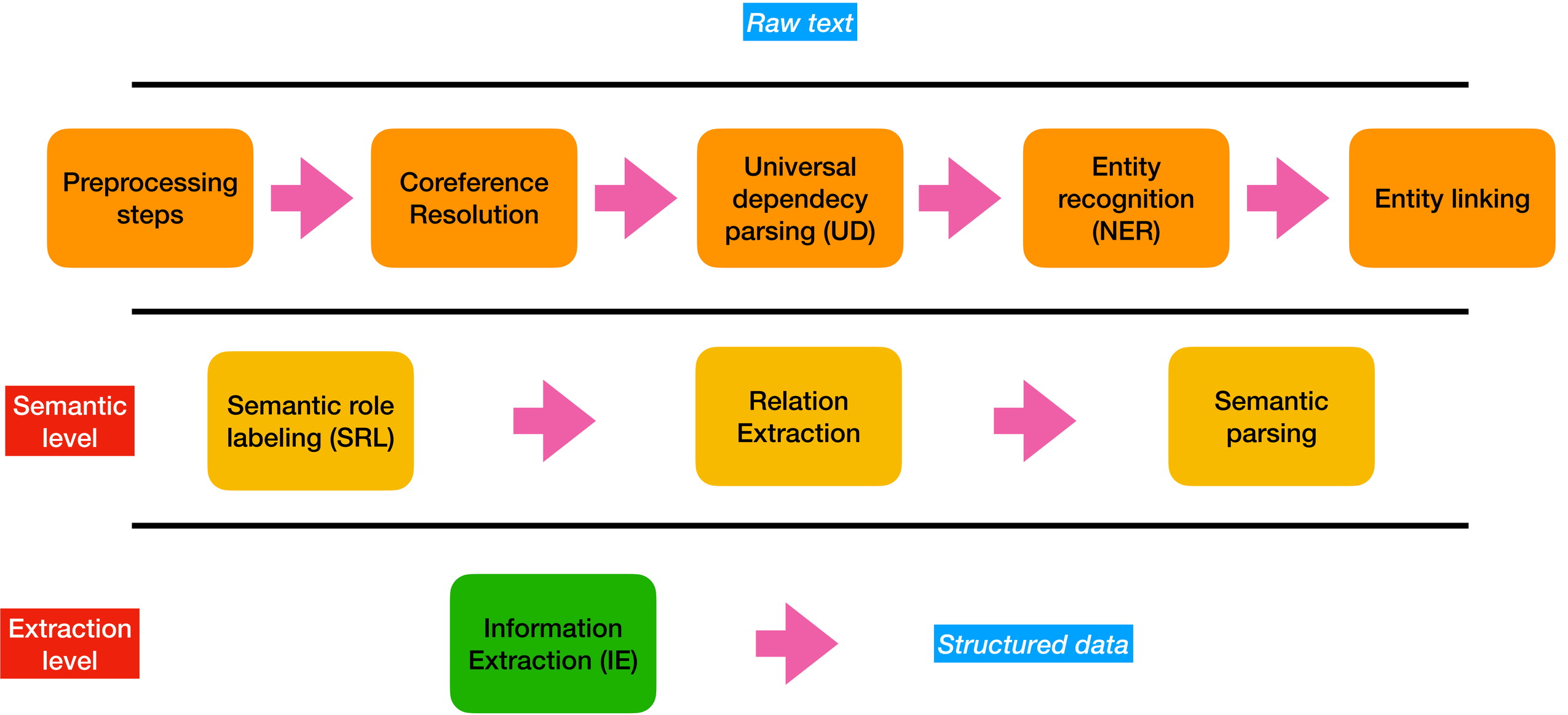
Semantic pipeline

Adam Kovacs

High-level pipeline


Raw text
Preprocessing steps
Coreference Resolution
Universal dependecy parsing (UD)
Entity recognition (NER)
Entity linking
Semantic level
Semantic role labeling (SRL)
Relation Extraction
Semantic parsing
Extraction level
Information Extraction (IE)
Structured data

