

The evaluation metric in generative grammar

John Goldsmith

July 14, 2013

1 Introduction

The subject which I would like to treat in this paper is the evaluation metric in generative grammar. Why? Arguably, the evaluation metric is both the most novel and the most important concept in the development of generative grammar by Noam Chomsky. And yet it is at the same time one of the least recognized and surely most misunderstood of the core concepts of generative grammar. So there you are: the evaluation metric is critically important, it is arguably novel, it is misunderstood, and at some times and in some places, it has even been reviled. What better reasons could there be for spending our time today talking about it?

I would like, first, to explain the idea of the evaluation metric in early generative grammar; this will mean exploring the separate ideas of (1) a prior over the set of grammars and (2) a measure of goodness of fit to the data. Second, I will very briefly trace how those two ideas have been developed in the world of machine learning over the last few decades, and as you will see, the picture I see is one of lost opportunity for us linguists, though the opportunities remain open to us.

To repeat, then: The notion of an evaluation metric played a central role in the first decades of generative grammar.

The idea of an evaluation metric was tightly bound up with the notion of explanatory adequacy. I am going to sketch an idealized picture of what I take it that Noam Chomsky proposed in his early work, *The Logical Structure of Linguistic Theory* [notably in Chapters 2 and 4 there] and *Syntactic Structures*.

First, a grammar can be thought of as a scientific theory of a language, and any grammar's predictions can be tested against real data (whether corpus-based or judgment-based), and in most cases we can compare any pair of grammars on the basis of linguistic data. If that were the whole story, though, we would not need an evaluation metric. But there is more. Early on in the lifespan of the enterprise, we linguists learn about the formal properties of the very best grammars, and as we do so, we develop a higher-order description language, a language in which grammars are written; and as we do this, we will ensure that those phenomena which are common across grammars can be expressed in a compact way, so that within a brief period of disciplinary evolution, a metalanguage will emerge in which compactness of expression will fit well with what is found in a wide range of languages. Out of this process will naturally *emerge* a body of meta-knowledge about the nature of language, and this body of meta-knowledge lies simply in the way in which grammars are written.

This conception of the scientific character of linguistic research—of linguistics as a science—was developed explicitly to counter three tendencies that were present or in any event that *seemed* to be present, either explicitly or just below the surface, in the work of a number of linguists working in the Sapir-Bloomfield tradition:

1. Any analysis is as good as any other.
2. It will often be the case that there are multiple best analyses, depending on what it is you are interested in studying or accomplishing.
3. The analysis of one language should have only a very indirect effect on the analysis of another: and the only acceptable indirect effect is that what happens in one language could stimulate the linguist's imagination to come up with novel ways of treating data in another.

All three of those views could be rejected as long as we were committed to developing and deploying an evaluation metric. The first and second view can be rejected in favor of the idea that there is a *right* analysis, because even if we are unable on language-internal grounds to eliminate one of the two, we will nonetheless be able to eliminate one once we have an evaluation metric that we can rely on. Third, the

evaluation metric provides just the right degree of “hands off” attitude that work on the grammar of one language should have on future or on-going work on a different language.

In order to better understand this development, I think it is essential to paint a picture of the intellectual milieu in which generative grammar arose in the middle of the 20th century. For people following what was going on in probability and related fields, the notion of an evaluation metric was not entirely strange or unfamiliar. And the 1950s and 1960s saw the development of ideas regarding algorithmic complexity which were at their heart very similar to what the generativist’s evaluation metric was designed to do. Work by Ray Solomonoff, Jorma Rissanen, and others over the last 50 plus years has seen a marvelous development that meshes well with the notion of an evaluation metric. Let us turn to that now.

2 What is a good theory?

A good account of a large collection of observations is one that is both simple and which fits the facts well. The whole artistry of doing science right is based on the sad fact that the path to a theory that is both simple and capable of fitting the facts well is a long and arduous one, in which a great deal of time, measured in years rather than weeks, must be spent in close quarters with a theory that is either complicated but fits the facts well—or else with a theory that is simple but does not fit the facts well at all. Success in sailing between these barrier reefs and reaching the high seas nonetheless is the challenge that all scientists face.

