chunking_grammar

February 14, 2023

1 Chunking Grammar

The purpose of this notebook is to explore a simple theory of sequence analysis using a context-free grammar that incorporates sequential dependencies. The theory is derived from fundamental principles using the notion of *chunking*. The initial rationale for this model comes from the short (and somewhat cryptic) papers of (Kupiec), although its motivation from the idea of chunking is my own invention, and ultimately the model equations do not agree with those of (Kupiec).

Further motivation for this model comes from the hierarchical HMMs of (Fine, Singer and Tishby) and (Bui, Phung and Venkatesh).

1.1 Introduction

The context-free grammar $\mathcal{G}$ under consideration is not restricted to Chomsky normal form (CNF). A CNF grammar has only binary rules, e.g. $\texttt{NP} \rightarrow \texttt{D} \oplus \texttt{N}$, and unary rules, e.g. $\texttt{N} \rightarrow \textit{cat}$, where here NP (noun-phrase), D (determiner) and N (noun) are non-terminal symbols and *cat* is a terminal symbol. Instead, we shall allow arbitrary-length non-terminal rules, e.g. $\texttt{NP} \rightarrow \texttt{D} \oplus \texttt{J} \oplus \texttt{N}$. The productions of such rules form contiguous *chunks* of non-terminal symbols (henceforth called *states*), e.g. $(\texttt{D}, \texttt{J}, \texttt{N})$, having sequential dependencies between the states.

The characteristic of being context-free is interpreted here in the Markov sense as meaning that each state depends only upon the previous state, and not upon past history. An example of a context-free derivation exhibiting sequential dependencies is shown in the figure below.

The grammar $\mathcal{G}$ defines a finite vocabulary $\mathcal{Y} = \{\nu_1, \nu_2, \ldots\}$ of discrete terminal symbols, called *tokens*, and a finite set $\mathcal{S} = \{\sigma_1, \sigma_2, \ldots\}$ of discrete non-terminal symbols, called *states*. The subset $\mathcal{S}_{\texttt{leaf}} \subseteq \mathcal{S}$ of states that may directly be used to generate tokens are called *leaf* states. The subset $\mathcal{S}_{\texttt{root}} \subseteq \mathcal{S}$ of states that may be used at the root of a derivation are called *root* states. The subset $\mathcal{S}_{\texttt{int}} \subseteq \mathcal{S}$ of states that may appear in a derivation between the root state and the leaf states are called *intermediate* states. Although $\mathcal{S} = \mathcal{S}_{\texttt{root}} \cup \mathcal{S}_{\texttt{int}} \cup \mathcal{S}_{\texttt{leaf}}$, there is, in general, no further restriction regarding whether the various subsets overlap or are mutually exclusive. Such restrictions, if required, must be built into the grammar by the presence of so-called *structural zeroes* in the conditional probability tables that dictate the stochastic nature of the grammar.

1.1.1 Sequence generation

The stochastic grammar $\mathcal{G}$ should be capable of generating sequences. For the example sentence "*The black cat purred.*", the specific derivation shown above has a particular probability of being generated. In abbreviated form, this probability is

$$P(\texttt{D} \mid \triangleleft)\, P(\textit{The} \mid \texttt{D})\, P(\oplus \mid \texttt{D})\, P(\texttt{J} \mid \texttt{D})\;\; P(\textit{black} \mid \texttt{J})\, P(\oplus \mid \texttt{J})\, P(\texttt{N} \mid \texttt{J})\;\; P(\textit{cat} \mid \texttt{N})\, P(\square \mid \texttt{N}) \qquad (1)$$

$$P(\texttt{NP} \mid \texttt{N})\, P(\oplus \mid \texttt{NP})\, P(\texttt{V} \mid \texttt{NP})\, P(\textit{purred} \mid \texttt{V})\, P(\square \mid \texttt{V})\, P(\texttt{VP} \mid \texttt{V})\, P(\square \mid \texttt{VP})\, P(\texttt{S} \mid \texttt{VP})\, P(\square \mid \texttt{S}) \quad (2)$$

Mnemonically, the leaf state sequence is $\triangleleft\texttt{D} \oplus \texttt{J} \oplus \texttt{N}\square\texttt{V}\triangleright$, which corresponds to the partitioning, or *chunking*, $\langle(\texttt{D}, \texttt{J}, \texttt{N})(\texttt{V})\rangle$ of the complete token sequence $\langle \textit{The}, \textit{black}, \textit{cat}, \textit{purred}\rangle$.

Some explanation is clearly in order here. Firstly, the *marker* symbol '$\triangleleft$' is used to indicate the start of a sequence. Marker symbols are used to denote the internal context of the stochastic process. Marker symbols are never externalised, and hence operate in conjunction with the context-free state-to-state transitions. Thus, the corresponding marker '$\triangleright$' indicates the end of a complete sequence. In addition, the marker '$\square$' denotes the end of a subsequence (or *chunk*), and the corresponding marker '$\oplus$' denotes a continuation of the subsequence.

Starting a new sequence automatically starts a new *chunk* (explained further in a later section) at the *leaf* state level, which is the lowest non-terminal level. Starting a new chunk triggers the generation of an initial state. The leaf level is special in that the production of a leaf state triggers the generation of its corresponding *token* (or terminal symbol). The production of tokens also operates in conjunction with the context-free state-to-state transitions.

While a chunk is *open*, i.e. the rule has not yet completed, a stochastic choice is made as to whether to *close* the chunk or keep it open. As mentioned, the marker '$\square$' is used to indicate closure, such that $P(\square \mid X)$ is the probability of closing the chunk immediately after state X. Conversely, $P(\oplus \mid X) = 1 - P(\square \mid X)$ is the probability of keeping the chunk open. If the chunk remains open, then there is a transition to a leaf state in the next position, and this state is appended to the open chunk (hence the reason for the marker '$\oplus$').

When a chunk is closed, this (usually) triggers the generation of a parent state assigned to the chunk at a higher level. If no open parent chunk currently exists, then one is created. The parent state is then appended to the open parent chunk. At this higher level, a stochastic decision is again made as to whether to close the parent chunk or keep it open. If the chunk remains open, then a new chunk is started at the leaf level, and a leaf state is generated.

The generation process terminates when a single-state chunk is closed and no parent chunk exists at the next higher level. The closure of this highest-level *root* chunk automatically triggers the closure of the derivation.

1.1.2 Sequence parsing

The converse of sequence generation, as discussed in the previous section, is sequence parsing. Here the goal is to start with an observed sequence of tokens, and to produce the (or a) most probable derivation. However, since most generative grammars are designed in a top-down fashion, and parsing usually proceeds in a bottom-up fashion, there is typically a disconnection between the two approaches.

However, the aim is to design a simplified grammar that can easily be used for both sequence generation and sequence parsing. For example, one parsing model of the derivation shown in the previous section might be

$$P(\texttt{D} \mid \triangleleft, \mathit{The})\, P(\oplus \mid \texttt{D})\, P(\texttt{J} \mid \texttt{D}, \mathit{black})\, P(\oplus \mid \texttt{J})\, P(\texttt{N} \mid \texttt{J}), \mathit{cat})\, P(\square \mid \texttt{N}) \tag{3}$$

$$P(\texttt{NP} \mid \texttt{N})\, P(\oplus \mid \texttt{NP})\, P(\texttt{V} \mid \texttt{NP}, \mathit{purred})\, P(\square \mid \texttt{V})\, P(\texttt{VP} \mid \texttt{V})\, P(\square \mid \texttt{VP})\, P(\texttt{S} \mid \texttt{VP})\, P(\square \mid \texttt{S})\,. \tag{4}$$

As noted above, the generative grammar defines terms like $P(\texttt{D} \mid \triangleleft)$ and $P(\mathit{The} \mid \texttt{D})$, not $P(\texttt{D} \mid \triangleleft, \mathit{The})$. However, via Bayes' rule we find that

$$P(\texttt{D} \mid \triangleleft, \mathit{The}) \quad = \quad \frac{P(\texttt{D} \mid \triangleleft)\, P(\mathit{The} \mid \texttt{D})}{\sum_{\sigma \in \mathcal{S}_{\texttt{leaf}}} P(\sigma \mid \triangleleft)\, P(\mathit{The} \mid \sigma)}\,, \tag{5}$$

and so we may define the probabilities required for parsing in terms of the probabilities required for generating a derivation.

Conversely, we may (in some circumstances) define the derivation probabilities in terms of the parsing probabilities. The conversion between derivation and parsing relies on the fact that, again via Bayes' rule, we have

$$\frac{P(\nu_m \mid \sigma_i)}{P(\nu_m)} \quad = \quad \frac{P(\sigma_i \mid \nu_m)}{P(\sigma_i)}\,, \tag{6}$$

for all leaf states $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$ and all tokens $\nu_m \in \mathcal{Y}$. Consequently, although we typically do not know the unconditional token probability $P(\nu_m)$ for $\nu_m \in \mathcal{Y}$, we may substitute

$$P(\nu_m \mid \sigma_i) \quad \propto \quad \frac{P(\sigma_i \mid \nu_m)}{P(\sigma_i)}\,, \tag{7}$$

on the basis that the unknown term $P(\nu_m)$ cancels out when conditioning on the observed tokens, which typically involves only summations over the leaf states $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$, e.g.

$$P(\texttt{D} \mid \triangleleft, \mathit{The}) \quad = \quad \frac{\frac{P(\texttt{D} \mid \triangleleft)\, P(\texttt{D} \mid \mathit{The})}{P(\texttt{D})}}{\sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \frac{P(\sigma_i \mid \triangleleft)\, P(\sigma_i \mid \mathit{The})}{P(\sigma_i)}}\,. \tag{8}$$

1.2 Chunking

Chunking is the process of partitioning a sequence into contiguous subsequences, such that each subsequence, henceforth called a *chunk*, has self-consistent semantics with respect to the grammar $\mathcal{G}$. For example, an English sentence could be chunked into noun phrases and verb phrases, et cetera. Note that although a chunk is a subsequence, an arbitrary subsequence is not necessarily a chunk. Also note that chunks may themselves be chunked, leading to a nested derivation tree. However, this tree is not restricted to binary branches, nor does it exclude unary 'branches' from parent state to child state.

Suppose the stochastic process has generated a complete sequence $\mathbf{y}_{1:T} = \langle y_1, y_2, \ldots, y_T \rangle$ of tokens $y_t \in \mathcal{Y}$. Here the marker symbols '<' and '>' respectively denote the start and end of a complete (i.e. terminated) sequence. For our example sentence, with the derivation shown in the introduction, we have the token sequence $\mathbf{y}_{1:4} = \langle \mathit{The}, \mathit{black}, \mathit{cat}, \mathit{purred} \rangle$, with the corresponding chunking $\mathbf{y}_{1:4}^{\texttt{chunk}} = \langle (\mathit{The}, \mathit{black}, \mathit{cat})(\mathit{purred}) \rangle$, where the marker symbols '(' and ')' respectively denote the start and end of a chunk of contiguous elements.

The leaf states of the derivation have also been chunked, namely as $\mathbf{s}_{1:4}^{\texttt{leaf}} = \langle(\texttt{D}, \texttt{J}, \texttt{N})(\texttt{V})\rangle$, and the *intermediate* states (those at levels above the leaf states but below the *root* state) have been chunked as $\mathbf{s}_{1:2}^{\texttt{int}} = \langle(\texttt{NP}, \texttt{VP})\rangle$. The root level chunk is $\mathbf{s}_{1}^{\texttt{root}} = \langle(\texttt{S})\rangle$.

During the process of chunking, a key notion is whether a given state subsequence $\mathbf{s}_{r:t}$ comprises a complete chunk or only part of a chunk. A complete chunk, represented as $\mathbf{s}_{r:t} = (s_r, \ldots, s_t)$, has a definite start and a definite end, where '(' denotes the immutable start of the chunk, and ')' denotes the immutable end of the chunk. This is called *closed* because no further states may be appended.

Conversely, an incomplete chunk has a definite start but (as yet) only an indefinite end, and is represented as $\mathbf{s}_{r:t} = (s_r, \ldots, s_t]$, where the marker ']' denotes the *mutable* 'end' of the chunk. This is called *open* because it may potentially have zero, one or more additional states appended to it, before being closed.

Hence, from the derivation, we have $\mathbf{s}_{1:1}^{\texttt{leaf}} = (\texttt{D}]$ and $\mathbf{s}_{1:2}^{\texttt{leaf}} = (\texttt{D}, \texttt{J}]$, but $\mathbf{s}_{1:3}^{\texttt{leaf}} = (\texttt{D}, \texttt{J}, \texttt{N})$.

