Semantic parsing

Adam Kovacs
TU Wien

Evaluation criteria: System

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Parser	Parser (amrlib) is available under MIT license	Packages under MIT license: **Stanza**, **Spacy**	Open source parser available under MIT license, end-to-end	Open-source parser available under MIT license	Open-source parser/matcher/language
Domains	Domain free, (around 60k training data), No domain specific model	Domain free, has lots of domain specific models also	Generic domain, no domain specific data	20k sentences from wiki data, relatively domain free	Starts from UD, small data to converge to semantic representation
Modularity	all parts open, end-to-end	Open-source, industry ready, Layered parser	End-to-end	ucca structure is **layered**	layered-structure
Documentation	AMR and the parser are well documented	The UD project is well documented	Less documentation, only the papers	Tutorial at COLING with lots of details	Extensive documentation
Correct mistakes	End-to-end, with post processing	The UD models are great, layered, can correct segmentation, tokenization, etc..	End-to-end, with post-processing	Layered, but the foundation layer is a neural structure	Can correct UD-semantic conversion

Evaluation criteria: Living project

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Lasting support	AMR is very actively developed, most widespread parser in NLP	spaCy is developed by Explosion. They even tailor pipelines to specific needs	The parser is updated, but not much happened since 2019	The authors were very active, didn't find many things after the COLING tutorial. Parser is not updated	The package is developed since 2012, funded by ERC grant
Availability on GitHub	amrlib: https://github.com/bjascob/amrlib	spaCy: https://github.com/explosion/spaCy	SuPar: https://github.com/yzhangcs/parser	Tupa: https://github.com/danielhers/tupa	GraphBrain parser: https://github.com/graphbrain/graphbrain
Production readiness	all parts open, end-to-end	Spacy: "Industrial-Strength Natural Language Processing"	End-to-end	ucca structure is **layered**	layered-structure
Living community	AMR and the parser are well documented	The UD project is well documented	Less documentation, only the papers	Tutorial at COLING with lots of details	Extensive documentation

Evaluation criteria: Languages

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Language Support	Focused only on English. There are multilingual parsers, but proper training data only in English (recent progress towards Turkish)	Supports 80+ languages in good quality	English, Chinese and Czech	English, German, French, Hebrew, Russian. Also support cross-lingual parsing.	Only English, but easy to map to other languages (small data ML)

Evaluation criteria: Performance consumption

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Resources	Transformers based parser. CPU: ~20min/100 sen GPU: ~50sec/100 sen on RTX 2080TI	Very fast, 3 sec/100 sentences on CPU. GPU support available natively	LSTM based parser, 40 sec/100 sentences	LSTM based parser neural parser. 50 sec/100 sentences.	After UD only trivial computations, around 4 sec/100 sentences

Evaluation criteria: closeness to HyperKnow

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Resources	Pretty close with PropBank frames. Maybe can define a set of rules to convert from AMR structure.	Pretty far, but can be useful in other ways.	Basically enhanced SRL structure, the argument structure can be pretty useful	Does SRL, handles semantic phenomena. Layered, could be easy to expand. SRL not in parser.	Not close, can be useful for extracting information from text. SOTA in generating triplets from text

