Universal Semantic Parsing with Neural Networks

Daniel Hershcovich
Joint work with Ari Rappoport and Omri Abend

June 11, 2019

Natural Language Processing: What's It Good For?

Natural Language Processing: What's It Good For?

Machine translation:

$$\boxed{\text{After graduation, John moved to Copenhagen}}$$
$$\downarrow$$

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.

After graduation, John moved to Copenhagen

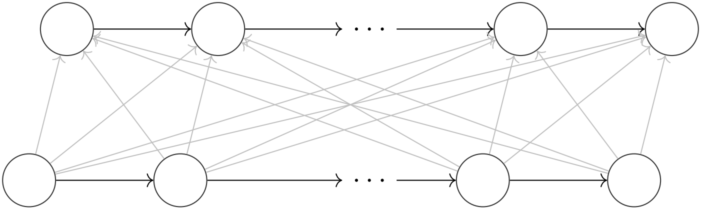
↓

ג'ון עבר לקופנהגן אחרי שסיים את הלימודים

↓

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

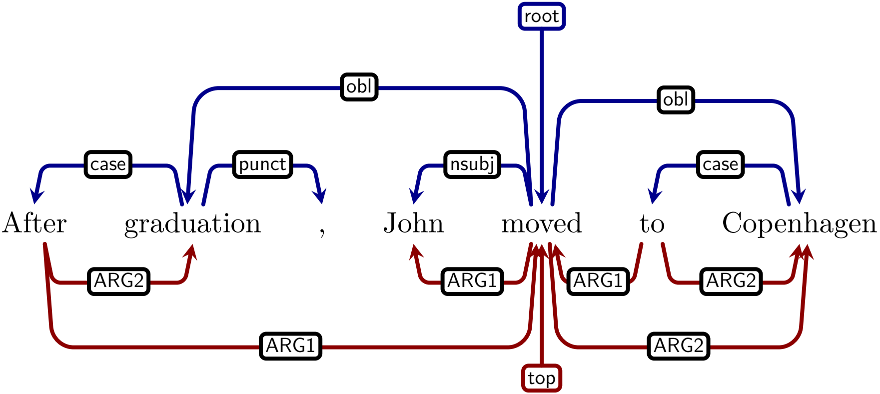
Sequence-to-sequence sometimes works, but lacks inductive bias.



Linguistic Structured Representations

Model explicit relations between words or concepts.

Example: syntactic/semantic bi-lexical dependencies.



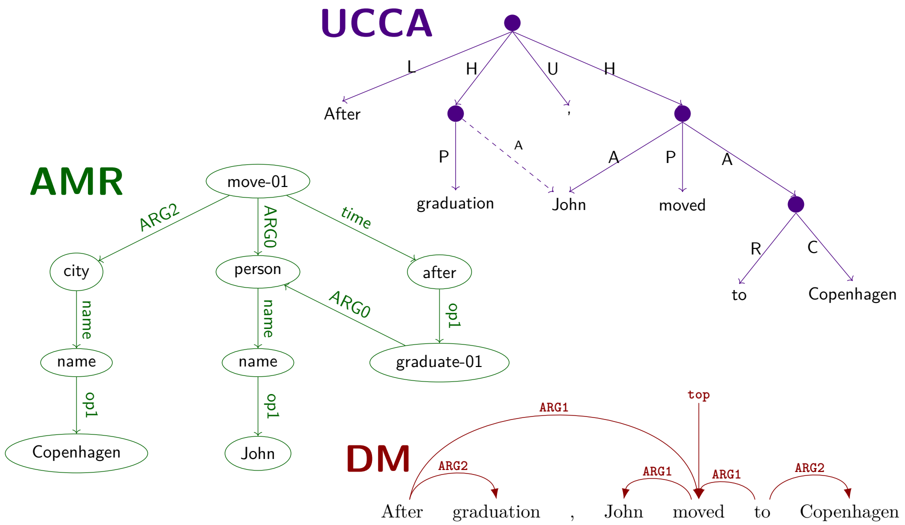
Semantic Representations

Abstract away from detail that does not affect meaning:

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

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

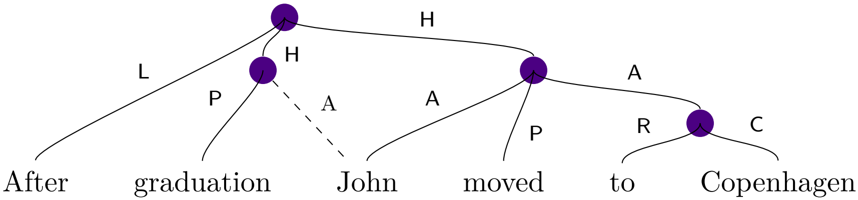
Semantic Representations



Outline

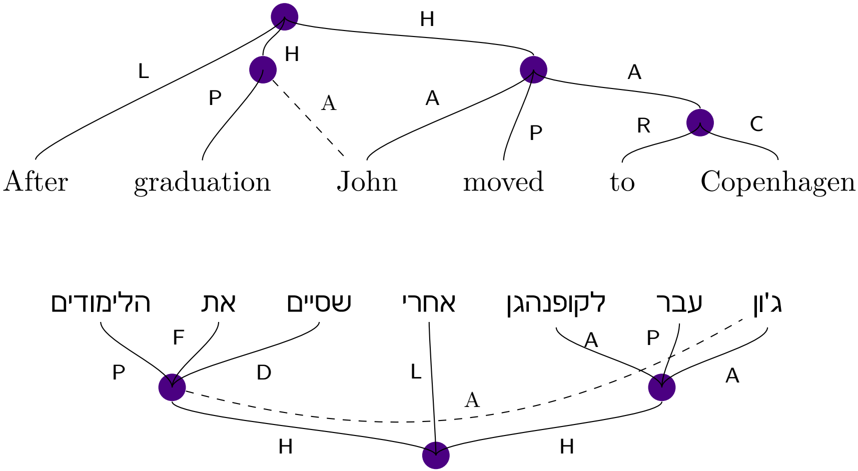
Universal Conceptual Cognitive Annotation (UCCA)

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



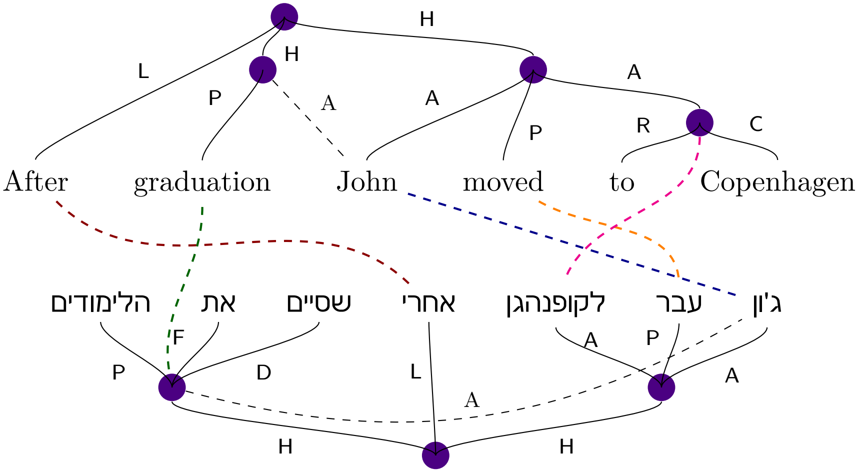
Universal Conceptual Cognitive Annotation (UCCA)

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



Universal Conceptual Cognitive Annotation (UCCA)

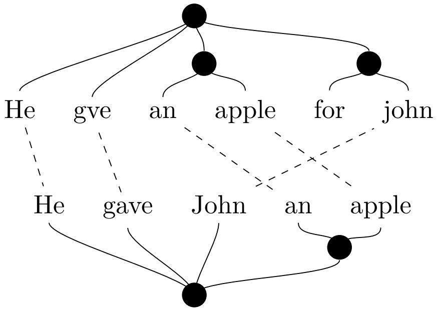
Supports rapid and intuitive annotation of linguistic semantic phenomena.
Cross-linguistically applicable and stable [Sulem et al., 2015].



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].

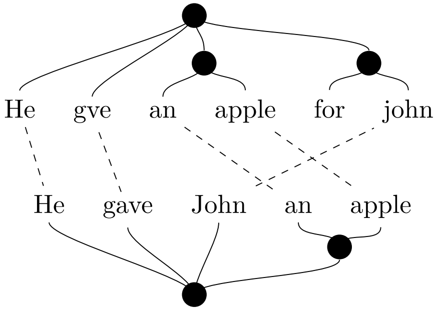


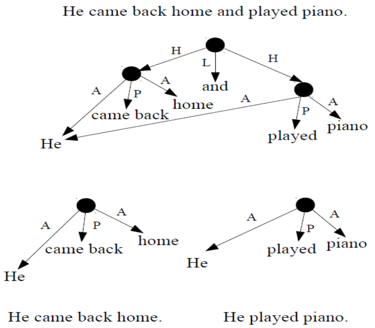
UCCA Applications

Semantics-based **evaluation** of

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

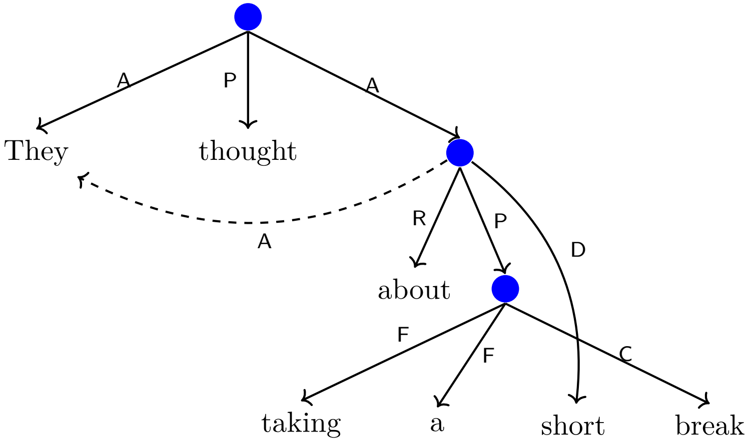
Sentence splitting for text simplification [Sulem et al., 2018b].





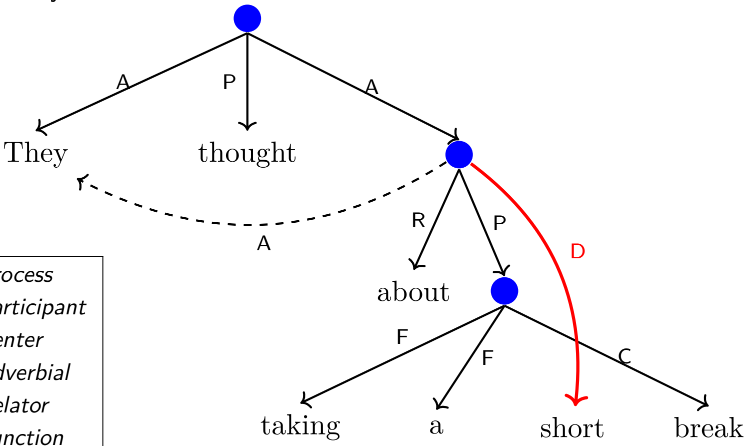
Graph Structure

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



Graph Structure

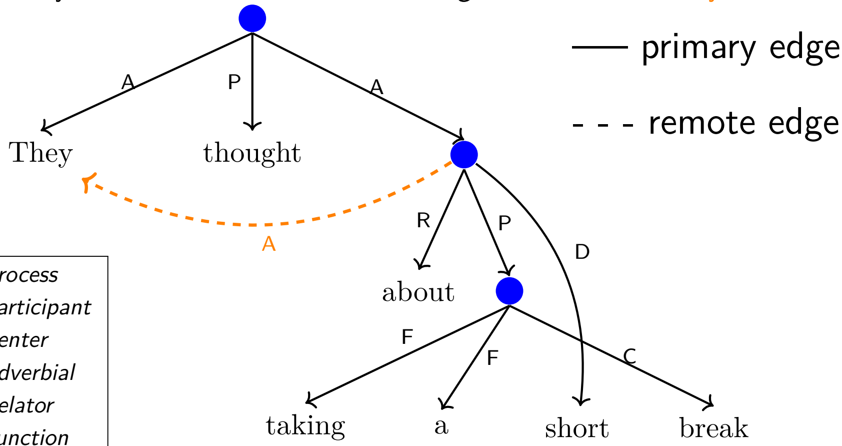
UCCA structures are directed acyclic graphs (DAGs) with labeled edges.
Text tokens are terminals, complex units are non-terminal nodes.
Phrases may be discontinuous.



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

Graph Structure

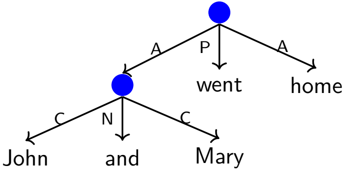
UCCA structures are directed acyclic graphs (DAGs) with labeled edges.
Text tokens are terminals, complex units are non-terminal nodes.
Phrases may be discontinuous. *Remote edges* enable reentrancy.



