Research Group for
Quantitative Linguistics
Fack
Stockholm 40
SWEDEN

KVAL PM 339
June 19, 1967

The Entropy of Recursive Markov Processes

By

BENNY BRODDA

The work reported in this paper has been sponsored by Humanistiska forskningsrådet, Tekniska forskningsrådet and Riksbankens Jubileumsfond, Stockholm, Sweden.

THE ENTROPY OF RECURSIVE MARKOV PROCESSES

By

BENNY BRODDA

KVAL, Fack, Stockholm 40, Sweden

Summary

The aim of this communication is to obtain an explicit formula for calculating the entropy of a source which behaves in accordance with the rules of an arbitrary Phrase Structure Grammar, in which relative probabilities are attached to the rules in the grammar. With this aim in mind we introduce an alternative definition of the concept of a PSG as a set of self-embedded (recursive) Finite State Grammars; when the probabilities are taken into account in such a grammar we call it a Recursive Markov Process.

1. In the first section we give a more detailed definition of what kind of Markov Processes we are going to generalize later on (in sec. 3), and we also outline the concept of entropy in an ordinary Markov source. More details of information may be found, e.g., in Khinchins "Mathematical Foundations of Information Theory", N.Y., 1957, or "Information Theory" by R. Ash, N.Y., 1965.

A Markov Grammar is defined as a Markov source with the following properties:

Assume that there are n + 1 states, say S_0, S_1, ..., S_n, in the source. S_0 is defined as the initial state and S_n is defined as the final state and the other states are called intermediate states. We shall, of course, also have a transition matrix, $M = (p_{ij})$, containing the transition probabilities of the source.

a) A transition from state S_i to state S_k is always accompanied by a production of a (non-zero) letter a_{ik} from a given finite alphabet. Transition to different states from one given state always produce different letters.

b) From the initial state, S_0, direct or indirect transitions should be possible to any other state in the source. From no state is a transition to S_0 allowed.

c) From any state, direct or indirect transitions to the final state S_n should be possible. From S_n no transition is allowed to any other state (S_n is an "absorbing state").

The work reported in this paper has been sponsored by Humanistiska forskningsrådet, Tekniska forskningsrådet and Riksbankens Jubileumsfond, Stockholm, Sweden.

A (grammatical) sentence should now be defined as the (left-to-right) concatenation of the letters produced by the source, when passing from the initial state to the final state.

The length of a sentence is defined as the number of letters in the sentence. To simplify matters without dropping much of generality we also require that

d) The greatest common divisor for all the possible lengths of sentences is = 1 (i.e., the source becomes an aperiodic source, if it is short-circuited by identifying the final and initial states).

With the properties a - d above, the source obtained by identifying the final and initial states is an indecomposable, ergodic Markov process (cf. Feller, "Probability Theory and Its Applications", ch. 15, N.Y., 1950).

In the transition matrix M for a Markov grammar of our type all elements in the first column are zero, and in the last row all elements are zero except the last one which is = 1. For a given Markov grammar we define the uncertainty or entropy, H_i, for each state S_i, $i = 0, 1, \ldots, n$, as:

$$H_i = -\sum_{j=0}^{n} p_{ij} \log p_{ij}; \; i = 1, 2, \ldots, n.$$

We also define the entropy, H or H(M), for the grammar as

$$(1). \qquad H = \sum_{i=0}^{n=1} x_i H_i$$

where $x = (x_0, x_2, \ldots, x_{n-1})$ is defined as the stationary distribution for the source obtained when S_0 and S_n are identified; thus x is defined as the (unique) solution to the set of simultaneous equations

$$(2) \qquad \begin{aligned} xM_1 &= x \\ x_0 + x_1 + \ldots + x_{n-1} &= 1 \end{aligned}$$

where M_1 is formed by shifting the last and first columns and then omitting the last row and column. The mean sentence length, μ, of the set of grammatical sentences can now be easily calculated as

(3) $\mu = 1/x_0$

(cf. Feller, op. cit.)

