Semantic parsers

Ádám Kovács

TU Wien

adam.kovacs@tuwien.ac.at

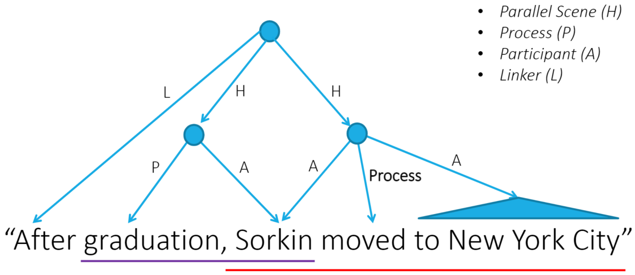
UCCA Abend and Rappoport (2013)

Universal Conceptual Cognitive Annotation (UCCA)

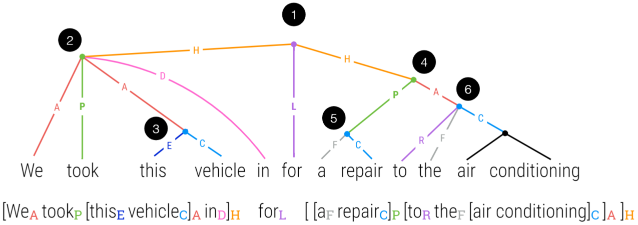
UCCA aims to to meet two goals

- Portability
  - The same set of categories can be applied accross different languages
- Stability
  - similar semantic structure is given to literal translations

UCCA scenes


L
H
H
P
A
A
Process
A
"After graduation, Sorkin moved to New York City"
Parallel Scene (H)
Process (P)
Participant (A)
Linker (L)

UCCA Foundational Layer



- Based on tokenized text
- Subset of tokens form units
- Units nest within larger units via edges labeled with categories
- At the top level, passage is segmented into units with Parallel Scenes (H) and (L)inkers

UCCA FL categories

Unit type:	**Superparallel unit**	**Scene unit**	**Sub-scene unit**	**Lexical unit**
Required elements	Parallel Scene (H)	Process xor State	Center	Token(s)
Optional elements	Linker	Participant (A), Adverbial (D), Time, Ground	{Elaborator, Quantity} xor Connector (N)	
	Function, Relator			
Legal parentage	root, A, E, C	A, E, C, H	any but F, R, root	any category

Secondary categories:
UNAnalyzable may be combined with any category in the table on a lexical unit;
Coordinated Main Relation (CMR) may occur with P or S

UCCA parsing

Three recent shared tasks

- SemEval 2019 shared task on UCCA parsing in English, French, German
- CoNLL 2019 and 2020 shared tasks on Cross-framework and Cross-Lingual Meaning Representation Parsing

Number of parsers available

- TUPA: Transition-based, multi-framework parser that serves as a baseline
- HLT@SUDA: won the SemEval task

UCCA applications

- Evaluation of text2text systems:
  - Semantic measure for Grammatical Error Correction
  - Semantic measure for (structural) text simplification
  - Human evaluation guided by semantic structure for MT
- Text simplification:
  - Text simplification using UCCA-based rules for preprocessing improves results
  - UCCA-guided simplification can also support MT in some settings
- UCCA-based machine translation and relation extraction

UCCA

- Abstracts away from much syntactic variation
- Demonstrated applicability to a number of languages
- Corpora and parsers available for a number of languages
- Already showing utility in evaluation of text2text systems and applications such as sentence simplification

UCCA extensions

- UCCA is a multi-layered structure
- The basic layer (the foundational layer) has a relatively flat structure
- Additional layers can capture semantic phenomena

Semantic roles

[Antoinette$_A$ drew$_P$ [a sheep]$_A$ [for the princess]$_A$ [in the desert]$_A$]

- FL does not distinguish Participants' roles
  - E.g., AGENT, THEME, CIRCUMSTANCE, PURPOSE, ...
- Expressed by various linguistic markers:
  - Word order [Mary$_A$ saw John$_A$] vs [John$_A$ saw Mary$_A$]
  - Case [**Er** sah [**den** Fuchs]] vs [**Ihn** sah [**der** Fuchs]]
  - Prepositions [The conquest [**of** Britain] [**by** the Romans]]

Semantic roles

- Several existing frameworks for role annotation: FrameNet, VerbNet, PropBank
- They chose SNACS for its independence of any one language or lexicon: https://arxiv.org/abs/1704.02134, https://aclanthology.org/P18-1018.pdf
  - 50 hierarchical categories
  - Designed to disambiguate prepositions and case

UCCA resources

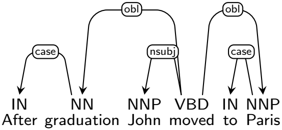
corpus	train/trial sentences	train/trial tokens	dev sentences	dev tokens	test sentences	test tokens	total passages	total sentences	total tokens
English-Wiki	4,113	124,935	514	17,784	515	15,854	367	5,142	158,573
English-20K	0	0	0	0	492	12,574	154	492	12,574
French-20K	15	618	238	6,374	239	5,962	154	492	12,954
German-20K	5,211	119,872	651	12,334	652	12,325	367	6,514	144,531

Table 2: Data splits of the corpora used for the shared task.

