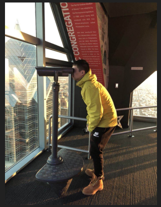
EACL 2021

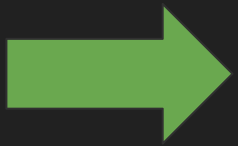
Process-Level Representation of Scientific Protocols with Interactive Annotation

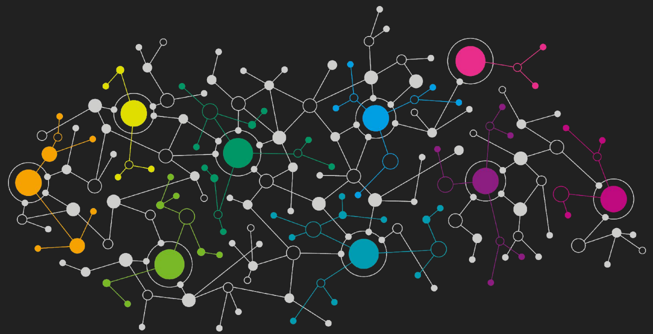
Ronen Tamari, Fan Bai, Alan Ritter, Gabriel Stanovsky

Motivation

- Scientific literature contains millions of unstructured natural language texts describing experimental material synthesis procedures ("recipes")
- Vast potential for data-driven research
- To date, methods have focused on text-mining applications





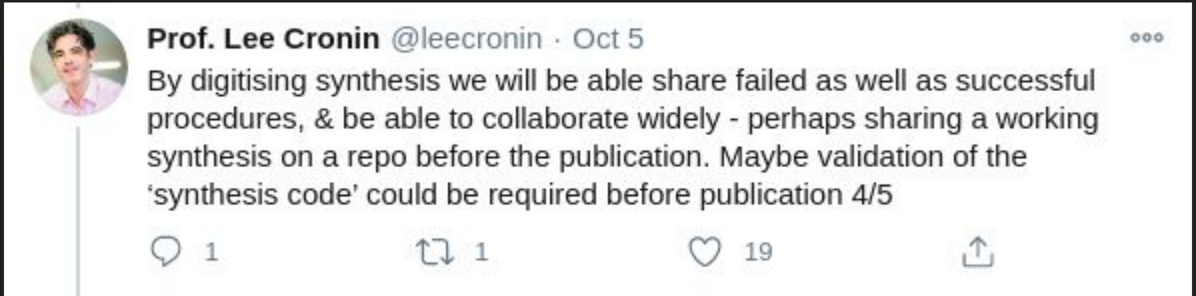
The next frontier: digital synthesis

- Serious reproduceability concerns have raised the bar
- The next frontier: convert textual descriptions to automated execution of experimental procedures!

Strateos cloud laboratory

The next frontier: digital synthesis

- Serious reproduceability concerns have raised the bar
- The next frontier: convert textual descriptions to automated execution of experimental procedures!



Text-to-experiment

- Objective: parse texts into structured format for integration into automated workflows
- Poses formidable challenges for current natural language understanding (NLU) systems:
  - Meaning representation (semantics)
  - Annotation
  - Modelling
  - Evaluation

Text-to-experiment

- Objective: parse texts into structured format for integration into automated workflows
- Poses formidable challenges for current natural language understanding (NLU) systems:
  - Meaning representation (semantics)
  - Annotation
  - Modelling
  - Evaluation

Why rethink semantics?

Can we just use text mining approaches?

“Although [text mining] approaches are useful for mining literature [...], we need a system that could output a machine-readable representation of a procedure with **sufficient process details to unambiguously execute the procedure on an automated platform**. This **goes beyond simply tagging chemical entities found in literature procedures**”

Can we just parse directly into robot instructions?

“Automated platforms from different companies or research groups all have bespoke instruction sets **with no obvious semantic link among them or to the literature**. This broken link has prevented the digitization of chemistry”

(Mehr et. al, 2020)

Why rethink semantics?

Can we just use text mining approaches?

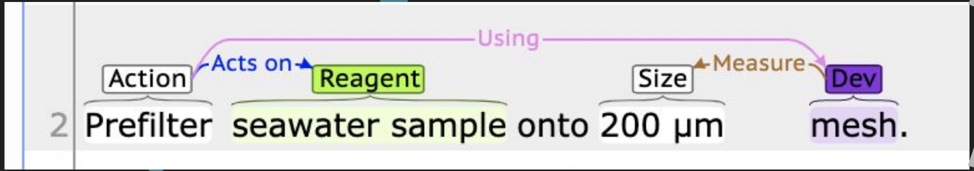
Hardware-agnostic representation as scaffold for execution

Can we just parse directly into robot instructions?

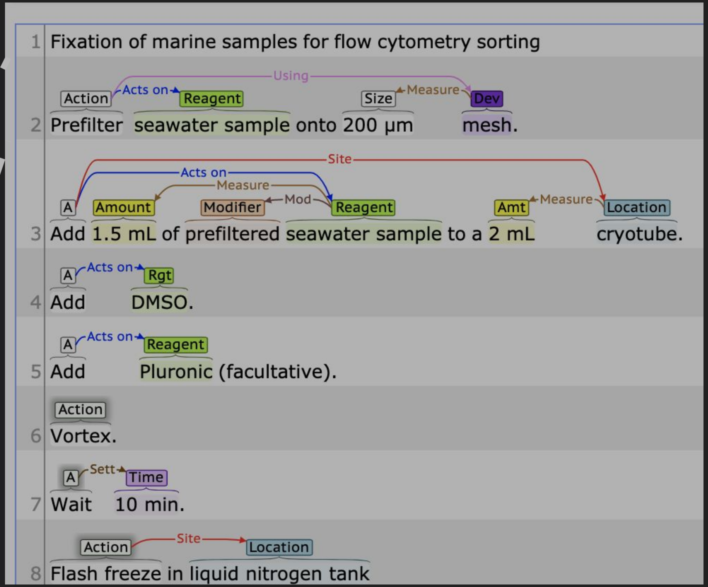
Text-mining approaches

> Action-graph semantics: shallow, sentence-level annotations
  - Biology: Wet Labs Protocols (WLP, Kulkarni et. al, 2018) - 622 documents
  - Materials science: Materials Science Procedural Text Corpus (MSPTC, Mysore et. al, 2019) - 230 documents