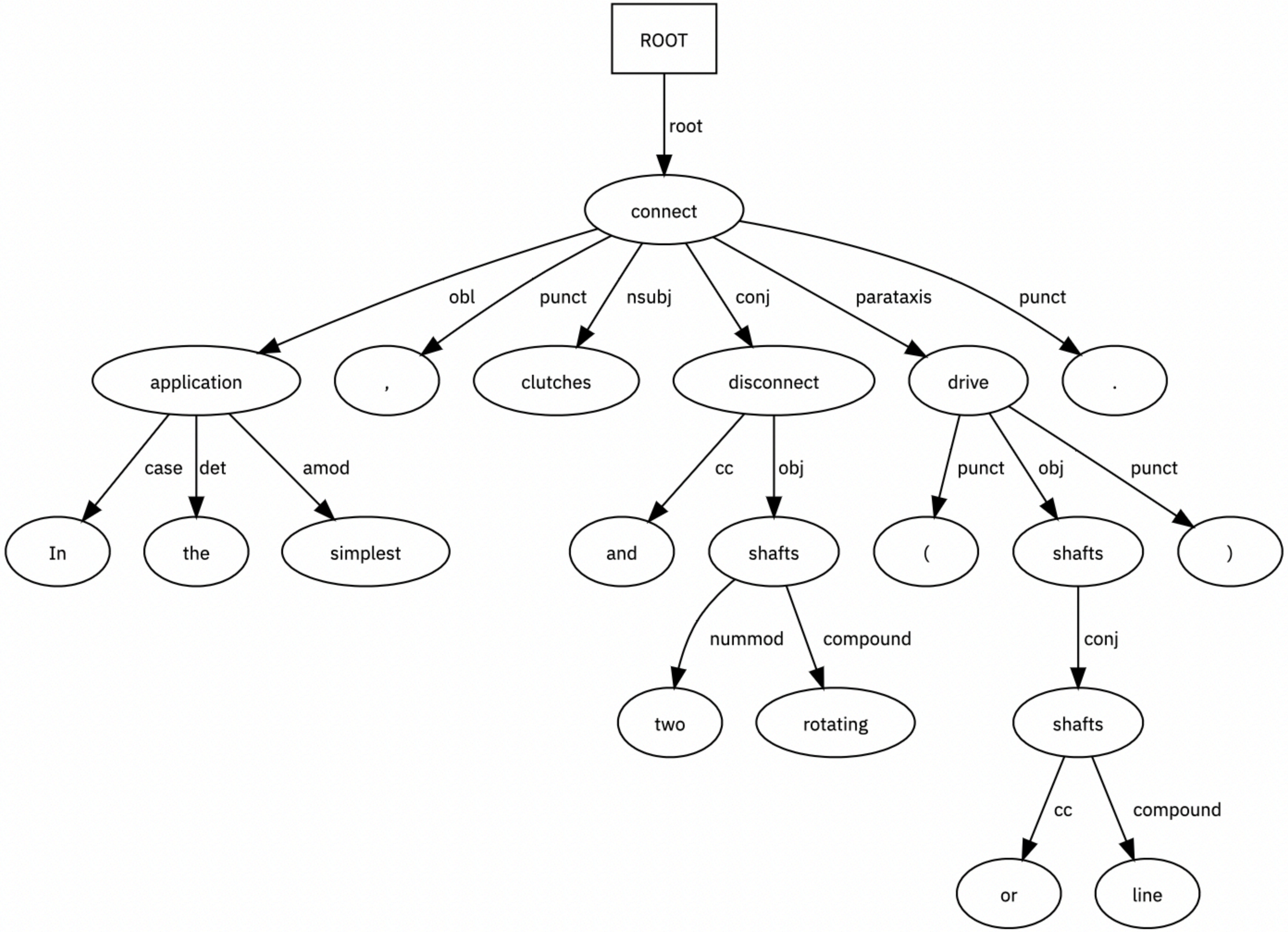
Evaluation criteria: Individual benefits

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Resources	Most all-in-on parser. Connects to PropBank frames, does NER, predicate disambiguation	Most versatile parser. SpaCy universe is huge, lots of very useful pipelines. Connect to wikidata, wordnet, dbpedia. Other tools: Holmes, negations, questions Other perks: domain specific models, tailored pipelines by the core developers, etc..	Recognizes whole structure instead of just verb predicates. Three parallel formalism for semantic dependencies.	Does SRL, connects to lexicon (SNACS). Cross-lingual mindset, layered structure, support for more languages	A complete hyper-graph representation, and best graph manipulation tools. Best for writing patterns over graphs (great pattern language framework)

