The neural basis of structure in language: bridging the gap between symbolic and connectionist models of language processing

Borensztajn, G.

Publication date
2011

Chapter 6

Episodic grammar

In this and the following chapters I will introduce episodic-HPN, an extension of HPN with an episodic memory. The model is based on an original hypothesis about the interaction of semantic and episodic memory in language processing. It shows how language processing can be understood in terms of memory retrieval, or as a priming effect, and language acquisition in terms of memory consolidation. I will point out that the perceived dichotomy between rule-based versus exemplar-based language modeling can be interpreted in a neuro-biological perspective in terms of the interaction between a semantic memory system that encodes linguistic knowledge in the form of abstract rules, and an episodic memory that stores concrete linguistic events. Before I present the full episodic-HPN model in Chapter 8, I will consider in this chapter the concept of parsing with an episodic memory for the supervised and symbolic case, using nonterminal labels learned from a treebank. I will implement a probabilistic, episodic grammar and evaluate its performance as a reranker on a realistic corpus of natural language, the Wall Street Journal.

6.1 Episodic memory

The previous chapter (section 5.6.3) identified several limitations of the current version of HPN. For instance, since all information available to the model are the

metric distances between network units, the model cannot represent (sentence) context, or do contextual conditioning. As a consequence, although HPN is suited for emulating (probabilistic) *context free* grammars (as was shown in section 5.2), it is ill-equipped for realistic language processing, where decisions depend on structural and lexical sentence context. The HPN network encodes, through the substitution space, context free relations between abstract (encapsulated) syntactic units, and as such qualifies as a *semantic* memory for the syntactic domain.

As defined in section 2.7, *semantic memory* refers to a person's general world knowledge, including language, in the form of abstract concepts that are systematically related to each other; *Episodic memory*, on the other hand, is a person's memory of personally experienced events or episodes, embedded in a temporal, spatial and emotional context (see section 2.7 for an extensive discussion of the human memory system).

The ideas developed in this chapter start from the observation that the scientific debate on the relation between semantic and episodic memory parallels, in a striking manner, an ongoing controversy about modeling language: one side in the debate is focusing on evidence for abstract, rule-based grammars [e.g. Marcus, 2001], and the other side emphasizes the item-based nature of grammar with a role (particularly in acquisition) for concrete sentence fragments larger than rules [e.g., Tomasello, 2000b]. While a rule-based grammar can be conceived of as an instance of semantic memory, as it encodes abstract, relational linguistic knowledge, the item-based approach suggests a role for episodic memory in sentence processing, since it reuses concrete (rather than abstract) linguistic experiences that have been memorized.

Assuming that the language domain mirrors cognitive processes from other domains, one expects that it would be illuminating to incorporate the notion of a semantic-episodic memory interaction within a computational model of language processing. In this chapter I will formulate a theory of episodic-semantic memory interaction, and based on this an 'episodic' model of syntax, called 'episodic grammar', that links language processing to memory processes, or more precisely to episodic memory retrieval. While the current chapter as a first step only introduces a *symbolic* episodic grammar – leaving out the topology – in Chapter 8 I will enrich the 'semantic' HPN model of the previous chapter with an episodic memory for sentences, thus lifting its limitation for dealing with sentence context. The episodic grammar model should take into account the following basic empirical facts about episodic memory

- *Physical traces.* All episodic experiences that occur during the lifespan of an individual, and that can be consciously remembered, leave physical *memory traces* in the brain. This includes memories of sentences that have been processed by the language system.
- *Chronological order preservation.* Most people are able to recover the ap-

proximate *chronological order* of their episodic memories. Thus, the relative order of the episodes must be somehow encoded in the representations of their traces.

- *Content addressability.* Priming effects demonstrate that static memories (for instance the memory of a smell) trigger episodic memories that are strongly associated with them. It is commonly believed that retrieval of episodic memories is contingent on cues from semantic memory. To account for *content addressability* an episodic memory must support local access from semantic memory units to their associated episodes (as implemented for instance in Hopfield networks [e.g., Hopfield, 1982]).
- *Sequentiality.* In the Memory Prediction Framework it is emphasized that the function of memory is to make temporal inferences (i.e., predict). According to [e.g., Eichenbaum, 2004] episodes are construed as temporal sequences of (time-less) semantic elements, bound together within a certain context (see Figure 2.7 in section 2.7.1).
- *Separatibility and identifiability.* The memory system must be able to identify and disambiguate an episode, even if it overlaps with another episode that is partly composed of the same semantic units. It is thought that to this end special 'context neurons' exist, that fire only for the duration of a specific episode [e.g., Levy, 1996].

There exist several connectionist models of episodic memory in the literature [e.g., Hopfield, 1982, Miikkulainen, 1999, McQueen, 2005, McClelland et al., 1995]. In section 8.5.1 I will discuss an instantiation of the latter, [O'Reilly and Norman, 2002] in the context of memory consolidation. Yet, as far as I know, to date there exists no theory of episodic-semantic memory interaction that is applicable to syntactic processing.

