Factoring with Statistical Linguistics

Linas Vepštas

January 2023

Abstract

An attempt is made to develop a mathematical formalism for factoring large language graphs into factors that have a symbolic interpretation. The OpenCog language learning effort has been attempting to induce grammar, syntax and semantics from corpora. Most of this work is purely experimental, "seat of the pants" exploration. This document attempts to provide a mathematical foundatation for that work, thus perhaps making it clearer and easier to grasp.

The primary focus is on exploring the nature of probability in extremely high-dimensional spaces ("hyperspaces"), and how traditional linguistic ideas can be applied to factorize probability distributions into components.

Introduction

A probabilistic description of natural language posits that probability theory can be validly applied to word-sequences. Given a sequence of words $(w_1, w_2, \cdots, w_n)$ representing a sentence, a paragraph, or a longer text..., one make the *a priori* assumption that it is possible to assign a probability $p(w_1, w_2, \cdots, w_n)$ to this sequence. It is not philosophically or scientifically obvious that this is a valid assumption: the collection of say-able sentences is presumably infinite; language changes over time; ever human speaker internalizes slightly different grammars and idiomatic expressions. Vocabularies are different for technical texts and literary texts. While it is true that present-day computers can gather up billions of sentences by scraping the web, it can hardly be assumed that these are converging to some overall stable probability distribution $p(w_1, w_2, \cdots, w_n)$. Thus, assuming that $p(w_1, w_2, \cdots, w_n)$ exists is intellectually dangerous.

None-the-less, we make this assumption, for two reasons. First, it is useful. Second, if a specific finite-sized corpus is selected and fixed, it is a simple and unambiguous matter of counting words and phrases to obtain frequency distributions. However, even in this case, one should not be naive: the probability space of of all sentences in a modest sized corpus is immense. It would be nice if one could work with smaller factors. Linguists have already exposed what these factors could be: nouns, verbs, grammatical relationships. One goal of this text is to formalize the relationship between grammar and probability spaces, using mathematical notation rather than hot air. It is hoped this will make things clearer. Another goal is to extend this analysis into the domain of semantics and "common sense". This second goal won't be met, other than

to suggest that exactly the same methods that allowed low-level syntactic factorization to be performed, can also be applied, again, at more abstract levels. A third goal is to use this mathematical machinery to guide the development of software for performing this analysis, to guide future experiments, and to provide a better theoretical foundation for what has so far been a seat-of-the-pants effort.

Parsing

The seat-of-the-pants effort so far has been focused on the automated extraction of a grammar from a text corpus. Issues of text segmentation are completely avoided: it is assumed that the corpus consists of words, unambiguously separated by blank spaces. These are avoided because segmentation is a deep, difficult and interesting problem, and tackling it takes us afield. Likewise, issues of morphology are also ignored. Both of these are fundamentally important. It is hoped (believed by the author) that the techniques described here will also be applicable there. But for now, its easiest to presume that there is a text corpus, consisting of well-defined "words".

One statistical approach to parsing is to simply count word-pairs in the sample corpus, and then to compute the pairwise point mutual information $MI(u,w)$ for all pairs. This mutual information can be used to create a Maximum Spanning Tree (MST) parse: to consider all possible trees spanning all words in a sentence (or block of text) and then select the one that maximizes the grand total MI, summed over the word-pairs in the tree. It is also useful to consider the Maximal Planar Graph (MPG) parse: starting with the MST tree, add edges to create cycles (loops), while still maximizing the total MI. That such MST parses correspond to reasonable linguistic structure has been widely explored over several decades.

A grammar can be extracted by taking such MST/MPG parses and cutting each edge in half, and retaining, as a "connector label", what word that edge used to connect to. The result of such chopping-up are the so-called "disjuncts" or "jigsaw puzzle pieces" of Link Grammar. These can be reassembled again, to obtain syntactic parses of sentences. Link Grammar works: there are extensive hand-curated dictionaries for English, Russian and Thai, with smaller dictionaries for another dozen natural languages. The English dictionary might be the most accurate/sophisticated parsing system currently available.

Link Grammar grammars can be converted to other formalisms; *e.g.* Head-Phrase Structure Grammar (HPSG) and so on. It can be shown that Link Grammar is "isomorphic" to Combinatory Categorial Grammars (CCG). The quotes around "isomorphic" have less to do about the math, than what a typical linguist might find acceptable in the mapping. For the remainder of this text, we assume that any grammar formalism is acceptable, and that they are all inter-convertible, interchangeable with one another, at least weakly, if not strongly. The goal of this text is to expose the relationship between statistics and grammar, rather than to quibble the finer points of linguistics. When the text below says things like "a relationship $r(w_1,w_2,w_3)$ between three words (w_1,w_2,w_3)" you are free to imagine any grammar formalism that you wish, involving subjects, verbs and objects and so on. However, Link Grammar will remain the touchstone, as it is the most compatible with probability theory. Thus, a general acquaintance with Link Grammar is strongly recommended.

Literature Review

The goal of the present text is to talk about the factorization of graphs, in general. There has been, of course, much related prior work.

The idea of statistical parsing[1] has been around for decades. Among the earliest work is Charniak's Maximum Entropy Parser.[1] No lit review, XXX reference Wikipedia instead?

The idea of matrix factorization is central large consumer businesses, who wish to estimate future shopping patterns as a function of prior behavior. Vast numbers of papers have been written...

Hypervectors are a relatively newer approach to computing ...

Factorization

The notion of factorization is to take some large blob, and pick it apart into components: to factor a matrix into block-diagonal components, to factor an integer into primes. Probability distributions over statistically independent variables factorize trivially: this is what is meant by the words "statistically independent". Probability distributions over language are not, of course, statistically independent, and thus are not strictly factorizable. None-the-less, they are almost so; the goal is to identify the strongly connected components and separate them from one-another, identifying the weaker connections.

Lets try to capture this idea using mathematical notation. To recap the story so far: Let $p(w_1, w_2, \cdots, w_n)$ be the probability of observing n words in a sentence (or block of text). The space of sequences $\{(w_1, w_2, \cdots, w_n)\}$ is a Cartesian product space, and p is a measure upon that space.