1.2.1 Hierarchical chunking

The procedure described for sequence generation essentially produces a sequence derivation that represents hierarchical chunking. A summary of this procedure, slightly modified, is as follows:

1. The first token y_1 in a sequence $\mathbf{y}_{1:T}$ is paired with its corresponding leaf state s_1. This state starts an open leaf chunk, $\mathbf{s}_{1:1}^{\texttt{leaf}} = (s_1]$.
2. For $1 \leq r \leq t \leq T$, consider the current open leaf chunk $\mathbf{s}_{r:t}^{\texttt{leaf}} = (s_r, s_{r+1}, \ldots, s_t]$.
   - If $t < T$, then a stochastic decision is made whether to close the chunk or keep it open. However, if $t = T$, then every open chunk will be closed, in order from the lowest level to the highest level.
   - If the chunk is kept open, then the state s_{t+1} of the next token y_{t+1} is appended to the chunk, giving $\mathbf{s}_{r:t+1}^{\texttt{leaf}} = (s_r, \ldots, s_t, s_{t+1}]$. The chunking process now loops to position $t + 1$.
3. However, if the chunk is closed, then it is now represented by $\mathbf{s}_{r:t}^{\texttt{leaf}} = (s_r, \ldots, s_t)$.
   - A higher level parent state $s_{r:t}^{\texttt{int}}$ is now assigned to the closed leaf chunk, and this parent state is appended to the open parent chunk (which is created as necessary).
   - If $t < T$, then a stochastic decision is made whether to close the parent chunk or keep it open. However, if $t = T$, then the parent chunk will be closed.
   - If the parent chunk remains open, then the closed lead chunk $\mathbf{s}_{r:t}^{\texttt{leaf}}$ is succeeded by an adjacent open leaf chunk $\mathbf{s}_{t+1:t+1}^{\texttt{leaf}} = (s_{t+1}]$. The chunking process now loops to step 2 at position $t + 1$.
   - However, if the parent chunk is closed, then a grandparent state is assigned, and a closure decision is made at the higher level.

We shall not pursue hierarchical chunking any further here, although a full treatment is required for parsing. Instead, we consider a simplified process that uses only the leaf state level and a single, intermediate state level.

1.2.2 Two-level chunking

For our simplified model, we consider only the complete sequence $\mathbf{y}_{1:T}$ of tokens, along with a corresponding sequence $\mathbf{s}^{\texttt{leaf}}_{1:T}$ of leaf states, and an arbitrary-length, secondary sequence $\mathbf{s}^{\texttt{int}}$ of intermediate states. For convenience, we drop the superscript '`leaf`', on the understanding that use of a state variable S_t implies a leaf state $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$ spanning token y_t. Similarly, use of the state variable $S_{r:t}$ implies an intermediate state $\sigma_p \in \mathcal{S}_{\texttt{int}}$ assigned to a closed chunk that spans leaf states $\mathbf{s}_{r:t}$ and thus tokens $\mathbf{y}_{r:t}$. In addition, we recognise that the sequence generation process has internal context, which we have represented using marker symbols. Hence, we let the variable M_t denote the context at token position t. For more precision, we also let M_t^- denote the context immediately before position t but after position $t-1$, and also let M_t^+ denote the context immediately after position t but before position $t+1$.

The first state s_1 in the leaf sequence $\mathbf{s}_{1:T}$ is generated with probability $P(S_1 = \sigma_i \mid M_1^- = \triangleleft) \doteq \iota_i^{\triangleleft}$, where $\sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \iota_i^{\triangleleft} = 1$. The vector $\iota^{\triangleleft}$ specifies the *start-of-sequence* leaf state probabilities. Note, however, that the start of a sequence implies the start of the first chunk. Thus, for an arbitrary (open or closed) chunk $\mathbf{s}_{r:t}$ that starts at position r, we can also define $P(S_r = \sigma_i \mid M_r^- = \square) \doteq \iota_i^{\square}$, where $\iota^{\square}$ specifies the *start-of-chunk* leaf state probabilities. Since this latter quantity will see frequent use, we often drop the superscript '$\square$' for convenience. Hence, instead of $P(S_1 = \sigma_i \mid M_1^- = \triangleleft)$ we could use $P(S_1 = \sigma_i \mid M_1^- = \square)$, if we do not care to distinguish the first chunk from consequent chunks.

Similarly, the last state s_T in the leaf sequence $\mathbf{s}_{1:T}$ closes a complete sequence, and the probability of closure is given by $P(M_T^+ = \triangleright \mid S_T = \sigma_i) \doteq \tau_i^{\triangleright}$, where the marker symbol '$\triangleright$' denotes the *end-of-sequence*. Conversely, for an incomplete sequence the probability of being left open is $\bar{\tau}_i^{\triangleright} \doteq 1 - \tau_i^{\triangleright}$. Once again, the closure of a sequence implies the closure of the last chunk, and hence we could instead use $P(M_T^+ = \square \mid S_T = \sigma_i)$. More generally, we let $P(M_t^+ = \square \mid S_t = \sigma_i) \doteq \tau_i^{\square}$ be the probability of closing an open chunk $\mathbf{s}_{r:t}$, and let $P(M_t^+ = \oplus \mid S_t = \sigma_i) \doteq \bar{\tau}_i^{\square}$ denote the complementary probability of leaving the chunk open. We typically drop the superscript '$\square$' due to the frequent use of *end-of-chunk* probabilities.

Since both the sequence generation process and sequence parsing process traverse the tokens from left to right, in general we consider a single, arbitrary chunk $\mathbf{s}_{r:t} = (s_r, \ldots, s_t]$ that starts at position r with context $M_r^- = \square$, and continues without closure up to and including position t. By default, we consider a chunk as being open until explicitly closed. In other words, unless otherwise specified, we assume that the closure decision M_t^+ has yet to be made. However, in some circumstances we do have further information. For instance, if we know the chunk is closed with context $M_t^+ = \square$, then we use the representation $\mathbf{s}_{r:t} = (s_r, \ldots, s_t)$ to indicate that the subsequence closure symbol ')' has additional probability. Alternatively, if we know the chunk has been closed at position t but do not yet know the starting position of the chunk, then we have context $M_r^- = \oplus$ with representation $\mathbf{s}_{r:t} = [s_r, \ldots, s_t)$. In exceptional circumstances, we might know only $M_r^- = \oplus$ and $M_t^+ = \oplus$, giving representation $\mathbf{s}_{r:t} = [s_r, \ldots, s_t]$.

1.2.3 Forward chunk recursion

Consider the open chunk $\mathbf{s}_{r:t} = (s_r, \ldots, s_t]$ that spans the subsequence $\mathbf{y}_{r:t}$ of tokens. Now, by consideration of the derivation shown in the introduction, we observe in general that a chunk starting at position r is usually preceded by some intermediate state $S_{*:r-1} = \sigma_p \in \mathcal{S}_{\texttt{int}}$, where the index symbol '*' indicates that the start of the previous chunk is indeterminate. The exception is the first chunk with $r = 1$, which is preceded by the context $M_1^- = \triangleleft$.

We further suppose that the 'last' position t of the chunk has some leaf state $s_t = \sigma_i \in \mathcal{S}_{\texttt{leaf}}$. Hence, we define the *start-chunk* probability $\alpha_{r:t}(p, i)$ as

$$\alpha_{r:t}(p,i) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, S_t = \sigma_i \mid M^+_{r-1} = \square, S_{*:r-1} = \sigma_p)\,, \tag{9}$$

where $\mathbf{Y}_{r:t} = (Y_r, \ldots, Y_t)$ and Y_t is a variable denoting the stochastic choice of token $\nu_m \in \mathcal{Y}$ at position t. The exceptional first-chunk case is defined via

$$\alpha_{1:t}(\triangleleft, i) \doteq P(\mathbf{Y}_{1:t} = \mathbf{y}_{1:t}, S_t = \sigma_i \mid M^-_1 = \triangleleft)\,. \tag{10}$$

As previously discussed, the first chunk starts with state $s_1 = \sigma_i$ with probability

$$\alpha_{1:1}(\triangleleft, i) \doteq P(Y_1 = y_1, S_1 = \sigma_i \mid M^-_1 = \triangleleft) = P(S_1 = \sigma_i \mid M^-_1 = \triangleleft)\, P(Y_1 = y_1 \mid S_1 = \sigma_i) \tag{11}$$

As an aside, note that if we are parsing some observed sequence $\mathbf{y}$, then we may pre-compute the *leaf-token* probabilities $\breve{\mathbf{B}} = [\breve{b}_{it}]$ for each $\breve{b}_{it} \doteq P(Y_t = y_t \mid S_t = \sigma_i)$. Alternatively, if the process is generating tokens, then arbitrary token $\nu_m \in \mathcal{Y}$ may be generated from state $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$ with probability

$$P(Y_t = \nu_m \mid S_t = \sigma_i) \doteq b_{im}\,, \tag{12}$$

via the pre-specified *emission* matrix $\mathbf{B} = [b_{im}]$. Hence, for consistency between generation and parsing, we define

$$\breve{b}_{it} \doteq \sum_{\nu_m \in \mathcal{Y}} \delta(y_t = \nu_m)\, b_{im}\,. \tag{13}$$

Consequently, the first chunk has starting probability

$$\alpha_{1:1}(\triangleleft, i) \doteq \iota^{\triangleleft}_i \breve{b}_{i1}\,. \tag{14}$$

More generally, the first state of an arbitrary chunk starting at position $r > 1$ has probability

$$\alpha_{r:r}(p,i) \doteq P(Y_r = y_r, S_r = \sigma_i \mid S_{*:r-1} = \sigma_p) = d_{pi}\, \breve{b}_{ir}\,, \tag{15}$$

where the *intermediate-to-leaf* state transition matrix $\mathbf{D} = [d_{pi}]$ specifies the generation of leaf state $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$ after intermediate state $\sigma_p \in \mathcal{S}_{\texttt{int}}$ with probability

$$P(S_r = \sigma_i \mid S_{*:r-1} = \sigma_p, M^+_{*:r-1} = \oplus) \doteq d_{pi}\,. \tag{16}$$

Note that here we take $M^+_{*:r-1} = \oplus$ to mean that the intermediate chunk containing state σ_p remains open, which in turn means that the sequence does not close here, such that there **must** be a transition to a successive leaf state at position r.

In the special case where the previous intermediate state $S_{*:r-1}$ is unknown, we define

$$\alpha_{r:r}(\square, i) \doteq P(Y_r = y_r, S_r = \sigma_i \mid M^-_r = \square) = \iota^{\square}_i \breve{b}_{ir}\,, \tag{17}$$

and more generally

$$\alpha_{r:t}(\square, i) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, S_r = \sigma_i \mid M^-_r = \square)\,. \tag{18}$$

Having selected the initial leaf state s_r of the open chunk $\mathbf{s}_{r:r}$, the process makes a choice to either close the chunk with probability $P(M_r^+ = \square \mid S_r = \sigma_i) = \tau_i$, or leave it open with probability $P(M_r^+ = \oplus \mid S_t = \sigma_i) = \bar{\tau}_i$. If the chunk is left open, then it **must** be expanded to include position $r+1$, and a subsequent leaf state $s_{r+1} = \sigma_j \in \mathcal{S}_{\texttt{leaf}}$ will be chosen with probability

$$P(S_{r+1} = \sigma_j \mid S_r = \sigma_i, M_r^+ = \oplus) \quad \doteq \quad a_{ij}\,, \tag{19}$$

via the *leaf-to-leaf* state transition matrix $\mathbf{A} = [a_{ij}]$. This stochastic process repeats iteratively to ultimately generate the chunk $\mathbf{s}_{r:t}$. At this point, if the chunk $\mathbf{s}_{r:t}$ remains open, then we obtain the *forward* recurrence relation

$$\alpha_{r:t+1}(p,j) \quad = \quad \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha_{r:t}(p,i)\,\bar{\tau}_i\,a_{ij}\,\breve{b}_{j,t+1}\,. \tag{20}$$

1.2.4 Backward chunk recursion

In forward recursion, we considered an open-right chunk $\mathbf{s}_{r:t} = (s_r, \ldots, s_t]$ that formed the start of some complete chunk. The analogue to forward recursion is therefore *backward* recursion, commencing from an open-left chunk $\mathbf{s}_{t+1:w} = [s_{t+1}, \ldots, s_w)$ that forms the end of some complete chunk. Since this chunk is closed on the right, the last leaf state s_w will have triggered a transition to an intermediate state, say $s_{*:w} = \sigma_p \in \mathcal{S}_{\texttt{int}}$. We also assume that since the chunk is open (by default) on the left, then at position t there is either a continuation of the same chunk or the end of a previous chunk, with some leaf state, say $s_t = \sigma_i \in \mathcal{S}_{\texttt{leaf}}$. In other words, we do not yet know the context M_{t+1}. Since the chunk $\mathbf{s}_{t+1:w}$ spans the tokens $\mathbf{y}_{t+1:w}$, we now define the *end-chunk* probability $\beta_{t:w}(i,p)$ as

$$\beta_{t:w}(i,p) \quad \doteq \quad P(\mathbf{Y}_{t+1:w} = \mathbf{y}_{t+1:w}, M_w^+ = \square, S_{*:w} = \sigma_p \mid S_t = \sigma_i)\,. \tag{21}$$

Observe that if the chunk at position t remains open, with probability $P(M_t^+ = \oplus \mid S_t = \sigma_i) = \bar{\tau}_i$, then this represents part of the *same* chunk. Hence, the recurrence relation is

$$\beta_{t:w}(i,p) \quad = \quad \bar{\tau}_i\,a_{ij}\,b_{j,y_{t+1}}\,\beta_{t+1:w}(j,p)\,. \tag{22}$$