6.1.1 Proposal for the representation of episodes as distributed traces in semantic units

I propose that the episodic memory of a sentence is distributed across semantic memory units (i.e., the HPN nodes), and consists of physical traces, contained inside the nodes, that keep a record of the nodes participation in the derivation of the processed sentence. (In general, I claim that the episodic memory of a complex event consists of physical traces, distributed across the primitive semantic units that took part in structurally encoding the particular event.) This is illustrated schematically in Figure 6.1, which shows the episodic memory traces in the HPN network after hearing the sentences *girl who dances likes tango* (light colored traces) and *boy likes mango* (dark colored traces). Each of the traces of a processed sentence points to its succeeding and preceding node in the derivation, which allows HPN to reconstruct the original derivation from the traces. Conceptually, all that is required to upgrade from semantic HPN to episodic HPN is to

turn the existing local *short term memories* in the slots, where pointers to bound nodes are temporarily stored, into *long term memories* after a sentence has been successfully processed. (Note that this proposal implies that the semantic units involved in encoding an episode, like those involved in an HPN derivation, are *dynamically* bound: see section 8.4.2 for a neural perspective on episodic memory encoding in HPN.)

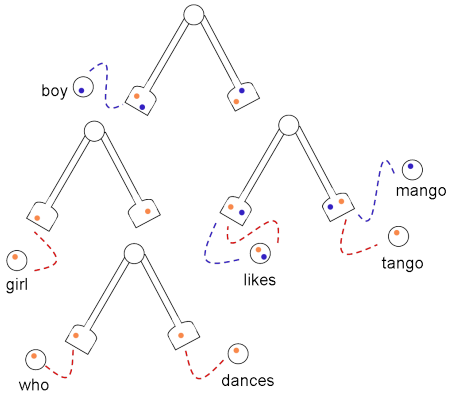

Figure 6.1: Episodic traces of a sentence (drawn as colored dots) are stored in local memories of visited nodes in the HPN network. In HPN the nodes and slots are situated in a topology.

6.2 Episodic grammar — model outline

For a more formal introduction to the topic of episodic grammar, let us leave for the moment the framework of HPN, and first deal with a symbolic implementation of episodic grammar. I will come back to the HPN formalism in Chapter 8, when I will work out the details of episodic-HPN. Also in the symbolic approach it is useful to take the point of view of a grammar as a network of interconnected *treelets*, that can combine with each other through substitution. I will assume that context-free rules from traditional grammars correspond one-to-one to such treelets, which thus play the role of the *compressor nodes* in HPN. I will also assume that the treelets possess a register (an internal memory, corresponding to a *slot* in HPN) that keeps track of the correct order of application of the syntactic operations.

As in HPN, in the episodic grammar a derivation is a sequence of visits to treelets, whereby treelets are bound through serial binding. The standard approach assumes a *top-down*, left-to-right derivation: each next rule is combined through left-most substitution with the partial tree derived so-far. I will also consider *left-corner derivations* in the next section.

In order to remember the correct order of derivation (which can vary depending on the chosen derivation strategy) the episodic traces encode the sentence

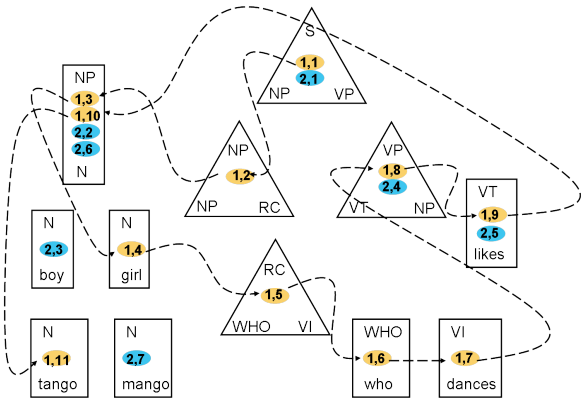

Figure 6.2: Episodic traces of two sentences (drawn as colored ovals) are stored in local memories of visited treelets (indicated by triangles and rectangles) in the *symbolic* episodic network. Note that by virtue of their ordinal number the traces implement pointers to successor treelets in a derivation (drawn for the first sentence alone).

number (s) as well as the position (k) of the treelet within the derivation. In Figure 6.2 the traces (for a top-down derivation) are identified by these two numbers, indicated as $\langle s, k \rangle$ inside the treelets. Note that after hearing many sentences a single treelet will store traces for all sentences that have visited it, which are distinguished by their sentence number, and possibly multiple visits from the same sentence.

The episodic sentence memories stored in the traces can also be recruited for the purpose of processing novel, unseen sentences. The idea is that when the derivation of a novel sentence arrives at a treelet, the traces encountered within the treelet trigger memories of stored exemplars. These receive an activation value whose strength depends on how close the stored derivation is to the pending derivation (see section 6.2.3). Every next step in the derivation is determined by competition between traces of different exemplars, each having its own preference for a successor treelet, and its own activation strength. In this view sentence processing (or parsing) can be interpreted as being subject to a *priming* effect: the traces prime or reactivate derivations of previously processed sentences (through content addressability), and restore the memory of previous parser decisions.

The above proposal satisfies the requirements of an episodic memory, as mentioned in the previous section, and it conforms to the view that episodes consist of pointers that bind semantic memory units into temporal sequences [e.g., Shastri, 2002]; content addressability is satisfied because activation of a single trace in a semantic unit triggers an entire episode. Parts of episodes are thus *reconstructed* on-the-fly at test time, rather than searched for. Further, chronological order preservation, as well as sequentiality and separatibility are trivially satisfied by the way that traces are encoded. Given a probabilistic interpretation, the episodic

grammar offers an explicit computational instantiation of the reinstatement hypothesis of episodic retrieval.