Much ink has been spilled over the question as to when a scientist ought to give up on a theory when its mesh with the observed facts is fair to poor. I have no intention of addressing this question at all. Indeed, from the perspective that I am describing here, that is a silly question to ask. The right question to ask is: how do we characterize these two notions of scientific simplicity and of snug fit to the data? And just as importantly, is it possible to speak meaningfully of a trade-off between the two? Can we speak meaningfully and usefully about when we should be willing to pay the price of poor data-fit and when we should not?

Here is the central idea, then, in two parts:

1. Part the First: Given a set of data d and a grammar g , g can be judged in two ways: first, how well does g *fit* the data? What is the goodness-of-fit of g to d ? We would like to provide a measurement that is both explicit and rigorous.

And second, how good is g as an analysis at all, quite independent of the data? That is the question that was never asked before generative grammar came around; that is the question for which the evaluation metric was designed; and that is the question which (without most linguists being aware of it at all) [which] has been developed with great elegance and mathematical sophistication in the last several decades in connection with notions like algorithmic complexity.

Part one, then, is this: we want a theory that is, yes, simple and elegant; and we want a theory that fits the data well. If we can give two independent answers to these measurements, that would already be great. But Part Two tries to go one step further.

2. Part the Second: How do we relate the two answers given in Part One? The two answers had to do with the degree of goodness of fit to data, and goodness of a grammar independent of data. How can the answers to those questions be combined? Generative grammar gave some hints as to the answer. As we will see below, better and well-worked out answers would be given by Minimum Description Length analysis, utilizing the core notions of information theory; it turns out to be possible to measure *both* of these quantities in the same abstract unit, the *bit*. But we are getting ahead of ourselves.

You all recall that in *Syntactic Structures*, Chomsky laid out the proper goal of linguistic theory, which is to serve as a box into which a triple consisting of three items (a set of data, and two candidate grammars) are sent, and from which an answer is produced selecting one of the two grammars as the better grammar for that data. Let’s put that in mathematical form:

Find the right function—call it UG , if you’d like—such that

$$UG(g_1, g_2, d) = -UG(g_2, g_1, d) \tag{1a}$$

$$\text{and } UG(g_1, g_2, d) > 0 \text{ iff } g_1 \text{ is a better grammar for } d \text{ than } g_2 \text{ is.} \tag{1b}$$

The central question of linguistic theory will have been solved when UG has been established

Now consider this hypothesis: the mathematical function UG can be factored into two simpler functions, each a function of only two arguments: one function compares the grammars, independent of considering anything about the data—in other fields, this is what is called a *prior* that is established over models or grammars; and then a second function which takes two arguments, a grammar and a corpus. This is essentially a goodness of fit function. So the hypothesis, never made explicit in a generative context even if it lurks elusively just beneath the surface, is that the grammar selection measure UG is composed of these two simpler functions, f and g , and some method of combining them, a third function Θ , and Θ may be linear, or it may not be.

$$UG(g_1, g_2, d) = \Theta(m(g_1, g_2), n(g_1, d), n(g_2, d)) \quad (2)$$

Looking ahead to what is called “Minimum Description Length” analysis, or MDL, we see a simple answer to what this function is:

$$UG_{MDL}(g_1, g_2, d) = |g_2| - |g_1| + \log_2 \frac{pr_{g_1}(d)}{pr_{g_2}(d)} \quad (3)$$

In other words,

$$m(g_1, g_2) = |g_2| - |g_1| \quad (4a)$$

$$n(g, d) = \log_2 pr_g(d) \quad (4b)$$

$$\Theta(x, y, z) = x + (y - z) \quad (4c)$$

To get there, we will look briefly at each of the two critical aspects of this question: developing an account of goodness of fit, and developing an account of a *prior over grammars*, that is, a way of evaluating grammars independent of their goodness of fit to the data.

2.1 Part One: Goodness of fit

The notion of ‘goodness of fit’ is a good deal more subtle than it may appear at first blush.

