# Purely Symbolic Induction of Structure

Linas Vepštas[0000−0002−2557−740X]

OpenCog Foundation<linasvepstas@gmail.com>

**Abstract.** Techniques honed for the induction of grammar from text corpora can be extended to visual, auditory and other sensory domains, providing a structure for such senses that can be understood in terms of symbols and grammars. This simultaneously solves the classical "symbol grounding problem" while also providing a pragmatic approach to developing practical software systems that can articulate the world around us in a symbolic, communicable fashion.

## Introduction

The symbolic approach to cognition is founded on the idea that observed nature can be categorized into distinct entities which are involved in relationships with one another. In this approach, the primary challenges are to recognize entities, and to discover what relationships there are between them.

The recognition problem is to be applied to sensory input. That is, we cannot know nature directly, as it is, but only by means of observation and sensing. Conventionally, this can be taken to be the classical five senses: hearing, touch, smell, vision, taste; or, more generally, scientific instruments and engineered detectors. Such sensors generate collections of data; this may be time-ordered, or simply a jumbled bag of data-points.

Out of this jumble of data, the goal of entity detection is to recognize groupings of data that *always* occur together. The adverb "*always*" here is key: entities are those things that are not events: they have existence over extended periods of time (Heidegger's "Dasein"). The goal of relationship detection is to determine both the structure of entities (part-whole relationships) as well as events (statistical co-occurrences and causation). If one is somehow able to detect and discern entities, and observe frequent relationships between them, then the path to symbolic processing becomes accessible. Each entity can be assigned a symbol (thus resolving the famous "symbol grounding problem"), and conventional ideas about information theory can be applied to perform reasoning, inference and deduction.

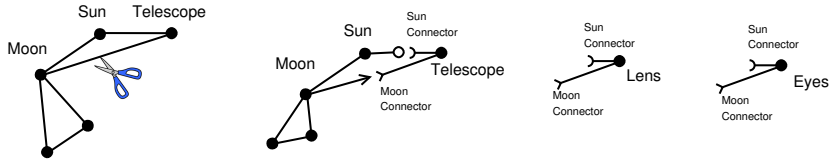
The goal of this paper is to develop a general theory for the conversion of sensory data into symbolic relationships. It is founded both on a collection of mathematical formalisms and also on a collection of experimental results. The experimental results are presented in a companion text; this text focuses on presenting the mathematical foundations in as simple and direct a fashion as possible.

In the first section, the general relationship between graphs and grammars is sketched out, attempting to illustrate just how broad, general and all-encompassing this is. Next, it is shown how this symbolic structure can be extended to visual and auditory perception. After this comes a mathematical deep-dive, reviewing how statistical principles

can be used to discern relationships between entities. Working backwards, a practical algorithm is presented for extracting entities themselves. To conclude, a collection of hypothesis and wild speculations are presented.

## From Graphs to Grammar

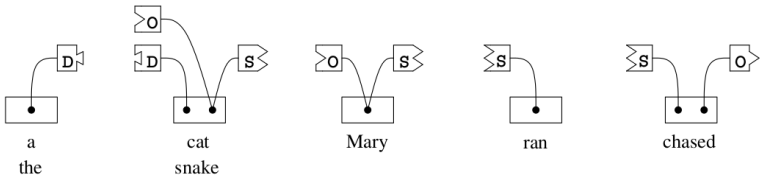
Assuming that sensory data can be categorized into entities and relationships, the natural representation is that of graphs: each entity is represented by a vertex, each relationship is represented by an edge. Vertexes are labeled with symbols, edges with symbol pairs. An example is illustrated below.




On the left is a conventional sparse graph of relationships between entities. On the right is the same graph, with some of the edges cut into half-edges, with the half-edge connectors labeled with what they can connect to. The connectors are drawn with distinct shapes, intended to convey what they are allowed to connect to. Such vertices, together with a collection of connectors, can be imagined to be jigsaw puzzle pieces, waiting to be connected.