6.2.1 The left corner episodic grammar

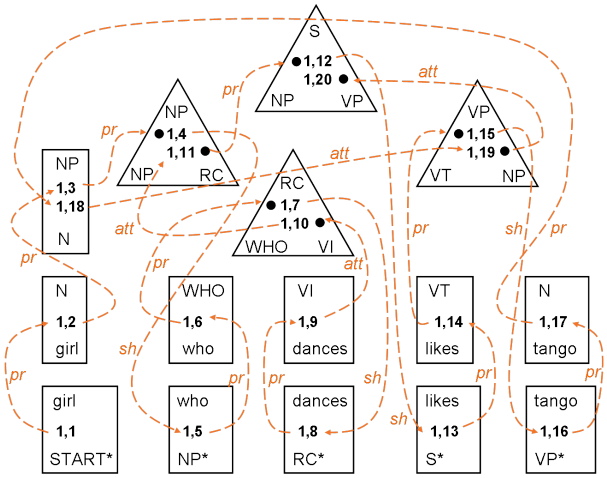

Figure 6.3: Episodic memory traces in the left corner episodic grammar after deriving the sentence *girl who dances likes tango*.

One of the advantages of the episodic approach is that it allows for comparing different derivation strategies within a single framework, and find out what the effect is of a different order of application of operations on treelets. An interesting parsing strategy from a cognitive point of view is *left corner parsing* [Rosenkrantz and Lewis II, 1970], since it proceeds incrementally from left to right, and combines top-down and bottom-up processing.

As explained in section 3.1.3, in left corner parsing the grammar rules are introduced bottom-up by a *project* operation to the *left corner* of the rule. The 'left corner' is the left-most symbol on the right hand side of a phrase structure rule; in the episodic framework it refers to the bottom-left nonterminal of a treelet. As long as there are no *completed* (i.e., fully processed) treelets, the next word in the sentence is introduced by a *shift* operation; otherwise the derivation can either *project* to a new treelet, or *attach* to a not yet completed treelet that has been previously introduced. Figure 6.3 shows an episodic left corner derivation for the sentence *girl who dances likes tango*. The shift, project and attach operations are indicated in the figure by their abbreviations.

Whereas most standard probabilistic left corner parsers compute the parse probability of a given sentence [e.g., Moore, 2004, Manning and Carpenter, 1997], hence assume a deterministic *shift* move, here we are interested in the joint probability of the parse and the sentence. It will be assumed that the *shift* move requires an additional step in the derivation, connecting an 'incomplete' treelet

(after attach or project) with a word, as illustrated in Figure 6.3. Thus, the derivation is *connected*, and proceeds according to a fixed linear order, which is a prerequisite for the episodic approach. To this end special treelets have been introduced that execute the shift to the next word (e.g., $RC* \rightarrow dances$).[1] These treelets employ special *starred* nonterminals (e.g., $RC*$): one or more stars indicate the register position in the treelet from where the shift operation originates (e.g., $RC \rightarrow WHO \ * \ VI$). The derivation starts with a shift operation from the special $START*$ symbol to the first word of the sentence.

One important difference with the top-down derivation strategy is that upon every attach operation treelets are reengaged in the derivation. It is therefore important to distinguish treelets by their register state, which keeps track of the operations (project, attach) performed on the treelet. Episodic traces are thus associated with and stored in a treelet *in a specific register state*, which is indicated in Figure 6.3 by adding a dot before or after the trace.[2]

6.2.2 Training the episodic grammar

To evaluate the concept of episodic grammar quantitatively a probabilistic version is implemented that is trained on a corpus of realistic language. Probabilistic grammars assign probabilities to different parses of a sentence and select the most probable one, hence can be evaluated on their ability to disambiguate between parses. As explained in section 3.1.5, one estimates the parameters of the probabilistic episodic grammar from a treebank, which is a corpus consisting of natural language sentences manually annotated with phrase structure trees.

After deciding on a derivation strategy (i.e., top-down or left-corner), the training proceeds by distributing a trace $e = \langle s, k \rangle$ in every visited treelet t_k of derivation $x = \langle t_0, \ldots, t_k, \ldots, t_n \rangle$ of sentence number s in the treebank. Specifically, given a treebank, then

1. Create an empty treelet for every unique context free production extracted from the treebank. In case of a left corner derivation one must also create separate treelets for distinct visits to the same production (i.e., after an attach), that is one must distinguish register positions of a treelet. Further, in case of a left corner derivation, create special shift treelets (as described in section 6.2.1) corresponding to the *shift* moves (to terminals) of the left corner parser.
2. For every treebank parse determine the sequential order of (register-indexed) treelets according to the chosen derivation strategy.

[1] This strategy is based on the probabilistic Left Corner Shifting Grammar (LCSG), which will be developed in the next chapter. The LCSG includes shift probabilities, hence defines a language model, which allows for the calculation of sentence probabilities.

[2] In general, there can be as many register positions as there are children in the treelet. In the top-down episodic grammar the register is always in position 0, hence it is not indicated in Figure 6.2.

3. For every step k in the derivation of sentence number s, leave a (register-indexed) trace in the visited treelet, encoded as $\langle s, k \rangle$.

At every derivation step the probability of moving to the next treelet in the derivation can be computed based on the traces in the current treelet and their activations, according to Equation 6.2.