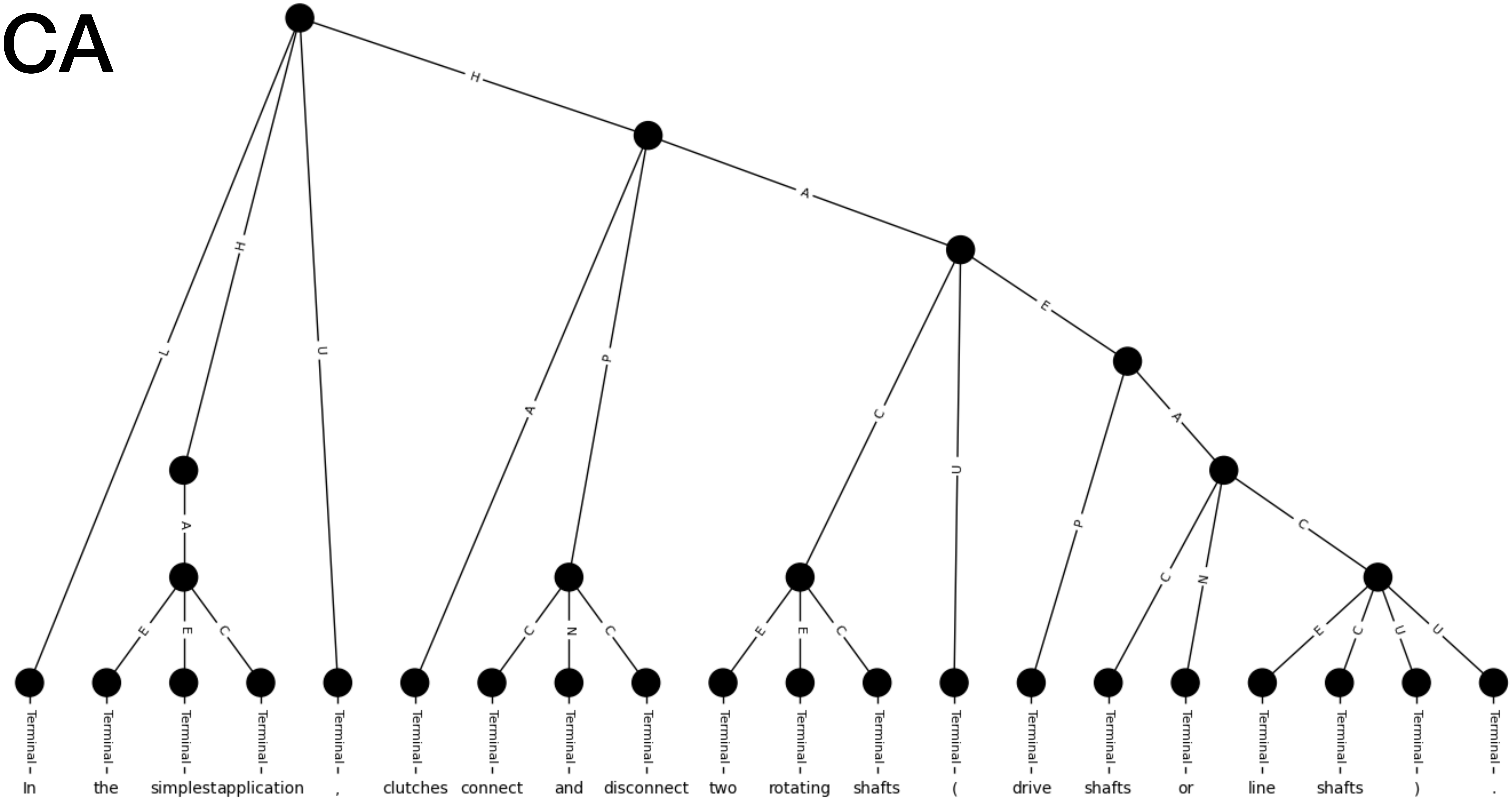
Semantic concepts

Criteria	AMR	UD	Semantic Dep parsing	UCCA	GraphBrain
Tokenization	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
NER	Approximately 80 named-entity types, including person, country, etc..	18 classes (cardinal, date, location, money, person, etc..)	Can be derived from UD	**Yes**, partly: Quantity, time, etc..	**Yes**, can be derived from UD
Semantic relationships	**Yes**, partly (e.g. Copular verbs)	**No**	**No**	**Yes**	**Yes**, partly, e.g. Relation specification (e.g. condition, time, …)
Semantic role labeling	**Yes** (e.g. :cause, :compared-to, :degree)	**No**	**Yes**	**Yes**	Has its own hyper edge types
NP chunking	Can detect compounds	Not directly	Not directly	**Yes**	**Yes**
Conjunctions	**Yes**	**Not directly, but able to get coordinates**	**No**	**Yes**	**Yes**
Comparators	**Yes**	**No**	**No**	**No**	**No**
Coreference Resolution	**Yes**	**No**	**No**	**Yes**	**Yes**
Detect questions	**Yes**	**Not directly**	**No**	**No**	**No**
Nominalization	**No**	**No**	**No**	**No**	**No**