The simplicity of the above diagram is deceptive. There is a deep and broad mathematical foundation: jigsaw pieces are the elements of a "monoidal category".[7] The connectors themselves are type-theoretic types. The jigsaw pieces are the syntactical elements of a grammar. These last three statements arise from a relatively well-known generalization of Curry–Howard correspondence: for every category, there is a type theory, a grammar and a logic; from each, the others can be determined.[2]

The jigsaw paradigm in linguistics has been repeatedly rediscovered.[11][13][4][23] The diagram below is taken from the first paper on Link Grammar.[17] Syntactically valid sentences are formed whenever all of the jigsaw connectors fully mated. This fashion of specifying a grammar may feel unconventional; such grammars can be automatically (i.e. algorithmically) transformed into equivalent HPSG, DG, LFG, *etc.* style grammars. Link Grammar is equivalent to Combinatory Categorial Grammar (CCG).[21]




### Compositionality and Sheaves

The naive replacement of entities by vertexes and relationships by edges seems to have a problem with well-foundedness. If an entity is made of parts, does this mean that a

vertex is made of parts? What are those parts made of? Is there an infinite regress? How might one indicate the fact that some entity has a composite structure? These questions are resolved by observing that a partially-assembled jigsaw puzzle resembles a singular jigsaw piece: it externalizes as-yet unconnected connectors, while also showing the connectivity of the assembled portions. Jigsaws resolve the the part-whole conundrum: the "whole" is a partially assembled jigsaw; the parts are the individual pieces. The way that an entity can interact with other entities is determined entirely through the as-yet unconnected connectors.

Sheaf theory[8] provides the formal setting for working with such part-whole relationships. The sheaf axioms describe how jigsaw pieces connect.[20] The appeal of sheaf theory is it's broad foundational and descriptive power: the sheaf axioms describe topology and logic (via the extended Curry–Howard correspondence mentioned above). Natural language can be taken in this broader setting.

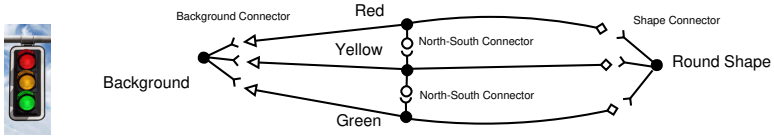
### Pervasiveness

After becoming familiar with the jigsaw paradigm, it becomes evident that it is absolutely pervasive. A common depiction of DNA uses jigsaw connectors for the amino acids ATGC. The antibody (immunoglobulin) is conventionally depicted in terms of jigsaws. Chemical reactions can be depicted as the assembly of jigsaw pieces.

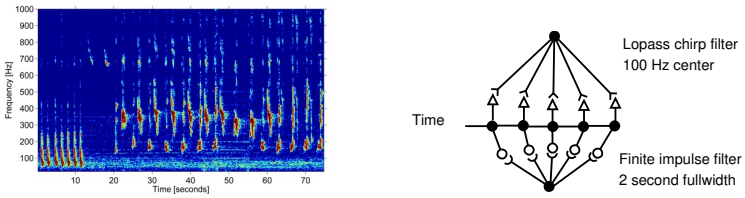
Composition (beta reduction) in term algebra can be seen as the act of connecting jigsaws. Consider a term (or "function symbol") $f(x)$ with typed variable $x$. Constants are type instances; for example, the integer 42. Beta reduction is the act of "plugging in": $f(x) : 42 \mapsto f(42)$. Re-interpreted as jigsaw connectors, the term $f(x)$ is a female-coded jigsaw , and 42 is a male-coded jigsaw. To connect, the types must match (the variable $x$ must be typed as integer). This kind of plugging-in or composition (with explicit or implicit type constraints) is rampant throughout mathematics. Examples can be found in proof theory,[19] lambda calculus,[3] term algebras[1] and model theory.[5]

### Vision and Sound

Shapes have a structural grammar, too. The connectors can specify location, color, shape, texture. The structural decomposition is that it is *not about pixels*! The structural decomposition is scale-invariant (more or less, unless some connector fixes the scale) and rotationally invariant (unless some connector fixes direction). The structural grammar captures the morphology of the shape, it's general properties It can omit details when they are impertinent, and capture them when they are important.