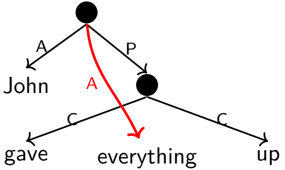
P	*Process*
A	*Participant*
C	*Center*
D	*Adverbial*
R	*Relator*
F	*Function*

Structural Properties

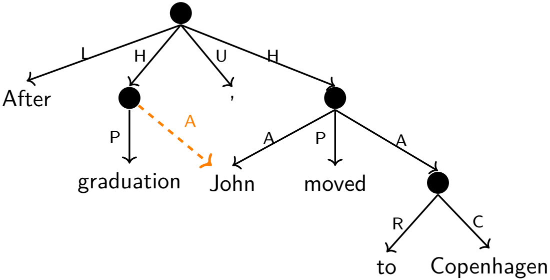
(1) non-terminal nodes



(2) discontinuity

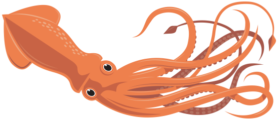


(3) reentrancy



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

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

stack	
●	They

					buffer
thought	about	taking	a	short	break

graph

Example: TUPA Transition Sequence

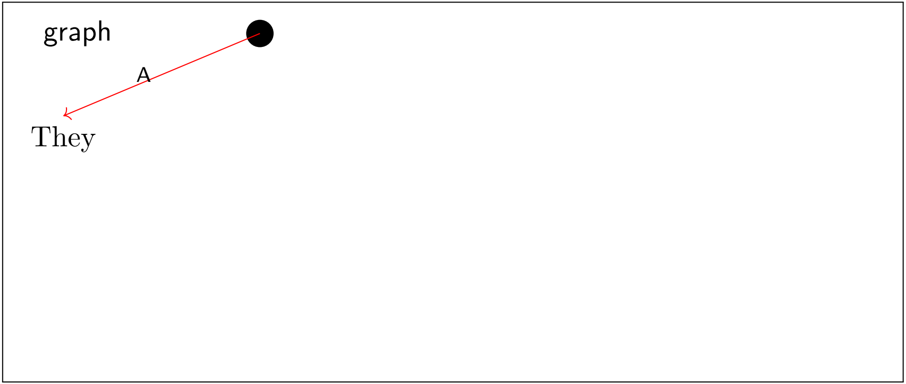
$\Rightarrow$ RIGHT-EDGE$_A$

stack

●	They

buffer

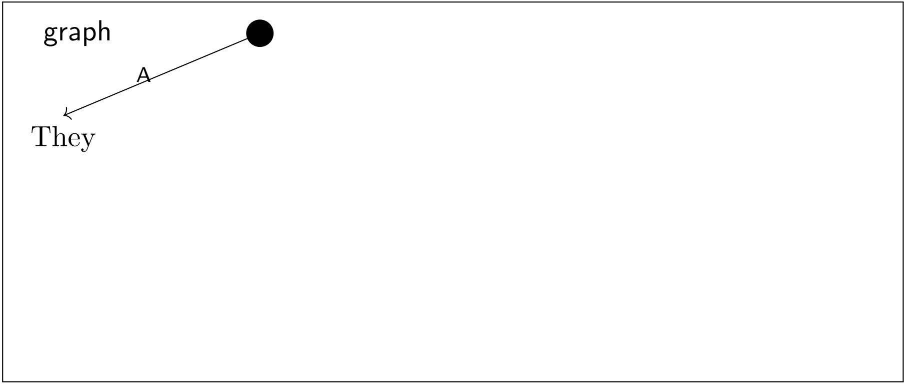
thought	about	taking	a	short	break



Example: TUPA Transition Sequence

⇒ SHIFT

stack				buffer				
●	They	thought		about	taking	a	short	break



Example: TUPA Transition Sequence

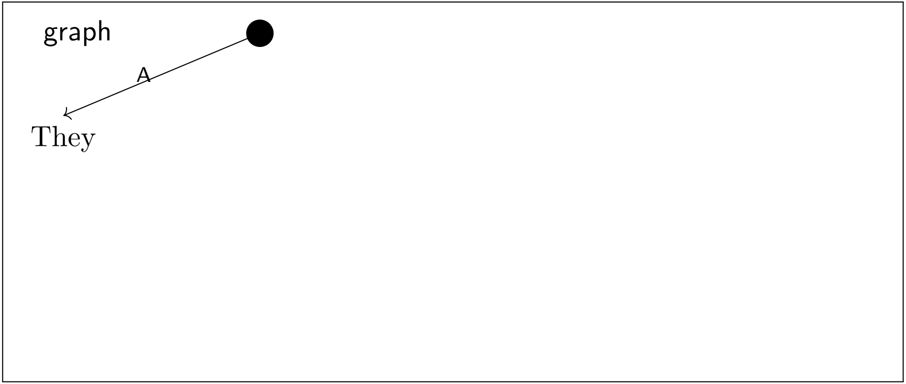
⇒ Swap

stack

●	thought

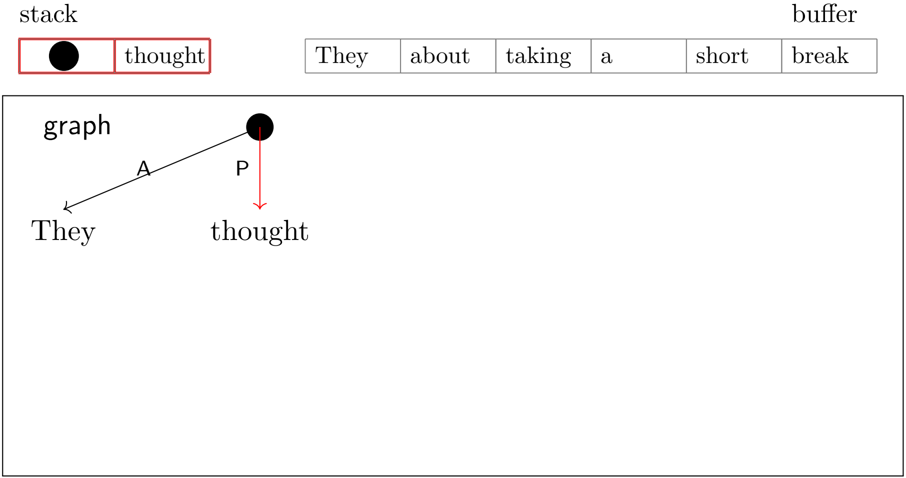
buffer

They	about	taking	a	short	break



Example: TUPA Transition Sequence

$\Rightarrow$ Right-Edge$_P$



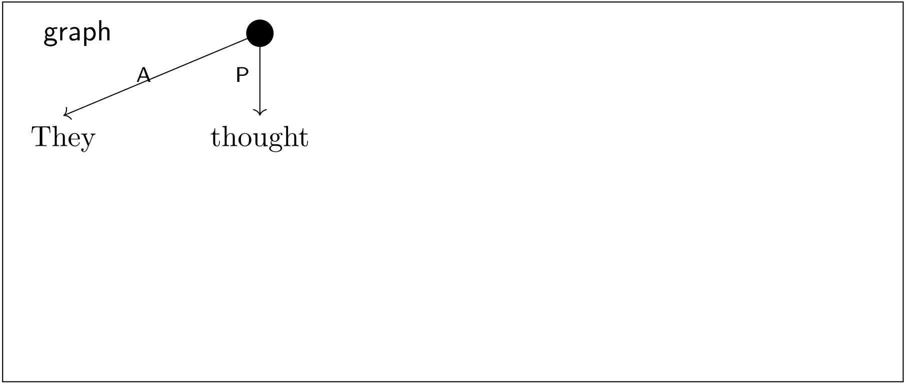
Example: TUPA Transition Sequence

$\Rightarrow$ REDUCE

stack

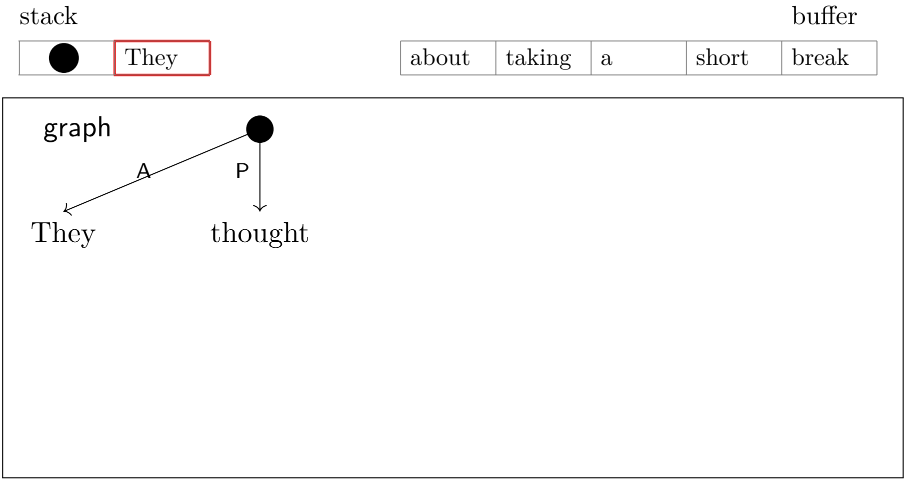
buffer

They	about	taking	a	short	break



Example: TUPA Transition Sequence

⇒ SHIFT

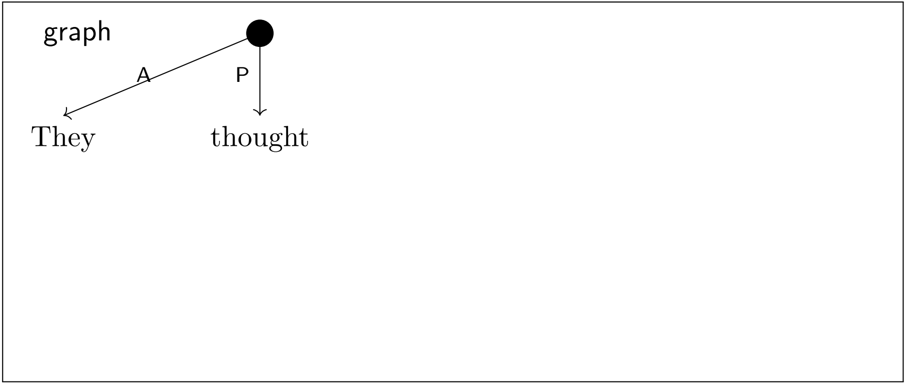


Example: TUPA Transition Sequence

⇒ SHIFT

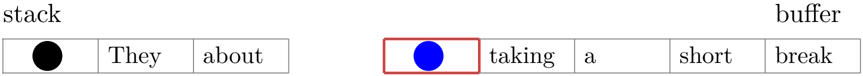
stack		
●	They	about

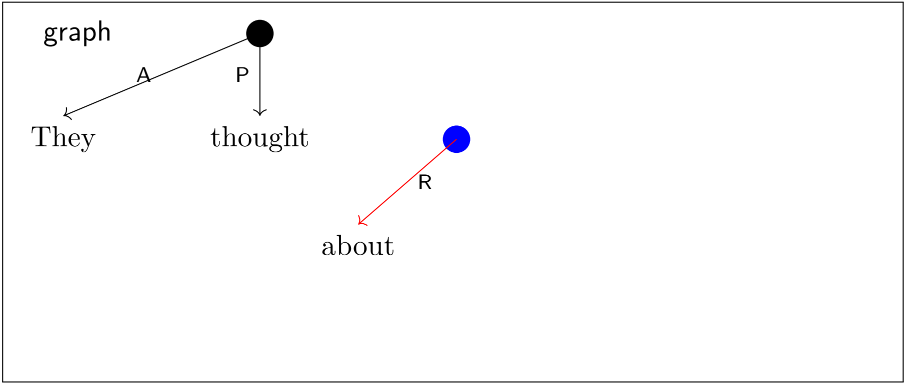
			buffer
taking	a	short	break



Example: TUPA Transition Sequence

$\Rightarrow$ $\textsc{Node}_R$





Example: TUPA Transition Sequence

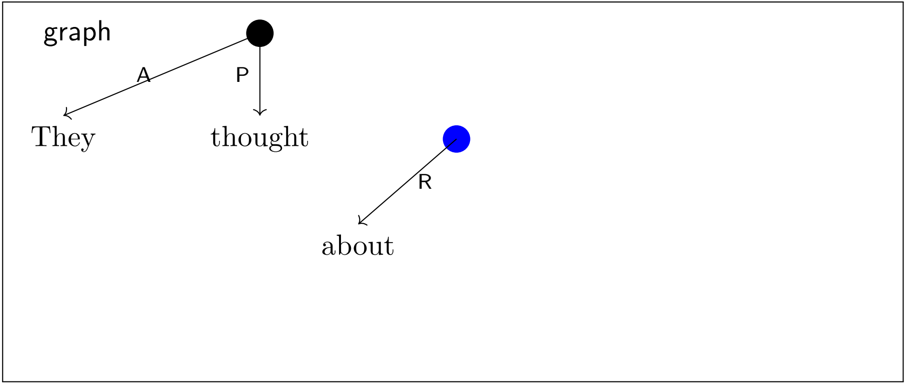