Preprocessing steps

- Using the **spaCy** [1] module
- Very fast
- Production ready
- Open-source
- Tailored pipelines [2]

Functions:

- **Pipeline** approach
- Sentence segmentation
- Tokenization
- Lemmatization
- NP chunking
- Punctuation, stopwords

Coreference Resolution

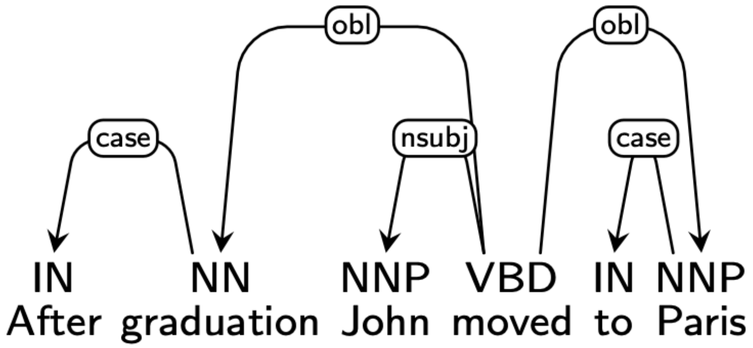
- **Coreferee** [1] is a spaCy pipeline, supports SOTA models in coreference tasks

- Open-source

- Both between sentences and within sentence:

Although he was very busy with his work, Peter had had enough of it. He and his wife decided they needed a holiday. They travelled to Spain because they loved the country very much.

Universal Dependency Parsing (UD)



Universal dependency graph (UD)

- Open-community effort
- Treebanks for 100+ languages
- Same format for all of the languages
- **spaCy** parser
- Coverage for various domains (legal, medical, technical, etc..)
- Well-established

Entity Recognition (NER)

- Available pipelines for more than 20+ languages
- Available named entities in the models:
  - CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART
- Can train spaCy model directly if there is available gold, annotated data
- Or use a few-shot learning approach: Concise-concept
- Use NLTK nombank for nominalization: [1]

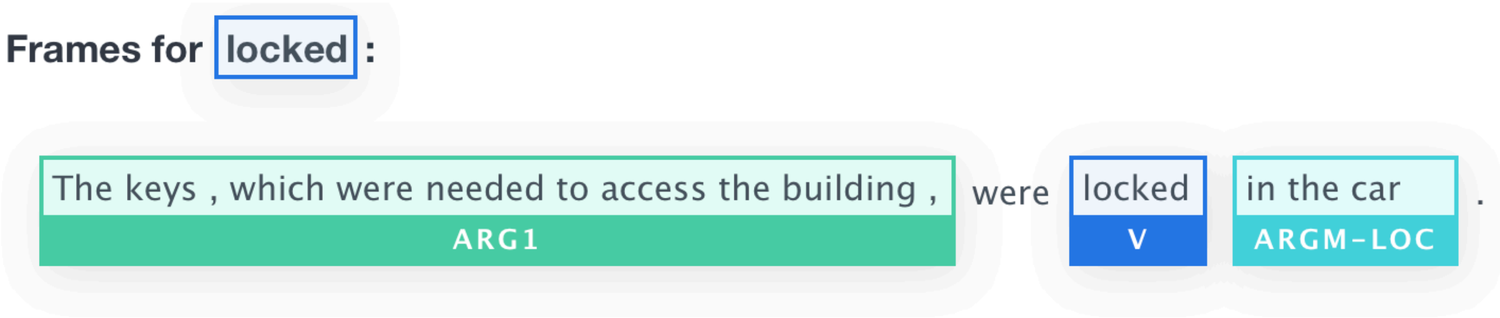
Entity linking

- spaCy entity linking to **Wikipedia**, **Wikidata**: [1], [2]; **Dbpedia spotlight**: [3]
- Or external package [3]
- Interesting paper in the topic: *“Named Entity Recognition for Entity Linking: What Works and What’s Next”*
- Example: “Google LLC is an American multinational technology company.”
  - Output: [('Google LLC', 'http://dbpedia.org/resource/Google', '0.9999999999999005'), ('American', 'http://dbpedia.org/resource/United_States', '0.9861264878996763')]