Anyone who has worked with experimental data understands quickly that the last thing we want is a theoretical model that explains everything about the data, because in every case like that we find that the theory then *overfits* the data. Some aspects of the data we do not want to explain with our models. In the case of language, there are many aspects we do not want to explain—the standard example of this being the *content* of a sentence, or a paragraph, or a book. The linguist only wants to *explain* about any given corpus those things that will still be true when we turn to a different corpus in the same language. That which is particular to a corpus should *not* be explained by the linguist: that would be a case of a model that over-fits the data at hand.

I want to emphasize that this point is quite different from the meta-observation that a theory can be taken seriously (and even be viewed as a success) if it does not account for all of the data. That is a different point.

One of the few places that I have seen in the generative literature where the significance of this point for the larger enterprise is in an essay of Chomsky’s in 1965 Berkeley lectures, *Language and Mind* (1968, pp. 76-77):

A third task is that of determining just what it means for a hypothesis about the generative grammar of a language to be “consistent” with the data of sense. Notice that it is a great oversimplification to suppose that a child must discover a generative grammar that accounts for all the linguistic data that has been presented to him and that “projects” such data to an infinite range of potential sound-meaning relations....The third subtask, then, is to study what we might think of as the problem of “confirmation”—in this context, the problem of what relation must hold between a potential grammar and a set of data for this grammar to be confirmed as the actual theory of the language in question.

2.1.1 Harwood

But one of the few explicit discussions of this question appeared in an article in *Language* in 1955 by F. W. Harwood, called *Axiomatic syntax: The construction and evaluation of a syntactic calculus*, which begins, “This paper discusses methods for presenting syntactic information in the form of a calculus, and for measuring its goodness of fit to a language.” It is worth our attention, if only for a few minutes, because it gives us a good sense of where linguists’ thoughts were at this critical time in the history of American linguistics. Harwood explains that his account of syntax will be divided into two components, one containing *formation* rules, which put sequences of morphemes together to form sentences, and another component containing *transformation* rules, such as the passive transtormation. The terminology he employs is a confluence of two traditions, one from Rudolf Carnap and the other from Zellig Harris. The one from Carnap derives from the English translation of Rudolf Carnap’s *Logical Syntax of Language* – Carnap’s translator in the 1930s proposed these terms, *rules of formation* and *rules of transformation*, to refer to rules of sentence formation and rules of proper implication, respectively.

Harris had begun using the term “transformation” around the same time as Carnap’s translation appeared, and I have found no evidence (despite having looked for it) that there was any documentable influence between these two streams of usage, one in philosophy and the other in linguistics. The coincidence is perhaps curious, but we may have to accept it and live with it.

In any event, Harwood uses these two terms, *rules of formation* and *rules of transformation*, which together constituted a syntactic grammar of a language. His theory of syntax was terribly simple: the model consisted of an assignment of words to lexical categories, plus a list of sequences of categories; each sequence of categories would correspond to a whole set of sentences of the language, each derived by placing a word from the right category into the right slot. Another set of sentences would be added to set of sentences that are predicted to be grammatical by a set of transformation rules. For a given word-length p , Harwood says, we can easily calculate the number of sequences of categories permitted by our grammar (for example, det N V det N might be one such category sequence); the set of such sequences (all of length p , remember) he calls C_p , and uses vertical strokes, like $|C_p|$, to indicate the number of items in that set. From this, and a known and finite lexicon, we can count the total number of sentences of length p generated by this grammar. Harwood calls this set K_p , and its size is therefore indicated by $|K_p|$.¹ For concreteness, he uses r to indicate the number of words in the lexicon. There are therefore r possible or conceivable sequences of words of length p ; call this set U_p . The set of *actually* ungrammatical sentences of length p is $U_p - K_p$.

Harwood establishes certain measures applied to a grammar and the corpus to which the grammar is offered as the correct analysis, as in Table 1.

| | |
|-------------|---|
| C_p | Set of category sequences of length p permitted by grammar |
| K_p | Set of word sequences from finite lexicon permitted by grammar |
| L_p | Set of all grammatical sentences of length p in L |
| U_p | Set of all sequences of length p of words from lexicon |
| $U_p - L_p$ | Set of all ungrammatical sentences of length p |
| $U_p - K_k$ | Set of all strings of length p predicted to be ungrammatical by grammar |

Table 1: F.W.Harwood’s terms

“We can now, ” he wrote,

define a measure of the goodness of fit of a syntactic system S to a language L . We define the positive fit (F) ... of a syntactic system s to a language L as:

$$F = \frac{|K_p \cap L_p|}{|L_p|} \quad (5)$$

This is a figure which today we call the *recall*, in the context of computational linguistics. Harwood gives as his *negative fit* f another expression.