6.2.3 Statistical parsing with the episodic grammar

After training the grammar one can use the model to assign probabilities to candidate parses of a new sentence. Given an ongoing derivation d of a sentence, that has arrived at a certain treelet $t_{q,\,r}$, in register position r, one defines the probability of continuing the derivation to any other treelet $t_{q',\,s}$ in register position s based on the activation values of the episodic traces of earlier derivations stored in treelet $t_{q,\,r}$. The activation $A(e_{x_i})$ of the trace e_{x_i} (in $t_{q,\,r}$) of earlier derivation x is a function of the common history $CH(e_{x_i}, d)$ of derivation x (of which e_{x_i} is the i^{th} trace) with the ongoing derivation d. The CH is simply given by the number of derivation steps (i.e., treelets) that the stored derivation x and the pending derivation d have shared the same path before arriving at $t_{q,\,r}$. Episodic traces that share a long common history should contribute relatively much to the parser decision. A convenient choice for the activation of a trace is

$$A(e_{x_i}) = \lambda_0^{CH(e_{x_i}, d)} \tag{6.1}$$

where λ_0 is a parameter of the model. Depending on the chosen derivation strategy (e.g., top-down or left corner), the traces have different CH's, hence receive different activations.

All information to calculate these activations is stored inside treelet $t_{q,\,r}$; computations are thus local, and compatible with the constraints imposed by a neurally plausible, or connectionist design.

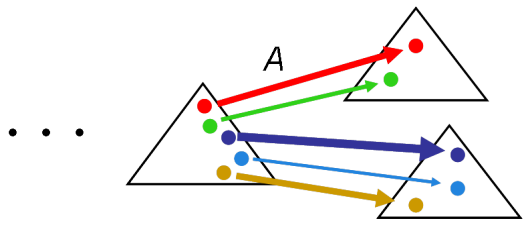

Figure 6.4: Probability of continuing a derivation from treelet t_q to treelet $t_{q'}$ is determined by competition between traces. The width of the arrows indicates the trace activation A.

The probability of moving to $t_{q'}$ in the next step of the derivation is simply the sum of activations of traces that point to $t_{q'}$, divided by the sum of all activations

(see Figure 6.4).[3] Let $E_{t_q}^{t_{q'}}$ be the set of traces in treelet t_q that point to treelet $t_{q'}$, and E_{t_q} the full set of traces in treelet t_q. Then, the probability of moving the derivation to treelet $t_{q'}$ is

$$P_{episodic}(t_{q'}|t_q) = \frac{\sum_{e_i \in E_{t_q}^{t_{q'}}} A(e_i)}{\sum_{e_j \in E_{t_q}} A(e_j)} \tag{6.2}$$

The (episodic) probability of a complete derivation D is given by:

$$P_{episodic}(D = \langle t_0, t_1, \ldots, t_n \rangle) = \prod_{i=1}^{n} P(t_i|t_{i-1}) \tag{6.3}$$

This probability can be computed dynamically, while simultaneously updating the common histories (and activations) of all traces at every step of the derivation. Let t_q and $t_{q'}$ be two successive treelets in the pending derivation d, and let $e' = \langle s, j \rangle$ be a trace stored in $t_{q'}$. Then its CH is updated according to

$$CH(e', d_{q'}) = CH(e, d_q) + 1 \tag{6.4}$$

if there exists a trace $e = \langle s, j-1 \rangle$ in t_q (i.e., a predecessor of e'). Otherwise, $CH(e', d_{q'}) = 0$.

A similar probability model can be derived for the episodic HPN model. There is a complementary role for the *semantic memory* component of HPN (i.e., the metric), namely to provide prior probabilities for transitions between treelets where there is no evidence from previous episodes (i.e., smoothing). One then has

$$P(t_{q',\ s}|t_{q,\ r}) = (1-\lambda) \cdot P_{episodic}(t_{q',\ s}|t_{q,\ r}) + \lambda \cdot P_{semantic}(t_{q',\ s}|t_{q,\ r}) \tag{6.5}$$

For now we focus on the symbolic episodic grammar (with labels), and a full treatment of the episodic-HPN model will be given in Chapter 8.

6.2.4 Smoothing and binarization

In order to obtain a non-zero parse probability for all sentences of the test corpus standard smoothing techniques were performed. Unknown words in the test set were replaced by word classes, which were created from rare words (occurring less than 5 times) in the training set. The word class labels were based on the word's morphology, capitalization, and whether the word occurred at sentence initial position. See [Petrov et al., 2006] for details about the algorithm.

In order to deal with missing productions in the test parse trees, as a first step the rules of the treebank parses were binarized, using horizontal Markovization as

[3] For clarity of notation I have left out the register positions r and s from this point on.

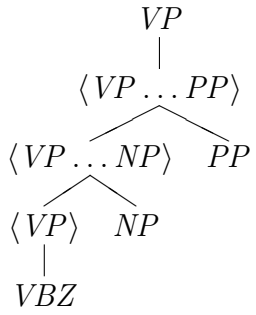

Figure 6.5: Markovization of the tree $VP \rightarrow VBZ\ NP\ PP$ (adapted from [Klein and Manning, 2003])

proposed by [Klein and Manning, 2003]. For any rule with two or more daughters, the right daughters are split off recursively, while the remaining left daughters are replaced with an internal node, as shown in Figure 6.5. In the Figure, the angled brackets (e.g., $< VP \ldots PP >$) indicate internal labels, and the dots summarize all internal labels that expand to VP as their leftmost daughter and PP as their rightmost daughter.

Subsequently, three levels of back-off smoothing (i.e., deleted interpolation) were used, where every level conditioned on less context (see Equation 6.6). The first level back-off probabilities, P_1, backs off to a non-episodic version of the chosen derivation strategy. In the top-down episodic grammar these are the PCFG rule probabilities, which condition the application of a treelet on a single, expanding nonterminal label; In the case of a left corner episodic grammar the first level backs off to a standard probabilistic left corner model. This conditions the application of a treelet on the left corner and goal category, following [Manning and Carpenter, 1997].

