Universal Semantic Parsing

Daniel Hershcovich Omri Abend Ari Rappoport

The Hebrew University

Introduction

Structure plays a crucial role in language understanding and generation. Meaning cannot be accurately inferred without *parsing* the input to find the relations between words and the semantic units they represent.

It reminded me what I forgot ≠ I forgot what it reminded me

Word order reflects a structural difference in meaning

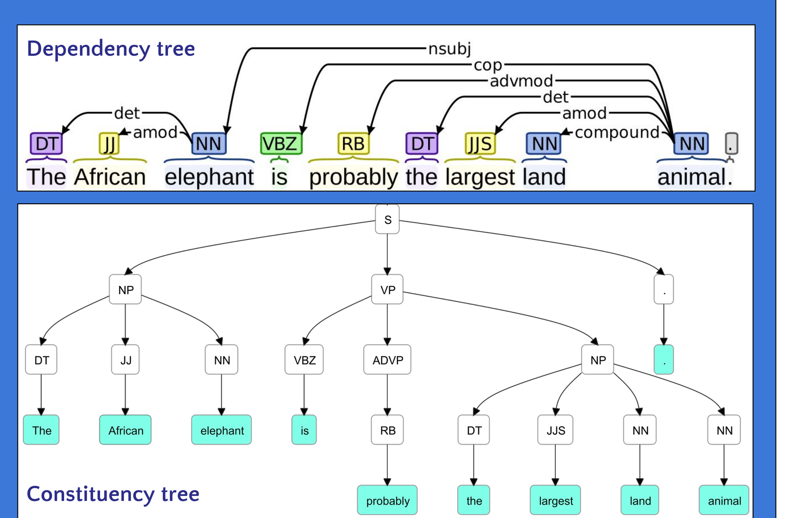
Understanding the cognitive processing of natural language requires a parsing model that reflects the most general structures found in it.

Grammatical Schemes

The most commonly used parsing models are *syntactic*: constituency and dependency grammars.

Common semantic schemes have a shallow structure, which is partial.

The majority of parsers are limited to only handle *projective trees*.



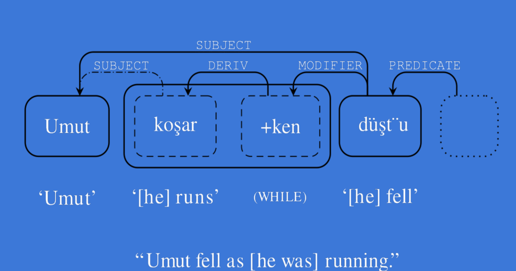
In general, grammatical models may have the following constraints:

1. **Single parents.** A unit may not be a child of more than one parent.
2. **Projectivity.** The tokens spanned by a unit may not contain gaps.
3. **Dependency.** All relations are between tokens, without non-terminals.

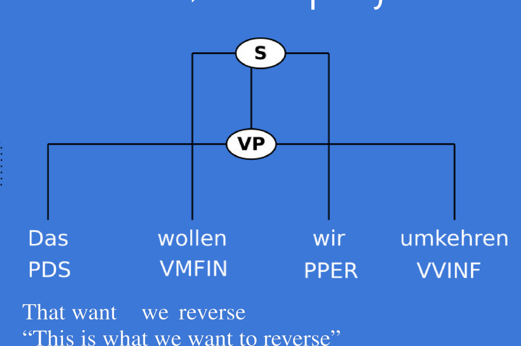
Dependency parsers have all three limitations (except DAG parsers and non-projective parsers), while constituency parsers are limited by 1 and 2.

General Structure

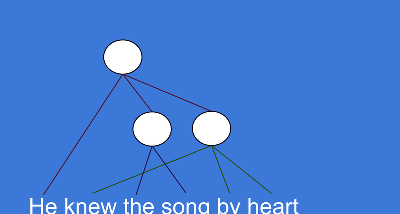
Natural language exhibits structures that are not subject to the above constraints: they may be non-tree, non-projective constituent graphs.



