Explainable Patterns

Unsupervised Learning of Symbolic Representations

Linas Vepštas

15-18 October 2021

Interpretable Language Processing (INLP) – AGI–21

Introduction – Outrageous Claims

Old but active issues with symbolic knowledge in AI:

- Solving the Frame Problem
- Solving the Symbol Grounding Problem
- Learning Common Sense
- Learning how to Reason

A new issue:

- Explainable AI, understandable (transparent) reasoning.

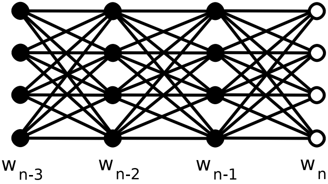
It's not (just) about Linguistics, its about about Understanding
Symbolic AI can (still) be a viable alternative to Neural Nets!

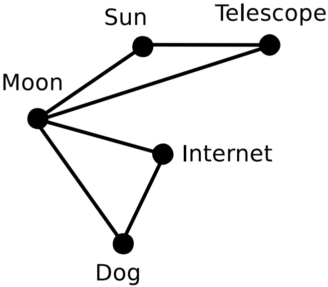
You've heard it before. Nothing new here...

... Wait, what?

Everything is a (Sparse) Graph

The Universe is a sparse graph of relationships.
Sparse graphs are (necessarily) symbolic!



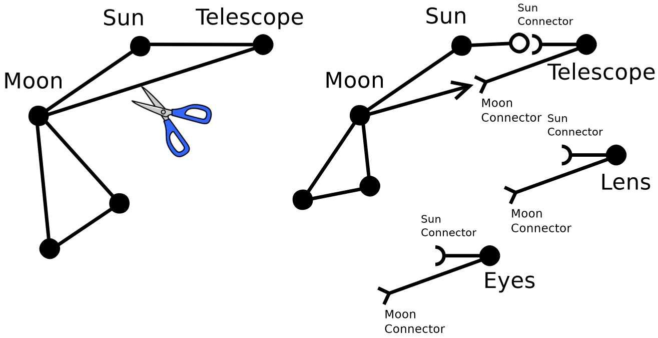


Not sparse..............................Sparse!

Edges are necessarily labeled by the vertices they connect!
Labels are necessarily symbolic!

Graphs are Decomposable

Graphs can be decomposed into interchangeable parts.
Half-edges resemble jigsaw puzzle connectors.



Graphs are syntactically valid if connectors match up.

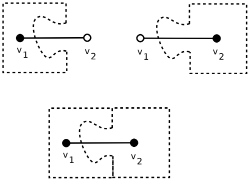
- Labeled graphs (implicitly) define a syntax!
- Syntax == allowed relationships between “things”.

Graphs are Compositional

Example: Terms and variables (Term Algebra)

- A term: $f(x)$ or an n-ary function symbol: $f(x_1, x_2, \cdots, x_n)$
- A variable: x or maybe more: $x, y, z, \cdots$
- A constant: 42 or "foobar" or other type instance
- Plug it in (beta-reduction): $f(x) : 42 \mapsto f(42)$
- "Call function f with argument of 42"

Jigsaw puzzle connectors:



Connectors are (Type Theory) Types.

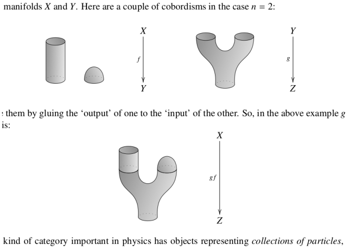
- Matching may be multi-polar, complicated, not just bipolar.

Examples from Category Theory

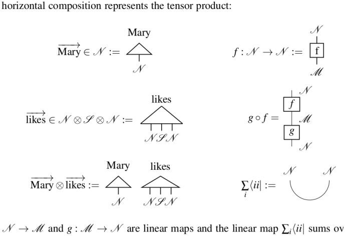
Lexical jigsaw connectors are everywhere!