Audio data can also be given a jigsaw structure. On the left is a spectrogram of a whale song; time along the horizontal axis, frequency on the vertical, intensity depicted as a color.




A midsection of the song is shown as jigsaws: the number of repetitions (six), the frequency distribution (its a chirp, which can be discovered with a chirp filter.) Individual repetitions can be spotted with a finite impulse response filter. Sensory information can be described in grammatical terms.

## Symbolic Learning

In order for a graphical, sheaf-theoretic, grammatical theory of structure to serve as a foundation stone for AGI, there most be a practical algorithm for extracting structure from sensory data. This can be achieved in three steps. The first step is chunking (tokenization), the division of sensory data into candidate entities and interactions. The second step takes a collection of candidate graphs, splits them into jigsaw pieces, and then classifies jigsaw pieces into common categories, based on their commonalities. The third step is a recursive step, to repeat the process again, but this time taking the discovered structure as the sensory input. It is meant to be a hierarchical crawl up the semantic ladder.

Tokenization, induction of grammar, entity detection and predicate-argument structure have been experimentally explored in linguistics for decades; a review cannot be given here. What has been missing until now is a unified framework in which sensory (visual and audio) data can be processed on the same footing as linguistic structure. The OpenCog system, specifically the AtomSpace and the Learn project,[1] provide an implementation of that unified framework. Research has focused on the second step of the above algorithm; extensive research diaries log the results.[2] A summary of these results is presented as a companion paper to this one. Explorations of the first and third steps have hardly begun. It is easiest to describe the second step first.

### Grammatical Induction

In linguistics, one is presented with a tokenized sequence of words; the conversion of raw sound into phonemes and then words is presumed to have already occurred. The task is to extract a more-or-less conventional lexical grammar, given a corpus of text. This may be done as follows. First, perform a Maximum Spanning Tree (MST) parse; next, split the MST parse into jigsaw pieces; finally, classify those pieces into lexical vectors. The process is inherently statistical.

[1] See the "AtomSpace" and "Learn project" in github.
[2] See the diaries in the aforementioned project.

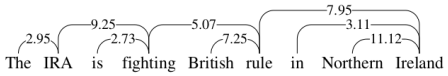
### Maximum Planar Graph Parsing

MST parsing is described by Yuret.[22] Starting with a corpus, maintain a count $N(u,w)$ of nearby word-pairs $(u,w)$. The frequentist probability $p(u,w) = N(u,w)/N(*,*)$ is the count of a given word-pair divided by the total count of all word-pairs. The star indicates a marginal sum, so that $p(u,*) = \sum_w p(u,w) = N(u,*)/N(*,*)$. The Lexical Attraction between word-pairs is

$$MI(u,w) = \log_2 \frac{p(u,w)}{p(u,*)\,p(*,w)}$$

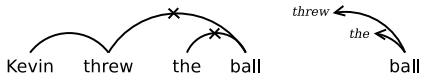
This lexical attraction is just the mutual information; it has a somewhat unusual form, as word-pairs are necessarily not symmetric: $(u,w) \neq (w,u)$. The MI may be negative! The range of values depends on the size of the corpus; for a "typical" corpus, it ranges from -10 to +30.

The MST parse of a sentence is obtained by considering all possible trees, and selecting the one with the largest possible total *MI*. The example below is, taken from Yuret's thesis. The numbers in the links are the MI between the indicated words.