Multiple parents in Turkish



Discontinuity in German



Multi-word units in English

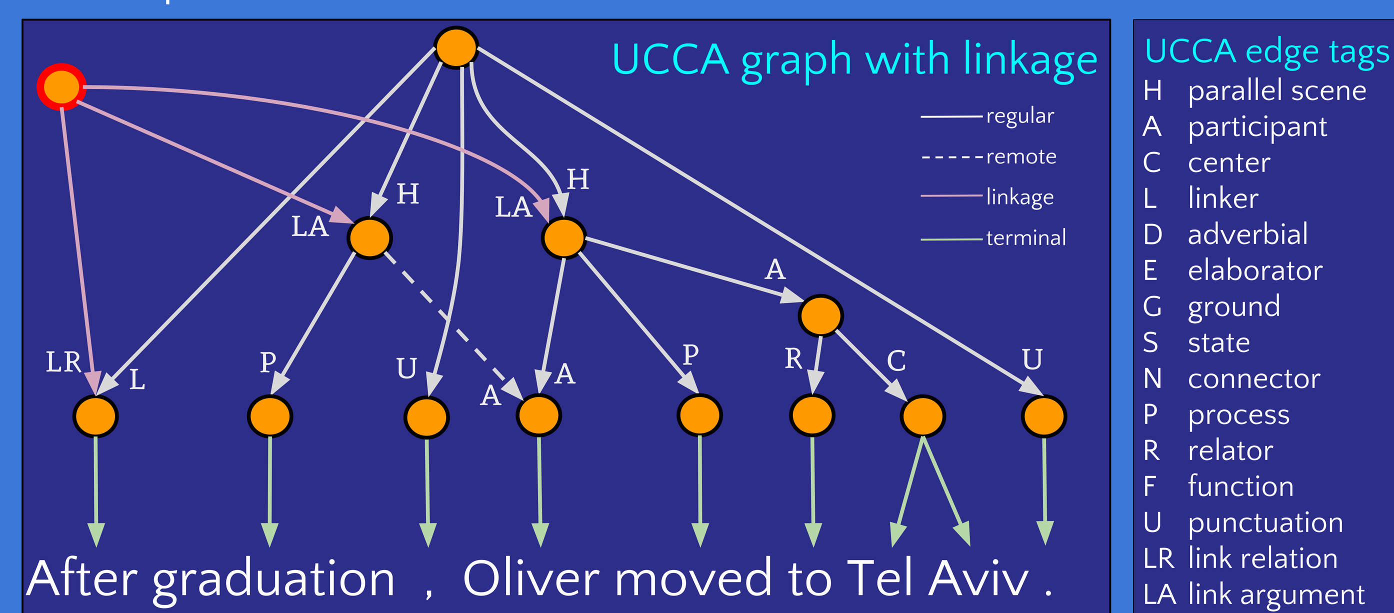
DAG (directed acyclic graph) parsers [6,8] are not limited to trees, but do not allow discontinuities. Some parsers can handle discontinuous constituents [4] but they can still only parse trees.

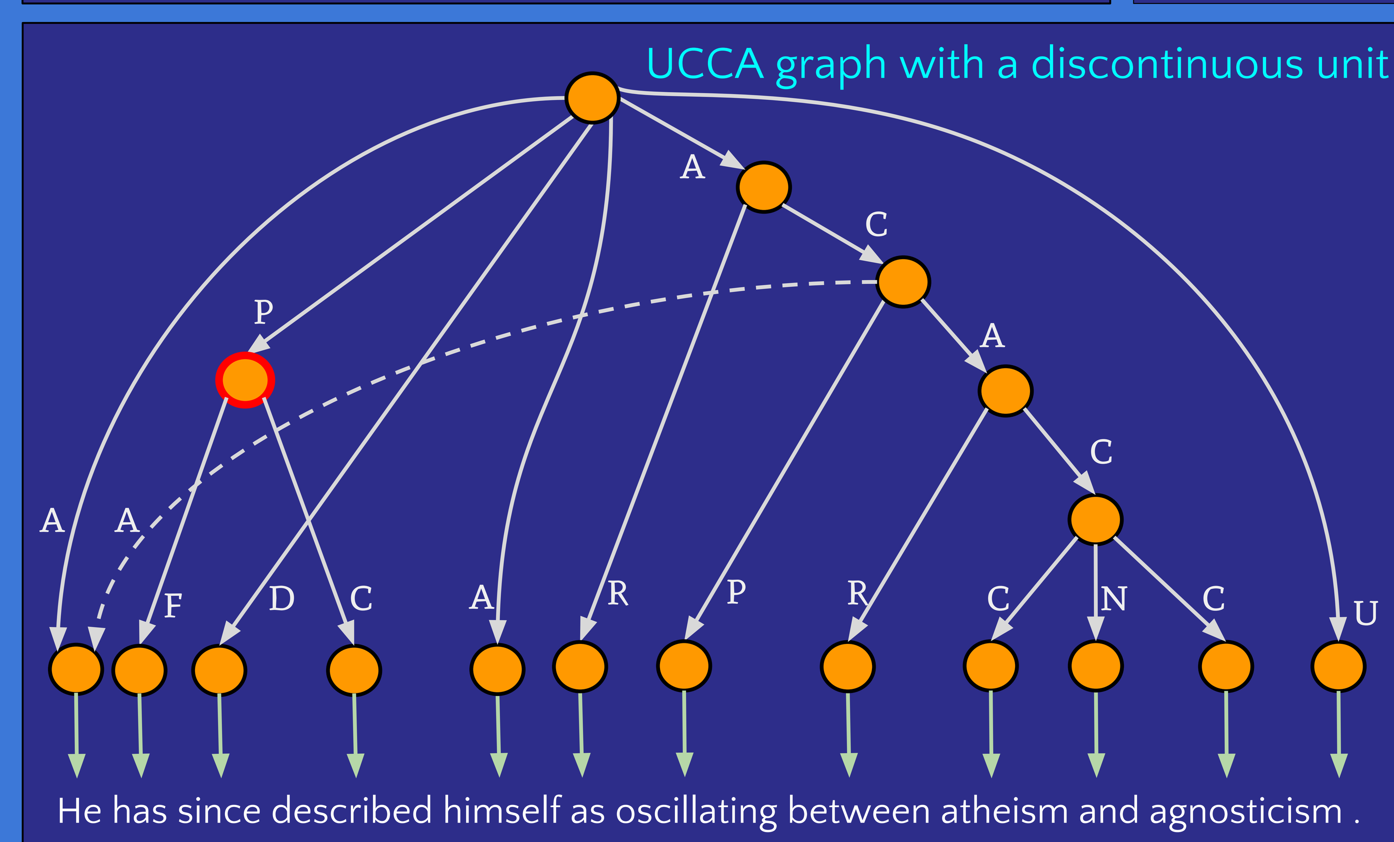
No parser currently supports the most general structure of language. This is not a minor phenomenon, and it must be addressed by parsers.