- Good coverage on multiple languages
- Shared tasks (SemEval, CoNLL)
- Multi layered, already connects to ontology
- Available parsers (e.g. tupa), not that resource hungry
- Tutorials at big conferences such as COLING
- But didn't really seen anything from them since 2020 (no commit, fresh models, etc..)
- Domain dependence wasn't really measured

Dependency parsing

Universal Dependency parsing



Universal dependency graph (UD)

- Syntactic UD parsing has seen great advances recently
- The UD project: https://universaldependencies.org/
- open community effort with over 300 contributors producing nearly 200 treebanks in over 100 languages
- Same format for all languages
- Active community, great models and industry ready parsers (stanza, spacy)
- Good for basically all domains (never had issues in legal, techical, medical domain for us)

Semantic Dependency Parsing

- Semantic analysis cannot be limited to tree structures
- E.g. node will often be the argument of multiple predicates
- Task 18 at SemEval 2015, Broad-Coverage Semantic Dependency Parsing (SDP 2015)
- Direct analysis of Who did What to Whom?
- SDP target representations aim to be task- and domain-independent.
- node re-entrancies, partial connectivity, higher degree of non-projectivity

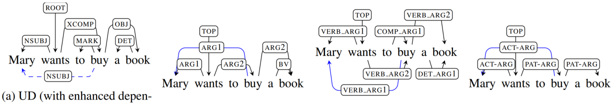
Data and languages

- Resources for English, Czech and Chinese
- Wall Street Journal (WSJ) and Brown segments of the Penn Treebank (PTB; Marcus et al., 1993) for English
- Around 30 000 sentences for English, Czech and Chinese

Representations

- DM: DELPH-IN MRS-Derived Bi-Lexical Dependencies
- PAS: Enju Predicate Argument Structures
- PSD: Prague Semantic Dependencies

Representations



(a) UD (with enhanced dependencies dashed) (b) DM (c) PAS (d) PSD

Figure 1: Comparison between syntactic and semantic dependency schemes

- PSD appearing linguistically most fine-grained
- PAS showing the smallest label inventory
- DM recognizes mwe-s better

Parsers

- Good easy to use parsers: https://github.com/yzhangcs/parser
- Good papers about the parsing: https://aclanthology.org/P18-2077.pdf
- Not many papers since 2018

AMR

AMR Banarescu et al. (2013)

- Abstract Meaning Representation (AMR) Annotation Release 3.0 (2020): https://catalog.ldc.upenn.edu/LDC2020T02
- AMR utilizes PropBank frames, non-core semantic roles, within-sentence coreference, named entity annotation, modality, negation, questions, quantities, and temporal expressions.
- Especially good for disambiguation:
  - run.01 , operate, proceed, operate
  - run.02 , walk quickly, a course or contest, run/jog

AMR

```
# ::id sdl_0002.1 ::date 2013-07-02T03:00:13 ::annotator SDL-AMR-09 ::preferred
# ::snt Rockstrom said that when rich countries increase their consumption , they also accelerate the expl
# with the result that they emit more greenhouse gases .
# ::save-date Thu Jul 11, 2013 ::file sdl_0002_1.txt
(s / say-01
        :ARG0 (p / person :wiki "Johan_Rockstrom"
            :name (n / name :op1 "Rockstrom"))
        :ARG1 (a / accelerate-01
            :ARG0 (c / country
                    :mod (r / rich))
            :ARG1 (e / exploit-01
                    :ARG0 c
                    :ARG1 (r2 / resource
                        :mod (n2 / nation)
                        :poss (w / world)))
            :mod (a2 / also)
            :time (i / increase-01
                    :ARG0 c
                    :ARG1 (c2 / consume-01
                        :ARG0 c))
            :ARG0-of (c3 / cause-01
                    :ARG1 (e2 / emit-01
                        :ARG0 c
                        :ARG1 (g / gas
                                :mod (g2 / greenhouse)
                                :mod (m / more))))))
```

