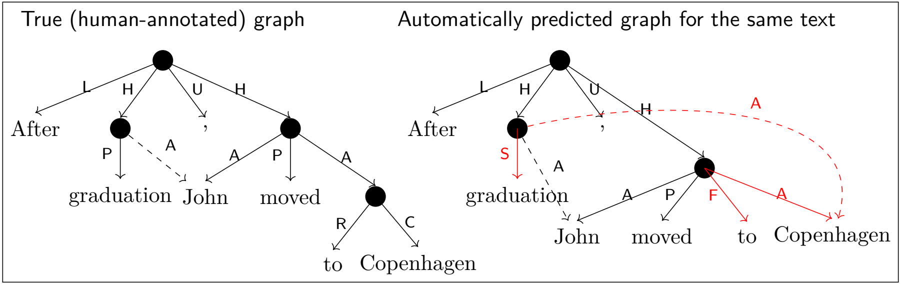


Evaluation



1. Match primary edges between the graphs by terminal yield and label.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

Evaluation



1. Match primary edges between the graphs by terminal yield and label.
2. Calculate **precision, recall and F1** scores.
3. Repeat for remote edges.

Primary

P	R	F1
$\frac{6}{9} = 67\%$	$\frac{6}{10} = 60\%$	64%

Remote

P	R	F1
$\frac{1}{2} = 50\%$	$\frac{1}{1} = 100\%$	67%

Results

TUPA$_{\text{BiLSTM}}$ outperforms all other methods on the English Wiki test set:

	English Wiki	
	Primary **F1**	Remote **F1**
TUPA		
Sparse	64.1	16
MLP	64.9	16.9
BiLSTM	**73.2**	**46.8**
Baselines		
DAGParser	58.6	1
TurboParser	51.2	3.7
MaltParser	60.2	
StackLSTM	69.9	
UPARSE	61.1	

Results

...and also on the **out-of-domain** English 20K:

	English Wiki		English 20K	
	Primary **F1**	Remote **F1**	Primary **F1**	Remote **F1**
TUPA				
Sparse	64.1	16	59.8	11.5
MLP	64.9	16.9	62.5	9.7
BiLSTM	**73.2**	**46.8**	**67.9**	**23.0**
Baselines				
DAGParser	58.6	1	53.4	
TurboParser	51.2	3.7	43.1	0.8
MaltParser	60.2		55.3	
StackLSTM	69.9		63.5	
UPARSE	61.1		52.8	

Results

	English Wiki		English 20K		French 20K		German 20K	
	Primary **F1**	Remote **F1**	Primary **F1**	Remote **F1**	Primary **F1**	Remote **F1**	Primary **F1**	Remote **F1**
TUPA								
Sparse	64.1	16	59.8	11.5				
MLP	64.9	16.9	62.5	9.7				
BiLSTM	**73.2**	**46.8**	**67.9**	**23.0**	44.0	3.8	73.9	47.2
Baselines								
DAGParser	58.6	1	53.4					
TurboParser	51.2	3.7	43.1	0.8				
MaltParser	60.2		55.3					
StackLSTM	69.9		63.5					
UPARSE	61.1		52.8					

Interim Summary

- Structured meaning representation benefits language understanding.
- UCCA's semantic distinctions require a graph structure including non-terminals, reentrancy and discontinuity.
- TUPA is an accurate transition-based UCCA parser, and the first to support UCCA and any DAG over the text tokens.
- Outperforms strong conversion-based baselines.

Interim Summary

- Structured meaning representation benefits language understanding.
- UCCA's semantic distinctions require a graph structure including non-terminals, reentrancy and discontinuity.
- TUPA is an accurate transition-based UCCA parser, and the first to support UCCA and any DAG over the text tokens.
- Outperforms strong conversion-based baselines.

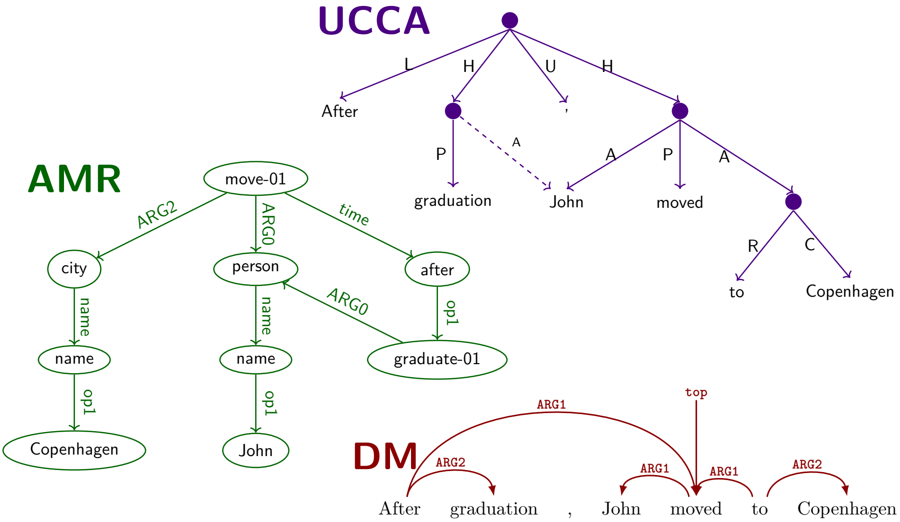
Up next:

- Parsing other semantic representations.
- Comparing representations through conversion.