¹It is essentially the product of the size of each of the categories in the sequence.

$$f = \frac{|(U_p - K_p) \cap (U_p - L_p)|}{|U_p - L_p|} \quad (6)$$

The ideal case is where $F = f = 1$, i.e. where s generates all and only the sequences in L_p . The S with $F = 1$, $f=0$ states that all of the N possible sequences may occur.

In general, Harwood reminds us, “it is the general aim to have S small in relation to” the number of sentences in the language, both observed and predicted:

Compactness is an important and measurable feature, but we need to consider the effect on the goodness of fit of S to L when S is modified to increase its compactness. In particular there is little point in securing compactness at the expense of negative fit. How close an approximation to perfect fit we require will depend on our purposes, but it is always necessary to have an estimate of what the fit is, i.e. of the values of F and f ... (411)

2.1.2 Generative grammar

Within generative grammar at the time, the suggestion was made – by Chomsky, who developed all of the infrastructure for this approach – that what we have called ‘goodness of fit’ could be described by in a very non-quantitative way: on this account, grammars would either have ‘goodness of fit’ or they wouldn’t.² The phrase used at the time, in the 1950s, was that we said that grammars would, or would not, “satisfy external criteria of adequacy.” (*Syntactic Structures* 1957 p. 54.)³

In early work on generative grammar, this condition was taken to be abrupt and discontinuous, rather than smooth, when it was considered at all: it was assumed that some grammars *did* describe the data, others did *not*; so the UG function would be expressed along the following lines:

$$UG(g_1, g_2, d) = UG'(m(g_1, g_2), n(g_1, d), n(g_2, d)) \quad (7)$$

$$m(g_1, g_2) = |g_2| - |g_1| \quad (8)$$

$$n(g, d) = \begin{cases} 1 & \text{if } g \text{ satisfactorily generates } d; \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

$$UG(g_1, g_2, d) = \begin{cases} 1 & \text{if } n(g_1, d) = 1 \text{ and } n(g_2, d) = 0 \\ -1 & \text{if } n(g_1, d) = 0 \text{ and } n(g_2, d) = 1 \\ m(g_1, g_2) & \text{otherwise.} \end{cases} \quad (10)$$

As Chomsky notes, If we had two grammars that both satisfied that criterion, then we would turn to grammar complexity as our way to selecting between the grammars.

²Carl Hempel, in the classic 1945 paper cited below, provides the locus classicus for such pessimism.

³Any literate reader in the 1950s would understand that the phrase is an allusion to the work of Carl Hempel on the logic of confirmation (notably “Studies in the logic of confirmation” 1945); he says there, for the first time, what would eventually come to be taken to be simple common sense.

The establishment of a general theory of confirmation may well be regarded as one of the most urgent desiderata of empirical science. Indeed it seems that a precise analysis of the concept of confirmation is a necessary condition for an adequate solution of various fundamental problems concerning the logical structure of scientific procedures... while certain “inductivist” accounts of scientific procedure seem to assume that relevant evidence, or relevant data, can be collected in the context of an inquiry prior to the formulation of any hypothesis, it should be clear upon brief reflection that relevance is a relative concept; experiential data can be said to be relevant or irrelevant only with respect to a given hypothesis; in other words, if it either confirms or disconfirms the hypothesis... the quest for rules of induction in the original sense of canons of scientific discovery has to be replaced, in the logic of science, by the quest for general objective criteria determining (a) whether and—if possible—even (b) to what degree, a hypothesis H may be said to be corroborated by a given body of evidence E . This approach differs essentially from the inductivist conception of the problem in that it presupposes not only E , but also H as given and then seeks to determine a certain logical relationship between them... as follows:

1. To give precise definitions of the two non-quantitative relational concepts of confirmation and of disconfirmation ...
2. To lay down criteria defining a metrical concept ‘degree of confirmation of H with respect to E ’, whose values are real numbers, or failing this, to lay down criteria defining two relational concepts, ‘more highly confirmed than’ and ‘equally well confirmed with’, which make possible a non-metrical comparison of hypotheses...