- Compositionality in anything tensor-like:

Cobordism[1]



Quantum Grammar[2]



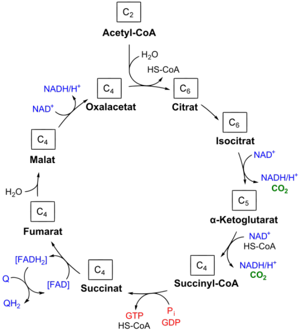
[1] John Baez, Mike Stay (2009) "Physics, Topology, Logic and Computation: A Rosetta Stone"

[2] William Zeng and Bob Coecke (2016) "Quantum Algorithms for Compositional Natural Language Processing"

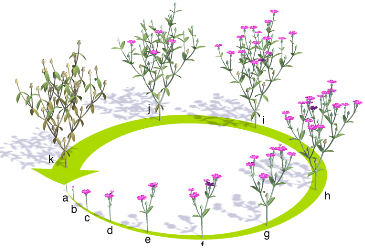
Examples from Chemistry, Botany

Lexical Compositionality in chemical reactions.
Generative L-systems explain biological morphology!

Krebs Cycle



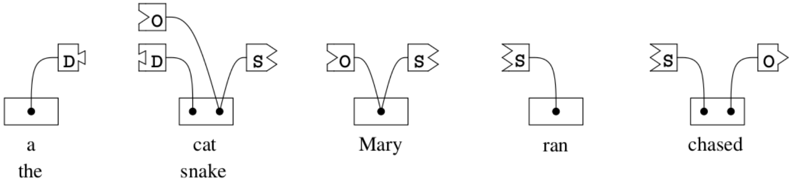
Algorithmic Botany[3]

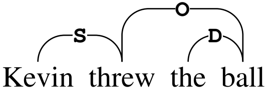


[3]Przemyslaw Prusinkiewicz, etal. (2018) “Modeling plant development with L-systems” – http://algorithmicbotany.org

Link Grammar

Link Grammar as a Lexical Grammar[4]



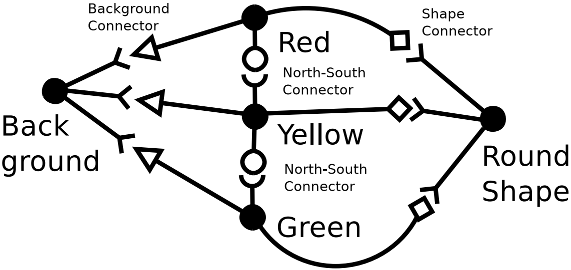


Can be (algorithmically) converted to HPSG, DG, CG, FG, ...
Full dictionaries for English, Russian.
Demos for Farsi, Indonesian, Vietnamese, German & more.

[4] Daniel D. K. Sleator, Davy Temperley (1991) "Parsing English with a Link Grammar"

Vision

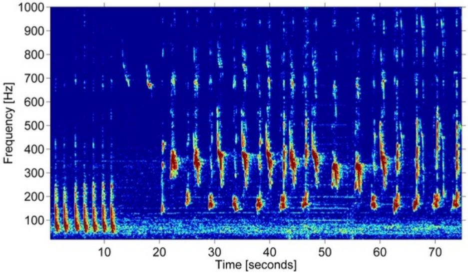
Shapes have a structural grammar.
The connectors can specify location, color, shape, texture.

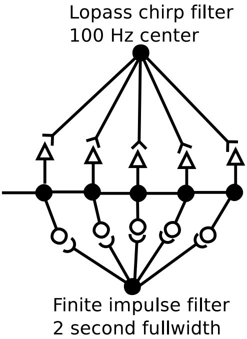


A key point: *It is not about pixels!*

Sound

Audio has a structural grammar.
Digital Signal Processing (DSP) can extract features.





Where do meaningful filters come from?