The edge case occurs for $t = w$, at which point the chunk at position t must become closed with probability $P(M_t^+ = \square \mid S_t = \sigma_i) = \tau_i$, and the process will transition to intermediate state $s_{*:t} = \sigma_p \in \mathcal{S}_{\texttt{int}}$ with probability

$$P(S_{*:t} = \sigma_p \mid S_t = \sigma_i, M_t^+ = \square) \quad = \quad u_{ip}\,, \tag{23}$$

as specified by the *leaf-to-intermediate* state transition matrix $\mathbf{U} = [u_{ip}]$. Hence, we obtain

$$\beta_{t:t}(i,p) \quad \doteq \quad P(M_t^+ = \square, S_{*:t} = \sigma_p \mid S_t = \sigma_i) \quad = \quad \tau_i\,u_{ip}\,. \tag{24}$$

Finally, note that if we do not know or do not care about the final intermediate state σ_p, then we may marginalise over it to obtain

$$\beta_{t:w}(i,\square) \quad \doteq \quad P(\mathbf{Y}_{t+1:w} = \mathbf{y}_{t+1:w}, M_w^+ = \square \mid S_t = \sigma_i)\,. \tag{25}$$

Also note that at the end of a complete sequence $\mathbf{y}_{1:T}$ we may use $P(M_T^+ = \triangleright \mid S_T = \sigma_i) \doteq \tau_i^{\triangleright}$ in place of $P(M_T^+ = \square \mid S_T = \sigma_i) \doteq \tau^{\square}$, whereupon

$$\beta_{t:T}(i,p) \quad \doteq \quad P(\mathbf{Y}_{t+1:T} = \mathbf{y}_{t+1:T}, M_T^+ = \triangleright, S_{*:T} = \sigma_p \mid S_t = \sigma_i)\,. \tag{26}$$

Thus, if we again ignore intermediate state σ_p, then we have

$$\beta_{t:T}(i,\triangleright) \quad \doteq \quad P(\mathbf{Y}_{t+1:T} = \mathbf{y}_{t+1:T}, M_T^+ = \triangleright \mid S_t = \sigma_i)\,. \tag{27}$$

1.2.5 Chunks and multi-chunks

We now consider a single, closed chunk $\mathbf{s}_{r:t} = (s_r, \ldots, s_t)$. For $r > 1$ the chunk must have started from a previous closed chunk with some intermediate state, say $s_{*:r-1} = \sigma_q \in \mathcal{S}_{\texttt{int}}$. Likewise, the chunk ends at position t and must have transitioned from leaf state to another intermediate state, say $s_{r:t} = \sigma_p \in \mathcal{S}_{\texttt{int}}$. In general, the complete chunk may be split into a *start-chunk* subsequence and an *end-chunk* subsequence. The split may occur at any position $r \leq w \leq t$, with arbitrary leaf state $s_w = \sigma_i \in \mathcal{S}_{\texttt{leaf}}$. Hence, the probability of the complete chunk is

$$\gamma_{r:t}(q,p) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t^+ = \square, S_{r:t} = \sigma_p \mid M_{r-1}^+ = \square, S_{*:r-1} = \sigma_q) \tag{28}$$

$$= \sum_{w=r}^{t} \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha_{r:w}(q,i)\, \beta_{w:t}(i,p)\,. \tag{29}$$

Note from forward recursion that we also have a variant for the start of a sequence, which corresponds to computing $\gamma_{1:t}(\triangleleft, p)$ from $\alpha_{1:w}(\triangleleft, p)$, and also a variant for the start of an arbitrary chunk (without further context), which corresponds to computing $\gamma_{r:t}(\square, p)$ from $\alpha_{r:w}(\square, p)$. Similarly, from backward recursion, we have variants for marginalising over the final intermediate state σ_p, both for the end of a sequence with $\gamma_{r:T}(q, \triangleright)$ computed via $\beta_{w:T}(q, \triangleright)$, and the end of a chunk with $\gamma_{r:T}(q, \square)$ computed from $\beta_{w:T}(q, \square)$. Thus, in general, the probability of observing a subsequence $\mathbf{y}_{r:t}$ of tokens as a single, complete chunk is given by

$$P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t^+ = \square \mid M_r^- = \square) \;=\; \gamma_{r:t}(\square, \square)\,. \tag{30}$$

In practice, these definitions are interesting but not very useful. More precisely, the formulae derived so far could possibly (with more theory) be used for partitioning a sequence of tokens into chunks, but they are not helpful from the point of view of estimating the grammar $\mathcal{G}$, nor from the point of view of analysing an entire sequence. Let us therefore turn from consideration of a single leaf chunk to consideration of a subsequence of one or more contiguous leaf chunks, henceforth called a *multi-chunk*. If the multi-chunk contains more than one chunk, then all chunks bar the last one must be closed on the right, and all chunks bar the first one must be closed on the left. We require more context before we can know if the multi-chunk is itself closed on the left and/or the right.

There are various ways we could define a multi-chunk. One way is to extend our previous definition of a single chunk. For example, we could combine two chunks, say $\gamma_{r:s}(q,v)$ and $\gamma_{s+1:t}(v,p)$. More generally, for a closed multi-chunk comprised of an arbitrary number of adjacent, closed chunks, we have

$$\bar{\gamma}_{r:t}(q,p) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t^+ = \square, S_{r:t} = \sigma_p \mid M_{r-1}^+ = \square, S_{*:r-1} = \sigma_q)\,. \tag{31}$$

Unfortunately, our notation is ambiguous, because this definition takes exactly the same form as that for a single chunk $\gamma_{r:t}(q,p)$. Rather than modify the notation, we shall simply rely on the context as to whether we are dealing with a single chunk or a multi-chunk. Now, for $r = t$ the multi-chunk reduces to a single chunk, with probability

$$\bar{\gamma}_{t:t}(q,p) \doteq P(Y_t = y_t, M_t^+ = \square, S_{t:t} = \sigma_p \mid M_{t-1}^+ = \square, S_{*:t-1} = \sigma_q) \;=\; \gamma_{t:t}(q,p)\,. \tag{32}$$

For $r < t$, the recurrence relation is given by

$$\bar{\gamma}_{r:t}(q,p) \;=\; \gamma_{r:t}(q,p) + \sum_{s=r}^{t-1} \sum_{\sigma_v \in \mathcal{S}_{\texttt{int}}} \bar{\gamma}_{r:s}(q,v)\, \gamma_{s+1:t}(v,p)\,. \tag{33}$$

Note that this definition essentially determines all of the ways that a multi-chunk may be partitioned, which in practice might not be very efficient. Also note that since we are marginalising over all internal structure of the multi-chunk, we need only consider combining a multi-chunk with a single chunk (on either the left or the right), otherwise the summation over the combination of two multi-chunks will count internal chunks multiple times.

As an alternative formulation, let us now simplify matters by ignoring the initial dependence on the previous intermediate state, which implies that we now have no prior context at the start of the multi-chunk. Similarly, let us ignore the final intermediate state, and consider instead only the final leaf state. Thus, consider an open multi-chunk $\mathbf{s}_{r:t} = (s_r, \ldots, s_t]$ that is closed on the left and open (by default) on the right. We may model this situation via

$$\alpha^{\texttt{multi}}_{r:t}(\square, i) \;\doteq\; P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, S_t = \sigma_i \mid M_r^- = \square)\,. \tag{34}$$

Note that this takes the same form as the single-chunk forward probability $\alpha_{r:t}(\square, i)$. The difference is that for a single chunk we implicitly assume there are no intra-chunk closures, whereas now for a multi-chunk we potentially have internal closures representing closed chunks. Clearly our notation is somewhat ambiguous.

Now, given this multi-chunk, a decision is made to either close the last chunk in the multi-chunk with probability τ_i, or leave it open with probability $\bar{\tau}_i$. If it is closed, then there must be a transition to some intermediate state. This situation is modelled via

$$\alpha^{\texttt{int}}_{r:t}(\square, p) \;\doteq\; P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t^+ = \square, S_{*:t} = \sigma_p \mid M_r^- = \square) \;=\; \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{r:t}(\square, i)\, \tau_i\, u_{ip}\,. \tag{35}$$

Initially, for $r = t$ the multi-chunk contains only a single chunk, and the (open) multi-chunk probability reduces to

$$\alpha^{\texttt{multi}}_{t:t}(\square, i) \;\doteq\; P(Y_t = y_t, S_t = \sigma_i \mid M_t^- = \square) \;=\; \iota_i^{\square}\, \breve{b}_{it}\,. \tag{36}$$

Alternatively, at the start of the sequence we may instead use

$$\alpha^{\texttt{multi}}_{1:1}(\triangleleft, i) \;\doteq\; P(Y_1 = y_1, S_1 = \sigma_i \mid M_1^- = \triangleleft) \;=\; \iota^{\triangleleft}\, \breve{b}_{i1}\,. \tag{37}$$

For $r < t$, the multi-chunk may contain one or more single chunks. In general, either the last chunk in the multi-chunk started at position t with the closure of a previous chunk at position $t-1$, or else the last chunk also remained open (on the left) at position $t-1$. For the former case, the leaf state $s_t = \sigma_i$ must be the result of an intermediate-to-leaf state transition, and for the latter case it results from a leaf-to-leaf state transition. Hence, the recurrence relation is

$$\alpha^{\texttt{multi}}_{r:t}(\square, i) \;=\; \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{r:t-1}(\square, j)\, \bar{\tau}_j\, a_{ji}\, \breve{b}_{it} + \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \alpha^{\texttt{int}}_{r:t-1}(\square, p)\, d_{pi}\, \breve{b}_{it}\,. \tag{38}$$

Despite differences in notation, this matches the relation given by (Kupiec), even though our model here for $\alpha^{\texttt{int}}$ completely differs from the model of (Kupiec).

For our purposes, we may now dispense with $\alpha^{\texttt{int}}$ by observing that

$$\alpha^{\texttt{multi}}_{r:t}(\square, i) = \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{r:t-1}(\square, j)\, \bar{\tau}_j\, a_{ji}\, \breve{b}_{it} + \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{r:t-1}(\square, j)\, \tau_j\, u_{jp}\, d_{pi}\, \breve{b}_{it} \tag{39}$$

$$= \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{r:t-1}(\square, j) \left\{ \bar{\tau}_j\, a_{ji} + \tau_j \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} u_{jp}\, d_{pi} \right\} \breve{b}_{it} \tag{40}$$

$$= \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{r:t-1}(\square, j)\, \tilde{a}_{ij}\, \breve{b}_{it}\,, \tag{41}$$

where now $\tilde{a}_{ij}$ represents either a direct leaf-to-leaf state transition or an indirect combination of a leaf-to-intermediate state transition and an intermediate-to-leaf state transition. In other words, $\tilde{a}_{ij}$ operates both within a chunk and across chunk boundaries. In matrix form, this corresponds to

$$\tilde{\mathbf{A}} \doteq \texttt{diag}(\mathbf{1} - \boldsymbol{\tau})\, \mathbf{A} + \texttt{diag}(\boldsymbol{\tau})\, \mathbf{U}\mathbf{D}\,, \tag{42}$$

which may be pre-computed, making the forward recursion efficient to compute.

Next, consider a multi-chunk $\mathbf{s}_{r:t} = [s_r, \ldots, s_t)$ that is open on the left and closed on the right. Analogously to the closure $\beta_{r:t}(i, \square)$ of a single chunk, we define the closure $\beta^{\texttt{multi}}_{r:t}(i, \square)$ of a multi-chunk via

$$\beta^{\texttt{multi}}_{r:t}(i, \square) \doteq P(\mathbf{Y}_{r+1:t} = \mathbf{y}_{r+1:t}, M_t^+ = \square \mid S_r = \sigma_i)\,. \tag{43}$$

Note that since the last chunk in the multi-chunk must be closed at position t, we have

$$\beta^{\texttt{multi}}_{t:t}(i, \square) \doteq P(M_t^+ = \square \mid S_t = \sigma_i) = \tau_i^{\square}\,. \tag{44}$$

Alternatively, at the end of a complete sequence $\mathbf{y}_{1:T} = \langle y_1, \ldots, y_T \rangle$, we could instead use

$$\beta^{\texttt{multi}}_{T:T}(i, \triangleright) \doteq P(M_T^+ = \triangleright \mid S_T = \sigma_i) = \tau_i^{\triangleright}\,. \tag{45}$$

In general, for $r < t$ we suppose that either the last chunk in the multi-chunk was started at position $r+1$ after the previous chunk was closed at position r, or else the last chunk extends back to include position r. Hence, we obtain the recurrence relation

$$\beta^{\texttt{multi}}_{r:t}(i, \square) = \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \tilde{a}_{ij}\, b_{j, y_{r+1}}\, \beta^{\texttt{multi}}_{r+1:t}(j, \square)\,. \tag{46}$$

Finally, we note that occasionally (e.g. for grammar estimation) we might need to complete a multi-chunk from an intermediate state $\sigma_p \in \mathcal{S}_{\texttt{int}}$ rather than from a leaf state $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$. When the context (i.e. leaf level versus intermediate level) is clear then we additionally define

$$\beta^{\texttt{int}}_{r:t}(p, \square) \doteq P(\mathbf{Y}_{r+1:t} = \mathbf{y}_{r+1:t}, M_t^+ = \square \mid M_r^+ = \square, S_{*:r} = \sigma_p) \tag{47}$$

$$= \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} d_{pi}\, b_{i, y_{r+1}}\, \beta^{\texttt{multi}}_{r+1:T}(i, \square) \tag{48}$$

for $r < t$, and

$$\beta^{\texttt{int}}_{t:t}(p, \square) \doteq P(M_t^+ = \square \mid M_t^+ = \square, S_{*:t} = \sigma_p) = 1\,, \tag{49}$$

for $r = t$. Once again, for the end of a sequence we may instead use

$$\beta^{\texttt{int}}_{r:T}(p, \triangleright) \doteq P(\mathbf{Y}_{r+1:T} = \mathbf{y}_{r+1:T}, M_T^+ = \triangleright \mid M_r^+ = \square, S_{*:r} = \sigma_p) \tag{50}$$

$$= \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} d_{pi}\, b_{i,y_{r+1}}\, \beta^{\texttt{multi}}_{r+1:T}(i, \triangleright) \tag{51}$$

for $r < t$, and

$$\beta^{\texttt{int}}_{T:T}(p, \triangleright) \doteq P(M_T^+ = \triangleright \mid M_T^+ = \triangleright, S_{*:T} = \sigma_p) = 1\,. \tag{52}$$

1.2.6 Sequence analysis

We now have enough information for inference. In particular, the likelihood of a complete sequence $\mathbf{y}_{1:T} = \langle y_1, \ldots, y_T \rangle$ is

$$P(\mathbf{y}_{1:T}) \doteq P(M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_T^+ = \triangleright) \tag{53}$$

$$= P(M_1^- = \triangleleft) \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{1:t}(\triangleleft, i)\, \beta^{\texttt{leaf}}_{t:T}(i, \triangleright)\,, \tag{54}$$

for $t = 1, 2, \ldots, T$. We typically assume that $P(M_1^- = \triangleleft) = 1$, i.e. that both the sequence and the first chunk must start at position 1.