2.1.3 Ray Solomonoff

Ray Solomonoff was one of the founders of modern algorithmic information theory, and a participant in the storied 1956 Dartmouth Summer Research Conference on Artificial Intelligence. Solomonoff got his bachelor's and master's degrees from the University of Chicago, where he studied philosophy with Rudolf Carnap, and mathematical biology with Anatol Rapoport and Nicolas Rashevsky.

By 1959, Solomonoff's formulations of his ideas featured Chomsky's generative grammar in a central way.

On reading Chomsky's "Three Models for the Description of Language" (Cho 56), I found his rules for generating sentences to be very similar to the techniques I had been using in the 1957 paper to create new abstractions from old, but his grammars were organized better, easier to understand, and easier to generalize. It was immediately clear that his formal languages were ideal for induction. Furthermore, they would give a kind of induction that was considerably different from techniques used in statistics up to that time. The kinds of regularities it could recognize would be entirely new.

At the time of Chomsky's paper, I was trying to find a satisfactory utility evaluation function for my own system. I continued working on this with no great success until 1958, when I decided to look at Chomsky's paper more closely. It was easy for me to understand and build upon. In a short time, I devised a fast left to right parser for context free languages and an extremely fast matrix parser for context sensitive languages. It took advantage of special 32 bit parallel processing instructions that most computers have. My main interest, however, was learning. I was trying to find an algorithm for the discovery of the "best" grammar for a given set of acceptable sentences. One of the things sought for: Given a set of positive cases of acceptable sentences and several grammars, any of which is able to generate all of the sentences—what goodness of fit criterion should be used? It is clear that the "Ad-hoc grammar", that lists all of the sentences in the corpus, fits perfectly. The "promiscuous grammar" that accepts any conceivable sentence, also fits perfectly. The first grammar has a long description, the second has a short description. It seemed that some grammar half way between these, was "correct"—but what criterion should be used?⁴

The real breakthrough came with my invention of probabilistic languages and their associated grammars. In a deterministic (non-probabilistic) language, a string is either an acceptable sentence or it is not an acceptable sentence. Taking a clue from Korzybski—we note that in the real world, we usually don't know for sure whether anything is true or false—but we can assign probabilities. Thus a probabilistic language assigns a probability value to every possible string. In a "normalized" language, the total probability of all strings is one. It is easy to give examples of probabilistic grammars: any context free or context sensitive generative grammar can be written as a set of rewrite rules with two or more choices for each rewrite. If we assign probabilities to each of the choices, we have a probabilistic grammar.

The way probabilistic grammars define a solution to the "positive examples only" induction problem:

Each possible non-probabilistic grammar is assigned an a priori probability, by using a simple probabilistic grammar to generate non-probabilistic grammars. Each non-probabilistic grammar that could have created the data set can be changed to a probabilistic grammar by giving it probabilities for each of its choices. For the particular data set of interest, we adjust these probabilities so the probability that the grammar will create that data set is maximum. Given the data d_i , the probability that a particular grammar G_j , created d_i is the a priori probability of G_j multiplied by the probability that d_i would be generated if we knew G_j to be the generating grammar (Bayes' Theorem). We then chose the grammar for which this product is largest. The promiscuous grammar has high a priori probability, but assigns low probability to the data. The ad-hoc grammar has very low a priori probability, but assigns probability 1 to the data. These are two extreme grammar types: the best choice is usually somewhere between them.

