

Neurosymbolic Ontology and Knowledge Graph Creation

Cogan Shimizu
cogan.shimizu@wright.edu
coganshimizu.com

Knowledge and Semantic Technologies Lab
Wright State University

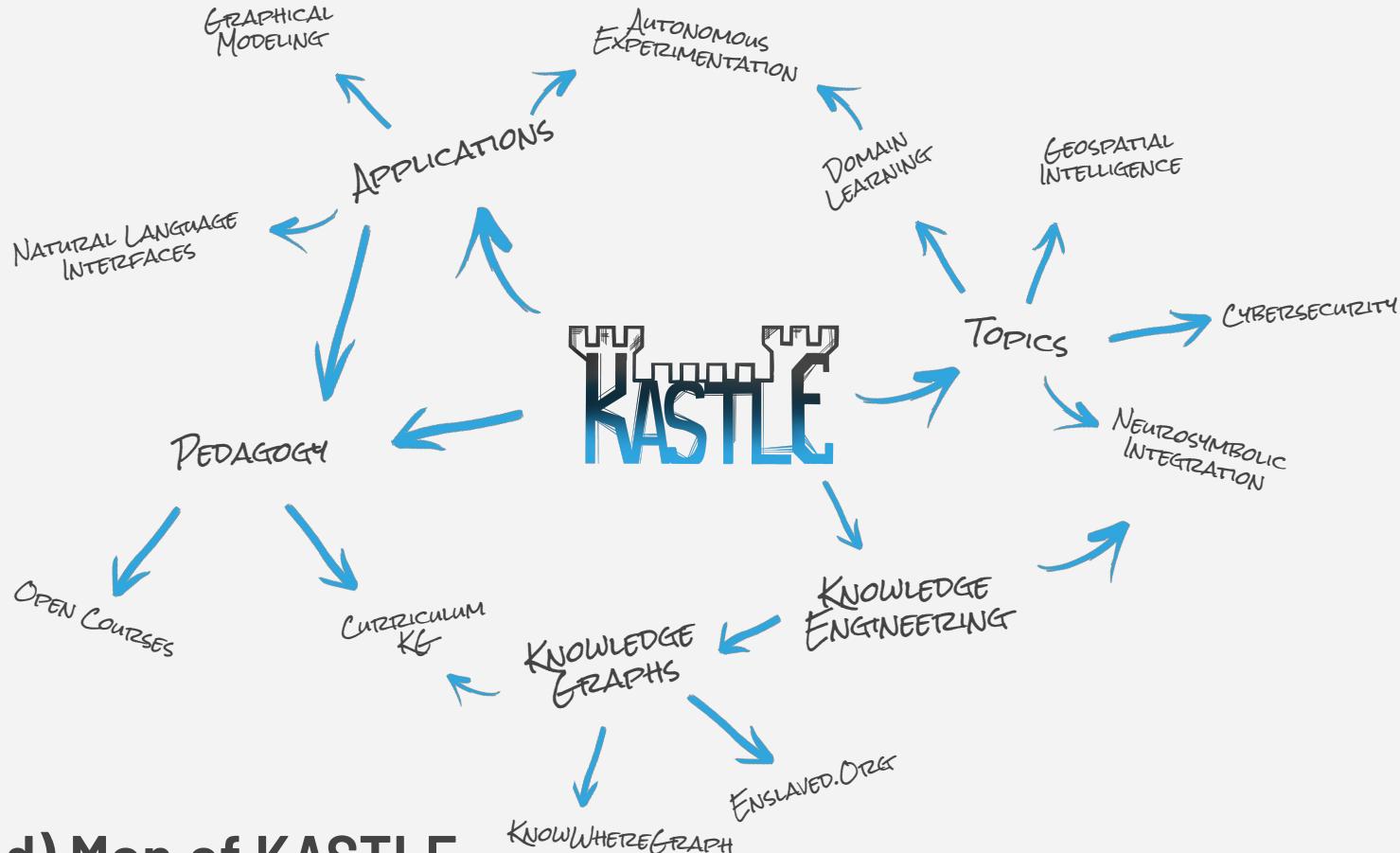
Cogan Shimizu



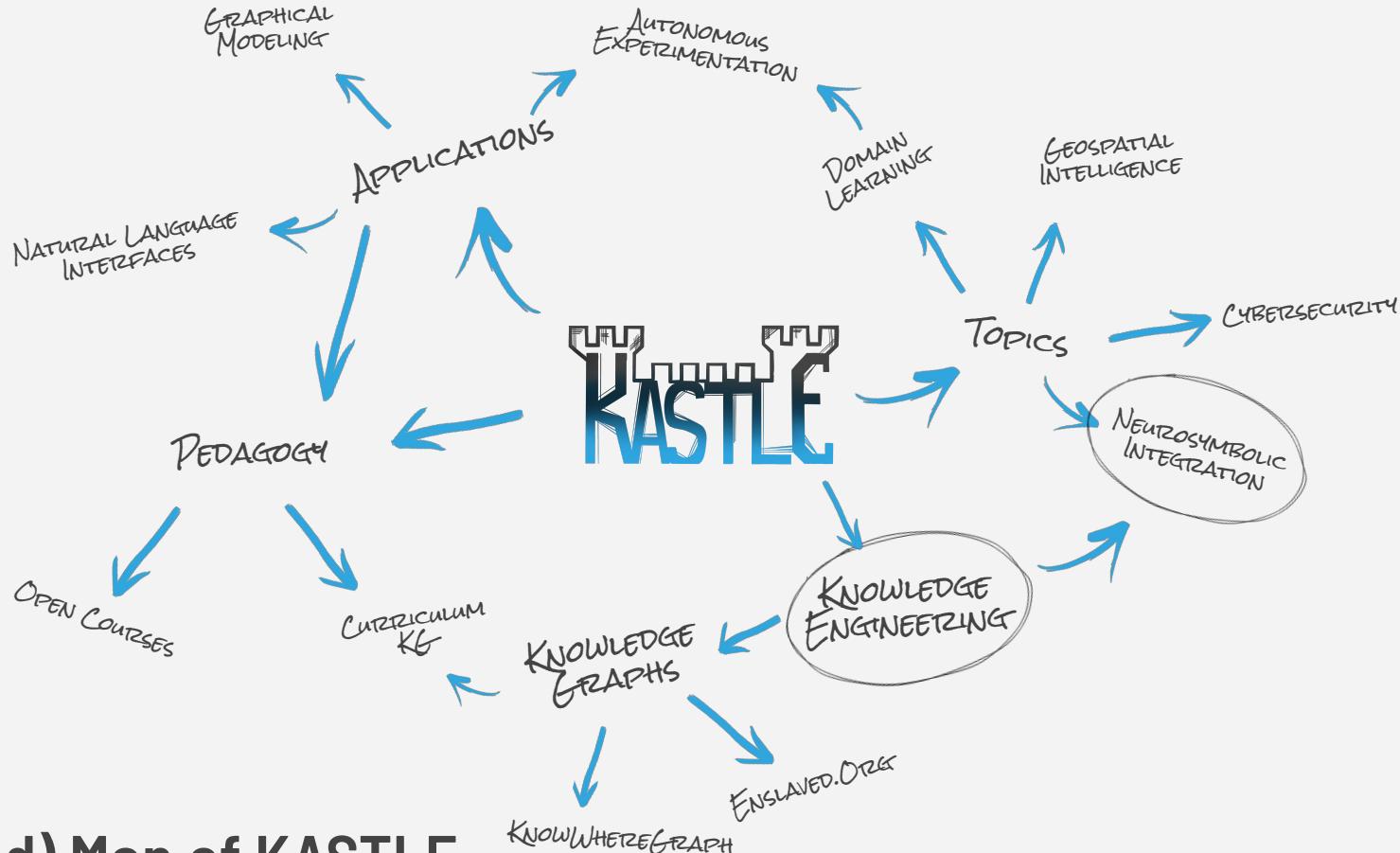
Assistant Professor

Department of Computer Science & Engineering
Wright State University





A (Sliced) Map of KASTLE



A (Sliced) Map of KASTLE

Outline & Objectives

An Overview

**Neurosymbolic Ontology
& KG Creation (NOK)**

NOK's Next Steps

A Conversational Ontologist

Outline & Objectives

An Overview

Neurosymbolic Ontology
& KG Creation (NOK)

NOK's Next Steps

A Conversational Ontologist

Overview: *Knowledge Engineering*

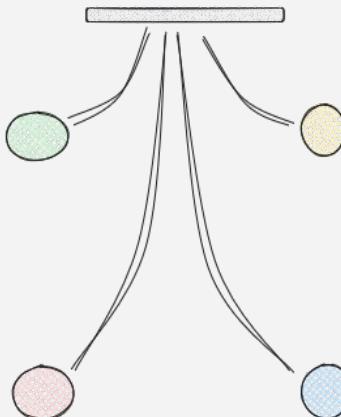