The second level, P_2, backs off the conditioning context of any compound nonterminal (originating from the Markovization step) by reducing the conditioning context of a label to its left element alone (e.g. X in $< X \ldots Y >$). Thus, given a unary or binary PCFG rule with a compound root label, e.g., $< X \ldots Z > \rightarrow < X \ldots Y >\ Z$, the backed off probabilities $P(< X \ldots Y >\ Z|X)$ generalize over all such rules with arbitrary Z that have X as the left element of their root nonterminal. Similarly, in the left corner grammar the second level backs off a compound left corner label to its left-most element.

The third level, P_3, assigns uniform probabilities to all possible unary and binary context free productions (that can be constructed from the nonterminals of the grammar), irrespective of context. The three levels are parametrized by back-off parameters λ_1, λ_2 and λ_3, yielding

$$P(t_{q'}|t_q) = (1-\lambda_1) \cdot P_{episodic} + \lambda_1 \cdot ((1-\lambda_2) \cdot P_1 + \lambda_2 \cdot ((1-\lambda_3) \cdot P_2 + \lambda_3 \cdot P_3)) \quad (6.6)$$

In this equation the λ's are fixed, and all back-off probabilities are estimated from the training corpus.

6.2.5 Evaluation and reranking

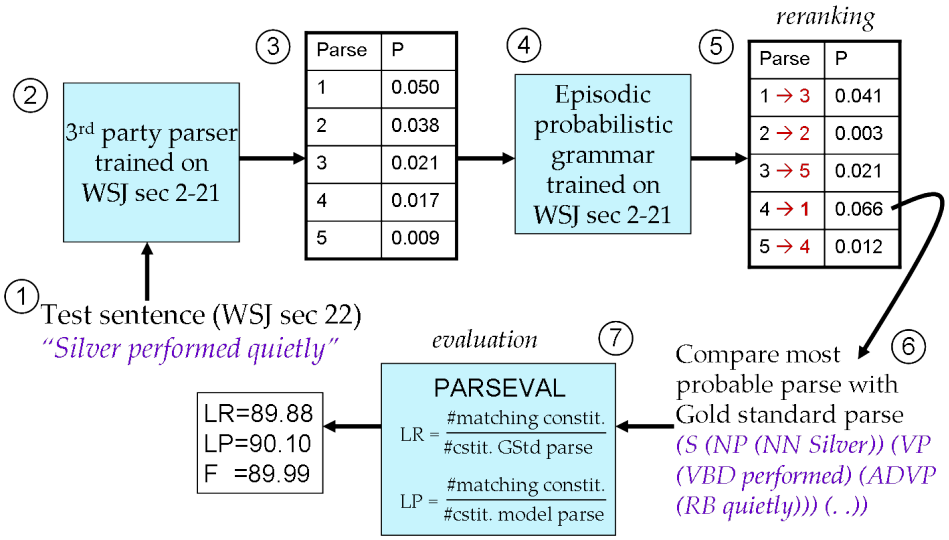

Figure 6.6: The reranking process.

As has become the standard, the episodic grammar was trained on sections 2-21 from the Penn Wall Street Journal corpus (WSJ) [Marcus et al., 1993] and evaluated on section 22 of WSJ. For the test section labeled precision and recall of the most probable parses according to the model were measured using the PARSEVAL metric (see section 3.1.6).

While in Chapter 7 I will develop a specialized left corner chart parser for the episodic grammar, at this stage it is interesting to study the properties of the episodic probability model, and a straight forward way to do this is to use the model as a *reranker*. This means that one takes a list of n best parses for every sentence produced by a third party parser (in this case Charniak's maximum entropy parser [Charniak, 2000]), and reranks the list by assigning a probability to each parse under the model of interest [Sangati et al., 2009]. One can then use the standardized PARSEVAL metric to evaluate labeled precision (LP), labeled recall (LR) and their harmonic mean (F-score) of the parses that receive the highest probability under the reranker [Manning and Schütze, 2000, p. 432]. Figure 6.6 illustrates the reranking process step by step.

Reranking does have some limitations as an assessment of the model's performance, since the n best parses list produced by the third party parser has upper and lower bound precision and recall scores. For comparison the scores are given of a random reranker, that selects a parse from the list by chance. Confidence in

the results of the reranker increases with the size n of the list of the best third party parses (NBest list) (e.g., see Figure 6.8).

6.3 Experiments and results

The precision and recall results of the episodic top-down reranker, applied to the top 5 Charniak parses, are given in the first three columns of Table 6.1 as a function of the maximum common history that is taken into account by the episodic grammar (the column *max his*). CH's larger than the maximum history are capped in equation 6.1. The bottom 2 rows give the Charniak scores and the scores for a random reranker; As is common practice, only sentences of 40 words or less were included. I have experimented with different parameterizations of $\lambda_0, \ldots, \lambda_3$ on the development set . Optimal results were obtained for λ_0=4, and $\lambda_1, \ldots, \lambda_3$ in the range between 0.1-0.3, with only little variance. In Table 6.1 and