Part Two: Learning

Graph structure can be learned from observation!
Outline:

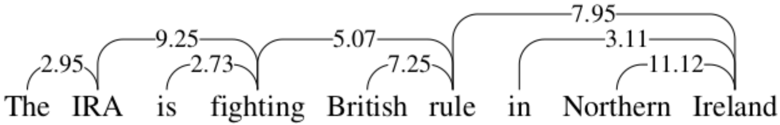
- Lexical Attraction (Mutual Information, Entropy)
- Lexical Entries
- Similarity Metrics
- Learning Syntax
- Generalization as Factorization
- Composition and Recursion

Lexical Attraction AKA Entropy

Frequentist approach to probability.
Origins in Corpus Linguistics, N-grams.
Relates ordered pairs (u, w) of words, ... or other things ...
Count the number $N(u, w)$ of co-occurrences of words, or ...
Define $P(u, w) = N(u, w)/N(*, *)$

$$LA(w, u) = \log_2 \frac{P(w,u)}{P(w,*)P(*,u)}$$

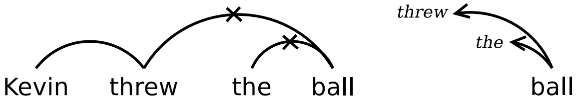
Lexical Attraction is mutual information.[5]
This LA can be positive *or* negative!



[5] Deniz Yuret (1998) "Discovery of Linguistic Relations Using Lexical Attraction"

Structure in Lexical Entries

Draw a Maximum Spanning Tree/Graph.
Cut the edges to form half-edges.



Alternative notations for Lexical entries:

- ball: the- & throw-;
- ball: $|the-\rangle \otimes |throw-\rangle$
- word: connector-seq; is a (w, d) pair

Accumulate counts $N(w, d)$ for each observation of (w, d).
Skip-gram-like (sparse) vector:

- $$\vec{w} = P(w, d_1)\widehat{e_1} + \cdots + P(w, d_n)\widehat{e_n}$$

Plus sign is logical disjunction (choice in linear logic).

Similarity Scores

Probability space is not Euclidean; it's a simplex.

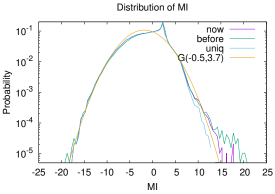
- Dot product of word-vectors is insufficient.
- $$\cos\theta = \vec{w} \cdot \vec{v} = \sum_d P(w,d)\,P(v,d)$$
- Experimentally, cosine distance low quality.

Define vector-product mutual information:

- $$MI(w,v) = \log_2 \vec{w} \cdot \vec{v} \,/\, (\vec{w} \cdot \vec{*})\ (\vec{*} \cdot \vec{v})$$

  where $\vec{w} \cdot \vec{*} = \sum_d P(w,d)\,P(*,d)$

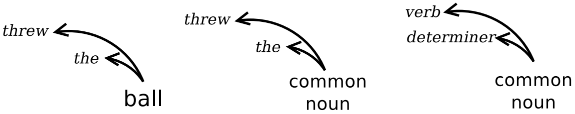
Distribution of (English) word-pair similarity is Gaussian!



- What's the theoretical basis for this? Is it a GUE ???

Learning Syntax; Learning a Lexis

Word-disjunct vectors are skip-gram-like.
They encode conventional notions of syntax:



Agglomerate clusters using ranked similarity:

$$\text{ranked } MI(w,v) = \log_2 \vec{w} \cdot \vec{v} \Big/ \sqrt{\left(\vec{w} \cdot \vec{*}\right)\ \left(\vec{*} \cdot \vec{v}\right)}$$

Generalization done via “democratic voting”:

- Select an “in-group” of similar words.
- Vote to include disjuncts shared by majority.