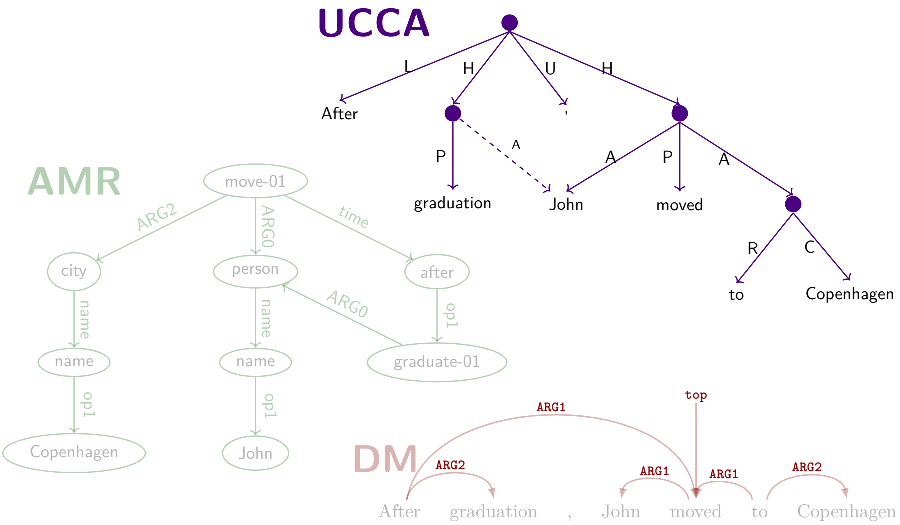
Outline

1. Background: The UCCA Semantic Representation Scheme
2. A Transition-Based DAG Parser for UCCA (ACL'17)
3. Multitask Parsing across Semantic Representations (ACL'18)
4. Content Differences between Syntactic and Semantic Representations (under submission)

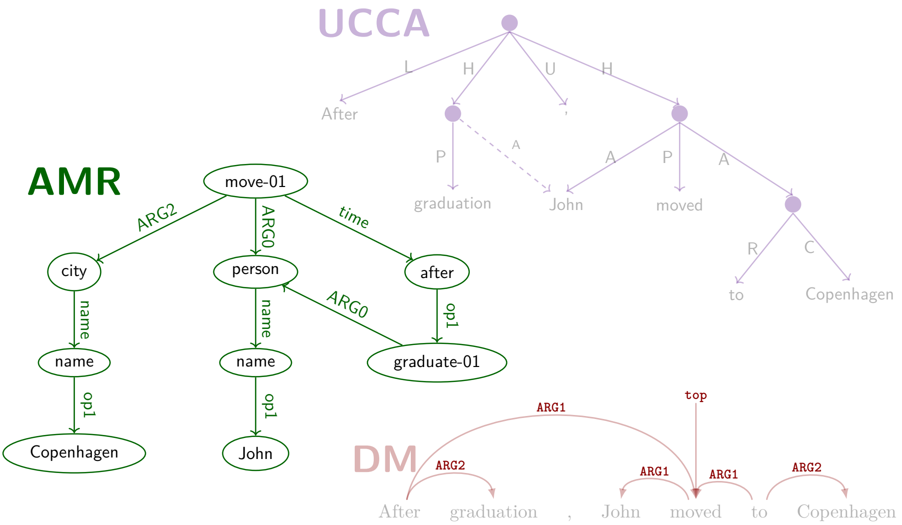
Semantic Representations



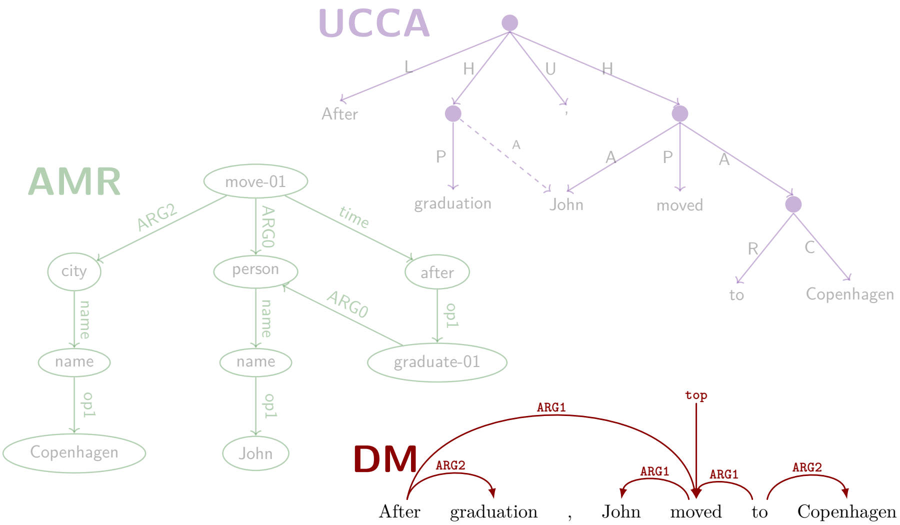
Semantic Representations



Semantic Representations

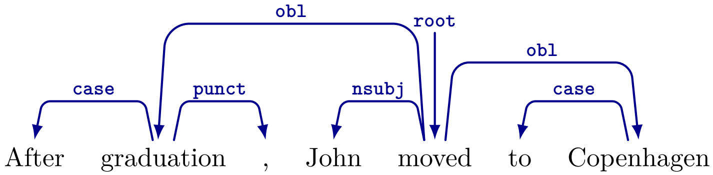


Semantic Representations



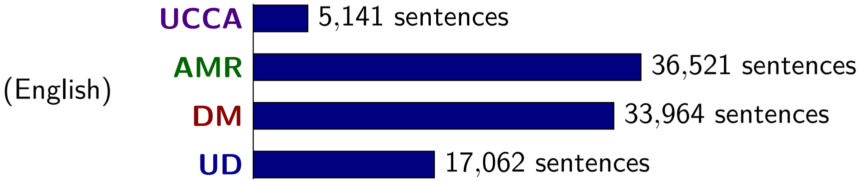
Syntactic Representations

UD (Universal Dependencies)