The goal of factorization is to approximate this measure by factorizing it into parts, where the parts are given by parsing via conventional linguistic theory. That is, we presume that relations r_i between small sets of words can be found, such that the following holds, approximately:

$$p(w_1, w_2, \cdots, w_n) \approx p(r_1\{w\})\, p(r_2\{w\}) \cdots p(r_k\{w\})$$

where $r_1, r_2, \cdots, r_k$ are syntactic relations (subject, verb, object...) and the $\{w\}$ are the set of words taking part in that particular relationship. For example, the relation might be a subject-verb-object relationship; the set $\{w\}$ then consists of only three words. The point here is that the r_i are "small", whereas $(w_1, w_2, \cdots, w_n)$ is "large". The goal is to grapple with complexity be finding suitable recurring patterns. Linguists have already shown what these patterns should be; now the task is to actually extract them from text.

The factorization is successful if

$$\log_2 \frac{p(w_1, w_2, \cdots, w_n)}{p(r_1)\, p(r_2) \cdots p(r_k)} \approx 0$$

[1] See Wikipedia, Statistical parsing.

With such a factorization in hand, one can now aim for higher and more abstract levels of structure, using the r_i as the building blocks, rather than individual words. One should imagine a perturbative structure, each level giving a foundation for the next.

Example: the Binomial MI Formula

As a concrete example of the above, consider the mutual information $MI(w_1, w_2, \cdots, w_n)$ over n variables. It is defined as

$$MI(w_1, w_2, \cdots, w_n) = \sum_{k=0}^{n} (-1)^{n-k} \sum_{w\backslash k} \log_2 p(\{w\backslash k\}) \tag{1}$$

where $\{w\backslash k\}$ is the set of words $\{w\} = \{w_1, w_2, \cdots, w_n\}$ with k of them removed. The sum over $w\backslash k$ is a sum over every combinatoric possibility of removal. By "removed", it is meant "summed over", so that, for example, if w_2 is removed, then

$$p(\{w\backslash w_2\}) = p(w_1, *, w_3, \cdots, w_n) = \sum_{w_2} p(w_1, w_2, w_3, \cdots, w_n)$$

The $*$ is the wild-card; it just denotes that "anything" can occupy that slot, and that, for probabilities, that slot should be summed over. Formally, $\{w\backslash w_2\} = (w_1, *, w_3, \cdots, w_n)$ is called a "cylinder set" and $p(\{w\backslash w_2\})$ is a cylinder set measure.[2]

The sum over $w\backslash k$ for $k = 1$ is then a sum over all possible wildcard locations, for a single wildcard:

$$\sum_{w\backslash 1} \log_2 p(\{w\backslash 1\}) = \sum_{i=1}^{n} \log_2 p(\{w\backslash w_i\})$$

Likewise, for $k = 2$ wildcards,

$$\sum_{w\backslash 2} \log_2 p(\{w\backslash 2\}) = \sum_{i=1}^{n} \sum_{j=1; j\neq i}^{n} \log_2 p\left(\left\{w\backslash \left\{w_i, w_j\right\}\right\}\right)$$

and so on.

The alternating sign is such that the singletons $p(w_j) = p(*, *, \cdots, w_j, \cdots, *)$ always have a minus sign in front of their log, while the first term is for the total space $p(\{w\backslash w\}) = p(\{\varnothing\}) = p(*, *, \cdots, *)$. If $p(\{\varnothing\}) = 1$ is a conventional probability, then of course $\log_2 p(\{\varnothing\}) = 0$. However, this MI sum works just fine if $p(\{\varnothing\}) \neq 1$. For example, the binomial formula eqn 1 still holds if N a count is used instead of p. This is because the normalizing factor $N(\{\varnothing\})$ can be pulled back through all of the terms.

The size of the set $\{w\backslash k\}$ is given by the binomial coefficient:[3]

$$|\{w\backslash k\}| = \binom{n}{k}$$

[2] See Wikipedia, Cylinder set and Cylinder set measure.

[3] See Wikipedia, Binomial coefficient

Note the resemblance of the formula for MI to the binomial theorem.[4] This is not accidental; it is a generalization of the binomial formula that holds for non-uniform intervals and non-independent correlations. It reduces to exactly the binomial formula if all probabilities are independent and uniform in size, *i.e.* if $p(a,b) = p(a)\,p(b)$ and if $p(w_j) = 1/n$. In this case, it becomes

$$\begin{aligned} MI(w_1, w_2, \cdots, w_n) &= \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} \log_2 \frac{1}{n^k} \\ &= -\log_2 n \cdot \sum_{k=0}^{n} (-1)^{n-k} k \binom{n}{k} \\ &= 0 \end{aligned}$$

The last part follows from

$$\frac{d}{dx}(1+x)^n = n(1+x)^{n-1} = \sum_{k=1}^{n} k \binom{n}{k} x^{k-1}$$

and setting $x = -1$.

More generally, the binomial-MI formula follows from the Cartesian-product nature of the topology of sequences. It is a formula that holds generically for cylinder set measures; there is nothing language-specific in this example.

The point of this example is to show that something seemingly "large" and "complex", such as $MI(w_1, w_2, \cdots, w_n)$ can be reduced into smaller, perhaps more manageable components, which can then be recombined back into the whole, with an exact (not approximate) expression, a summation over pieces-parts.

But there is also a second lesson here: the binomial MI formula is not suitable for natural language tasks. Although it is an exact expression, and individual words and word pairs appear in the lower summation terms, one gains no insight applying this to natural language. MI values can be both negative and positive; the alternating sign introduces more chaos into the mix. Basically, one has a series of small and large terms summing up and mostly canceling one-another. (Literally: try this experimentally, if possible. You will find that the various MI's bounce around, getting large and small, and that the sum is almost always smaller than the largest term.) The quest is to find a similar expression, ideally, an exact expression, where most of the terms are strictly positive. This would allow the structure, the factorization to be approached perturbatively, as a strictly convergent sequence of corrections, each applied to the last.

