Broad-Coverage Transition-Based UCCA Parsing

Daniel Hershcovich[1,2] & Omri Abend[2] & Ari Rappoport[2]

[1]Edmond and Lily Safra Center for Brain Sciences
[2]School of Computer Science and Engineering
Hebrew University of Jerusalem

Learning Club
November 17, 2016

Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions

Semantic Annotation Schemes

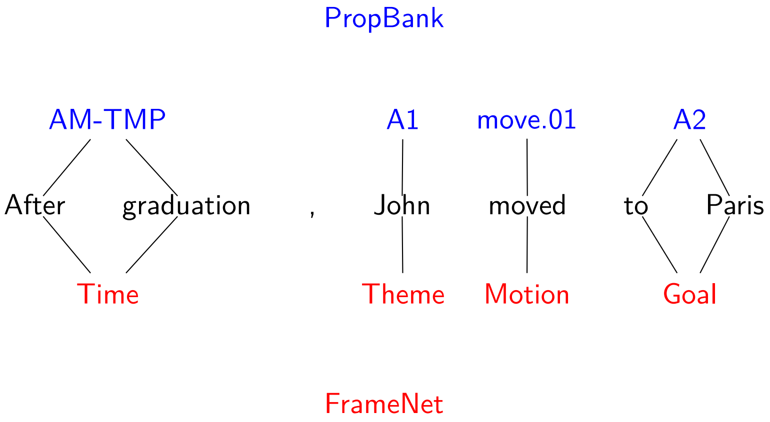
Represent semantic structure of text as a graph.
Used by NLP applications for features and structure, providing information such as *who did what to whom?*

Examples:

- Semantic Role Labeling
- Semantic Dependencies
- Abstract Meaning Representation
- Universal Conceptual Cognitive Annotation

Semantic Role Labeling (SRL)

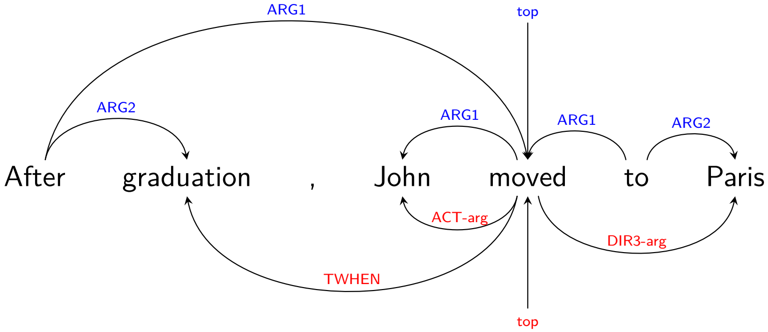
Annotate predicates and their arguments as a flat structure. Examples:



Semantic Dependency Parsing (SDP)

Graph on the text tokens, including internal structure of arguments.
Examples:

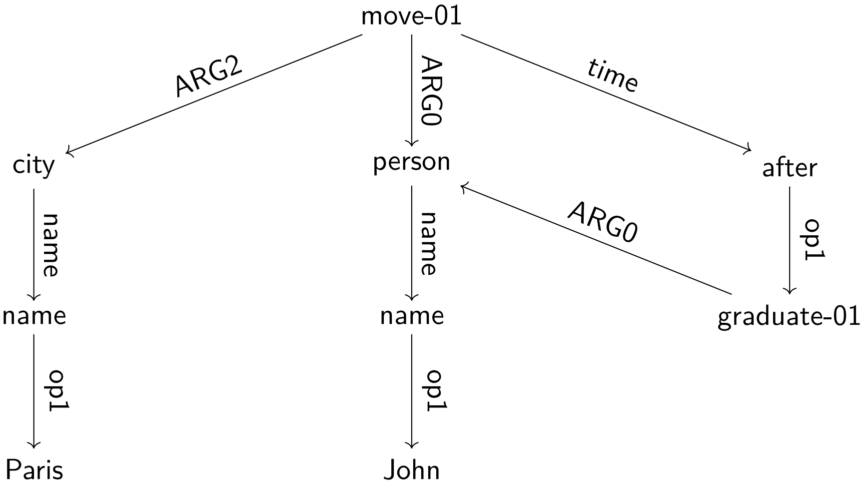
DELPH-IN MRS-derived bi-lexical dependencies (DM)



Prague Dependency Treebank tectogrammatical layer (PSD)

Abstract Meaning Representation (AMR)

Graph on knowledge resource entries inferred from the tokens.