Some Examples...

Schema Development

Develop a schema for a KG with a methodology, using available data and subject matter expertise

Schema Learning

Automatically generate a schema for knowledge extracted from unstructured text corpora



Knowledge Alignment

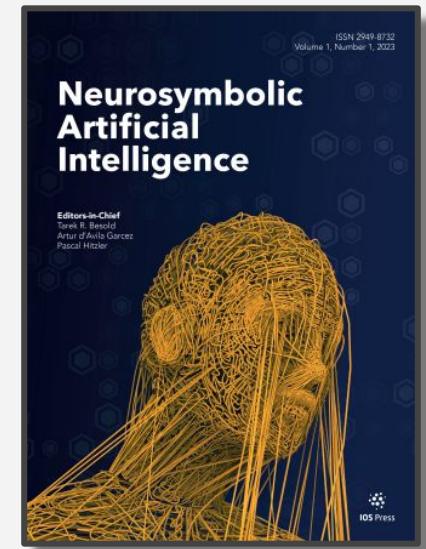
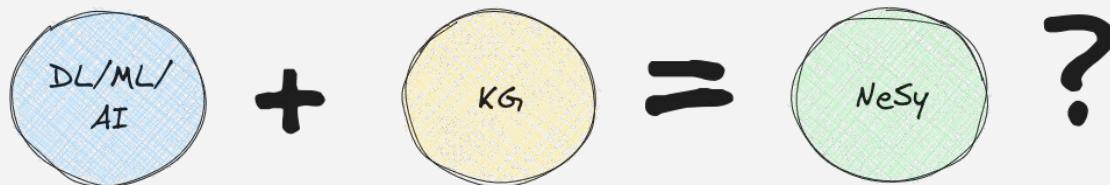
Integrate knowledge fragments modeled in conceptually distinct ways into a singular KG

KG Deployment

Understand the technology stack for efficiently deploying and exposing a knowledge graph

Overview: Neurosymbolic Artificial Intelligence

"Neurosymbolic AI is an emerging field of AI aiming to build rich computational AI models, systems and applications by combining neural and symbolic learning and reasoning."