Edges = Relations



Nodes (text spans) = Entities

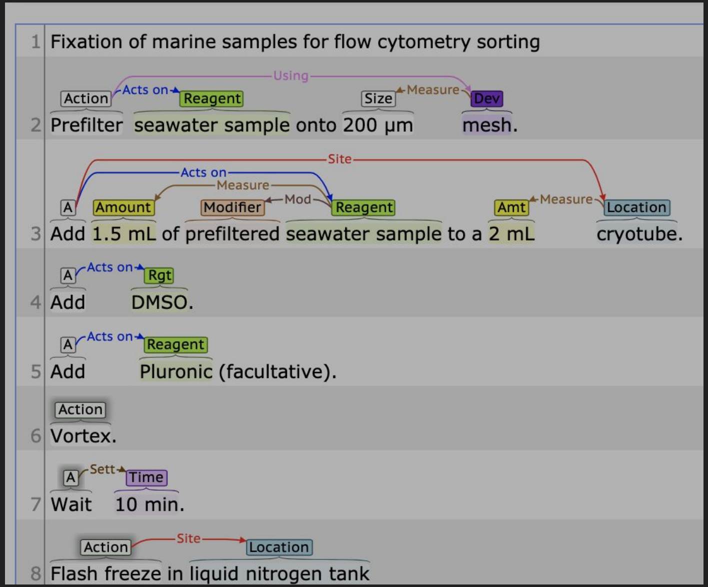


WLP instance

Text-mining approaches

- Action-graph semantics: shallow, sentence-level annotations
  - Biology: Wet Labs Protocols (WLP, Kulkarni et. al, 2018) - 622 documents
  - Materials science: Materials Science Procedural Text Corpus (MSPTC, Mysore et. al, 2019) - 230 documents

Complex texts, require expert & common-sense knowledge

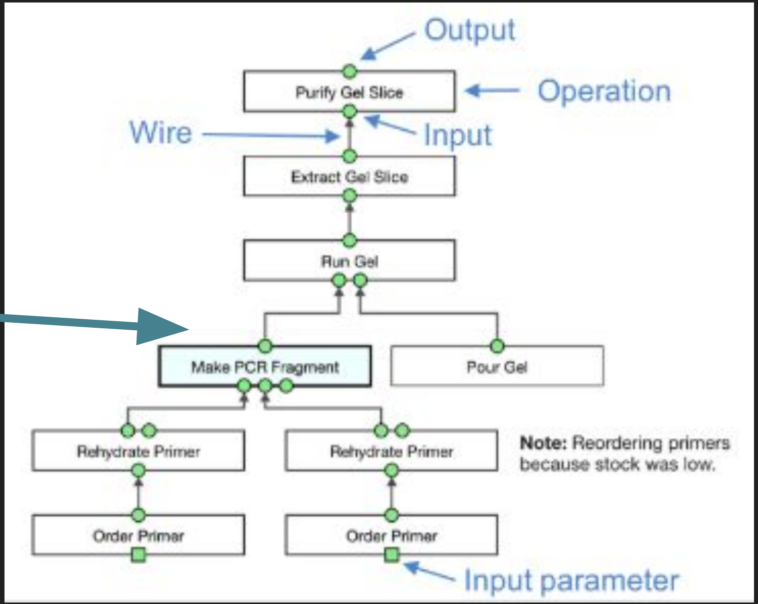


WLP instance

What's missing?

- Inspiration from executable biology protocols research
- Protocols as:
  - Computation graphs

Process-level representation: connected graph tracking operation inputs and outputs



Aquarium (Keller et. al, 2019)

What's missing?

> Inspiration from digital synthesis research
> Protocols as:
  - Computation graphs
  - Sequence of API calls