UCCA

Universal Conceptual Cognitive Annotation (UCCA) [1,7] is a semantic grammatical scheme that represents the meaning of natural language directly, in the most general way. It is portable across domains and languages, extensible, intuitive and supported by typological theories.

A corpus of 160K tokens from English Wikipedia was annotated by non-experts and is available online[1].



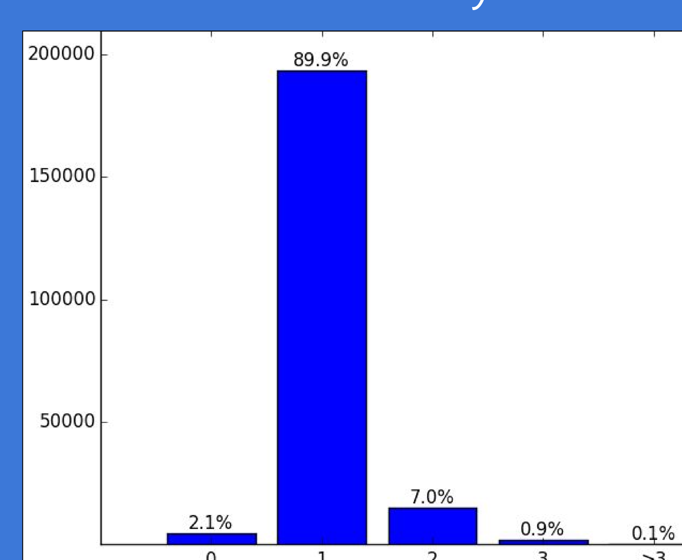


Corpus Statistics

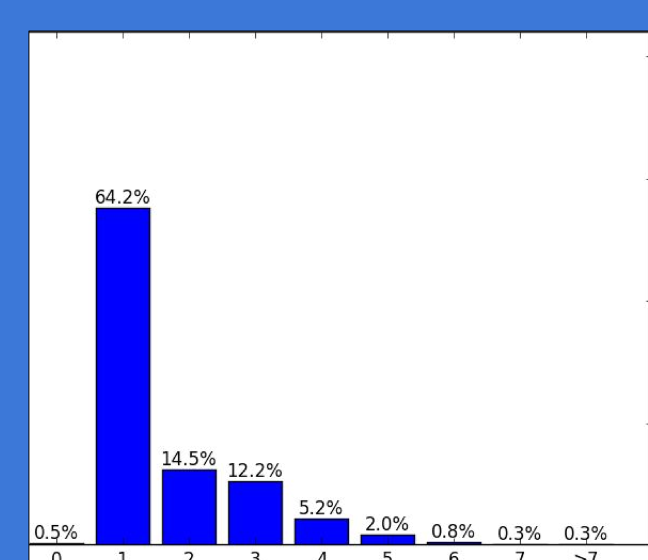
In the UCCA corpus, 57% of all nodes are non-terminal nodes.

Out of all non-terminal nodes, 8% have multiple parents and 0.9% span a discontinuous set of tokens.

