Messaging

March 8, 2020

Abstract

Thoughts about messaging and a relationship to grammar. Some old ideas, rephrased, some new ideas, incomplete.

Introduction

Two almost inrelated ideas, and one theme.

Lets start with an obvious observation: natural language is used to transmit messages, from one human mind to another. Language carries messages. Painfully obviously. Now pair this with another idea: message-passing algorithms are a good way of solving NP-complete graphical constraint satisfaction problems. This pairing suggests a wild insight: that a collection of human minds, working together, are using message passing as a technique for collectively solving some difficult problem. To what degree can this insight be developed into a formal theory of the mind, and speciffically, a formalm theory of the collective, social mind? I don't know, and what follows below will be mostly unrelated to that, and will instead tackle grammar again, the grammar of natural language.

Grammar via Beleif Propagation

This is a reprise of a recurring theme. The goal is to find effective and tractable computational strategies for extracting meaning from natural language. The current work concerns itself with a very basic layer, that of extracting a lexical grammar, describing the syntax of the language, and a crude level of semantics that follows therefrom. By a "lexical grammar", it is meant that sentences of the language can be broken down into words, and that the relatioinship between words can be obtained from a lexicon, that is, from a dictionary where each word can be looked up to discover the grammatical relationships that word can engage in. It is useful to note that such lexicons are redily extended to include idioms, set phrases, institutional expressions, colocations; such multi-word constructions do not alter the underlying concept.

Lexicality implies that language can be analyzed in terms of words (or set phrases, *etc.*). But language is also fundamentally statistical and probabilisitic: there is no ultimate, final truth to syntax and semantics, but only likely meanings and interpretations. In this setting, lexicality means not only that language

can be viewed as a graph of relationships between words, but also that the graph can be factored into local components. Specifically, each local component consists of a word, and the other nearby words that it may interact with: a word, and its syntactico-semantically-nearest neighbors.

Modern probability theory has a standard formulation using the terminology and notation of statistical mechanics. In this formulation, one begins by asserting that the universe is described by a summation over all possibilities: everything that might happen, can happen, with some associated probability. This sum is called the partition function; it is symbolized by Z, and the partitioning is simply the statement each possibility has a probability. For natural language, this just means that every possible sequence of words (a "sentence") occurs with some probability; ungrammatical sentences have a low, approximately vanishing probability.

It is a theorem of Boltzmann that paritition functions can be written as sums over exponentials, and that the most likely possibility is given by maximizing the entropy. This is not an assumption that has to be artificially forced onto the system; rather, it is the factual statement that, if you beleive in probability, then there is no other way: it is a theorem. Combining natural language with probability then suggests that it is fruitful to articulate the statistical mechanics thereof.

In what follows, the formal grammar of choice is Link Grammar.[1, 2] This choice is made for several reasons. First, one may argue that the actual choice of a grammar formalism is immaterial, as all grammars are effectvely interconvertable between one-another by algorithmic means. Thus, the choice of formalism boils down to convenience; what notational system is most convenient? Here, Link Grammar stands out. First, it is effectively a form of dependency grammar, and so is natural to linguists trained in that tradition. Second, by expressing grammar in terms of link types, it leverages type theory, and has a very natural bridge to categorial grammars and pregroup grammars: link types are just the type-theoretical types of the relationships that categorial grammars articulate. As categorial grammars are normally considered to be a form of phrase-structure grammars, exposing the relationships as types provides the natural bridge between the phrase-structure and dependency grammar schools of thought. Finally, Link Grammar is appealing from the tradition of mathematics: Link Grammar is a tensor algebra. Lexical entries are tensors, and lexical entries are composed into sentences in exactly the same way that one composes tensors in a tensor algebra. It is precisely this tensorial nature that then enables Curry–Howard correspondance; Link Grammar is the "internal language" of monoidal categories.

The next section articulates the statistical mechanics Link Grammar. This presents Link Grammar as both a constraint-statistfaction problem as well as a maximum entropy problem. This is followed by a section looking at the belief-propagation aglorithm, inspired by and devloped along the lines described by Mézard and Mora[3].

A Frequentist Model of Language

The goal of this section is to formulate a model of language, simultaneously invoking both its graph-theoretical and statistical properties. This mostly requires the introduction and review of fairly mainstream ideas and notation, and an articulation of the notation so that it's meaning becomes clear.