⇒ REDUCE

stack

●	They

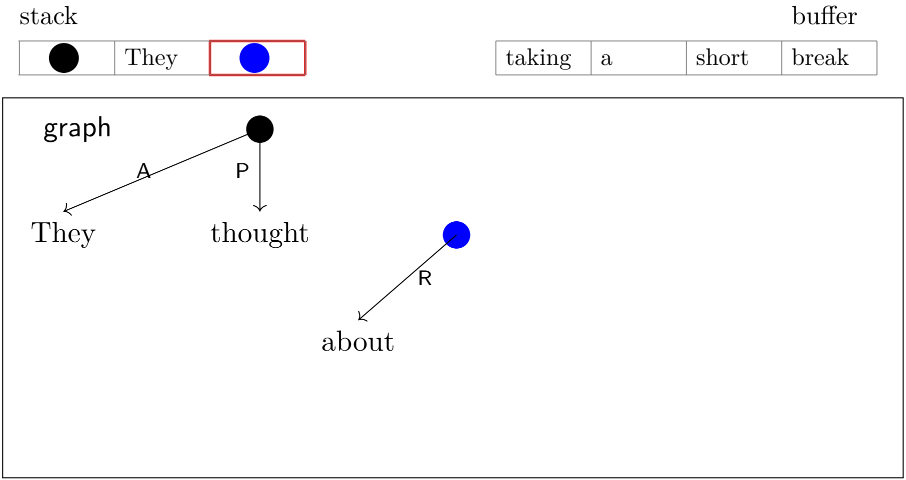
buffer

●	taking	a	short	break



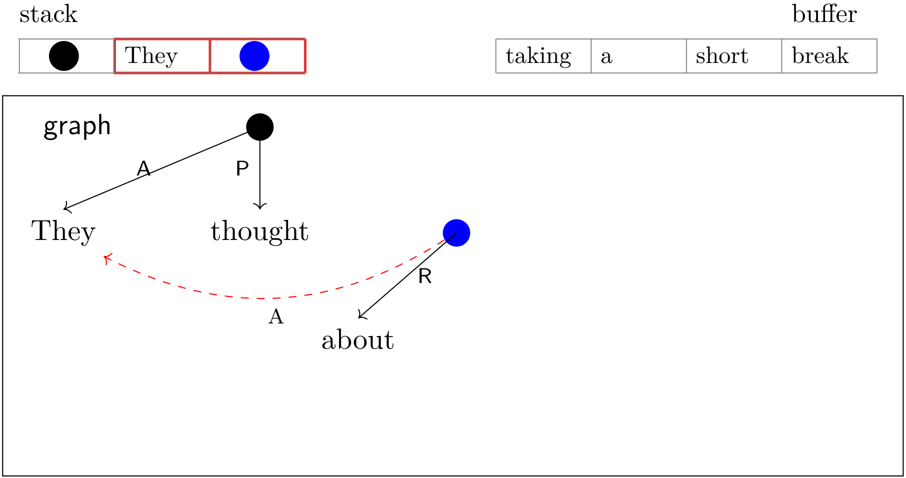
Example: TUPA Transition Sequence

⇒ SHIFT



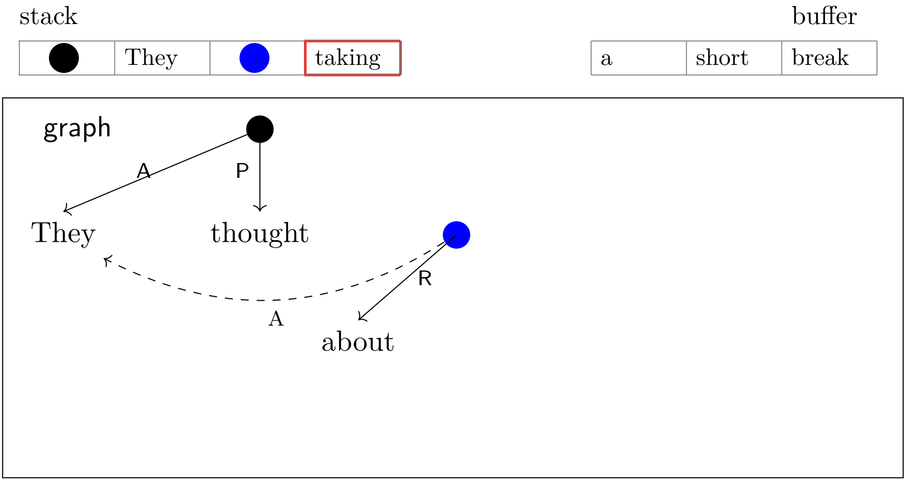
Example: TUPA Transition Sequence

⇒ LEFT-REMOTE$_A$



Example: TUPA Transition Sequence

⇒ SHIFT



Example: TUPA Transition Sequence

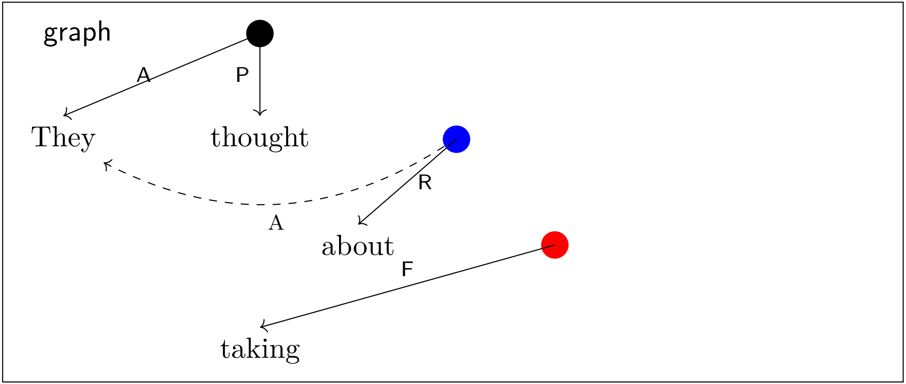
$\Rightarrow$ NODE$_C$

stack

	They		taking

buffer

	a	short	break



Example: TUPA Transition Sequence

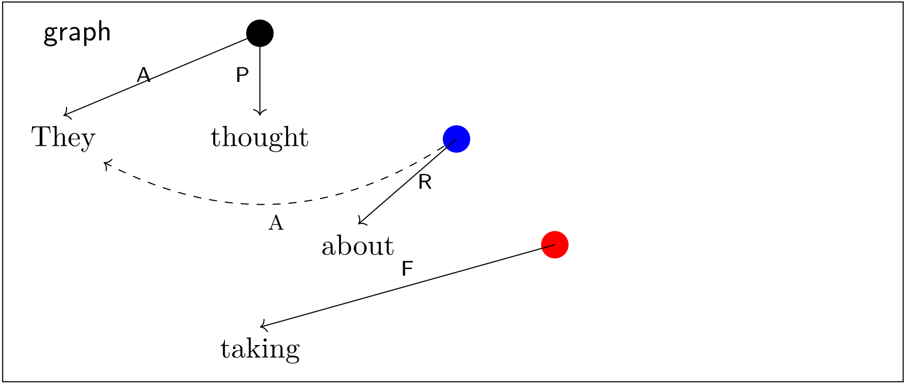
⇒ REDUCE

stack

●	They	●

buffer

●	a	short	break



Example: TUPA Transition Sequence

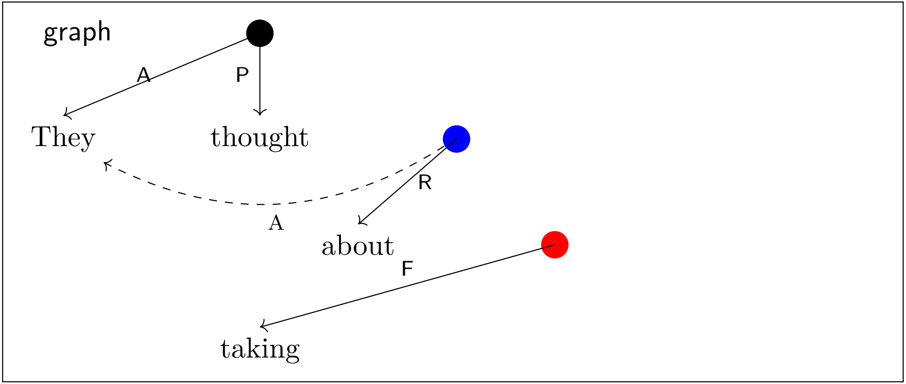
⇒ SHIFT

stack

●	They	●	●

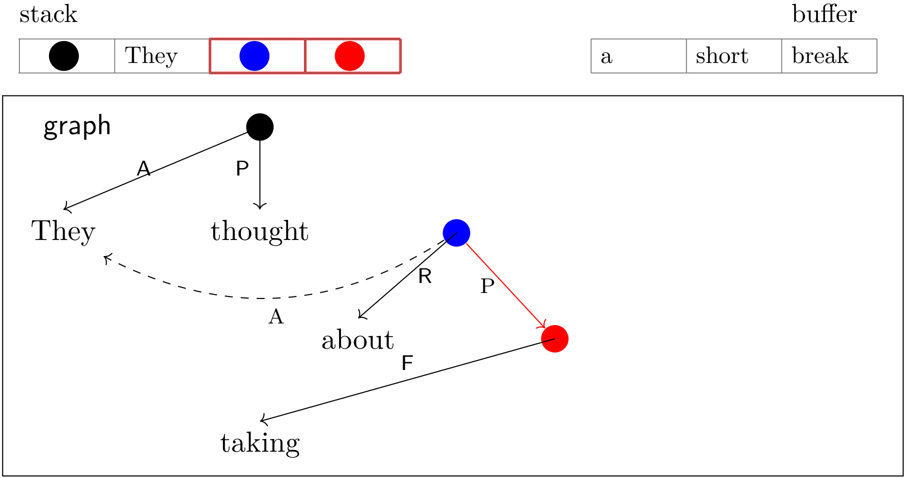
buffer

a	short	break



Example: TUPA Transition Sequence

⇒ Right-Edge$_P$



Example: TUPA Transition Sequence

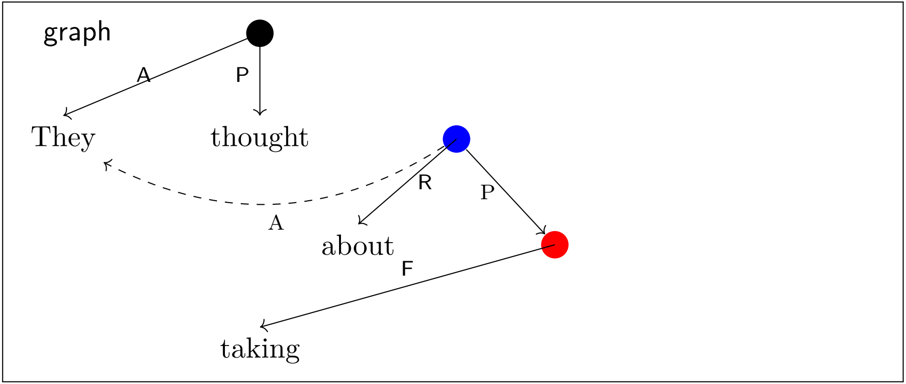
⇒ SHIFT

stack

●	They	●	●	a

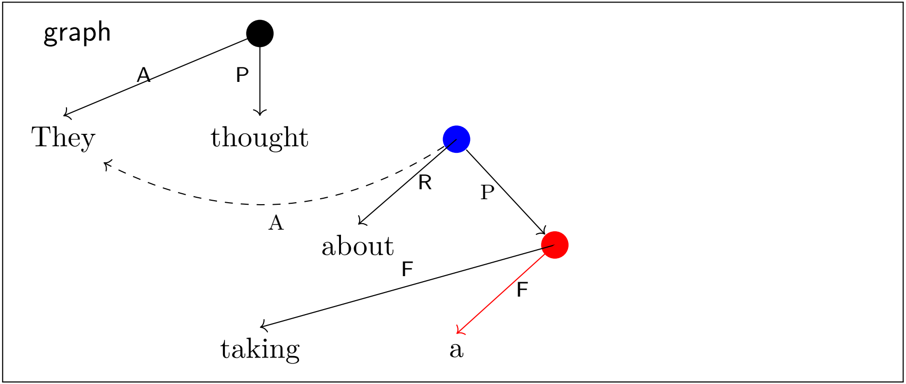
buffer

short	break



Example: TUPA Transition Sequence

$\Rightarrow$ RIGHT-EDGE$_F$

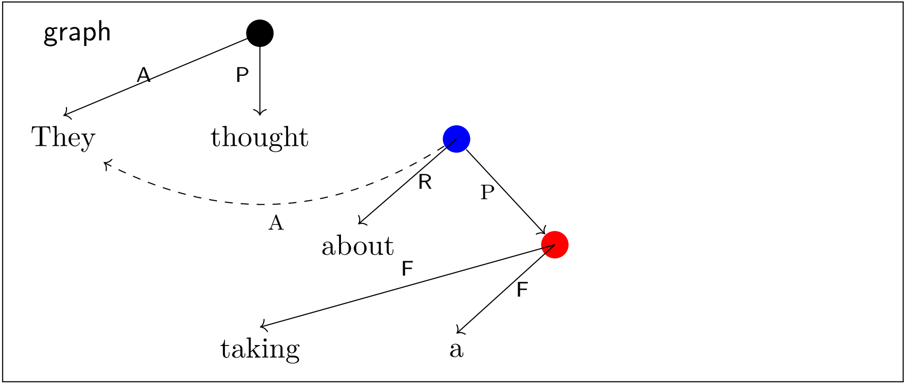


Example: TUPA Transition Sequence

⇒ REDUCE

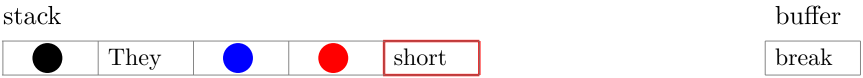
stack

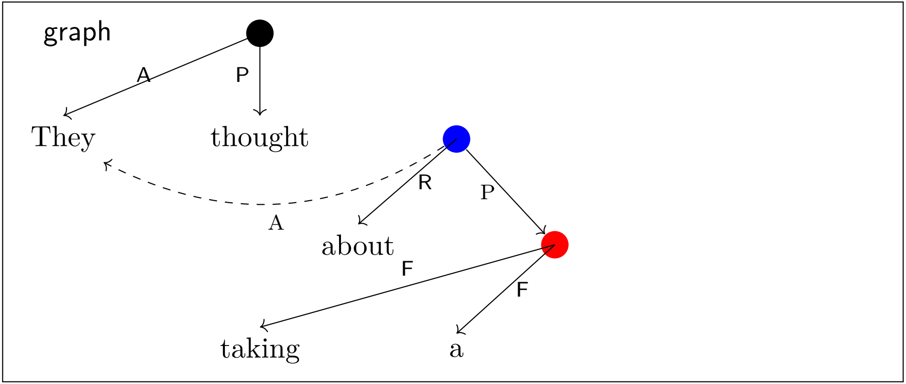
buffer



Example: TUPA Transition Sequence

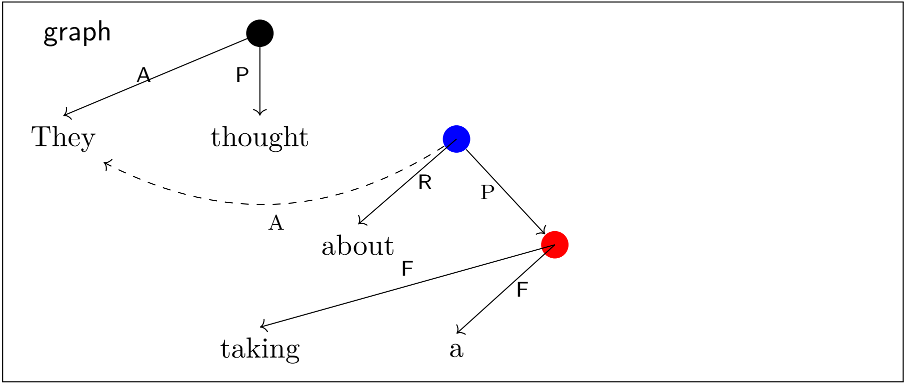
⇒ SHIFT





Example: TUPA Transition Sequence

⇒ SWAP



Example: TUPA Transition Sequence

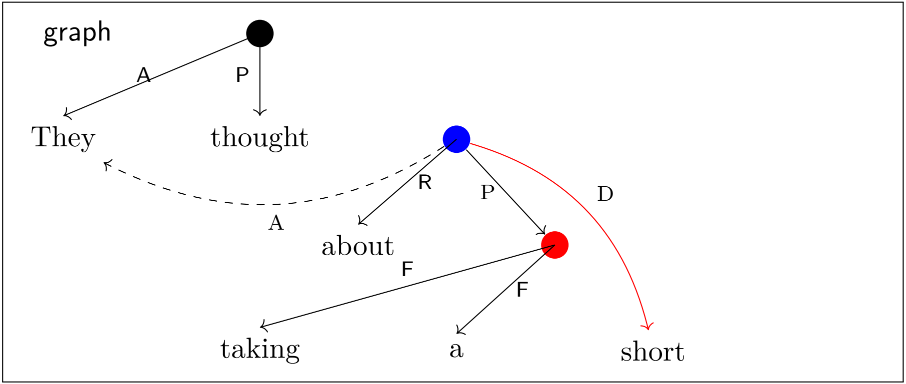
$\Rightarrow$ RIGHT-EDGE$_D$

stack

●	They	●	short

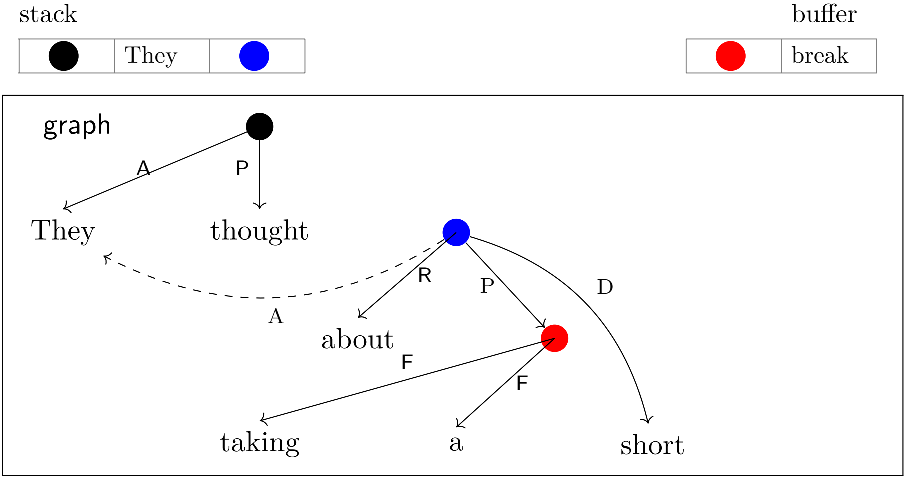
buffer

●	break



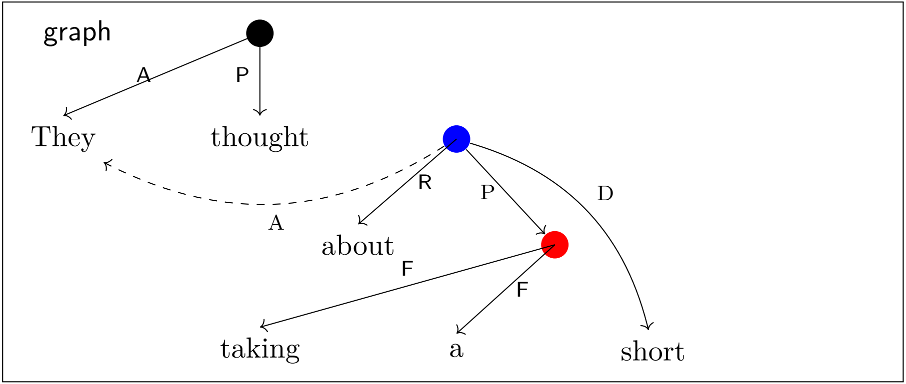