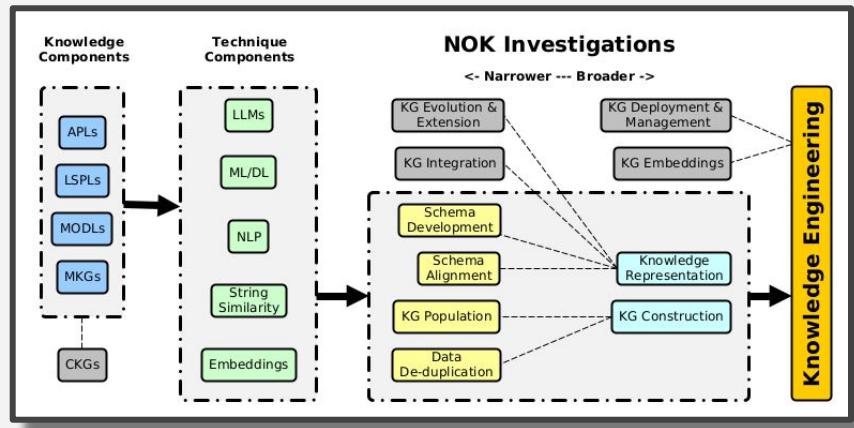


The Spaghetti Monster

Overview: NOK

Neurosymbolic Ontology & KG Creation

- Based off of a proposal
- Spurred by the hype cycle and an obvious understanding that this should be next!
- Use the power and “speed” of DL systems to create symbolic systems (and other combinations)



Outline & Objectives

An Overview

Neurosymbolic Ontology & KG Creation (NOK)

NOK's Next Steps

A Conversational Ontologist

Meme Interlude

- This is also my face
the last few days
staring at this
presentation...



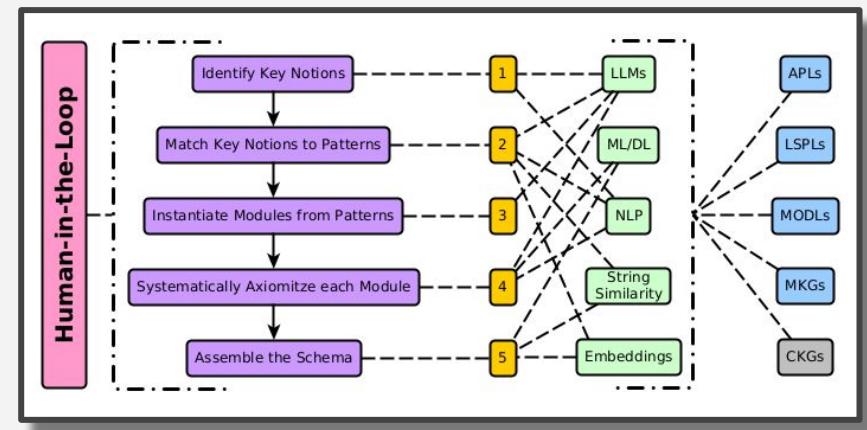
Modular Ontology Modeling

1. Define the use case
 2. Make competency questions
 3. Identify key notions
 4. Match patterns to key notions
 5. Instantiate the patterns
 6. Systematic axiomatization
 7. Assemble the modules
 8. Review final product
 9. Produce artifacts
- Modular Ontology Modeling (MOMo) is a well-documented, iterative methodology that emphasizes the use of modules as the primary component for creating a knowledge graph schema
 - Divide and conquer approach

NOK: Redux

Neurosymbolic Ontology & KG Creation

- Take M^oM_o and automate it
- Use Libraries of knowledge & patterns
- Integrate with ML/DL techniques



NOK: Knowledge Components

Axiom Patterns

Common axiomatic structure
independent of *meaning*

Lexico-syntactic Patterns

Common axiomatic structure

Ontology Design Patterns

Domain-invariant modeling solutions

- | | | |
|------------------------------------|-------------------------------------|--|
| 1. $A \sqsubseteq B$ | 7. $A \sqsubseteq R.B$ | 13. $\top \sqsubseteq \leq 1R^-. \top$ |
| 2. $A \sqcap B \sqsubseteq \perp$ | 8. $B \sqsubseteq R^- . A$ | 14. $\top \sqsubseteq \leq 1R^-. A$ |
| 3. $\exists R. \top \sqsubseteq A$ | 9. $\top \sqsubseteq \leq 1R. \top$ | 15. $B \sqsubseteq \leq 1R^-. \top$ |
| 4. $\exists R. B \sqsubseteq A$ | 10. $\top \sqsubseteq \leq 1R.B$ | 16. $B \sqsubseteq \leq 1R^-. A$ |
| 5. $\top \sqsubseteq \forall R. B$ | 11. $A \sqsubseteq \leq 1R. \top$ | 17. $A \sqsubseteq \geq 0R.B$ |
| 6. $A \sqsubseteq \forall R. B$ | 12. $A \sqsubseteq \leq 1R.B$ | |

NOK: Knowledge Components

Axiom Patterns

Common axiomatic structure
independent of *meaning*

Lexico-syntactic Patterns

Common axiomatic structure

Ontology Design Patterns

Domain-invariant modeling solutions

NL Formulation

- Engineering project managers participate in writing specifications, researching, and selecting suppliers and materials.
- Players are involved in competitions.

LSP Formalization

```
NP<object> participate/take part in/be involved in (NP<event>,* and] NP<event>
```

Reusable JAPE code: PA_1.jape

NOK: Knowledge Components

Axiom Patterns

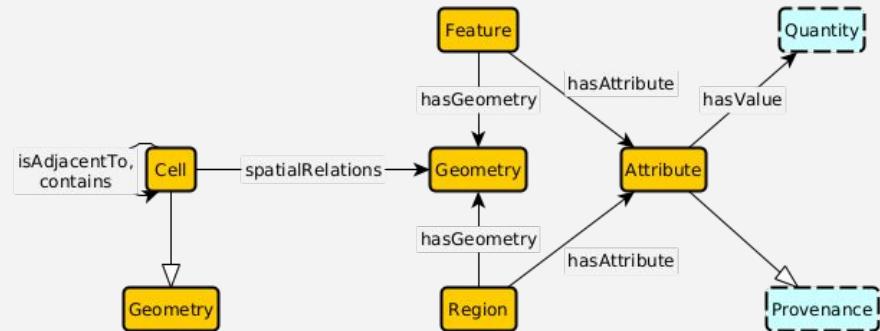
Common axiomatic structure
independent of *meaning*