UD

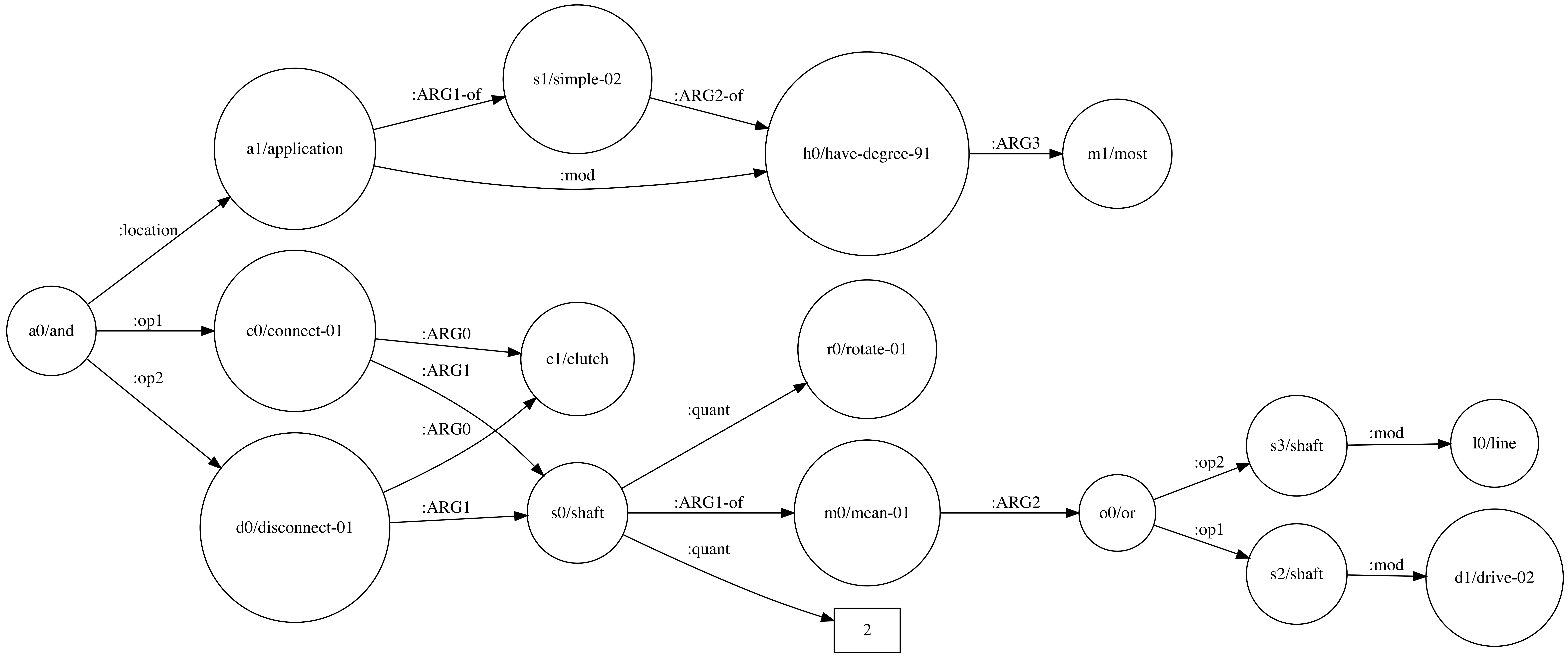

ROOT
root
connect
obl
punct
nsubj
conj
parataxis
punct
application
,
clutches
disconnect
drive
.
case
det
amod
cc
obj
punct
obj
punct
In
the
simplest
and
shafts
(
shafts
)
nummod
compound
conj
two
rotating
shafts
cc
compound
or
line