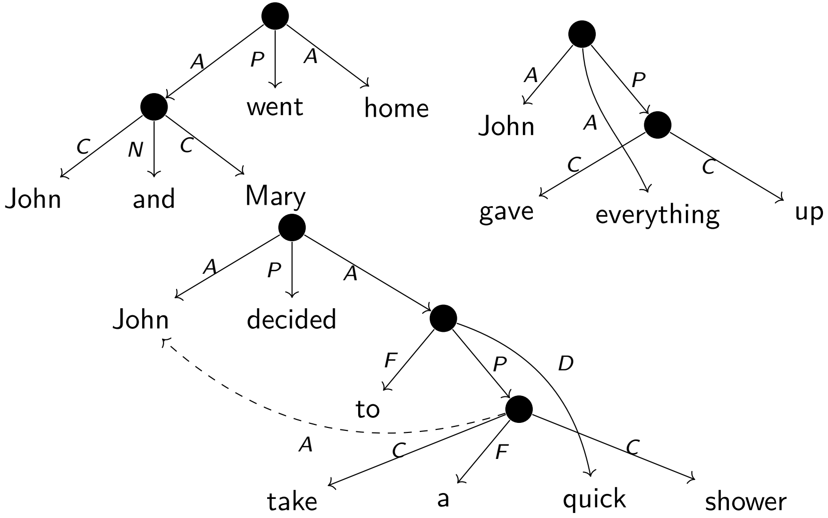
Universal Conceptual Cognitive Annotation (UCCA)

Cross-linguistically applicable semantic representation scheme.

- Builds on typological [Dixon, 2012] and Cognitive Linguistics literature [Croft and Cruse, 2004].
- Demonstrated applicability to English, French, German & Czech.
- Support for rapid annotation.
- Semantic stability in translation [Sulem et al., 2015].
- Proven useful for machine translation evaluation [Birch et al., 2016].
- Applicability has been so far limited by the absence of a parser.

Structural Properties

(1) non-terminal nodes, (2) reentrancy, (3) discontinuity



UCCA Corpora

	Wiki Train	Wiki Dev	Wiki Test	20K Leagues
# passages	300	34	34	154
# sentences	4267	453	518	506
# nodes	298,665	33,263	37,262	29,315
% terminal	42.95	43.62	42.89	42.09
% non-term.	58.30	57.46	58.31	60.01
% discont.	0.53	0.51	0.47	0.81
% reentrant	2.31	1.76	2.18	2.03
# edges	287,381	32,015	35,846	27,749
% primary	98.29	98.81	98.75	97.73
% remote	1.71	1.19	1.25	2.27
Average per non-terminal node				
# children	1.67	1.68	1.66	1.61

[a]

[a] Excluding root node, implicit nodes, and linkage nodes and edges.

Outline

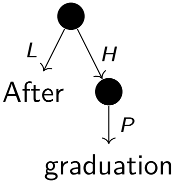
1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions

Transition-Based Parsing

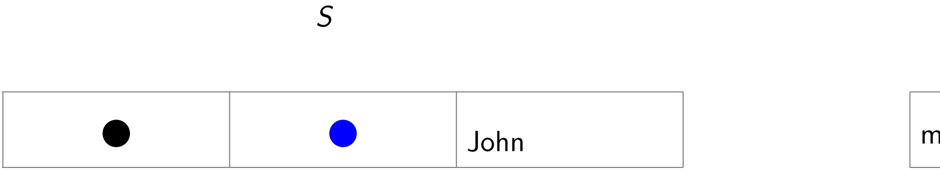
Parse sentence $w_1 \ldots w_n$ to graph $G = (V, E, \ell)$ incrementally, using buffer B and stack S. Classifier determines transition to apply at each step. Transition-based parsers work by applying a *transition* at each step to the parser state, defined using a buffer B of tokens and nodes to be processed, a stack S of nodes currently being processed, and a graph $G = (V, E, \ell)$ of constructed nodes and edges. A classifier selects the next transition based on the current state's features. It is trained by an oracle based on gold-standard annotations.