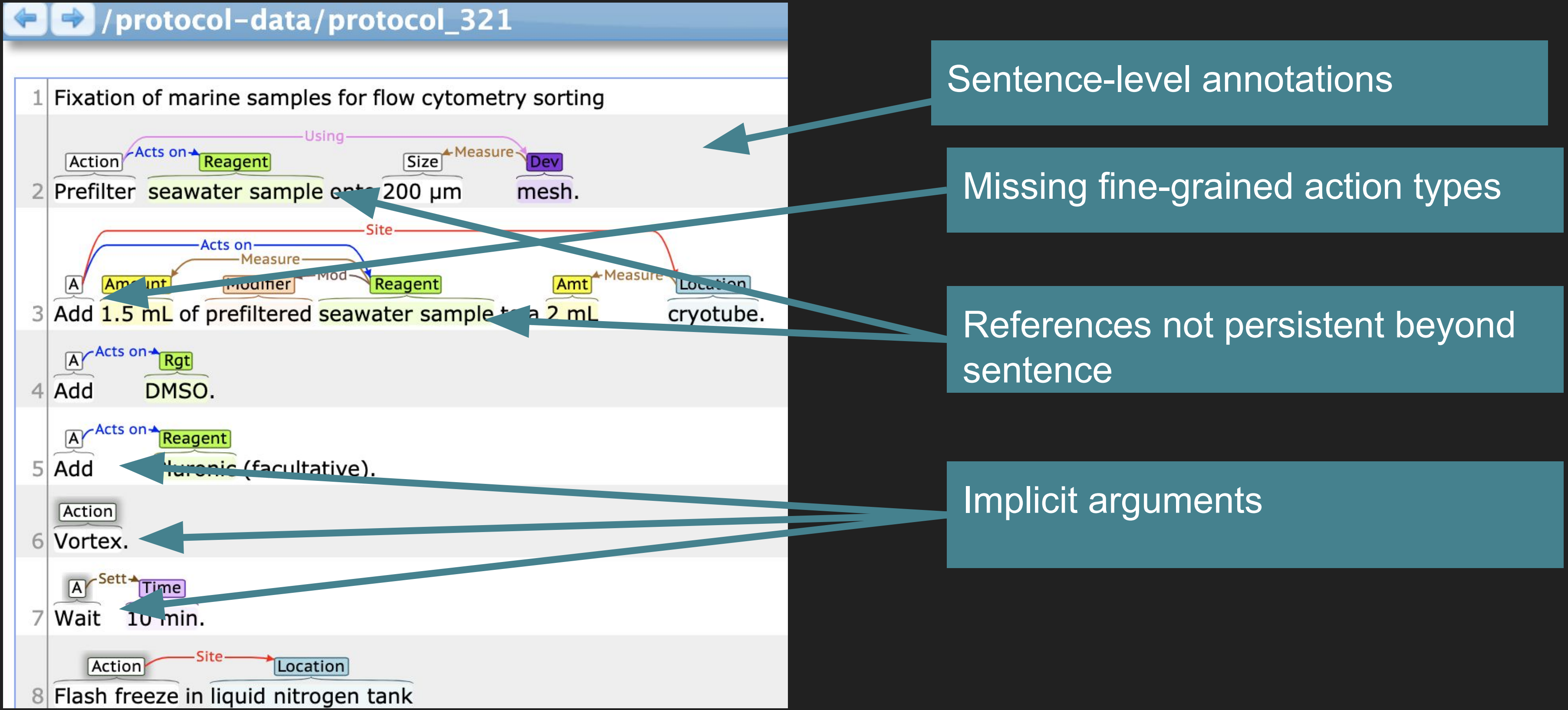
Typed operations

References as persistent objects rather than text spans

```
# transfer the first primer
protocol.transfer(params.primer_pairs[0][0], params.taq_tubes[0],
                  params.primer_vol_total)
# transfer the second primer and mix the solution
protocol.transfer(params.primer_pairs[0][1], params.taq_tubes[0],
                  params.primer_vol_total,
                  mix_after=True, mix_vol=params.primer_vol_total,
                  repetitions=5)
# distribute the master mix from the first Taq tube to the first
# quadrant of the PCR plate, mix before aspirating the solution
protocol.distribute(params.taq_tubes[0], dest_plate.quarter(0),
                    params.mm_vol,
                    allow_carryover=True, mix_before=True,
                    mix_vol=params.primer_vol_total, repetitions=4)

# add the DNA from the 96-well plate to the first quadrant and mix
protocol.stamp(params.dna_plate, dest_plate, 0, params.dna_vol,
               mix_after=True, mix_vol=params.dna_vol, repetitions=3)
# seal and thermocycle
protocol.seal(dest_plate)
protocol.thermocycle(dest_plate, [{
```

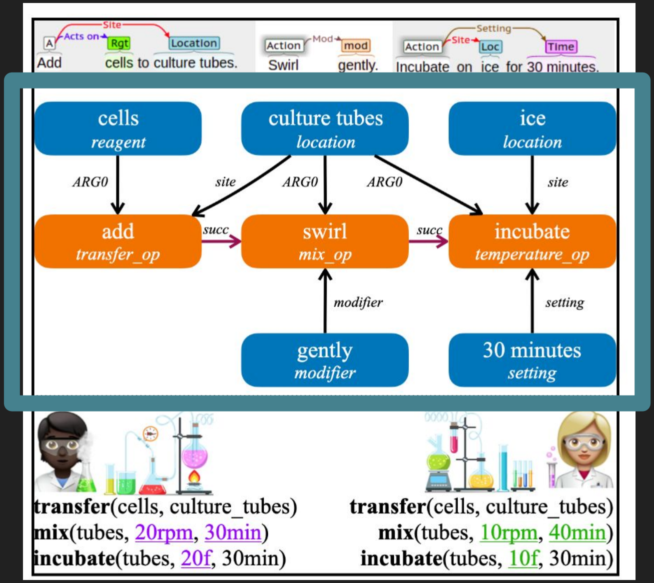
Autoprotocol (Lee & Miles, 2018)

Action-graphs: limitations



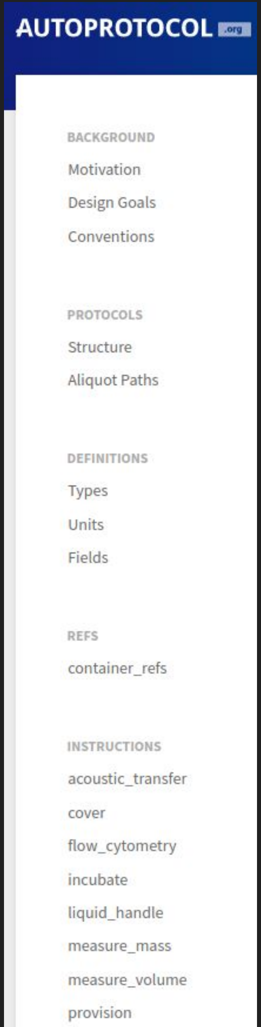
Our proposal: Process Execution Graphs (PEG)

- Process-level abstract executable representation
- Bridges between procedural text and downstream automated workflows



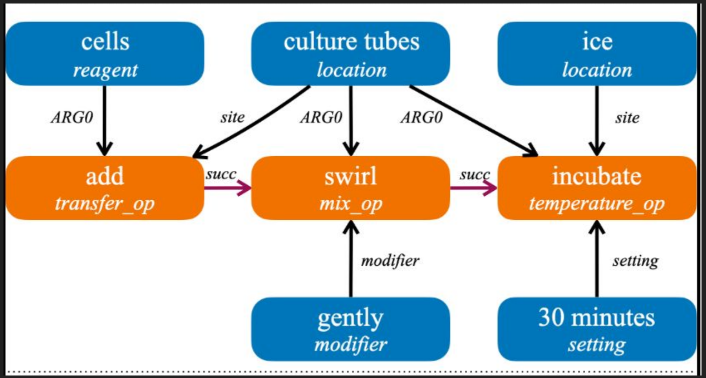
PEG: Definitions

- Directed, a-cyclic labeled graph
- Ontology based on Autoprotocol



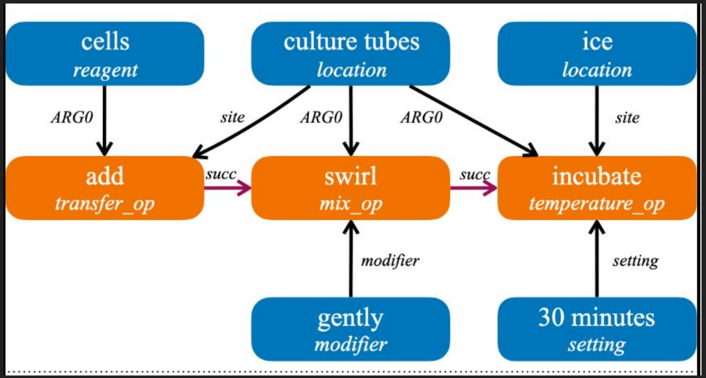
PEG: Definitions

- Directed, a-cyclic labeled graph
- Ontology based on Autoprotocol
- Nodes
  - Predicates (**mix**, **transfer**)
  - Arguments
    - Physical lab entities (device, reagent, etc)
    - Abstract entities like amounts or modifiers