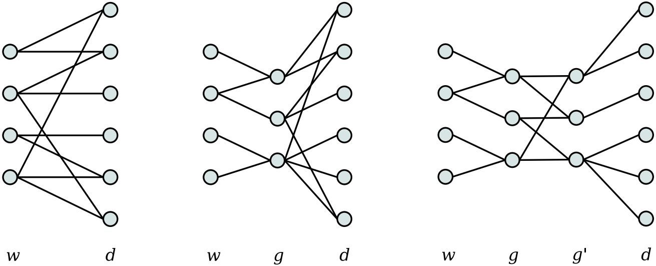
Yes, this actually works! There’s (open) source code, datasets.[6]

[6] OpenCog Learn Project, https://github.com/opencog/learn

Generalization is Factorization

The word-disjunct matrix $P(w,d)$ can be factored:

- $P(w,d) = \sum_{g,g'} P_L(w,g)\,P_C(g,g')\,P_R(g',d)$
- g = word class; g' = grammatical relation ("LG macro").
- Factorize: $P = LCR$ Left, central and right block matrices.
- L and R are sparse, large.
- C is small, compact, highly connected.



- This is the *defacto* organization of the English, Russian dictionaries in Link Grammar!

Key Insight about Interpretability

The last graph is ultimately key:

- Neural nets can accurately capture the dense, interconnected central region.
- That's *why* they work.
- They necessarily perform dimensional reduction on the sparse left and right factors.
- By erasing/collapsing the sparse factors, neural nets become no longer interpretable!
- Interpretability is about regaining (factoring back out) the sparse factors!
- That is what this symbolic learning algorithm does.

Boom!

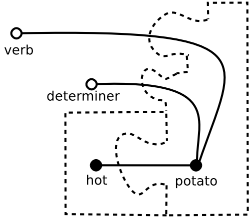
Summary of the Learning Algorithm

- Note pair-wise correlations in a corpus.
- Compute pair-wise MI.
- Perform a Maximum Spanning Tree (MST) parse.
- Bust up the tree into jigsaw pieces.
- Gather up jigsaw pieces into piles of similar pieces.
- The result is a grammar that models the corpus.
- This is a conventional, ordinary linguistic grammar.

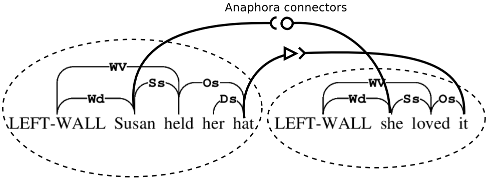
Compositionality and Recursion

Jigsaw puzzle assembly is (free-form) hierarchical!
Recursive structure exists: the process can be repeated.

Idioms, Institutional phrases



Anaphora resolution



Part Three: Vision and Sound

Not just language!

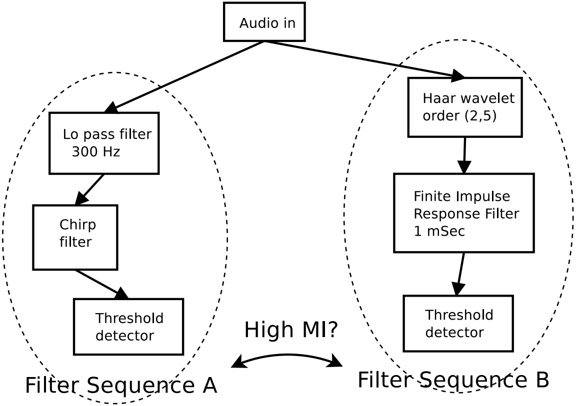
- Random Filter sequence exploration/mining
- Symbol Grounding Problem
- Affordances
- Common Sense Reasoning

Something from Nothing

What is a relevant audio or visual stimulus?

- We got lucky, working with words!

Random Exploration/Mining of Filter sequences!

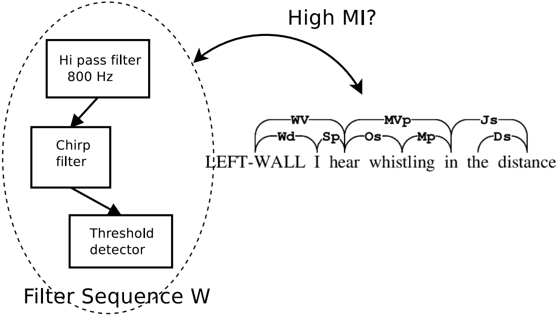


Salience is given by filters with high Mutual Information!