Data

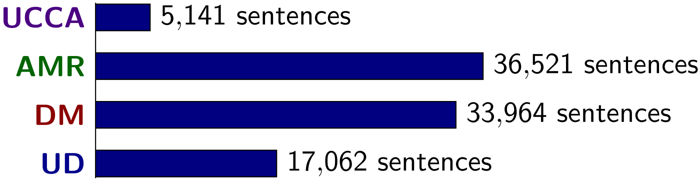
UCCA training data is scarce



Data

UCCA training data is scarce

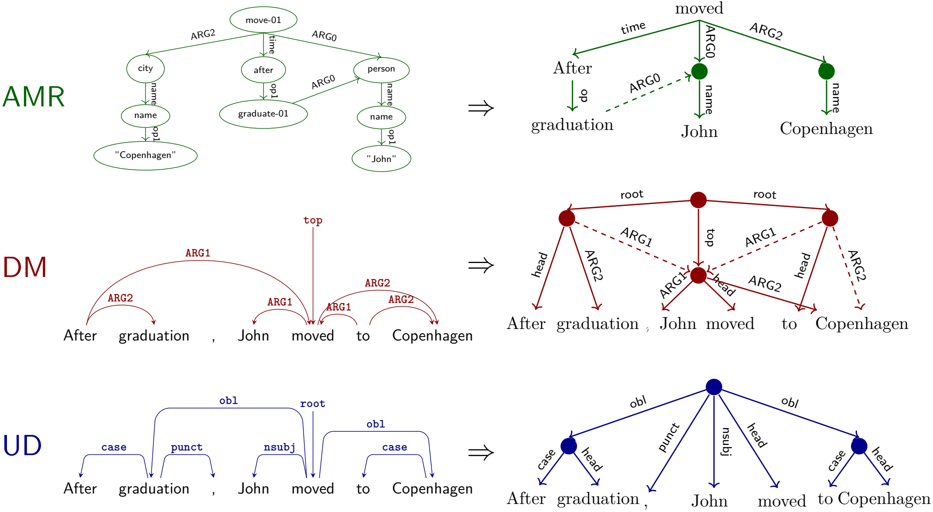
(English)



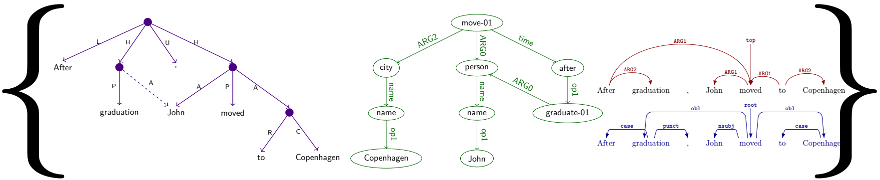
and domains are limited.

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

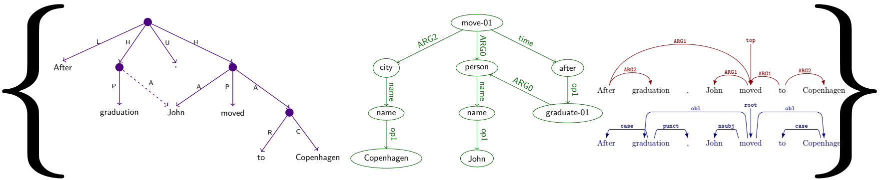
Conversion


AMR
move-01
ARG2
time
ARG0
city
after
person
name
op1
name
ARG0
name
graduate-01
name
op1
op1
"Copenhagen"
"John"
⇒
moved
time
ARG0
ARG2
After
op
ARG0
name
name
graduation
John
Copenhagen
DM
top
ARG1
ARG2
ARG1
ARG2
ARG1
ARG2
After graduation , John moved to Copenhagen
⇒
root
root
top
ARG1
ARG1
head
ARG2
ARG2
head
ARG2
ARG1
head
After graduation , John moved to Copenhagen
UD
obl
root
obl
case
punct
nsubj
case
After graduation , John moved to Copenhagen
⇒
obl
obl
punct
nsubj
head
case
head
case
head
After graduation , John moved to Copenhagen

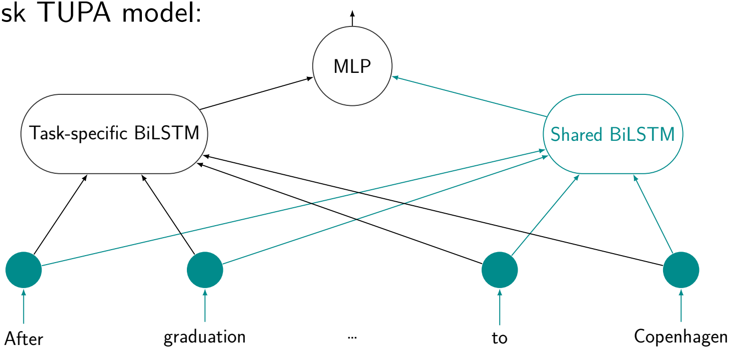
Multitask


L
H
U
H
After
P
A
,
A
P
A
graduation
John
moved
R
C
to
Copenhagen
move-01
ARG2
ARG0
time
city
person
after
name
name
ARG0
op1
name
name
graduate-01
op1
op1
Copenhagen
John
ARG1
top
ARG2
ARG1
ARG1
ARG2
After graduation , John moved to Copenhagen
obl
root
obl
case
punct
nsubj
case
After graduation , John moved to Copenhage

Multitask



Multitask TUPA model:



Results

	Primary F1	**Remote F1**
English Wiki (in-domain)		
Single-task	73.2	46.8
+AMR	72.7	52.7
+DM	**74.0**	**53.8**
+UD	72.2	48.0
+AMR+DM	73.6	48.5
+AMR+UD	73.3	51.2
+DM+UD	73.9	52.2
All	73.8	52.1