	top down reranker			left corner reranker		
max his	LR	LP	F	LR	LP	F
0	87,11	90,01	88,54	87,93	90,31	89,10
1	89,53	90,27	89,90	89,35	90,22	89,79
2	89,64	90,23	89,94	89,49	90,30	89,89
3	90,15	90,45	90,30	89,64	90,43	90,04
4	90,15	90,39	90,27	89,79	90,53	90,16
5	**90,27**	**90,45**	**90,36**	89,91	90,63	90,27
6	90,23	90,41	90,32	89,96	90,58	90,27
7	90,19	90,37	90,28	90,13	90,76	90,44
8	90,09	90,21	90,15	**90,32**	**90,90**	**90,61**
9	90,14	90,27	90,20	90,29	90,84	90,56
10	90,03	90,16	90,09	90,23	90,79	90,51
11	89,98	90,14	90,06	90,10	90,74	90,42
12	89,91	90,11	90,01	90,07	90,67	90,37
Ch	90,23	90,15	90,19	90,23	90,15	90,19
Ran	88,15	87,89	88,02	88,17	87,84	88,00

Table 6.1: Precision and recall scores of the episodic *top-down* reranker (columns 1-3) and *left corner* reranker (columns 4-6) as a function of the maximum history considered ($nBest$=5; λ_0=4; λ_1=λ_2=λ_3=0.2).

Figure 6.7 one can see a clear effect of conditioning history, peaking at history 5 for the top-down reranker, and at history 8 for the left corner reranker (best scores are indicated in boldface). For histories 3-7 the episodic top-down reranker surpasses the Charniak F-scores by a slight margin, and overall does much better than the PCFG reranker (corresponding to *history* 0) and the random reranker.

As can be seen from Table 6.1, the LCE grammar performs better across the board than the TDE grammar, and this is mainly due to improved labeled precision scores. It also does better than the probabilistic left corner model of [Manning and Carpenter, 1997], which corresponds to the top row in the Table. Note that for the LCE reranker the peak is reached at history 8, and the F-scores

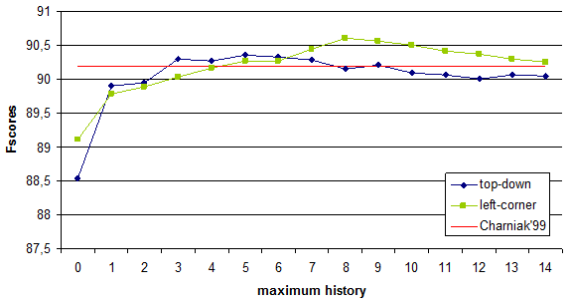

Figure 6.7: F-scores compared between the top-down and the left corner episodic reranker as a function of conditioning history.

stay high until history 14; this could be an indication that the order of conditioning in a LCE derivation better approximates human sentence processing than in a TDE derivation. It is remarkable that the LCE grammar robustly improves on the Charniak parser, because i) unlike the latter it does not implement head annotation or other non-trivial preprocessing steps, ii) it makes several non-standard assumptions about the derivation process, such as a left-corner sequential order and the inclusion of special shift treelets in the derivation for transitions from incomplete productions to words.

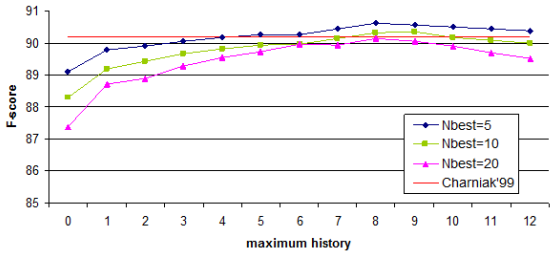

Figure 6.8: F-scores of the left corner episodic reranker applied to the top 5, top 10 and top 20 Charniak parses.

To assess the robustness of the reranking method I have also applied the LCE reranker to the top 10 and the top 20 lists of Charniak parses. In the latter case the random reranker baseline is significantly lower than for the top 5 reranker (F-score = 86.2 resp. 88.0). Therefore it is meaningful that the top 20 reranker still performs almost as good as Charniak (F-score=90.15 for history 8), and the top 10 reranker does even better (F-score=90.34 for history 9). In Figure 6.8 it can further be seen that although the differences in performance between the top 5, top 10 and top 20 reranker are large for low histories, they converge for histories of 6-10, when the episodic approach starts to make a difference. On the other hand, the TDE reranker breaks down when applied to the top 20 Charniak parses, peaking at an F-score of 89.66 for history 6.

6.3.1 Discontiguous episodes

An interesting way to extend the episodic grammar is by including *discontiguous* episodes. Often one can reuse a memorized sentence fragment, even if it does not exactly match the sentence that is currently being processed, but differs from it by a single word or clause. I implemented a variation of update rule for the common history (CH) in order to include episodes with 'gaps'. In Equation 6.4, whenever an episode is interrupted (i.e., its CH is set to 0) it is pushed together with its current activation on an external stack of discontiguous episodes (a separate stack is used for every exemplar). If at a later stage in the derivation a trace of the same exemplar is found, which has no predecessor, then one can pop up an interrupted episode from the top of the stack of that exemplar, and copy (a fixed fraction f of) its activation to the new trace.

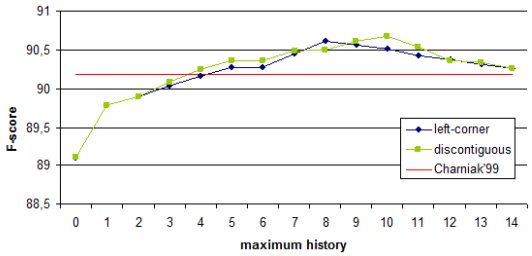

Figure 6.9: F-scores of the LCE reranker with and without counting discontiguities (d=0.95; f=0.6)

.