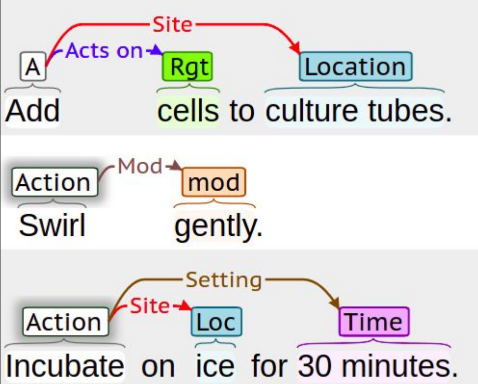
PEG: Definitions

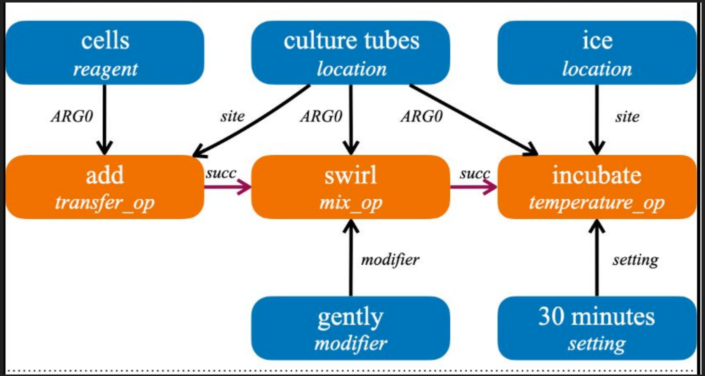
- Directed, a-cyclic labeled graph
- Ontology based on Autoprotocol
- Edges
  - Core-roles (~positional arguments)
  - Non-core roles (predicate agnostic)
  - Temporal dependency relation



Comparison with action-graphs

- Fine-grained operation types
- Cross-sentence relations
- Argument re-use: arguments can be persistent objects
- Enforcing required arguments



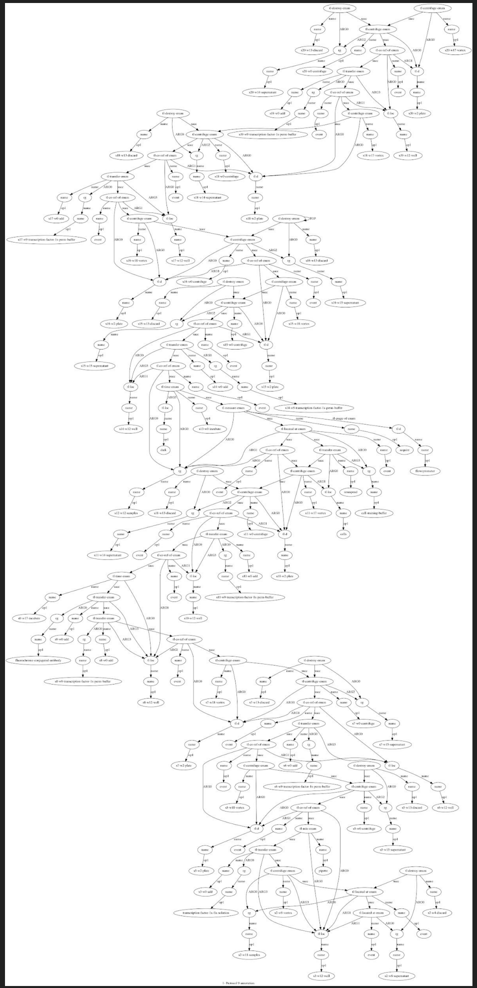


Annotation interface desiderata

- Predicate specific execution semantics (container moves -> containee moves)
- Tracking temporal dependencies and entity states over long texts
- Argument validation

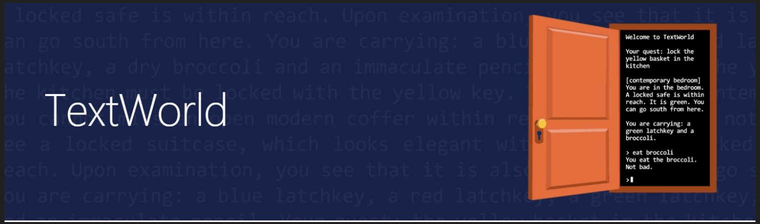
Too complex for span-based annotation!

PEG visualization in AMR using AMRICA (Safra & Lopez, 2015)



Text-based games as annotation interface

- Convenient platform providing cheap interface
- Suited for tracking and representing dynamic world state
- We’ll adapt it for annotation, too!

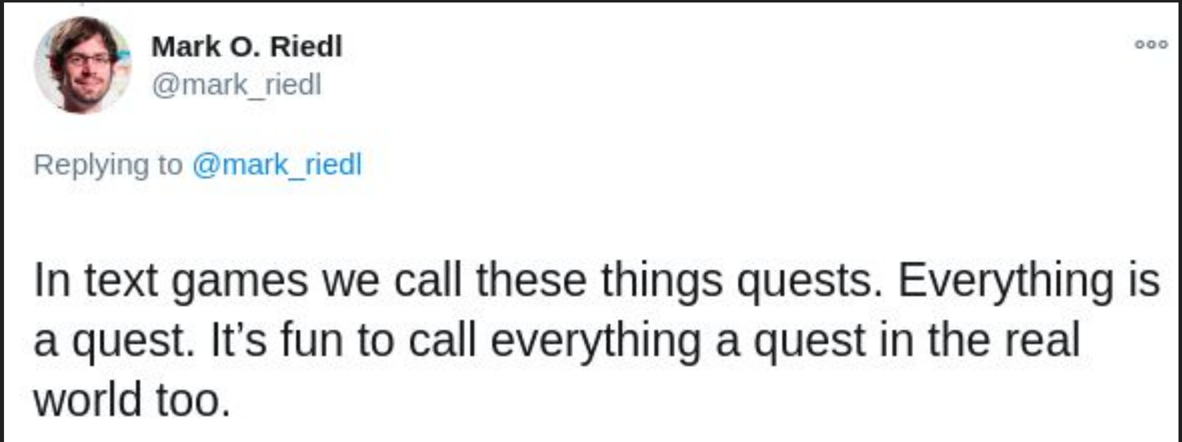


(Côté et al, 2018)

(Tamari et al, 2019)

Text-based games as intuitive annotation interface?

- Procedures most naturally thought of as sequential execution
- “Questification”: maybe grounded annotations are easier for humans (and machines)?