Entity linking

- **VerbAtlas** SRL parser (later) links to frames in PropBank
  - “The quick brown fox jumps over the lazy dog.” - Frame: jump.03
  - “Roleset id: jump.03 , physically or metaphorically leap, physical motion, “
- VerbNet semantic parser: https://github.com/jgung/verbnet-parser
  - Links to 329 verb classes

Semantic Role Labeling (SRL)



lock.01: attach, fastening, secured with a lock,

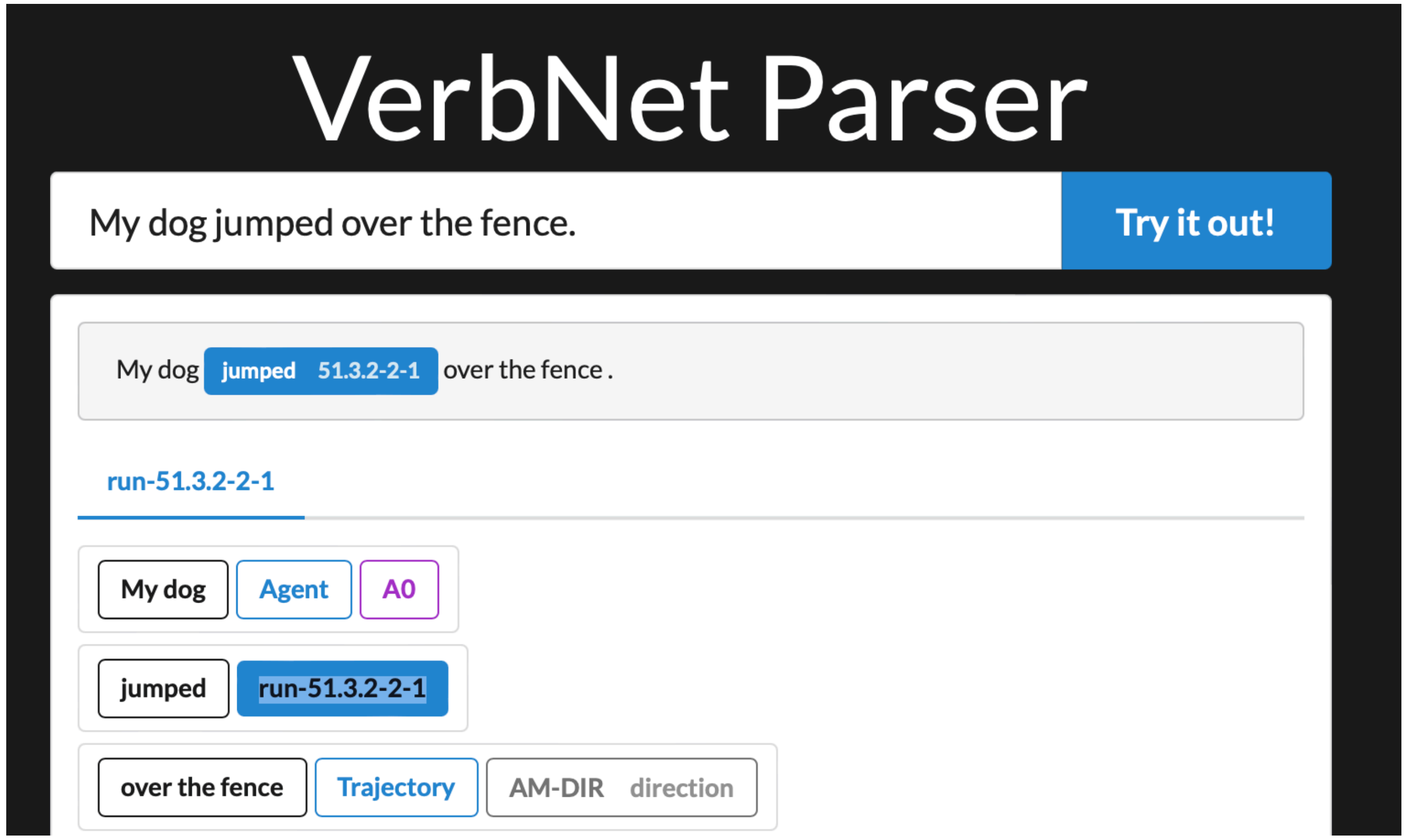
- **VerbAtlas** [1] : is a hand-crafted lexical-semantic resource whose goal is to bring together all verbal synsets from BabelNet into semantically-coherent frames.
- Both API and offline models (pretrained) are available, paper: [2]
- The semantic roles are fixed