Results

	Primary F1	Remote F1
English Wiki (in-domain)		
Single-task	73.2	46.8
+AMR	72.7	52.7
+DM	**74.0**	**53.8**
+UD	72.2	48.0
+AMR+DM	73.6	48.5
+AMR+UD	73.3	51.2
+DM+UD	73.9	52.2
All	73.8	52.1

	Primary F1	Remote F1
English 20K (out-of-domain)		
Single-task	67.9	23.0
+AMR	67.0	31.2
+DM	**69.1**	27.5
+UD	67.4	23.9
+AMR+DM	68.9	25.4
+AMR+UD	68.2	**31.4**
+DM+UD	68.6	29.1
All	**69.1**	25.8

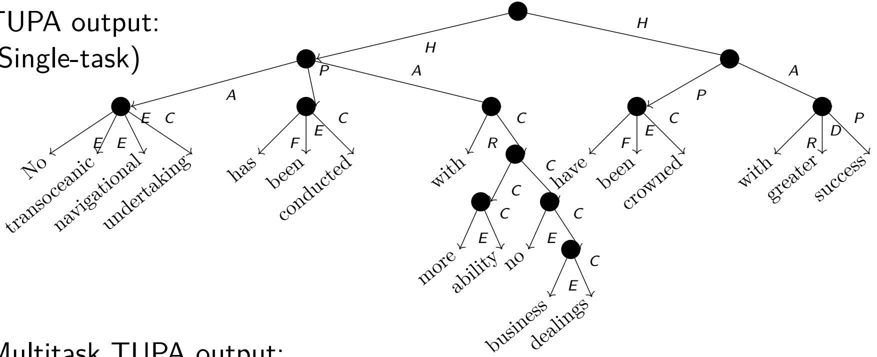
Results

	Primary F1	Remote F1
English Wiki (in-domain)		
Single-task	73.2	46.8
+AMR	72.7	52.7
+DM	**74.0**	**53.8**
+UD	72.2	48.0
+AMR+DM	73.6	48.5
+AMR+UD	73.3	51.2
+DM+UD	73.9	52.2
All	73.8	52.1

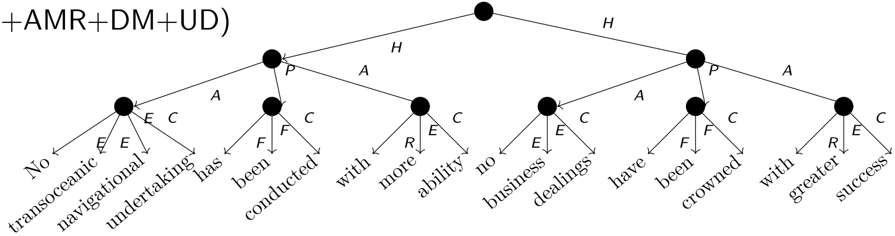
	Primary F1	Remote F1
English 20K (out-of-domain)		
Single-task	67.9	23.0
+AMR	67.0	31.2
+DM	**69.1**	27.5
+UD	67.4	23.9
+AMR+DM	68.9	25.4
+AMR+UD	68.2	**31.4**
+DM+UD	68.6	29.1
All	**69.1**	25.8

	Primary F1	Remote F1
French 20K (in-domain)		
Single-task	44.0	3.8
+UD	**49.6**	1.6
German 20K (in-domain)		
Single-task	73.9	47.2
+UD	**80.1**	**59.8**

TUPA output:
(Single-task)

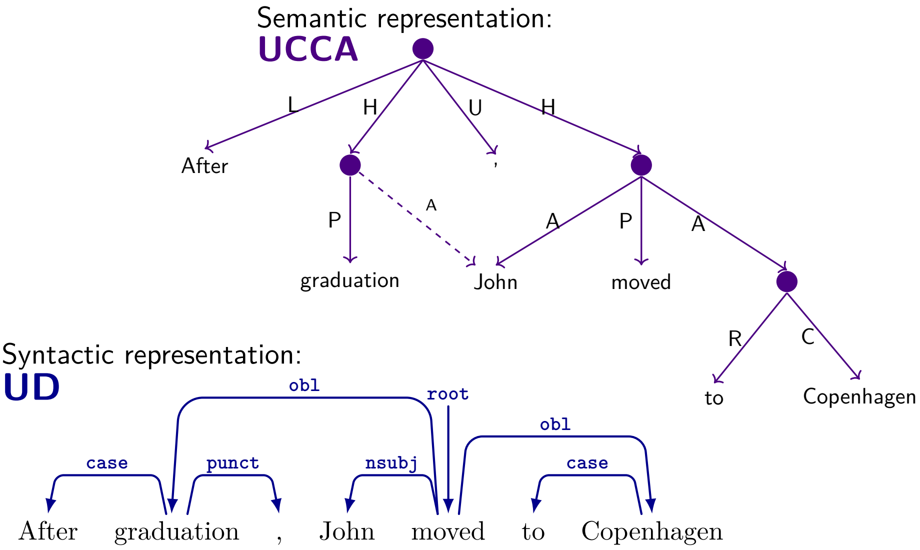


Multitask TUPA output:
(+AMR+DM+UD)