For subsequences of tokens, the situation is more complex, since we have to allow for chunk boundaries. Of particular interest for sequence prediction is the incomplete subsequence $\mathbf{y}_{1:t} = \langle y_1, \ldots, y_t]$ for $t < T$. If token y_t has some leaf state $s_t = \sigma_j \in \mathcal{S}_{\texttt{leaf}}$, then either position t is the last position in its chunk with probability $\tau_j^{\square}$, or else the chunk remains open with probability $\bar{\tau}_j^{\square}$. Since $\tau_j^{\square} + \bar{\tau}_j^{\square} = 1$, the probability of the subsequence is

$$P(\mathbf{y}_{1:t}) \doteq P(M_1^- = \triangleleft, \mathbf{Y}_{1:t} = \mathbf{y}_{1:t}) \tag{55}$$

$$= P(M_1^- = \triangleleft) \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{1:t}(\triangleleft, i)\,. \tag{56}$$

One-step prediction for $t + 1 < T$ is then obtained via

$$P(y_{t+1} \mid \mathbf{y}_{1:t}) \doteq \frac{P(\mathbf{y}_{1:t+1})}{P(\mathbf{y}_{1:t})} = \frac{\sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{1:t+1}(\triangleleft, i)}{\sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{1:t}(\triangleleft, i)}\,. \tag{57}$$

The remainder of the complete sequence is also predicted as

$$P(\mathbf{y}_{t+1:T} \mid \mathbf{y}_{1:t}) \doteq P(\mathbf{Y}_{t+1:T} = \mathbf{y}_{t+1:T}, M_T^+ = \triangleright \mid \mathbf{Y}_{1:t} = \mathbf{y}_{1:t}, M_1^- = \triangleleft) \tag{58}$$

$$= \frac{P(\mathbf{y}_{1:T})}{P(\mathbf{y}_{1:t})} = \frac{\sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{1:t}(\triangleleft, i)\, \beta^{\texttt{multi}}_{t:T}(i, \triangleright)}{\sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha^{\texttt{multi}}_{1:t}(\triangleleft, i)}\,. \tag{59}$$

1.2.7 Grammar estimation

From forward recursion and backward recursion, we see that the stochastic nature of the chunking grammar $\mathcal{G}$ is determined by a number of conditional probability tables (CPTs). Like a standard HMM, we require the *leaf-to-leaf* state transition matrix $\mathbf{A} = [a_{ij}]$, the token *emission* matrix $\mathbf{B} = [b_{im}]$, and the probability vector $\boldsymbol{\iota}^{\triangleleft} = [\iota_i^{\triangleleft}]$ of *initial* states. Also, for complete sequences, we require the vector $\boldsymbol{\tau}^{\triangleright} = [\tau_i^{\triangleright}]$ of sequence *termination* probabilities. Finally, for chunking we

require the probability vector $\boldsymbol{\iota}^{\square} = [\iota_i^{\square}]$ of *initial* chunk states and the vector $\boldsymbol{\tau}^{\square} = [\tau_i^{\square}]$ of chunk *termination* probabilities, as well as the *leaf-to-intermediate* state transition matrix $\mathbf{U} = [u_{ip}]$ and the *intermediate-to-leaf* state transition matrix $\mathbf{D} = [d_{pi}]$.

In analogy to the estimation process for a HMM, we assume that an expectation-maximisation (EM) formulation leads to a maximum likelihood (ML) estimate, by which the various probability vectors and matrices are simply normalised forms of vectors and matrices of various joint counts of interest. EM is an iterative process that starts with prior estimates, e.g. $\mathbf{A}'$, $\mathbf{B}'$, etc., and produces posterior re-estimates, e.g. $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, et cetera. For notational convenience, we henceforth drop the prime from our prior estimates.

In order to estimate $\mathbf{A}$, we need to compute the number $N_{ij}^{\mathtt{leaf}}$ of times that leaf state $\sigma_i \in \mathcal{S}_{\mathtt{leaf}}$ has been immediately followed by leaf state $\sigma_j \in \mathcal{S}_{\mathtt{leaf}}$ within an open chunk. This is a stochastic value, and so we estimate the expected value $\hat{N}_{ij}$ given the observed sequence $\mathbf{y}_{1:T}$. The require computation is

$$\begin{aligned}
\hat{N}_{ij}^{\mathtt{leaf}} &\doteq \mathbb{E}[N_{ij}^{\mathtt{leaf}} \mid \mathbf{y}_{1:T}] && (60)\\
&= \sum_{t=1}^{T-1} P(S_t = \sigma_i, S_{t+1} = \sigma_j, M_t^+ = \oplus \mid M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_T^+ = \triangleright) && (61)\\
&= \frac{P(M_1^- = \triangleleft)}{P(\mathbf{y}_{1:T})} \sum_{t=1}^{T-1} \alpha_{1:t}^{\mathtt{multi}}(\triangleleft, i)\, \bar{\tau}_i\, a_{ij}\, b_{j,y_{t+1}}\, \beta_{t+1:T}^{\mathtt{multi}}(j, \triangleright)\,. && (62)
\end{aligned}$$

Hence, the latest estimate of the *leaf-to-leaf* state (or *within-chunk*) transition probability matrix $\mathbf{A} = [a_{ij}]$ is obtained via $\hat{a}_{ij} \doteq \frac{\hat{N}_{ij}^{\mathtt{leaf}}}{\hat{N}_{i\cdot}^{\mathtt{leaf}}}$.

Similarly, let $N_{ip}^{\mathtt{up}}$ be the number of times that any chunk closes with leaf state $\sigma_i \in \mathcal{S}_{\mathtt{leaf}}$ and intermediate state $\sigma_p \in \mathcal{S}_{\mathtt{int}}$. Then

$$\begin{aligned}
\hat{N}_{ip}^{\mathtt{up}} &\doteq \mathbb{E}[N_{ip}^{\mathtt{up}} \mid \mathbf{y}_{1:T}] && (63)\\
&= \sum_{t=1}^{T-1} P(S_t = \sigma_i, M_t^+ = \square, S_{*:t} = \sigma_p \mid M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_T^+ = \triangleright) && (64)\\
&= \frac{P(M_1^- = \triangleleft)}{P(\mathbf{y}_{1:T})} \left\{ \sum_{t=1}^{T-1} \alpha_{1:t}^{\mathtt{multi}}(\triangleleft, i)\, \tau_i^{\square}\, u_{ip}\, \beta_{t+1:T}^{\mathtt{int}}(p, \triangleright) + \alpha_{1:T}^{\mathtt{multi}}(\triangleleft, i)\, \tau_i^{\triangleright}\, u_{ip} \right\}, && (65)
\end{aligned}$$

and the *leaf-to-intermediate* state (or *end-chunk*) transition probability matrix $\mathbf{U} = [u_{ip}]$ is re-estimated via $\hat{u}_{ip} \doteq \frac{\hat{N}_{ip}^{\mathtt{up}}}{\hat{N}_{i\cdot}^{\mathtt{up}}}$.

Additionally, let $N_{pi}^{\mathtt{down}}$ be the number of times that a chunk closes with intermediate state $\sigma_p \in \mathcal{S}_{\mathtt{int}}$ and the next chunk opens with leaf state $\sigma_i \in \mathcal{S}_{\mathtt{leaf}}$. Then

$$\begin{aligned}
\hat{N}_{pi}^{\mathtt{down}} &\doteq \mathbb{E}[N_{pi}^{\mathtt{down}} \mid \mathbf{y}_{1:T}] && (66)\\
&= \sum_{t=1}^{T-1} P(M_t^+ = \square, S_{*:t} = \sigma_p, S_{t+1} = \sigma_i \mid M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_T^+ = \triangleright) && (67)\\
&= \frac{P(M_1^- = \triangleleft)}{P(\mathbf{y}_{1:T})} \sum_{t=1}^{T-1} \alpha_{1:t}^{\mathtt{int}}(\triangleleft, p)\, d_{pi}\, \breve{b}_{i,t+1}\, \beta_{t+1:T}^{\mathtt{multi}}(i, \triangleright)\,, && (68)
\end{aligned}$$

and the *intermediate-to-leaf* state (or *start-chunk*) transition probability matrix $\mathbf{D} = [d_{pi}]$ is re-estimated via $\hat{d}_{pi} \doteq \frac{\hat{N}_{pi}^{\texttt{down}}}{\hat{N}_{p\cdot}^{\texttt{down}}}$.

Finally, we want to re-estimate the *initial* state probability vectors, namely $\boldsymbol{\iota}^{\square} = [\iota_i^{\square}]$ for the start of chunks, and $\boldsymbol{\iota}^{\triangleleft} = [\iota_i^{\triangleleft}]$ for the start of sequences. Likewise, we want to re-estimate the state *termination* probabilities, namely $\boldsymbol{\tau}^{\square} = [\tau_i^{\square}]$ for the end of chunks, and $\boldsymbol{\tau}^{\triangleright} = [\tau_i^{\triangleright}]$ for the end of sequences.

We consider the start and end of a sequence first. The *initial* sequence probabilities are given by

$$\hat{\iota}_i^{\triangleleft} \doteq P(S_1 = \sigma_i \mid M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_T^+ = \triangleright) = \frac{P(M_1^- = \triangleleft)}{P(\mathbf{y}_{1:T})} \iota_i^{\triangleleft} b_{i,y_1} \beta_{1:T}^{\texttt{multi}}(i, \triangleright)\,. \quad (69)$$

However, the *terminal* sequence probabilities are more difficult. We note that the posterior we want cannot be computed directly due to the Markov nature of the model, since

$$P(M_T^+ = \triangleright \mid S_T = \sigma_i, M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}) = \frac{P(M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, S_T = \sigma_i, M_T^+ = \triangleright)}{P(M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, S_T = \sigma_i)} \quad (70)$$

$$= \frac{P(M_1^- = \triangleleft)\, \alpha_{1:T}^{\texttt{multi}}(\triangleleft, i)\, \tau_i^{\triangleright}}{P(M_1^- = \triangleleft)\, \alpha_{1:T}^{\texttt{multi}}(\triangleleft, i)} = \tau_i^{\triangleright}\,. \quad (71)$$

Instead, we note that only the last leaf state s_T in a complete sequence contributes to sequence termination, and all previous the leaf states s_t for $t < T$ contribute to non-termination. Hence, we define the per-token leaf state posterior

$$\hat{N}_{it}^{\texttt{token}} \doteq P(S_t = \sigma_i \mid M_1^- = \triangleleft, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T} M_T^+ = \triangleright) \quad (72)$$

$$= \frac{P(M_1^- = \triangleleft)\, \alpha_{1:t}^{\texttt{multi}}(\triangleleft, i)\, \beta_{t:T}^{\texttt{multi}}(i, \triangleright)}{P(\mathbf{y}_{1:T})}\,, \quad (73)$$

and re-estimate $\hat{\tau}_i^{\triangleright} \doteq \frac{\hat{N}_{iT}^{\texttt{token}}}{\hat{N}_{i\cdot}^{\texttt{token}}}$.

Lastly, we want re-estimate the *start-chunk* (initial) probability vector $\boldsymbol{\iota}^{\square} = [\iota_i^{\square}]$, and the *end-chunk* (terminal) probability vector $\boldsymbol{\tau}^{\square} = [\tau_i^{\square}]$. Note that the end of every chunk is followed by a *leaf-to-intermediate* state transition, which we have counted via $\hat{N}_{ip}^{\texttt{up}}$. Additionally, within every chunk we have counted the *leaf-to-leaf* state transitions, namely $\hat{N}_{ij}^{\texttt{leaf}}$, which do not terminate the chunk. Hence, the *terminal* chunk probabilities are re-estimated as $\hat{\tau}_i^{\square} \doteq \frac{\hat{N}_{i\cdot}^{\texttt{up}}}{\hat{N}_{i\cdot}^{\texttt{up}} + \hat{N}_{i\cdot}^{\texttt{leaf}}}$.