Lexico-syntactic Patterns

Common axiomatic structure

Ontology Design Patterns

Domain-invariant modeling solutions



NOK: AI Components

LLMs & ML/DL

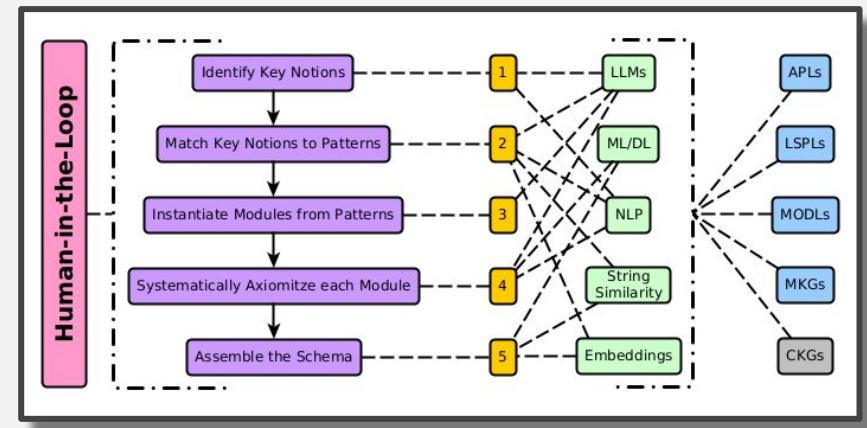
Topic Modeling, Keyword extraction

NLP & SS

Part of Speech tagging

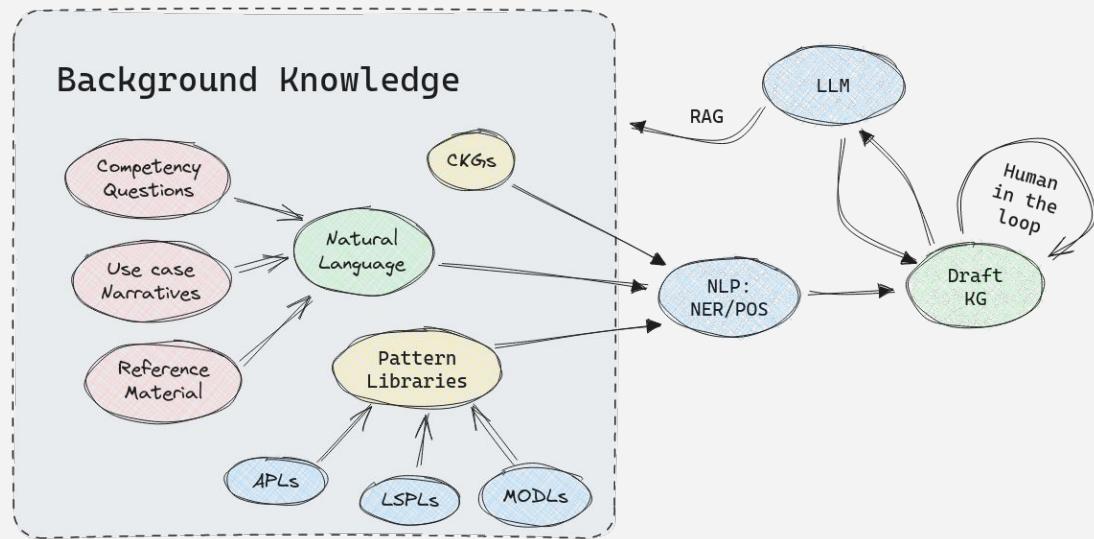
Embeddings

Co-reference resolution, de-duplication, link prediction



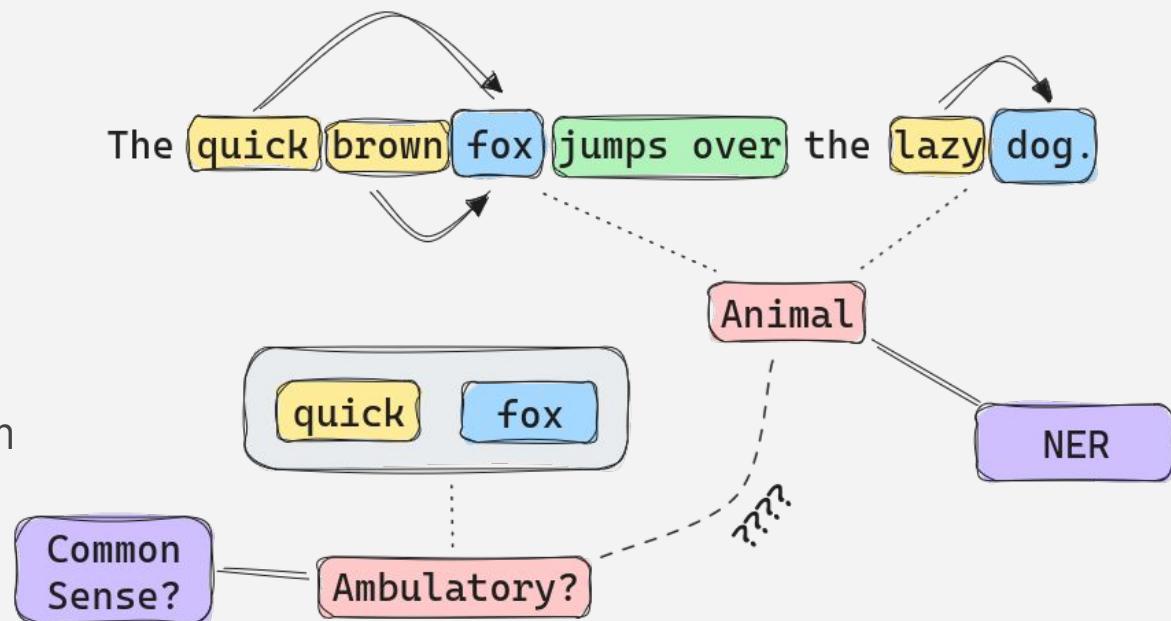
NOK: Implementation Overview

- "The Devil's in the Details"
 - Pascal
- Making the components work in concert
- Lots of pairwise augmentations
 - improving NER with CKG entities
 - Fine tuning LLMs
 - PoS to LSPs