Mark O. Riedl
@mark_riedl

Replying to @mark_riedl

In text games we call these things quests. Everything is a quest. It’s fun to call everything a quest in the real world too.

Creating the dataset: eXecutable WLP (X-WLP)

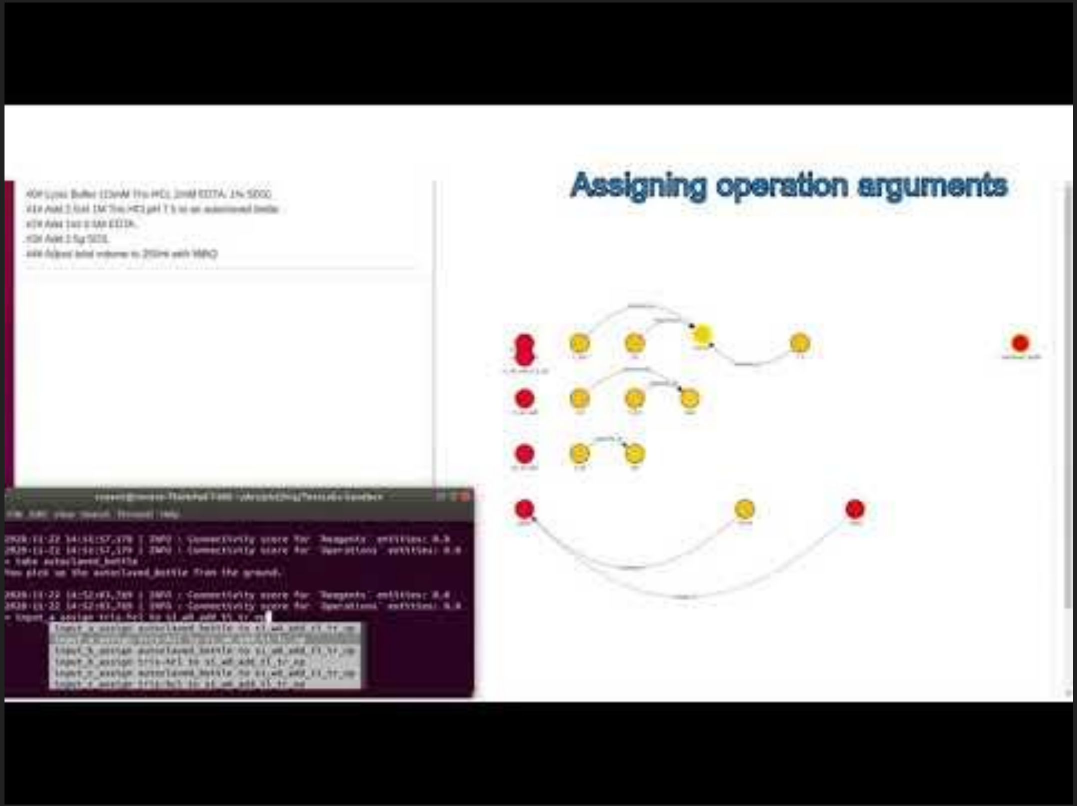
- Pre-populate PEGs with WLP entities and subset of within-sentence relations
- Annotators enrich them by adding:
  - Type grounding
  - Argument labelling
  - Execution dependencies
  - Cross-sentence relations

PEG annotation process

- Annotators interact with text-based game engine
- Engine ensures semantic validity
- Further provided assistance:
  - “Linter” for type checking
  - Autocomplete
  - Simple scoring to encourage annotating connected graphs

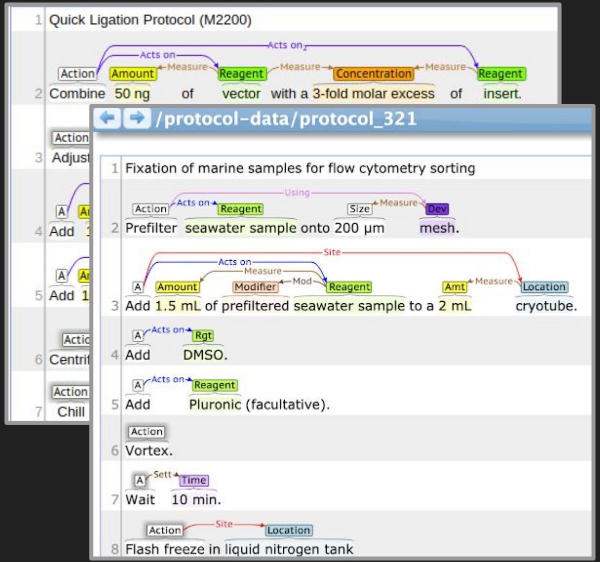
```
> op_type incubate to temp_type
Set operation type!

> op_run incubate
Failed: missing “ARG0” input for incubate!

> take culture_tubes
You pick up the culture_tubes.

> ARG
      ARG0_assign culture_tubes to incubate
      ARG1_assign culture_tubes to incubate
      ARG2_assign culture_tubes to incubate
```

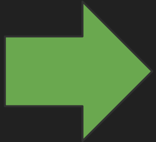
Demo

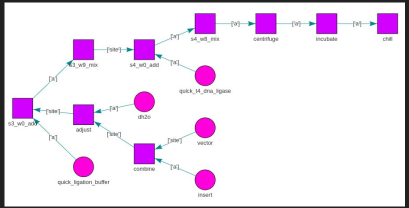


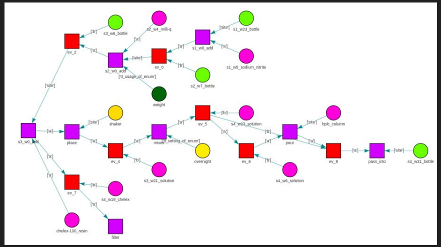
X-WLP stats

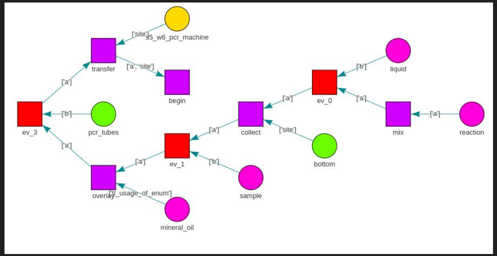
> 3 annotators, enriched 279/622 (45%) WLP protocols to PEG format