Similarly, $\hat{N}_{pi}^{\texttt{down}}$ counts the expected number of *intermediate-to-leaf* state transitions that start each chunk except the first chunk of a sequence. The initial states of the first chunk have already been estimated via $\hat{\iota}_i^{\triangleleft}$. Consequently, the *initial* chunk probabilities are re-estimated as $\hat{\iota}_i^{\square} \doteq \frac{\hat{\iota}_i^{\square} + \hat{N}_{\cdot i}^{\texttt{down}}}{\hat{\iota}_{\cdot}^{\square} + \hat{N}_{\cdot\cdot}^{\texttt{down}}}$.

we want re-estimate the *start-chunk* (initial) probability vector $\boldsymbol{\iota} = [\iota_i]$, and the *end-chunk* (terminal) probability vector $\boldsymbol{\tau} = [\tau_i]$. For the former quantity, recall that $\hat{N}_{pi}^{\texttt{down}}$ counts the joint occurrences of leaf state $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$ at the start of a chunk and intermediate state $\sigma_p \in \mathcal{S}_{\texttt{int}}$ at the end of the previous chunk. Hence, it follows that $\hat{\iota}_i \doteq \frac{\hat{N}_{\cdot i}^{\texttt{down}}}{\hat{N}_{\cdot\cdot}^{\texttt{down}}}$. Similarly, for the latter quantity, recall that $\hat{N}_{ij}^{\texttt{leaf}}$ counts every non-terminating transition, and $\hat{N}_{ip}^{\texttt{up}}$ counts every terminating transition. Hence, $\hat{\tau}_i \doteq \frac{\hat{N}_{i\cdot}^{\texttt{up}}}{\hat{N}_{i\cdot}^{\texttt{leaf}} + \hat{N}_{i\cdot}^{\texttt{up}}}$.

1.3 Simplified Chunking

Having gone through the complicated details and assumptions of chunking in the previous section, let us now revisit the key ideas with the aim of simplifying the grammar $\mathcal{G}$ still further. As before, we consider a finite set $\mathcal{Y} = \{\nu_1, \nu_2, \ldots\}$ of terminal tokens, and a finite set $\mathcal{S} = \{\sigma_1, \sigma_2, \ldots\}$ of non-terminal states. We also retain the set $\mathcal{M} = \{\triangleleft, \oplus, \square, \triangleright\}$ of markers that denote the internal context of the process.

However, we now simplify the sequence generation process as follows. For each complete sequence, let the process always start in context $M_0 = \triangleleft$ with probability $P(M_0 = \triangleleft) = 1$. Let this context correspond to the chunking symbols $C_0 = \langle($, which means that the start of a sequence also opens the first chunk. Next, the process chooses some state $S_1 = s_1$, generates token $Y_1 = y_1$, and then transitions to context $M_1 = m_1$. Iteratively, the process has context $M_{t-1} = m_{t-1}$, chooses state $S_t = s_t$, generates token $Y_t = y_t$, and then transitions to context $M_t = m_t$. Finally, for a complete sequence of length $|\mathbf{y}| = T$, the process terminates with context $M_T = \triangleright$. This corresponds to the chunking symbols $C_T =)\rangle$, which means that the last chunk is closed at the end of a sequence.

In the interior of a sequence, for $t = 1, 2, \ldots, T-1$, the process has a choice of context, namely $M_t = \oplus$ or $M_t = \square$. The former context indicates that both the sequence and the current chunk will continue to the next token, with corresponding chunking symbols $C_t =][$. The latter context indicates that the current chunk will be closed and the sequence will continue to the next token in a new chunk, with corresponding chunking symbols $C_t =)($.

The chunking grammar $\mathcal{G}$ is expressed by a Markov process that generates an arbitrary-length (but non-empty) complete sequence $\mathbf{Y}_{1:T} = \langle Y_1, \ldots, Y_T \rangle$, driven by the dependencies $\overset{M_0}{\to} S_1 \overset{M_1}{\to} S_2 \overset{M_2}{\to} \cdots \overset{M_{T-1}}{\to} S_T \overset{M_T}{\to}$, with hidden states $\mathbf{S}_{1:T} = (S_1, \ldots, S_T)$ and hidden contexts $\mathbf{M}_{0:T} = (M_0, \ldots, M_T)$. The grammar is thus comprised of distinct types of rules. For token $\nu_m \in \mathcal{Y}$, context $\kappa \in \mathcal{M}$, and states $\sigma_i, \sigma_j \in \mathcal{S}$, the types of rules are:

1. *Context-to-state* transition rules of the form $\overset{\kappa}{\to} \sigma_i$ with probability $P(S_t = \sigma_i \mid M_{t-1} = \kappa) \doteq \iota_i^\kappa$.
2. *State-to-context* transition rules of the form $\sigma_i \overset{\kappa}{\to}$ with probability $P(M_t = \kappa \mid S_t = \sigma_i) \doteq \tau_i^\kappa$.
3. *Token* generation rules of the form $\sigma_i \to \nu_m$ with probability $P(Y_t = \nu_m \mid S_t = \sigma_i) \doteq b_{im}$.
4. *State-to-state* transition rules of the form $\sigma_i \overset{\kappa}{\to} \sigma_j$ with probability $P(S_{t+1} = \sigma_j, M_t = \kappa \mid S_t = \sigma_i) = P(M_t = \kappa \mid S_t = \sigma_i)\, P(S_{t+1} = \sigma_j \mid S_t = \sigma_i, M_t = \kappa)$, where $P(S_{t+1} = \sigma_j \mid S_t = \sigma_i, M_t = \kappa) \doteq a_{ij}^\kappa$.

Let $\mathcal{R}$ be the set of rules in the grammar $\mathcal{G}$, over all contexts $\kappa \in \mathcal{M}$, states $\sigma_i, \sigma_j \in \mathcal{S}$, and tokens $\nu_m \in \mathcal{Y}$. Why do we need so many rules? In practice, although the process always generates complete sequences, we might not observe the entire sequence $\langle y_1, \ldots, y_T \rangle = \langle Y_1, \ldots, Y_T \rangle$. Instead, we might have observed an incomplete sequence, e.g. $\langle y_1, \ldots, y_t] = \langle Y_1, \ldots, Y_t]$ or $[y_1, \ldots, y_t\rangle = [Y_{T-t+1}, \ldots, Y_T\rangle$, or even a subsequence, e.g. $[y_1, \ldots, y_t] = [Y_r, \ldots, Y_{t+r-1}]$. Hence, we re-index the stochastic process above to locally match the observed sequence, rather than the true (but unknown) process. Consequently, we now have a choice of starting context M_0, internal context M_t, and ending context M_T, depending upon what we know of the observation process.

However, note that some contexts do not make sense. Thus, at the start of an observed sequence we set $P(M_0 = \triangleright) = 0$, since the current sequence will not have been observed if the process terminated beforehand. Similarly, the process cannot restart during a sequence, such that $P(M_t = \triangleleft \mid S_t) = 0$.

Additionally, only the contexts $M_t \in \mathcal{M}^+ \doteq \{\oplus, \square\}$ designate the continuation of a sequence, and thus we set $P(S_{t+1} \mid S_t, M_t) = 0$ for $M_t \in \mathcal{M}^- \doteq \{\triangleleft, \triangleright\}$.

The joint model of the local process is therefore

$$\begin{aligned} P(\mathbf{s}, \mathbf{m}, \mathbf{y} \mid \mathcal{G}) &= P(M_0 = m_0)\, P(S_1 = s_1 \mid M_0 = m_0) && (74) \\ &\quad \times \prod_{t=1}^{T-1} \{P(Y_t = y_t \mid S_t = s_t)\, P(M_t = m_t \mid S_t = s_t)\, P(S_{t+1} = s_{t+1} \mid S_t = s_t, M_t = m_t)\} && (75) \\ &\quad \times P(M_T = m_T \mid S_T = s_T)\,, && (76) \end{aligned}$$

and the marginal probability of the sequence is

$$P(\mathbf{y} \mid \mathcal{G}) = \sum_{\mathbf{s} \in \mathcal{S}^T} \sum_{\mathbf{m} \in \mathcal{M}^{T+1}} P(\mathbf{s}, \mathbf{m}, \mathbf{y} \mid \mathcal{G})\,. \tag{77}$$

1.3.1 Forward-Backward algorithm

In the previous formulation, we had difficulty succinctly expressing the difference between the start of a sequence with initial states $\boldsymbol{\iota}^{\triangleleft}$, and the start of a chunk with initial states $\boldsymbol{\iota}^{\square}$. Similarly, there was confusion between the end of a sequence with terminal probabilities $\boldsymbol{\tau}^{\triangleright}$, and the end of a chunk with terminal probabilities $\boldsymbol{\tau}^{\square}$. For convenience, let us now define a new, polymorphic marker symbol '$\diamond$', which denotes $M_0 = \triangleleft$ at the start of a complete sequence $\mathbf{y}_{1:T}$, $M_T = \triangleright$ at the end of the sequence, and $M_t = \square$ internally within the sequence. This correspondence is deterministic, and depends only upon the starting and ending positions of any chosen subsequence.

As shown in a previous section, the process permits a HMM-like view of a sequence by making chunks implicit. Hence, for an open multi-chunk that starts at position r, the *forward* probabilities are given by the definition

$$\alpha_{r:t}(i) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, S_t = \sigma_i \mid M_{r-1} = \diamond)\,, \tag{78}$$

which for $r = t$ reduces to

$$\alpha_{t:t}(i) = P(Y_t = y_t, S_t = \sigma_i \mid M_{t-1} = \diamond) = \iota^{\diamond}\, \breve{b}_{it}\,, \tag{79}$$

and for $r < t$ gives the recurrence relation

$$\alpha_{r:t}(i) = \sum_{\sigma_j \in \mathcal{S}} \alpha_{r:t-1}(j)\, \tilde{a}_{ji}\, \breve{b}_{it}\,, \tag{80}$$

where

$$\tilde{a}_{ij} \doteq \sum_{m \in \mathcal{M}^+} \tau_i^m\, a_{ij}^m\,. \tag{81}$$

Likewise, for an open multi-chunk that ends at position t, the *backward* probabilities are given by the definition

$$\beta_{r:t}(i) \doteq P(\mathbf{Y}_{r+1:t} = \mathbf{y}_{r+1:t}, M_t = \diamond \mid S_r = \sigma_i)\,, \tag{82}$$

which for $r = t$ reduces to

$$\beta_{t:t}(i) = P(M_t = \diamond \mid S_r = \sigma_i) = \tau^{\diamond}\,, \tag{83}$$

and for $r < t$ gives the recurrence relation

$$\beta_{r:t}(i) \;=\; \sum_{\sigma_j \in \mathcal{S}} \tilde{a}_{ij}\, \breve{b}_{j,r+1}\, \beta_{r+1:t}(j)\,. \tag{84}$$

Note that if we neglect internal subsequences spanning $r:t$, i.e. we insist that either $r = 1$ or $t = T$, then this formulation reduces to the standard forward-backward algorithm. Hence, the probability of a complete sequence $\mathbf{y} = \langle y_1, \ldots, y_T \rangle$ is given by

$$P(\mathbf{y}) \;=\; \sum_{\sigma_i \in \mathcal{S}} \alpha_{1:t}(i)\, \beta_{t:T}(i)\,, \tag{85}$$

for any $t = 1, 2, \ldots, T$.

1.3.2 Inside-Outside algorithm

Whereas the forward-backward algorithm of the previous section deliberately obscures the start and end of chunks, here we want to explicitly handle chunks, or at least multi-chunks. Hence, if $\alpha_{r:t}(i)$ is the probability of an open multi-chunk starting at position r, then the corresponding probability of a closed multi-chunk is given by

$$\bar{\alpha}_{r:t} \;\doteq\; P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t = \diamond \mid M_{r-1} = \diamond) \;=\; \sum_{\sigma_i \in \mathcal{S}} \alpha_{r:t}(i)\, \tau_i^{\diamond}\,. \tag{86}$$

Recall that, by construction, an arbitrary chunk or multi-chunk has only a known start, and that by assumption the last state of the previous closed chunk is unknown. Hence, under our simplified model, once a multi-chunk is permanently closed, its the terminal state is of no further relevance to the adjacent multi-chunk.