Maximal planar graphs (MPG) (graphs with loops, but no intersecting links) appear to offer experimentally-observable advantages over trees, they constrain the grammar more tightly and offer advantages similar to those of catena-based linguistic theory.[14] MST parses are linguistically plausible: they correspond, more or less, to what trained linguists would write down for a parse. The accuracy is reasonably high. Perfect accuracy is not needed, as later stages make up for this. Yuret indicates that the best results are obtained when one accumulates at least a million sentences. This is not outrageous: work in child psychology indicates that human babies hear several million sentences by the age of two years.

**Lexical Entries** Given an MST or MPG parse, the lexis is constructed by chopping up the parse into jigsaw pieces, and then accumulating the counts on the jigsaw pieces. This is shown below.




Several kinds of notation are in common use such lexical entries. In tensorial notation, ball : $\left|\overleftarrow{\text{the}}\right\rangle \otimes \left|\overleftarrow{\text{throw}}\right\rangle$. In Link Grammar, `ball` : `the` − & `throw`−; the minus signs indicate connections to the left. The ampersand is the conjunction operator from a fragment of linear logic; it demands that both connectors be present. Linear logic is the logic of tensor algebras (by the aforementioned Curry–Howard correspondence.) Unlike tensor algebras, natural language has a distinct left-right asymmetry, and so the corresponding logic (of the monoidal category of natural language) is just a fragment

of linear logic. Note that all of quantum mechanics lies inside of the tensor algebra; this explains why assorted quantum concepts seem to recur in natural language discussions.

Connector sequences such as $\left|\overleftarrow{\text{the}}\right\rangle \otimes \left|\overleftarrow{\text{throw}}\right\rangle$ are disjoined in the lexis; each such sequence is called a disjunct. Given a word $w$, a lexical entry consists of all word-disjunct pairs $(w,d)$ together with their observed count $N(w,d)$. The normalized frequency is $p(w,d) = N(w,d)/N(*,*)$ where $N(*,*)$ is the sum over all word-disjunct pairs. A lexical entry is thus a sparse skip-gram-like vector:

$$\overrightarrow{w} = p(w,d_1)\,\widehat{e_1} + \cdots + p(w,d_n)\,\widehat{e_n}$$

The logical disjunction "or" can be used in place of the plus sign; this would be the "choice" operator in linear logic (as in "menu choice": pick one or another). The basis vectors $\widehat{e_k}$ are short-hand for the skip-gram disjuncts $\left|\overleftarrow{\text{the}}\right\rangle \otimes \left|\overleftarrow{\text{throw}}\right\rangle$.

**Similarity** The lexis generated above contains individual words with connectors to other, specific words. Taken as a matrix, the lexis is sparse but still quite large. To obtain a conventional grammar in terms of nouns, verbs and adjectives, dimensional reduction must be performed. This can be achieved by clustering with respect to a similarity metric. A conventional similarity metric is the cosine distance

$$\cos\theta = \overrightarrow{w}\cdot\overrightarrow{v} = \sum_d p(w,d)\,p(v,d)$$

As a metric, it fails, because the space spanned by these vectors is *not Euclidean space*! It is a probability space, with unit-length probability vectors: $1 = \sum_{w,d} p(w,d)$. The correct similarity is the mutual information:

$$MI(w,v) = \log_2 \frac{\overrightarrow{w}\cdot\overrightarrow{v}}{\left(\overrightarrow{w}\cdot\overrightarrow{*}\right)\left(\overrightarrow{*}\cdot\overrightarrow{v}\right)} \qquad \text{where} \qquad \overrightarrow{w}\cdot\overrightarrow{*} = \sum_d p(w,d)\,p(*,d)$$

Experimentally, the distribution of the MI for word pairs is Gaussian.[3] This is remarkable: it implies that the word vectors are uniformly distributed on the surface of a (high-dimensional) sphere: a Gaussian Orthogonal Ensemble (a spin glass).[18] In this sense, one can see that natural language is maximally disambiguating.