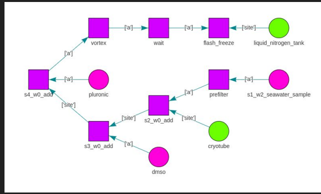












X-WLP stats

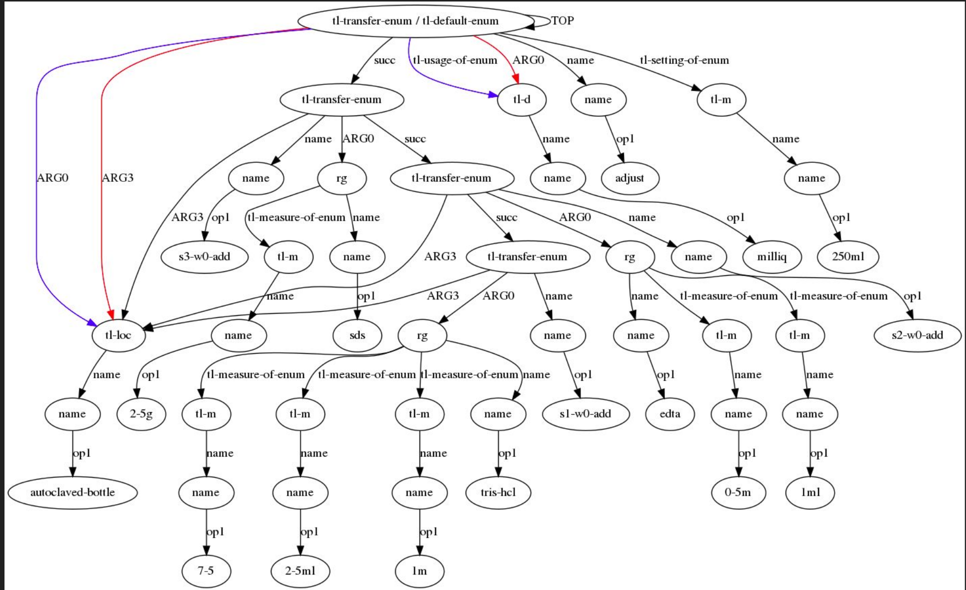
- 3 annotators, enriched 279/622 (45%) WLP protocols to PEG format
- Comparable with other procedural text datasets

Table 5: Statistics of our annotated corpus (X-WLP), compared with the ProPara corpus [Dalvi et al., 2018], material science (MSPTC; Mysore et al. [2019]) and chemical synthesis procedures (CSP; Vaucher et al. [2020]). CSP is comprised of annotated sentences (document level information is not provided)

	X-WLP (ours)	MSPTC	CSP	ProPara
# words	54k	56k	45k	29k
# words / sent.	12.7	26	25.8	9
# sentences	4,262	2,113	1,764	3,300
# sentences / docs.	15.28	9	N/A	6.8
# docs.	279	230	N/A	488

Quantitative analysis: annotator agreement

- Use Abstract Meaning Representation (AMR) format for established graph agreement metrics (Smatch, Cai & Knight, 2013)



1- Protocol 604 annotation.

Quantitative analysis: annotator agreement

- Use Abstract Meaning Representation (AMR) format for established graph agreement metrics (Smatch, Cai & Knight, 2013)
- Mean 84.99 F1 Smatch comparable to AMR datasets (69 - 89 F1)

Benefits from underlying WLP annotations

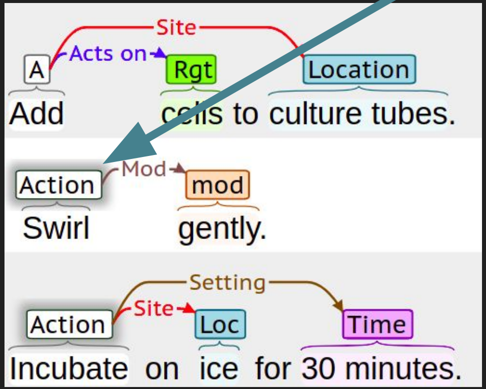
Longer-range, often cross sentence relations

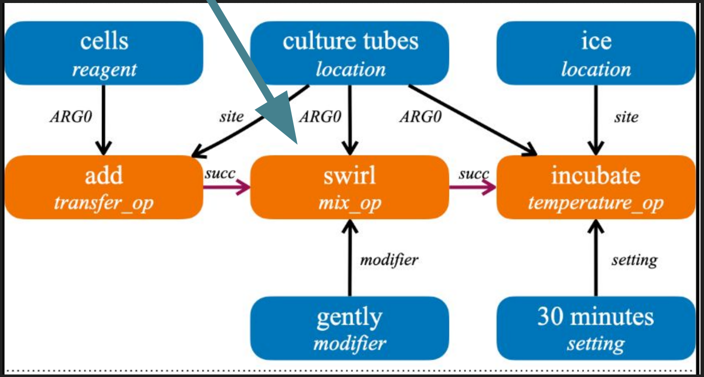
Agreement Metric	F1
Smatch	84.99
Argument identification	89.72
Predicate identification	86.68
Core roles	80.52
Re-entrancies	73.12

Quantitative analysis: operation arguments

- Simulator input validation prevents semantic underspecification, increases overall argument count per op.

Dataset	Avg. #args/op	#Ops. w/o core arg.	#Ops.	Pct.
WLP	1.87	3297	17485	18.9
X-WLP	3.01	0	3915	0.0