Note that since $\bar{\alpha}_{r:t}$ spans tokens $\mathbf{y}_{r:t}$, these define *inner* probabilities, and their computation over all $1 \le r \le t \le T$ forms the *inside* pass. Consequently, the *outside* pass corresponds to computing the *outer* probabilities $\bar{\beta}_{r:t}$ that complete the rest of the sequence. We therefore define

$$\bar{\beta}_{r:t} \;\doteq\; P(M_0 = \triangleleft, \mathbf{Y}_{1:r-1} = \mathbf{y}_{1:r-1}, M_{r-1} = \diamond, \mathbf{Y}_{t+1:T} = \mathbf{y}_{t+1:T}, M_T = \triangleright \mid M_t = \diamond)\,. \tag{87}$$

Note that on the left for $r = 1$ this reduces to

$$\bar{\beta}_{1:t} \;\doteq\; P(M_0 = \triangleleft, \mathbf{Y}_{t+1:T} = \mathbf{y}_{t+1:T}, M_T = \triangleright \mid M_t = \diamond)\,, \tag{88}$$

and on the right for $t = T$ becomes

$$\bar{\beta}_{r:T} \;\doteq\; P(M_0 = \triangleleft, \mathbf{Y}_{1:r-1} = \mathbf{y}_{1:r-1}, M_{r-1} = \diamond \mid M_T = \triangleright)\,. \tag{89}$$

Now, if $r > 1$ then there is room on the left of the current multi-chunk to place an adjacent multi-chunk. Hence, for some position $s < r$, we adjoin the closed multi-chunk $\bar{\alpha}_{s:r-1}$, and what remains forms the outer probability $\bar{\beta}_{s:t}$. Likewise, if $t < T$ then there is room on the right of the current multi-chunk to place an adjacent multi-chunk. Hence, for some position $s > t$, we adjoin the closed multi-chunk $\bar{\alpha}_{t+1:s}$, and what remains forms the outer probability $\bar{\beta}_{r:s}$. Summing over all such adjacent multi-chunks gives rise to the recurrence relation

$$\bar{\beta}_{r:t} \;\doteq\; \sum_{s=1}^{r-1} \bar{\alpha}_{s:r-1}\, \bar{\beta}_{s:t} + \sum_{s=t+1}^{T} \bar{\alpha}_{t+1:s}\, \bar{\beta}_{t:s}\,. \tag{90}$$

Note that now $\bar{\alpha}_{r:t}\,\bar{\beta}_{r:t}$ gives the joint probability of the token sequence **and** the fact that a closed multi-chunk spans positions $r:t$, since

$$\begin{aligned}\bar{\alpha}_{r:t}\,\bar{\beta}_{r:t} &= P(\mathbf{Y}_{r:t}=\mathbf{y}_{r:t}, M_t=\diamond \mid M_{r-1}=\diamond)\,P(M_0=\triangleleft, \mathbf{Y}_{1:r-1}=\mathbf{y}_{1:r-1}, M_{r-1}=\diamond, \mathbf{Y}_{t+1:T}=\mathbf{y}_{t+1:T}, M_T=\triangleright \mid M\\ &= P(M_0=\triangleleft, \mathbf{Y}_{1:T}=\mathbf{y}_{1:T}, M_T=\triangleright, M_{r-1}=\diamond, M_t=\diamond)\,.\end{aligned}$$

The marginal probability of the complete sequence $\mathbf{y}_{1:T}$ is

$$P(\mathbf{y}_{1:T}) \doteq \bar{\alpha}_{1:T}\,\bar{\beta}_{1:T}\,, \tag{93}$$

due to the polymorphic nature of the marker '$\diamond$', and the fact that a complete sequence can always be partitioned into closed chunks. For incomplete sequences, one may replace'$\triangleleft$' and/or '$\triangleright$' by '$\square$', as necessary, in the definition of $\bar{\beta}_{r:t}$. However, this assumes that the incomplete sequence can also be chunked.

1.3.3 An alternative formulation

In terms of the process depicted earlier, note that each *complete-data* case $(\mathbf{s}, \mathbf{m}, \mathbf{y})$ corresponds to a structure T, where, in grammatical terms, T is a derivation or parse of the token sequence $\mathbf{y}$. Hence, we restrict our attention to the set $\mathcal{T} = \mathcal{T}(\mathbf{y})$ of all such parses that are consistent with $\mathbf{y}$.

Next, we note that the Markov process takes the form of a graph, which can be re-expressed as the Bayesian network shown below.

Thus, each parse $T \in \mathcal{T}$ has a structural interpretation as a network (or graph) $G(T)$ of nodes (or vertices), where each node $v \in G(T)$ has some designated context, state or token that conditionally depends on the current or previous context and/or state, denoted by $\boldsymbol{\pi}(v, T)$. Consequently, the conditional model has the form

$$P(T \mid \mathbf{y}) = \frac{P(\mathbf{s}, \mathbf{m}, \mathbf{y} \mid \mathcal{G})}{P(\mathbf{y} \mid \mathcal{G})} = \frac{1}{Z} \prod_{v \in G(T)} P(v \mid \boldsymbol{\pi}(v, T))\,, \tag{94}$$

where we have normalised the distribution via the partition function $Z = Z(\mathbf{y}) \doteq P(\mathbf{y} \mid \mathcal{G})$.

Now, following (Eisner), for every rule $R \in \mathcal{R}$ we define $\theta_R \doteq \ln P(R)$, such that these log-probabilities parameterise the grammar $\mathcal{G}$. Next, we introduce the feature function $f_R : \mathcal{T} \mapsto \mathbb{N}$ that counts the number of occurrences of rule R in parse T. Hence, the conditional model now becomes

$$P(T \mid \mathbf{y}) = \frac{1}{Z} \prod_{R \in \mathcal{R}} P(R)^{f_R(T)} = \frac{1}{Z} \exp\left\{ \sum_{R \in \mathcal{R}} f_R(T)\, \theta_R \right\}, \tag{95}$$

with normaliser

$$Z = \sum_{T \in \mathcal{T}} \exp\left\{ \sum_{R \in \mathcal{R}} f_R(T)\, \theta_R \right\}. \tag{96}$$

It follows that

$$\frac{\partial \ln Z}{\partial \theta_R} = \frac{1}{Z} \frac{\partial Z}{\partial \theta_R} = \frac{1}{Z} \sum_{T \in \mathcal{T}} f_R(T) \exp\left\{ \sum_{R' \in \mathcal{R}} f_{R'}(T)\, \theta_{R'} \right\} \tag{97}$$

$$= \sum_{T \in \mathcal{T}} f_R(T)\, P(T \mid \mathbf{y}) = \mathbb{E}_{\mathcal{T}}[f_R(T) \mid \mathbf{y}]\,. \tag{98}$$

The last term gives the expected count $\hat{N}_R$ of the number of times rule R can appear across all possible parses of sequence $\mathbf{y}$.

Given this relation, (Eisner) goes on to show how automatic differentiation of $\ln Z$ provides the update equations for computing θ_R. This is demonstrated by applying back-propagation to the inside algorithm to efficiently obtain the both the outside algorithm and rule count estimation. In particular, we have

$$\hat{N}_R \doteq \frac{\partial \ln Z}{\partial \theta_R} = \frac{\partial \ln Z}{\partial Z} \frac{\partial Z}{\partial P(R)} \frac{\partial P(R)}{\partial \theta_R} = \frac{P(R)}{Z} \frac{\partial Z}{\partial P(R)} . \tag{99}$$

Now, recall that the probability $P(R)$ of rule R is assumed to be invariant to the substructure in which it occurs. Consequently, the back-propagation gradient $\frac{\partial Z}{\partial P(R)}$ represents the marginalisation over all possible substructures in which rule R can occur.

1.4 Hierarchical Chunking

Now that we have looked at leaf-level chunking in detail, it is time to revisit hierarchical chunking. This means we need once again to distinguish between leaf states $\mathcal{S}_{\texttt{leaf}}$ and intermediate states $\mathcal{S}_{\texttt{int}}$. Essentially, each leaf-level chunk will be assigned an intermediate state, and the chunks will be combined by higher-level rules, according to the grammar $\mathcal{G}$.

Previously, the intermediate state was assigned at the end of a chunk. However, this meant that the initial states of arbitrarily-placed chunks (as opposed to adjacent chunks) all shared the same fixed distribution. A viable alternative, therefore, is to assign the intermediate state at the start of a chunk and to condition the initial leaf state of each chunk on the chunk's intermediate state. As a consequence, we no longer need to distinguish between the start of the sequence and the start of a chunk at the leaf level, since this distinction will be handled by the higher-level grammar. This process is demonstrated by the example shown in the figure below.

In abbreviated form, the probability of this derivation is

$$P(\triangleleft\texttt{S}\triangleright)\, P(\square\texttt{NP}, \square\texttt{VP} \mid \texttt{S}) \tag{100}$$

$$P(\texttt{D} \mid \square\texttt{NP})\, P(\textit{The} \mid \texttt{D})\, P(\oplus\texttt{J} \mid \texttt{D})\; P(\textit{black} \mid \texttt{J})\, P(\oplus\texttt{N} \mid \texttt{J})\; P(\textit{cat} \mid \texttt{N})\, P(\square \mid \texttt{N}) \tag{101}$$

$$P(\texttt{V} \mid \square\texttt{VP})\, P(\textit{purred} \mid \texttt{V})\, P(\square \mid \texttt{V})\ . \tag{102}$$

1.4.1 Forward pass

We now consider an open chunk that starts at position r with intermediate state $S_{r:*} = \sigma_p \in \mathcal{S}_{\texttt{int}}$, and 'ends' at position t with leaf state $S_t = \sigma_i \in \mathcal{S}_{\texttt{leaf}}$. The probability of this chunk is

$$\alpha_{r:t}(p, i) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, S_t = \sigma_i \mid M_{r-1} = \square, S_{r:*} = \sigma_p) . \tag{103}$$

The initial leaf state of the chunk is determined via

$$\alpha_{t:t}(p, i) \doteq P(Y_t = y_t, S_t = \sigma_i \mid M_{t-1} = \square, S_{t:*} = \sigma_p) = d_{pi}\, \breve{b}_{it} , \tag{104}$$

where now

$$d_{pi} \doteq P(S_t = \sigma_i \mid M_{t-1} = \square, S_{t:*} = \sigma_p) . \tag{105}$$

The open chunk is then continued at the leaf level via the recurrence relation

$$\alpha_{r:t+1}(p,i) \doteq \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha_{r:t}(p,j)\, \bar{\tau}_j\, c_{ji}\, \breve{b}_{i,t+1}\,, \tag{106}$$

where once again $\tau_i \doteq \tau_i^{\square}$ and $\bar{\tau}_i = 1 - \tau_i \doteq \tau_i^{\oplus}$ with position-invariant probability

$$P(M_t = \square \mid S_t = \sigma_i) \doteq \tau_i\,, \tag{107}$$

and now $c_{ij} \doteq a_{ij}^{\oplus}$ with position-invariant probability

$$P(S_{t+1} = \sigma_j \mid S_t = \sigma_i, M_t = \oplus) \doteq c_{ij}\,. \tag{108}$$

Note that all $\alpha_{r:t}(p,i)$ comprise the *forward* probabilities of the *foward* pass.

Eventually, every open chunk must be closed. The probability of a closed chunk is simply

$$\gamma_{r:t}(p) \doteq P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t = \square \mid M_{r-1} = \square, S_{r:t} = \sigma_p) = \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha_{r:t}(p,i)\, \tau_i\,. \tag{109}$$

Note that after closure of the chunk, the sequence generation process notionally returns to the (end of the) chunk's intermediate state. Hence, we no longer need to know the terminal leaf state of the chunk.

1.4.2 Backward pass

The converse of the forward pass is a *backward* pass. Whereas the forward pass extends an open chunk on the right with some leaf state, say $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$, the backward pass extends an open chunk on the left given the previous state σ_i. Consequently, we define

$$\beta_{r:t}(p,i) \doteq P(\mathbf{Y}_{r+1:t} = \mathbf{y}_{r+1:t}, M_t = \square \mid S_r = \sigma_i, S_{*:t} = \sigma_p)\,, \tag{110}$$

with

$$\beta_{t:t}(p,i) = P(M_t = \square \mid S_t = \sigma_i) = \tau_i\,. \tag{111}$$

For $r < t$, the recurrence relation is

$$\beta_{r:t}(p,i) \doteq \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \bar{\tau}_i\, c_{ij}\, \breve{b}_{j,r+1}\, \beta_{r+1:t}(p,j)\,. \tag{112}$$

Eventually, this open chunk will become closed on the left at some position $s \le r$ with probability $\alpha_{s:r}(p,i)$, and hence the general probability of a closed chunk is

$$\gamma_{s:t}(p) \doteq P(\mathbf{Y}_{s:t} = \mathbf{y}_{s:t}, M_t = \square \mid M_{s-1} = \square, S_{s:t} = \sigma_p) \tag{113}$$

$$= \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \alpha_{s:r}(p,i)\, \beta_{r:t}(p,i)\,, \tag{114}$$

for every $s \le r \le t$.

1.4.3 Inside pass

We recall that the (closed) chunk with state $S_{r:s} = \sigma_p \in \mathcal{S}_{\texttt{int}}$ that spans tokens $\mathbf{y}_{r:s}$ has *inner* probability $\gamma_{r:s}(p)$. Furthermore, if that chunk is followed by an adjacent chunk, say $\gamma_{s+1:t}(q)$ with intermediate state $\sigma_q \in \mathcal{S}_{\texttt{int}}$, then the two chunks may be combined (via high-level rules) into a multi-chunk with state $S_{r:t} = \sigma_w \in \mathcal{S}_{\texttt{int}}$, say, with probability $\bar{\gamma}_{r:t}(w)$, where the bar indicates a (closed) multi-chunk of one or more (closed) chunks. We define the probability of a multi-chunk to be

$$\bar{\gamma}_{r:t}(w) \quad = \quad P(\mathbf{Y}_{r:t} = \mathbf{y}_{r:t}, M_t = \square \mid M_{r-1} = \square, S_{r:t} = \sigma_w)\,, \tag{115}$$

which, ambiguously, has the same form as for a single chunk.