S	After	graduation	B	John	moved	to	Paris

G



Transition-Based Parsing



Transitions: SHIFT, RIGHT-EDGE$_{\text{L}}$, REDUCE, SHIFT,

TUPA (Transition-Based UCCA Parser)

Our parser supports the structural properties of UCCA.[1]

Before Transition					After		
Stack	**Buffer**	**Nodes**	**Edges**		**Stack**	**Buffer**	**N**
S	$x \mid B$	V	E	SHIFT	$S \mid x$	B	V
$S \mid x$	B	V	E	REDUCE	S	B	V
$S \mid x$	B	V	E	NODE$_X$	$S \mid x$	$y \mid B$	V
$S \mid y, x$	B	V	E	LEFT-EDGE$_X$	$S \mid y, x$	B	V
$S \mid x, y$	B	V	E	RIGHT-EDGE$_X$	$S \mid x, y$	B	V
$S \mid y, x$	B	V	E	LEFT-REMOTE$_X$	$S \mid y, x$	B	V
$S \mid x, y$	B	V	E	RIGHT-REMOTE$_X$	$S \mid x, y$	B	V
$S \mid x, y$	B	V	E	SWAP	$S \mid y$	$x \mid B$	V
[root]	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	V

Table: TUPA transitions. $(\cdot,\cdot)_X$ denotes a primary X-labeled edge, and $(\cdot,\cdot)^*_X$ a remote X-labeled edge.

2

TUPA Classifiers

We experiment with three classifiers:

TUPA$_{\textbf{sparse}}$	Perceptron, sparse features: words, POS tags & edge label co
TUPA$_{\textbf{dense}}$	Perceptron, dense embedding features: word2vec [Mikolov et
TUPA$_{\textbf{NN}}$	2-layer MLP, learned embedding features, logistic activation +

For all classifiers, inference is performed greedily, i.e., without beam search.

Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions

Experimental Setup

We conduct our main experiment on the UCCA Wikipedia corpus, and use the English part of the UCCA *Twenty Thousand Leagues Under the Sea* English-French parallel corpus as out-of-domain data.

Evaluation

We report two variants of labeled precision, recall and F-score: one where we consider only primary edges, and another for remote edges. Given graphs $G_p = (V_p, E_p, \ell_p)$ and $G_g = (V_g, E_g, \ell_g)$ over terminals $W = \{w_1, \dots, w_n\}$, the yield $y(e) \subseteq W$ of an edge $e = (u, v)$ in either graph is the set of terminals in W that are descendants of v. The *mutual edges* between the graphs are:

$$M(G_p, G_g) = \{(e_1, e_2) \in E_p \times E_g \mid y(e_1) = y(e_2) \wedge \ell_p(e_1) = \ell_g(e_2)\}$$

and we define

$\mathrm{LP} = |M(G_p, G_g)|/|E_p|$ $\mathrm{LR} = |M(G_p, G_g)|/|E_g|$

Baselines

Since there are no existing UCCA parsers, we use bilexical DAG parsers:

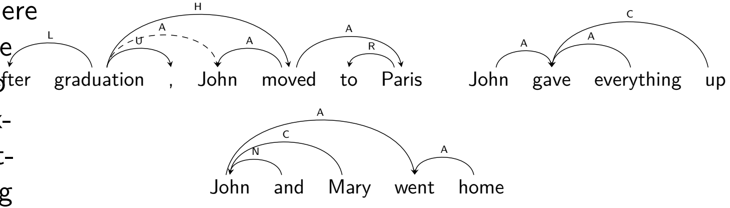


Figure: Bilexical approximation for UCCA graphs.

1. Convert UCCA into bilexical dependencies.
2. Train parsers on the resulting training set.
3. Apply trained parsers to the test set.
4. Reconstruct UCCA graphs.

Results

TUPA_{NN} obtains the highest scores in nearly all metrics:

	Wiki (in-domain)				
	Primary			Rem	
	LP	**LR**	**LF**	**LP**	**L**
Bilexical Approximation					
Upper Bound	93.4	83.7	88.3	73.9	49.
DAGParser [Ribeyre et al., 2014]	63.7	56.1	59.5	0.8	9.
TurboParser [Almeida and Martins, 2015]	60.2	47.4	52.9	2.2	7.
Direct Approach					
$\text{TUPA}_{\text{sparse}}$	64	55.6	59.5	16	11.
$\text{TUPA}_{\text{dense}}$	55	54.8	54.9	15.2	**16.**
TUPA_{NN}	**65**	**62.5**	**63.7**	**20.7**	11.