Outline

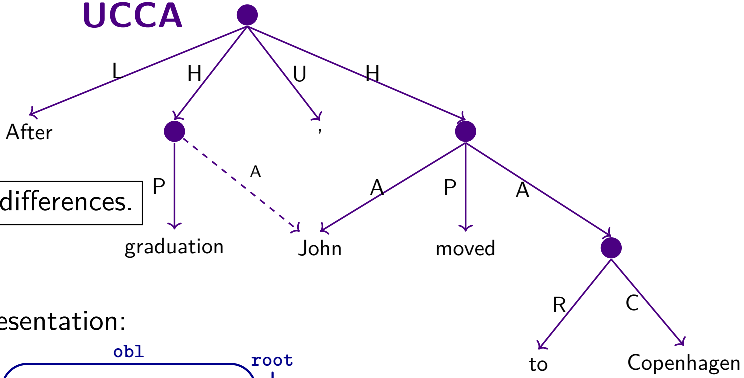
UCCA vs. UD


Semantic representation:
UCCA
L
H
U
H
After
,
P
A
A
P
A
graduation
John
moved
R
C
to
Copenhagen
Syntactic representation:
UD
obl
root
obl
case
punct
nsubj
case
After
graduation
,
John
moved
to
Copenhagen

UCCA vs. UD

Semantic representation:

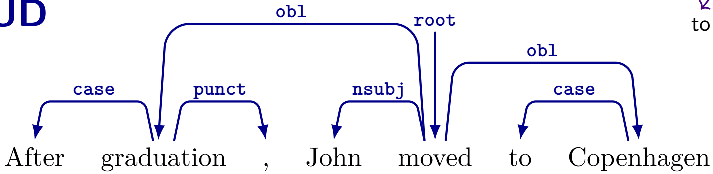
UCCA



Many formal differences.

Syntactic representation:

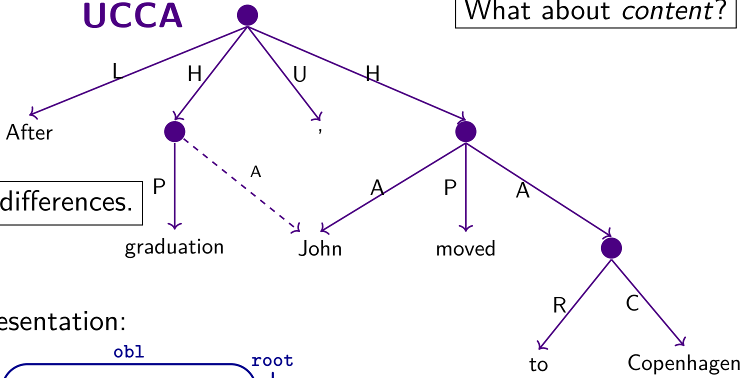
UD



UCCA vs. UD

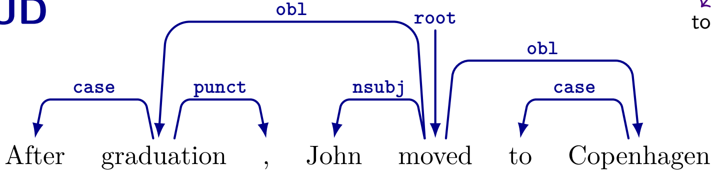
Semantic representation:
UCCA

What about *content*?



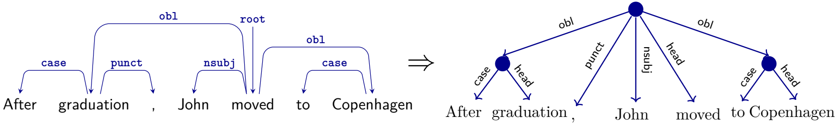
Many formal differences.

Syntactic representation:
UD



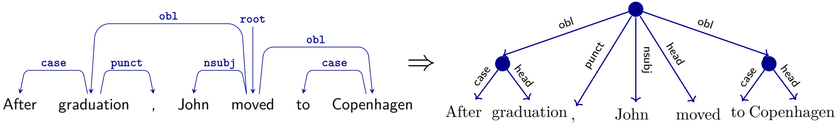
Assimilating the Graph Structures

UD



Assimilating the Graph Structures

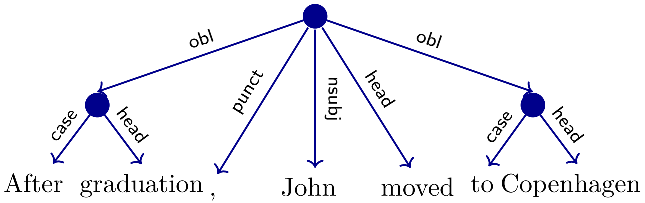
UD



Now we can evaluate by matching edges (UCCA unlabeled evaluation)

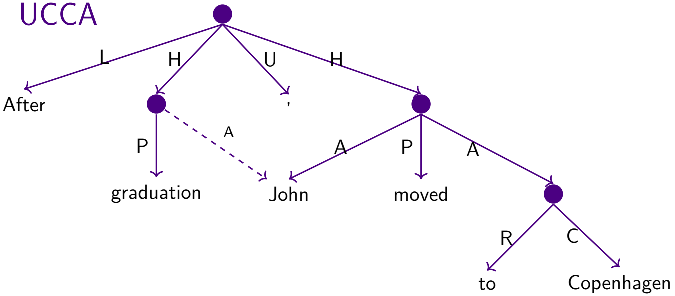
Assimilating the Graph Structures

UD



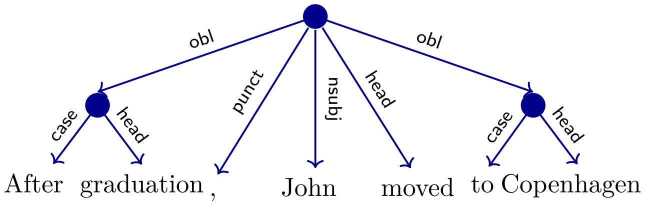
Now we can evaluate by matching edges (UCCA unlabeled evaluation)