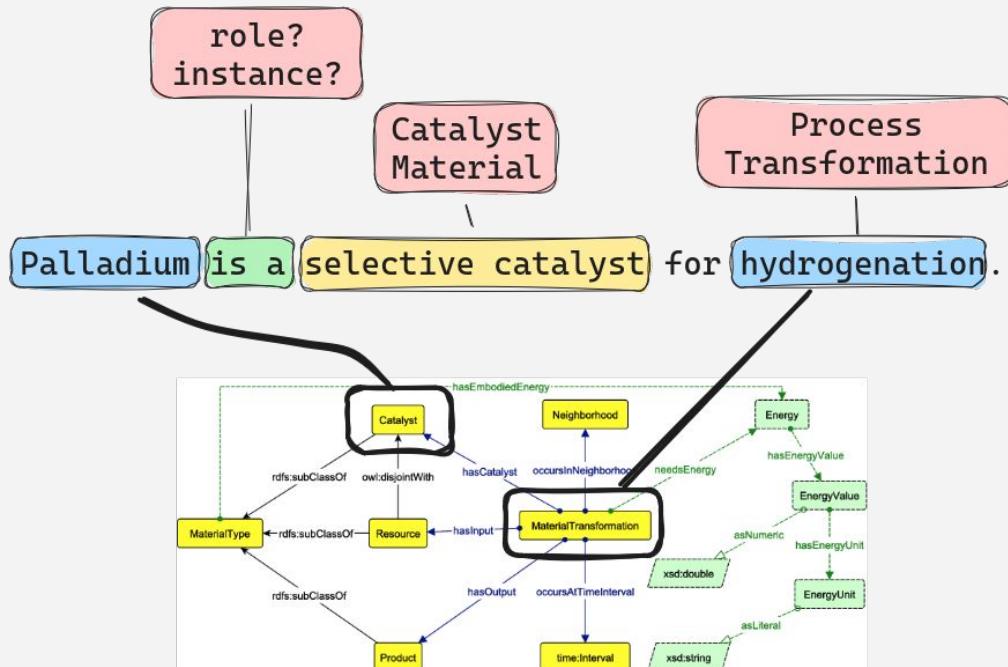
NOK: Implementation Thoughts

- Draft an initial naïve KG
E.g., one with no schema
- Using word sense, frame semantics, ontolemon (ontolex), NER, NLP to guess at key notions from “instance data”
- Rearrange triples according to patterns



NOK: Implementation Thoughts

- LSPs can be used to match sentences to fragments of ODPs.
- Looking at missing data from a pattern might help extract more from the text.



Outline & Objectives

An Overview

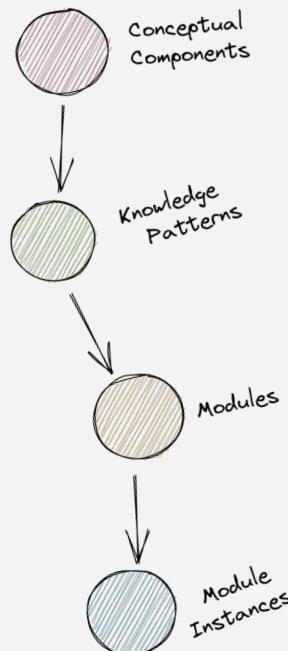
Neurosymbolic Ontology
& KG Creation (NOK)

NOK's Next Steps

A Conversational Ontologist

NOK: More Patterns

- Need more patterns at different levels of abstraction
- Need more modular ontologies – modules are just (very) specific patterns ;)



Category	Patterns
Metapatterns	Explicit Typing Property Reification Stubs
Organization of Data	Aggregation, Bag, Collection Sequence, List Tree
Space, Time, and Movement	Spatiotemporal Extent Spatial Extent Temporal Extent Trajectory Event
Agents and Roles	AgentRole ParticipantRole Name Stub
Description and Details	Quantities and Units Partonymy/Meronymy Provenance Identifier

ONE DAY WE'LL HAVE MODL 2.0

NOK: More LSPs and APLs and ...

- Current crop of **LSPs** are very basic.
 - Set of LSPs for each pattern
- **APLs** need more investigation
 - Are there domains with significantly different axiom representation?
- Can we have “pre-embeddings” of data structured to a pattern?
- Shapes for every pattern.
- Can we have word embeddings of data which conforms to these patterns?

Outline & Objectives

An Overview

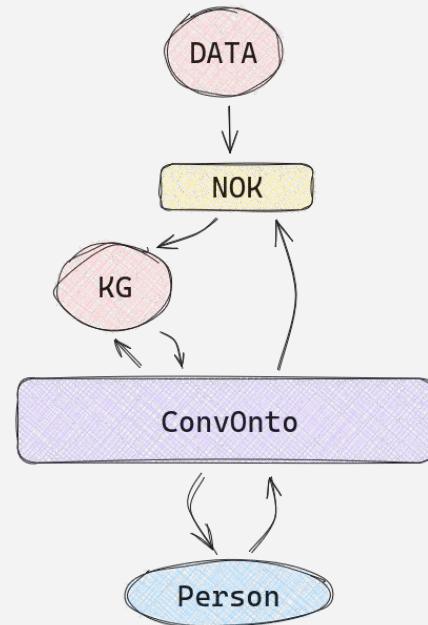
Neurosymbolic Ontology
& KG Creation (NOK)