2. Embedded Grammars

We now assume that we have two Markov grammars, M and M_1, with states S_0, S_1, ..., S_n, and T_0, T_1, ..., T_m, respectively, where S_0 and S_n, T_0 and T_m are the corresponding initial and final states. Now consider two states S_i and S_k in the grammar M; assume that the corresponding transition probability is = p_{ik}. We now transform the grammar, M_1, into a new one, M_1', by <u>embedding</u> the grammar M_2 in M_1 between the states S_i and S_k, an operation which is performed by identifying the states T_0 and T_m with the states S_i and S_k respectively. Or, to be more precise, assume that in the grammar M_1 the transitions to the states T_j, $j \geq 1$, has the probabilities q_{0j}. Then, in the grammar M', transitions to a state T_j from the state S_i will take place with the probability = $p_{ik}q_{0j}$. A return to the state S_k in the "main" grammar from an intermediate state T_j in M_1 takes place with the probability q_{jm}.

With the conditions above fulfilled, we propose that the entropy for the composed grammar be calculated according to the formula:

$$(4) \qquad H(M') = \frac{H(M) + x_i p_{ik} \cdot \mu_1 \cdot H(M_1)}{1 + x_i p_{ik} (\mu_1 - 1)}$$

where H(M) is the entropy of the grammar M when there is an ordinary connection (with probability p_{ik}) between the states S_i and S_k, and where x_i is the inherent probability of being in the state S_i under the same conditions. μ_1 is the mean sentence length of the sentences produced by the grammar M_1 alone. (It is quite natural that this number appears as a weight in the formula, since if one is producing a sentence according to the grammar M and arrives at the state S_i and from there "dives" into the grammar M_1, then μ_1 is the expected waiting time for emerging again in the main grammar M.) The factor $x_i p_{ik}$ may be interpreted as the combined probability of ever arriving at S_i and there choosing the path over to M_1 (you may, of course, choose quite another path from S_i).

The proof of formula (4) is very straightforward, once the premises according to the above have been given, and we omit it here, as it does not give much extra insight to the theory. The formula may be extended to the case when there are more than one sub-grammar embedded in the grammar M', by adding similar terms as the one standing to the right in the numerator and the denominator. The important thing here is that the factors of the type $x_i p_{ik}$ depend only on the probability matrix for the grammar M and are dependent of the sub-grammars involved.

3. <u>Recursive or Self-embedded Sources</u>

It is now quite natural to allow a grammar to have itself as a sub-grammar or to allow a grammar M_1 to contain a grammar M_2 which, in its turn, contains M_1, and so on. The grammars thus obtained cannot, however, be rewritten as an ordinary Markov grammar. The relation between an ordinary Markov grammar and a <u>recursive</u> one is exactly similar to the relation between Finite State Languages and Phrase Structure Languages.

To be more precise, assume that we have a set of Markov grammars M'_0, M'_1, ..., M'_N where M'_0 is called the <u>main</u> grammar and in the sense that the process always starts at the initial state in M'_0 and ceases when it reaches the final state in M_0. Each of the grammars may contain any number of the others (and itself) as sub-grammars. The only restriction is that from any state in any one of the grammars there should exist a path which ends up at the final state of M_0.

<u>Remark</u>

If we interpret a source of our kind as a Phrase Structure Language, the rewriting rules are all of the following kind:

$$(5) \qquad S_i \rightarrow A_{ik} + S_k \qquad \underline{\text{or}} \qquad S_n \rightarrow \#;$$

where the S's are all non-terminal symbols. (They stand for the names of the states in the sources - M'_0, M'_1, ..., M'_N and where S_0 is assumed to be the initial symbol /the Chomskyan S/ and S_n is the terminating state which produces the sentence delimiter #. The symbols A_{ik} are either terminal symbols /letters from a finite alphabet/ or non-terminal symbols equal to the name of the initial state in one of the grammars M'_0, M'_1, ..., M'_N /one may also say that A_{ik} stands as an abbreviation for an arbitrary sentence of that grammar/.)