Example: TUPA Transition Sequence

⇒ Reduce



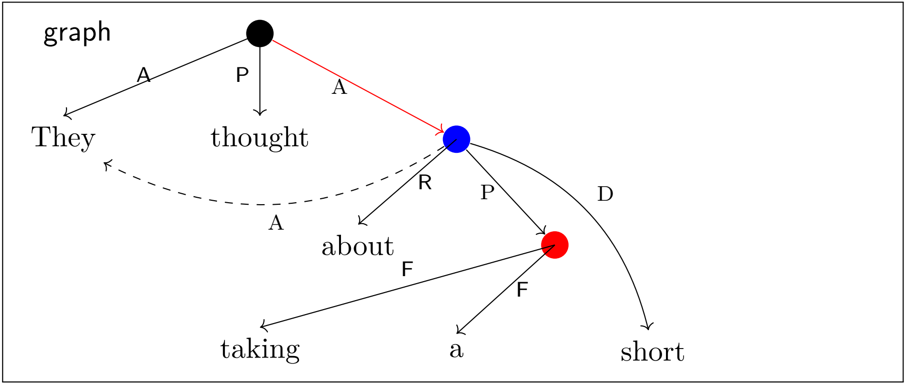
Example: TUPA Transition Sequence

⇒ SWAP



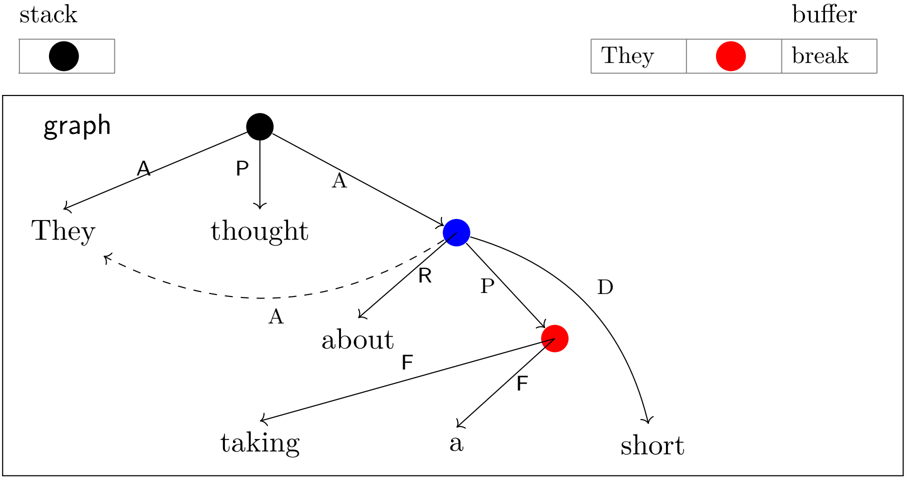
Example: TUPA Transition Sequence

⇒ Right-Edge$_A$



Example: TUPA Transition Sequence

⇒ REDUCE



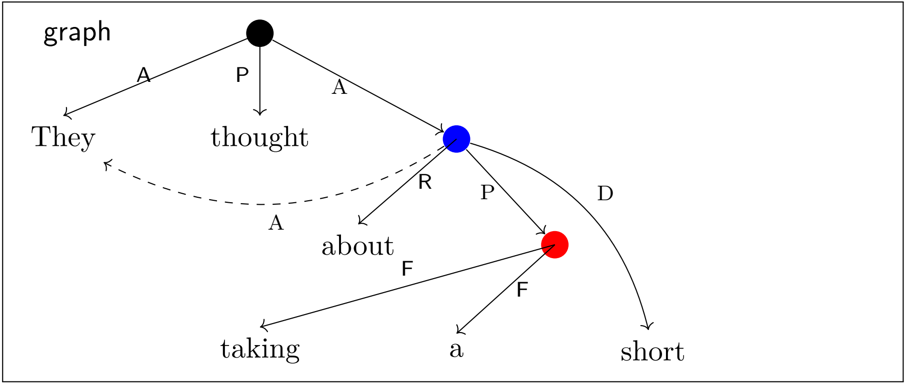
Example: TUPA Transition Sequence

⇒ Reduce

stack

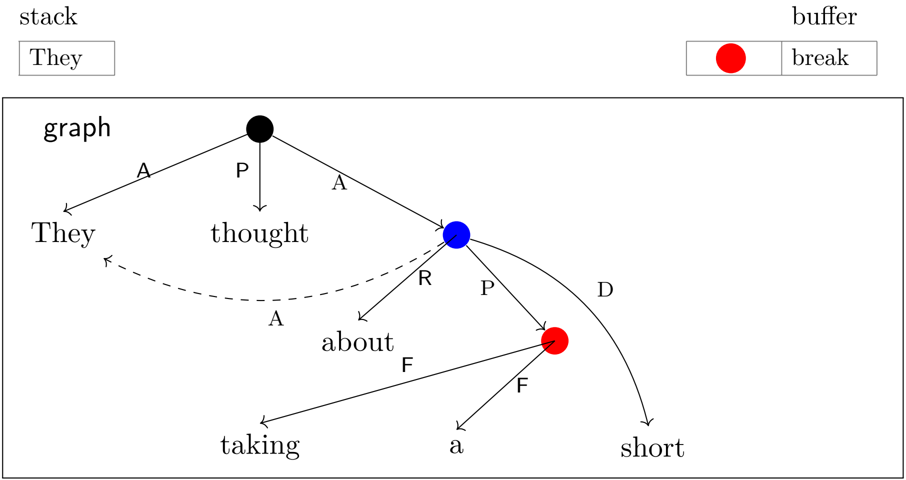
buffer

They	●	break



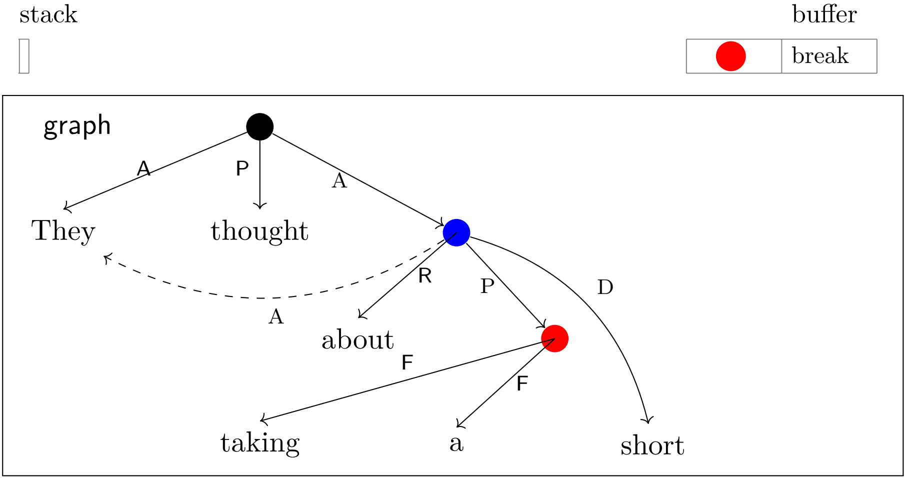
Example: TUPA Transition Sequence

⇒ SHIFT



Example: TUPA Transition Sequence

⇒ REDUCE

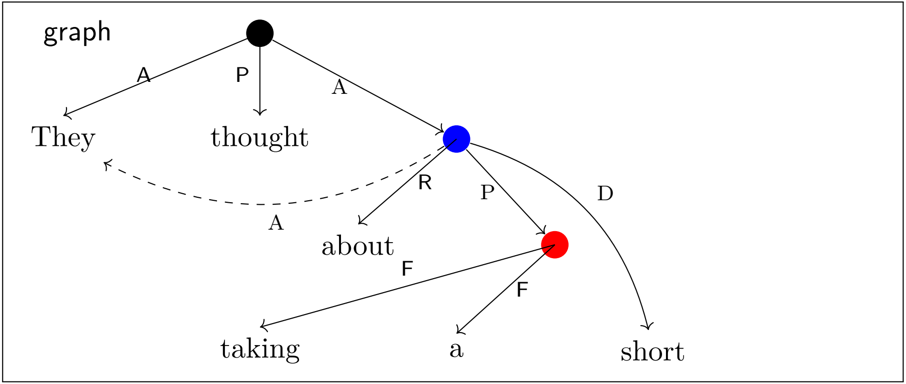


Example: TUPA Transition Sequence

⇒ SHIFT

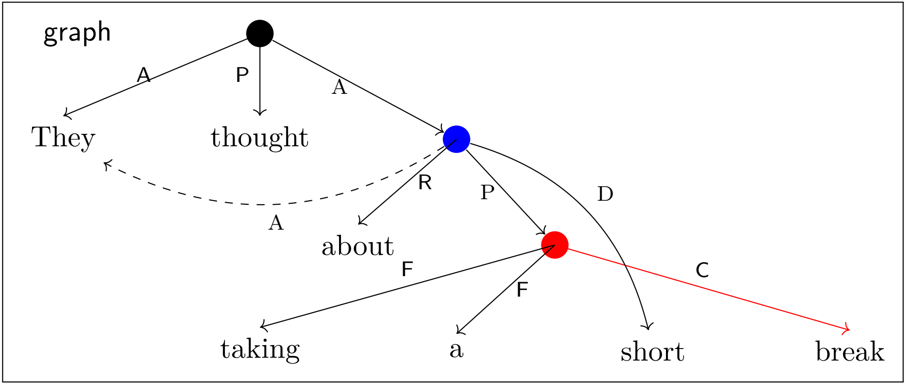
stack

buffer



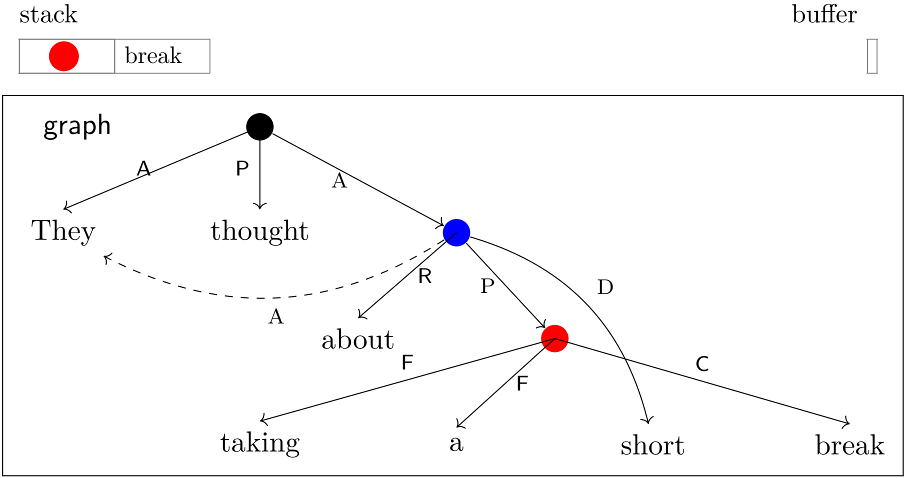
Example: TUPA Transition Sequence

⇒ RIGHT-EDGE$_C$



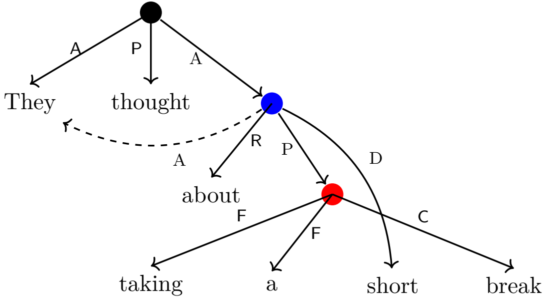
Example: TUPA Transition Sequence

⇒ FINISH



Training

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



$\Downarrow$

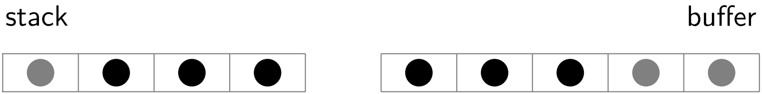
SHIFT, RIGHT-EDGE$_A$, SHIFT, SWAP, RIGHT-EDGE$_P$, REDUCE, SHIFT, SHIFT, NODE$_R$, REDUCE, LEFT-REMOTE$_A$, SHIFT, SHIFT, NODE$_C$, REDUCE, SHIFT, RIGHT-EDGE$_P$, SHIFT, RIGHT-EDGE$_F$, REDUCE, SHIFT, SWAP, RIGHT-EDGE$_D$, REDUCE, SWAP, RIGHT-EDGE$_A$, REDUCE, REDUCE, SHIFT, REDUCE, SHIFT, RIGHT-EDGE$_C$,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

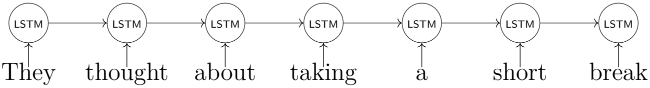
Features include:
{words, parts of speech, syntactic dependencies, existing edge labels}
from the stack and buffer + parents, children, grandchildren.



TUPA Model

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