In principle, there are many ways of defining how chunks may be combined. For example, we could define a *head-driven* grammar such that exactly one of states, $S_{r:s} = \sigma_p$ or $S_{s+1:t} = \sigma_q$, would be chosen as the overall *head* state $S_{r:t}$ of the combination. Each such combination would therefore represent a *dependency* where either *satellite* chunk $S_{r:s}$ attaches to head chunk $S_{s+1:t}$ on its right, or satellite chunk $S_{s+1:t}$ attaches to head chunk $S_{r:s}$ on its left.

Alternatively, we could allow production rules, such as n-ary rules of the form $\mathcal{S}_{\texttt{int}} \to \mathcal{S}_{\texttt{int}}^n$. For simplicity, and consistency with the usual context-free grammar in Chomksy normal form (CNF), we utilise binary rules of the form $\mathcal{S}_{\texttt{int}} \to \mathcal{S}_{\texttt{int}} \oplus \mathcal{S}_{\texttt{int}}$, and unary rules of the form $\mathcal{S}_{\texttt{leaf}} \to \mathcal{Y}$. However, we now need to also include additional unary rules of the form $\mathcal{S}_{\texttt{int}} \to \mathcal{S}_{\texttt{leaf}}$ and $\mathcal{S}_{\texttt{leaf}} \to \mathcal{S}_{\texttt{leaf}}$, such that the grammar $\mathcal{G}$ no longer has the CNF property but is still context-free. Note that these latter chunking rules implicitly correspond to unconstrained n-ary rules of the form $\mathcal{S}_{\texttt{int}} \to \mathcal{S}_{\texttt{leaf}}^n$ that provide the additional sequential dependencies.

Consequently, if a multi-chunk spanning tokens $\mathbf{y}_{r:t}$ has intermediate state $\sigma_w \in \mathcal{S}_{\texttt{int}}$, then any dichotomous partitioning of the multi-chunk via some binary rule $\sigma_w \to \sigma_p \oplus \sigma_q$ has position-invariant probability

$$P(S_{r:s} = \sigma_p, S_{s+1:t} = \sigma_q \mid S_{r:t} = \sigma_w) \quad \doteq \quad P(\sigma_w \to \sigma_p \oplus \sigma_q) \quad \doteq \quad a_{wpq}\,, \tag{116}$$

for every $s = r, \ldots, t-1$.

In general, we do not care about the internal structure of a multi-chunk. Hence, the *inside* pass sums over the probabilities of all internal chunking of a multi-chunk. Thus, appropriately modifying the model from a previous section, the *inner* probability of a multi-chunk is given by the recurrence relation

$$\bar{\gamma}_{r:t}(w) \quad = \quad \gamma_{r:t}(w) + \sum_{s=r}^{t-1} \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \sum_{\sigma_q \in \mathcal{S}_{\texttt{int}}} a_{wpq}\, \bar{\gamma}_{r:s}(p)\, \gamma_{s+1:t}(q)\,, \tag{117}$$

for $r < t$, with

$$\bar{\gamma}_{t:t}(w) \quad = \quad \gamma_{t:t}(w)\,. \tag{118}$$

This is very similar to the standard inside pass (with both γ and $\bar{\gamma}$ replaced by β), except that whereas there two adjoining subparses make a bigger subparse, here two adjoining chunks make a multi-chunk, not a single chunk. Also note that here we are explcitly combining a multi-chunk with a single chunk (to its right), rather than a multi-chunk with a multi-chunk, since (as noted

previously) marginalising over the latter would count some internal chunks multiple times. The extra leading term in the recurrence relation comes from the fact that a multi-chunk may also be comprised of a single chunk.

Due to the similarity with the standard inside pass, we choose to also denote the probability of a multi-chunk via $\bar{\beta}_{r:t}(p) \doteq \bar{\gamma}_{r:t}(p)$. This is not to be confused with the backward probability $\beta_{r:t}(p, i)$.

As a quick example of the difference between a chunk and a multi-chunk, consider the sentence "*The cat sat on the mat.*", with possible chunking $[\mathit{The} \oplus \mathit{cat}]_{\texttt{NP}}[\mathit{sat}]_{\texttt{VP}}[\mathit{on} \oplus \mathit{the} \oplus \mathit{mat}]_{\texttt{PP}}$. However, an alternative chunking might be $[\mathit{The} \oplus \mathit{cat}]_{\texttt{NP}}[\mathit{sat} \oplus \mathit{on} \oplus \mathit{the} \oplus \mathit{mat}]_{\texttt{VP}}$. The single chunk $[\mathit{sat} \oplus \mathit{on} \oplus \mathit{the} \oplus \mathit{mat}]_{\texttt{VP}}$ is not the same as the multi-chunk $\{[\mathit{sat}]_{\texttt{VP}}[\mathit{on} \oplus \mathit{the} \oplus \mathit{mat}]_{\texttt{PP}}\}_{\texttt{VP}}$, even though both span the same tokens, and have the same intermediate state. Furthermore, the probability $P(\{[\mathit{sat}]_{\texttt{VP}}[\mathit{on} \oplus \mathit{the} \oplus \mathit{mat}]_{\texttt{PP}}\}_{\texttt{VP}})$ is only one way of contributing to the total multi-chunk probability $\bar{\bar{\alpha}}_{3:6}(\texttt{VP})$. However, the single chunk has probability $\bar{\alpha}_{3:6}(\texttt{VP}) = P([\mathit{sat} \oplus \mathit{on} \oplus \mathit{the} \oplus \mathit{mat}]_{\texttt{VP}})$ exactly.

1.4.4 Outside pass

Now, recall from the previous section that a multi-chunk spanning tokens $\mathbf{y}_{r:t}$ has inside probability $\bar{\beta}_{r:t}(p)$, marginalising over all inner structure. Hence, the remainder of the derivation of the sequence $\mathbf{y}_{1:T}$ forms the *outer* structure with outer probability $\bar{\alpha}_{r:t}(p)$, defined as

$$\bar{\alpha}_{r:t}(p) \quad \doteq \quad P(\mathbf{Y}_{1:r-1} = \mathbf{y}_{1:r-1}, M_{r-1} = \square, S_{r:t} = \sigma_p, \mathbf{Y}_{t+1:T} = \mathbf{y}_{t+1:T}) \,. \tag{119}$$

This is not to be confused with the forward probability $\alpha_{r:t}(p, i)$. As a direct consequence of this definition, we now obtain

$$P(\mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_{r-1} = \square, M_t = \square) \quad = \quad \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \bar{\alpha}_{r:t}(p)\, \bar{\beta}_{r:t}(p) \,, \tag{120}$$

and thus

$$P(\bar{\mathbf{y}}_{1:T}) \quad \doteq \quad P(M_0 = \square, \mathbf{Y}_{1:T} = \mathbf{y}_{1:T}, M_T = \square) \quad = \quad \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \bar{\alpha}_{1:T}(p)\, \bar{\beta}_{1:T}(p) \,, \tag{121}$$

where we have now defined the logical proposition

$$\bar{\mathbf{y}}_{1:T} \quad \doteq \quad M_0 = \square \wedge \mathbf{Y}_{1:T} = \mathbf{y}_{1:T} \wedge M_T = \square \tag{122}$$

for brevity.

We now suppose that the entire sequence has some over-arching root state $\sigma_p \in \mathcal{S}_{\texttt{root}}$ with probability

$$P(S_{1:T} = \sigma_p \mid M_0 = \square) \quad \doteq \quad \iota_p \,. \tag{123}$$

Consequently, the outside pass commences with the edge case

$$\bar{\alpha}_{1:T}(p) \quad \doteq \quad P(M_0 = \square, S_{1:T} = \sigma_p) \quad = \quad P(M_0 = \square)\, \iota_p \,. \tag{124}$$

We take $P(M_0 = \square) = 1$ on the basis that we have observed either a complete sequence or at least a closed multi-chunk, since otherwise chunking a partial multi-chunk would be very difficult.

The recurrence relation for the outside probabilities is derived by following the usual reasoning for the inside-outside algorithm. In particular, for $r > 1$ there exists some position $1 \leq s < r$

giving rise to a closed chunk spanning tokens $\mathbf{y}_{s:r-1}$ with probability $\gamma_{s:r-1}(q)$. The chunk and the multi-chunk can now be combined via binary rules of the form $\sigma_w \to \sigma_q \oplus \sigma_p$ to form a larger multi-chunk spanning tokens $\mathbf{y}_{s:t}$ with inner probability $\bar{\beta}_{s:t}(w)$. Consequently, what remains is a smaller outer structure with outer probability $\bar{\alpha}_{s:t}(w)$.

Similarly, for $t < T$ there exists some position $t < s \leq T$ leading to a closed chunk spanning tokens $\mathbf{y}_{t+1:s}$ with probability $\gamma_{t+1:s}(q)$. Hence, the multi-chunk and the chunk may be combined via binary rules of the form $\sigma_w \to \sigma_p \oplus \sigma_q$ into a larger multi-chunk spanning tokens $\mathbf{y}_{r:s}$ with inner probability $\bar{\beta}_{r:s}(w)$. What remains forms a smaller outer structure with probability $\bar{\alpha}_{r:s}(w)$.

Consequently, marginalising over all the possible ways of expanding a multi-chunk to either the left or to the right, the outer probabilities are computed via the recurrence relation

$$\bar{\alpha}_{r:t}(p) \quad = \quad \sum_{s=1}^{r-1} \sum_{\sigma_q \in \mathcal{S}_{\texttt{int}}} \sum_{\sigma_w \in \mathcal{S}_{\texttt{int}}} a_{wqp}\, \gamma_{s:r-1}(q)\, \bar{\alpha}_{s:t}(w) + \sum_{s=t+1}^{T} \sum_{\sigma_q \in \mathcal{S}_{\texttt{int}}} \sum_{\sigma_w \in \mathcal{S}_{\texttt{int}}} a_{wpq}\, \gamma_{t+1:s}(q)\, \bar{\alpha}_{r:s}(w) \tag{125}$$

Once again, this resembles the standard outside algorithm (with α instead of $\bar{\alpha}$, and β instead of γ).

1.4.5 Grammatical restrictions

We noted via an example at the end of a previous section that chunking ambiguity may arise due to the existence of nested rules, such as $\texttt{VP} \to \texttt{VP} \oplus \texttt{PP}$. One possible way of avoiding such situations is to label each state with an explicit role, e.g. $\texttt{VP}_{\texttt{bin}} \to \texttt{VP}_{\texttt{chunk}} \oplus \texttt{PP}_{\texttt{chunk}}$, such that $\texttt{VP}_{\texttt{bin}} \neq \texttt{VP}_{\texttt{chunk}}$. More generally, such role labelling corresponds to the separation of intermediate states $\mathcal{S}_{\texttt{int}}$ into states $\mathcal{S}_{\texttt{bin}}$ that may appear at the head of binary rules, and other states $\mathcal{S}_{\texttt{chunk}}$ that may produce leaf states $\mathcal{S}_{\texttt{leaf}}$, such that $\mathcal{S}_{\texttt{int}} = \mathcal{S}_{\texttt{bin}} \cup \mathcal{S}_{\texttt{chunk}}$. The binary rules would therefore take the form $\mathcal{S}_{\texttt{bin}} \to (\mathcal{S}_{\texttt{bin}} \cup \mathcal{S}_{\texttt{chunk}})^2$, and the non-token unary rules would take the form $\mathcal{S}_{\texttt{chunk}} \to \mathcal{S}_{\texttt{leaf}}$, e.g. $\texttt{VP}_{\texttt{chunk}} \to \texttt{V}_{\texttt{leaf}}$.

Note that we do not necessarily require that $\mathcal{S}_{\texttt{bin}} \cap \mathcal{S}_{\texttt{chunk}} = \emptyset$, just as we do not require that $\mathcal{S}_{\texttt{int}} \cap \mathcal{S}_{\texttt{leaf}} = \emptyset$. However, the existence of unary rules of the form $\mathcal{S}_{\texttt{chunk}} \to \mathcal{S}_{\texttt{leaf}}$ in the grammar $\mathcal{G}$ implies a degree of separation between leaf states and intermediate states, such that the grammatical restriction $\mathcal{S}_{\texttt{int}} \cap \mathcal{S}_{\texttt{leaf}} = \emptyset$ would be justified. Note, however, that the existence of single-token sequences precludes the exclusion $\mathcal{S}_{\texttt{root}} \cap \mathcal{S}_{\texttt{int}} = \emptyset$, although it does not prevent choosing $|\mathcal{S}_{\texttt{root}}| = 1$.

Such restrictions on the grammar, specifically mutual exclusions between subsets of states, would typically be imposed by the explcit setting of zero-valued probabilities (known as *structural zeros*) within the corresponding conditional probability tables. Structural zeros are distinct from *estimated zeros* caused by a lack of training data, and hence parameter estimation should typically allow non-zero prior probabilites at all places other than structural zeros.