In this way, after transforming to a sphere, a plain cosine distance be used. The sphere vectors are given by $\overrightarrow{w} = \sum_v MI(w,v)\widehat{v}$. The center of the sphere must be subtracted, and the vectors normalized to unit length before taking a dot product.

**Classification** In practice, clustering is not straightforward. One wishes to first cluster the most frequent words first, whereas the highest MI pairs are very rare. This suggests defining a ranked-MI, adjusted by the average log frequency:

$$MI_{\text{rank}}(w,v) = MI(w,v) + \frac{\log_2 p(w,*) + \log_2 p(v,*)}{2} = \log_2 \frac{\overrightarrow{w}\cdot\overrightarrow{v}}{\sqrt{\left(\overrightarrow{w}\cdot\overrightarrow{*}\right)\left(\overrightarrow{*}\cdot\overrightarrow{v}\right)}}$$

Experimentally, this does not affect the shape of the Gaussian; it only shifts it to the right; one still has an orthogonal ensemble.

[3] See the Language Learning Diary Part Three, *op. cit.*

**Word-sense disambiguation** Words can have multiple meanings. Two words may be deemed to be similar, but not all of the disjuncts can be dumped into a common class; some of the disjuncts may belong to other word-senses. For example, a portion of the word-vector for "saw" can be clustered with other cutting tools, while the remainder can be clustered with viewing verbs. This presents a practical difficulty: off-the-shelf clustering algorithms cannot perform word-sense disambiguation

A further difficulty is that connectors must also be merged. The rewriting of connector sequences is subtle, as it affects word-vectors outside of those being merged (the merged connectors might appear anywhere). To maintain coherency, "detailed balance" must be preserved: the grand total counts must remain the same both before and after merge. The details of "detailed balance" can be found in the diary, *op cit.*

**Factorization** The clustering described above can be understood to be a form of matrix factorization. The word-disjunct matrix $p(w,d)$ is factorized into three matrices $LCR$ as

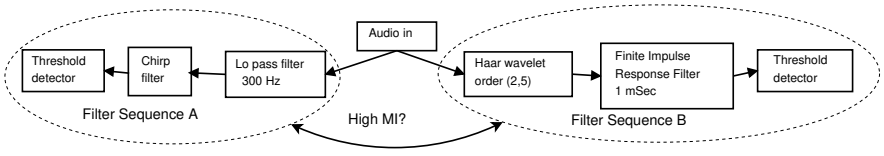
$$p(w,d) = \sum_{g,g'} p_L(w,g)\, p_C\left(g,g'\right) p_R\left(g',d\right)$$

where $g$ is a "word class" and $g'$ is a "grammatical relation". These two are commonly conflated in linguistic theory; here, they will be distinguished. The reason for this is that this is the *de facto* organization of the Link Grammar dictionaries for English, Russian and other languages. Examples of grammatical relations are "subject", "object", "modifier"; examples of word classes are "common noun" or "preposition". The matrices $L$ and $R$ are very sparse, which $C$ is compact, dense and highly connected. A sense of the scale of factorization can be obtained from the hand-curated English-language dictionary. It consists of about 100K words, 2K word classes, several hundred grammatical relations (LG "macros") and 30 million disjuncts. In other words, the central component is quite small.

Factorization provides an aid to interpretability. Instead of a morass of matrix elements, word-classes (nouns, verbs) are recognizable as such. So are relationships (subjects, objects). This is the power of a symbolic, lexical approach.

## Chunking/Tokenization

The relatively straightforward tokenization of written English hides the difficulty of chunking in general. How can one obtain a comparable chunking of raw audio or visual data? The goal is to obtain, by automatic means, a sequence of transducers, from sounds to phonemes and syllables and words.