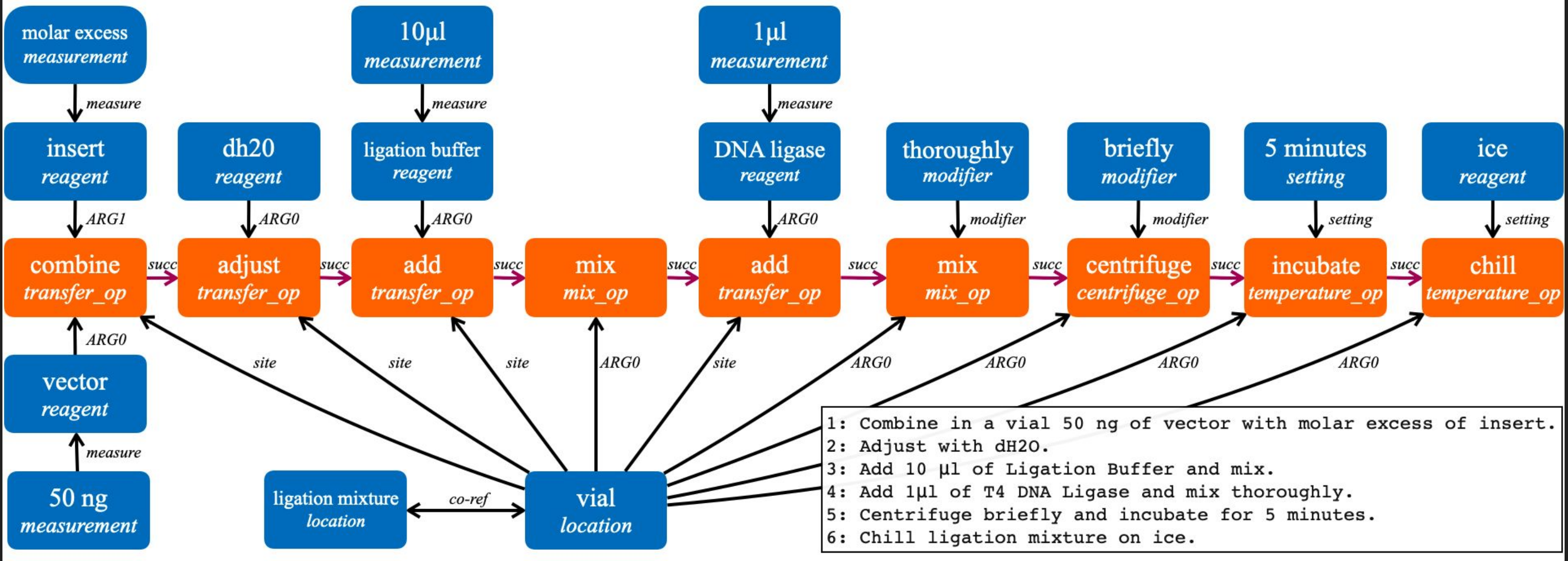
Quantitative analysis: relation types

- Significant proportion of arguments are re-entrancies (>30%)
- Many cross-sentence co-reference relations (>90%) - provide process-level structure

Relation	# Intra.	# Inter.	Total	# Re-entrancy
Core				
• ARG0	2962	952	3914	1645
• ARG1	560	127	687	3
• ARG2	84	123	207	77
Total (core)	3606	1202	4808	1725
Non-Core				
• site	1306	325	1631	360
• setting	3499	2	3501	-
• usage	1114	24	1138	-
• co-ref	129	1575	1704	-
• located-at	199	72	271	-
• measure	2936	18	2954	-
• modifier	1861	2	1863	-
• part-of	72	65	137	-
Total (non-core)	11116	2083	13199	360

Qualitative Analysis: complex co-reference

- Many long-range metonymic co-references - what does “ligation mixture” refer to?

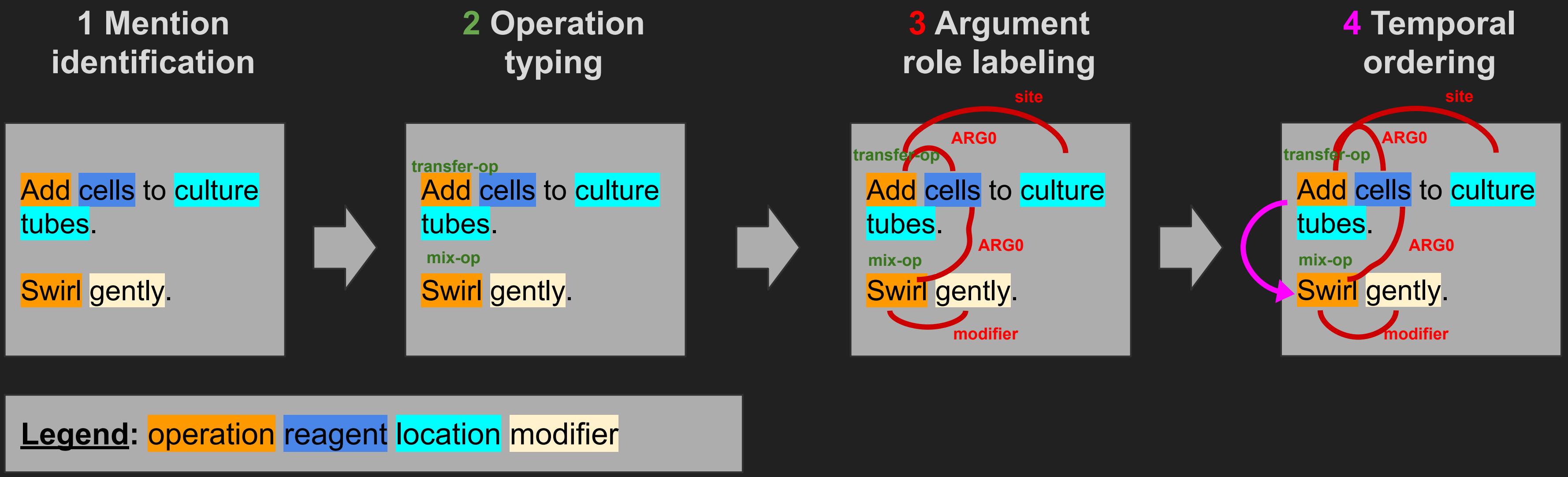


Modelling

- Tried two approaches:
  - Standard pipeline model based on SciBERT (Beltagy et. al, 2019)
  - Multi-task: jointly learn to predict entire PEG. Based on DyGIE++ (Wadden et. al, 2019)

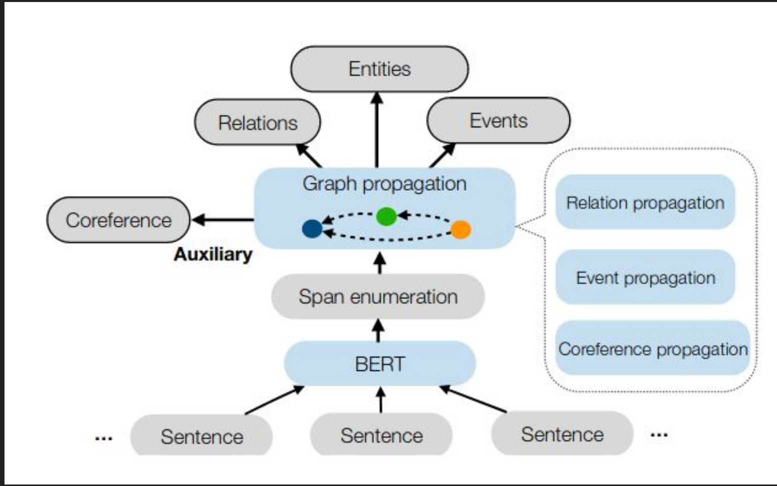
Modelling (1): Pipeline Approach

> Train model for each sub-task, chain together to obtain full PEG



Modelling (2): Multi-task Approach

- Adapted DyGIE++ (Wadden et. al, 2019) for our protocols
  - Used sliding window as length exceeded SciBERT 512-token limit