Semantic roles in VerbAtlas

AGENT	ASSET	ATTRIBUTE	BENEFICIARY
CAUSE	CO_AGENT	CO_PATIENT	CO_THEME
DESTINATION	EXPERIENCER	EXTENT	GOAL
IDIOM	INSTRUMENT	LOCATION	MATERIAL
PATIENT	PRODUCT	PURPOSE	RECIPIENT
RESULT	SOURCE	STIMULUS	THEME
TIME	TOPIC	VALUE	

VerbNet: Cluster verbs

- Parser to disambiguate verbs into VerbNet categories: https://github.com/jgung/verbnet-parser



Go To Comments

run-51.3.2

Members: 97, Frames: 5

Post Comment

Class Hierarchy

- run-51.3.2*
  - run-51.3.2-1
  - run-51.3.2-2
    - run-51.3.2-2-1

Members

AMBLE (FN 1; WN 1; G 1)
AMBULATE (WN 1; G 1)
BACKPACK (WN 1)
BOLT (FN 1, 2, 3, 4; WN 4; G 1)
BOUND (FN 1; WN 1; G 1)
BREEZE
BUSTLE (WN 1)
CAPER (WN 1)
CAROM (WN 1, 2)
CAVORT (WN 1)
CHARGE (FN 1; WN 1, 4; G 5)
CHUNTER
CLAMBER (FN 1; WN 1; G 1)
CLIMB (FN 1, 2, 3; WN 1, 2; G 1, 3)
CLUMP (FN 1; WN 3; G 3)
CRAWL (FN 1; WN 1; G 1)
CREEP (FN 1; WN 1, 2; G 1)
DODDER (WN 1)
GOOSE_STEP (WN 1)
HIKE (FN 1; WN 2; G 1)
HITCHHIKE (WN 1)
HOPSCOTCH
JOUNCE
LIMP (FN 1; WN 1, 2)
LOLLOP (WN 1)
LUMBER (FN 1; WN 1)
LURCH (FN 1; WN 1, 2, 3; G 1, 2)
MEANDER (FN 1; WN 1)
MINCE (FN 1; WN 2; G 2)
MOSEY (FN 1; WN 1)
NIP
PACE (WN 1, 2)
POOTLE
POUNCE (WN 1)
PUSSYFOOT
REPAIR (WN 3; G 2)
SCUD (WN 1)
SCURRY (FN 1; WN 1)
SCUTTER
SCUTTLE (FN 1; WN 1; G 2)
SEESAW
SHAMBLE (WN 1)
SHUFFLE (FN 1, 2; WN 1; G 1)
SIDLE (FN 1; WN 1, 2)
SKEDADDLE (WN 1)
SKID
SKULK (FN 1; WN 3)
SLEEPWALK (FN 1; WN 1)
SLINK (FN 1; WN 1)
SLITHER (FN 1; WN 1)
SLOG (FN 1; WN 2)
SLOUCH (WN 2)
SOMERSAULT (WN 1)
SPRING (WN 1; G 2)
STUMP (WN 2; G 2)
SWAG (WN 2; G 2)
SWAGGER (FN 1; WN 1)
SWAN
SWERVE
TEAR (WN 3; G 2)
TIPTOE (FN 1; WN 1)
TODDLE (FN 1; WN 1)
TOIL
TOOTLE
TOTTER (FN 1; WN 1, 2, 3)
TREAD (WN 2; G 1)
TROOP (FN 1; WN 2)
TRUNDLE (FN 1; WN 1)
WADDLE (FN 1; WN 1)
WEAVE (FN 1, 2)
WEND
WHIZ