AMR data

Dataset	Training	Dev	Test	Totals
BOLT DF MT	1061	133	133	1327
Broadcast conversation	214	0	0	214
Weblog and WSJ	0	100	100	200
BOLT DF English	7379	210	229	7818
DEFT DF English	32915	0	0	32915
Aesop fables	49	0	0	49
Guidelines AMRs	970	0	0	970
LORELEI	4441	354	527	5322
2009 Open MT	204	0	0	204
Proxy reports	6603	826	823	8252
Weblog	866	0	0	866
Wikipedia	192	0	0	192
Xinhua MT	741	99	86	926
Totals	55635	1722	1898	59255

AMR models

- Parse (StoG) model_parse_xfm_bart_large gives an 83.7 SMATCH score with LDC2020T02.
- Generation (GtoS) generate_t5wtense gives a 54 BLEU with tense tags or 44 BLEU with untagged LDC2020T02.
- Parsing 100 sentences on CPU (i5-8400) took 20 minutes (20 seconds to parse ud)

Graphbrain

Graphbrain

Semantic Hypergraph (SH) model

- Well-known semantic parsers (UCCA, AMR) usually take advantage of Deep Learning models
- Graphbrain combines symbolic and ML approaches
- The work was performed in the context of a computational social science (CSS) research team
- It is a formal language representation
- They define a parser with
  - modern NLP ML based building blocks
  - and random forest classifier
  - simple search tree to parse NL to SH

From natural language to SH

- It is a hard task to annotate enough data to train a NL-SH parser
- Better infer SH from grammatically-enriched representation - UD
- Two staged approach
  - **alpha-stage**: a classifier that assigns a type to each token in a given sentence.
  - **beta-stage**: is a search tree-based algorithm that recursively applies rules to build SH from sequence of atoms
- ML only in alpha-stage, a trivial classification problem

alpha-stage

With the usage of spaCy:

- Segmentation
- Tokenization
- POS tags
- Dependency labels
- NER
- Generate features for each token
- spaCy model: ”*en_core_web_lg*”

Data

- fiction books
- non-fiction books
- news
- scientific articles
- Wikipedia articles
- Selected randomly 60 sentences from each (300 sentences)
- 6936 tokens in full for classification
- They were annotated manually for classification
- Trained a random-forest classifier

Features

- Initial feature set from dependency features - 33 features total
- Feature selection algorithm - Iterative ablation procedure
- The final feature set contained 5 feature: F5 = TAG, DEP, HDEP, HPOS, POS_AFTER
- Accuracy around 95%

beta-stage

- This stage transforms the sequence of atoms into a semantic hyperedge
- It follows the alpha-stage
- A bottom up process:
  - Aggregates the deeper structure into a more complex one
  - Recursively combines hyperedges into a final, well-formed hyperedge

Final SH

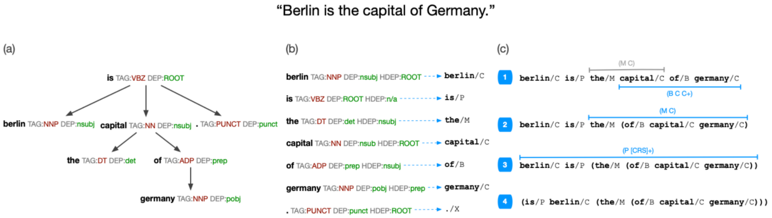


Figure 2: **(a)** Dependency parse tree with dependency labels (green) and fine grained part-of-speech tags (red). **(b)** α-stage classification of atom types. **(c)** β-stage structuring of sentence by iterative application of the patterns from table 2. A non-selected pattern is greyed-out.

SH concepts

Code	Type	Purpose	Example	Atom	Non-atom
C	concept	Define atomic concepts	apple/C	×	×
P	predicate	Build relations	(is/P berlin/C nice/C)	×	×
M	modifier	Modify a concept, predicate, modifier, trigger	(red/M shoes/C)	×	×
B	builder	Build concepts from concepts	(of/B capital/C germany/C)	×	
T	trigger	Build specifications	(in/T 1994/C)	×	
J	conjunction	Define sequences of concepts or relations	(and/J meat/C potatoes/C)	×	
R	relation	Express facts, statements, questions, orders,...	(is/P berlin/C nice/C)		×
S	specifier	Relation specification (e.g. condition, time,...)	(in/T 1976/C)		×

Table 1: Hyperedge types with use purposes and examples. Connector types are emphasized with a gray background. The rightmost columns specify whether this type may be encountered in atomic or non-atomic hyperedges.