Example: Syntactic Factorization

Parses imply factorizations. Consider a sentence with a fixed single parse. Suppose that there is a location i in this parse, such that when considering the block of all words to the left of i, and the block of all words to the right of i, there is only a single edge connecting the two sides. For example, this might be the edge connecting word w_i, say, the verb, to word w_j, say, the object. (One cannot assume that $j = i+1$ since the object

[4] See Wikipedia, Binomial Theorem

might have adjectives and determiners that precede it. One should assume that $i < j$.) Then this parse implies a factorization

$$p(w_1, w_2, \cdots, w_n) \approx p(w_1, \cdots, w_i)\, p\left(r\left\{w_i, w_j\right\}\right) p(w_{i+1}, \cdots, w_n)$$

The is, the likelihood of the block of words to the left is effectively independent of the block of words on the right.

This factorization follows from the intuitive grammatical structure of natural language. Consider the sentence fragment "On alternate Tuesdays, John goes ..." How can it be completed? One imagines almost anything can complete it: "... fishing in Georgetown." "... to the doctor." That is, the completion of the sentence seems independent of the start of the sentence, and so the probability expression should factor like this as well.

Yet, this is just an approximation. Realizing that the first half of the sentence implies activity undertaken by a human, then the last half of the sentence must be an activity that humans can perform. So there is a linkage, a connection between these two parts of the sentence that extend beyond the grammatical relations between pairs of words. So again: the proposal here is to first factor according to syntax, providing a baseline, and then consider corrections to that initial factorization.

The above was written with a factor $p\left(r\left\{w_i, w_j\right\}\right)$ specifically tying together the two specific words w_i, w_j connected by the parse edge. This factor is made explicit because one imagines that the specific word-choice connecting the left and right halves helps further isolate or make independent these two halves. In the example, it is presumed that $p\left(r\left\{w_i, w_j\right\}\right)$ might capture at least some of the idea that the left side of the sentence is about humans, and the right side is about human activities.

Note that $p\left(r\left\{w_i, w_j\right\}\right) \neq p(w_i, w_j)$ and that the probability depends on the relation r. That is, this factor is presumed to not be a simple word-pair co-occurrence probability $p(\text{"goes"}, \text{"fishing"})$, but also includes a weighting for this word-pair being an auxiliary-verb pair. This now gives a first hint of the appearance of semantics in a syntactic discussion: The $p(w_i, w_j)$ captures a syntactic relation between a pair of words; the $p\left(r\left\{w_i, w_j\right\}\right)$ captures something more.

The notation $p\left(r\left\{w_i, w_j\right\}\right)$ may feel strange; a more conventional approach would be to write this as a conditional probability: $(w_i, w_j | r)$ and read this as "the pair (w_i, w_j) conditioned on the relation $r(w_i, w_j)$". But this seems awkward, and invites inappropriate applications of Bayes theorem. For the present case, the non-standard notation used here seems easier to write and more direct to think about. It can always be re-imagined as conditional probabilities, on an as-needed basis.

Connectors and Disjuncts

Syntactic relations are not just pair-wise connections, though. Syntactic elements have more complex structure. Thus, the above factorization might be more correctly written as

$$p(w_1, w_2, \cdots, w_n) \approx \left[p(w_1, \cdots, w_i) \sqrt{p\left(r\left\{w_i, w_j\right\}\right)}\right] \times \left[\sqrt{p\left(r\left\{w_i, w_j\right\}\right)}\, p(w_{i+1}, \cdots, w_n)\right]$$

so that half of $p\left(r\left\{w_i, w_j\right\}\right)$ rides along with the left side, as the probability of making a connection, and the other half rides with the other side. This square root $\sqrt{p\left(r\left\{w_i, w_j\right\}\right)}$ term can be referred to as the "connector probability". The "connectors" are then locations in the factorized tensor that have the potential to make a connection. This apportions the probability away from the actual connection, from the actual linkage, and moves it to the two endpoints of the connection.

Of course, real grammatical relations are more complex; they are not just a compendium of pair-wise relationships. For example, a transitive verb *must* make a connection to both a subject on the left and an object on the right. This is effectively a triple (S,V,O). Any factorizations involving transitive verbs should be factored in terms of a tri-variable $p(\text{TrVb}\{S,V,O\})$. A transitive verb V has the possibility of connecting to a subject S, and the possibility of connecting to an object O.

This last paragraph is a sneaky introduction to Link Grammar. The specific grammatical relation is $\text{TrVb}\{S,V,O\} = \texttt{V} : \texttt{S}- \,\&\, \texttt{O}+$. The right-hand side is the conventional Link Grammar notation stating that the lexical entry `V` has the connector `S`$-$ pointing to the left, and the connector `O`$+$ pointing to the right. The combined expression $\texttt{S}- \,\&\, \texttt{O}+$ is called a "disjunct"; the name stems from it's being disjoined from other lexical entries for the verb `V`.

The other sneaky thing being done above is to introduce the idea of a "word class" (subjects, verbs, objects). Thus, during factorization, we expect to see indicator functions, such as $p(w_j \in \texttt{V})$ which takes a value of 1 if w_j is verb, and zero, otherwise. Keep in mind, though, that it might be useful to assign fractional values to $p(w_j \in \texttt{V})$, for any number of technical reasons. It is premature to sketch these reasons, just yet.

The intended factorization, for transitive verbs, is to say that a tri-variable probability $p\left(\text{TrVb}\left\{w_i, w_j, w_k\right\}\right)$ can be factored into a a probability $p(w_i \in \texttt{S}+)$ of w_i belonging to some (any) class of words that can make subject-type connections, a probability $p(w_k \in \texttt{O}-)$ of w_k belonging to some (any) class of words that can make object-type connections, a probability $p(w_j \in \texttt{V})$ of w_j belonging explicitly to the transitive verb class `V`, and an overall probability of observing the relation $p(\texttt{V} : \texttt{S}- \,\&\, \texttt{O}+)$. How should this be written? The resulting factorization must be consistent with the notion of "connector probabilities". It must also be consistent with the lexical entry $\texttt{N} : \texttt{S}+ \texttt{ or } \texttt{O}-$, which states that there is a word-class of common nouns that can act as a subject, when to the left of a verb, or as an object, when to the right of a verb. But this is not the only such lexical entry with `S`$+$ or with `O`$-$ connectors: certainly, pronouns can make these connections as well. This is the reason for writing $p(w_i \in \texttt{S}+)$ instead of $p(w_i \in \texttt{N})$: what matters is not that w_i is a noun, but that w_i can serve as the subject of a sentence.