NOK's Next Steps

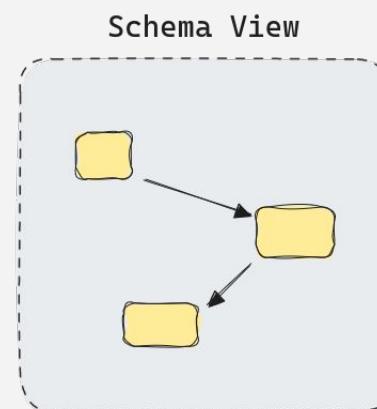
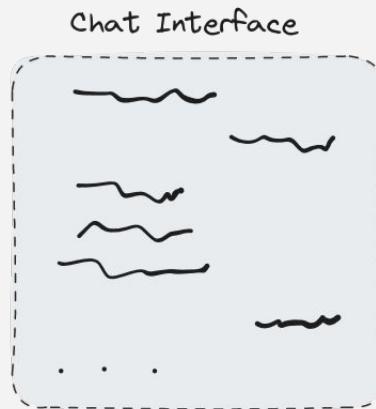
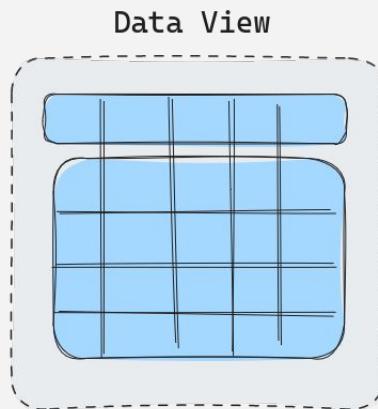
A Conversational Ontologist

ConvOnto: Overview

- Layering an interactive LLM agent over the NOK process as an additional, dynamic input of natural language to NOK
- Can we replace ourselves?



ConvOnto: Interface



- Shapes (e.g., SHACL)
- R2RML / Foundry
- LLM Agent
- Question Answering
- Ontology Drafting
- Current epistemological view of ConvOnto

ConvOnto: Integrations

The image displays two side-by-side software interfaces for ontology management and exploration.

Left Interface (Ink-browser):

- Top Left:** Statistics, Schema, Map, Search.
- Table View:** Earthquakes (ak02115pjn6u, ak02153n5s1, ak021b6xmj, nn00792158, pr2020359052) with columns Depth, Gap, Error, Mag.
- Map View:** A map of the Northeastern United States showing a location in Springfield, Massachusetts.
- Search Bar:** Search, Lookup Schema, SEARCH.
- Search Results:** A list of URLs related to Earthquakes and Geospatial features.

Right Interface (CoModIDE):

- Top Bar:** Active ontology, Entities, Individuals by class, DL Query, CoModIDE.
- Left Panel:** Core constructs (XSD datatypes), Class, Property, Subclass.
- Middle Panel:** A large diagram showing the relationships between classes like EarthquakeObservationCollection, ObservationCollection, Feature, FeatureOfInterest, and AdministrativeRegion. A red circle labeled **1** points to this panel.
- Bottom Panel:** A smaller diagram showing a simplified view of the ontology structure. A red circle labeled **2** points to this panel.
- Right Panel:** A configuration sidebar with sections for Entity naming, Module annotations placement, Edge creation, and a Pattern Library. A red circle labeled **3** points to the "Internal (using parent ontology)" checkbox under Entity naming.

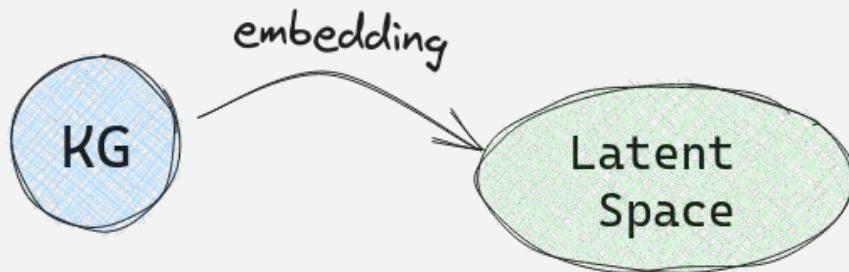
- Explore produced instance data & relations with InK Browser?
- Manually adjust the model with a CoModIDE style interface?

THANKS!

Cogan Shimizu
cogan.shimizu@wright.edu
coganshimizu.com

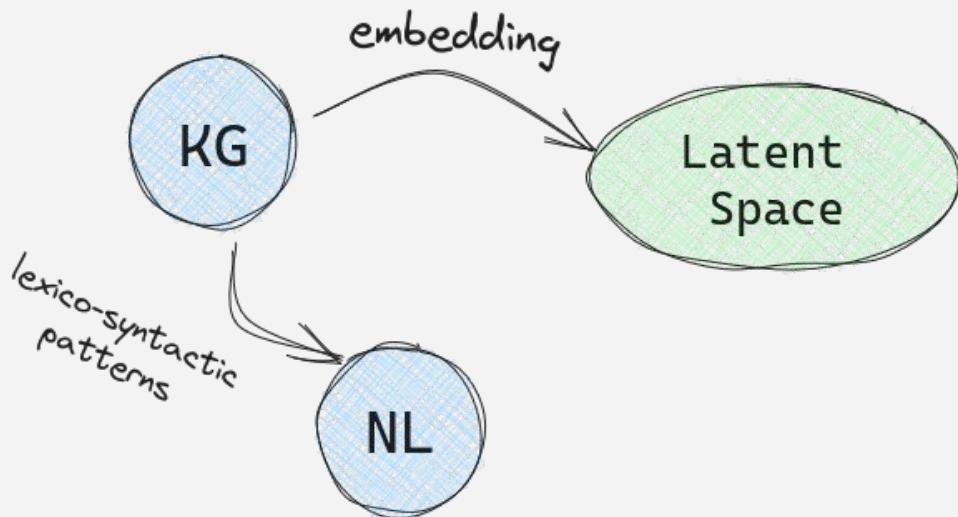
Knowledge and Semantic Technologies Laboratory