Consider first a word sequence $\underline{w} = (w_1, \cdots, w_n)$ which can be taken to be a sentence that is n words long; it need not be grammatical; that is, it need not be a valid sentence. One possible way of defining the statistics of langauge is to claim that the probability of this word sequence is

$$P(\underline{\mathrm{w}}) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \delta(\underline{\mathrm{w}}, \underline{\mathrm{w}}_i) \tag{1}$$

where N is the number of sentences in a sample corpus representative of the langauge, so that each $\underline{\mathrm{w}}_i$ is a grammatically valid sentence. The above states that, basicly, a word sequence $\underline{\mathrm{w}}$ is valid if and only if it occurs in the sample corpus; else it is not. This is a naive and simplistic definition of language, dounded on a frequentist view of probability. It is inadequate on multiple fronts. It's worth articulating these, lest they become an impediement later.

- The notation above assigned a fixed sentence length of n. This seems to be a notational inconvenience, rather than a fundamental limitation. From here on, sentences are assumed to be varying in length, and can be chosen as desired.

- Morphology is ignored. For the most part, this should not be an impediment; Link Grammar is able to deal adequately with morpho-syntax, and even enforce phonetic agreement, so this presents no particular stumbling block.

- Semantic structure on a scale larger than one sentence is ignored. Most of what follows will be focused on syntax, and the lower reaches of semantics, and so this simplification seems reasonable at this time.

- Corpora are assumed to be finite in size. This is naturally the case for natural langauge, but it can cause difficulties for certain mathematical approaches, which are more naturally expressed in the continuum limit. This is also glossed over, as it rarely presents any practical difficulties. By contrast, formal langauges with generative grammars are capable of producing infinite corpora, and so the distinction between finite and infinite can be blurred.

- The above definition completely ignores the obvious fact that language is compositional: one can form sentences from sentence fragments; phrases can have meanings; language is a collection of recurring word-patterns plainly visible at a sub-sentence level. Yet, one of the goals of research into linguistics is to elucidate teh compositional structure of language.

For all of these various reasons, the above probabilistic description of language is patently absurd. Yet it is often quoted as a foundational cornerstone, and so is worth repeating here. Despite these abusrdities, one might like the final formulation of language be such that eqn 1, or at least something similar, can arise naturally from the theory.

For the remainder of this section, the assumption is made that the compostional nature of langauge can be adequately captured by means of a lexis. This is closely related too the assumption that language is syntactic, but is, in some sense, strictly weaker, or, at least, should be interpreted in a broader setting. A lexical description of langauge is one where each word is associated with a collection of properties and relations that specify how that word can occur in grammatical contexts, and how rough, basic meaning can be pinned down. This is in contrast to "syntax", which is usally taken to be synonym for the existence of a (planar) parse tree. The point here is that the graphical structure of languge need not be tree-like: the parse may contain loops; it may contain non-planar edges between words, and it may contain relations operating at different conceptual levels. For example, graphical relations enforcing phonetic agreement might typically operate at a different level than count and tense agreement, and that in turn operates at a different level than anaphora agreement.

It is possible that natural language has structure that cannot be captured by probabilistic graphical representations. That author, however, is currently unable to imagine what this might be. Therefore, in all that follows, the assumption is made that the entirety of meaning and structure in language can be captured by graphs that encode functions and relations with statistical, probabilistic properties.

Link Grammar

The remainder of this paper asumes that Link Grammar is well suited to capture most of the lexical, syntactic properties of language. There are multiple reasons that this assumption seems justified. These include:

- The Link Grammar formalism can be more-or-less directly related to dependency grammars; it can be taken as a certain kind of dependency grammar, and is rich enough that it can be mapped or translated to different styles of dependency.

- The Link Grammar formalism can be directly related to categorial and pregroup grammar-style grammars; insofar as those can be taken as examples of phrase-structure grammar, a route exists to map Link Grammar into phrase structure, and the kinds of phenomena exhibited there.