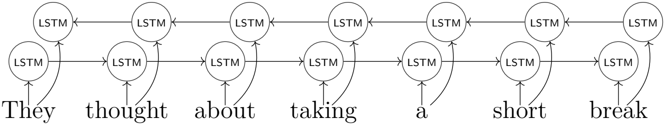
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.



TUPA Model

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

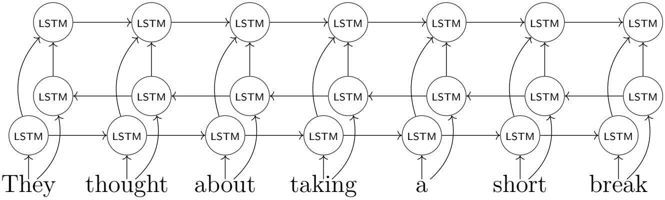
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.



TUPA Model

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

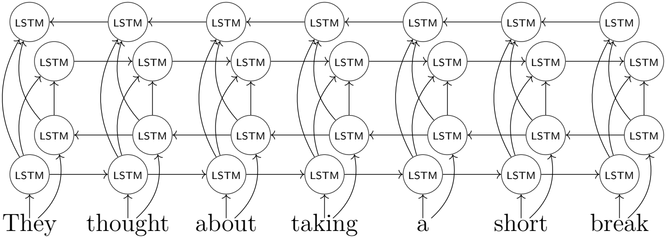
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.

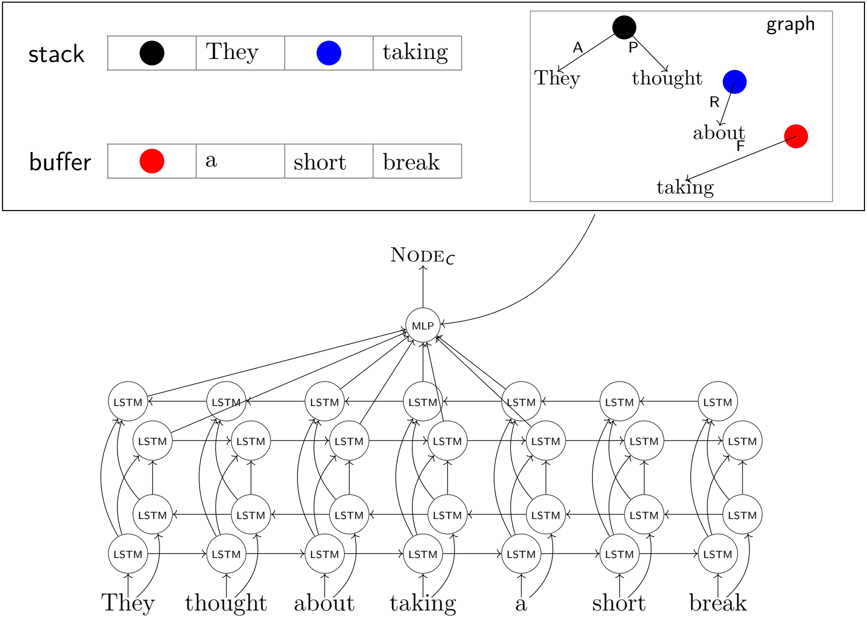


TUPA Model

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

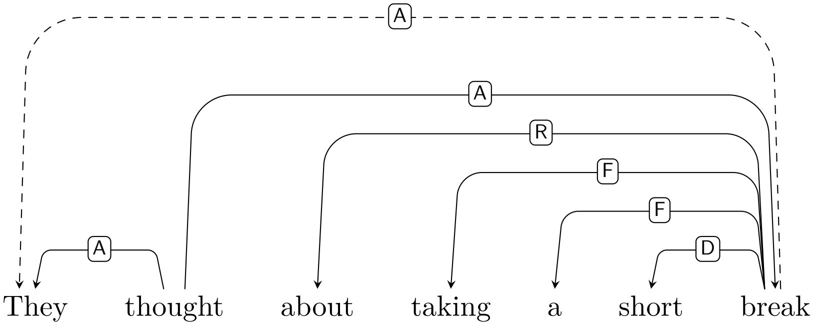
Sparse	Perceptron with sparse features.
MLP	Word embeddings + MLP.
BiLSTM	Word embeddings + **bidirectional RNN** + MLP.




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

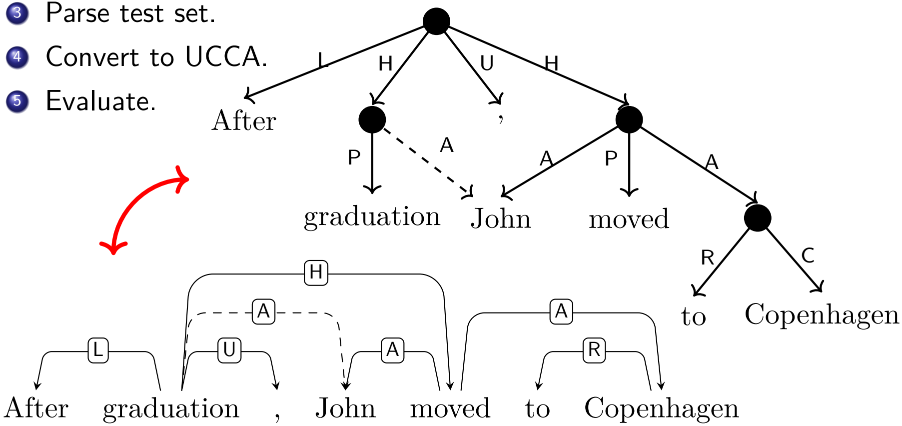
Using conversion-based approximation as baseline,
with bi-lexical DAG parsers and transition-based tree parsers.



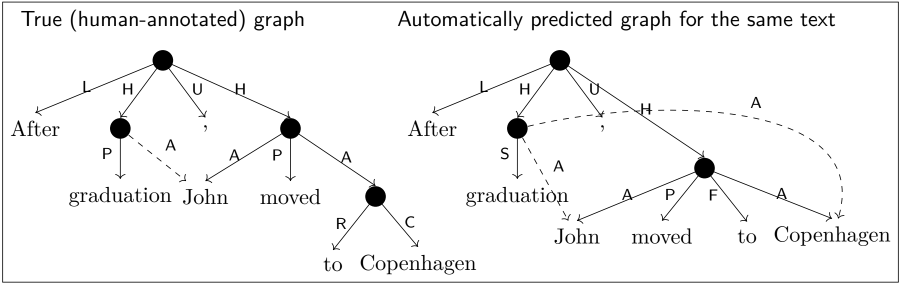
UCCA bi-lexical DAG approximation.

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.

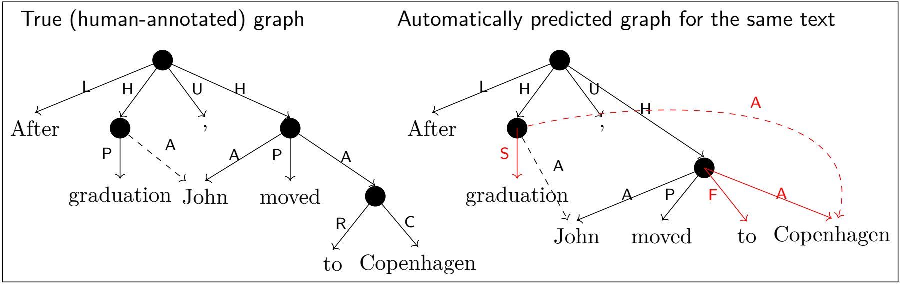


Evaluation



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

$\text{TUPA}_{\text{BiLSTM}}$ 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

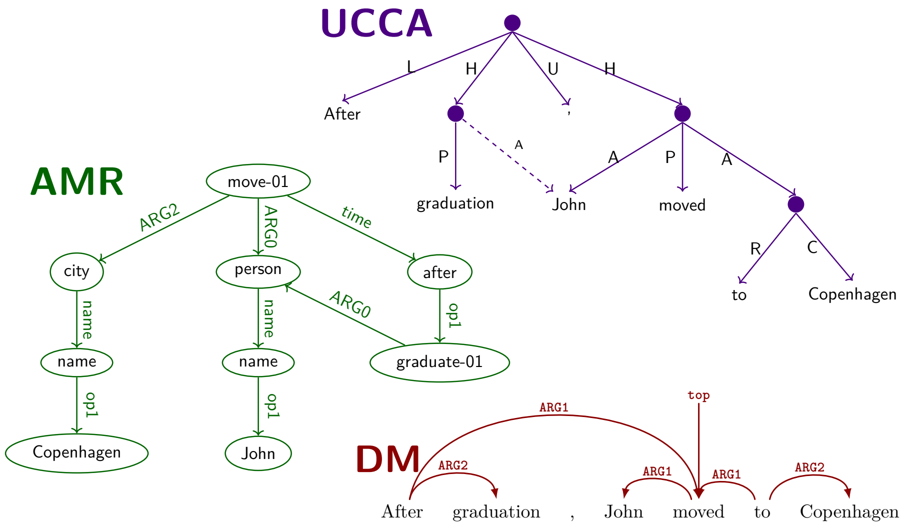
- Structured meaning representation benefits language understanding.
- UCCA's semantic distinctions require a graph structure including non-terminals, reentrancy and discontinuity.
- TUPA is an accurate transition-based UCCA parser, and the first to support UCCA and any DAG over the text tokens.
- Outperforms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