Counting the parents and children for all non-terminal non-root nodes reveals the variety in the data:



Parent count histogram



Child count histogram

Parser

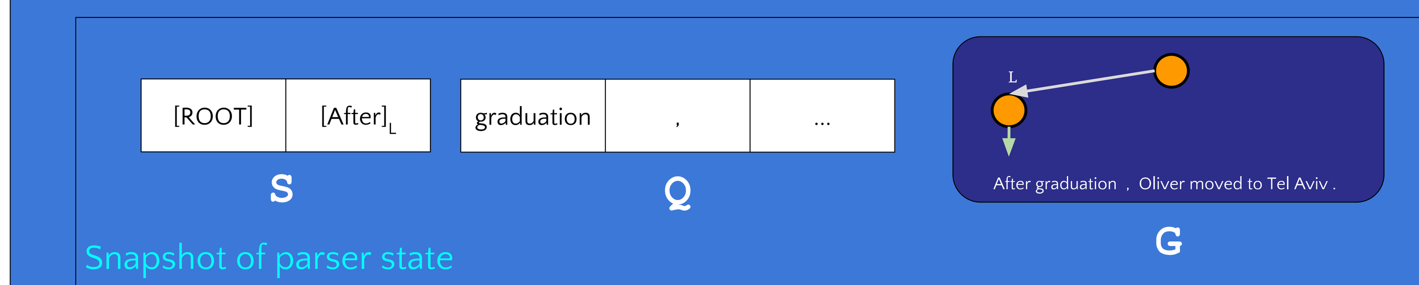
We present the first parser for UCCA: a novel transition-based system[2].

Transition-based (also called shift-reduce) parsers build the graph structure incrementally, maintaining the following data structures:

`Q:` Queue of nodes to process, initialized to the list of tokens.

`S:` Stack of nodes being processed, initially containing just the root.

`G:` Graph of already constructed nodes and edges.



Snapshot of parser state

The parser advances incrementally by predicting and applying actions:

- `Shift` moves one node from the queue to the top of the stack.
- `Reduce` discards the node at the top of the stack.
- `Left/Right-Edge`$_X$ create an X-edge between the top two stack elements.
- `Left/Right-Remote`$_X$ are the same, but they create remote edges.
- `Node`$_X$ creates a parent for the top of the stack, with an X-edge.
- `Implicit`$_X$ creates an implicit child for the top of the stack, with an X-edge.
- `Swap` places the second stack item back on the queue.
- `Finish` ends the parse and returns the constructed graph.

Actions are learned from the corpus using a structured perceptron.

Results

State-of-the-art dependency parsers pose a strong baseline. We converted UCCA to dependency annotation for running them, by removing remote and linkage edges (leaving at most one parent for each node), and omitting non-terminal nodes (leaving the tokens only). Performance is measured by F1 score on the graph's edges.

	regular edges		remote edges	
Parser	labeled	unlabeled	labeled	unlabeled
MaltParser [5]	0.589	0.782	0	0
LSTM Parser [3]	0.695	0.845	0	0
Our parser	0.296	0.531	0.026	0.061

Performance of parsers on the UCCA dataset

The baselines are incapable of producing the full general structure, and our parser should be able to surpass them with more tuning and improvements to the learning algorithm.

References

1. Abend, Omri, and Ari Rappoport. "Universal Conceptual Cognitive Annotation (UCCA)." ACL 2013.
2. Banarescu et al. "Abstract Meaning Representation for Sembanking". ACL LAW & ID 2013.
3. Dyer, Chris, Miguel Ballesteros, Wang Ling, Austin Matthews and Noah A. Smith. "Transition-based Dependency Parsing with Stack Long Short-Term Memory." ACL 2015.
4. Maier, Wolfgang. "Discontinuous Incremental Shift-Reduce Parsing." ACL-IJCNLP 2015.
5. Nivre, Joakim, Johan Hall, and Jens Nilsson. "MaltParser: A Data-Driven Parser-Generator for Dependency Parsing." LREC 2006.
6. Sagae, Kenji, and Jun'ichi Tsujii. "Shift-reduce dependency DAG parsing." COLING 2008.
7. Sulem, Elior, Omri Abend and Ari Rappoport. "Conceptual Annotations Preserve Structure Across Translations: A French-English Case Study." ACL S2MT 2015.
8. Alper Tokgöz, Gülşen Eryiğit. "Transition-based Dependency DAG Parsing Using Dynamic Oracles." ACL-IJCNLP SRW 2015.

[1] UCCA resource page: www.cs.huji.ac.il/~oabend/ucca.html

[2] UCCA parser source code: github.com/danielhers/ucca