2.2 Part Two: Grammar complexity and algorithmic complexity

Let's turn now to the second major question, which is how we can evaluate grammars independent of, or (as we say) *prior to*, bringing the grammar into contact with the data. That this is a meaningful question is the

⁴The discovery of algorithmic probability. Ray Solomonoff.

crucial insight of generative grammar, and is implemented by developing an evaluation metric for grammars. Time does not permit elaborating the whole system, but it rests crucially on the following six ideas:

1. There are an infinite number of grammars;
2. *Given* that the grammars are expressed in a finite alphabet, they can be meaningfully ranked in terms of length (which is a stand-in for complexity)
3. The same effects will be the result of many grammars (an infinite number of grammars, in fact), but we will be OK if we only consider the shortest of these; the imprecision won't hurt us in the long run;
4. The specifics of how we do all this depend (but only to a certain degree) on the choice of the underlying 'machine' (the UTM);
5. But choice of that underlying machine can be thrashed out in the court of scientific competition.
6. An analysis of data that will be evaluated by the length of the grammar, or program for a Turing machine, need not be written in machine code; it can be written in a higher order language for grammars, as long as we include a 'compiler' that turns grammars into machine code, and as long as we recognize that we have to take the length of the compiler into account as part of the complexity of the analysis.

I don't have time to go through each of these points in detail today, but I hope I have described enough for you to see how these pieces can be made to fit together to form the structure that we need.

3 Minimum Description Length analysis: Rissanen

With the work of Jorma Rissanen in the 1970s and 1980s we find a natural resting point for the trends that we have been looking at. Rissanen proposed what he called Minimum Description Length analysis, a development of many of the ideas presented already, with some additions that are strictly his own.

In machine learning, the value of probabilistic models has very little to do—almost nothing to do, really—with fuzzy or uncertain data (though that is probably what most linguists associate with it). Probability theory has traditionally been linked to the treatment of problems which may always leave us in some uncertainty as to the correct answer. Probability theory, viewed this way, aims to teach us

Given a set of data d , we seek the most likely hypothesis \hat{g} :

$$\begin{aligned}
 \hat{g} &= \operatorname{argmax}_g p(g|d) \\
 &= \operatorname{argmax}_g p(d|g) p(g) \\
 &= \operatorname{argmin}_g \underbrace{-\log \operatorname{pr}(d|g)}_{\text{Goodness of fit}} \underbrace{-\log \operatorname{pr}(g)}_{\text{Length of grammar in binary format}}
 \end{aligned}$$

Table 2: Essence of MDL (Minimum description length)

This last expression —the one for which we try to find the hypothesis that minimizes it — is a sum of two terms, and these two terms will be very familiar. The first term is the measure of the goodness of fit of hypothesis or grammar g to the data d , while the second term, $-\log \operatorname{pr}(g)$, is the length of the grammar g when expressed in some simple binary format [I leave the details out here, because they don't matter for our basic point].

For any given set of data, and a way of generating it with a probabilistic grammar or model, we can compute a *description length* of the data. For a fixed set of data, there will be a different description length for each probabilistic grammar that generates it. The description length of the data is a sum of two terms in Table 1: the first is the measure of the goodness of fit of the data to the model, and the second is the measure of complexity of the model (independent of the data that is being modeled).

$$DL(\text{data}, \text{grammar}) = \text{Length}(\text{grammar}) + \log_2 \left(\frac{1}{\operatorname{pr}(\text{data}|\text{grammar})} \right) \quad (11)$$

The goal is to find the grammar that minimizes this expression: one that is both short and which fits the data as well as possible, with an appropriate trade-off between these two tendencies, always in opposition to each other.

What is most astonishing about this insight is the realization that two very different quantities are measured in precisely the same units: the complexity of a grammar, and the inverse goodness of fit of a grammar to the data. This is a very hard idea to wrap one's head around—but it's true!

4 Conclusion

4.1

The conclusion that I draw from this brief history of generative grammar and machine learning is that the notion of the evaluation metric still has a lot of good life to it, and it deserves our continued consideration. Minimum Description Length (MDL) analysis strongly suggests that it is possible to offer a serious measure of goodness-of-fit between our theory and a body of data. For some, that is very good news; for others, it may not be. For myself, I think it is excellent news, and it promises that we can give explicit accounts of how a change in grammatical analysis provides better (or, God forbid, worse) goodness of fit to the data we have from a language. I think that measurement of grammar complexity should form a serious and central part of the concerns of the theorist in syntax, morphology, and phonology, and I look forward to working on this with all of you.