Up next:

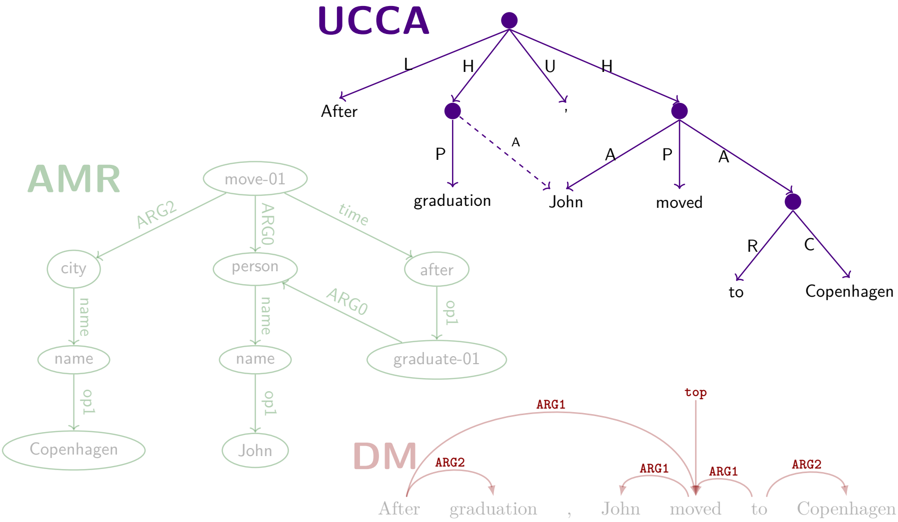
- Parsing other semantic representations.
- Comparing representations through conversion.

Outline

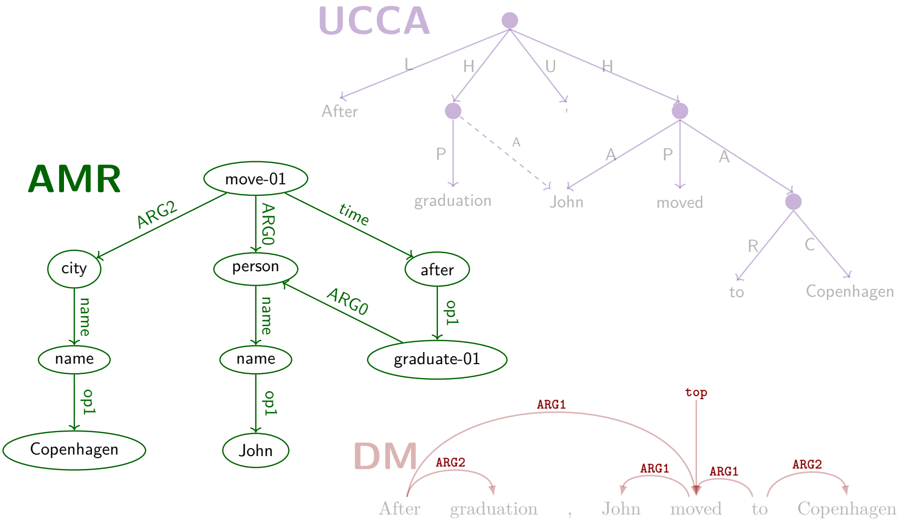
Semantic Representations



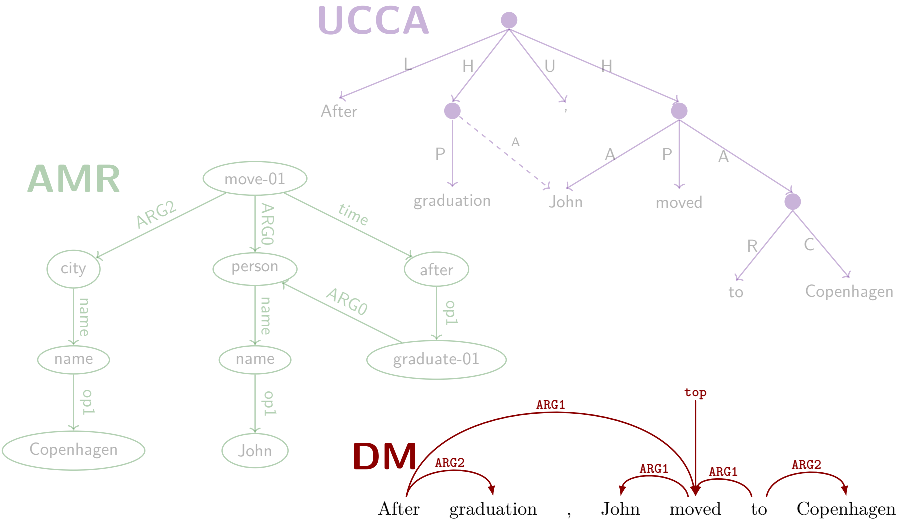
Semantic Representations



Semantic Representations

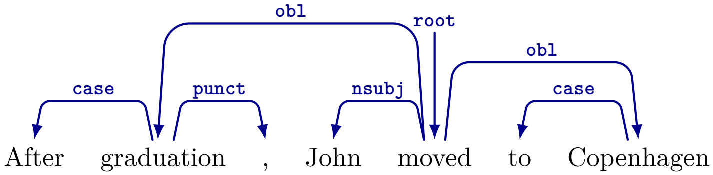


Semantic Representations



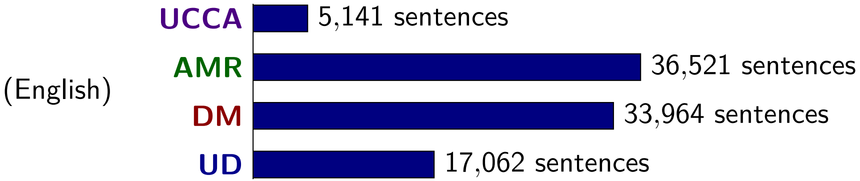
Syntactic Representations

UD (Universal Dependencies)



Data

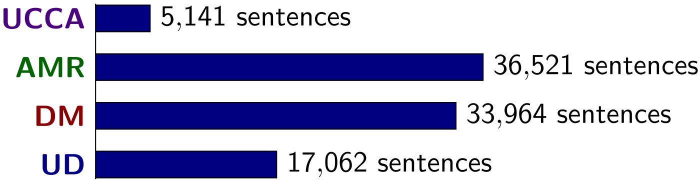
UCCA training data is scarce



Data

UCCA training data is scarce

(English)

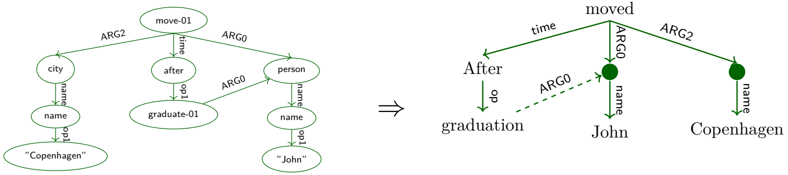


and domains are limited.

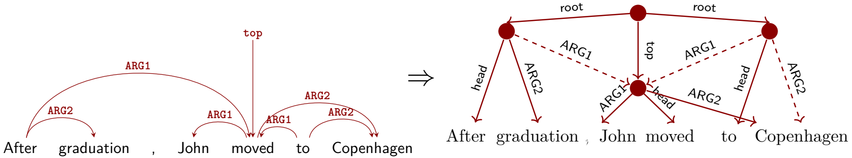
UCCA	**AMR**	**DM**	**UD**
Wikipedia	blogs	news	blogs
books	news		news
	emails		emails
	reviews		reviews
			Q&A

Conversion

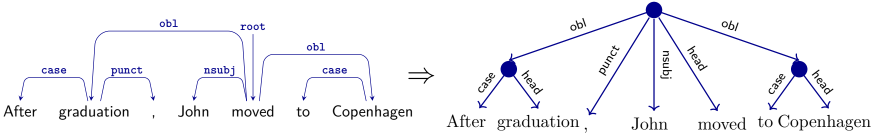
AMR



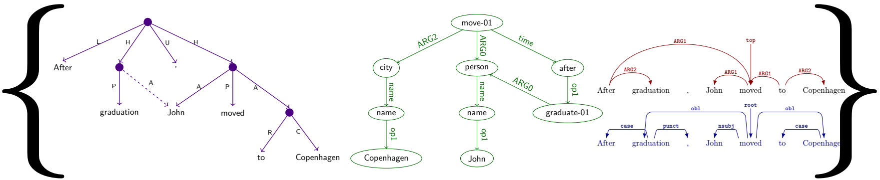
DM



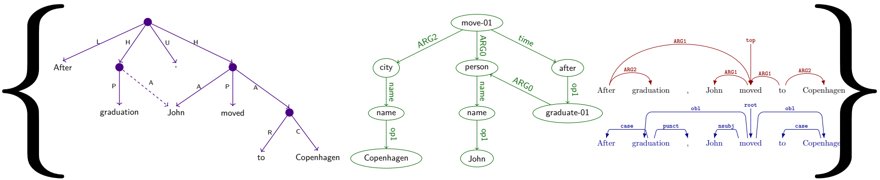
UD



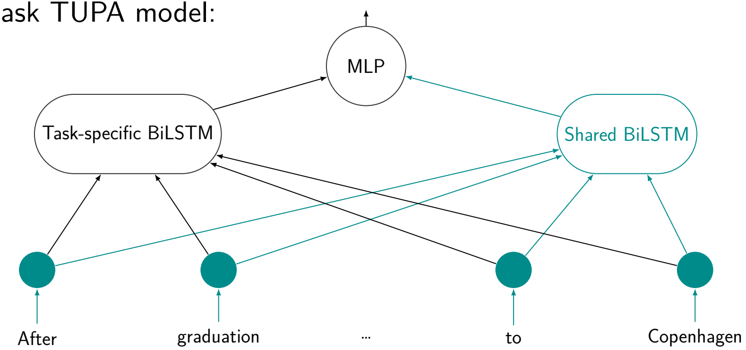
Multitask


After
graduation
John
moved
to
Copenhagen
move-01
ARG2
ARG0
time
city
person
after
name
op1
graduate-01
Copenhagen
John
ARG1
top
ARG2
After
graduation
moved
Copenhagen
obl
root
case
punct
nsubj

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

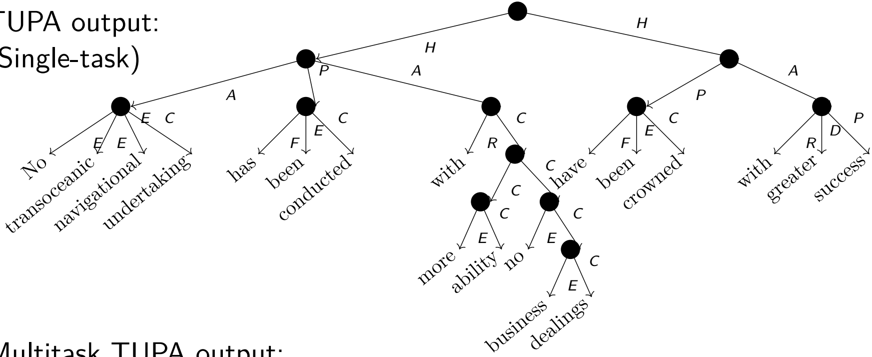
Results

	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

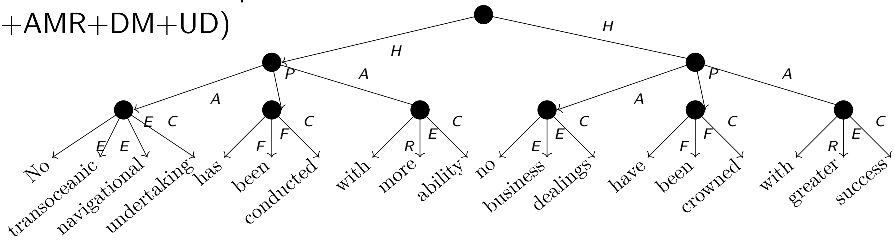
	Primary F1	Remote F1
English 20K (out-of-domain)		
Single-task	67.9	23.0
+AMR	67.0	31.2
+DM	**69.1**	27.5
+UD	67.4	23.9
+AMR+DM	68.9	25.4
+AMR+UD	68.2	**31.4**
+DM+UD	68.6	29.1
All	**69.1**	25.8