Semantic parsers

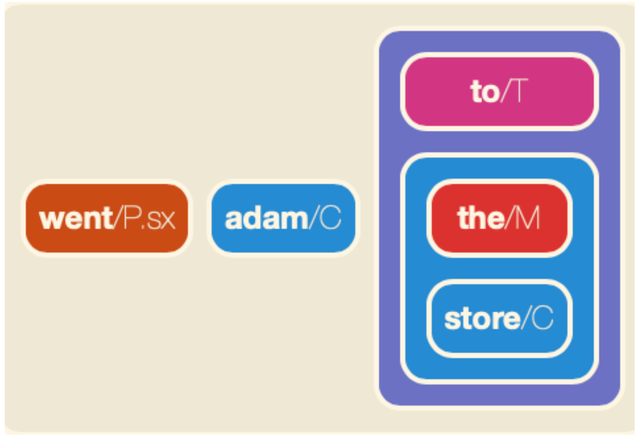
- Linking my previous presentations:
  - Possible choices for evaluation: https://github.com/adaamko/presentations/blob/main/semantic_parsers/parsers.ipynb
  - Analysis of semantic parsers: https://github.com/adaamko/presentations/blob/main/semantic_parsers/parsers.pdf
  - Evaluation of semantic parsers: https://github.com/adaamko/presentations/blob/main/semantic_parsers/semantic_parsing.pdf

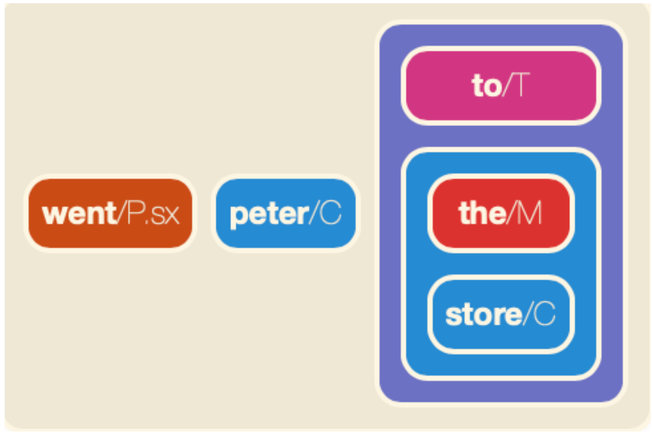
Semantic parsers

- To summarize, useful resources:
  - **AMR** - mainly for English; takes lots of resources to parse; open-source parser; can handle lots of semantic phenomena
  - **UCCA** - multi and cross-lingual; less-resource hungry;
  - **Semantic Dependency Parsing (SDP)** - similar to SRL but tries to parse the whole structure of the sentence not just verbs and arguments; multi-lingual parsers are available
  - **GraphBrain** - only English; converts UD graphs into a more semantic format; very good pattern language and repository; easy to extend with new functionalities

Handling conjunctions

- Multiple graph formalisms can handle that (e.g. AMR, GraphBrain):





Handling Semantic Relationships

- Called **Relation Extraction (RE)** in NLP
  - Relation extraction (RE) is the task of extracting semantic relationship between entities from a text
  - Usually between two or more entitites; Semantic categories (e.g. Destination, Component, Employed by, Founded by, etc..)
- Curated list on Github: [1]
- **Semeval** task: [2]
  - Cause-Effect, Instrument-Agency, Product-Producer, Content-Container, Entity-Origin, Entity-Destination, Component-Whole, Member-Collection, Message-Topic
- **Component-Whole:** my apartment has a large kitchen
- **Member-Collection:** there are many trees in the forest
- **Entity-Destination:** the boy went to bed