Structured Knowledge to Latent Space



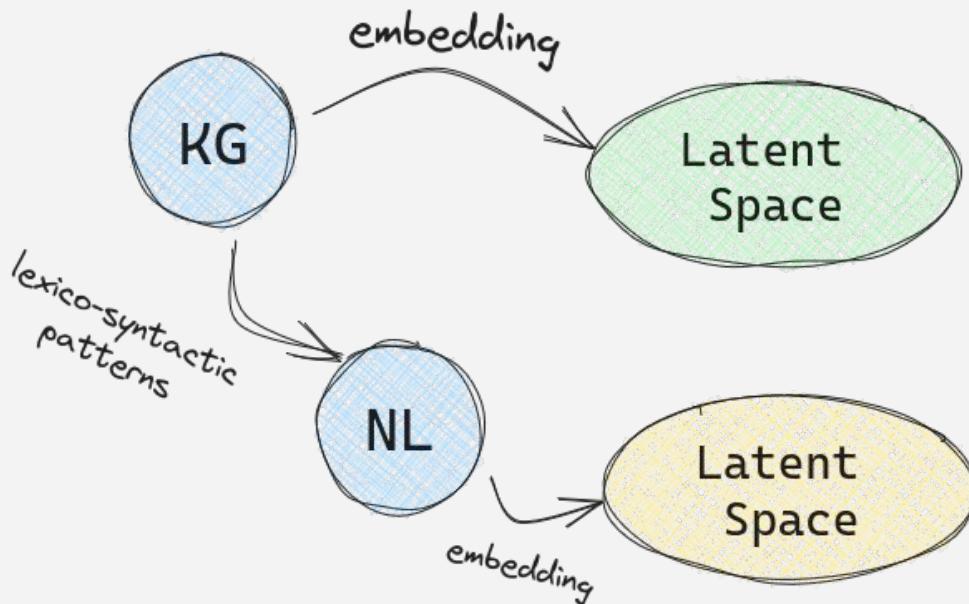
First, start with a KG and embed it into a latent space.

Structured Knowledge to Natural Language



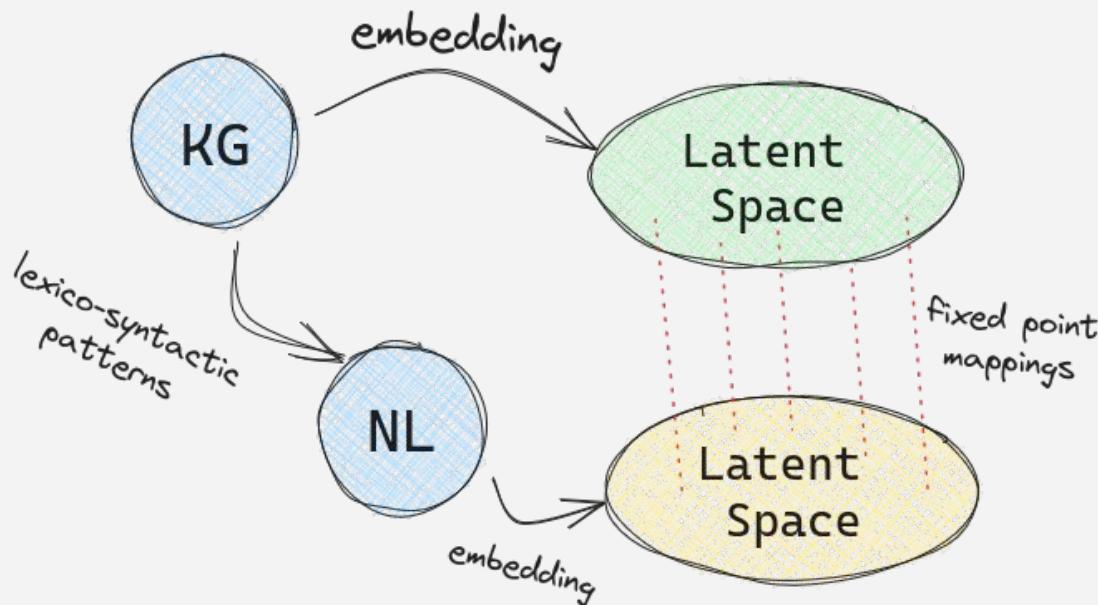
Then, using a library of LSPs, convert the KG into an NL corpus.

Natural Language to Latent Space



Take the NL Corpus and embed that into its own latent space.

Fixed-Point Mapping

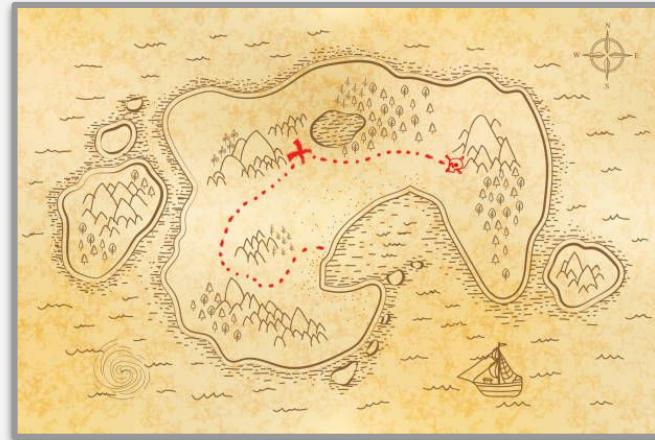


Now, train some model to map between the fixed points in the latent spaces.

Follow-up

So what does this get us?

- Extended domain transferability and augmentation
 - Describe an image in Natural Language, do we then have structured knowledge about it?
- An “easy” way to translate natural language back into structured knowledge
 - Leverage metadata and ontology once there?
- What else? (discuss!)