Tree Approximation

For completeness, we also explore lossily converting UCCA into trees, resulting in a simplified task for the underlying parser, in addition to the maturity of tree-based parsers.

Although remote edges are of pivotal importance, exploring tree approximation methods can inform the future development of DAG parsers in general and of UCCA parsers in particular.

Constituency Tree Approximation			
Upper Bound	100	100	100
UPARSE [Maier and Lichte, 2016]	63	64.7	63.7
Dependency Tree Approximation			
Upper Bound	93.7	83.6	88.4
MaltParser [Nivre et al., 2007]	64.9	57.9	61
LSTM Parser [Dyer et al., 2015]	**74.9**	**66.4**	**70.2**
Direct Tree Parsing			
TUPA$_{Sparse}$ − REMOTE	65.5	57.5	61.3

Outline

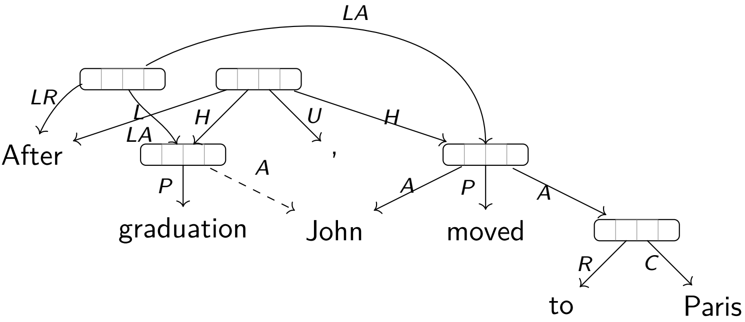
1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions

UCCA-Based Distributed Representation

Vector representation for sentences and documents, based on recursive composition on the UCCA graph.

Impact:

- General automatic semantic feature extractor for text.
- Accurate measure for text similarity.
- Understand the semantic contribution of different elements.



Outline

1 Semantic Annotation Schemes

2 Transition-based UCCA Parsing

3 Experiments

4 Future Work

5 Conclusions

Conclusions

We present TUPA, the first parser for UCCA, and evaluate it in both in-domain and out-of-domain settings, showing it surpasses bilexical DAG parsers on the task of UCCA parsing.
Future work will incorporate LSTMs into TUPA, and apply the parser to more languages such as German, demonstrating the importance of broad-coverage parsing. We will also improve the conversion-based methods and explore different target representations. A UCCA parser will enable using the scheme for representation in NLP tasks.

- UCCA exhibits formal properties important for semantic representation.
- We present the first parser for UCCA and the first to support these properties.

References I

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

Almeida, M. S. C. and Martins, A. F. T. (2015).
Lisbon: Evaluating TurboSemanticParser on multiple languages and out-of-domain data.
In *Proc. of SemEval*, pages 970–973, Denver, Colorado. Association for Computational Linguistics.

Birch, A., Haddow, B., Bojar, O., and Abend, O. (2016).
Hume: Human UCCA-based evaluation of machine translation.
arXiv preprint arXiv:1607.00030.

Croft, W. and Cruse, D. A. (2004).
Cognitive linguistics.
Cambridge University Press.

Dixon, R. M. W. (2010/2012).
Basic linguistic theory.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015).
Transition-based dependeny parsing with stack long short-term memory.
In *Proc. of ACL*, pages 334–343.

Maier, W. and Lichte, T. (2016).
Discontinuous parsing with continuous trees.
In *Proc. of Workshop on Discontinuous Structures in NLP*, pages 47–57, San Diego, California. Association for Computational Linguistics.

References II

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013).
Efficient estimation of word representations in vector space.
arXiv preprint arXiv:1301.3781.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007).
MaltParser: A language-independent system for data-driven dependency parsing.
Natural Language Engineering, 13(02):95–135.

Ribeyre, C., Villemonte de la Clergerie, E., and Seddah, D. (2014).
Alpage: Transition-based semantic graph parsing with syntactic features.
In *Proc. of SemEval*, pages 97–103, Dublin, Ireland.

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.