UCCA



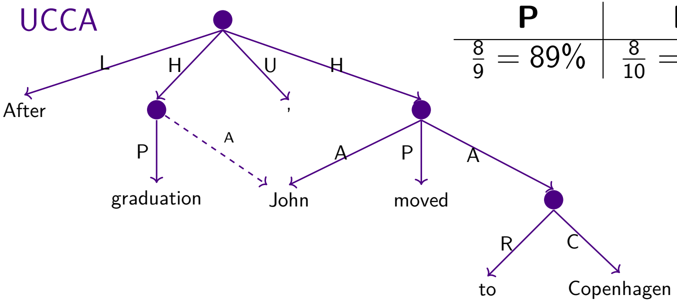
Assimilating the Graph Structures

UD



Now we can evaluate by matching edges (UCCA unlabeled evaluation)

UCCA

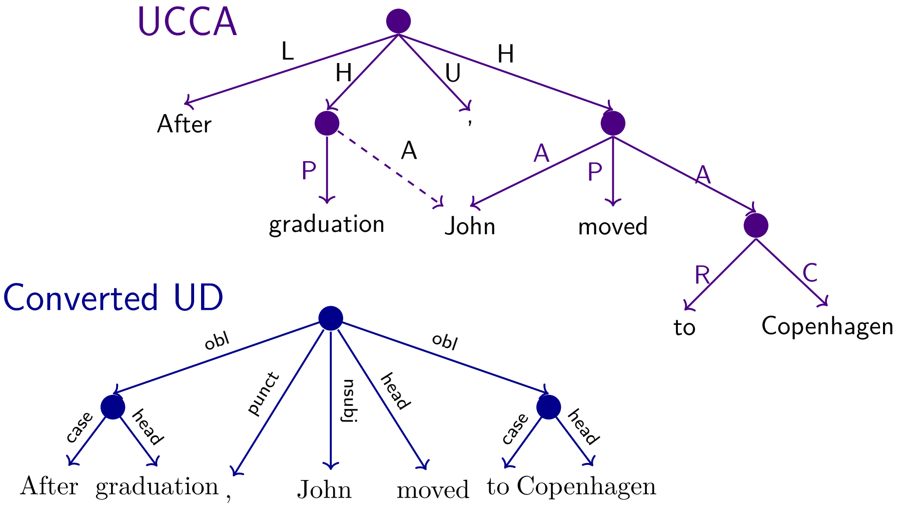


P	**R**	**F1**
$\frac{8}{9} = 89\%$	$\frac{8}{10} = 80\%$	84%

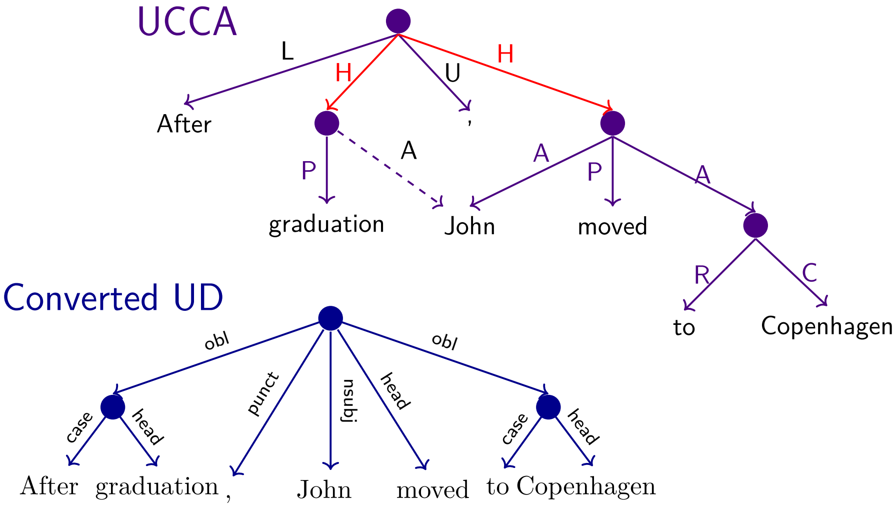
Confusion Matrix

	A	C	D	E	F	G	H	L	N	P	Q	R	S	T	∅
acl	8			101	2		15						1		49
advcl	2	2					103	1		4					97
advmod	61	9	399	51	12	33	3	61		2	10	6	5	117	71
amod	1	33	99	197	2		7			3	27		97	2	60
appos	1	10		8			5				1		4		10
aux			96		285										2
case	1	5	2	14	34			48	6	1	1	489	50		75
cc			1					305	71		1	1			11
ccomp	78						8						1		41
compound	23	24	8	176	2					1	1	1	3	3	164
conj	2	88	1				265			2			3		90
cop					333					3		1	24		3
csubj	2														8
dep												1			
det	2	1	19	763	1	1					19	2			26
discourse		1			1	6	13	3					1		1
expl					22										2
iobj	19														
list		2					8								2
mark		2	3		186	1		134	1			53	1	1	18
nmod	100	1	1	233			6						3	4	110
nsubj	993				14		2	9		3		24	1		37
nummod	4	7		6			3				50				24
obj	439	7	5	1	1		1	1		8	1	6		4	92
obl	247	1	21	7	2	4	4	4			3	2		69	132
orphan	1														1
parataxis	1			1		2	79			1			2		39
vocative	9					3									
xcomp	44	1	2	2			1			5			7		116
head	125	1402	152	37	91	18	652	2	1	961	18	9	353	1	524
∅	329	172	34	56	6	5	466	29		141	27	7	98	11	