	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

Shared Task

- Data: English Wiki, English-French-German 20K

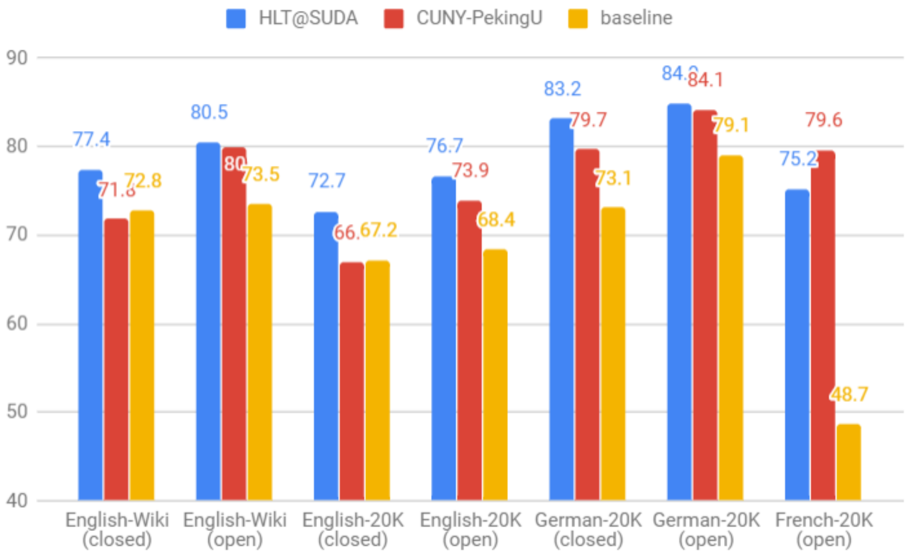
	sentences	tokens
English-Wiki	5,142	158,573
English-20K	492	12,574
French-20K	492	12,954
German-20K	6,514	144,531

- Tracks:
  - English {in-domain/out-of-domain} × {open/closed}
  - German in-domain {open/closed}
  - French *low-resource* (only 15 training sentences)
- Baseline: TUPA
- Evaluation period: January 10–31, 2019

Participating Systems

8 groups in total:

- *MaskParse@Deskiñ* Orange Labs, Aix-Marseille University
- *HLT@SUDA* Soochow University
- *TüPa* University of Tübingen
- *UC Davis* University of California, Davis
- *GCN-Sem* University of Wolverhampton
- *CUNY-PekingU* City University of New York, Peking University
- *DANGNT@UIT.VNU-HCM* University of Information Technology VNU-HCM
- *XLangMo* Zhejiang University


HLT@SUDA
CUNY-PekingU
baseline
90
80
70
60
50
40
77.4
71.8
72.8
80.5
80
73.5
72.7
66.
67.2
76.7
73.9
68.4
83.2
79.7
73.1
84.
84.1
79.1
75.2
79.6
48.7
English-Wiki (closed)
English-Wiki (open)
English-20K (closed)
English-20K (open)
German-20K (closed)
German-20K (open)
French-20K (open)

Main Findings

- HLT@SUDA won 6/7 tracks:
  Neural constituency parser + multi-task + BERT
  French: trained on all languages, with language embedding

Main Findings

- HLT@SUDA won 6/7 tracks:
  Neural constituency parser + multi-task + BERT
  French: trained on all languages, with language embedding
- CUNY-PekingU won the French (open) track:
  TUPA ensemble + synthetic data by machine translation

Main Findings

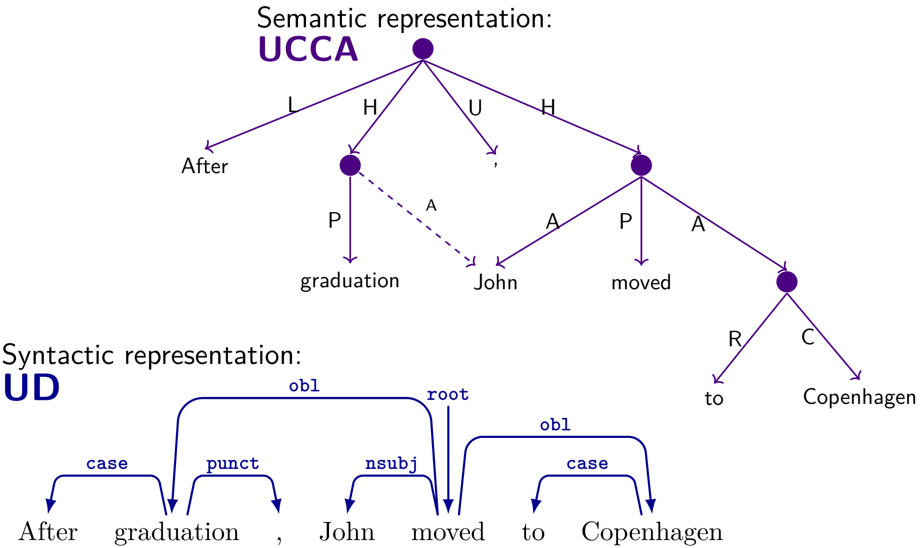
- HLT@SUDA won 6/7 tracks:
  Neural constituency parser + multi-task + BERT
  French: trained on all languages, with language embedding
- CUNY-PekingU won the French (open) track:
  TUPA ensemble + synthetic data by machine translation

Surprisingly, 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?

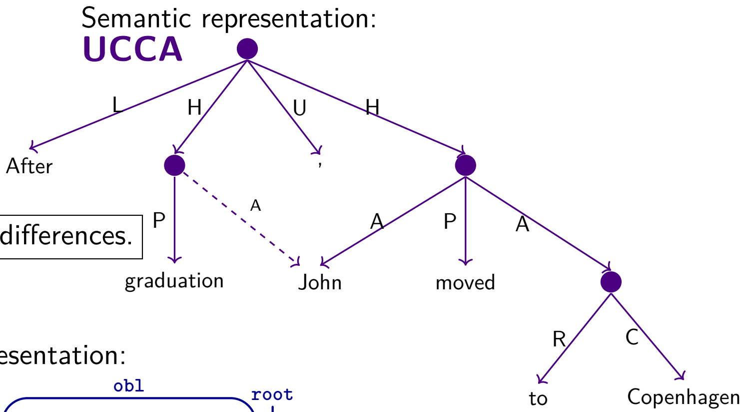
Outline

UCCA vs. UD

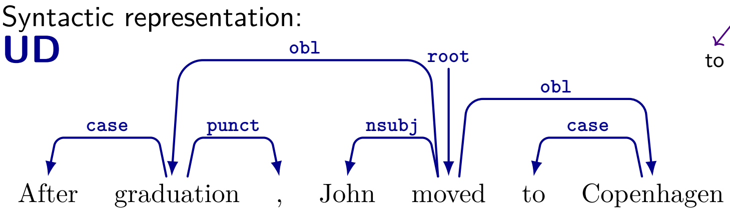


Semantic representation:
UCCA
L
H
U
H
After
,
P
A
A
P
A
graduation
John
moved
R
C
to
Copenhagen
Syntactic representation:
UD
obl
root
obl
case
punct
nsubj
case
After graduation , John moved to Copenhagen

UCCA vs. UD



Many formal differences.

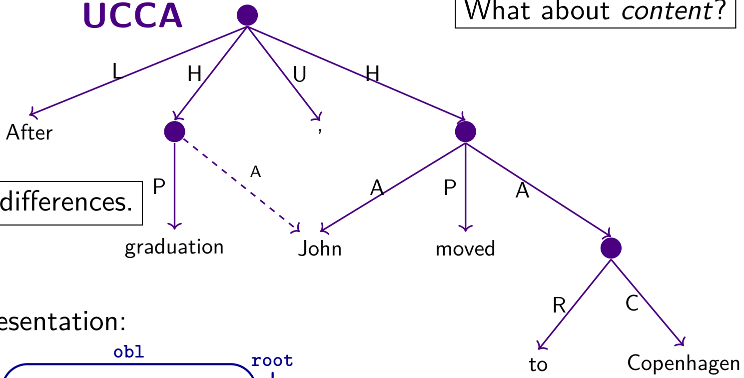


UCCA vs. UD

Semantic representation:

UCCA

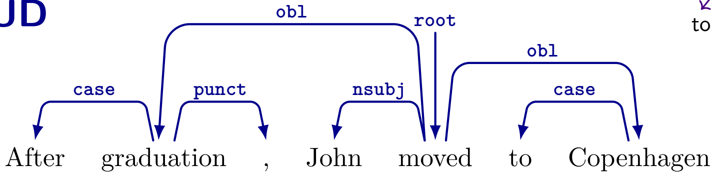
What about *content*?



Many formal differences.

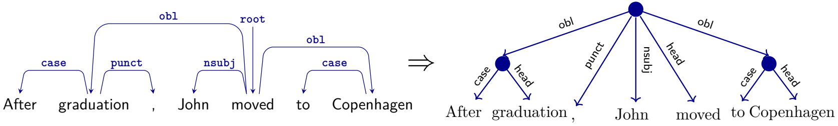
Syntactic representation:

UD



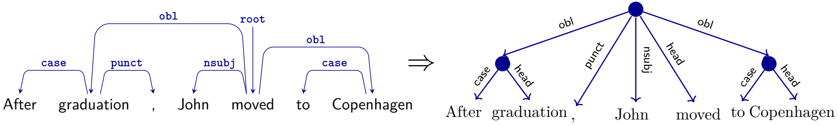
Assimilating the Graph Structures

UD



Assimilating the Graph Structures

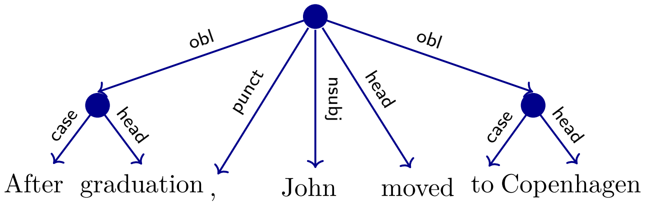
UD



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

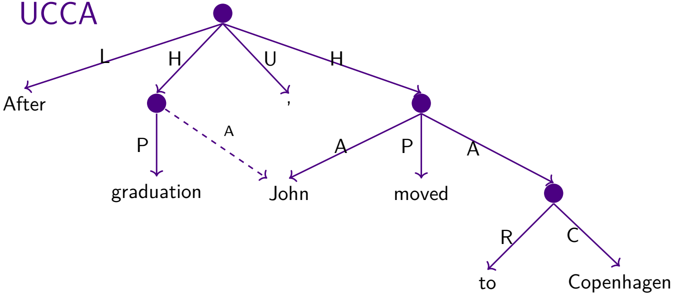
Assimilating the Graph Structures

UD



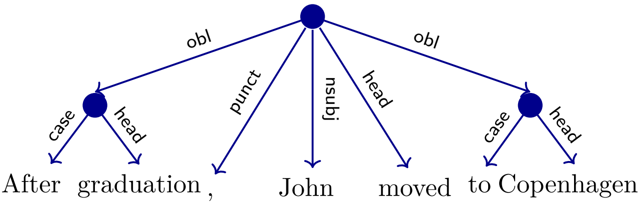
Now we can evaluate by matching edges (UCCA unlabeled evaluation)