One concludes that transitive verbs contribute a factor

$$p\left(\text{TrVb}\left\{w_i, w_j, w_k\right\}\right) = p(\texttt{V} : \texttt{S}- \,\&\, \texttt{O}+)\, p(w_j \in \texttt{V})\, p(w_i \in \texttt{S}+)\, p(w_k \in \texttt{O}-) \times \sqrt{p\left(\texttt{S}\left\{w_i, w_j\right\}\right) p\left(\texttt{O}\left\{w_j, w_k\right\}\right)}$$

where each of the first four p's can be imagined to be zero or one, exactly, and a square-root probability for each word participating in a specific linkage. The square-root appears because there is a corresponding square root at the other side of the link.

To complete the example, consider the case where the subject and object are common nouns. Then these are covered by the lexical entry $\texttt{N}:\texttt{S}+\ \texttt{or}\ \texttt{O}-$ consisting of two disjoined disjuncts: one that says common nouns can act as a subject: $\texttt{N}:\texttt{S}+$ and another where they act as objects: $\texttt{N}:\texttt{O}-$. That is, for the subject,

$$p\left(\text{Subj}\left\{w_i,w_j\right\}\right) = p\left(\texttt{N}:\texttt{S}+\right) p\left(w_i \in \texttt{N}\right) p\left(w_j \in \texttt{S}-\right) \sqrt{p\left(\texttt{S}\left\{w_i,w_j\right\}\right)}$$

and similarly for the object. A three-word sentence (w_1,w_2,w_3) which has *exactly one parse* as SVO then has the probability

$$p\left(w_1,w_2,w_3\right) = p\left(\text{Subj}\left\{w_1,w_2\right\}\right) p\left(\text{TrVb}\left\{w_1,w_2,w_3\right\}\right) p\left(\text{Obj}\left\{w_2,w_3\right\}\right)$$

A longer sentence, say, one with adverbs, adjectives and determiners, having a transitive verb will then have a block factor $p\left(w_i,w_j,w_k\right)$ of this same form.

The goal here is to describe factorization along the lines of conventional linguistic grammars. Although an explicit Link Grammar notation is used, the arguments above can be transposed to any grammatical theory. The building blocks are simply the vertexes and edges that are drawn by that grammatical theory. The factorization above is two-fold. First, a graph corresponding to the links drawn by a (single) parse in that grammatical theory, in terms of grammatical classes, and an adjustment for the actual words employed.

Phrase Structure

The descriptions above are primarily couched in a Dependency Grammar[5] setting. A few words are in order about Chomsky-style production grammars, such as Phrase-Structure Grammars[6]. Such grammars consist of production rules, the first of which is conventionally $S \rightarrow NP, S\backslash NP$, stating that a sentence S consists of a noun phrase NP and the rest of the sentence $S\backslash NP$. This can be directly mapped to the assertion that

$$p\left(w_1,w_2,\cdots,w_n\right) = p\left(NP\left\{w_1,\cdots,w_i\right\}\right) p\left(S\backslash NP\left\{w_{i+1},\cdots,w_n\right\}\right)$$

for some yet-to-be-determined word index i. Such phrase structure grammars are necessarily trees, as production rules do not allow the creation of graphs with loops. The leaves of these trees are necessarily the words in the (fully-parsed) sentence. These trees, however, are not strict dependency trees: they also have non-leaf vertexes, labeled by the production rule that produced everything below. This does nothing to change the overall conception of factorization: in the example above, the factor $p\left(\texttt{V}:\texttt{S}-\ \&\ \texttt{O}+\right)$ plays the same role as a production rule vertex.

Cliques and Spanning Trees

What if there are multiple parses? How should this be understood? In short, as a many-worlds summation.

[5] See Wikipedia, Dependency grammar.
[6] See Wikipedia, Phrase sturcture grammar.

Given n words in a sentence, consider first the clique or complete graph[7] of degree n: this is the graph where every word, a vertex, is joined to every other word by an edge. A specific parse of the sentence then corresponds to a spanning tree[8] of this graph. This tree can be described in terms of an indicator function[9] on the edges of the clique. That is, make a list $\{E\}$ of all of the edges E in the complete graph, and then provide a function $\delta(E)$ that is zero or one on each edge. A specific parse T then corresponds to a specific indicator function $\delta_T : \{E\} \to \{0,1\}$. As a block factorization, one has that

$$p(T\{w_1, w_2, \cdots, w_n\}) = \prod_{(w_i,w_j)\in T} p\left(r\{w_i, w_j\}\right)$$

where T is the set of edges where the indicator function is one. The per-edge factors $p(w_i, w_j)$ may be indicator functions themselves, or may be weighted, or may be the result of a more complex factorization, as described in the previous section. That is, some of the pair-wise terms in the product should have been written as triples or quads:

$$p(T\{w_1, w_2, \cdots, w_n\}) = \prod_{r\in T} p\left(r\{w_i, w_j, \cdots, w_k\}\right) \tag{2}$$

If there is more than one possible parse of a sentence, then presumably one parse is preferred over another, and each can be weighted, with some probability $p(T)$ for each parse T. Each of these contributes to the overall analysis of the sentence:

$$p(w_1, w_2, \cdots, w_n) = \sum_{T\in\{T\}} p(T\{w_1, w_2, \cdots, w_n\}) \tag{3}$$

This summation implies that, in general, $p(w_1, w_2, \cdots, w_n)$ probably cannot be factored into blocks, although each term in the sum is explicitly block-factored. It might happen that all parses have a common sub-block; in this case, the sub-block can be pulled out of the summation, leaving only the ambiguous part inside the summation. Classic ambiguous parses are "I saw the man with the telescope", and so on.