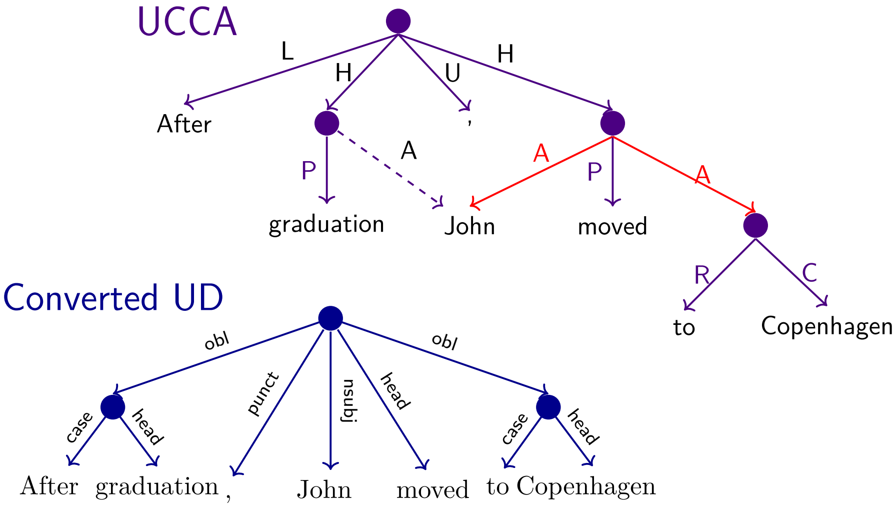
Scenes and non-Scenes, Relations and Participants



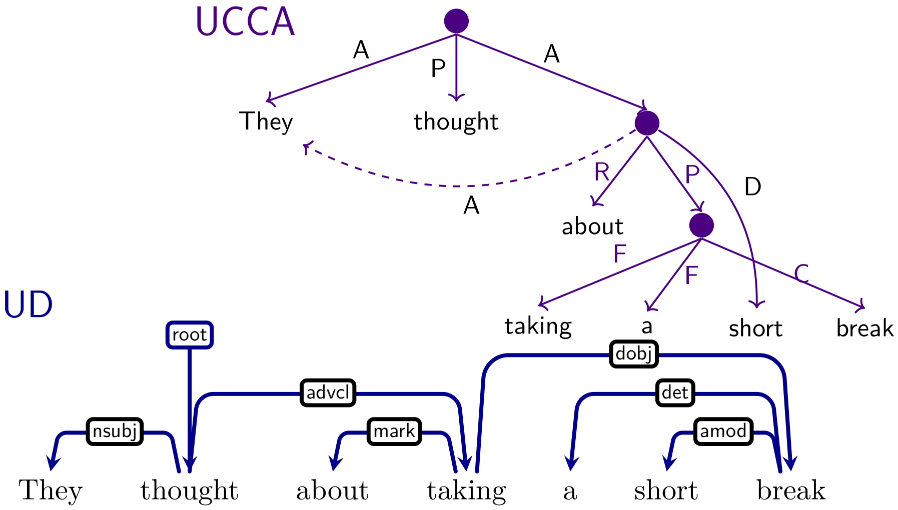
Scenes and non-Scenes, Relations and Participants



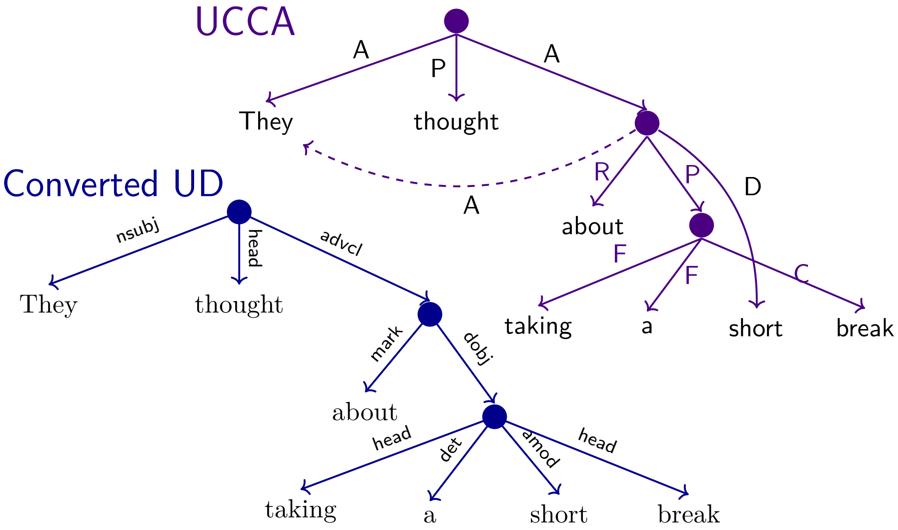
Scenes and non-Scenes, Relations and Participants