UCCA



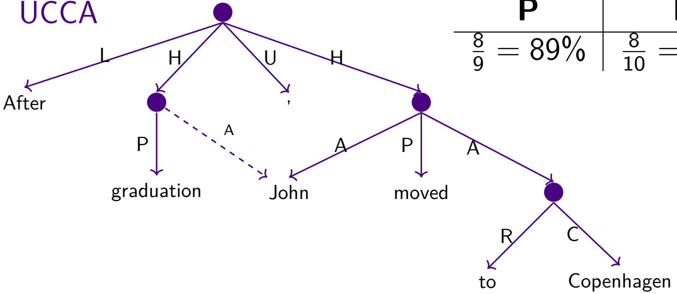
Assimilating the Graph Structures

UD



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

UCCA

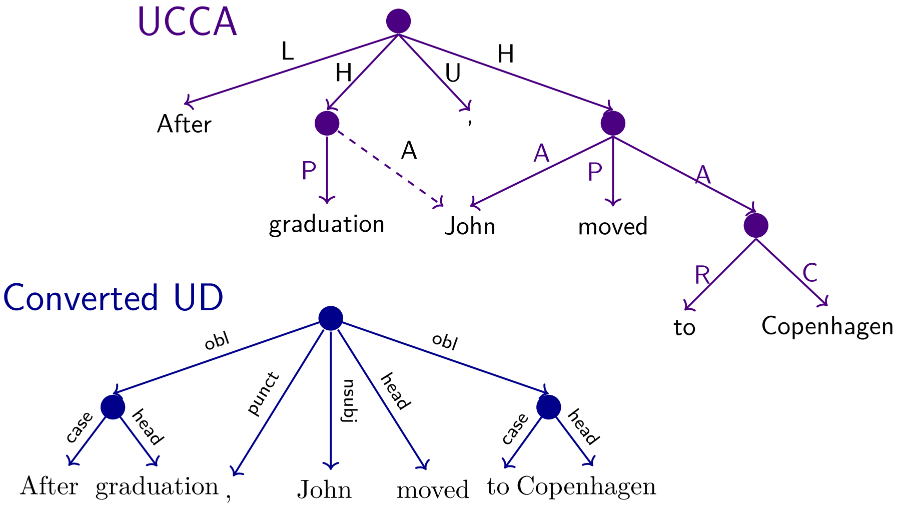


P	**R**	**F1**
$\frac{8}{9} = 89\%$	$\frac{8}{10} = 80\%$	84%

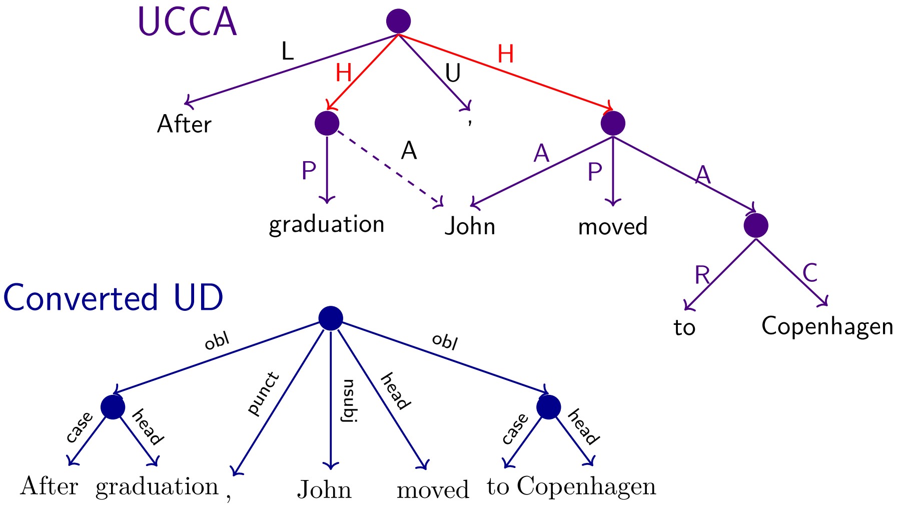
Confusion Matrix

	A	A|P	A|S	C	D	E	F	G	H	L	N	P	Q	R	S	T	No Match
acl	58			1	4	249	1		48			6			1	1	409
advcl	14			12	2	2		6	512	4		11					423
advmod	225		1	69	1778	332	27	135	14	258	2	2	15	44	9	368	273
amod	25			134	647	837		1	28			7	130	3	269	25	176
appos	21			39	2	34			18						8		33
aux					384	2	1335			2		1		1			17
case	11			31	27	25	123			213	26	11	1	2629	154	1	262
cc				8	4	1	4	1	1	1567	381		6	12			52
ccomp	345			1		1			36			2			1	1	166
compound	225			116	67	586	21		2			32	19	1	12	24	683
conj	10			449	4	5		1	1262	1		6	2		10		497
cop				1			1312			1		9		10	178		7
csubj	13								3								46
det	10			17	119	440	2963				1		129	16	1		124
discourse	1			2	1		25	29	27	16					5		19
expl	21			1			98								17		3
iobj	131			1			1										10
list	3			7	2	1			27						1		6
mark				9	7	1	531	1		654				407	1	5	143
nmod	844	1	1	20	9	786	8	4	12	1	1	20	2	2	11	27	488
nsubj	4296	7	21	25	3	2	55	1	5	61		58	1	80	14	4	247
nummod	2			33	12	17		4		4			334				64
obj	1845		1	54	21	6	11	1	4	23		52	1	23	3	11	583
obl	1195			19	115	41	1	17	39	34		6	6	26	7	302	611
parataxis	6		1	5		4		6	285						3		180
vocative	17							8									
xcomp	121			4	25				8			38			38		526
head	445	48	159	6388	717	142	564	83	2462	42	1	4163	120	52	1547	32	2235
No Match	1421	37	58	640	417	291	14	33	2291	146	6	802	94	52	369	96	

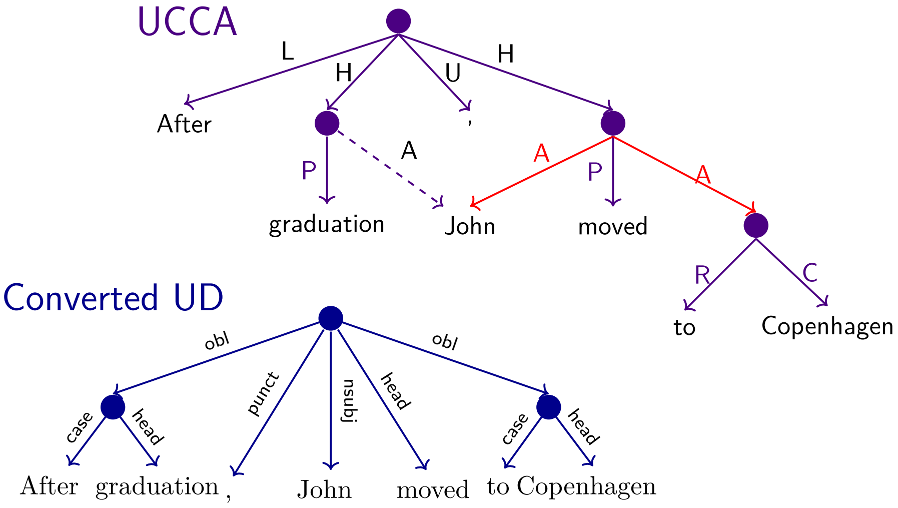
Scenes and non-Scenes, Relations and Participants



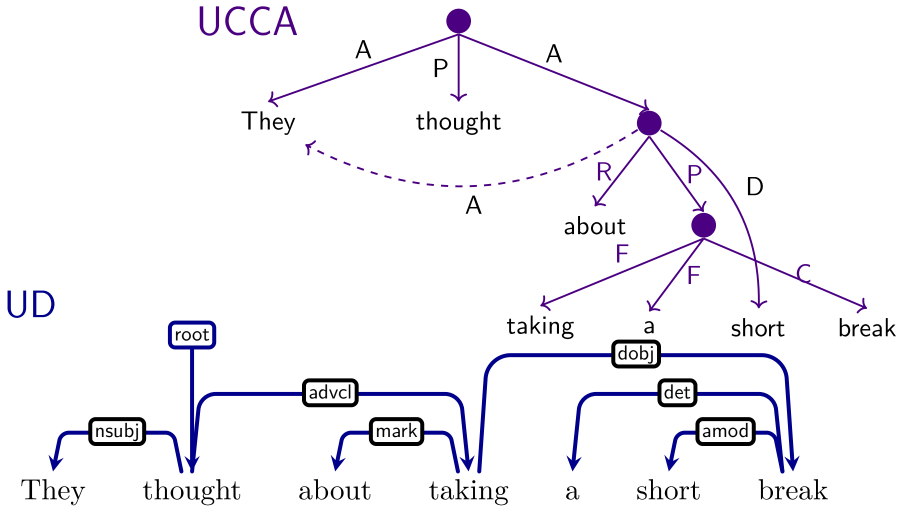
Scenes and non-Scenes, Relations and Participants



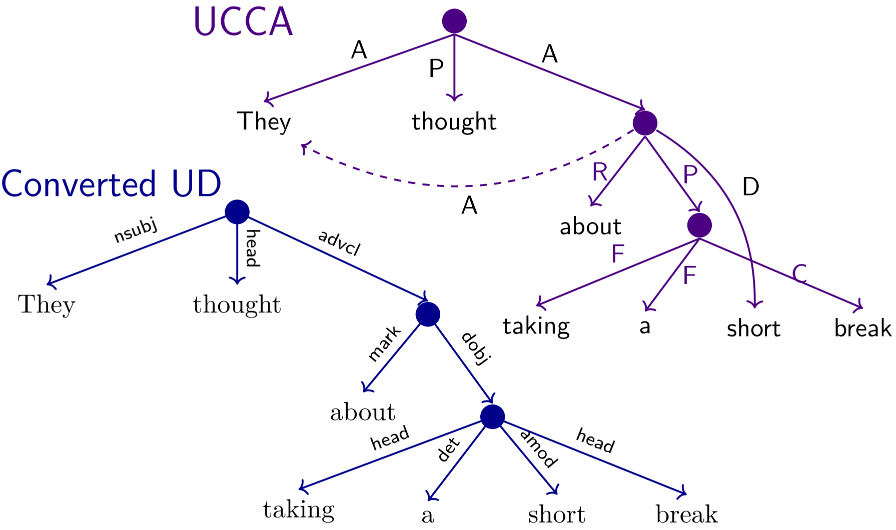
Scenes and non-Scenes, Relations and Participants



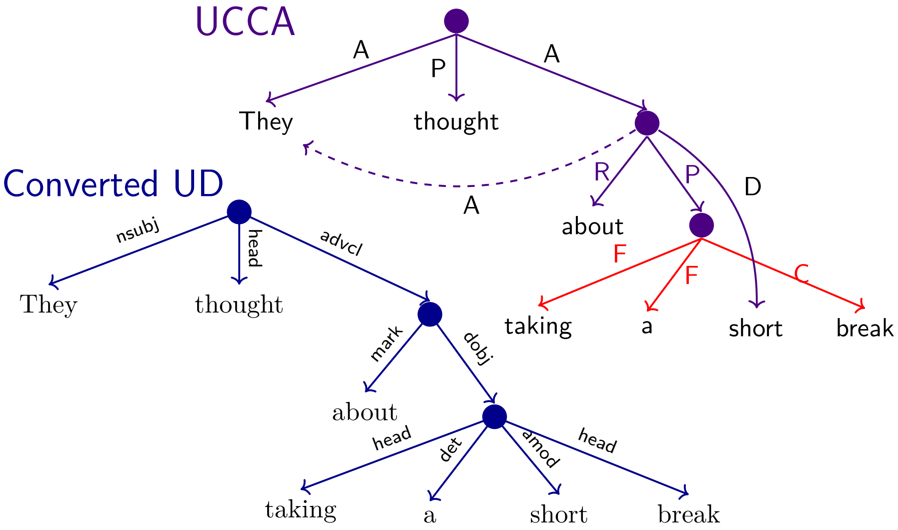
Multi-word Expressions



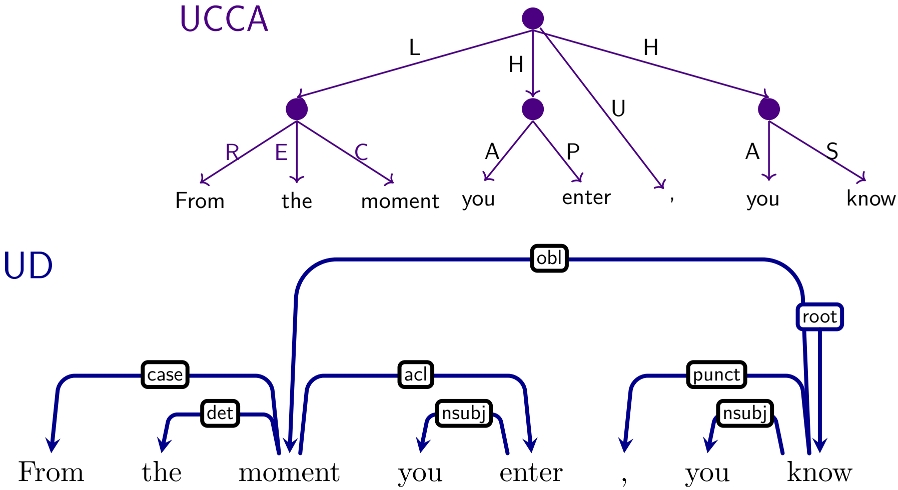
Multi-word Expressions



Multi-word Expressions

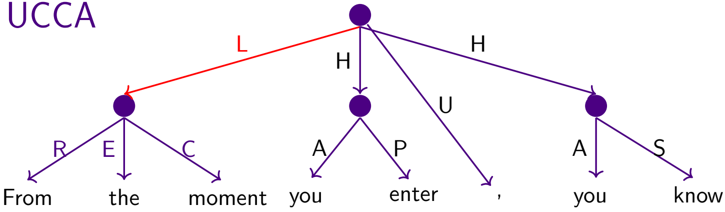


Linkage between Scenes

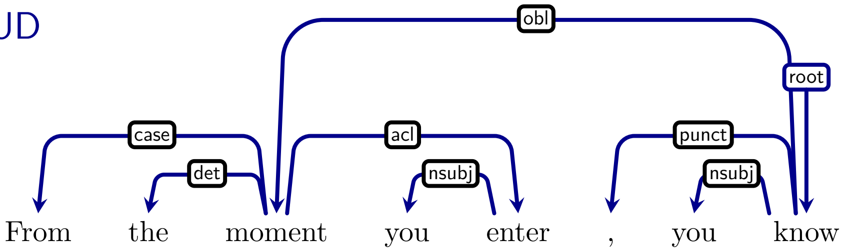


Linkage between Scenes

UCCA



UD

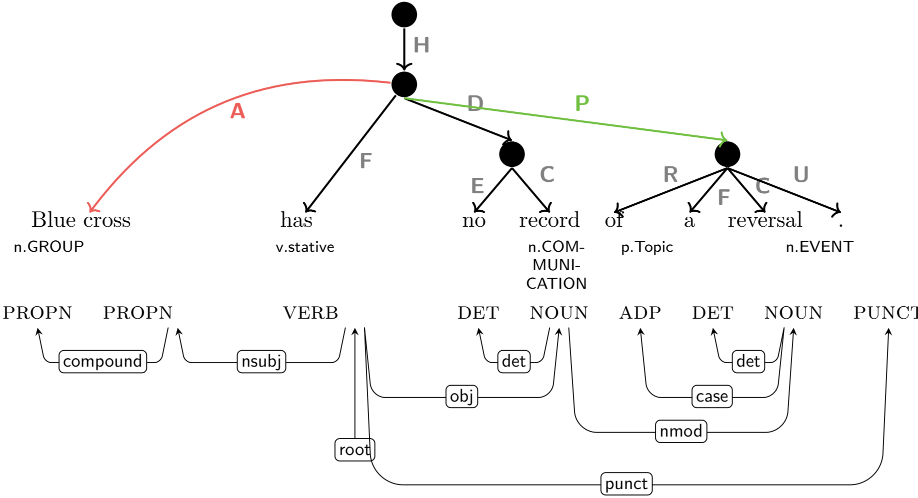


Outline

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. Cross-lingual Semantic Parsing with UCCA (SemEval'19)
5. Content Differences in Syntactic and Semantic Representations (NAACL'19)
6. Ongoing Work

UCCA in Terms of Syntax and Lexical Semantics

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



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.

CoNLL 2019 Shared Task:
Cross-Framework Meaning Representation Parsing
SDP, EDS, AMR and UCCA

`mrp.nlpl.eu`

Evaluation Period: July 8–22, 2019

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*.