Deformation Retracts

In mathematics, Homotopy Theory[10] concerns the structure of spaces with non-trivial topologies. In the present situation, the primary concern is how to work with graphs that have cycles (loops) in them, as opposed to those that do not. Trees are the prototypical example of a graph that has a trivial homotopy: the edges can always be shortened, until the two end-points have been collapsed into one. This is termed a "deformation retract". In the present case, it can be understood to mean that the factorization of a graph along an edge can be written as if the edge has shrunk to a point. Explicitly:

$$p(w_1, \cdots, w_i)\, p\left(r\{w_i, w_j\}\right)\, p(w_{i+1}, \cdots, w_j, \cdots, w_n) = p(w_1, \cdots, K)\, p(K)\, p(w_{i+1}, \cdots, K, \cdots, w_n)$$

[7] See Wikipedia, Complete graph.
[8] See Wikipedia, Spanning tree.
[9] See Wikipedia, Indicator function.
[10] See Wikipedia, Homotopy theory.

where a new kind of "word" K has been introduced. It's a compound word, a multi-word expression, a set phrase, a phraseme, an idiomatic expression, an institutional expression.[11] Collections of words can be "retracted", congealed down to single lexical units.

Twines

A more problematic situation arises for graphs that have cycles. Although conventional phrase-structure and and dependency parses are trees, loops can appear in dependency parses. The very simplest case would be an HSV parse, where H is the left-wall or head of the parse, S is the subject, a noun, and V is a verb: this forms a triangle: there are three edges. H is used to indicate both the dominant noun (the subject) and also the dominant verb.

Perhaps this can be seen more clearly in dependent or relative clauses. An example from the Link Grammar documentation:[12]

```
     +--B------+
     +-R-+--S--+
     |   |     |
The dog  I  chased was black
```

In the above the R link points at the head noun of the relative clause; the B link connects to the head verb of the relative clause, and the S link is the conventional subject-verb link. Loops may be larger than triangles:

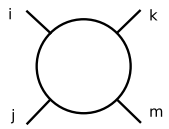
```
     +-----B------+
     +-R--+C-+-S--+
     |    |  |    |
The dog  who I chased was black
```

The C link connects head nouns to subordinating conjunctions.[13]

Such loops present a potential complication to factoring. In order to factor a block of text into a left and a right component, there are now two links to be cut. There are several ways in which to imagine this issue. One is to presume that the relative clause is an irreducible block of the form

$$p\left(\text{RelCl}\left\{w_i, w_j; w_k, w_m\right\}\right)$$

with i, j linking to the left-hand block and k, m linking to the right. In the last example, it would be $w_i =$ `dog`, $w_j =$ `who`, $w_k =$ `I` and $w_m =$ `chased`, so that the entire loop of the relative clause is unreduced, and has four connectors grand-total, emanating from it:

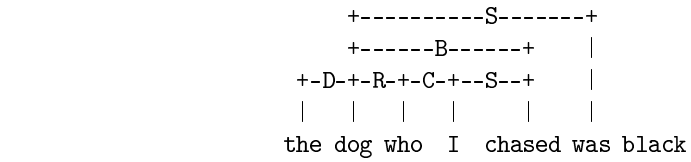


[11] See Wikipedia, Phraseme and Multiword expression and Idiom.
[12] See the Link Grammar Guide-to-Links Section R, Section B, Section S, and Section C.
[13] See Wiktionary, subordinating conjunction.

Alternately, this loop is clearly composed four three-point vertexes and so following the earlier conventions, there is a pair probability for each linkage (there are four links: R,B,C and S), a square-root for each exposed connector (there are two: D for determiner, and S), and a vertex factor for each vertex:

$$\begin{aligned} p\left(\text{RelCl}\left\{w_i, w_j; w_k, w_m\right\}\right) = & p\left(\texttt{R}\left\{w_i, w_j\right\}\right) p\left(\texttt{C}\left\{w_j, w_k\right\}\right) \times \\ & p\left(\texttt{S}\left\{w_k, w_m\right\}\right) p\left(\texttt{B}\left\{w_i, w_m\right\}\right) \times \\ & p\left(\text{Vertex}\right) \times \\ & \sqrt{p\left(\texttt{D}\left\{\texttt{the}, w_i\right\}\right) p\left(\texttt{S}\left\{w_{i,}\texttt{was}\right\}\right)} \end{aligned}$$

the factors of which follows from the fuller diagram:



In order to not clog the above, the vertex factor was separated out:

$$\begin{aligned} p\left(\text{Vertex}\right) = & p\left(\texttt{who} : \texttt{R} - \,\&\, \texttt{C}+\right) \times \\ & p\left(\texttt{I} : \texttt{C} - \,\&\, \texttt{S}+\right) \times \\ & p\left(\texttt{chased} : \texttt{S} - \,\&\, \texttt{B}-\right) \times \\ & p\left(\texttt{dog} : \texttt{D} - \,\&\, \texttt{R}+ \,\&\, \texttt{B}+ \,\&\, \texttt{S}+\right) \end{aligned}$$

As before, the individual word-mentions could have been pulled out into grammatical classes, so that, for example:

$$p\left(\texttt{chased} : \texttt{S} - \,\&\, \texttt{B}-\right) = p\left(\texttt{chased} \in \texttt{V}\right) p\left(\texttt{V} : \texttt{S} - \,\&\, \texttt{B}-\right)$$

and so on.

Summary

In this way, and ordinary seven-word sentence "the dog I chased was black", having a joint probability $p\left(w_1, \cdots, w_7\right)$ can be factored into independent blocks. This factorization is more complex than a conventional Hidden Markov Model, and properly should be called a Markov random field. [14] We stop short of calling this a Bayesian Network, because we've stopped short of using any Bayesian priors $p\left(\theta\right)$, nor of combining them with likelihoods $p\left(x|\theta\right)$ to obtain posteriors $p\left(\theta|x\right) \sim p\left(x|\theta\right) p\left(\theta\right)$.

One reason to avoid Bayesian priors is to instead allow the use of Gibbs measure[15] and so to rephrase the probabilities in terms of entropies (or free energies):

$$p\left(x_i\right) = \frac{1}{Z} e^{-\beta_i E_i}$$

[14] See Wikipedia, Hidden Markov model, Markov random fieldand Bayesian network.
[15] See Wikipedia, Gibbs measure.

This also distinguishes the graphs here from the concept of Conditional Random Fields[16] which are conventionally formulated in terms of priors. Another reason to avoid Bayesian formulations is the problem of ambiguity, sketched below.