TACRED dataset

- TACRED is a large-scale relation extraction dataset with
- 106,264 examples built over newswire and web text
- Examples in TACRED cover 41 relation types or are labeled as no_relation if no defined relation is held.
- Example relations: '*org:founded_by*', '*no_relation*', '*per:identity*', '*org:alternate_names*', '*per:children*', '*per:origin*', '*per:countries_of_residence*', '*per:employee_of*', '*per:title*', '*org:city_of_branch*', '*per:religion*', *‘per:age’, etc..*

RE parsers

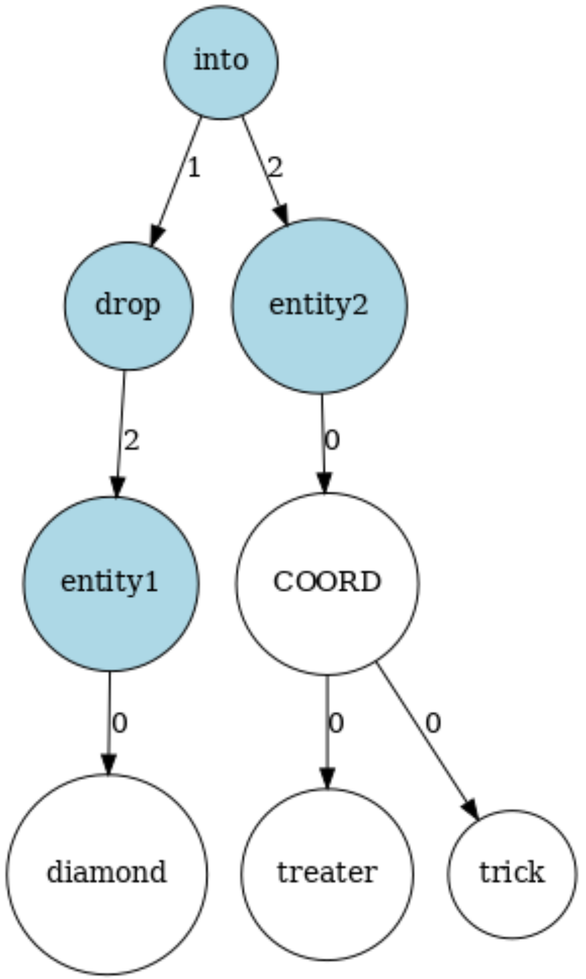
- **REBEL**: seq2seq model that simplifies Relation Extraction (EMNLP 2021). [Github]
- **OpenNRE** [Github]: Open-Source Package for Neural Relation Extraction (NRE)

Other relations

- Verb clustering: based on VerbNet
- NP chunking: built in component in spaCy
- Attribution: GraphBrain

Case-study for RE

- The diamond ring was dropped into a trick-or-treater's bag.
- Method:
  - Parse text into a 4lang graph
  - Use POTATO and the Semeval training data to discover rules for the Entity-Destination label
  - Details in our paper