- Link Grammar bridges naturally to Lambek calculus. The link types of Link Grammar can be interpreted as the types of type theory; the "disjuncts" of Link Grammar are manifestly tensorial, and so Link Grammar can be taken as a kind of tensor algebra. As such, it can be understood via category theory: Link Grammar is the "internal language" of a monoidal

category. This makes it simultaneously very general, and abstractly powerful.

To summarize, the Link Grammar formalism lies at the nexus of a multitude of different viewpoints and theories of language. One need not go very far, or work particularly hard, to see how it captures different linguistic (and mathematical) phenomena and theories.

Statistical Mechanics of Link Grammar

Lexical entries are then of the form

$$((w, d), h)$$

where w is a word, d is a disjunct[1] describing one of the grammatical relationships the word can engage in, and h is a "cost" or entropy associated with this word-disjunct pair.[2] It is convenient to write the cost as a function of the word-disjunct pair: $h = h(w, d)$ as only the lowest cost is meaningful, and it does not make sense for a given disjunct to have more than one cost. Impossible word-disjunct pairs have infinite cost, rendering thier probability zero. That is, the probability of a word-disjunct pair, up to an overall normalization, can be taken as

$$P(w, d) \sim \exp -h(w, d)$$

The probability of a word sequence $\underline{w}$ is then

$$P(\underline{w}) = \frac{1}{Z} \sum_{(d_1, \cdots, d_n)} \prod_{j=1}^{n} P(w_j, d_j) \, \Delta(d_1, \cdots, d_n)$$

where the formal grammar is encoded as a boolean satisfiability factor[3]

$$\Delta(d_1, \cdots, d_n) = \begin{cases} 1 & (d_1, \cdots, d_n) \text{ is a valid parse} \\ 0 & \text{otherwise} \end{cases}$$

The probabilistic aspects, that some parses are more likely than others, are encoded in the lexical factors $P(w, d)$. Replacing probabilities by thier logs, one can equivalently write

[1] A definition of "disjunct" will be given shortly; the next few paragraphs do not rely on a precise defintion, and hold true generally for any lexical formulation. In particular, one can imagine that a disjunct is an n-gram or a skip-gram; this is useful, as it makes contact with nueral-net/deep-learning approaches to language.

[2] A single word might be associated with multiple disjuncts. During grammatical analysis, only one of these may be used at a time; thus the disjuncts are disjoined from one-another, whence the name.

[3] This is just the indicator function for a predicate. Common alternate notation is $\mathbb{I}$ or $\mathbf{1}$. There is an implicit assumption that a parse, if it exists, is unique. If this is not the case, and multiple distinct parses exist with a given fixed $\underline{d}$, then Δ needs to count these with multiplicity.

$$P(\underline{\mathrm{w}}) = \frac{1}{Z} \sum_{\underline{\mathrm{d}}} \Delta(\underline{\mathrm{d}}) \exp -\mathcal{A}(\underline{\mathrm{w}}, \underline{\mathrm{d}})$$

using as shorthand $\underline{\mathrm{d}} = (d_1, \cdots, d_n)$ as the string of n disjuncts associated with the n words $\underline{\mathrm{w}} = (w_1, \cdots, w_n)$. The sum is taken over all possible lists $\underline{\mathrm{d}}$ of disjuncts d_j; the Δ term excludes those sequences that do not correspond to valid parses in the formal grammar.[4] The normalization Z is famously known as the partition function.[5]

The $\mathcal{A}$ is the "action"[6], given by

$$\mathcal{A}(\underline{\mathrm{w}}, \underline{\mathrm{d}}) = \sum_{j=1}^{n} h(w_j, d_j)$$

As the cost h is infinite for those words-disjunct pairs that do not occur in the lexis, this sum excludes impossible pairings; no addtional effort is needed to otherwise exclude them.