UCCA

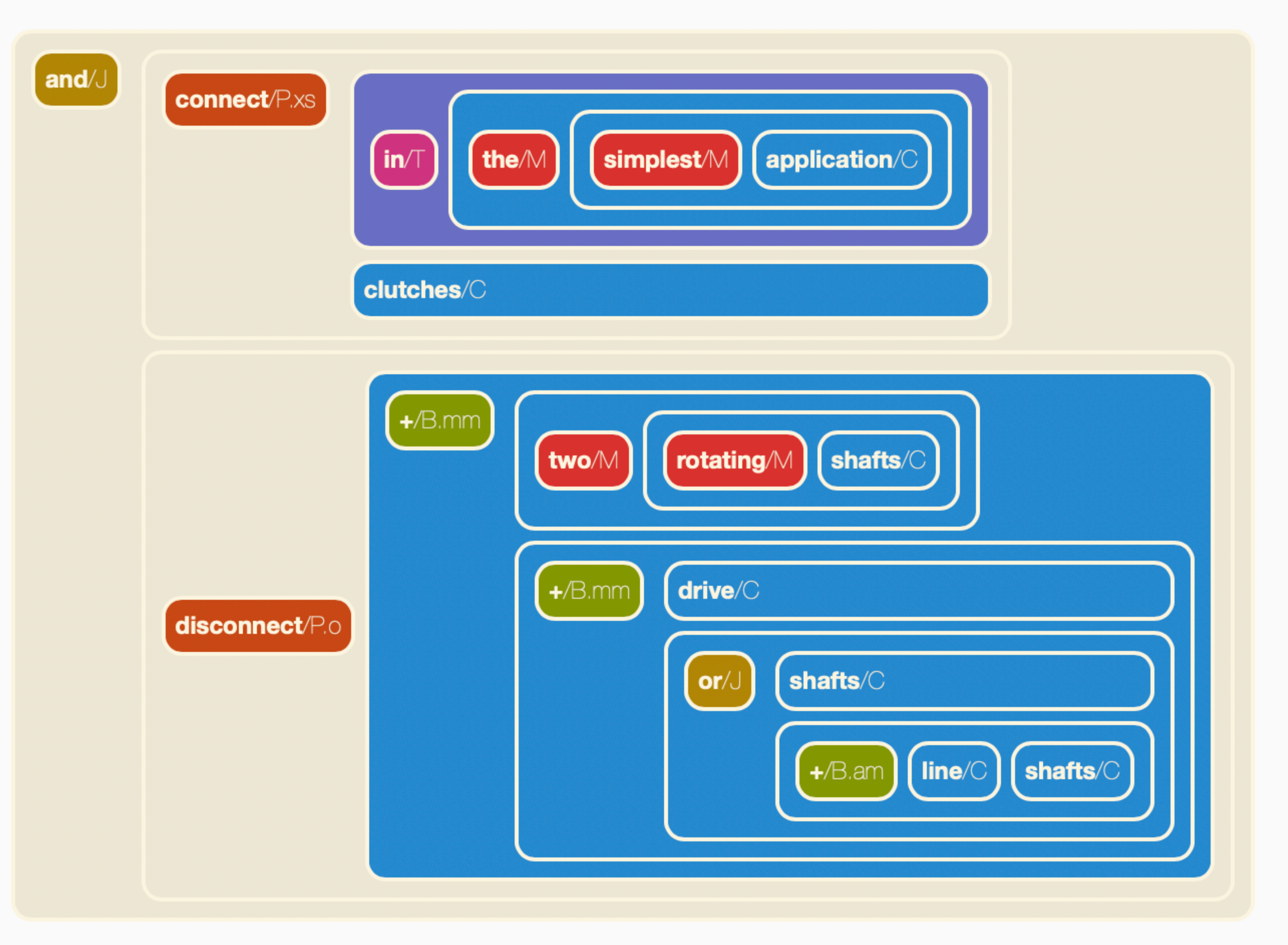


H
L
H
U
A
P
A
A
E
C
U
A
E
E
C
C
N
C
E
E
C
P
C
N
C
E
C
U
U
Terminal
In
the
simplest
application
,
clutches
connect
and
disconnect
two
rotating
shafts
(
drive
shafts
or
line
shafts
)
.