We associate each grammar M_j' with the grammar M_j, $j = 0, 1, \ldots, N$, by just considering it as a non-recursive one, that is, we consider all the symbols A_{ik} as terminal symbols (even if they are not). The grammars thus obtained are ordinarily Markov grammars according to our definition, and the entropies $H_j = H(M_j)$ are easily computed according to formula (1), as are the stationary distributions /formula (2)/. The follwoing theorem shows how the entropies H_j' for the fully recursive grammars M_j' are connected with the numbers H_j.

Theorem

The entropy H_j' for a set of recursive Markov grammar M_j', $j = 0, 1, \ldots, N$, can be calculated according to the formula

$$(6) \qquad H_j' \{1 + \sum_k y_{jk} (\mu_k - 1)\} - \sum_k y_{jk} \mu_k H_k' = H_j$$

$$j = 0, 1, \ldots, N.$$

Here the factors y_{jk} are dependent only of the probability matrix of the grammar and the numbers μ_k defined as the mean sentence length of the sentences of the grammar M_k', $k = 0, 1, \ldots, N$, and computable according to lemma below.

H_0' is the entropy for the grammar.

The theorem above is a direct application for the grammar of formula (4), sec. 2.

The coefficients y_{jk} in formula (6) can, more precisely, be calculated as a sum of terms of the type $x_i p_{im}$ with the indices (i, m) are where the grammar M_k' appears in the grammar M_j'; x_i and p_{im} are the components the stationary distribution and probability matrix for the grammar M_j.

Assume now that we have a Markov grammar of our type, but for which each transition will take a certain amount of time. A very natural question is then: "What is the expected time to produce a sentence in that language?" The answer is in the following lemma.

Lemma

Let M be a Markov grammar with states S_i, $i = 0, 1, \ldots, n$, where S_0 and S_n are the initial and final states respectively.

Assume that each transition $S_i \to S_k$ will take y_{ik} time units.

Denote the expected time for arrival at S_n given that the grammar is in state S_i by t_i, $i = 0, 1, \ldots, n$, (thus t_0 is the expected time for producing a sentence). The times t_1 will then fulfill the following set of simultaneously linear equations:

$$(7) \qquad t_i = \sum_k p_{ik} (t_{ik} + t_k)$$

Formula (7) is itself a proof of the lemma.

With more convenient notations we can write (7) as

$$(E - P)\, t = p_t$$

where E is the unit matrix, P is the probability matrix (with $P_{nn} = 0$) and p_t is the vector with components

$$p_i\,(t) = \sum_m p_{im}\, t_{im}, \quad i = 0, 1, \ldots, n.$$

The application of the lemma for computing the numbers μ_k in formula (6) is now the following.

The transition times of the lemma are, of course, the expected time (or "lengths" as we have called it earlier) for passing via a sub-grammar of the grammar under consideration. Thus the number t_{ik} is itself the unknown entities μ_k.

For each of the sub-grammars M_j', $j = 0, 1, \ldots, N$, we get a set of linear equations of type (7) for determining the vectors t of lemma. The first component of this vector, i.e., the number t_0, is then equal to the expected length, μ, of the sentences of that grammar. (Unfortunately, we have to compute extra the expected time for going from any state of the sub-grammars to the corresponding final state.)

The total number of unknowns involved when computing the entropy of our grammar (i.e., the entropy H_0') is equal to

(the total number of states in all our sub-grammars) plus
(the number of sub-grammars).

This is also the number of equations for we have $n + 1$ equations from formula (6) and then $(n + 1)$ sets of equations of the type (7). We assert that all these simultaneous equations are solvable, if the grammar fulfills the conditions we earlier stated for the grammar, i.e., that from each state in any sub-grammar exists at least one path to the final state of that grammar.