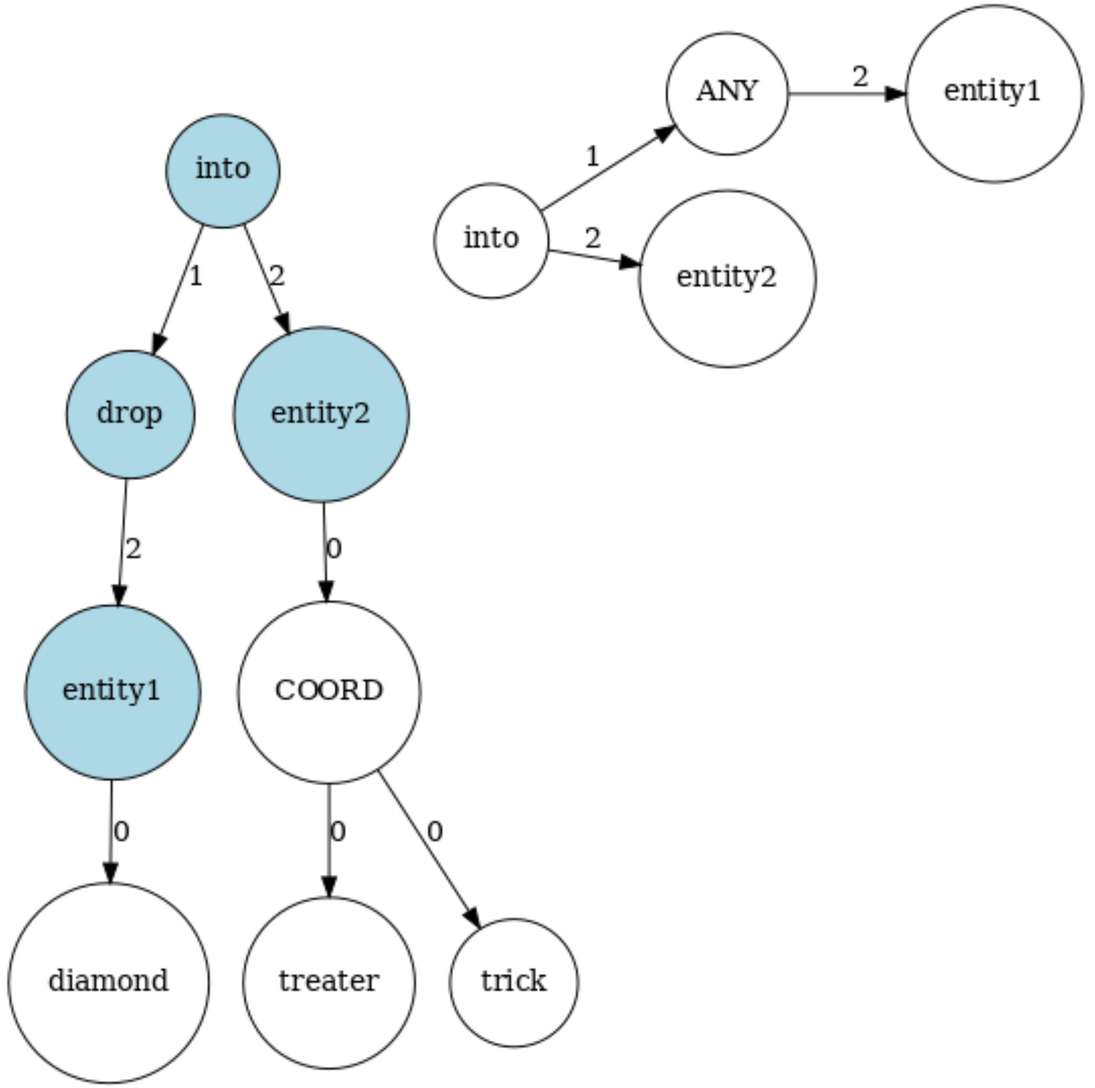


Case-study for RE

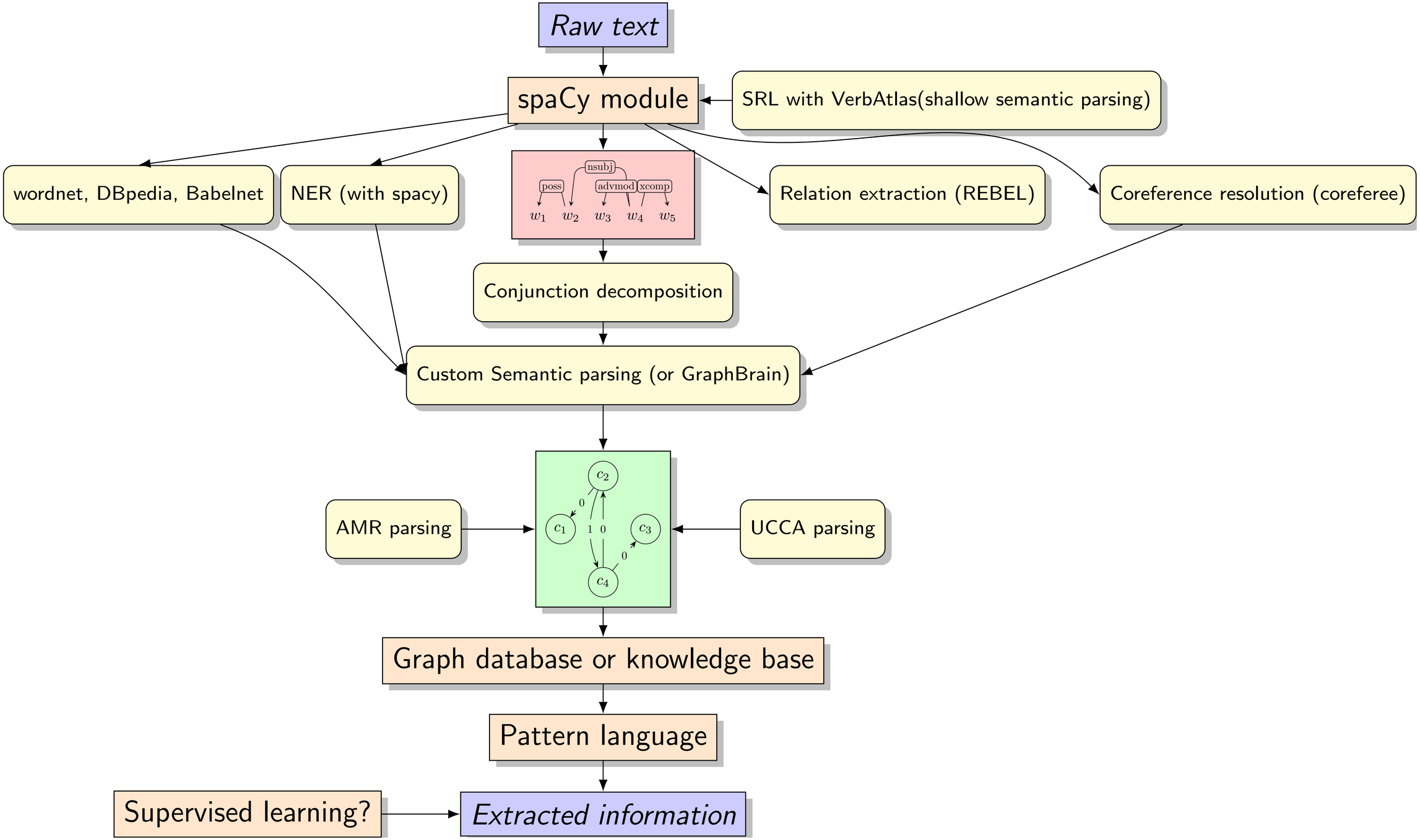
- The diamond ring was dropped into a trick-or-treater's bag.
- Method:
  - *(u_1 / into :2 (u_2 / entity2) :1 (u_3 / .* :2 (u_4 / entity1)))*

```
(u_1 / into
  :2 (u_2 / entity2)
  :1 (u_3 / misplace|invest|drop|import|transport|
            fetch|pack|insert|pour|implant|remove|
            leak|add|dump|release|stuff
       :2 (u_4 / entity1)))
```

  - This pattern now matches 138 true positives and only 1 false positive, achieving 99.3% precision on the training dataset. On the validation data the same pattern achieves 95.6% precision and 21.9% recall.



Suggested pipeline


Raw text
spaCy module
SRL with VerbAtlas(shallow semantic parsing)
wordnet, DBpedia, Babelnet
NER (with spacy)
nsubj
poss
advmod
xcomp
w_1 w_2 w_3 w_4 w_5
Relation extraction (REBEL)
Coreference resolution (coreferee)
Conjunction decomposition
Custom Semantic parsing (or GraphBrain)
AMR parsing
c_1 c_2 c_3 c_4
0 1 0 0
UCCA parsing
Graph database or knowledge base
Pattern language
Supervised learning?
Extracted information