Many Worlds

The factorizations being described above are *not those of maximum entropy approaches*! *Nor are they Bayesian*! We must now be careful not to discard the baby with the bathwater: most of the equations above are a stream of bathwater; the baby is eqn.3. That is, we are *not pursing a singular and best parse* (arrived by via MaxEnt or via Bayesian inference.) The goal is to explicitly avoid a single model. Avoid some Bayesian distribution over likelihoods. The central and key idea that we are struggling to expose is that one must insist that all of the various likelihoods are not only possible, but are intertwined. There is no one true reality that is merely unknown and needs to be found out; there are many, and they are necessarily tangled together.

Perhaps this sounds like philosophical quantum woo. It is not meant to be; it needs to be unpacked. Linguistics is relatively arid, but there are a few examples: again , the classic "I saw the man with the telescope." In a spy thriller, perhaps the protagonist is looking through the telescope. In the biography of a famous astronomer, the protagonist may be standing in an observatory. Without that context, it is ambiguous. "Ah ha!" you may say, "but eventually it becomes clear, one or the other! And we can update our priors when it becomes clear!". Alas, it will never become clear. I will not tell you which context I am actually thinking of; you will be left hanging. I won't tell you, not because I'm secretive or coy, but because this is a meta-conversation about linguistics and not about telescopes. In this situation, it is fundamentally impossible for you to update your priors.

Poetry provides a richer example: what was the poet thinking, when he, she, wrote those verses? What did they want you, the reader, to think? A good poet will want you to think of a multitude of things, to feel many emotions, all at once, at the same time. The name of the game is not to update our Bayesian priors and select one and only one emotion on which we will fiercely focus (unless, of course, it is nationalistic poetry 😀😀).

More broadly, ambiguity is primal to common sense: our visual field is filled with a myriad of items, a rainbow of events and happenings. There's (usually) no particular reason for focus attention on one or another; the brain, in default mode, wanders across the field of possibilities. A mathematical formulation of AGI must also capture this freedom to wander about. And yet, also, some things are distinct. Distinctness is captured in eqn 2: my coffee cup is distinct from my coffee. The world of possibilities is captured in eqn 3: there are many and they compete with one-another.

If one is given only the string $p(w_1, w_2, \cdots, w_n)$ and the task of factoring it, then sure, use eqn 2 and be prepared to lose, whenever there are any admixtures of anything else in there. But there always will be admixtures of something else! Those admixtures cloud the factors, because the admixtures are entangled into the total. In the end, all language is necessarily poetry, an artful attempt to capture vague thoughts and set them

[16] See Wikipedia, Conditional random field.

into words, in such a way that someone else might happen upon them, and perhaps extract something meaningful, perhaps what the author meant, and yet unavoidably unclear, because the author was never precise enough, and the reader was never clever enough to understand. This is the human condition.

To reiterate: the goal here is both to factor, to obtain factors that are relatively unambiguous, and then at the same time, bracket the ambiguities so that they are each corralled in their own paddock, and can be recombined as needed. This need is what makes the machinery daunting.

Although neuroscientists will eventually find the neurological basis for human ambiguity, the origin of ambiguity is not the human mind. Ambiguity is fundamental, in nature, as it stands. This is the message delivered by eqn. 1 and eqn. 3.

MST Factorization

The sections above provide theoretical arguments. Let quickly review the experimental situation. The Yuret-style MST factorization being used in the language-learning effort is effectively the presumption that

$$MI(w_1, w_2, \cdots, w_n) \approx \sum_{(w_i, w_j) \in MST} MI(w_i, w_j)$$

This sum runs only over the maximum spanning tree, and contains only $n-1$ links. The corresponding complete graph would have $n(n-1)/2$ links in it, and so most of these are ignored. Specifically, the presumption is that these ignored links actually cancel higher-order terms in the full MI expansion.

Put more plainly, it appears that

$$\log_2 \frac{p(w_1, w_2, \cdots, w_n)}{p(w_1)\, p(w_2) \cdots p(w_n)}$$

is, in general, "freakishly high", and that much of it can be "knocked down to size" by using the MST, instead.

Lacking is a coherent theoretical argument as to *why* an MST parse provides a reasonable approximation to the factorization. Also lacking is any comprehensive experimental exploration comparing the full, formal factorization to the MST approximation. Gut feel implies that MST is OK or even "pretty good", but no one has characterized the structure of the difference

$$\Delta(w_1, w_2, \cdots, w_n) = MI(w_1, w_2, \cdots, w_n) - \sum_{(w_i, w_j) \in MST} MI(w_i, w_j)$$

When is Δ small? When is it large? What does it mean, when it is large? Are there tricks that can describe such deviations?

The next step is, of course, to use disjuncts, but again, without any clear argument about why the disjuncts provide a good approximation to the higher order terms in the sum of binomial-MI equation 1.

More precisely, the binomial-MI equation is correct, but unwieldy, because it contains large canceling terms. What is unclear is why these terms cancel, and how to

best obtain a diagonalized, factorized perturbative expansion. The MST->disjunct path is just a gut-feel approach to obtaining that perturbative expansion. It lacks formal justification for why it works.

Factorization as Dimensional Embedding

Syntax alone is not enough to convey the meaning of an expression, and so the above approximations, of working with parses, are necessarily mediocre. The factorization we are groping for should more properly be written as a change of variable

$$p(w_1, w_2, \cdots, w_n) \approx p(r_1, r_2, \cdots, r_k)$$

where the variables $r_1, r_2, \cdots, r_k$ are in some sense "more independent" than the word-sequence $w_1, w_2, \cdots, w_n$.

Note that in general, $k \neq n$. For spanning tree parses, the r_i are understood to be links between word-pairs, and so k is counting the number of links in the parse. Thus $k = n - 1$ for word-pair relationships. If the parse has fundamental cycles (loops), then $k = n - 1 + \ell$ where ℓ is the number of fundamental cycles (*e.g.* in an MPG parse).

Note that parsing performs a "dimensional oxidation": there are far more r_i's than there are w_i's. If the size of the base vocabulary is $\mathcal{O}(|w|) \sim N$ then $\mathcal{O}(|r|) \sim N^2$ where I'm using $\mathcal{O}$ notation because counting the size of the vocabulary is hard, when vocabulary words have a Zipfian distribution. Also, the claim that $\mathcal{O}(|r|) \sim N^2$ is somewhat misleading. Its actually more like $\mathcal{O}(|r|) \sim N^\gamma$ for some $\gamma < 2$ because this is what the Zipfian distributions do to us. We've seen this experimentally, when we measure the sparsity and rarity,[17] but we haven't explicitly measured this.