Although the above formulation uses the word "disjunct" for d_j, it was intentionally vague; none of the development required a more precise definition. In a lexical formulation, the term $\Delta(\underline{\mathrm{d}})$ can also be factorized locally, as it is determined by a product of lexical elements. In the Link Grammar formulation, the lexical elements are explicitly graphical: a vertex surrounded by half-edges or connectors, with two half-edges required to make a full edge (link) connecting two vertexes. Algebraically, a disjunct is a list of connectors, which either can connect, or not. That is, an arity-m disjunct is a conjunction of connectors c written as

$$d = c_1 \& \cdots \& c_m$$

with each connector being a half-link, that is, a link with a direction indicator:

$$c = (\ell, \sigma)$$

with $\ell \in L$ being one of the Link Grammar link types, and $\sigma \in \{-, +\}$ being a direction indicator.[7] Define the conjugate direction indicator $\bar{\sigma}$ as

$$\bar{\sigma} = \begin{cases} - & \text{if } \sigma = + \\ + & \text{if } \sigma = - \end{cases}$$

[4] The summation $\sum_{\underline{\mathrm{d}}} \Delta(\underline{\mathrm{d}})$ has the form that makes it clear that Δ has the form of an integration measure. An obvious generalization is to replace it by a fuzzy or fractional measure.

[5] See Wikipedia.

[6] This word comes from physics, specifically, the Lagrangian formulation of classical mechanics.

[7] In standard link grammar, these direction indicators point to the left and right, respecively, and encode the directional dependence of word order. For languages with free word-order, it is convenient to use symbols for head and tail instead of, or in addition to the direction indicators. Head/tail markings are also useful for indicating dependency directions, when these are desired, or for indicating catena.

Thus, two connectors c_a and c_b connect if the link types match, and the direction indicators are conjugate. Thus, one may define

$$\delta(c_a, c_b) = \delta(\ell_a, \ell_b)\,\delta(\sigma_a, \bar{\sigma}_b)$$

A sentence is parsable when all connectors are connectable, and so

$$\Delta(\underline{d}) = \Delta(d_1, \cdots, d_n) = \prod_{d_j=(c_{j1}\&\cdots\&c_{jm})} \delta(c_{jk}, c_{j'k'}) \tag{2}$$

with the product being over all of the individual connectors in each disjunct. That is, $\Delta = 1$ if and only if every connector can be uniquely paired with some other connector, and no dangling connectors remain. To avoid double-counting, the product is meant to extend only over the links (edges) in the graph, with one term per edge.

It is from this that the interpretation as a tensor algebra arises: Each disjunct $d = d_j$ can be thought of as a tensor having m indexes on it; each index must be contracted with some other index on some other tensor. The tensor indexes are always contracted pair-wise, and once connected (consumed) cannot be connected to any other index.[8] A parse is valid if and only if Δ is a scalar, having no remaining uncontracted indexes. Unlike symmetric tensor algebras, the connectors are directional: they can contract only to the left, or to the right.

Implicit in the above is a further constraint that the parse graph be a planar graph, *i.e.* that there is a no-links-crossing constraint. This constraint is very useful for controlling the combinatorial explosion of possible parses; unfortunately, it is a non-local constraint, and thus cannot be easily written in a factorizable manner. Rather than tackling the difficulty of obtaining an adequate notation for such a non-local constraint, it is easier, for now, to implicitly keep this in the background. There are multiple techniques that can be used when dealing with this; these, and planarity in general, are for now secondary concerns.

Taking the logarithm, so as to turn products into sums, the constraint can be written in the form

$$\Delta(\underline{d}) = \Delta(d_1, \cdots, d_n) = \exp - \left[\sum_{d_j=(c_{j1}\&\cdots\&c_{jm})} \Xi(c_{jk}, c_{j'k'}) \right]$$

where Ξ can be interpreted as a kinetic term, having a value of zero when c_{jk} can be contracted with $c_{j'k'}$ and is infinite otherwise.

In this form, the constraint can be pulled into the action $\mathcal{A}$, redefining it as a summation of local interactions:

$$\mathcal{A}(\underline{w}, \underline{d}) = \sum_{j=1}^{n} \left[h(w_j, d_j) + \sum_{d_j=(c_{j1}\&\cdots\&c_{jm})} \Xi(c_{jk}, c_{j'k'}) \right]$$

[8] This is the content of the "no cloning" theorem.

The intended reading of the above expression is that it is a summation over Feynmann diagrams, with h corresponding to a m-point vertex (when the disjunct d_j has arity m), and the Ξ terms corresponding to propagators connecting vertexes. The propagators are exactly the Link Grammar links, weighted in such a way that they contribute zero to the action, when the link is allowed, and otherwise contributing infinity.