Symbol Grounding Problem

What is a "symbol"? What does any given "symbol" mean?

- It means what it is! Filters are interpretable.



- Solves the Frame Problem![7]
- Can learn Affordances![8]

---

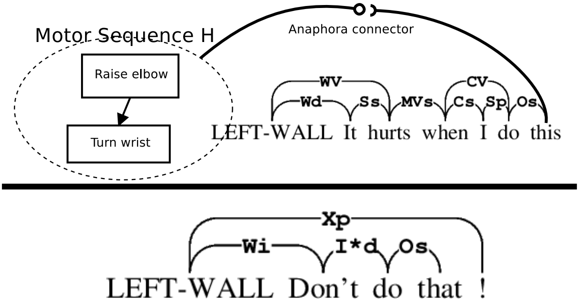
[7] Frame Problem, *Stanford Encyclopedia of Philosophy*

[8] Embodied Cognition, *Stanford Encyclopedia of Philosophy*

Common Sense Reasoning

Rules, laws, axioms of reasoning and inference can be learned.

$$\frac{A \wedge A \rightarrow B}{B}$$



Naively, simplistically: Learned Stimulus-Response AI (SRAI)[9]

[9]Metaphorical example: Mel'čuk's Meaning Text Theory (MTT) SemR + Lexical Functions (LF) would be better.

Part Four: Conclusions

- Leverage idea that everything is a graph!
- Discern graph structure by frequentist observations!
- Naively generalize recurring themes by MI-similarity clustering!
- (Magic happens here)
- Repeat! Abstract to next hierarchical level of pair-wise relations

Looking to the future:

- Better software inrastructure is needed; running experiments is hard!
- Engineering can solve many basic performance and scalability issues.
- Shaky or completely absent theoretical underpinnings for most experimental results.

Thank you!

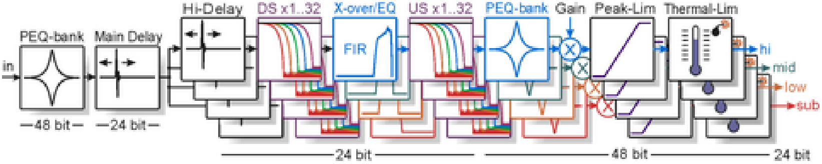
Questions?

Part Five: Supplementary Materials

- Audio Filters
- MTT SemR representation, Lexical Functions
- Curry–Howard–Lambek Correspondence

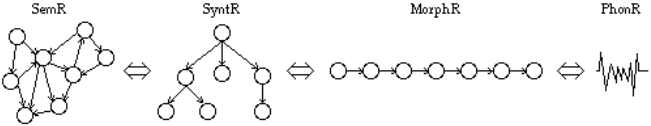
Audio Filters

A stereotypical audio processing filter sequence:



Meaning-Text Theory

Aleksandr Žolkovskij, Igor Mel'čuk[10]



Lexical Function examples:

- Syn(helicopter) = copter, chopper
- A0(city) = urban
- S0(analyze) = analysis
- Adv0(followV [N]) = after [N]
- S1(teach) = teacher
- S2(teach) = subject/matter
- S3(teach) = pupil
- ...

More sophisticated than Predicate-Argument structure.

[10] Sylvain Kahane, "The Meaning Text Theory"

Curry–Lambek–Howard Correspondence

Each of these have a corresponding mate:[11][12]

- A specific Category
  - Cartesian Category vs. Tensor Category
- An “internal language”
  - Simply Typed Lambda Calculus vs. Semi-Commutative Monoid (distributed computing with mutexes, locks e.g. vending machines!)
- A type theory[13]
- A logic
  - Classical Logic vs. Linear Logic
- Notions of Currying, Topology
  - Scott Topology, schemes in algebraic geometry

---

[11] Moerdijk & MacLane (1994) “Sheaves in Geometry and Logic”

[12] Baez & Stay (2009) “Physics, Topology, Logic and Computation: A Rosetta Stone”

[13] The HoTT Book, Homotopy Type Theory