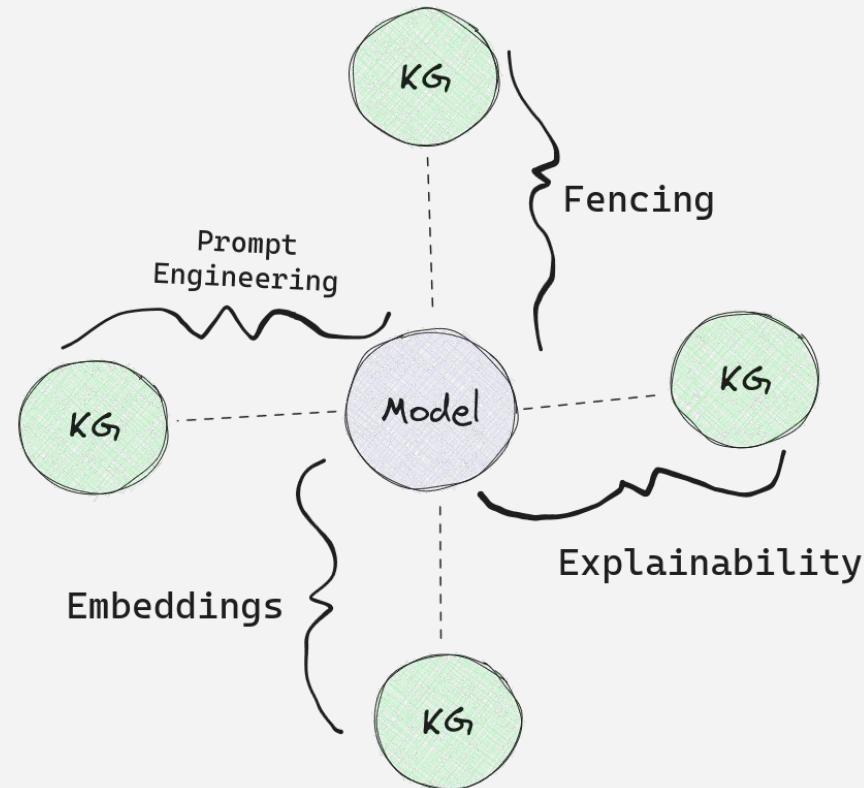


From iStock photos

Nesycule 4

What even is this thing?

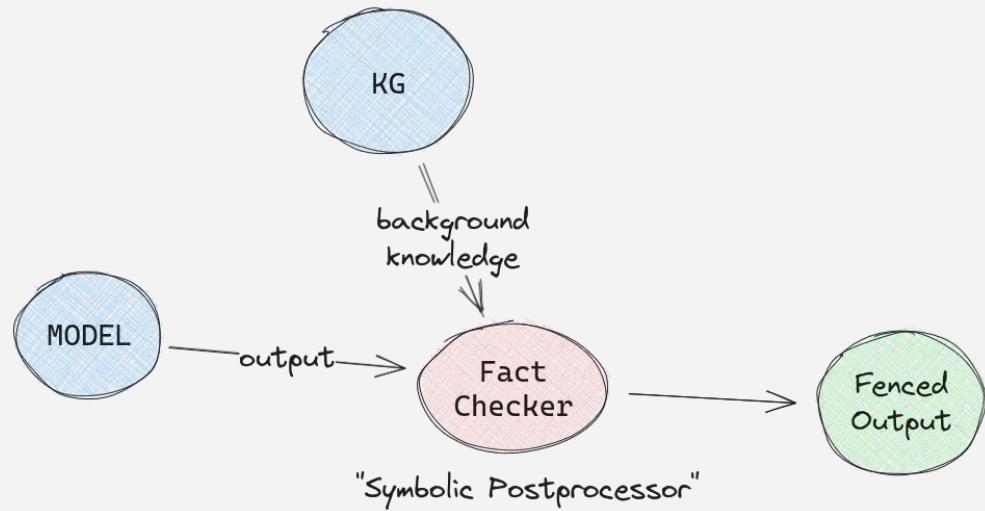
- Imagining relations between an AI/ML/DL Model in an intuitive "spatial" dimension
- KG "over" a model
- KG "alongside" a model
- Model "over" a KG



Nesycule 4: Fencing

Symbolic Post-processing

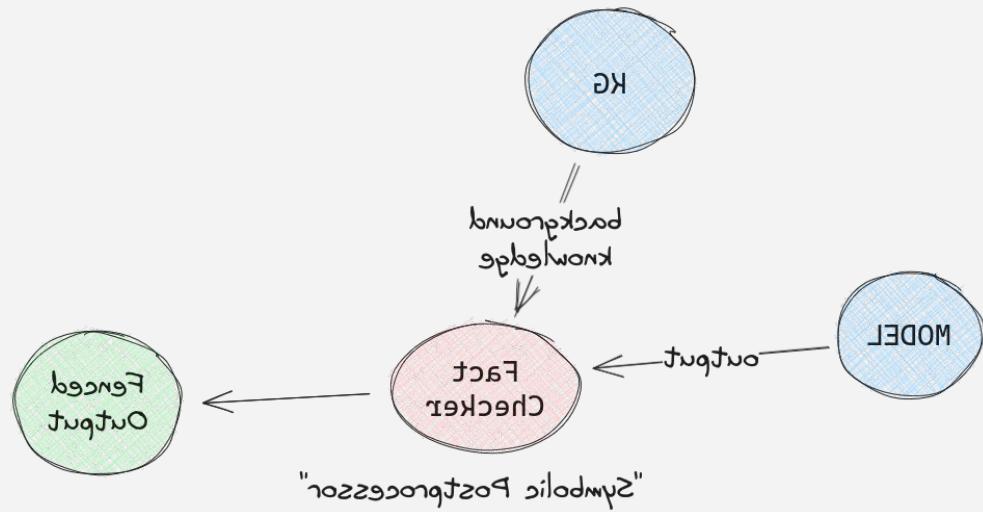
- The process by which output from an LLM (or some model) is fact-checked (common-sense, domain knowledge) for correctness
- Bridging structured knowledge to natural language
 - Lexico-syntactic patterns?
 - Controlled english?
 - Both!?



Nesycule 4: !Fencing

Flip it on its head?

- text



Nesycule 4: NOK

Symbolic Post-processing

- The process by which output from an LLM (or some model) is fact-checked (common-sense, domain knowledge) for correctness
- Bridging structured knowledge to natural language
 - Lexico-syntactic patterns?
 - Controlled english?
 - Both!?