Thus, MST parsing is a form of dimensional embedding, where the strings living in the relatively low-dimensional space $\mathcal{O}(|(w_1, w_2, \cdots, w_n)|) \sim N^n$ are embedded into the vastly larger space $\mathcal{O}(|r_1, r_2, \cdots, r_k|) \sim N^{2k}$.

In Link Grammar parsing, the embedding is not into a word-pair space, but into a disjunct space, which is explosively larger. That is, the relations r are actually disjuncts d. I assume its $\mathcal{O}(|d|) \sim N^\gamma$ for some $\gamma > 2$, but again, we've monitored this size without actually ever measuring it's scaling dependence. This is an experimental to-do: got to fix that.

Paths in hyperspace

Lets try to paint a mental image of this. Consider a vector space of dimension N, with N the size of the vocabulary. Each w_k is then a unit vector e_k in this space. The word-sequence $w_1, w_2, \cdots, w_n$ is a path in this space. The probabilities $p(w_1, w_2, \cdots, w_n)$ are hard to factorize, because there are many of these paths, and they overlap a lot.

Consider now a vector space of dimension D, with D being the number of disjuncts. Very roughly, $D \sim N^\gamma$ for some $\gamma > 2$ or something like that. So this is a much larger space. A single Link Grammar parse of a sentence $S = (w_1, w_2, \cdots, w_n)$

[17] See e.g. page 34 or Diary Part Five for rarity. I could have sworn I had this in other tables, but I can't find it right now.

provides a unique sequence of disjuncts $G = (d_1, d_2, \cdots, d_n)$ fixed by that parse. As before, $d_1, d_2, \cdots, d_n$ specifies a path through the disjunct space. However, this time, the space is much larger, and so the accidental intersection of two different paths is much less likely. There's disambiguation.

Another difference is that the path $d_1, d_2, \cdots, d_n$ is constrained. A syntactically valid path $d_1, d_2, \cdots, d_n$ is necessarily one where *all* of the connectors on all of the disjuncts d_i in that path are fully connected. Other paths are simply not valid. This stands in sharp contrast to word sequences $w_1, w_2, \cdots, w_n$ which are unconstrained: one is free to write any word-sequence, even if it's nonsense.

Metric spaces

The space of words is endowed with several metrics. One of the simplest ones is given by the word-pair MI. Fixing a word w, consider the vector $\vec{w}$ of length $2N$, whose vector components are given by $x_j = MI(w, w_j)$ for $j < N$ and by $x_j = MI(w_j, w)$ for $N \leq j < 2N$. That is, using entirely conventional notation, write $\vec{w} = \sum_j x_j \hat{e}_j$ with the x_j being just real numbers, and the $\hat{e}_j$ being the unit basis vector for the vector space.

Experimentally, it has been seem that the distribution of the MI of word-pairs is approximately Gaussian, perhaps even to a surprising degree.[18] This implies that the word vectors $\vec{w}$ are uniformly randomly distributed on a unit sphere: that is, the word-vectors form a Gaussian Orthogonal Ensemble (GOE). Because these vectors are distributed on a sphere, the cosine distance between the vectors can be used as a metric. Intuitively, this metric judges two words to be similar, when they have similar neighbors.

This is not the only such metric. One can construct a different word-pair MI, from disjuncts. This is obtained by considering a pair-wise correlation $f(w_i, w_j) = \sum_d p(w_i, d)\, p(w_j, d)$ and then constructing an MI from f. The sum over d is the sum over disjuncts. This, too, appears to be described by a GOE.[19] Intuitively, this metric judges two words to be similar, when they appear in similar grammatical contexts. This is a bit stronger and more constrained than saying the words have similar neighbors: they must also be grammatically similar. Experimentally, these two different distances are correlated; I don't have r-values offhand.

Presumably, the metric obtained from disjuncts is semantically more accurate, although we don't have any way of measuring semantic similarity.

Nearby paths

These word-pair metrics, together with disjunct sequences, provide an opportunity to explore similar sentences. Given a grammatically correct sentence $w_1, w_2, \cdots, w_n$ one can now explore other nearby words to obtain other sequences. In general, though, these other nearby strings will not be grammatically correct. Several possibilities exist. One is to just sample such sequences, parse them, and hope to stumble across a sequence that parses.

[18] See Diary part Five, Part Nine, the 2008 word-pairs report and also the AGI 2022 paper.

[19] This is reviewed in Diary Part Eight, if I recall correctly.

A more clever approach is to start with the path $d_1, d_2, \cdots, d_n$, as this is already grammatically correct. For any fixed d_i, one then examines the set of words $\{w|\,(w,d_i)\ \text{exists}\}$ and order these according to the cosine distance between w and w_i. The result is then necessarily a grammatically correct sentence. In fact, it will have exactly the same parse as the original sentence. This is perhaps overly strict: it turns into an exercise of finding synonyms that can work as drop-in syntactic replacements. By necessity, the resulting sentence will also be of exactly the same length, as only a word-for-word substitution is done. BTW, this assumes that connectors have been classified into grammatical classes, as otherwise the set $\{w|\,(w,d_i)\ \text{exists}\}$ will contain exactly one element.

Syntactically similar sentences, with a different number of words, can be obtained by means of idioms and "institutional phrases". These would be word-sequences that, taken as a unit, have a disjunct d_i. That is, all of the unconnected connectors on the idiom provide d_i. Thus, we expand the set of nearby words to nearby phrases: $\{\text{phr}|\,(\text{phr},d_i)\ \text{exists}\}$. Fishing from this set allows syntactically identical sentences to be constructed, with varying numbers of words (syntactically identical, ignoring syntactic structure in the phrase itself).