Final SH results

Category	**Correct**	**Defect**	**Wrong**	**Total**	**Mean relative defect size**
Non-fiction	87 (.87)	8 (.08)	5 (.05)	100	.188
Wikipedia	81 (.81)	12 (.12)	7 (.07)	100	.190
News	77 (.77)	16 (.16)	7 (.07)	100	.147
Fiction	79 (.79)	5 (.05)	16 (.16)	100	.140
Science	71 (.71)	19 (.19)	10 (.10)	100	.290
All	**395 (.79)**	**60 (.12)**	**45 (.09)**	**500**	**.206**

Table 4: Global NL to SH parser evaluation.

Summary

- Lots of explainability in SH
- Only in English, but mapping to another language is not resource hungry (only annotated 300 sentences for English)
- The pattern language is very convenient and easy to use for IE tasks!!
- Very easy to manipulate the graphs with their own tools
- Recently held a seminar about the topic at the university: https://nlp.ec.tuwien.ac.at/seminar/

4lang

Background - 4lang

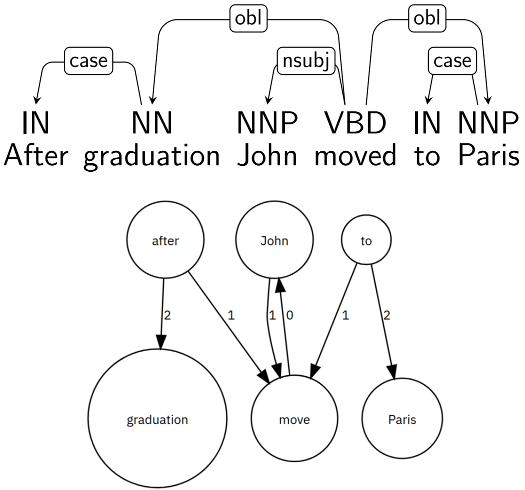


Figure 1: 4lang graph and UD parse of sentence "After graduation John moved to Paris"

4lang

Dependency	Edge
advcl advmod amod nmod nummod obl:npmod	$w_1 \xrightarrow{0} w_2$
nsubj csubj	$w_1 \underset{0}{\overset{1}{\rightleftharpoons}} w_2$
obj ccomp xcomp	$w_1 \xrightarrow{2} w_2$
appos	$w_1 \underset{0}{\overset{0}{\rightleftharpoons}} w_2$
nmod:poss	$w_2 \xleftarrow{1} \texttt{HAS} \xrightarrow{2} w_1$
nmod:npmod	$w_1 \xleftarrow{1} \texttt{NPMOD} \xrightarrow{2} w_2$
nmod:tmod obl:tmod	$w_1 \xleftarrow{1} \texttt{AT} \xrightarrow{2} w_2$
$w_1 \xrightarrow{\text{obl}} w_2 \xrightarrow{\text{case}} w_3$	$w_1 \xleftarrow{1} w_3 \xrightarrow{2} w_2$
$w_1 \xrightarrow{\text{acl:relcl}} w_2 \xrightarrow{\text{nsubj}} w_3$	$w_1 \xrightarrow{0} w_2$

Table 1: Mapping from UD relations to 4lang subgraphs.

4lang results

- 4lang is a set of rules from UD to semantic representation
- Used in multiple projects
- BRISE (Building Regulations Information for Submission Envolvement)[1] [2] is a smart city project of the City of Vienna
- Paper about the construction of semantic graphs for the task and rule-based classification in Recski et al. (2021)
- Lexical inference: Kovács et al. (2022)
- Offensive text detection Gémes and Recski (2021); Gémes et al. (2021)

[1] https://www.uia-initiative.eu/en/uia-cities/vienna-call4
[2] https://digitales.wien.gv.at/site/projekt/brisevienna/

A plus: Graphical Knowledge Representation (GKR)

- http://gkrparser.nlitoolkit.de/
- https://github.com/kkalouli/GKR_semantic_parser

Abend, O. and Rappoport, A. (2013). Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Gémes, K., Kovács, A., Reichel, M., and Recski, G. (2021). Offensive text detection on English Twitter with deep learning models and rule-based systems. to appear in FIRE 2021: Forum for Information Retrieval Evaluation.

Gémes, K. and Recski, G. (2021). TUW-Inf at GermEval2021: Rule-based and hybrid methods for detecting toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Workshop on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, pages 69–75, Heinrich Heine University Düsseldorf, Germany.

Kovács, Á., Gémes, K., Kornai, A., and Recski, G. (2022). Explainable lexical entailment with semantic graphs. *Natural Language Engineering*, page 1–24.

Recski, G., Lellmann, B., Kovács, A., and Hanbury, A. (2021). Explainable rule extraction via semantic graphs. In *Proceedings of the Fifth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2021)*, pages 24–35, São Paulo, Brazil. CEUR Workshop Proceedings.