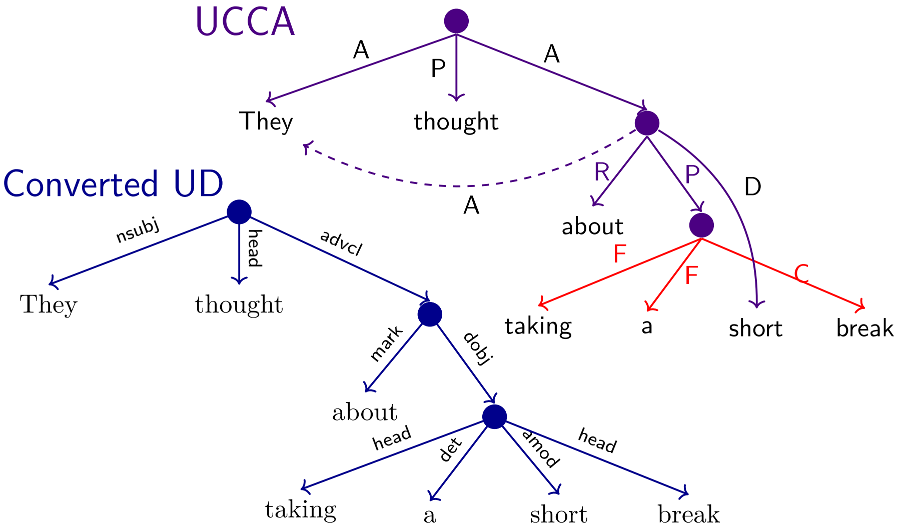
Multi-word Expressions



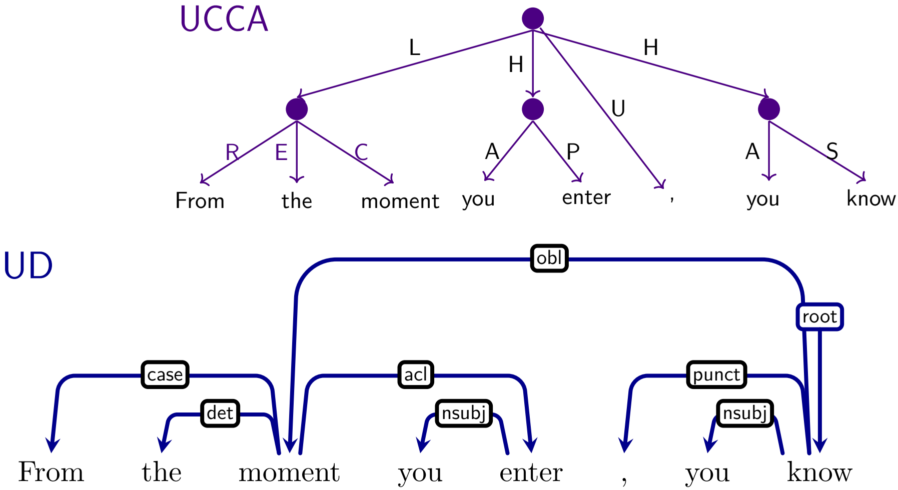
Multi-word Expressions



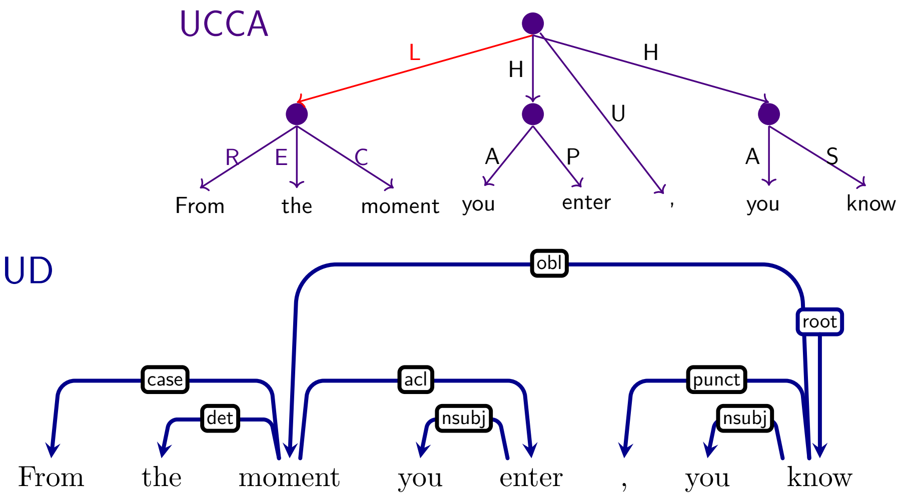
Multi-word Expressions



Linkage between Scenes



Linkage between Scenes



Conclusion

- Meaning representation is valuable for language understanding.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.
- Multitask learning allows useful shared generalizations to emerge.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.
- Multitask learning allows useful shared generalizations to emerge.
- Divergences limit inter-scheme gain, but highlight relative strengths.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.
- Multitask learning allows useful shared generalizations to emerge.
- Divergences limit inter-scheme gain, but highlight relative strengths.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.
- Multitask learning allows useful shared generalizations to emerge.
- Divergences limit inter-scheme gain, but highlight relative strengths.

Ongoing work:

- Complement syntax with *lexical* semantics to make up for differences.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.
- Multitask learning allows useful shared generalizations to emerge.
- Divergences limit inter-scheme gain, but highlight relative strengths.

Ongoing work:

- Complement syntax with *lexical* semantics to make up for differences.
- Establish cross-framework meaning representation parsing as a task.

Conclusion

- Meaning representation is valuable for language understanding.
- TUPA, an accurate UCCA parser, is suited to many representations.
- Multitask learning allows useful shared generalizations to emerge.
- Divergences limit inter-scheme gain, but highlight relative strengths.

Ongoing work:

- Complement syntax with *lexical* semantics to make up for differences.
- Establish cross-framework meaning representation parsing as a task.

Long term goal: learning semantic parsing as a means to learn language.

References I

Abend, O. and Rappoport, A. (2013).
Universal Conceptual Cognitive Annotation (UCCA).
In *Proc. of ACL*, pages 228–238.

Birch, A., Abend, O., Bojar, O., and Haddow, B. (2016).
HUME: Human UCCA-based evaluation of machine translation.
In *Proc. of EMNLP*, pages 1264–1274.

Choshen, L. and Abend, O. (2018).
Reference-less measure of faithfulness for grammatical error correction.
In *Proc. of NAACL-HLT*.

Sulem, E., Abend, O., and Rappoport, A. (2015).
Conceptual annotations preserve structure across translations: A French-English case study.
In *Proc. of S2MT*, pages 11–22.

Sulem, E., Abend, O., and Rappoport, A. (2018a).
Semantic structural annotation for text simplification.
In *Proc. of NAACL*.

Sulem, E., Abend, O., and Rappoport, A. (2018b).
Simple and effective text simplification using semantic and neural methods.
In *Proc. of ACL*.