Best results were obtained when the activation of unused discontiguous episodes decays by some percentage d at every step of the derivation. With $d = 0.95$ and $f = 0.6$ the addition of discontiguous episodes gives a minor improvement over the non-discontiguous case, as can be seen from Figure 6.9. The highest F-score is 90.68, which is reached for history 10. The effect of the inclusion of discontiguous fragments seems to be that longer histories play a more prominent role.

If one looks at the individual sentences from the test set (WSJ section 22) for which the F-scores increased most by including discontiguous fragments, one finds that those are indeed sentences that employ frequent discontiguous expressions. For instance, within the top 5 of these sentences one contains the discontiguous fragment *rose to ... from ...*, which occurs more than 100 times in the training corpus.

6.3.2 Shortest derivation reranker

Assuming that language users understand and produce novel sentences by reusing fragments of stored episodes, then intuitively they will try to do so by retrieving not only the most frequent, but also as few as possible fragments from memory,

since this demands the least cognitive effort. This amounts to a preference for the shortest derivation of a novel sentence.

Such a preference can be implemented in the episodic grammar framework by greedily selecting fragments from stored exemplars that share the largest common history with the derivation of a novel sentence (not including fragments from exemplars that are identical to the novel sentence). When the shortest derivation principle is used together with the LCE reranker to select those derivations of the Nbest list that use the fewest episodes (followed by selection of the derivation with the highest likelihood in case of a draw) then an F-score of 90.44 is obtained (for history 9). Thus, the shortest derivation LCE reranker performs worse than the maximum likelihood LCE reranker, but still better than the Charniak parser.

In Data Oriented Parsing the principle of the shortest derivation has been successfully explored as an alternative to a probabilistic parsing strategy [Bod, 2000]. The multi-word fragments employed in the shortest derivations (or in the most probable derivations) are assumed to have some cognitive reality as the primitive building blocks of speech. In the DOP framework however a top-down derivation is always assumed, whereas in the episodic framework one can also find fragments of a left corner derivation. Figure 6.10 shows some examples of frequent fragments that occur in the shortest derivations of the Tuebingen Corpus of English Spontaneous Speech (www.sfs.uni-tuebingen.de/en/tuebaes.shtml), when the parse trees are derived with a left corner episodic grammar.

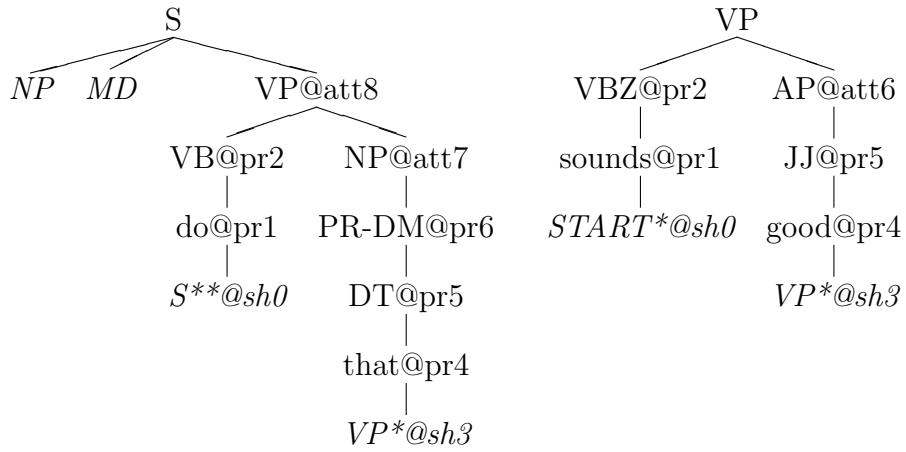

Figure 6.10: Examples of frequent fragments used in the shortest derivations of the Tuebingen corpus. The letters after the @-symbol indicate the applied operation (sh(ift), att(ach), pr(oject)), and the order of application.

6.4 Relation to other work

As was discussed in section 3.1.8, current research in statistical NLP and parsing increasingly focuses on ways to weaken the context independence assumptions of probabilistic context free grammars (PCFGs). Context free grammars fail to take advantage of two relatively independent sources of contextual information for disambiguating between parses: *context!structural and lexical*, which captures the dependency on previous words in the sentence, and *structural context*, which captures the dependency on the relative position in a parse tree. In section 3.1.8 I have discussed some of the solutions that have been investigated, such as *head lexicalization* and *parent annotation*; all of these involve transferring contextual information to the labels of the trees as to preserve the context free backbone of the grammar.

In the episodic grammar both lexical context and structural context are integrated in the conditioning history without any need for preprocessing of the labels. For instance, in the LCE grammar all words to the left of the currently processed word weigh in the parser move decision. As such, the LCE grammar should be considered as a good candidate for language processing.

Parsing with episodic grammars is in some respects comparable to the tradition of *history based parsing*, which exploits the idea that the parser moves are conditioned on n previous parser decisions in the derivation history. A weakness of the latter approach is however that it leads to very large grammars and data sparsity, since all conditioning events are saved explicitly in equivalence classes [e.g., Black et al., 1993, Collins, 1999, p.57]. In the episodic grammar parser decisions are conditioned on arbitrary long histories, at no cost to the grammar size, because conditioning context is implicit in the representation, and is constructed explicitly only during on-line processing of a novel sentence. Since every exemplar is stored only once in the network, the space complexity of the episodic grammar is linear in the number of exemplars.