1.4.6 Parameter estimation

The hierarchical chunking grammar $\mathcal{G}$ is now parameterised by a collection of conditional probability tables. The *chunk combination* rules are specified by the tensor $\mathbf{A} = [a_{wpq}]$, for $\sigma_w \in \mathcal{S}_{\texttt{bin}}$ and $\sigma_p, \sigma_q \in \mathcal{S}_{\texttt{bin}} \cup \mathcal{S}_{\texttt{chunk}}$. The *token generation* rules are specified by the matrix $\mathbf{B} = [b_{im}]$, for $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$ and $\nu_m \in \mathcal{Y}$. The *chunk transition* rules are specified by the matrix $\mathbf{C} = [c_{ij}]$, for $\sigma_i, \sigma_j \in \mathcal{S}_{\texttt{leaf}}$. The *chunk initiation* rules are specified by the matrix $\mathbf{D} = [d_{pi}]$, for $\sigma_p \in \mathcal{S}_{\texttt{chunk}}$ and $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$. Finally, the *chunk termination* rules are specified by the vector $\boldsymbol{\tau} = [\tau_i]$, for $\sigma_i \in \mathcal{S}_{\texttt{leaf}}$, and the *sequence initiation* rules are specified by the vector $\boldsymbol{\iota} = [\iota_p]$, for $\sigma_p \in \mathcal{S}_{\texttt{root}}$.

The maximum likelihood (ML) estimates of these probabilities are obtained via iterations of the expectation-maximisation (EM) procedure. The individual estimates of the rule probabilities are obtained as normalisations of the expected joint counts of each rule, namely

$$\hat{a}_{wpq} \;=\; \frac{\hat{N}^A_{wpq}}{\hat{N}^A_{w\cdot\cdot}}\,, \hat{b}_{im} \;=\; \frac{\hat{N}^B_{im}}{\hat{N}^B_{i\cdot}}\,, \hat{c}_{ij} \;=\; \frac{\hat{N}^C_{ij}}{\hat{N}^C_{i\cdot}}\,, \hat{d}_{pi} \;=\; \frac{\hat{N}^D_{pi}}{\hat{N}^D_{p\cdot}}\,, \hat{\tau}_i \;=\; \frac{\hat{N}^{\square}_{i}}{\hat{N}^{\square}_{i} + \hat{N}^{\oplus}_{i}}\,, \hat{\iota}_p \;=\; \frac{\hat{N}^{\triangleleft}_{p}}{\hat{N}^{\triangleleft}_{\cdot}}\,, \tag{126}$$

Note that, in general, these counts may be summed over all sequences in the training corpus, and may also be initialised with prior counts.

As explained briefly in a previous section, to compute the expected count $\hat{N}_R$ of each rule $R \in \mathcal{R}$ for a given sequence $\mathbf{y}$, we essentially count the number $f_R(T)$ of times rule R appears in each parse T, weight this count by the conditional probability $P(T \mid \mathbf{y})$ of the parse, and sum these weighted counts over every possible parse $T \in \mathcal{T}(\mathbf{y})$. More traditionally, we may (loosely speaking) enumerate each distinct parse structure S (which does not specify the states of the nodes), and for each such S compute the conditonal probability $P(S, R \mid \mathbf{y})$ of the rule R (which does specify the states) occuring within that structure. The sum of these conditonal probabilities over all structures then gives the expected count as $\hat{N}_R = P(R \mid \mathbf{y})$.

Thus, for the chunk combination rule $R^A_{wpq} : \sigma_w \to \sigma_p \oplus \sigma_q$ we have

$$\hat{N}^A_{wpq} \;=\; \frac{1}{Z} \sum_{r=1}^{T-1} \sum_{t=r+1}^{T} \sum_{s=r}^{t-1} P(\bar{\mathbf{y}}_{1:T}, S_{r:t} = \sigma_w, S_{r:s} = \sigma_p, S_{s+1:t} = \sigma_q)\,, \tag{127}$$

where

$$Z \;\doteq\; P(\bar{\mathbf{y}}_{1:T})\,. \tag{128}$$

Now, from the inside pass, the innards of a multi-chunk comprised of two or more chunks may be exposed via

$$\bar{\beta}_{r:t}(w) - \gamma_{r:t}(w) \;=\; \sum_{s=r}^{t-1} \sum_{\sigma_p \in \mathcal{S}_{\text{int}}} \sum_{\sigma_q \in \mathcal{S}_{\text{int}}} a_{wpq}\, \bar{\beta}_{r:s}(p)\, \gamma_{s+1:t}(q)\,, \tag{129}$$

Furthermore, the inner probability $\bar{\beta}_{r:t}(w)$ has corresponding outer probability $\bar{\alpha}_{r:t}(w)$ that completes the derivation. Consequently, the joint probability factors as

$$P(\bar{\mathbf{y}}_{1:T}, S_{r:t} = \sigma_w, S_{r:s} = \sigma_p, S_{s+1:t} = \sigma_q) \;=\; a_{wpq}\, \bar{\beta}_{r:s}(p)\, \gamma_{s+1:t}(q)\, \bar{\alpha}_{r:t}(w)\,, \tag{130}$$

giving

$$\hat{N}^A_{wpq} \;=\; \frac{a_{wpq}}{Z} \sum_{r=1}^{T-1} \sum_{t=r+1}^{T} \sum_{s=r}^{t-1} \bar{\beta}_{r:s}(p)\, \gamma_{s+1:t}(q)\, \bar{\alpha}_{r:t}(w)\,. \tag{131}$$

Similarly, the within-chunk transition rule $R^C_{ij} : \sigma_i \xrightarrow{\oplus} \sigma_j$ has expected count

$$\hat{N}^C_{ij} \;=\; \frac{1}{Z} \sum_{t=1}^{T-1} P(\bar{\mathbf{y}}, S_t = \sigma_i, M_t = \oplus, S_{t+1} = \sigma_j)\,. \tag{132}$$

Now, from the backward pass, we may expose the innards of a closed chunk via

$$\gamma_{r:s}(p) \;=\; \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha_{r:t}(p,i)\, \bar{\tau}_i\, c_{ij}\, \breve{b}_{j,t+1}\, \beta_{t+1:s}(p,j)\,, \tag{133}$$

for $r < s$. Next, since the chunk determines an inner probability, we close the derivation with outer probability $\bar{\alpha}_{r:s}(p)$, and then marginalise across the structure of the chunk. Consequently, we obtain

$$\hat{N}^C_{ij} \;=\; \frac{c_{ij}}{Z} \sum_{r=1}^{T-1} \sum_{s=r+1}^{T} \sum_{t=r}^{s-1} \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \alpha_{r:t}(p,i)\, \bar{\tau}_i\, \breve{b}_{j,t+1}\, \beta_{t+1:s}(p,j)\, \bar{\alpha}_{r:s}(p)\,. \tag{134}$$

Next, the chunk initiation rule $R^D_{pi} : \sigma_p \xrightarrow{\square} \sigma_i$ has expected count

$$\hat{N}^D_{pi} \;=\; \frac{1}{Z} \sum_{t=1}^{T} P(\bar{\mathbf{y}}_{1:T}, M_{t-1} = \square, S_{t:*} = \sigma_p, S_t = \sigma_i)\,. \tag{135}$$

Now, from the both the forward and backward passes, we may expose the start of a closed chunk via

$$\gamma_{t:s}(p) \;=\; \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} d_{pi}\, \breve{b}_{it}\, \beta_{t:s}(p,i)\,. \tag{136}$$

Hence, we obtain the expected count

$$\hat{N}^D_{pi} \;=\; \frac{d_{pi}}{Z} \sum_{t=1}^{T} \sum_{s=t}^{T} \breve{b}_{it}\, \beta_{t:s}(p,i)\, \bar{\alpha}_{t:s}(p)\,. \tag{137}$$

Similarly, the token generation rule $R^B_{im} : \sigma_i \to \nu_m$ has expected count

$$\hat{N}^B_{im} \;=\; \frac{1}{Z} \sum_{t=1}^{T} P(\bar{\mathbf{y}}_{1:T}, S_t = \sigma_i, Y_t = \nu_m) \;=\; \frac{1}{Z} \sum_{t=1}^{T} \delta(y_t = \nu_m)\, P(\bar{\mathbf{y}}_{1:T}, S_t = \sigma_i)\,. \tag{138}$$

Once again, the leaf state S_t occurs within an arbitrary chunk with span $r \leq t \leq s$. For $r = t$, we have

$$\gamma_{t:s}(p) \;=\; \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} d_{pi}\, \breve{b}_{it}\, \beta_{t:s}(p,i)\,. \tag{139}$$

Alternatively, for $r < t$, we have

$$\gamma_{r:s}(p) \;=\; \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \sum_{\sigma_j \in \mathcal{S}_{\texttt{leaf}}} \alpha_{r:t-1}(p,j)\, \bar{\tau}_j\, c_{ji}\, \breve{b}_{it}\, \beta_{t:s}(p,i) \tag{140}$$

$$\;=\; \sum_{\sigma_i \in \mathcal{S}_{\texttt{leaf}}} \frac{\alpha_{r:t}(p,i)}{\breve{b}_{it}}\, \breve{b}_{it}\, \beta_{t:s}(p,i)\,. \tag{141}$$

Consequently, the expected count is

$$\hat{N}^B_{im} \;=\; \frac{b_{im}}{Z} \sum_{t=1}^{T} \delta(y_t = \nu_m) \sum_{s=t}^{T} \sum_{\sigma_p \in \mathcal{S}_{\texttt{int}}} \left\{ d_{pi}\, \bar{\alpha}_{t:s}(p) + \sum_{r=1}^{t-1} \frac{\alpha_{r:t}(p,i)}{\breve{b}_{it}}\, \bar{\alpha}_{r:s}(p) \right\} \beta_{t:s}(p,i)\,. \tag{142}$$

Alternatively, we may simply use the fact that

$$P(\bar{\mathbf{y}}_{1:T}, S_t = \sigma_i) \;=\; \sum_{r=1}^{t} \sum_{s=t}^{T} \sum_{\sigma_p \in \mathcal{S}_{\text{int}}} \alpha_{r:t}(p,i)\, \beta_{t:s}(p,i)\, \bar{\alpha}_{r:s}(p)\,, \tag{143}$$

which does not expose the prior probability b_{im} as a back-propagation factor.

The chunk termination rule $R_i^{\square} : \sigma_i \stackrel{\square}{\rightarrow}$ requires expected counts

$$\hat{N}_i^{\square} \;=\; \frac{1}{Z} \sum_{t=1}^{T} P(\bar{\mathbf{y}}_{1:T}, S_t = \sigma_i, M_t = \square) \tag{144}$$

$$\;=\; \frac{\tau_i}{Z} \sum_{r=1}^{T} \sum_{t=r}^{T} \sum_{\sigma_p \in \mathcal{S}_{\text{int}}} \alpha_{r:t}(p,i)\, \bar{\alpha}_{r:t}(p)\,, \tag{145}$$

and

$$\hat{N}_i^{\oplus} \;=\; \frac{1}{Z} \sum_{t=1}^{T-1} P(\bar{\mathbf{y}}_{1:T}, S_t = \sigma_i, M_t = \oplus) \tag{146}$$

$$\;=\; \frac{\bar{\tau}_i}{Z} \sum_{r=1}^{T-1} \sum_{s=r+1}^{T} \sum_{t=r}^{s-1} \sum_{\sigma_p \in \mathcal{S}_{\text{int}}} \sum_{\sigma_j \in \mathcal{S}_{\text{leaf}}} \alpha_{r:t}(p,i)\, c_{ij}\, \breve{b}_{j,t+1}\, \beta_{t+1:s}(p,j)\, \bar{\alpha}_{r:s}(p) \tag{147}$$

$$\;=\; \frac{1}{Z} \sum_{r=1}^{T-1} \sum_{s=r+1}^{T} \sum_{t=r}^{s-1} \sum_{\sigma_p \in \mathcal{S}_{\text{int}}} \alpha_{r:t}(p,i)\, \beta_{t:s}(p,i)\, \bar{\alpha}_{r:s}(p)\,, \tag{148}$$

where the last expression does not expose the prior porability $\bar{\tau}_i$ as a back-propagation term. Alternatively, we recall that

$$\hat{N}_{ij}^{C} \;=\; \frac{1}{Z} \sum_{t=1}^{T-1} P(\bar{\mathbf{y}}, S_t = \sigma_i, M_t = \oplus, S_{t+1} = \sigma_j)\,, \tag{149}$$

and thus $\hat{N}_i^{\oplus} = \hat{N}_{i\cdot}^{C}$.

Finally, the sequence initiation rule $R_p^{\triangleleft} : \stackrel{\triangleleft}{\rightarrow} \sigma_p$ has expected count

$$\hat{N}_p^{\triangleleft} \;=\; \frac{1}{Z} P(\bar{\mathbf{y}}_{1:T}, S_{1:T} = \sigma_p) \;=\; \frac{\iota_p}{Z} P(M_0 = \square)\, \bar{\beta}_{1:T}(p)\,. \tag{150}$$

1.5 References

[1a] J. Kupiec (1992): "*Robust part-of-speech tagging using a hidden Markov model*", Computer speech & language 6(3): 225–242.

[1b] J. Kupiec (1992): "*An Algorithm for Estimating the Parameters of Unrestricted Hidden Stochastic Context-Free Grammars*", COLING 1992 Vol. 1.

[2] S. Fine, Y. Singer, and N. Tishby (1998) "*The Hierarchical Hidden Markov Model: Analysis and Applications*", Machine Learning 32. (PDF)

[3] H.H. Bui, Q. Phung and S. Venkatesh (2004) "*Hierarchical Hidden Markov Models with General State Hierarchy*", AAAI-04 (National Conference on Artificial Intelligence). (PDF)

[4] J. Eisner (2016): "*Inside-Outside and Forward-Backward algorithms are just backprop*", Proc. Workshop on Structured Prediction for NLP. (PDF)