AMR


a0/and
:location
:op1
:op2
a1/application
:ARG1-of
s1/simple-02
:ARG2-of
h0/have-degree-91
:ARG3
m1/most
:mod
c0/connect-01
:ARG0
:ARG1
c1/clutch
d0/disconnect-01
:ARG0
:ARG1
s0/shaft
:quant
r0/rotate-01
:ARG1-of
m0/mean-01
:quant
2
:ARG2
o0/or
:op2
:op1
s3/shaft
:mod
l0/line
s2/shaft
:mod
d1/drive-02

GraphBrain (semantic hypergraph)



A Semantic Role Labeler

- CONLL 2012 dataset: https://paperswithcode.com/dataset/conll-2012-1
- https://aclanthology.org/W12-4501.pdf
- https://github.com/Riccorl/transformer-srl

Semantic role labeler

- Implementation of Shi et al 2019

- Trained on PropBank

- Demo with Allennlp

- E.g.: *[**ARGM-LOC**: In the simplest application] , [**ARG0**: clutches] [**connect.01**: connect]* and disconnect *[**ARG1**: two rotating shafts (drive shafts or line shafts)] .*

- Doesn’t cover variety of semantic phenomena— for example negation and other scopal embedding, comparatives, possessives, various types of modification, and even conjunction—typically remain unanalyzed in SRL

Semantic role labeling on VerbAtlas

- Cross-lingual Semantic Role Labeler trained on VerbAtlas
- Open models
- *[**LOCATION**: In the simplest application] , [**AGENT**: clutches] [**JOIN-CONNECT**: connect]* and disconnect *[**PATIENT**: two rotating shafts (drive shafts or line shafts)]*
- *[**argM-loc**: In the simplest application] , [**arg0-cau**: clutches] [**connectar.a2**: connect]* and disconnect *[**arg1-pat**: two rotating shafts (drive shafts or line shafts)]*

Resources

- spaCy link to Wikipedia: https://spacy.io/universe/project/spacyopentapioca
- Nominalization can be done with Wordnet
- UCCA SNACS: https://aclanthology.org/W19-3316.pdf
- Spacy also have a rule-based matcher: https://spacy.io/usage/rule-based-matching
- *ExtEnD*: Extractive Entity Disambiguation: https://github.com/SapienzaNLP/extend
- Few shot classification models: https://github.com/Pandora-Intelligence/classy-classification
- Few shot NER: https://github.com/Pandora-Intelligence/concise-concepts
- *Timexy*: A spaCy custom component that extracts and normalizes dates and other temporal expressions: https://github.com/paulrinckens/timexy
- ClausIE: a novel, clause-based approach to open information extraction, which extracts relations and their arguments from natural language text!: https://github.com/mmxgn/spacy-clausie
- Handling negation: https://spacy.io/universe/project/negspacy
- Knowledge extraction: https://github.com/erre-quadro/spikex
- Link to DBPedia: https://github.com/MartinoMensio/spacy-dbpedia-spotlight
- Link to Wordnet: https://spacy.io/universe/project/spacy-wordnet

Resources

- AMR:
  - Paper: https://aclanthology.org/W13-2322.pdf, Parser: https://amrlib.readthedocs.io/en/latest/
- Graphbrain:
  - Documentation: http://graphbrain.net, https://arxiv.org/abs/1908.10784, Parser: https://github.com/graphbrain/graphbrain
- UCCA
  - Tutorial: https://github.com/UniversalConceptualCognitiveAnnotation/tutorial, Parser: https://github.com/danielhers/tupa
- Semantic Dependency parsing
  - Shared-task: https://aclanthology.org/S15-2153/, Parser: https://github.com/yzhangcs/parser
- SRL:
  - Parser: https://github.com/Riccorl/transformer-srl, Paper: https://arxiv.org/abs/1904.05255