Loosening the concept of idiom to, say a "common noun with a adjective", or a "verb with an adverb", we can consider the set $\{w|\,(w,c_A-\&\,d_i)\ \text{exists}\}$ where c_A- is an adjectival connector. That is, the d_i connects to the rest of the sentence, as before, but that now, there is an adjective connector, to which an adjective can be connected. Thus, again, we explore the space of nearby sentences, but now with locations that can be "decorated" with extra words.[20]

Now, most of what has been written in the three paragraphs above is old hat; ideas such as this have been articulated in linguistic theory for many decades. What is different here (what I hope is different) is the provision of an actual metric, a way of actually measuring the distances between sentences that moves beyond the concepts of a Hamming distance or a Levenshtein distance,[21] and gets us closer to a semantic distance.

The metric above is a "syntax-respecting distance". Lets write down a formal formula for it. Given two strings $W=(w_1,w_2,\cdots,w_n)$ and $V=(v_1,v_2,\cdots,v_n)$ having the same parse $d_1,d_2,\cdots,d_n$, the syntax-respecting distance srd is

$$\text{srd}\,(W,V)=\sum_i f\,(w_i,v_i)$$

where f is a word-pair distance, presumably just the MI between the two words (which is why it is summed, instead of e.g. taking the Euclidean sum-of-squares distance.) This strict distance can even be extended to accommodate idioms, adjectives, etc. as described above.

[20] The word "decorated" here is similar to the idea of mutation in genetic programming. There, one takes an existing expression tree, and randomly adds "knobs" to it, in various places. The "knobs" can be "turned" to have different settings, and the fitness of the expression tree, with a given knob-setting is then evaluated. The fitness is used for evolutionary guidance of a population of individuals. In the present case, the "knob" would be the vacant slot for an adjective. The knob setting would be the selection of a specific adjective. The fitness can be evaluated extrinsically: "does this new sentence express the idea better?" or intrinsically: "does the mutual information of the total sentence increase?"

[21] See Wikipedia, Hamming distance and Levenshtein distance.

Note that the MI tools also allow the definition of $MI(d_i, d_j)$ so that, for any given disjunct d_i, there is also a local neighborhood of "similar" disjuncts d_j. However, these are necessarily grammatically different. Thus, one cannot take a syntactically valid parse, and just substitute $d_i \mapsto d_j$ without braking the parse. But perhaps this can be rescued in some way. At this time, there is no experimental characterization of $MI(d_i, d_j)$ beyond the fact that it is distributed as a Gaussian (and so again is a GOE). Experimental results remain a bottleneck to theorizing.

Representing ideas with word sequences

Consider the task of a writer who wishes to express an idea. At the basic level, this requires a search for different collections of sentences that convey the same meaning. At this point, we do not have any precise definition of what "an idea" is, or how to thread sentences through it. Presumably, an "idea" is some region of space, a volume, and an expression of the idea is a collection of sentences, paragraphs, that form a space-filling curve through this space. The sentences are the flight of a moth about a light-bulb, filling the surrounding space with a trajectory.

Given the present set of developments, what could an "idea" be? Well, per usual, a concept: "a chair", with all of it's extensive and intensive properties: has legs, can be sat on, is movable, has a flat surface, etc. This can be treated as a "bag of words": {legs, sit, movable, flat}, a primary word "chair", the associated relations: {has, can be, is}. The description & expression of a chair is then the threading of this space by strings of words, visiting all of the space, without repetition. These last two: "visit all of the space" and "without repetition", require a metric space. For the first, "visit all of the space", some Hausdorff topology of balls, with a sense of the volume of the balls: the metric gives us this. For the second, a sense of separation or distance, to maximize distance between sentence-strings (while staying within the concept-space). We want two sentences to repel each other, when they get too close. Again, this is provided by the metric.[22]

Recursive relationships

The ultimate hypothesis presented here is that these ideas can be applied recursively. Starting with word-pair MI, we arrive at MST parses. Starting with MST parses, we arrive at disjuncts and structural analysis of sentences. This process can be repeated again. Specifically, the $MI(d_i, d_j)$ between a pair of disjuncts is directly available, from direct observation. Given a parse $d_1, d_2, \cdots, d_n$ of a sentence, we can now obtain the MST parse of $d_1, d_2, \cdots, d_n$. Where does this lead? In what sense does

[22] Would it be better to have a "Pauli exclusion principle"? To disallow two sentences from occupying the same space? This is not the same as saying that two sentences are repulsive, when they are too near each other. How might this work? I am not aware of any description of any (measure-preserving) dynamical system that exhibits or makes use of some exclusion principle. In Riemannian geometry, the exclusion principle arises from fermions, which provide a certain square-root of a connection, i.e. a "spin manifold" or a "spin structure" as mild generalizations of Riemannian manifolds. But we have not defined this "space of words and sentences" closely enough to map it into the machinery of fiber bundles and connections, from which we could describe spin structures. So, for the moment, this idea remains out of grasp.

this provide the higher-order perturbative expansion of the original object of study, $MI(w_1, w_2, \cdots, w_n)$?

Let's not lose sight of the meta-goals. At short range, we wish to obtain structure across multiple sentences, at the paragraph level. Insofar as human communication is about topics, then the goal is to identify a common theme. This can be reduced to collecting assertions made about objects: a very traditional KR task. At longer range, the grammatical relationships between objects is nothing other than "common sense": if you hit your thumb with a hammer, it will hurt. This is not a "logical deduction", this is the perception of a structural relationship in sensory data. Is the procurement of MST parses of disjuncts the next step in the automated extraction of common sense?

The answer is cloudy. Certainly, hypothesis can be readily cooked up for all of this. We can limit the kinds of hypothesis to those involving MI analysis across larger expanses of text. But still, there are many of these; which work best? Can we go meta at this level, also? To ask for an algorithm that generates hypothesis, involving the factorizations of MI, that automatically explores the various alternatives? Perhaps. For me, the limiting factors are not an ability to hypothesize and theorize, but to perform actual experiments and measure results. This is in turn limited by lack of suitable infrastructure. The degree to which this aligns with the meta-goal of AGI remains unknown.

The End

This is the end of the factoring paper, for now. Perhaps more will be added, later. A suitable conclusion must be written.

References

[1] Eugene Charniak, "A Maximum-Entropy-Inspired Parser", *Proceedings of NAACL-2000*, 2000, URL https://cs.brown.edu/people/echarnia/papers/shortMeP.ps.gz.