A pair of transducers in block-diagram form is shown. Each block corresponds to a digital signal processing (DSP) function. The generation of such sequences can be

managed through genetic program (GP) learning techniques. An example of a GP system is provided by MOSES.[9,10] Given a collection of "okay" filter sequences, GP can explore both the parameter space to provide a better tuning, and, by means of mutation and cross-over, generate other filter sequences. The goal is to find high-quality "feature recognizers", indicating the presence/absence of a salient feature in the sensory environment.

Learning in GP systems is guided by maximizing a utility (scoring) function. But what should that function be, in an unsupervised setting? Just as one discovered structure in language through entropy maximization, one can use the same ideas here. For all features (filter sets) currently under consideration, one looks for high-MI correlations. Features that are poorly detected have poor correlation and low information content; crisp recognizers should be sharply correlated.

### The Symbol Grounding Problem and the Frame Problem

An old problem in philosophy (dating back to Socrates) is the symbol grounding problem.[4] When one says the word "chair", what does that mean? One can attempt to make extensional lists of things one can sit on, but that list can never be complete. One can make intensional lists of the properties of a chair; such a list invariably fails to encompass all the possibilities. A third possibility is that of affordances: what must an object be like, to be sit-on-able? The DSP filter sequence presented above is precisely an affordance-detector.

Consider a simpler example. If someone says "I hear whistling in the distance", what does the word "whistling" actually mean? How to describe it? What is the grounding for the symbol "whistling"? Filter sequences explicitly manifest the grounding. "Whistling" is a certain kind of hi-pass filter attached to a chirp filter with a certain finite impulse response time. That is what "whistling" is. What else could it possibly have been?

The Frame Problem posits that there are too many objects, features and events in the environment to be able to pay attention to all of them, especially as almost all are irrelevant to the current focus. This is precisely the problem that the entropy-maximizing training of filter sequences is doing: mutual information tells you what things "go together", what things are relevant. The grammatical structure reveals *how* those things depend on one-another. The vast ocean of sensory stimulus is reduced to a trickle of symbolic relationships, arriving either in a regular, expected pattern (and thus perhaps unimportant), or arriving in unexpected, surprising ways. The surprise is what demands attention; the rest is but background noise.

## Abstraction and Recursion

The above presented techniques for moving from sensory input to the lower reaches of semantics. Can one go farther, and arrive at common-sense reasoning, one of the Holy Grails of AGI? The author wishes to argue that the techniques described above are

[4] See the Stanford Encyclopedia of Philosophy, "Frame Problem" and "Embodied Cognition".

sufficient to reach up into the highest levels of abstraction and general intelligence. It is a ladder to be climbed, repeating the same operations on each new layer of abstraction.

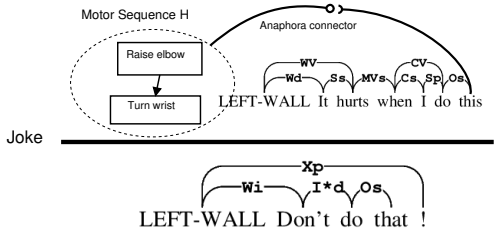
The next few rungs of the ladder can be found in linguistics. The MST parsing algorithm given above was presented at the word-pair level. When applied at the semantic level, it becomes the Mihalcea algorithm.[12]

In lexical semantics, there is an idea of "lexical implication rules"[15]. These are rules that control how words used in one context can be used in a different context. The discovery of these rules be automated: each rule has the form of a jigsaw, and the algorithm for inferring jigsaws has already been presented. Jigsaw assembly is parsing: given a set of constraints (for example, a sequence of words) parsing is the act of finding jigsaw pieces that fit the word-sequence. Parsing technologies, and their more general cousins, the theorem-provers, are well-understood.