Another difference with history-based parsers is that in the latter the association between the conditioning event and the sentence from which it originates is lost, whereas in the episodic grammar the identity of an exemplar that has contributed to a derivation step is preserved. In section 6.3.1 it was shown that this feature can be used for including discontiguous episodes.

It is also interesting to compare the episodic grammar with Data Oriented Parsing (DOP) [e.g., Bod, 1998] (see section 3.1.9). In DOP the primitive units of the grammar are not CF rules, but subtrees of arbitrary size, which are extracted from the parses of a treebank. In a certain sense DOP and episodic parsing are complementary: whereas in DOP the substitution of an arbitrary large subtree is conditioned on a single nonterminal, in the episodic parser the application of a local tree is conditioned on an arbitrary large episode. However, the shortest derivation variant of the episodic reranker effectively combines both conditioning on large histories and substitution of stored units larger than a single treelet.

Further, both approaches allow for non-local dependencies to be captured in primitive, discontiguous fragments of the grammar, but in the episodic framework this is less straight forward to implement than in DOP. An advantage of episodic grammar over DOP is that in the former the stored parse tree can be broken down into subtrees according to various generative processes (top-down, left corner, or any other decomposition) whereas in DOP always a top-down generative process is assumed. This opens the possibility to utilize the episodic grammar as a language model in speech recognition, for which a left corner strategy is more suitable than a top-down strategy.

As was mentioned before, in the episodic grammar it is not necessary to store every possible tree fragment explicitly. This is an advantage over DOP, which suffers from computational inefficiency due to very large grammars. The fact that stored episodes are automatically *reconstructed* from traces during the derivation of a novel sentence obviates a time-expensive search through an external memory (i.e., a treebank of fragments), and makes the episodic grammar *content-addressable*.

Table 6.2 shows how the present results compare to state-of-the-art parsers. Note that the latter are evaluated on section 23 of WSJ, while all the results of this work are on section 22. Note also that for the present results a reranker is used, that is parasitic on the Charniak (1999) parser.

Various parser strategies (on WSJ sec 23)

Parsing model	F ($\leq$ 40)	F (all)
Charniak (1999) (max. entropy)	90.1	89.6
Petrov and Klein (2007) (refinement-based)	90.6	90.1
Bansal and Klein (2010) (fragment-based)	88.7	88.1
Sangati & Zuidema (2011) (DOP)	89.7	89.1
Cohn et al. (2009) (Bayesian)	-	84.0
Charniak and Johnson (2005) (reranker, $n = 50$)	-	90.1
This paper (on WSJ sec 22)		
TDE reranker ($n = 5$)	90.4	-
LCE reranker ($n = 5$)	90.6	90.1
LCE + disctg ($n = 5$)	90.7	-

Table 6.2: Comparison of the episodic reranker to state-of-the-art parsers, for sentences of length up to 40, or all sentences.

6.5 Chapter conclusion

In this chapter I described a cognitively inspired implementation for contextual conditioning in statistical parsing, using episodic memory. It was shown that for the task of supervised parsing the episodic grammar is a viable alternative for standard, not cognitively motivated probabilistic grammars. At the same time the episodic grammar offers a neural perspective on human syntax, that unifies

the contrasting views that syntax is either encoded as a set of abstract rules, or as stored exemplars of (fragments of) sentences.

It will be even more interesting to see whether the episodic framework can be successful as an approach to the *unsupervised* induction of (neurally plausible) grammars from unannotated sentences. Since in episodic parsing all computations are done locally, the framework is in principle compatible with the constraints imposed by a connectionist design. This will be explored in Chapter 8, where I will evaluate an episodic version of HPN.

The current work should not only be seen as an exercise in computational linguistics, but also as a theoretical contribution to episodic memory research. As such, it is an instance of how cognitively inspired linguistic research can open a window on the study of memory processes in the brain. I proposed an original hypothesis for the representation of episodic memory, which expresses that an episodic memory is distributed in the form of traces, supplied with a time stamp, *inside* local stores of the semantic memory units that are involved in processing it. According to a free interpretation of this proposal one could imagine episodic memory as a life-long thread spun through semantic memory.

In contemporary theoretical neuroscience most models of episodic memory assume dedicated 'binding neurons', whose sole job it is to bind semantic 'content' nodes into episodic representations [e.g., MacKay, 2007, Shastri, 2002, O'Reilly and Rudy, 2001, O'Reilly and Norman, 2002]. Yet, this is not a very feasible solution for the representation of episodic memories, for every day of a person's life many thousands of new episodic memories are formed. If episodic memories were stored in binding units, this would require the neurogenesis of a massive number of neurons and the establishment of even more new connections. As will be explained in section 8.4.2, in the current proposal successive traces of an episode are assumed to be dynamically bound, hence binding neurons are not necessary. In this sense the current proposal, although simple, contributes to the episodic memory debate, because it shows a way out of the curse of connectivity.

In their essence, the ideas developed in this chapter are consistent with contemporary research in neuroscience, which emphasizes the construal of episodes in the hippocampus as contextually bound sequences of semantic memories [e.g., Eichenbaum, 2004] (see section 8.4 for a discussion of the episodic-HPN model in the neuro-biological context). The hippocampal model of Levy [1996] shows that during episodic sequence learning special 'context neurons' are formed that uniquely identify (part of) an episode. These may function as a neural correlate of the *counter* that was implemented in the traces. The episodic grammar model represents a first attempt to validate this theory of episodic memory within the language domain.