DyGIE++ Framework (Wadden et. al, 2019)

Results

1. Mention Identification

Data Split	System	F_1
original	Kulkarni et al. (2018)	78.0
	Wadden et al. (2019)	**79.7**
	PIPELINE	78.3

2. Fine-grained operation typing

System	P	R	F_1
MULTI-TASK	75.6	68.9	72.1
PIPELINE	69.2	78.4	73.6
• w/ gold mentions	78.6	81.0	79.8

Results

3 + 4: Argument role labeling + temporal ordering (relation classification)

Task	MULTI-TASK	PIPELINE	# gold
Core			
• All roles	**59.2**	49.1	2839
• All roles (gold mentions)	-	70.8	2839
• ARG0	**62.0**	52.2	2313
• ARG1	**39.4**	28.9	412
• ARG2	**70.7**	57.4	114
Non-Core			
• All roles	**55.6**	44.6	4827
• All roles (gold mentions)	-	72.3	4827
• site	**60.5**	52.5	962
• setting	**77.4**	62.7	974
• usage	**35.0**	29.5	297
• co-ref	**41.2**	30.8	1014
• measure	**64.0**	52.6	804
• modifier	**50.0**	42.4	519
• located-at	**13.4**	10.5	179
• part-of	**8.5**	8.5	78
Temporal Ordering	**60.3**	49.0	1200
Temp. Ord. (gold mentions)	-	67.0	1200

Multi-task better on all relation-classification tasks, likely due to less cascading errors

Local (intra-sentence) relations easier to predict than cross sentence relations

Results: intra vs inter sentence relations

- For core-roles:

Split	MULTI-TASK	PIPELINE	# gold
Intra-sentence	**63.3**	55.6	2160
Inter-sentence	**42.1**	29.4	679

- For co-reference (92% are inter-sentence):

co-reference	**41.2**	30.8	1014

Cross-sentence relations ~ process-level information -> key challenge for modelling!

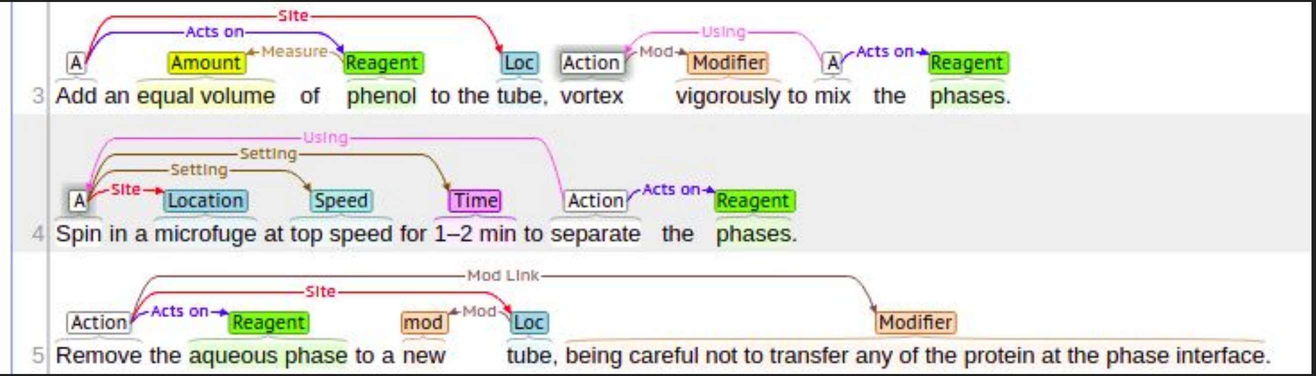
Future work: modelling

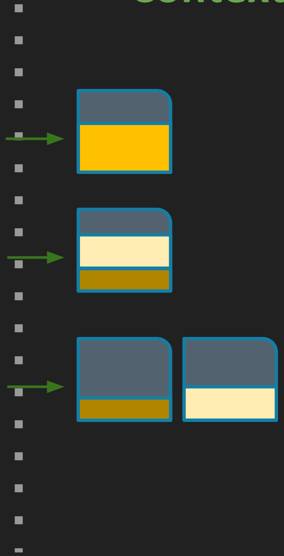
- Text-based game format allows alternate approach - predict instructions rather than predicting graph
- Interactive semantic parsing setting - allows executing instructions and utilizing intermediate game state as context.

Understanding protocols with context

Text

Context



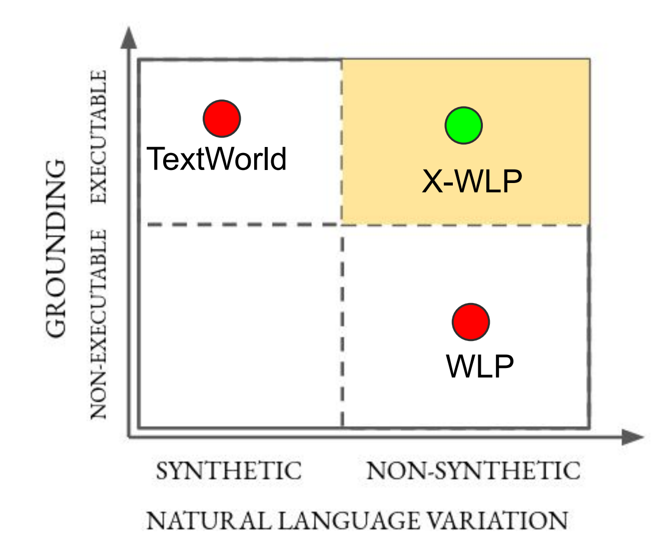


Future work: improve interface

- Reduce reliance on pre-existing annotations (e.g., allow creating new entities)
- Ad hoc implementation, lots of improvements possible to facilitate more general applicability
  - Requires writing both Inform7 and Python, should require just one language...

Towards grounded procedural text

> Possible to obtain grounded annotations of long, complex real-world scientific texts- given the right framework!



Concluding remarks

- Simulator improves data quality through questification & validation
- Modelling results highlight interesting process-level inference challenges
- New take on annotation: simulation rather than labelling
  - Facilitates new training, modelling and evaluation methods
- Data and simulator to be released with camera-ready!

THE END

Questions?