The partition function can then be formally written as

$$Z = \sum_{\underline{\mathrm{w}},\underline{\mathrm{d}}} \exp -\mathcal{A}\left(\underline{\mathrm{w}},\underline{\mathrm{d}}\right)$$

with the summation over $\underline{\mathrm{w}}$ running over all possible word-sequences of arbitrary length. As is conventional in partition function formulations, it is convenient to introduce external currents $\underline{\mathrm{J}}$ so that variational principles can be used to extract quantities of interest. Thus, for example, writing

$$Z\left[J\right] = \sum_{\underline{\mathrm{w}},\underline{\mathrm{d}}} \exp -\mathcal{A}\left(\underline{\mathrm{w}},\underline{\mathrm{d}}\right) + \underline{\mathrm{J}} \cdot \underline{\mathrm{w}}$$

and taking a variation δJ along the direction $\underline{\mathrm{w}}$, the limit of the logarithmic derivative gives the probability:

$$P\left(\underline{\mathrm{w}}\right) = \frac{1}{Z} \left.\frac{\delta Z\left[J\right]}{\delta J}\right|_{J=0} = \frac{1}{Z} \sum_{\underline{\mathrm{d}}} \exp -\mathcal{A}\left(\underline{\mathrm{w}},\underline{\mathrm{d}}\right)$$

The "current" J is just an algebraic device, a trick, used to single out one particular word-sequence out of the infinite sum. It is conventient to also introduce other currents coupling to other parts of the action, so that the variational principle can be used to extract other quantities of interest. The logarithm of the partition function $-\ln Z$ is the free energy; standard algebraic variations can be used to extract an entire zoo of "thermodynamic" variables, including correlation functions, mean values, mean-square deviations and the like.

Grammar via Beleif Propagation

The learning task begins by noting that the valid disjuncts are not known *a priori*, they must be discovered. To accomplish this, [4]

Its a constraint-satisfaction problem. Equation 2 defines a factorization of the constraints, that is, a factor-graph. For a given parse, the factor graph is a bipartite graph, connecting elements $d \in D$ (the disjuncts) to elements $\ell \in L$ (the links). In the factor graph, each link ℓ corresponds to a vertex, and, of course, links are always arity-2. To be precise, the vertexes of the factor graph are taken from $V = D \cup L$ and the edges are taken from $E = \{(d,\ell)\,|d \in D \text{ and } d = c_1 \& \cdots \& c_m \text{ and } c_k \in \ell\}$. This last just states that an edge in the factor graph must connect some connector (half-link) on a disjunct to the link (as links are just pairs of half-links).

XXX Need figure here XXX.

Here are the belief-prop eqns:

The first step is to replace each possible connection-pair $\delta(c_{jk}, c_{j'k'})$ by a belief of the possibility $p(c_{jk}, c_{j'k'})$ of such a connection being present, interpreted as a probability: $0 \leq p \leq 1$ with the goal of eventually driving each connectable pair to be zero or one. Similarly

x

if only one link-type, then its an unlablled dependency parse, and the solution is mean-field or Markovian. To make it tractable use the page-rank algorithm...

x

x but the shafiness of it all ....

x

x if multiple link types....

x

The other thing to point out is due to loops, etc. it won't be naive beleif propagation, but it will be Survey Propagation, as mentioned there...

The End

References

[1] Daniel Sleator and Davy Temperley., *Parsing English with a Link Grammar*, Tech. rep., Carnegie Mellon University Computer Science technical report CMU-CS-91-196, 1991, URL `http://arxiv.org/pdf/cmp-lg/9508004`.

[2] Daniel D. Sleator and Davy Temperley, "Parsing English with a Link Grammar", in *Proc. Third International Workshop on Parsing Technologies*, 1993, pp. 277–292, URL `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/link/pub/www/papers/ps/LG-IWPT93.ps`.

[3] Marc Mézard and Thierry Mora, "Constraint satisfaction problems and neural networks: a statistical physics perspective", , 2008, URL `https://arxiv.org/abs/0803.3061`, arXiv abs/0803.3061.

[4] A. Braunstein, et al., "Constraint Satsifaction by Survey Propagation", *Advances in Neural Information Processing Systems*, 9, 2002, URL `https://arxiv.org/pdf/cond-mat/0212451.pdf`.