Lexical implication rules generalize to the "lexical functions" (LF) of Meaning-Text Theory (MTT).[6] The MTT is a well-developed theory of the "semantic" layer of linguistics, sitting atop surface syntax. An algorithm for learning LF's is described by Poon & Domingos[16]. The relationship to the current work is obscured by their use of jigsaws written as lambdas; rephrasing as jigsaws makes it clear that it is just a hunt for equivalent jigsaw sub-assemblies (synonymous phrases). Anaphora resolution, reference resolution and entity detection are well-explored topics in computational linguistics. The jigsaw metaphor demonstrates precisely how one can climb the rungs of the ladder: from pair-wise correlations up to grammars. In the presence of a grammar, we once again know what is ordinary, and can then renew the search for surprising pair-wise correlations, this time at the next layer of abstraction

### Common Sense

Can this be used to learn common sense? I believe so. How might this work? Let me illustrate by explaining an old joke: "Doctor Doctor, it hurts when I do this! Well, don't do that!". The explanation is shown below, in the form of a rule, using the notation from proof theory. The thick horizontal bar separates the premises from the conclusions. It is labeled as "Joke" to indicate what kind of rule it is.




The "sequent" is the anaphora connector, which connects the word "this" the a specific motor sequence. Which motor sequence? Well, presumably one that was learned, by automatic process (perhaps GP), to move a limb. All of the components of this diagram are jigsaw pieces. All of the pieces can be discovered probabilistically. All of the connectors can be connected probabilistically. The learning algorithm shows how to discern structure from what is superficially seems like a chaotic stream of sensory input. Common sense can be learned.

## References


1. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
2. John C. Baez and Mike Stay. Physics, topology, logic and computation: A rosetta stone. *Arxiv/abs/0903.0340*, 2009.
3. H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*. North-Holland, 1981.
4. Bob Coecke. Quantum links let computers read. *New Scientist*, December 2010.
5. Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.
6. Sylvain Kahane. The meaning-text theory. *Dependency and Valency. An International Handbook of Contemporary Research*, 1:546–570, 2003.
7. Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1978.
8. Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic*. Springer, 1992.
9. Moshe Looks. *Competent Program Evolution*. PhD thesis, Washington University St. Louis, 2006.
10. Moshe Looks. Meta-optimizing semantic evolutionary search. In Hod Lipson, editor, *Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, July 7-11, 2007*, page 626. ACM, 2007.
11. Solomon Marcus. *Algebraic Linguistics; Analytical Models*. Elsevier, 1967.
12. Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 411–418, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
13. EA Nida. The molecular level of lexical semantics. *International Journal of Lexicography*, 10:265–274, 1997.
14. Timothy Osborne, Michael Putnam, and Thomas Groß. Catenae: Introducing a novel unit of syntactic analysis. *Syntax*, 15:354–396, 2012.
15. N. Ostler and B.T.S.Atkins. Predictable meaning shift: Some linguistic properties of lexical implication rules. *Proceedings of the First SIGLEX Workshop on Lexical Semantics and Knowledge Representation*, 1991.
16. Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore, August 2009. Association for Computational Linguistics.
17. Daniel Sleator and Davy Temperley. Parsing english with a link grammar. Technical report, Carnegie Mellon University Computer Science technical report CMU-CS-91-196, 1991.
18. Michel Talagrand. *Mean Field Models for Spin Glasses*. Springer-Verlag, 2010.
19. A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory, Second Edition*. Cambridge University Press, 1996.
20. Linas Vepstas. Sheaves: A topoloogical approach to big data. ArXiv abs/1901.01341, 2017.
21. Linas Vepstas. Combinatory categorial grammar and link grammar are equivalent. https://github.com/opencog/atomspace/raw/master/opencog/sheaf/docs/ccg.pdf, 2022.
22. Deniz Yuret. *Discovery of Linguistic Relations Using Lexical Attraction*. PhD thesis, MIT, 1998.
23. William Zeng and Bob Coecke. Quantum algorithms for compositional natural language processing. *Electronic Proceedings in Theoretical Computer Science (EPTCS)*, 221, 2016.